	UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA			
	Documento	Código	Fecha	Revisión
	FORMATO HOJA DE RESUMEN PARA TRABAJO DE GRADO	F-AC-DBL-007	08-07-2021	B
Dependencia	Aprobado		Pág.	
DIVISIÓN DE BIBLIOTECA	SUBDIRECTOR ACADEMICO		(38)	

RESUMEN – TRABAJO DE GRADO

AUTORES	Alvaro Andres Rincon Castro		
FACULTAD	Ingenierías		
PLAN DE ESTUDIOS	Ingeniería de sistemas		
DIRECTOR	Luis Fernando Morales Martínez		
TÍTULO DE LA TESIS	Desarrollo de un sistema de procesamiento de transacciones que mejore el flujo de los depósitos en cuentas gestionadas por terceros relacionados con la organización Wposs		
TITULO EN INGLES	Development of a transaction processing system that improves the flow of deposits in accounts managed by third parties related to the wposs organization		
RESUMEN (70 palabras)			
<p>Este informe describe el proceso de pasantías en la compañía WORLD POS SOLUTIONS, enfocado en el área de desarrollo de aplicaciones para dispositivos POS. Se brindaron nuevas funciones, basados en los requerimientos para la aplicación financiera desarrollada en el IDE Android Studio con lenguaje de programación Java. El informe detalla el proceso de desarrollo de la transacción de depósitos en un periodo de cuatro meses, incluyendo requerimientos, codificación y pruebas.</p>			
RESUMEN EN INGLES			
<p>This report describes the internship process at the company WORLD POS SOLUTIONS, focused on the area of application development for POS devices. New functions were provided, based on the requirements for the financial application developed in the Android Studio IDE with Java programming language. The report details the deposit transaction development process over a four-month period, including requirements, coding, and testing.</p>			
PALABRAS CLAVES	Desarrollo, Transacción, Codificación, Pruebas, Calidad.		
PALABRAS CLAVES EN INGLES	Development, Transaction, Coding, Testing, Quality.		
CARACTERÍSTICAS			
PÁGINAS: 37	PLANOS:	ILUSTRACIONES: 24	CD-ROM:



Desarrollo de un sistema de procesamiento de transacciones que mejore el flujo de los depósitos en cuentas gestionadas por terceros relacionados con la organización wposs

Alvaro Andres Rincon Castro

Facultad de Ingenierías, Universidad Francisco de Paula Santander Ocaña

Ingeniería de Sistemas

Mag. Luis Fernando Morales Martínez

11 mayo del 2023

Índice

Capítulo 1. Desarrollo de un sistema de procesamiento de transacciones que mejore el flujo de los depósitos en cuentas gestionadas por terceros relacionados con la organización WPOSS.	7
1.1. Descripción de la empresa	7
1.1.1. Misión.....	8
1.1.2. Visión	8
1.1.3. Objetivos de la empresa.....	8
1.1.4. Descripción de la estructura organizacional.....	8
1.1.5. Descripción de la dependencia y/o proyecto al que fue asignado	9
1.2 Diagnóstico inicial de la dependencia asignada	10
1.2.1 Planteamiento del Problema	11
1.3 Objetivos de la pasantía	11
1.3.1 General.....	11
1.3.2 Específicos.....	12
1.4 Descripción de las actividades a desarrollar en la misma.....	13
Capítulo 2. Enfoques referenciales	14
2.1 Enfoques conceptuales.....	14
2.1.1 Smart POS Terminal NEW9220	14
2.1.2 Android.....	15
2.1.3 Android Studio	15

2.1.4	Cuadro de dialogo (AlertDialog).....	16
2.1.5	Diseño.....	16
2.1.6	Vista.....	17
2.1.7	Activity.....	17
2.1.8	Checklist.....	18
2.1.9	Git.....	18
2.1.10	Bitbucket.....	19
2.1.11	Sourcetree.....	19
2.1.12	Java.....	20
2.2	Marco legal.....	20
2.2.1	ISO 8583.....	20
Capítulo 3. Informe de cumplimiento de trabajo.....		22
3.1	Analizar los requerimientos proporcionados para la transacción de depósitos con el fin de tener una visión clara del flujo de desarrollo del producto.	23
3.2	Desarrollar un componente funcional para la ejecución de transacciones, utilizando el lenguaje de programación java y el IDE Android Studio, apoyado en normatividades transaccionales ISO-8583 y en patrones de diseño de software para cumplir con los requerimientos del cliente.	26
3.3	Realizar las pruebas al sistema, revisando con ello el flujo del aplicativo y el envío correcto de tramas en su versión de producción.	27
3.4	Documentar e integrar cambios realizados al repositorio principal del aplicativo.....	33
Capítulo 4. Diagnóstico final.....		33

Capítulo 5. Conclusiones	35
Capítulo 6. Recomendaciones	36
Referencias	37

Lista de Figuras

Figura 1 Logo WPOSS	7
Figura 2 Organigrama Organizacional.....	9
Figura 3 Dispositivo NEWPOS9220	14
Figura 4 Logo de Android.....	15
Figura 5 Logo de Android Studio	16
Figura 6 Imagen representativa de cuadros de diálogo básicos.....	16
Figura 7 Ilustración de una jerarquía de vista, que define un diseño de IU.....	17
Figura 8 Logo de git.....	18
Figura 9 Logo de Bitbuckete.....	19
Figura 10 Logo de Sourcetree.....	20
Figura 11 Logo de java	20
Figura 12 Video explicativo de la base bancaria en Android Studio que usan la mayoría de proyectos en WPOSS.....	22
Figura 13 Video explicativo de introducción al estándar para transacciones financieras ISO 8583	22
Figura 14 Daily con el equipo de Desarrollo	24
Figura 15 Muestra de parte del código implementado.....	27
Figura 16 Lista de pruebas internas realizada.....	27
Figura 17 Mensaje de alerta de transacción exitosa.....	29
Figura 18 Mensaje de permiso para capturar la huellas.....	30
Figura 19 Ejemplo vóucher de la transacción depósitos.....	30
Figura 20 Realizando pruebas de lo reportado en QA	30
Figura 21 Ajustes de las incidencias reportadas por QA	31
Figura 22 Mensaje de transacción exitosa cuando finaliza la transacción.....	32
Figura 23 Vóucher de la transacción depósitos con la dirección y ciudad.	32
Figura 24 Evidencia de Tag del número de versión en Sourcetree.....	33

Lista de Tablas

Tabla 1 Matriz DOFA dependencia CMB en WPOSS.....	10
Tabla 2 Descripción de las actividades.....	13

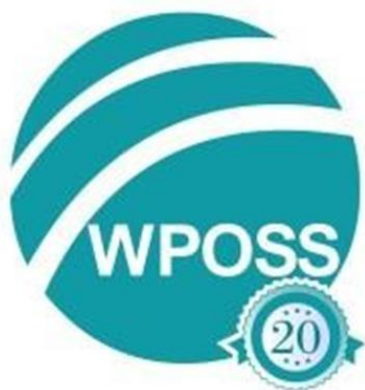
Capítulo 1. Desarrollo de un sistema de procesamiento de transacciones que mejore el flujo de los depósitos en cuentas gestionadas por terceros relacionados con la organización WPOSS.

