



Pontificia Universidad
JAVERIANA
Bogotá

Planeación de Proyecto de Grado

Plan de Administración de Proyecto

Integrantes:

“Cristobal Castrillón Balcazar”

“Angie Tatiana Peña Peña”

“Santiago Vasquez Sanchez”

2023

1 Historial de Cambios

Fecha de Cambio	Descripción	Responsable
27-10-2022	Visión del producto	Angie Tatiana Peña Peña
27-10-2022	Propósito, alcance y objetivos	Angie Tatiana Peña Peña
27-10-2022	Supuestos y restricciones	Angie Tatiana Peña Peña
27-10-2022	Entregables	Cristobal Castrillón Balcázar
27-10-2022	Resumen de calendarización	Cristobal Castrillón Balcázar
27-10-2022	Evolución del Plan	Cristobal Castrillón Balcázar
27-10-2022	Lenguajes y herramientas	Cristobal Castrillón Balcázar
27-10-2022	Plan de Aceptación del Producto	Cristobal Castrillón Balcázar
29-10-2022	Modelo de ciclo de vida	Cristobal Castrillón Balcázar
29-10-2022	Interfaces externas	Angie Tatiana Peña Peña
29-10-2022	Organigrama y descripción de roles	Angie Tatiana Peña Peña
29-10-2022	Métodos y herramientas de estimación	Angie Tatiana Peña Peña
29-10-2022	Entrenamiento del personal	Angie Tatiana Peña Peña
29-10-2022	Infraestructura	Angie Tatiana Peña Peña
29-10-2022	Descomposición de actividades	Angie Tatiana Peña Peña
29-10-2022	Asignación de recursos	Angie Tatiana Peña Peña
29-10-2022	Monitoreo y control de progreso	Santiago Vasquez Sanchez
29-10-2022	Cierre de proyecto	Santiago Vasquez Sanchez
29-10-2022	Entrega de proyecto	Santiago Vasquez Sanchez
30-10-2022	Ambiente de Trabajo	Santiago Vasquez Sanchez
30-10-2022	Análisis y administración de riesgos	Angie Tatiana Peña Peña
30-10-2022	Control de Calidad	Santiago Vasquez Sanchez
11-11-2022	Revisión final	Todo el equipo
19-02-2023	Actualización del documento	Santiago Vasquez Sanchez

Tabla 1: Historial de Cambio

Contenido

1	Historial de Cambios	1
2	Vista General del Proyecto	6
2.1	Visión del Producto	6
2.2	Propósito, Alcance y Objetivos	6
2.2.1	Alcance	6
2.2.2	Objetivos	7
2.3	Supuestos y Restricciones	8
2.3.1	Supuestos	8
2.3.2	Restricciones	8
2.4	Entregables	9
2.5	Resumen de Calendarización y Presupuesto	9
3	Glosario	10
4	Contexto del proyecto	11
4.1	Modelo de Ciclo de Vida	11
4.1.1	Fase de Investigación	11
4.1.2	Fase de Planificación	12
4.1.3	Fase de Implementación	18
4.2	Lenguajes y Herramientas	18
4.2.1	Planeación y Diseño	20
4.2.2	Desarrollo	21
4.3	Plan de Aceptación del Producto	22
4.4	Organización del Proyecto y Comunicación	23
4.4.1	Interfaces Externas	23
4.4.2	Organigrama y Descripción de Roles	24
5	Administración del Proyecto	26
5.1	Métodos y Herramientas de Estimación	26
5.2	Inicio del Proyecto	26
5.2.1	Entrenamiento del Personal	26
5.2.2	Infraestructura	27
5.3	Planes de Trabajo del Proyecto	27
5.3.1	Descomposición de Actividades	27
5.3.2	Calendarización	28
5.3.3	Asignación de Recursos	28
5.4	Asignación de Presupuesto y Justificación	29
5.4.1	Justificación de la Asignación de Presupuesto	29
6	Monitoreo y Control del Proyecto	30
6.1	Administración de Requerimientos	30
6.1.1	Gestión de Cambios en Requerimientos	30
6.1.2	Comunicación y Seguimiento	30
6.1.3	Validación y Verificación de Requerimientos	30
6.1.4	Mantenimiento de Trazabilidad	31
6.2	Monitoreo y Control de Progreso	31
6.3	Toma de Decisiones con Análisis DOFA	31
6.3.1	Cierre del Proyecto	32

