



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Sistemas

**Desarrollo de una aplicación móvil nativa bajo la
metodología Scrum para la ejecución de operaciones
monetarias de empresas en una entidad bancaria**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Daniel Enrique VERGARA CORNELIO

ASESOR

Augusto Parcemon CORTEZ VÁSQUEZ

Lima, Perú

2022



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Vergara, D. (2022). *Desarrollo de una aplicación móvil nativa bajo la metodología Scrum para la ejecución de operaciones monetarias de empresas en una entidad bancaria*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Daniel Enrique Vergara Cornelio
Tipo de documento de identidad	DNI
Número de documento de identidad	72789454
URL de ORCID	https://orcid.org/0000-0003-3671-6329
Datos de asesor	
Nombres y apellidos	Augusto Parcemon Cortez Vasquez
Tipo de documento de identidad	DNI
Número de documento de identidad	08634618
URL de ORCID	https://orcid.org/0000-0002-5188-7962
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Joel Fernando Machado Vicente
Tipo de documento	DNI
Número de documento de identidad	40476778
Miembro del jurado 1	
Nombres y apellidos	César Augusto Alcántara Loayza
Tipo de documento	DNI
Número de documento de identidad	09132297
Datos de investigación	
Línea de investigación	No aplica
Grupo de investigación	No aplica
Agencia de financiamiento	Propio
Ubicación geográfica de la investigación	País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima

	Jr. Carlos Amezaga No. 375 Universidad Nacional Mayor de San Marcos Latitud: -12.0564232 Longitud: -77.0843327
Año o rango de años en que se realizó la investigación	2021
URL de disciplinas OCDE	2.02.04 -- Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
Escuela Profesional de Ingeniería de Sistemas

Acta Virtual de Sustentación
del Trabajo de Suficiencia Profesional

Siendo las 15:45 horas del día 7 de enero del año 2022, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Mg. Machado Vicente Joel Fernando (Presidente), Mg. Alcántara Loayza Cesar Augusto (Miembro) y el Mg. Cortez Vásquez Augusto Parcemon (Miembro Asesor), usando la plataforma Meet (<https://meet.google.com/jjy-yahj-fza>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“DESARROLLO DE UNA APLICACIÓN MOVIL NATIVA BAJO LA METODOLOGÍA SCRUM PARA LA EJECUCIÓN DE LAS OPERACIONES MONETARIAS DE EMPRESAS EN UNA ENTIDAD BANCARIA”**, por el Bachiller **Vergara Cornelio Daniel Enrique**; para obtener el Título Profesional de Ingeniero de Sistemas.

Acto seguido de la exposición del Trabajo de Suficiencia Profesional, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los miembros del Jurado.

El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de **17 DIECISIETE**.

A continuación el Presidente de Jurados el Mg. Machado Vicente Joel Fernando, declara al Bachiller **Ingeniero de Sistemas**.

Siendo las 16:30 horas, se levantó la sesión.

Presidente

Mg. Machado Vicente Joel Fernando

Miembro

Mg. Alcántara Loayza Cesar Augusto

Miembro Asesor

Mg. Cortez Vásquez Augusto Parcemon

Dedicatoria

A mis padres y mis hermanos, por su apoyo incondicional, por sus consejos y el sacrificio que hicieron para que pudiera ser la persona y el profesional que soy hoy.

Gracias!

Agradecimiento

A la Universidad Nacional Mayor de San Marcos, mi alma mater, por darme los conocimientos necesarios para desempeñarme en el ámbito laboral y ser el profesional que soy.

A las empresas que confiaron en mi y apoyaron en mi carrera profesional.

A mi asesor Augusto Cortez Vasquez, por su supervisión y ayuda durante la elaboración de este trabajo.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMATICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

**DESARROLLO DE UNA APLICACIÓN MÓVIL NATIVA BAJO LA
METODOLOGÍA SCRUM PARA LA EJECUCIÓN DE OPERACIONES
MONETARIAS DE EMPRESAS EN UNA ENTIDAD BANCARIA.**

Autor: Vergara Cornelio, Daniel Enrique
Asesor: Cortez Vasquez, Augusto Parcemon
Titulo: Trabajo de Suficiencia Profesional para optar el Título Profesional de Ingeniero de Sistemas.
Fecha: Diciembre 2021

RESUMEN

El presente trabajo de suficiencia profesional describe el desarrollo de una aplicación netamente nativa para dispositivos con sistema operativo iOS para la ejecución de operaciones monetarias de empresas en una entidad bancaria.

Se utilizó como marco de trabajo Scrum ya que nos permite realizar constantes cambios durante la creación de la aplicación. La entidad bancaria utiliza Scrum por sus múltiples beneficios en el desarrollo de software.

La necesidad de desarrollar este proyecto surgió ya que existía una versión en producción poco estable en relación al tiempo de expiración de la sesión del usuario. Esto ayudó a que los usuarios tengan una aplicación más amigable y estable.

Palabras clave: Aplicación Móvil, iOS, SCRUM, Operaciones Monetarias.

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF SYSTEM ENGINEERING AND INFORMATICS
PROFESSIONAL SCHOOL OF SYSTEMS ENGINEERING

**DEVELOPMENT OF A NATIVE MOBILE APPLICATION UNDER THE SCRUM
METHODOLOGY FOR THE EXECUTION OF BUSINESS OPERATIONS IN A
BANKING ENTITY**

Author: Vergara Cornelio, Daniel Enrique
Advisor: Cortez Vasquez, Augusto Parcemon
Title: Professional Sufficiency Work Report for OPT for the Professional
title of System Engineer
Date: December 2021

ABSTRACT

The present work of professional sufficiency describes the development of a purely native application for devices with iOS operating system for the execution of monetary operations of companies in a banking entity.

Scrum was used as a framework since it allows us to make constant changes during the creation of the application. The bank uses Scrum for its multiple benefits in software development. The need to develop this project arises since there was a very stable production version in relation to the expiration time of the user's session. This helped users to have a more friendly and stable application.

Keywords: Mobile Application, iOS, SCRUM, Monetary Operations

Índice General

RESUMEN	VI
ABSTRACT	VII
ÍNDICE DE TABLAS.....	X
ÍNDICE DE FIGURAS.....	XI
INTRODUCCIÓN.....	12
CAPITULO I TRAYECTORIA PROFESIONAL.....	13
CAPITULO II CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA.....	16
2.1. EMPRESA – ACTIVIDAD QUE REALIZA	16
2.2. MISIÓN	16
2.3. VISIÓN.....	16
2.4. ORGANIZACIÓN DE LA EMPRESA.....	16
2.5. ÁREA, CARGO Y FUNCIONES DESEMPEÑADAS	17
2.6. EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN.....	17
CAPITULO III ACTIVIDADES DESARROLLADAS	19
3.1. SITUACIÓN PROBLEMÁTICA	19
3.1.1. Definición del problema.....	19
3.2. SOLUCIÓN	19
3.2.1. Objetivo.....	20
A. OBJETIVO GENERAL.....	20
B. OBJETIVOS ESPECÍFICOS	20
3.2.2. Alcance	20
A. ALCANCE FUNCIONAL	20
B. ALCANCE ORGANIZACIONAL	20