1.1. Descripción de la empresa

WPOSS es una empresa líder en el desarrollo de soluciones transaccionales, está enfocada en proteger la inversión de sus clientes, garantizándoles equipos seguros y tecnología avanzada que se ajusten a las necesidades de los mismos, brindando seguimiento continuo desde el área comercial y tecnológica, apoyados en un excelente servicio, procesos de certificación, asesoría y gestión de conocimiento. Actualmente WPOSS cuenta con más de 25 años de experiencia en el desarrollo de soluciones de software y venta de dispositivos electrónicos para medios de pago como impresoras, lectores de código de barras y terminales de punto de venta especializadas (POS).

Figura 1

Logo WPOSS



Nota. WPOSS (2022).

1.1.1. Misión

Nuestra misión es ser aliado estratégico de nuestros clientes en la innovación y evolución de modelos de negocios de soluciones integrales, que incluyen paquetes completos de productos tecnológicos y servicios asociados, cumpliendo con los estándares de seguridad, altos niveles de servicio y calidad. (World Pos Solutions, 2021).

1.1.2. Visión

“Convertirnos en la primera opción como socio estratégico y tecnológico de nuestros clientes en el mercado de América Latina.” (World Pos Solutions, 2021)

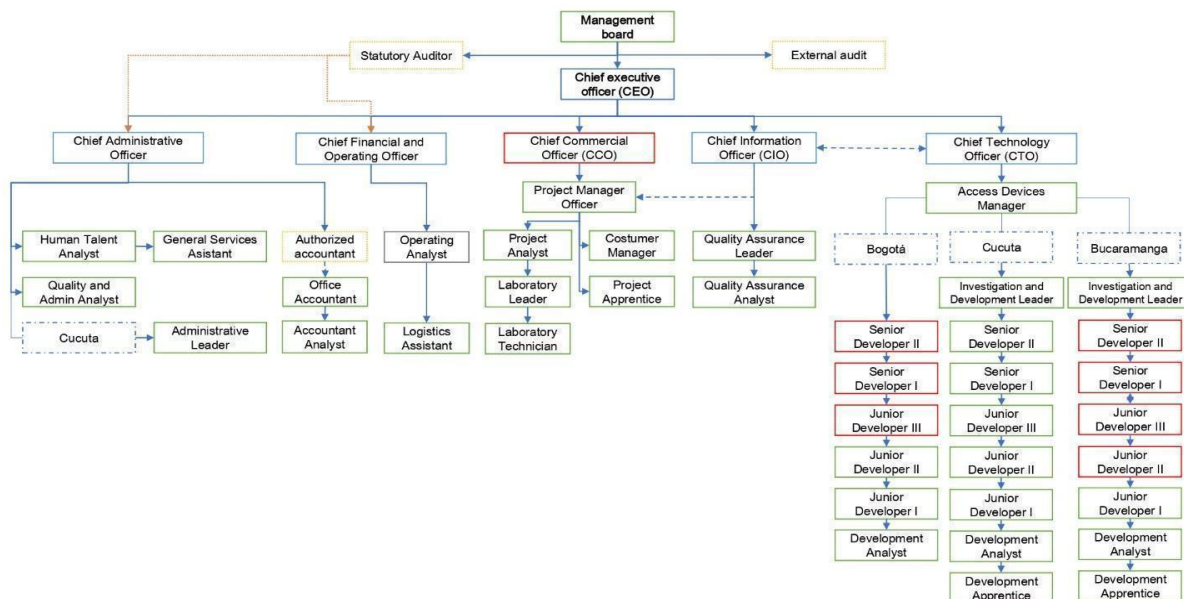
1.1.3. Objetivos de la empresa

- Obtener ventas de 15.000 millones de pesos.
- Tener un centro de desarrollo con altos estándares enfocado a brindar soluciones tecnológicas y de pago.
- Implementar un sistema de gestión de calidad.
- Desarrollar adaptaciones y personalizaciones en sistemas de información y aplicaciones.
- Proveer las mejores soluciones especializadas en los sectores bancarios como transaccionales y de medios de pago, teniendo como visión maximizar la rentabilidad de nuestros clientes. (WPOSS, 2021)

1.1.4. Descripción de la estructura organizacional

Para ofrecer un acercamiento más grande sobre la composición organizacional de la organización WPOSS, a continuación, se muestra la representación de la misma:

Figura 2

Organigrama Organizacional.

Nota. WPOSS (2022).

1.1.5. Descripción de la dependencia y/o proyecto al que fue asignado

Actualmente WPOSS está conformada por diferentes áreas dentro del ámbito de desarrollo y administración de software, entre ellas se encuentra QA (Quality Assurance), UX (User Experience), PMO (Project Management Office) y otras más. El área a la cual se me fue asignado es CMB (Corresponsal Multi Bancario), esta se encarga de desarrollar aplicativos que permiten la administración de servicios y pagos para entidades bancarias como potenciales clientes y aliados de WPOSS, de las cuales, algunas manejan terminales de puntos de venta llamados POS. Los dispositivos anteriormente mencionados son el punto en el cual el área de CMB está enfocada, desarrolla y ofrece soluciones de software para estos dispositivos con el fin de suministrar productos y servicios de calidad, de este modo, el área de CMB se conforma por diferentes células (grupos para cada proyecto), de las cuales cada una tiene su líder, quien toma las riendas de cada proyecto asignado y dos desarrolladores más que completan el equipo.

1.2 Diagnóstico inicial de la dependencia asignada

Tabla 1.

Matriz DOFA dependencia CMB en WPOSS

	Fortalezas (F)	Debilidades (D)
Matriz DOFA	<p>F1: Dentro del área de CMB es posible encontrar personal capacitado para el desarrollo de aplicativos empleando el IDE Android Studio.</p> <p>F2: La metodología SCRUM es la principal herramienta para el trabajo colaborativo entre las diferentes áreas de desarrollo.</p> <p>F3: Cuenta con un buen establecimiento y herramientas de cómputo de última generación.</p>	<p>D1: En su mayoría la documentación es deficiente y/o desactualizada de aquellos cambios que han sido establecidos en anteriores equipos de desarrollo.</p> <p>D2: El tiempo de desarrollo se ve ralentizado debido a que el personal dispuesto para dicha tarea debe rotar entre proyectos nuevos y antiguos.</p>
Oportunidades (O)	FO	DO
<p>O1: Interés por parte de los clientes en adquirir nuevos productos y servicios de la empresa.</p> <p>O2: Capacitaciones disponibles para el personal contratado en las distintas áreas de desarrollo.</p>	<p>F1O1: Desarrollar productos tecnológicos que generen un valor agregado a los modelos de negocio de los clientes y que a su vez aumente la competitividad de la empresa.</p>	<p>D2O2: Contratar a una cantidad mayor de desarrolladores y brindarles capacitación para que puedan trabajar en nuevos proyectos.</p> <p>D1O2: Capacitar al personal con el que cuenta la empresa para que puedan llevar a cabo una adecuada actualización de la documentación existente.</p>
Amenazas (A)	FA	DA
<p>A1: Incidencias encontradas en la aplicación en producción.</p>	<p>F1A1: Tomar apoyo del personal capacitado dentro del área para garantizar un buen desarrollo del producto.</p> <p>F2A1: Hacer buen uso de la metodología scrum para garantizar la agilidad en el desarrollo de los productos.</p>	<p>D1A1: Actualizar la documentación de los proyectos de desarrollo antiguos de la empresa, de manera que estén a la par con los nuevos desarrollos.</p>