7	Entrega del Producto	33
7.1	Producto Mínimo Viable (MVP)	33
7.1.1	Entrega del Producto Mínimo Viable (MVP)	33
7.1.2	Documentación del MVP	33
7.1.3	Capacitación y Transición	33
7.1.4	Entrega de Manuales	33
8	Soporte	34
8.1	Ambiente de Trabajo	34
8.2	Análisis y Administración de Riesgos	34
8.3	Métricas y Proceso de Medición	35
8.3.1	Administración de Configuración y Documentación	35
8.3.2	Métricas y Procesos de Medición	35
8.4	Control de Calidad	35
9	Referencias	36

Lista de Figuras

1	Metodología Implementada en CashFlow	11
2	Organigrama del Equipo	24
3	Matriz DOFA	32

Lista de Tablas

1	Historial de Cambio	1
2	Fechas importantes según fases del proyecto	9
3	Conceptos Clave para la Gestión Financiera Personal en la Aplicación de Cashflow . . .	10
4	Cruce de Atributos CashFlow	17
5	Tabla de Atributos con Marcas de Verificación y Cruces	17
6	Comparación entre Flutter y Swift	19
7	Comparación entre Firabese y Otras BD	20
8	Descripción de los roles	25
9	Entrega del Producto Mínimo Viable (MVP)	33
10	Documentación del MVP	33
11	Capacitación y Transición	33
12	Entrega de Manuales	33
13	Reglas de Trabajo	34
14	Faltas y Sanciones	34
15	Faltas y Sanciones	35
16	Métricas y proceso de medición	35

2 Vista General del Proyecto

2.1 Visión del Producto

CashFlow tiene como visión diseñar e implementar una aplicación móvil para la plataforma Android que combine algoritmos de clasificación y el reconocimiento óptico de caracteres (OCR) para brindar a los usuarios una forma intuitiva y eficiente de registrar sus gastos e ingresos. CashFlow permitirá a los usuarios capturar imágenes de recibos de caja e ingresos, y utilizará el OCR para extraer automáticamente la información relevante.

Adicionalmente, presentará información detallada y estadísticas relevantes para el usuario; Uno de los aspectos destacados de la aplicación será su capacidad para proporcionar análisis detallados de los datos financieros. Los usuarios podrán acceder a informes y gráficos interactivos que les mostrarán una visión clara de sus gastos e ingresos. Podrán realizar un seguimiento de sus patrones de gasto, identificar áreas en las que pueden ahorrar dinero y establecer metas financieras alcanzables. Además, la aplicación les permitirá comparar sus datos financieros a lo largo del tiempo y obtener información sobre tendencias y cambios en sus hábitos de gasto.

La seguridad y la privacidad de los datos serán aspectos prioritarios en el desarrollo de la aplicación. Todos los datos de los usuarios se almacenarán de forma segura y se utilizarán técnicas de encriptación para proteger la información confidencial. El objetivo es proporcionar una experiencia de usuario excepcional, simplificando el proceso de registro de transacciones y ofreciendo información valiosa para mejorar la toma de decisiones financieras.

2.2 Propósito, Alcance y Objetivos

El propósito del proyecto es optimizar la gestión financiera personal de los usuarios mediante una aplicación móvil impulsada por tecnología de vanguardia, como la inteligencia artificial (IA) y el reconocimiento óptico de caracteres (OCR). Esta innovadora solución tiene como objetivo abordar los desafíos comunes asociados con la falta de un seguimiento detallado de los gastos e ingresos, la dificultad para establecer un presupuesto efectivo, el exceso en la utilización de servicios digitales y la toma de decisiones financieras impulsivas.