3.2.3. Etapas y metodologías	21
3.2.4. Fundamentos utilizados	23
3.2.5. Implementación de sus áreas de proceso y sus buenas prácticas	30
3.3. EVALUACIÓN.....	37
3.3.1. Evaluación económica	37
3.3.2. Beneficios obtenidos	38
CAPITULO IV REFLEXION CRITICA DE LA EXPERIENCIA	40
CAPITULO V CONCLUSIONES Y RECOMENDACIONES	41
5.1. CONCLUSIONES	41
5.2. RECOMENDACIONES	41
5.3. FUENTES DE INFORMACION	42
5.4. GLOSARIO	43

ÍNDICE DE TABLAS

Tabla 1: Experiencia laboral.....	13
Tabla 2: Formación académica profesional.....	15
Tabla 3: Etapas y actividades del proyecto	21
Tabla 4: Roles definidos para el equipo de desarrollo	30
Tabla 5: Listado de funcionalidades existentes.....	31
Tabla 6: Listado de pantallas para cada funcionalidad existente	31
Tabla 7: Lista de features de la etapa de preparación de la fase creación del proyecto	33
Tabla 8: Épicas y feature de la fase 3.....	34
Tabla 9: Presupuesto del proyecto	37
Tabla 10: Incremento de usuarios y empresas luego de 6 meses de la publicación	38
Tabla 11: Incremento de la cantidad de ejecuciones de operaciones monetarias luego de 6 meses de la publicación	38

ÍNDICE DE FIGURAS

Figura 1: Organigrama de la entidad bancaria	17
Figura 2: Definición de las capas de acuerdo a Clean Architecture	24
Figura 3: Comunicación entre capas de acuerdo a Clean Architecture	25
Figura 4: Distribución de los components VIPER mediante el concepto Clean Architecture	33

INTRODUCCIÓN

El presente trabajo de suficiencia profesional describe el desarrollo de una aplicación nativa para la ejecución de operaciones monetarias de empresas en una entidad bancaria.

Esta necesidad surgió debido a que la versión en producción de la aplicación para la ejecución de operaciones monetarias de empresas era inestable en relación al tiempo de duración de la sesión del usuario. Esto generó una gran insatisfacción en el usuario llegando a tener comentarios negativos en la tienda de descarga de aplicaciones móviles de Apple.

El trabajo de suficiencia profesional está estructurado de la siguiente manera:

En el CAPITULO I, se describe la trayectoria profesional del autor y las funciones que desempeñó en cada empresa.

En el CAPITULO II, se realiza una breve descripción de la empresa donde el autor trabajó durante el desarrollo del proyecto.

En el CAPITULO III, se detalla el problema y la solución, teniendo en cuenta los objetivos y el alcance.

En el CAPITULO IV, se detallará el aporte del autor en el informe, además de una reflexión crítica de la experiencia.

En el CAPITULO V, se detallarán las conclusiones y recomendaciones.

CAPITULO I

TRAYECTORIA PROFESIONAL

El autor del presente trabajo es bachiller en Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos, con una experiencia laboral de más de 3 años en desarrollo de aplicaciones móviles para dispositivos iPhone. El autor del presente trabajo se desempeña como Sub-Gerente Adjunto de Análisis de Aplicaciones en el Banco de Crédito del Perú, brindando apoyo a la empresa en la programación y mantenimiento de aplicaciones móviles, seguimientos de buenas prácticas y lineamientos para el desarrollo de aplicaciones para dispositivos con sistema operativo iOS.

A continuación se mostrará la tabla relacionada a la experiencia laboral.

Tabla 1: Experiencia laboral

Banco de Crédito del Peru	
Marzo 2020 – Actualidad	
Cargo	Sub Gerente Adjunto de Análisis de Aplicaciones
Proyecto	Telecrédito
Funciones	- Mantener las buenas prácticas de programación de aplicaciones para dispositivos iPhone. - Fomentar la participación en grupos de conocimiento. - Seguir los lineamientos de programación de la empresa, - Coordinar la actualización de la aplicación en el Appstore con las áreas involucradas. - Mantenimiento de la aplicación. - Coordinar con diseño la implementación de nuevas funcionalidades. - Coordinar con el equipo la priorización de nuevas funcionalidades.
Logros	- Llegar a tener 3 sesiones del grupo de conocimiento en un mes, aumentando la cantidad de participantes en un 50%. - Implementar el rediseño de la bandeja en la aplicación.
Everis	
Diciembre 2019 – Marzo 2020	

Cargo	Solution Analyst
Cliente	Claro
Proyecto	Mi Claro
Funciones	- Mantenimiento de la aplicación para dispositivos iPhone. - Coordinar con el equipo y el cliente la priorización de las nuevas funcionalidades a implementar. - Coordinar con el area de Análisis la implementación de Firebase Analytics.
Logros	- Implementar Firebase Analytics en el 80% de la aplicación.

Everis

Noviembre 2018 – Diciembre 2019

Cargo	Solution Analyst
Cliente	Interbank
Proyecto	Tunki
Funciones	- Mantenimiento de la aplicación para dispositivos iPhone. - Implementar nuevas funcionalidades. - Seguimiento y ejecución de la actualización de la aplicación en el Appstore. - Coordinar con el área de diseño las nuevas funcionalidades que se desean implementar. - Coordinar con el equipo y el cliente la priorización de las nuevas funcionalidades a implementar.
Logros	- Actualizar la aplicación en el Appstore cada semana. - Implementar el pago de persona a persona con QR. - Modificar el diseño de las pantallas de la seccion “Mi perfil”.

Accenture

Junio 2017 – Noviembre 2018

Cargo	Associate
Cliente	BBVA

Proyecto	BBVA Peru
Funciones	- Mantenimiento de la aplicación para dispositivos iPhone. - Implementar nuevas funcionalidades. - Coordinar con diseño las nuevas funcionalidades que se desean implementar. - Coordinar con el área de Análisis la implementación de Adobe Analytics.
Logros	- Implementación de la funcionalidad “Mi día a día”. - Implementación de la funcionalidad “Préstamo vehicular”. - Implementación de la funcionalidad “Operaciones frecuentes”. - Implementación de la funcionalidad “Apertura de fondos mutuos”. - Implementar Adobe Analytics en el 80% de la aplicación.

Nota: Elaboración propia

A continuación se mostrará la formación académica profesional.

Tabla 2: Formación académica profesional

Formación recibida	Grado Académico de Bachiller en Ingeniería de Sistemas – Escuela Académica Profesional de Ingeniería de Sistemas – Facultad de Ingeniería de Sistemas e Informática
Institución que Acredita	Universidad Nacional Mayor de San Marcos
Documento de Acreditación	Grado Académico – Bachiller
Período	2012 - 2017

Nota: Elaboración propia

CAPITULO II

CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA

2.1. Empresa – actividad que realiza

La entidad bancaria viene liderando el sistema financiero a nivel nacional desde hace 130 años, contribuyendo con el desarrollo económico del país. Actualmente cuenta con una participación de más del 40% en créditos corporativos. Así mismo, la entidad bancaria ofrece servicios de gestión de activos, de transacciones de divisas, de tesorería, de custodia, de asesoramiento de inversión y actividades de investigación financiera.

Al tener una diversidad de negocios y áreas, permite ofrecer a sus trabajadores múltiples experiencias que mejoren sus habilidades y competencias.

2.2. Misión

De acuerdo a la misión de la entidad bancaria se manifiesta lo siguiente: Estar siempre al lado de los ciudadanos transformando sus sueños y planes en realidad con el objetivo de contribuir a la historia del Perú en su desarrollo y superación.