1.2.1 Planteamiento del Problema

En la actualidad, es evidente pensar que el mercado de tecnologías en su creciente desarrollo, innova y necesita de una más grande competitividad todos los años, por lo cual es sencillo creer, que este mismo está orientado a muchas organizaciones con diferentes necesidades, como tienen la posibilidad de ser esas empresas que hacen parte del mercado bancario, en donde una enorme proporción de información de alta trascendencia es tratada, y debido a esto, una y otra vez es solicitada la utilización de novedosas funciones o en su defecto resoluciones que se adapten a las necesidades de este sector.

De acuerdo con lo anterior, para la empresa WPOSS es importante mantener a sus clientes satisfechos demostrando su capacidad para atender y mantener sus productos de software en óptimas condiciones; por lo tanto, es así como uno de sus clientes de La Paz, Bolivia solicita los requerimientos de la transacción de depósitos y su flujo transaccional ; es allí donde entra un equipo de desarrollo capaz de ofrecer nuevas funcionalidades y soluciones para este aplicativo, haciendo uso del lenguaje de programación como lo es Java, trabajando sobre el IDE Android Studio.

1.3 Objetivos de la pasantía

1.3.1 General

- Desarrollar un sistema de procesamiento de transacciones, que mejore el flujo de los depósitos en cuentas gestionadas por terceros relacionados con la organización WPOSS.

1.3.2 Específicos

- Analizar los requerimientos proporcionados para la transacción de depósitos con el fin de tener una visión clara del flujo de desarrollo del producto.
- Desarrollar un componente funcional para la ejecución de transacciones, utilizando el lenguaje de programación java y el IDE Android Studio, apoyado en normatividades transaccionales ISO-8583 y en patrones de diseño de software para cumplir con los requerimientos del cliente.
- Realizar las pruebas al sistema, revisando con ello el flujo del aplicativo y el envío correcto de tramas en su versión de producción.

1.4 Descripción de las actividades a desarrollar en la misma

Tabla 2.

Descripción de las actividades

Objetivo General	Objetivos Específicos	Actividades a desarrollar
Implementar un esquema transaccional, que mejore el flujo funcional de depósitos en cuentas gestionadas por un tercero relacionado con la organización WPOSS.	Analizar los requerimientos proporcionados para la transacción de depósitos con el fin de tener una visión clara del flujo de desarrollo del producto.	Hacer análisis de los requerimientos proporcionados.
		Hacer propuestas para el desarrollo del producto.
		Seleccionar la propuesta que se acople al desarrollo del producto.
	Desarrollar un componente funcional para la ejecución de transacciones, utilizando el lenguaje de programación java y el IDE Android Studio, apoyado en normatividades transaccionales ISO-8583 y en patrones de diseño de software para cumplir con los requerimientos del cliente.	Codificación de las transacciones de depósito en la aplicación.
		Realización de modificaciones en el código actual de la aplicación en base a soluciones de eventos.
		Diseño del plan de pruebas.
Realizar las pruebas al sistema, revisando con ello el flujo del aplicativo y el envío correcto de tramases en su versión de producción.	Ajustar las incidencias presentadas en la aplicación.	
	Documentar e integrar cambios realizados al repositorio principal de la aplicación.	

Capítulo 2. Enfoques referenciales

2.1 Enfoques conceptuales

Los puntos principales que se discutirán a continuación son los conceptos de elementos que se consideran importantes en el desarrollo de las actividades propuestas para el cumplimiento de las pasantías.

2.1.1 *Smart POS Terminal NEW9220*

Actualmente los pagos, depósitos, ventas y demás procesos son manejados por los establecimientos bancarios en diversos establecimientos utilizando lo que comúnmente conocemos como datáfono, pero con el uso de las nuevas tecnologías están surgiendo nuevas formas de realizar los procesos antes mencionados de la misma manera; Y es aquí donde entra en escena el terminal punto de venta inteligente denominado NEW9220, un dispositivo táctil que puede ser utilizado como intermediario en las transacciones bancarias, con tecnologías de lectura de tarjetas como cinta, contactless y chip, impresión de recibos y QR o código de barras y todo ello con el sistema operativo para dispositivos móviles denominado Android.

Figura 3

Dispositivo NEWPOS9220



Nota. WPOSS (2022).

2.1.2 *Android*

“Sistema operativo que se emplea en dispositivos móviles, este se basa en Linux un programa libre que, a su vez, está basado en Unix, el objetivo principal de Android es promover los estándares abiertos en teléfonos y ordenadores móviles” (Pérez Porto & Merino , 2015). En la actualidad se puede decir que Android se ha convertido en el sistema operativo más potente, creciente y demandado.

Figura 4

Logo de Android



Nota Android Developers (2023).

2.1.3 *Android Studio*

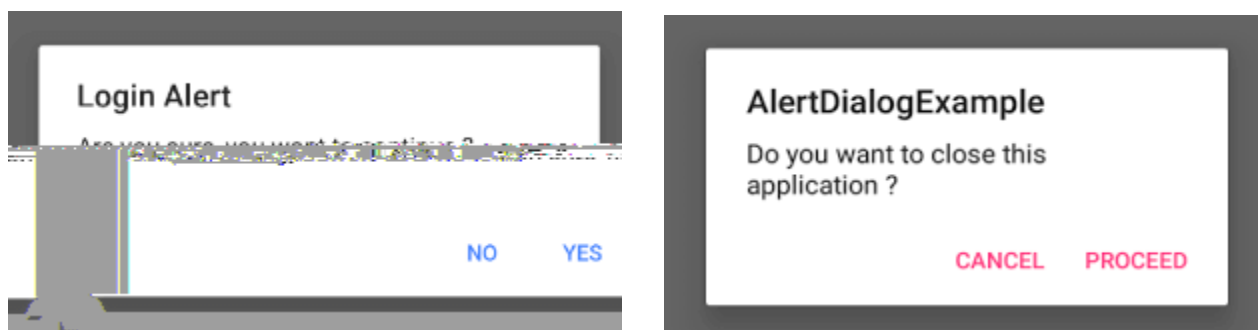
Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android.