Para cumplir con este propósito, la aplicación móvil con IA proporcionará una amplia gama de funcionalidades para mejorar la gestión financiera personal de los usuarios. Entre estas características destacan el registro automático de ingresos y gastos, la categorización automática de los gastos, el análisis exhaustivo de los patrones de gasto, la generación de un presupuesto personalizado y recomendaciones de ahorro específicas. Además, la aplicación brindará alertas y recordatorios oportunos para mantener a los usuarios informados sobre su estado financiero y fomentar una mayor conciencia de sus finanzas personales.

En resumen, el objetivo principal del proyecto es mejorar significativamente la organización financiera personal de los usuarios a través de una aplicación móvil intuitiva y eficaz. Al aprovechar la inteligencia artificial y el reconocimiento óptico de caracteres, la aplicación simplificará el proceso de registro, análisis y toma de decisiones financieras, impulsando así una vida financiera más saludable y consciente.

2.2.1 Alcance

El alcance del proyecto se centra en el desarrollo de un Producto Mínimo Viable (MVP) de una aplicación móvil de organización financiera personal que incluye la implementación de OCR. El obje-

tivo es proporcionar a los usuarios una forma eficiente de registrar sus gastos e ingresos mediante el reconocimiento óptico de caracteres en imágenes de recibos de caja e ingresos.

- **Registro mediante OCR:** El MVP permitirá a los usuarios capturar imágenes de sus recibos de caja utilizando la cámara del dispositivo móvil. A través de la implementación de OCR, la aplicación analizará y extraerá automáticamente la información relevante, como la fecha, el monto y la descripción de la transacción.
- **Categorización automática de gastos:** La aplicación utilizará el OCR para clasificar automáticamente los gastos en diferentes categorías, como alimentación, transporte, entretenimiento, etc. Esto ayudará a los usuarios a tener una visión clara de cómo se distribuyen sus gastos.
- **Análisis de patrones de gasto:** La aplicación utilizará los datos capturados mediante OCR para analizar los patrones de gasto del usuario. Esto permitirá generar informes y ofrecer gráficos personalizados para reducir gastos innecesarios y mejorar la gestión financiera.

Es importante tener en cuenta que, dado que se trata de un Producto Mínimo Viable, la implementación del OCR puede tener ciertas limitaciones y no garantizar una precisión del 100 %. Sin embargo, el objetivo principal es demostrar la viabilidad y utilidad de esta funcionalidad para mejorar la organización financiera personal de los usuarios.

2.2.2 Objetivos

- Desarrollar una aplicación móvil funcional para la plataforma Android que permita a los usuarios registrar y organizar sus gastos e ingresos de manera eficiente.
- Implementar el reconocimiento óptico de caracteres (OCR) en la aplicación móvil para facilitar el registro automático de los datos de los recibos de caja e ingresos.
- Clasificar automáticamente los gastos en diferentes categorías, como alimentación, transporte, entretenimiento, etc., utilizando la tecnología de inteligencia artificial (IA) y el análisis de los datos recopilados.
- Analizar los patrones de gasto del usuario utilizando la IA y proporcionar recomendaciones personalizadas para reducir gastos innecesarios y mejorar la gestión financiera.
- Generar un presupuesto personalizado basado en los ingresos y gastos registrados y actualizarlo automáticamente a medida que se añadan nuevas transacciones.
- Ofrecer recomendaciones personalizadas para ahorrar dinero y establecer un fondo de emergencia basado en la información recopilada y el análisis de los patrones de gasto del usuario.
- Enviar alertas y recordatorios al usuario para notificarle el pago oportuno de facturas, fechas de vencimiento y otros eventos importantes relacionados con sus finanzas personales.
- Demostrar la viabilidad y utilidad de la aplicación móvil con funcionalidades de OCR y análisis financiero en la mejora de la organización financiera personal de los usuarios.
- Obtener retroalimentación de los usuarios sobre la aplicación móvil y realizar mejoras iterativas en futuras versiones del producto.
- Proporcionar una solución innovadora y efectiva para abordar los problemas de falta de organización financiera personal y mejorar la educación financiera de las personas a través del uso de tecnologías de inteligencia artificial.