2.3. Visión

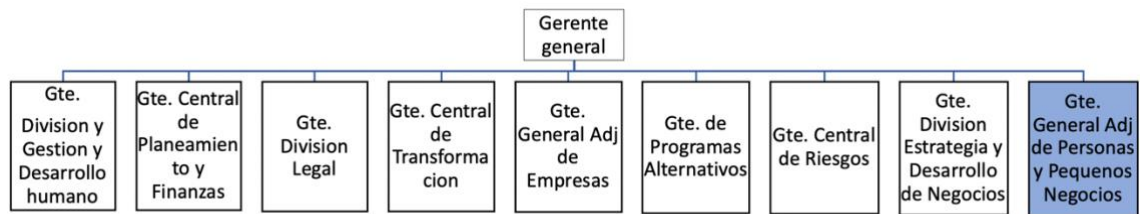
De acuerdo a la visión de la entidad bancaria, nos manifiesta lo siguiente:

- Ser la empresa peruana que brinda la mejor experiencia a los clientes. Simple, cercana y oportuna.
- Ser la comunidad laboral de preferencia en el Perú, que inspira potencia y dinamiza a los mejores profesionales.
- Ser referentes regionales en la gestión empresarial potenciando su liderazgo histórico y transformador de la industria financiera en el Perú.

2.4. Organización de la empresa

A continuación, se presenta el organigrama de la entidad bancaria:

Figura 1: Organigrama de la entidad bancaria



Nota: Elaboración propia

2.5. Área, cargo y funciones desempeñadas

El autor de este informe de experiencia profesional se desempeña como Sub Gerente Adjunto de Analisis de Aplicaciones dentro de la Gerencia General Adjunta de Personas y Pequeños Negocios, desde el 15 de marzo del 2020 hasta la actualidad.

Las funciones que presenta el autor están alineados a lo siguiente:

- Mantener las buenas prácticas de programación en aplicaciones móviles para dispositivos iPhone.
- Fomentar la participación en grupos de conocimiento.
- Seguir los lineamientos de programación de la empresa,
- Coordinar la actualización de aplicaciones móviles en el Appstore con las áreas involucradas.
- Mantenimiento de las aplicaciones móviles para dispositivos con sistema operativo iOS.
- Coordinar con el área de diseño la implementación de nuevas funcionalidades.
- Coordinar con el equipo de desarrollo la priorización de nuevas funcionalidades.

2.6. Experiencia profesional realizada en la organización

El autor del presente trabajo tuvo las siguientes experiencias dentro de la organización actual en la que se desempeña:

- Analizar, diseñar y desarrollar aplicaciones móviles para dispositivos con sistema operativo iOS.
- Participar en reuniones de priorización de funcionalidades con el negocio.
- Coordinar con otras áreas de acuerdo a las dependencias que la funcionalidad demande.

CAPITULO III

ACTIVIDADES DESARROLLADAS

3.1. Situación problemática

La entidad bancaria desarrolló una aplicación móvil para que las empresas puedan ejecutar sus operaciones monetarias, en su primera versión tuvo la funcionalidad de autenticación, siendo esta nativa. Adicionalmente el resto de funcionalidades eran accedidas por el usuario mediante un CWR (Componente Web Responsive).

En la primera versión se mostró mensajes a los usuarios indicando que su sesión había expirado luego de veinte a ciento veinte segundos de haberse logueado. Mediante un análisis correspondiente se concluyó que este problema se manifestaba cuando existía una comunicación desde CWR hacia los servicios Rest. Para dicha comunicación se necesitaba enviar un certificado de autenticación mutua para la seguridad de la transferencia de datos. El problema radicaba en que el CWR identificaba al certificado como expirado y esto provocaba que se mostrara el mensaje en mención, lo cual generaba una incomodidad ya que el usuario no podía culminar la ejecución de sus operaciones monetarias.

Una alternativa fue no utilizar el certificado de autenticación mutua para la comunicación entre la CWR y los servicios Rest, pero generaría una falla de seguridad en la transferencia de datos permitiendo que personas inescrupulosas puedan acceder a ellas. Riesgo que la empresa no estuvo dispuesta a asumir.

3.1.1. Definición del problema

El tiempo de expiración de la sesión del usuario era muy corta provocando que los usuarios no puedan culminar la ejecución de las operaciones monetarias generando una gran insatisfacción con la aplicación.

3.2. Solución

Para evitar la falla de seguridad en la transferencia de datos dentro de la aplicación se tuvo que considerar el uso del certificado de autenticación mutua y para evitar que el tiempo de expiración de la sesión del usuario sea muy corta se

dejó de usar el CWR, por lo que se propuso desarrollar una aplicación completamente nativa ya que de esta manera el certificado es detectado correctamente.

A continuación se detallará los objetivos y el alcance del proyecto.

3.2.1. Objetivo

a. Objetivo general

Desarrollar una aplicación móvil nativa bajo la metodología scrum para la ejecución de operaciones monetarias de empresas en una entidad bancaria.

b. Objetivos específicos

- Realizar un análisis de las funcionalidades que tiene la aplicación en producción para implementarla en la aplicación nativa.
- Aplicar metodología SCRUM utilizando las técnicas, herramientas y buenas prácticas para el beneficio de los involucrados y el producto.
- Definir la arquitectura con la que se desarrollará la aplicación.

3.2.2. Alcance

a. Alcance funcional

El presente trabajo tiene como alcance funcional el desarrollo de una aplicación para dispositivos móviles con sistema operativo iOS de una entidad bancaria considerando las siguientes funcionalidades:

- Autenticación
- Selección de empresa
- Perfil de usuario
- Cuenta
- Historial de movimiento
- Operaciones monetarias pendientes de firma
- Operaciones monetarias pendientes de envío

b. Alcance organizacional

El alcance organizacional para este proyecto involucró al área de banca mayorista con participación de las siguientes áreas:

- De diseño
- De experiencia de usuario
- De arquitectura de software
- De seguridad
- Devsecops.

3.2.3. Etapas y metodologías

El desarrollo de la aplicación nativa para la ejecución de operaciones monetarias de empresa en una entidad bancaria se realizó en cuatro fases: análisis de funcionalidades existentes, creación del proyecto, gestión de requerimientos y publicación del producto. En cada una de las fases mencionadas se considerará las etapas respectivas como se muestra en la tabla 3, mencionando las actividades por cada etapa.

Tabla 3: Etapas y actividades del proyecto

FASE	ETAPA	ACTIVIDADES
Análisis de funcionalidades existentes (Fase 1)	Análisis	Realizar un listado de todas las funcionalidades existentes de la aplicación en producción.
		Revisar del flujo de pantallas para cada funcionalidad existente con el área de diseño.
Creación del proyecto (Fase 2)	Definición	Definir la separación de la lógica.
		Definir el patrón de arquitectura de la aplicación
	Preparación	Crear del proyecto en el ambiente Apple Creación del proyecto en Xcode

Gestión de los requerimientos (Fase 3)	Planificación y estimación	Crear el backlog priorizado del producto
		Crear Épicas / Historias de usuario
		Estimar historias de usuario
		Identificar las tareas por historia
		Crear Sprint Backlog
	Implementación	Realizar reuniones daily
		Implementar las historias de usuario
		Realizar pruebas de humo
		Realizar reunión de refinamiento.
		Realizar demos
Publicación del producto (Fase 4)	Publicación	Realizar pruebas de certificación
		Realizar pruebas de seguridad
		Realizar pase a producción

Nota: Elaboración propia

Para mayor detalle de las fases y actividades vea la sección 3.6.

El presente proyecto fue desarrollado en 16 sprint, donde cada sprint tiene una duración de dos semanas.