(Android Developers, 2023)

Figura 5*Logo de Android Studio**Nota.* Android Developers (2023).

2.1.4 Cuadro de dialogo (*AlertDialog*)

Un diálogo es una ventana pequeña que le indica al usuario que debe tomar una decisión o ingresar información adicional. Un diálogo no ocupa toda la pantalla y, generalmente, se usa para eventos modales que requieren que los usuarios realicen alguna acción para poder continuar. (Android Developers, 2023)

Figura 6.*Imagen representativa de cuadros de diálogo básicos.**Nota* Android Developers (2023).

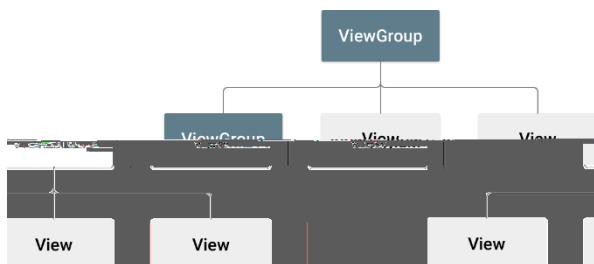
2.1.5 Diseño

Un diseño define la estructura de una interfaz de usuario en tu aplicación, por ejemplo, en una actividad. Todos los elementos del diseño se crean usando una jerarquía de objetos `View` y `ViewGroup`. Una `View` suele mostrar un elemento que el usuario puede ver y con

el que puede interactuar. En cambio, un ViewGroup es un contenedor invisible que define la estructura de diseño de View y otros objetos ViewGroup. (Android Developers, 2023)

Figura 7

Ilustración de una jerarquía de vista, que define un diseño de IU



Nota Android Developers (2023).

2.1.6 Vista

Esta clase representa el bloque de construcción básico para los componentes de la interfaz de usuario. Una Vista ocupa un área rectangular en la pantalla y es responsable del dibujo y manejo de eventos. View es la clase base para los widgets, que se utilizan para crear componentes de interfaz de usuario interactivos (botones, campos de texto, etc.). (Android developers, 2023)

2.1.7 Activity

Una Activity en Android se corresponde con una pantalla de nuestra App. En realidad, es un punto de entrada que Android puede cargar en cualquier momento. Se compone de:
 Una clase, que normalmente extiende de AppCompatActivity. Es donde definimos el código de lo que queremos que haga la App.

Un layout, que identifica la apariencia de la vista, el diseño. Tiene formato XML, pero se puede utilizar el diseñador para hacerlo más sencillo. (Leiva, 2021)

2.1.8 Checklist

Los listados de control, listados de chequeo, checklist u hojas de verificación, siendo formatos generados para realizar actividades repetitivas, controlar el cumplimiento de un listado de requisitos o recolectar datos ordenadamente y de manera sistemática. Se utilizan para hacer comprobaciones sistemáticas de actividades o productos asegurándose de que el trabajador o inspector no se olvida de nada importante. (ISOTools, 2022)

2.1.9 Git

Hoy en día, Git es, con diferencia, el sistema de control de versiones moderno más utilizado del mundo. Git es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente Linus Torvalds, el famoso creador del kernel del sistema operativo Linux, en 2005. Un asombroso número de proyectos de software dependen de Git para el control de versiones, incluidos proyectos comerciales y de código abierto. Los desarrolladores que han trabajado con Git cuentan con una buena representación en la base de talentos disponibles para el desarrollo de software, y este sistema funciona a la perfección en una amplia variedad de sistemas operativos e IDE (entornos de desarrollo integrados). (Atlassian, 2022)

Figura 8

Logo de git



Nota. Atlassian (2022).

2.1.10 Bitbucket

Es una herramienta de administración de repositorios de Git basada en los servidores de Atlassian y accesible a través de una URL normal, lo que significa que es un servidor en la nube accesible desde cualquier dispositivo informático. Bitbucket proporciona un lugar para administrar todo lo relacionado con los repositorios de Git para manejar el código de una manera mucho mejor. Tenga en cuenta que esta herramienta está dirigida a proyectos que usan Mercurial o Git como sus sistemas de control de revisión y desarrolladores de software con piezas de código de propiedad privada que desean revisar sus revisiones de código. (rootstack, 2021).

Figura 9.

Logo de Bitbucket



Nota. Rootstack (2021).

2.1.11 Sourcetree

Sourcetree es un cliente de escritorio de interfaz gráfica de usuario (GUI) gratuito que simplifica la forma en que interactúa con los repositorios de Git para que pueda concentrarse por completo en la codificación. Diga adiós a la línea de comandos: esta GUI facilita la visualización y administración de sus repositorios. También se integra con Mercurial para garantizar un proceso de desarrollo eficiente y consistente. Visualice su trabajo y ejecute comandos push con un nivel de confianza completamente nuevo. (GlobalLogic, 2022.)

Figura 10*Logo de Sourcetree*

Nota. GlobalLogic (2022).

2.1.12 Java.

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

(Oracle, 2022)

Figura 11*Logo de java*

Nota. Oracle (2022).

2.2 Marco legal

Para el desarrollo del proyecto se debe tener en cuenta el siguiente reglamento o norma:

2.2.1 ISO 8583

Por lo general las transacciones con tarjetas de crédito nacen en un punto de venta (POS) o cajero automático (ATM), y viajan hacia la red emisora de la tarjeta de crédito para obtener la autorización de dicha transacción, en el proceso de la transacción viaja

información que se obtiene a partir de la banda magnética de la tarjeta de crédito, como el número de tarjeta, fecha de expiración, y otra información que es agregada dinámicamente por el sistema que procesa la transacción, como el código de terminal, o el valor de la transacción. ISO 8583 define el formato del mensaje a ser enviado y recibido para cada transacción que se realice en el punto de venta, como son compras corrientes y diferidas, anulaciones, consultas y cierres. En general todas las redes que procesan transacciones financieras usan ISO 8583, aunque adaptan campos específicos del ISO de acuerdo a sus necesidades. ISO 8583 también define mensajes entre sistemas para intercambios seguros de claves, conciliación de totales y otros propósitos administrativos.