2.3 Supuestos y Restricciones

2.3.1 Supuestos

- **Funcionalidades Básicas:** Se asume que el MVP se enfocará únicamente en las funcionalidades principales y fundamentales para el registro de ingresos y gastos, categorización básica de gastos y generación de un presupuesto personalizado. Se espera que estas funcionalidades sean suficientes para proporcionar valor a los testers y posibles usuarios.
- **Limitaciones de IA:** Se supone que la inteligencia artificial (IA) implementada en el MVP tendrá ciertas limitaciones en cuanto a su capacidad de análisis y recomendaciones. Es posible que no pueda realizar análisis de patrones de gasto complejos ni ofrecer recomendaciones personalizadas avanzadas. Sin embargo, se espera que la IA pueda realizar clasificaciones básicas de gastos y proporcionar recomendaciones generales.
- **Capacidad OCR:** Considerando el enfoque en un MVP, se asume que la implementación del reconocimiento óptico de caracteres (OCR) tendrá una capacidad limitada. Puede que no sea capaz de reconocer y extraer datos de todos los tipos de recibos o facturas, y su precisión puede ser menor en comparación con soluciones más sofisticadas. Sin embargo, se espera que pueda manejar una cantidad razonable de casos comunes.
- **Interfaz de Usuario Simplificada:** Se supone que la interfaz de usuario del MVP será simplificada y se centrará en las acciones principales, como ingresar gastos e ingresos, ver el presupuesto y recibir recomendaciones básicas. No se espera que la interfaz cuente con todas las características y elementos de diseño avanzados, sino que se mantenga enfocada en las funcionalidades principales.
- **Retroalimentación de los Usuarios:** A pesar de ser un MVP, se espera que los usuarios brinden retroalimentación sobre la funcionalidad y usabilidad del producto. La retroalimentación será fundamental para iterar y mejorar el MVP en futuras versiones.

2.3.2 Restricciones

- **Funcionalidades Limitadas:** El MVP estará sujeto a restricciones en términos de funcionalidades. Solo se implementarán las características esenciales y fundamentales necesarias para el registro de ingresos y gastos, categorización básica de gastos, generación de un presupuesto personalizado y ofrecimiento de recomendaciones básicas.
- **Alcance Reducido:** El alcance del MVP se limitará a las características principales y críticas para el funcionamiento básico del producto. Se eliminarán funcionalidades secundarias y características adicionales que no sean imprescindibles para el objetivo principal del MVP. Esto se hace para agilizar el desarrollo y cumplir con el tiempo estimado.
- **Limitaciones Técnicas:** Dado que se trata de un MVP, es posible que existan limitaciones técnicas debido a la falta de tiempo y recursos para desarrollar soluciones más complejas. Estas limitaciones pueden incluir la precisión limitada del OCR, menor capacidad de procesamiento y análisis de datos, y falta de integración con plataformas y servicios externos. El MVP se centrará en cumplir con las funcionalidades básicas dentro de estas limitaciones.
- **Diseño Simplificado:** El diseño y la interfaz de usuario del MVP serán simplificados y enfocados en la usabilidad básica. Se dará prioridad a la funcionalidad sobre la estética y se utilizarán componentes y elementos de diseño estándar para agilizar el desarrollo. El diseño visual y las características de personalización más avanzadas se dejarán para futuras iteraciones del producto.
- **Retroalimentación Temprana de los Usuarios:** Dado que se trata de un MVP, se espera obtener retroalimentación de los usuarios de manera temprana y constante. Esto significa que el

producto, aunque no vaya a salir al mercado ni tenga un modelo de negocios, se espera que haya cierta retroalimentación de un reducido grupo de testers.