Para cada fase los participantes cumplieron con los siguientes roles:

- Product Owner: encargado de gestionar el valor del producto mediante reuniones de sprint planning y en el sprint review donde los requerimientos son revisado para su aceptación.
- Scrum Master: encargado de gestionar las reuniones scrum que viene a ser daily, sprint planning, sprint review y retrospective. Además de ello se encarga de apoyar al equipo de desarrollo en la eliminación de impedimentos.
- Developer: son los encargados de desarrollar el producto mediante la responsabilidad y cumplir con las historias asignadas en el sprint.

3.2.4. Fundamentos utilizados

Operaciones monetarias

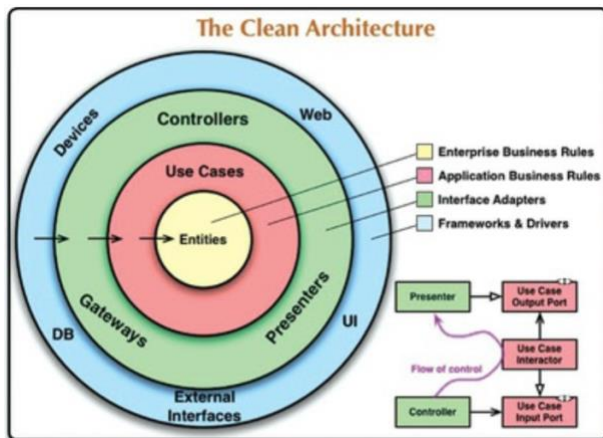
Según (Gray & Talbot, 2006) “las operaciones monetarias se refieren a la implementación de una política monetaria que garantice que la decisión política tenga el efecto deseado en los mercados financieros y en la economía en general. Bajo este amplio espectro, las autoridades monetarias necesitan decidir qué metas específicas deben tener como objetivo, y cuáles instrumentos de política deben utilizar. Para la mayoría de los países, el objetivo monetario final a largo plazo del banco central es mantener una inflación baja y estable. Para propósitos operacionales, sin embargo, la meta táctica día a día es, por lo general, lograr un nivel particular de tasas de interés o de tipo de cambio”.

Para el presente informe, las operaciones monetarias forman parte del flujo de dinero del proyecto en desarrollo que involucra transferencias, pago de planillas, tarjetas de crédito, etc.

Clean architecture

Según (Martin, 2017) “La arquitectura limpia es un diseño de software que separa los elementos de un diseño en niveles de anillo. Un objetivo importante de la arquitectura limpia es proporcionar a los desarrolladores una forma de organizar el código de tal manera que encapsule la lógica empresarial pero la mantenga separada del mecanismo de entrega.”

Figura 2: Definición de las capas de acuerdo a Clean Architecture



Nota: Fuente de (Martin, 2017)

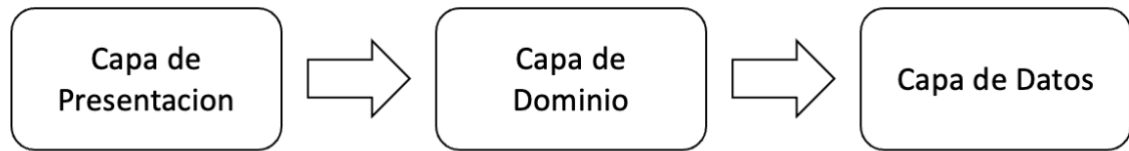
En el desarrollo del presente proyecto se toma el concepto de Clean Architecture para definir la arquitectura del software. De acuerdo a la referencia anterior, para el presente proyecto, se va a considerar como capas el concepto de anillos.

Clean Architecture nos ayuda a separar la lógica en tres capas:

- Capa de presentación: se implementa la vista y la lógica requerida para que los componentes visuales se comuniquen.
- Capa de Dominio: se implementan los casos de uso o las reglas de negocio que serán comunicados a la capa de presentación.
- Capa de Datos: se implementa las entidades necesarias para la interacción con los servicios Rest. Además de las diferentes formas de almacenamiento de data.

Como lo menciona en la Figura 3 la comunicación entre capas debe ser de afuera hacia adentro. Considerando que la capa exterior es la capa de presentación, la capa intermedia es la capa de dominio y la capa interna es la capa de datos.

Figura 3: Comunicación entre capas de acuerdo a Clean Architecture



Nota: Elaboracion propia

Esto quiere decir que la capa de presentación conoce la capa de dominio pero la capa de dominio no conoce la capa de presentación. Esta misma lógica se aplica entre la capa de dominio y la capa de datos, ya que la capa de dominio conoce la capa de datos pero la capa de datos no conoce la capa de dominio.

“A diferencia de una propuesta de modelo de arquitectura, Clean Architecture es mas bien un concepto. Si bien es cierto que especifica las áreas de las cuales debe estar compuesto el desarrollo, así como la relación entre ellas, no es un modelo de arquitectura cerrado y tampoco define realmente unas reglas de implementación, sino que debe ser diseñado específicamente para un desarrollo concreto.” (Nieto Sanchez, 2017).

Patrón de arquitectura de software

Según (Butani, 2020) “el patrón de arquitectura de software captura las estructuras de diseño de varios sistemas y elementos de software para que puedan reutilizarse. Durante el proceso de escribir código de software, los desarrolladores encuentran problemas similares varias veces dentro de un proyecto, dentro de la empresa y dentro de sus carreras. Una forma de abordar esto es crear patrones de diseño que brinden a los ingenieros una forma reutilizable de resolver estos problemas, lo que permite a los ingenieros de software lograr el mismo resultado estructural para un proyecto determinado”. Además, realiza un análisis de la importancia del patrón de arquitectura de software para el ingeniero de software y para el cliente. Desde la perspectiva de un ingeniero, según (Butani, 2020), “los patrones de arquitectura de software son importantes porque impulsan la eficiencia y la productividad. Los desarrolladores

pueden unirse a un proyecto existente en cualquier momento con una incorporación limitada, dado que ya comprenden el patrón de arquitectura utilizado en el proyecto. También se pueden agregar nuevas funcionalidades al proyecto sin ninguna dificultad, y los problemas comunes de las aplicaciones se pueden resolver fácilmente”. Desde la perspectiva del cliente, según (Butani, 2020) “los patrones de arquitectura optimizan los costos de desarrollo, aceleran el cronograma del proyecto y permiten al ingeniero entregar un producto de alta calidad. Las estimaciones de costos se basan en la comprensión de los sistemas de arquitectura, por lo que los gerentes de producto tienen un costo del proyecto más preciso, lo que permite una planificación y un presupuesto temprano. Además, una arquitectura claramente definida significa que el sistema ha sido validado y delimitado. Esto ayuda a los ingenieros a concentrarse en los aspectos esenciales del producto y también permite a los gerentes planificar adecuadamente la finalización del proyecto.”