Un mensaje ISO 8583 consta de las siguientes partes:

- Message Type Indicator (MTI) - Indicador de Tipo de Mensaje.
- Uno o más bitmaps, indicando qué elementos están presentes en el mensaje.
- Data elements, los campos del mensaje. (SOLIS, 2013, pág. 17)

Capítulo 3. Informe de cumplimiento de trabajo

Al comienzo del proceso de pasantía, todos los pasantes reciben una serie de sesiones de capacitación para comprender mejor el modelo de negocios que maneja la empresa. Dicha inducción se realiza en un video instructivo, como se ve en las imágenes a continuación:

Figura 12

Video explicativo de la base bancaria en Android Studio que usan la mayoría de proyectos en WPOSS

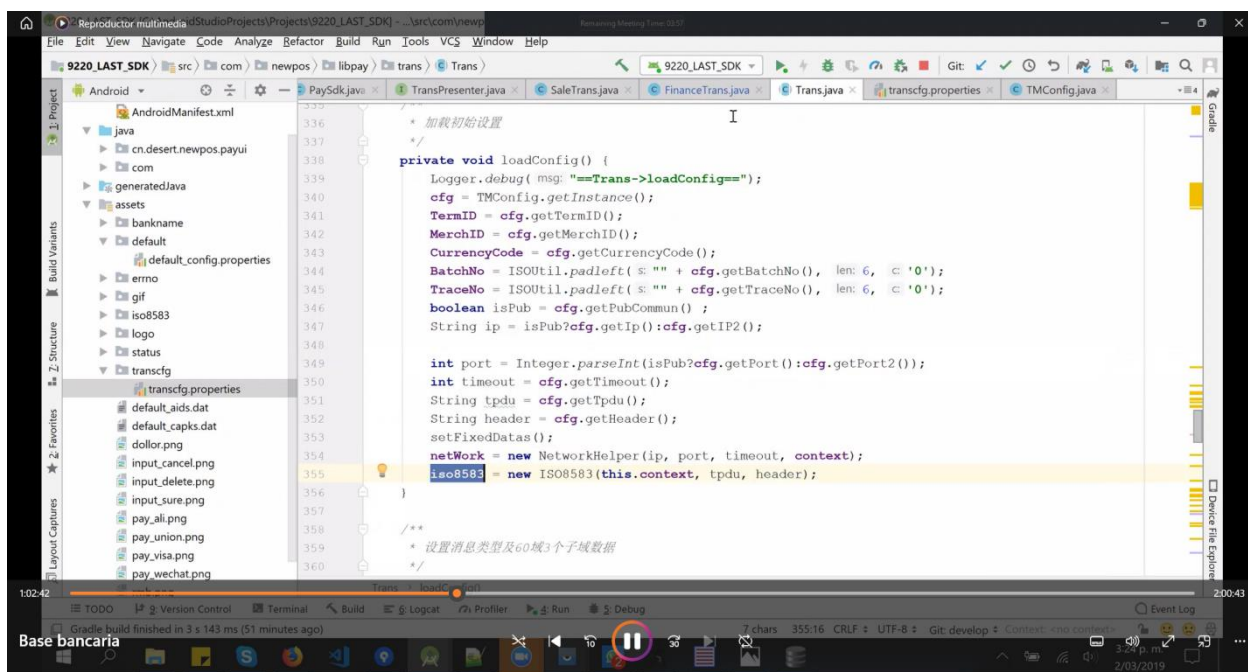
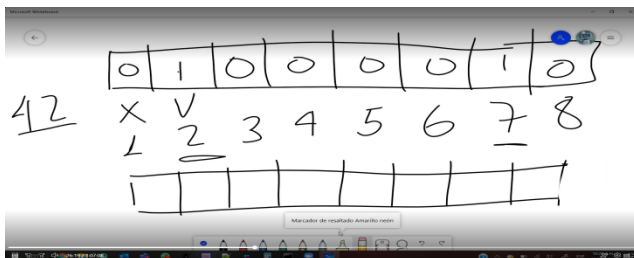


Figura 13

Video explicativo de introducción al estándar para transacciones financieras ISO 8583



Después de completar estas inducciones en el área de CMB, se pasa a lo que WPOSS llama una célula. Esta célula consta de un líder, una mano derecha y una mano izquierda o tres

desarrolladores. Los desarrolladores son responsables de apoyar o implementar nuevos desarrollos de proyectos designados por los clientes (entidades bancarias o empresas involucradas en procesos financieros en varios países).

3.1 Analizar los requerimientos proporcionados para la transacción de depósitos con el fin de tener una visión clara del flujo de desarrollo del producto.

Antes de comenzar el desarrollo, la empresa entrega un documento detallado que describe los requisitos funcionales y no funcionales definidos que deben verificarse. Este consiste en las funcionalidades que debe realizar la aplicación con la nueva transacción (Depósitos). En base a este documento, se realiza un análisis sobre su planteamiento y desarrollo, los cuales se presentan a continuación:

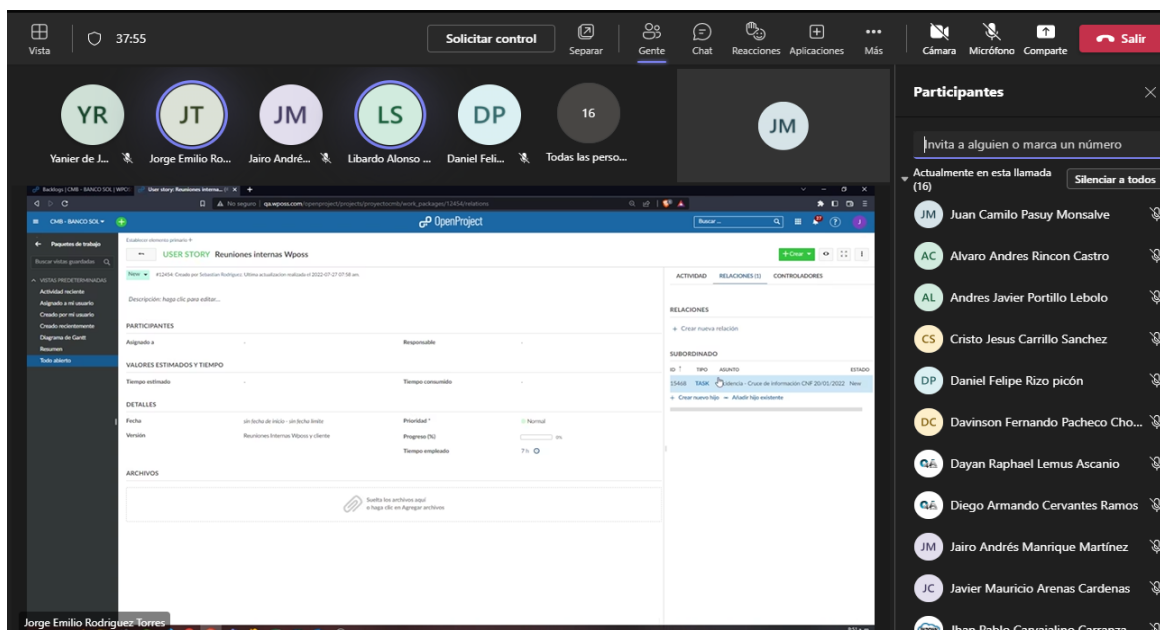
- **Requerimientos funcionales:**

- La aplicación debe permitir el envío de información en formato JSON, y leer y enviar la información a la plataforma.
- Se debe permitir al usuario elegir el modo de depósito, ya sea por huella o por efectivo.
- Si el usuario elige hacer el depósito por huella, se le debe solicitar colocar la huella en el lector del POS y pedirle el monto a depositar.
- Si el usuario elige hacer el depósito por efectivo, se le debe solicitar ingresar un número al cual se va a realizar el depósito y pedirle el monto a depositar.
- Al finalizar la transacción, se debe imprimir un vóucher con toda la información del depósito.