2.4 Entregables

Los entregables que el cliente recibirá y que se van a desarrollar durante la ejecución del proyecto, son los siguientes:

- **Documento de Proyecto de Grado:** Documento principal que describe el proyecto en su totalidad, incluyendo la introducción, antecedentes, problema de investigación, objetivos, metodología, resultados, conclusiones y recomendaciones. El documento de proyecto de grado debe contener toda la información relevante sobre el desarrollo y evaluación del MVP.
- **Prototipo Funcional:** Como parte del MVP, se espera que se entregue un prototipo funcional de la aplicación móvil. El prototipo debe demostrar las funcionalidades principales del MVP, como el registro de ingresos y gastos, categorización básica, generación de presupuesto y recomendaciones básicas. El prototipo puede ser presentado en forma de una aplicación móvil interactiva o una demostración en vídeo.
- **Informe de Diseño:** Este informe describirá el proceso de diseño de la aplicación móvil, incluyendo el análisis de requisitos, diseño de interfaz de usuario, arquitectura de software y detalles técnicos relevantes. El informe debe proporcionar una visión detallada del enfoque utilizado para desarrollar el MVP y las decisiones de diseño tomadas.
- **Informe de Pruebas y Evaluación:** En este informe se detallarán las pruebas realizadas durante el desarrollo del MVP, incluyendo pruebas de funcionalidad, pruebas de usabilidad y evaluación del rendimiento. Se deben presentar los resultados de las pruebas y análisis de los mismos, junto con cualquier mejora realizada en base a los hallazgos.
- **Análisis de Resultados y Conclusiones:** Este documento resumirá los resultados obtenidos durante la implementación y evaluación del MVP. Debe incluir un análisis de la efectividad del MVP en relación con los objetivos establecidos, así como conclusiones sobre las limitaciones, lecciones aprendidas y posibles mejoras futuras.
- **Otros Elementos:** Además de estos entregables, también se pueden considerar otros elementos como presentaciones orales, póster o material adicional que ayude a respaldar y explicar el proyecto de tesis.

2.5 Resumen de Calendarización y Presupuesto

En la Tabla 2 se detallan las fases del proyecto junto con sus respectivas fechas de inicio y finalización. Se debe tener en cuenta que, a pesar de que se hará todo lo posible por cumplir con estas fechas, pueden surgir imprevistos que las alteren.

Fase	Fecha de Inicio	Fecha de Fin
Identificación del contexto	23-07-2023	29-07-2023
Diseño de la arquitectura	30-07-2023	08-08-2023
Implementación	07-08-2023	14-10-2023
Pruebas de implementación	15-10-2023	30-10-2023
Integración y pruebas	01-11-2023	13-11-2023

Tabla 2: Fechas importantes según fases del proyecto

3 Glosario

En la Tabla 3 se pueden identificar conceptos relacionados al contexto:

Concepto	Descripción
Flujo de Efectivo	El flujo de efectivo se refiere al movimiento de dinero dentro y fuera de una cuenta financiera durante un período de tiempo específico. Es esencial para comprender la liquidez y la salud financiera a corto plazo.
Ingresos y Gastos	Los ingresos representan las cantidades de dinero que una persona recibe, ya sea a través de sueldo, ingresos por inversiones u otras fuentes. Los gastos son las cantidades de dinero gastadas en bienes y servicios.
Categorización y Etiquetado	La categorización implica organizar los ingresos y gastos en diferentes grupos o categorías (por ejemplo, alimentación, vivienda, transporte). El etiquetado implica asignar etiquetas específicas a transacciones individuales para un seguimiento detallado.
Presupuesto Personal	Un presupuesto personal es un plan que establece cómo se distribuirán los ingresos entre diferentes categorías de gastos. Ayuda a controlar los gastos y a garantizar que los ingresos sean suficientes para cubrir las necesidades.
Ahorro e Inversiones	El ahorro implica reservar una parte de los ingresos para futuros usos o emergencias. Las inversiones son colocaciones de dinero con el objetivo de obtener un rendimiento financiero a largo plazo.
Alertas y Notificaciones	Las alertas y notificaciones son mensajes o avisos automáticos que la aplicación envía al usuario para informar sobre eventos importantes, como gastos inusuales, vencimientos de pagos, etc.
Análisis de Tendencias y Patrones	Esta función permite al usuario identificar tendencias a lo largo del tiempo en sus patrones de gastos e ingresos, lo que facilita la toma de decisiones informadas y el ajuste del presupuesto.
Planificación de Metas Financieras	La planificación de metas financieras implica establecer objetivos específicos, como ahorrar para un viaje, comprar una casa o pagar deudas, y realizar un seguimiento del progreso hacia su logro.