Según (Butani, 2020) los patrones de arquitectura más esenciales para el desarrollo de una aplicación móvil en iOS son:

MVC: “considera que hay tres tipos de objetos: modelo, vista y controlador. El objeto modelo encapsula los datos específicos de una aplicación y definen la lógica y el cálculo que manipulan y procesan esos datos. El objeto vista es un objeto con el propósito de mostrar datos que el usuario final puede ver. El controlador tiene como propósito actuar como intermediario entre uno o más objetos de la vista y el modelo. Muchos objetos en esta aplicación tienden a ser mas reutilizables y sus interfaces tienden a estar mejor definidas.” (Butani, 2020)

MVP: “tiene seis componentes, que son una vista, modelo, comandos, selecciones, presentador e interactor. El elemento modelo indica un dato en la aplicación y comparte un concepto similar en MVC. El componente secciones especifica la parte de los datos a operar. El componente comandos entrega acciones que podrían ejecutarse. El componente presentador tiene como objetivo organizar y coordinar al igual que el controlador en MVC. El componente vista

representa una vista, que es similar en MVC. Aunque en el MVP original utiliza solo tres componentes, que son Modelo, Vista y Presentador. Por lo tanto, los comandos, las selecciones y el interactor están incluidos en Presenter.” (Butani, 2020)

MVVM: “es un patrón arquitectónico más comunmente utilizado en aplicaciones de Windows. El modelo son los datos o la lógica empresarial, completamente independiente de la interfaz de usuario que almacena el estado y realiza el procesamiento del dominio del problema. La vista consta de los elementos visuales que muestran los datos vinculados desde el modelo de vista. Vista-Modelo es la interacción entre el modelo y la vista. Los datos transmitidos también administran la lógica y el comportamiento de la vista.” (Butani, 2020)

VIPER: “La palabra VIPER es un acrónimo de View, Interactor, Presenter, Entity y Routing. El componente View nos permite mostrar la interfaz y los datos del usuario que irán hacia el presenter. Interactor se centra en la lógica empresarial como por caso de uso. El componente Presenter recibe un dato del interactor que muestra la preparación del contenido con lógica de visualización, solicitando nuevos datos del interactor, que dan reacción a las entradas del usuario. El interactor, como el componente Modelo, utiliza Entity como objetos de modelo primario. Routing contiene la lógica de navegación para describir que pantallas se muestran en que orden.” (Butani, 2020)

Adicional a ello es importante mencionar que los patrones de arquitectura “siempre han ayudado a construir un rendimiento de software comprobable, manejable y optimizado. Por lo general, ayuda a modularizar el software para que cada componente este separado y maneje una sola responsabilidad. También, mejoran significativamente la usabilidad de un código, que realiza una función crítica en la conexión del software. El proceso de construcción de software también se acelera drásticamente con los paradigmas de diseño ya probados, y los desarrolladores móviles obtienen más beneficios en el proceso de desarrollo siguiendo los patrones arquitectónicos. Dado que la aplicación móvil es cada vez más grande, los desarrolladores móviles deben considerar los patrones de diseño

antes de que entren en el proceso de desarrollo de la aplicación. Varios estudios han demostrado que del 50% al 70% del costo total del ciclo de vida de un sistema de software se gasta en la evolución del sistema. Los patrones de arquitectura de software tiene un papel esencial en el proceso del cambio, refactorización y reescritura de funciones o características en el proceso de desarrollo de software.” (Sholichin, Bin Isa, Abd Halim, & Bin Harun, 2019)

SCRUM

Para el desarrollo del presente proyecto la entidad bancaria ha trabajado bajo el marco de trabajo SCRUM. Según (Schwaber & Sutherland, 2020) define Scrum como “...un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos.”

Este marco de trabajo permite “reducir el riesgo, controlar los costos y la calidad del producto ayudando a completar el proyecto a tiempo” (Hayat, Ur Rehman, Sarmad Arif, & Wahab, 2019).

Además, según (Alejandria Niño, 2021) “Scrum describe una metodología de organización y ejecución del trabajo con una visión ágil y adaptable, para lo cual se sostiene sobre algunos pilares: la transparencia, adaptación e inspección”. Conceptos que serán detallados a continuación:

- Transparencia: genera que las decisiones tengan un valor importante y que el riesgo disminuya. Además, da paso a la inspección.
- Inspección: permite detectar variaciones o problemas potencialmente indeceables. Además, permite la adaptación.
- Adaptación: en caso el producto resultante es inaceptable, el proceso o materiales deben ajustarse. Esto debe hacerse lo antes posible para evitar mayor desviación.

(Urteaga Pecharroman, 2015) menciona que “todo proyecto en scrum se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas). Cada una de las iteraciones tiene que proporcionar un resultado completo, un incremento de producto que sea potencialmente

entregable, de tal forma que cuando el cliente (Product Owner) lo solicite solo sea necesario un esfuerzo mínimo para que el producto este disponible para ser utilizado. Durante cada iteración existe una estrecha colaboración de tal forma que se ejecuten las siguientes dinámicas: Cada día el equipo realiza una reunión de sincronización, donde cada miembro inspecciona el trabajo de los otros para poder hacer las adaptaciones necesarias, comunica cuales son los impedimentos, actualiza el estado de la lista de tareas de la iteración (Sprint Backlog) y los gráficos de trabajo pendiente, y que el facilitador (Scrum Master) se encarga de que el equipo pueda cumplir con su compromiso y que no se merme su productividad mediante la eliminación de los obstaculos que el equipo no puede resolver por si mismo o mediante la protección del equipo de interrupciones externas que puedan afectar su compromiso o su productividad”

El Scrum Team, según (Schwaber & Sutherland, 2020), está conformado por:

- Developer: son las personas que se encarga de darle el incremento de valor al producto.
- Product Owner: es el encargado de darle el máximo valor al producto entregado por los developer.
- Scrum Master: se encarga de apoyar al Scrum team en la mejora de sus prácticas, dentro del marco de trabajo Scrum. Además, de guiar a los miembros del equipo en ser autogestionados y mutifuncionales, entres otras funciones que impiden bloqueantes para el desarrollo del producto.

El equipo de trabajo o Scrum Team trabaja bajo los siguientes eventos, según (Schwaber & Sutherland, 2020):

- Sprint: son eventos de duración fija, que para este proyecto se definió que tendría una duración de dos semanas.
- Sprint planning: El Scrum Team establece el trabajo que se realizará para el Sprint, definiendo así la lista de requerimientos priorizados para el sprint que viene.

- Sprint review: su propósito es inspeccionar el resultado del Sprint y determinar futuras adaptaciones.
- Sprint Retrospective: su propósito es planificar formas de aumentar la calidad y la efectividad. Se centra en las personas, interacciones, procesos, herramientas y la definición de terminado.

Todos estos eventos tienen que ser representados en trabajos o valor. Según (Schwaber & Sutherland, 2020) se define como artefactos, que te permite medir el progreso:

- Para el Product Backlog, es el objetivo del Producto.
- Para el Sprint Backlog, es el objetivo del Sprint

Pasaremos a definir los objetivos en mención:

- Producto backlog: es una lista de requerimientos planteados por el negocio plasmados por el Product Owner para la discusión, definición de criterios y priorización en conjunto con los developers y el scrum master.
- Sprint Backlog: de acuerdo a como se ha definido el Producto Backlog, el Scrum Team define qué requerimientos se van a considerar para el trabajo en el sprint. Esta lista de requerimientos se denomina Sprint Backlog.

3.2.5. Implementación de sus áreas de proceso y sus buenas prácticas

Para el proyecto se definió los siguientes roles:

Tabla 4: Roles definidos para el equipo de desarrollo

Roles del equipo de desarrollo	Cantidad de recursos
Product Owner	1
Líder Técnico	1
Desarrollador Back-end	1
Desarrollador Front-end iOS	1
Desarrollador Front-end Android	1
Analista de calidad	1

Total	6
--------------	----------

Nota: Elaboración propia

Para la primera fase de análisis de funcionalidades existentes se revisó la aplicación en producción y se hizo un listado de todas las funcionalidades en la tabla 5.