- Se debe permitir al usuario elegir si desea o no imprimir una copia del v ucher.
- Despu es de imprimir el v ucher, la aplicaci n debe mostrar un mensaje de confirmaci n al finalizar la transacci n.
- **Requerimientos no funcionales:**
 - El protocolo utilizado debe ser ISO-8583.
 - El formato JSON se utiliza para la transmisi n de informaci n.
 - El protocolo ISO-8583 y el formato JSON deben garantizar una transacci n segura y eficiente.
 - El uso de huella digital o efectivo no debe generar un retraso significativo en la transacci n.
 - La impresi n del v ucher debe ser clara y legible.
 - El mensaje de confirmaci n al finalizar la transacci n debe ser visible y f cil de entender, “Transacci n exitosa”.

Una vez que se han descrito los requerimientos funcionales y no funcionales de las actividades asignadas, el l der de la c lula convoca a una reuni n con el resto del equipo para establecer qu  desarrollos deben ser priorizados. Durante la reuni n, cada miembro presenta sus ideas y dudas, las cuales son resueltas en conjunto. Si alguna duda persiste, se aborda en las reuniones diarias de seguimiento, que duran aproximadamente 15 minutos. El l der tambi n ofrece apoyo en cuanto a la forma en que se deben abordar las solicitudes del cliente y se inicia la codificaci n de las actividades asignadas. Adem s, se elabora un cronograma de actividades con el tiempo estimado necesario para cada una de ellas.

Figura 14
Daily con el equipo de Desarrollo



Se realizaron varias reuniones para poder llevar a cabo esta tarea, se contó con la participación de todo el equipo, tanto del área de desarrollo móvil, como otras áreas entre esas el área de plataforma y QA, logrando así aterrizar cada propuesta de desarrollo y aclarar las dudas surgidas.

Junto con el cliente también se realizaron varias secciones en video llamada para seleccionar las propuestas y aclarar qué datos se necesitaban en la transacción de depósitos, es especial se miró mucho el flujo de la aplicación en esta transacción.

Después de numerosas reuniones con el equipo de desarrollo y el cliente, y tras una revisión exhaustiva de los requisitos funcionales y no funcionales, se seleccionó la mejor propuesta. En resumen, se decidió que la transacción de depósitos debería contar con dos modalidades de pago: depósito en efectivo y depósito mediante huella dactilar. Se descartó la propuesta que incluía el uso de tarjeta electrónica, así como la que se limitaba a una sola forma de pago. Dado que las necesidades del cliente requerían dos modalidades de pago, y se procedió a su desarrollo.

3.2 Desarrollar un componente funcional para la ejecución de transacciones, utilizando el lenguaje de programación java y el IDE Android Studio, apoyado en normativas transaccionales ISO-8583 y en patrones de diseño de software para cumplir con los requerimientos del cliente.

Luego de levantar las actividades a realizar, se realiza una revisión del código manejado hasta el momento, para luego agregar la nueva transacción sin afectar las demás, cuyo objetivo es presentar un código escalable y estructurado. En su desarrollo se implementaron todos los requisitos analizados en el documento como ISO-8583 para enviar información en los diversos campos que posee, además se utilizó el formato JSON, construido sobre la información del POS como su dirección IP, Mac, batería nivel, también se envió información del depósito como el modo (huella o efectivo) , el monto, fecha, hora y otros datos importantes para realizar correctamente la transacción y se agregaron mensajes de advertencia que ocurren durante las transacciones, todo lo mencionado funciona con el lenguaje de programación Java usando un IDE como Android Studio.

Luego de la implementación de la propuesta, se realiza una revisión para reestructurar el código diseñado por etapas, lo que se hace respecto a esta actividad, por ejemplo, se reutilizan varias clases ya que la aplicación contaba con una transacción (pago de servicios), se utilización los mismos mensajes de alerta utilizados en la transacción pago de servicios, se realizan varias pantallas y se crea el formato del vóucher de acuerdo al de la tracción actual, todo eso codificado con el lenguaje Java.

La aplicación no había implementado previamente esta nueva transacción de depósitos, sólo contaba con la transacción de pago de servicios, una opción de administrativas la cual permite imprimir un vóucher e imprimir un reporte detallado de las transacciones, un login y

envío de logs, actualmente la transacción depósitos ya está incluida y se encuentra en producción.

Figura 15

Muestra de parte del código implementado

```

1 package com.                transactions.depositos;
2
3 import ...
4
15
16 public class Depositos extends TransData implements TransPresenter {
17
18     private String transENamePrincipal;
19
20     public Depositos(Context ctx, String transEName, TransInputPara p) {
21         super(ctx, transEName, p);
22         init(transEName);
23     }
24
25     private void init(String transEName) {
26         transUI = para.getTransUI();
27         isReversal = false;
28         isProcPreTrans = true;
29         isSaveLog = false;
30         isDebit = true;
31         transENamePrincipal = transEName;
32         hostId = ID_LOTE;
33     }
34
35
36     @Override
37     public void start() {
38         Logger.transaction("Inicio de transaccion" + transENamePrincipal);
39
40         if (!checkBatchAndSettle()) {
41             return;
42         }

```

3.3 Realizar las pruebas al sistema, revisando con ello el flujo del aplicativo y el envío correcto de tramas en su versión de producción.

Luego de realizar las actividades previamente planificadas, se realizan pruebas funcionales para observar el comportamiento de las nuevas transacciones junto con las existentes. Estas pruebas se realizan de forma manual, es decir, instalando la aplicación en el terminal (POS) y comprobando el funcionamiento, para esto se creó una lista de pruebas la cual fue realizada por el departamento de desarrollo.

Figura 16

Lista de pruebas internas realizadas

ID	PRUEBA
P01	Instalación manual de la aplicación
P02	Instalación de versión sobre la versión anterior.
P03	Verificar que al reiniciar el POS inicie con la aplicación financiera
P04	Verificación de versión en la pantalla de inicio.
P05	Validar que el pos luego de descargar e instalar remotamente, se reinicie e inicialice la app actualizada.
P06	Validar la correcta descarga remota de la versión.
P07	Verificación del menú principal: <i>PAGO DE SERVICIOS.</i> <i>DEPÓSITOS</i>
P08	Validar que la aplicación permite al usuario elegir el modo de depósito, ya sea por huella o por efectivo.
P09	Validar que al finalizar la transacción, se debe imprimir un voucher con toda la información del depósito.
P10	Se valida que no realice transacciones si no hay papel en la impresora y se muestre el mensaje correspondiente.
P11	Verificar límite mínimo y máximo del Monto de pago.
P12	Verificar cantidad de caracteres mínimos y máximos para cada campo.