Tabla 5: Listado de funcionalidades existentes

Listado de funcionalidades existentes
Autenticación
Selección de empresa
Perfil de usuario
Estado de cuenta
Historial de movimientos
Operaciones monetarias pendientes de firma
Operaciones monetarias pendientes de envío

Nota: Elaboración propia

Siendo la funcionalidad Autenticación la única construida en nativo y las demás accedidas por el CWR.

Se realizó una reunión con el área de diseño para que construyeran las funcionalidades con sus respectivas pantallas. En la tabla 6 se podrá ver dicha distribución.

Tabla 6: Listado de pantallas para cada funcionalidad existente

Funcionalidad	Pantallas
Autenticación	Ingresar numero de tarjeta
	Ingresar Pin 6
Selección de empresa	Seleccionar empresa

Perfil de usuario	Ver perfil
Cuenta	Ver cuenta
Movimientos	Movimientos asociados a la cuenta
	Detalle de movimiento
Operaciones monetarias pendientes de firma	Listado operaciones monetarias pendientes de firma
	Detalle de operacion monetaria
	Detalle de operaciones monetarias
	Resultado de la firma de operaciones monetarias
Operaciones monetarias pendientes de firma	Listado de operaciones monetarias pendientes de envio
	Detalle de operación monetaria
	Detalle de operaciones monetarias
	Resultado del envío de operaciones monetarias
Envio y firma de operaciones protegida (pantallas que comparten)	Operación monetaria protegida

Nota: Elaboración propia

Luego de haber definido las pantallas correspondientes de cada funcionalidad existente, se procede con la segunda fase (creación del proyecto) iniciando con la definición de la arquitectura de la aplicación móvil, para ello se siguió los lineamientos de implementación de software que es definida por el área de arquitectura. Se realizó una reunión con dicha área lo que llevó a recomendaciones claras que fueron las siguientes:

- Clean architecture: se considera este concepto ya que permite la separación de la lógica en capas. Además, de promover buenas prácticas de programación.

- Patrón de arquitectura: se recomendó VIPER ya que los elementos de dicho patrón se acomoda al concepto de clean architecture. Además, el patrón en mención nos da un ordenamiento para el empaquetamiento de archivos dentro de la programación.

En la figura 4 se visualiza la distribución de los componentes VIPER bajo el concepto Clean Architecture en capas.

Figura 4: Distribución de los components VIPER mediante el concepto Clean Architecture



Nota: Elaboración propia

Además, para reforzar la elección del patrón de arquitectura el estudio realizado por (Hayat, Ur Rehman, Sarmad Arif, & Wahab, 2019) menciona que “MVVM y VIPER son bastante sobresalientes en términos de capacidad de prueba, modificabilidad y rendimiento”. Este sustento más el de la distribución de elementos VIPER bajo el concepto de capas de Clean Architecture nos da un punto de inicio importante para el implementación del proyecto.

Con lo explicado anteriormente la primera etapa (definición) perteneciente a la fase 2 (creación del proyecto) es cubierta.

Para la segunda etapa (preparación) se realizaron los features mostrados en la tabla 7.

Tabla 7: Lista de features de la etapa de preparación de la fase creación del proyecto

Feature
F01: Crear el proyecto desde Xcode
F02: Crear el proyecto en la plataforma de Apple

F03: Coordinar con el area de Devsecops

F04: Agregar las librerías externas construidas por la entidad bancaria

F05: implementar el empaquetamiento

F06: Implementar aspectos de seguridad

Nota: Elaboración propia

El detalle de cada feature de la tabla 7 lo verá en el Anexo 1.

Para el término de la etapa preparación se deja todo listo para iniciar la implementación de las funcionalidades. Además de registrar el proyecto en Apple para el despliegue a tienda (Appstore). Adicional a ello, permite la configuración de Integración Continua con el área de Devsecops para el despliegue a ambiente no productivo y ambiente productivo.

Para la fase 3 (gestión de los requerimientos) del proyecto se llevó a cabo mediante épicas y features. En la tabla 8 se mostrará todos los features que se ha considerado para el proyecto.

Tabla 8: Épicas y feature de la fase 3

Épicas	Features
E01: Implementación de Autenticación	E0#F01: Realizar reunión con el área de diseño
E02: Implementación de Seleccionar Empresa	E0#F02: Realizar reunión con backend para el consumo de servicios rest
E03: Implementación de Perfil del usuario	E0#F03: Implementar pruebas unitarias
E04: Implementación de Cuenta	E0#F04: Implementar la lógica visual
E05: Implementación de Historial de movimiento	E0#F05: Implementar la lógica de negocio
E06: Implementación de operaciones monetarias pendientes de firma	E0#F06: Implementar el consumo de servicios Rest
E07: Implementación de operaciones monetarias pendientes de envío	

Nota: Elaboración propia

El detalle de cada feature de la tabla 8 lo verá en el Anexo 2, y en el Anexo 3 se podrá visualizar el flujo de pantallas.

A continuación se procederá a explicar cada una de las épicas.

E01: Implementación de autenticación

Esta funcionalidad le permite al usuario ingresar a la aplicación y poder ver su información mediante el ingreso del número de tarjeta y una contraseña de seis dígitos.

E02: Implementación de selección de empresa

Esta funcionalidad permite seleccionar la empresa que estén asociadas a la tarjeta empresarial para poder visualizar las cuentas asociadas.

E03: Implementación del perfil del usuario

Esta funcionalidad permite al usuario visualizar los equipos afiliados al número de tarjeta, visualizar la opción de tener un número de contacto hacia la entidad bancaria y la opción de cerrar sesión.

E04: Implementación de cuenta

Esta funcionalidad permite visualizar el monto total de soles y en dólares que tiene asociado a las cuentas, y las cuentas asociadas a la empresa seleccionada previamente.

E05: Implementación de historial movimientos

Esta funcionalidad permite al usuario visualizar los movimientos de la cuenta, además del detalle que tienen ellas.

E06: Implementación de operaciones monetarias pendientes de firma

Esta funcionalidad permite al usuario poder elegir una o varias operaciones monetarias para poder ver su detalle y poder firmarlas. La pantalla con el resultado final de la firma de operaciones te da un estado de las

operaciones donde te indica si ha sido firmada exitosamente, si está pendiente de aprobación o si se ha rechazado.

E07: Implementación de operaciones monetarias pendientes de enviar

Esta funcionalidad permite al usuario elegir una o varias operaciones monetarias para ver su detalle y enviarlas para que sean ejecutadas. La pantalla con el resultado final del envío de operaciones muestra el estado de las operaciones indicando si ha sido enviada exitosamente, si está pendiente de aprobación o si se ha rechazado.

Es importante mencionar que existen usuarios que tiene el rol firmante y rol enviador. Esto quiere decir que el usuario con rol firmante solo podrán ejecutar las operaciones monetarias de la pantalla “Listado de operaciones monetarias pendientes de firma”, y los usuarios con rol enviador solo podrán ejecutar las operaciones monetarias de la pantalla “Listado de operaciones monetarias pendientes de envío”. Además, existen usuarios que tienen los dos roles, tanto firmante como enviador, esto les permite visualizar las operaciones en las pantallas “listado de operaciones monetarias pendientes de firma” y “listado de operaciones pendientes de envío” y poder ejecutarlas sin restricción. Con lo mencionado culmina la fase 3 del proyecto.