En el desarrollo de estas pruebas no fueron al principio todas satisfactorias, todo se basó en prueba y error con cada una de estas, resolviéndolas al instante, por ejemplo una de las pruebas es validar la correcta instalación de la aplicación, validar el mínimo y máximo del monto a depositar, se hicieron en esta varias pruebas, algunas de ellas no satisfactoria, ya que tenía que salir una alerta que dijera si el monto estaba en el rango adecuado, se hizo una análisis mirando esto en tiempo real, logrando solucionarlo y así dando por OK esa prueba.

Al finalizar las pruebas con éxito, el equipo de desarrollo procede hacer entrega de la aplicación al área de QA para que sea certificado, todo esto se mencionara en la siguiente actividad.

Al finalizar las pruebas internas realizadas a la aplicación detallada anteriormente, se genera una APK para ser enviada al área de QA de la empresa. Esto se hace para que el área pueda hacer pruebas más detalladas para proceder a certificar la versión, si se encontró alguna

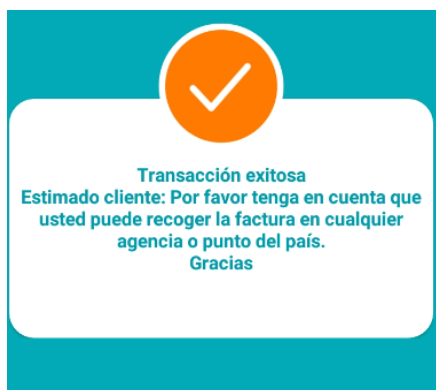
incidencia no se certifica el APK y se informa de lo encontrado al equipo de desarrollo para su respectiva solución.

Luego de recibir las incidencias encontradas por el área de QA, hacemos copias de las mismas para confirmar los errores del equipo de desarrollo, el equipo se adapta a las solicitudes solicitadas y realiza la corrección del código, A continuación, se presentan algunas de las incidencias encontradas por el área de QA.

A nivel de texto y pantalla para depósito en operaciones financieras no aplica factura, por tanto, en el texto final del mensaje de transacción exitosa no aplica la mención a recoger factura, se debe eliminar el mismo.

Figura 17

Mensaje de alerta de transacción exitosa.

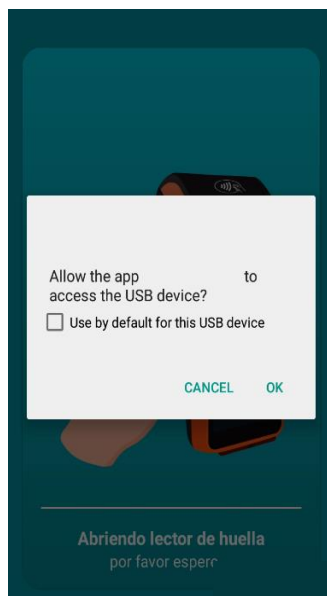


Se ha detectado que, al momento de capturar la huella en la aplicación, aparece un mensaje que ha sido reportado anteriormente en la fase 1. Solo si se da OK al mensaje, se permite continuar con la transacción, lo cual afecta negativamente la experiencia del usuario y ha sido objeto de quejas por parte de los comercios. Al parecer, este problema está relacionado con la capacidad de memoria de la aplicación. Por lo tanto, solicitamos que el mensaje no aparezca durante la ejecución de la transacción y no impida que se realice la

transacción de manera fluida.

Figura 18

Mensaje de permiso para capturar la huella.



Para todos los vouchers, en el encabezado donde se reflejan datos del comercio debería visualizarse dirección y ciudad. Elaboración Propia.

Figura 19

Ejemplo vóucher de la transacción depósitos

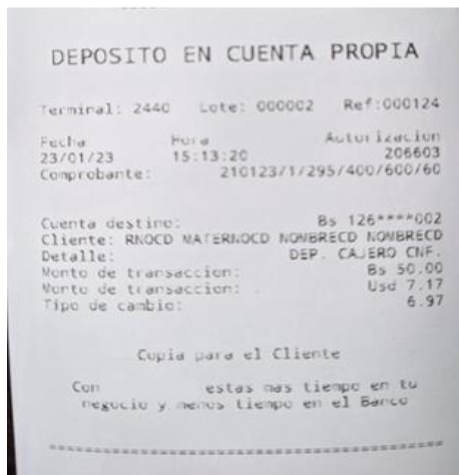


Figura 20

Realizando pruebas de lo reportado en QA.



Posteriormente de haber replicado lo reportado por el área de QA se procede hacer los respectivos ajustes.

Después de reconocer el error, se realizan las actividades necesarias para realizar la corrección. Una vez hecho esto, la nueva versión del APK se enviará nuevamente al área de QA para su posterior certificación.

A continuación, se muestran algunas modificaciones que se realizaron.

Figura 21

Ajustes de las incidencias reportadas por QA.

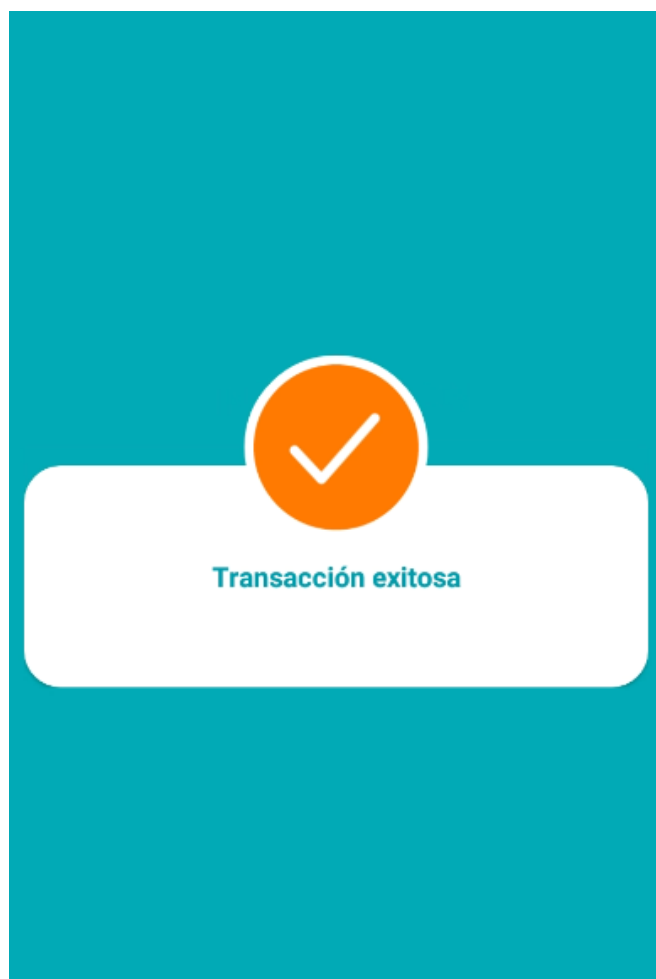
Modificaciones:

- Se baja la calidad de la imagen capturada de la huella de 35 a 25 para poder enviarla en el campo 64 y no exceda el tamaño del campo.
- Se ajusta el envío del campo del tipo de moneda, ya no será Bs. Si no 0 que es boliviano o 101 que sería dólares.
- Se ajusta para qué se muestra si la transacción de depósitos está vigente o no.
- Se ajusta la alerta de finalizar la transacción para que solo muestre la frase "Transacción exitosa".
- Se agrega dirección y ciudad del comercio en los vouchers.
- Se ajusta el modo de impresión por si en medio de la misma la impresora se queda sin papel, aparecerá una vista que pedirá insertar papel y un botón aceptar para volver a imprimir la información del voucher.

A continuación, se presentan algunas evidencias de los ajustes realizados.

Figura 22

Mensaje de transacción exitosa cuando finaliza la transacción.

**Figura 23**

Voucher de la transacción depósitos con la dirección y ciudad.

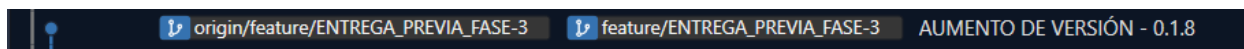


3.4 Documentar e integrar cambios realizados al repositorio principal del aplicativo.

Todos los proyectos dentro de la empresa WPOSS mantienen un repositorio compartido donde se almacenan todos los cambios y mejoras de cada uno. Es decir, después de que se completa cada desarrollo estos cambios se registran en el repositorio del proyecto que se me ha asignado; Usando Sourcetree pude integrar estos cambios para que permanezcan dentro de la estructura general del proyecto alojado en su propio repositorio. Como todo lo anterior, cada proyecto mantiene un documento denominado “release notes”, en el cual se documentan los mismos cambios registrados en el repositorio por la fecha en que se realizaron los cambios y la versión del APK generada para el cliente.

Se utiliza Sourcetree para cargar cada cambio realizado y por supuesto agrega una etiqueta (identificador) de las versiones generadas.

Figura 24
Evidencia de Tag del número de versión en Sourcetree.



Capítulo 4. Diagnóstico final

La experiencia de las pasantías brinda la oportunidad de afianzar y adquirir nuevos

conocimientos y aptitudes en el desarrollo de software para aplicaciones móviles Android, así como en el manejo de herramientas de desarrollo y el trabajo en equipo. Específicamente, en el caso de WORLD POS SOLUTIONS S.A.S (WPOSS), empresa especializada en servicios financieros y transacciones bancarias, se logró comprender con mayor profundidad el mercado relacionado con el desarrollo de software y la importancia de este tipo de servicios para las entidades bancarias.

En particular, como desarrollador, se fue capaz de contribuir significativamente a un proyecto que ofrece servicios financieros, agregando nuevas funcionalidades a la aplicación. El trabajo ha sido de alta calidad y se ha demostrado un nivel suficiente de habilidad y conocimiento en el campo de la programación.

Capítulo 5. Conclusiones

Al analizar detalladamente los requerimientos proporcionados para la transacción de depósitos, los desarrolladores pueden comprender mejor los requisitos del cliente y tener una visión clara del flujo de desarrollo del producto. Esto les permitirá identificar posibles problemas y desafíos, lo que a su vez puede ayudar a mejorar la calidad del producto final y, por ende, la satisfacción del cliente. El análisis de los requerimientos es un aspecto fundamental en el proceso de desarrollo de cualquier producto o servicio, ya que permite a los desarrolladores trabajar de manera efectiva y eficiente para cumplir con las expectativas del cliente y alcanzar el éxito del proyecto.

Al desarrollar un componente funcional para la ejecución de transacciones, utilizando herramientas y normatividades adecuadas como el lenguaje de programación Java, el IDE Android Studio y la normatividad ISO-8583, se puede garantizar la seguridad, eficiencia y calidad del producto final, cumpliendo con los requerimientos del cliente y mejorando la experiencia del usuario. Además, el uso de patrones de diseño de software puede facilitar la evolución y actualización del componente a lo largo del tiempo, lo que es fundamental para el éxito del proyecto y la satisfacción del cliente.

Realizar pruebas exhaustivas al sistema antes de su implementación en producción es esencial para identificar y corregir posibles errores, garantizando así el correcto funcionamiento y flujo del aplicativo en su versión final. Además, revisar el flujo del aplicativo y asegurar el envío correcto de tramas es fundamental para proporcionar una experiencia satisfactoria al usuario y evitar errores.

Capítulo 6. Recomendaciones

Antes de iniciar con cada desarrollo nuevo, se debe estudiar el proyecto en su estado actual y los cambios que ha tenido, saber cada funcionamiento tanto de forma visual como su código fuente, logrando así tener claro el proceso de negocio y poder aplicar nuevas funcionalidades más fácilmente.

Es ideal seguir la metodología SCRUM, se aplica en algunos casos de desarrollo, pero puede que en otros no se aplique del todo ya que se manejan de manera más esporádica.

Se recomienda ofrecer más capacitación a los empleados para mejorar sus habilidades como desarrollador y ser un recurso aún más valioso para la empresa, ya que las inducciones disponibles son útiles hasta cierto punto.

Es imprescindible implementar un proceso de documentación detallado para cada proyecto. Asegurándose de que la documentación sea clara, concisa y fácil de entender, incluso para aquellos que no están familiarizados con el proyecto en cuestión.

Referencias

Android Developers. (2021). Developers.

<https://developer.android.com/studio/intro?hl=es-419>

Atlassian. (2021). Git tutorials & training.

<https://www.atlassian.com/git/tutorials/what-is-git>

GlobalLogic. (2021). Software engineering services & solutions.

<https://www.globallogic.com/offerings/atlassian/products/sourcetree/>

Oracle. (s.f.). ¿Qué es la tecnología Java y para qué la necesito?

https://www.java.com/es/download/help/whatis_java.html

Pérez Porto, J., & Merino, M. (2015). Definición de Android:

<http://definicion.de/android/>

Rootstack. (2021). Bitbucket.

<https://www.rootstack.com/es/technology/bitbucket/>

Solis, C. (2013). Sistema de pagos basado en el estándar ISO 8583 y normas PCI DSS utilizando

lectores de banda magnética desarrollado en Java. (Tesis de grado). Universidad de

Guayaquil, Guayaquil. Obtenido de <http://repositorio.ug.edu.ec/handle/redug/2534>

World Pos Solutions. (2021).

<https://www.wposs.com/>

World Pos Solutions. (2021). Quienes Somos

<https://www.wposs.com/quienes-somos/>