La fase 4 (publicación del producto) se realiza en tres pasos: el primero es en coordinación con el líder técnico ya que se encarga que la implementación pase de un ambiente de desarrollo a un ambiente no productivo en donde el analista de calidad se encarga de realizar las pruebas necesarias para validar lo implementado. Luego de la confirmación del analista de calidad, el líder técnico en coordinación con el área de seguridad realizan una revisión de vulnerabilidad en el proyecto a nivel de programación. Luego de la confirmación del área de seguridad, el líder técnico en coordinación con el área Devsecops realizan el pase del ambiente no productivo al ambiente productivo, que trae como consecuencia el envío de la aplicación a la tienda Appstore.

3.3. Evaluación

Se detallará la evaluación económica y los beneficios del proyecto.

3.3.1. Evaluación económica

Para el desarrollo del proyecto se contó con un equipo conformado por 6 personas.

En la tabla 9 se mostrará el estimado del presupuesto del proyecto en base a los recursos utilizados.

Tabla 9: Presupuesto del proyecto

Recursos	Unidades	Costo x Mes	Meses del proyecto	Costo total
Product Owner	1	S/ 8000	8	S/ 64000
Líder Técnico	1	S/ 7000	8	S/ 56000
Desarrollador Front-end Android	1	S/ 5000	8	S/ 40000
Desarrollador Front-end iOS	1	S/ 5000	8	S/ 40000
Desarrollador Back-end	1	S/ 5000	8	S/ 40000
Analista de Calidad	1	S/ 5000	8	S/ 40000
			Total	S/ 280 000

Nota: Elaboración propia

En la tabla 9 no se ha considerado el costo del Scrum master ya que es compartido con distintos proyectos. Considerar que solo se esta tomando en cuenta los gastos de desarrollo de la aplicación.

El costo total del proyecto fue de **S/ 280,000.00 (aproximado)**.

3.3.2. Beneficios obtenidos

En relación a los beneficios obtenidos se tienen planteados tanto el beneficio hacia los usuario y hacia la entidad bancaria.

Para los usuarios se tiene una aplicación mucho más amigable y estable, que promueve su uso sin ningún inconveniente. Para la entidad bancaria el principal beneficio es la preferencia de los usuarios por el uso de la aplicación.

A continuación veremos que luego de 6 meses de la publicación en la tienda Appstore se tuvieron los resultados tales como se muestra en la tabla 10.

Tabla 10: Incremento de usuarios y empresas luego de 6 meses de la publicación

	# Antes	# Después	Incremento %
Empresas	20 000	22 040	10%
Usuarios	50 000	54 000	8%

Nota: Elaboración propia

Considerando que antes de la publicación el incremento mensual de las empresas era de 200 aproximadamente, y luego de la publicación es de 340 empresas mensuales aproximadamente. Se nota una mejora en la captación de empresas de 70%. Esto viene de la mano de la captación de usuarios ya que previo a la publicación el incremento mensual de los usuarios era de 400 aproximadamente y luego de la publicación es de 660 aproximadamente, notándose un incremento del 65%.

Tener un incremento de usuarios no me asegura que exista una preferencia por la aplicación. Es por ello que es importante ver la data que muestra la tabla 11.

Tabla 11: Incremento de la cantidad de ejecuciones de operaciones monetarias luego de 6 meses de la publicación

	# Antes	# Después	Incremento %
Ejecución de operaciones monetarias	400 000	660 000	65%

Nota: Elaboración propia

El incremento de las ejecuciones de operaciones monetarias si nos dan un panorama favorable para notar un aumento en la preferencia de los usuarios para la aplicación.

CAPITULO IV

REFLEXION CRITICA DE LA EXPERIENCIA

- El autor del presente informe ha participado en el proyecto como desarrollador de funcionalidades, dándole mantenimiento al código y velando por la implementación de buenas prácticas en la programación.
- El autor del presente informe ha participado en el desarrollo del proyecto mobile bajo la metodología scrum que le permitió liderar eventos en el que se considera la presencia de stakeholder. Eventos tales como sprint review y daily permitieron la mejora de habilidades blandas para realizar una comunicación acertada sobre las actividades realizadas durante el sprint.
- Como miembro del equipo de desarrollo, el autor del presente informe tuvo la oportunidad de interactuar con las distintas áreas involucradas para llevar a cabo el proyecto. Esto ayudó a tener mayor conocimiento sobre el negocio y qué dependencias con otras áreas se tiene para cumplir con los objetivos.
- El autor del presente informe vela por mantener los lineamientos de programación que la entidad bancaria propone para el desarrollo de software.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- El análisis de las funcionalidades de la primera versión de la aplicación en producción dió como resultado una lista de funcionalidades a implementar con sus respectivas interfaces y un punto de partida para el desarrollo de la nueva aplicación.
- Se realizó el seguimiento del desarrollo de la aplicación utilizando todos los conceptos que nos otorga el marco de trabajo Scrum. Este marco de trabajo nos da un panorama de lo realizado con lo cual pudimos mitigar lo riesgos que se presentaron durante el proyecto.
- Se utilizó el concepto de Clean Architecture y el patrón de arquitectura VIPER para la establecer la arquitectura de la aplicación. Esto ayudó a la implementación de buenas prácticas de programación y a un mayor ordenamiento de los archivos por cada funcionalidad que se fue implementando.

5.2. Recomendaciones

Se recomienda seguir con el concepto de clean architecture pero por módulos, esto quiere decir que cada requerimiento o funcionalidad tenga sus propias tres capas (presentación, dominio y data). Este punto ayudaría a prevenir cualquier tipo de conflicto cuando se requiere unir funcionalidades que involucran mucha codificación y que se hayan implementado de manera simultánea.

5.3. Fuentes de informacion

Martin, R. C. (2017). Clean Architecture. Prentice Hall.

Gilbert, J., & Conrad, S. (Junio de 2014). Obtenido de <https://www.objc.io/issues/13-architecture/viper>

Nieto Sanchez, J. (2017). Modelo de arquitectura de software para aplicaciones iOS basado en clean architecture. Obtenido de Google Academico: https://scholar.google.es/scholar?hl=es&as_sdt=0%2C5&q=arquitectura+pattern+software+viper&btnG=

Schwaber, K., & Sutherland, J. (2020). La Guia de Scrum. La Guia definitiva de Scrum: Las Reglas del Juego. Obtenido de SCRUM GUIDES: <https://scrumguides.org/download.html>

Hayat, F., Ur Rehman, A., Sarmad Arif, K., & Wahab, K. (2019). The ,nfluence of Agile Methodology (Scrum) on Software Project Management. Obtenido de IEEE Explore: <https://ieeexplore.ieee.org/document/8935813>

Alejandria Niño, E. (2021). Implementación de sistema web para la carga, edición y transferencia de datos de máquinas de extracción de hidrocarburos para Tekton Labs en el 2020. Obtenido de Cybertesis UNMSM: https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/17102/Alejandria_ne.pdf?sequence=1&isAllowed=y

Urteaga Pecharroman, A. (2015). Aplicación de la metodología de desarrollo ágil Scrum para el desarrollo de un sistema de gestión de empresas. Obtenido de Biblioteca de la Universidad Carlos III de Madrid: https://e-archivo.uc3m.es/bitstream/handle/10016/23750/TFG_Aitor_Urteaga_Pecharroman.pdf?sequence=1&isAllowed=y

Sholichin, F., Bin Isa, M., Abd Halim, S., & Bin Harun, M. (2019). Review of ios architectural pattern for testability, modifiability, and performance quality. Journal of Theoretical and Applied Information Techology.

Butani, A. (2020). 5 essential patterns of software architecture. Obtenido de Red Hat: <https://www.redhat.com/architect/5-essential-patterns-software-architecture>

Gray, S., & Talbot, N. (2006). *Operaciones monetarias*. Retrieved from Banco Central de Reserva del Peru: <https://www.bcrp.gob.pe/docs/Publicaciones/Publicaciones-de-Otras-Instituciones/pub-en-74.pdf>

5.4. Glosario

- Backend: son los encargados de manejar los servidores, mediante servicios rest y bases de datos.
- Certificado de autenticación mutua: sirve como medida de seguridad para comunicación entre el cliente (frontend) y el servidor.
- CWR: Componente Web Responsive, es un componente dentro de la programación nativa de aplicación móviles que te permite mostrar una web responsive.
- Épica: es una historia de usuario que no puede ser entregada tal y como se ha definido dentro de una solo sprint. Es suficientemente grande para ser dividida en historias de usuario mas pequenas.
- Frontend: son los encargados de crear el software con el que el usuario va a interactuar.
- Integración continua: es una práctica de desarrollo de software mediante la cual los desarrolladores combinan los cambios en el código en un repositorio central de forma periódica, tras lo cual se ejecutan versiones y pruebas automáticas.
- Servicios rest: es una interfaz para conectar varios sistemas basados en el protocolo HTTP y nos sirve para obtener y generar datos y operaciones, devolviendo esos datos en formatos muy especificos, como XML y JSON.
- Web responsive: es un diseño que es capaz de adaptarse a pantallas de diferentes tamaños con un solo sitio web. El sistema detecta automáticamente el ancho de la pantalla y a partir de ahí adapta todos los elementos de la página, desde el tamaño de letra hasta las imágenes y los menús, para ofrecer al usuario la mejor experiencia posible.

ANEXOS

Anexo 1: Feature de la etapa de preparación de la fase de creación del proyecto

A continuación se detallará cada uno de los features.

F01: Crear el proyecto desde Xcode

Código	E01F01
Feature	Crear el proyecto desde Xcode
Descripción	Realizar la creación del proyecto considerando la definición del bundleid. Además de agregarlo al repositorio correspondiente.
Criterio de aceptación	Validar que se haya creado correctamente el proyecto en el repositorio y con el bundleid correcto.

F02: Crear el proyecto en la plataforma de Apple

Código	E01F02
Feature	Crear el proyecto en la plataforma Apple
Descripción	Realizar la creación del proyecto en la cuenta de Apple de la entidad bancaria.
Criterio de aceptación	Que se hayan creado correctamente el ID del proyecto. Además, el provisional profile y el certificado desde la plataforma de Apple.

F03: Coordinar con el área de DevSecOps

Código	E01F03
Feature	Coordinar con el área de DevSecOps
Descripción	Se debe realizar la coordinación con el área de DevSecOps para que se creen los pipeline para subir la implementación tanto a los ambiente de

	desarrollo, no productivo y productivo de manera automática con el uso de integración continua.
Criterio de aceptación	Validar que los pipelines se hayan creado correctamente.

F04: Agregar librerías externas construidas por la entidad bancaria

Código	E01F04
Feature	Agregar librerías externas construidas por la entidad bancaria
Descripción	Mediante el manejador de dependencia Cocoapods instalar las librerías obligatorias que manda la entidad bancaria para el funcionamiento de la aplicación móvil.
Criterio de aceptación	Validar que las dependencias se hayan instalado correctamente.

F05: Implementar el empaquetamiento

Código	E01F05
Feature	Implementar el empaquetamiento
Descripción	Crear carpetas y archivos necesarios para que se cumpla con el concepto de capas de clean architecture y con el patrón de arquitectura VIPER
Criterio de aceptación	Revisar que la creación de las carpetas cumpla con el concepto de capas de clean architecture y el patrón de arquitectura VIPER

F06: Implementar aspectos de seguridad

Código	E01F06
Feature	Implementar aspectos de seguridad
Descripción	Agregar el certificado de autenticación mutua para que la comunicación entre front-end y back-end sea lo más segura posible.
Criterio de aceptación	Validar que el certificado se agregue de manera correcta.

ANEXO 2: Feature de la fase 3 (gestión de los requerimientos)

A continuación se detallará cada uno de los feature.

E0#F01: Realizar reunión con el área de diseño (considerar que el símbolo # equivale a un número desde el 2 hasta el 10)

Código	E0#F01
Feature	Realizar reunión con el área de diseño
Descripción	Coordinar con el área de diseño para poder concretar una reunion donde se confirme que las pantallas de cada funcionalidad se encuentren subidas a Zeplin para su implementacion en el proyecto.
Criterio de aceptación	Validar que las pantallas se encuentren en Zeplin.

E0#F02: Realizar reunión con back-end para el consumo de servicios Rest (Considerar que el símbolo # equivale a un número desde el 2 hasta el 10)

Código	E0#F02
Feature	Realizar una reunión con back-end para el consumo de servicios Rest
Descripción	Coordinar con back-end para poder saber la cantidad de servicios Rest que se tienen que consumir. Además, de la estructura del response y request que implica en cada uno de los servicios para la funcionalidad
Criterio de aceptación	Validar que tanto el response como el request de los servicios rest se hayan coordinado entre frontend y backend, asi como la cantidad de servicios.

E0#F03: Implementar pruebas unitarias (Considerar que el símbolo # equivale a un número desde el 2 hasta el 10)

Código	E0#F03
Feature	Implementar pruebas unitarias
Descripción	En conjunto con la analista de calidad se coordina para saber los casos de prueba que tiene que evaluar por funcionalidad, en base a esos casos de prueba se van creando las pruebas unitarias.
Criterio de aceptación	Validar que las pruebas unitarias tengan relación con los casos de prueba.

E0#F04: Implementar la lógica visual (Considerar que el símbolo # equivale a un número desde el 2 hasta el 10)

Código	E0#F04
Feature	Implementar la lógica visual
Descripción	De acuerdo a las pantallas que se definió con el área de diseño se procede a implementar tanto la interfaz visual como la lógica para que la interfaz visual converce con las reglas de negocio.
Criterio de aceptación	Validar que las pantallas y el funcionamiento de ellas esten acorde a como se coordinó con el área de diseño.

E0#F05: Implementar la lógica de negocio (Considerar que el símbolo # equivale a un número desde el 2 hasta el 10)

Código	E0#F05
Feature	Implementar la lógica de negocio
Descripción	De acuerdo con el análisis de la funcionalidad se tiene que definir los casos de uso, en base a ello se va creando la lógica de negocio.

Criterio de aceptación	Validar que los casos de uso tengan relación con la funcionalidad.
-------------------------------	--

E0#F06: Implementar el consumo de servicios Rest (Considerar que el símbolo # equivale a un número desde el 2 hasta el 10)

Código	E0#F06
Feature	Implementar el consumo de servicios Rest
Descripción	De acuerdo a lo acordado con back-end se implementará el consumo de los servicios Rest.
Criterio de aceptación	Validar que los servicios Rest sean consumidos correctamente.

E0#F07: Realizar pruebas (Considerar que el símbolo # equivale a un número desde el 2 hasta el 10)

Código	E0#F07
Feature	Realizar pruebas
Descripción	Realizar pruebas de la funcionalidad desarrollada, en coordinación con back-end para el correcto envío y recepción de la data.
Criterio de aceptación	Validar que lo desarrollado en frontend y backend esté de acuerdo al correcto funcionamiento de la funcionalidad

ANEXO 3: Flujo de pantallas de la aplicación

A continuación se mostrará la imagen de como es el flujo de pantallas.