Tabla 3: Conceptos Clave para la Gestión Financiera Personal en la Aplicación de Cashflow

4 Contexto del proyecto

4.1 Modelo de Ciclo de Vida

Para el planteamiento del modelo de ciclo de vida del proyecto, se definieron tres objetivos principales con el fin de desarrollarlo correctamente, estos son:

- Determinar las metodologías que mejor se ajustan al proyecto y, si es necesario, plantear una unión de éstas.
- Identificar las herramientas indicadas para la gestión, monitoreo y control de las actividades del proyecto.
- Implementar el modelo de ciclo de vida con el fin de alcanzar la meta propuesta.

Ahora, considerando los alcances, los objetivos y las visiones propuestas para el proyecto de grado, se partirá de dos metodologías y luego, se presentará una unión de éstas. Esto último, como fundamento principal para la planeación, desarrollo, gestión y seguimiento del proyecto. Con base en lo anterior, las metodologías que se usarán en cuestión como fundamento para el desarrollo de una propia son:

- **SCRUMBAN:** Es un marco de gestión de proyectos que combina características importantes de dos metodologías: *Scrum* y *Kanban*. Este junta la estructura y las rutinas de *Scrum* con la flexibilidad de *Kanban* para hacer que el equipo sea más ágil, eficiente y productivo. Considerando lo anterior, se mejora la eficiencia y la satisfacción general del cliente, priorizando un producto de alta calidad a través de la mejora continua. [10]

Con base en las metodologías anteriores, se buscó crear una adaptación que abarcara ambas. Por un lado, se considerarían los hitos y tareas asignadas para el desarrollo del sistema web y, por otro lado, se llevaría a cabo un proceso de tratamiento de datos en las resonancias. A continuación, se entrenaría un modelo de redes neuronales convolucionales, se evaluarían y compararían sus resultados y se realizaría un despliegue para integrar el modelo y el prototipo web. El grupo ha propuesto una metodología de diseño denominada ”**SCRUMBAN Modificado**”, que se describe a continuación:

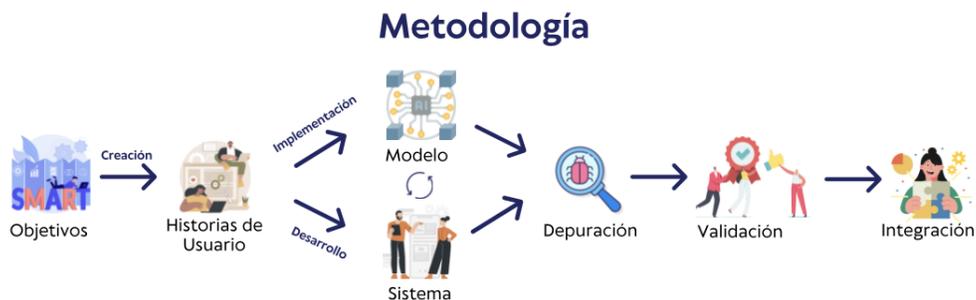


Figura 1: Metodología Implementada en CashFlow

En cada una de las fases, se mantuvo comunicación con la directora de forma semanal para llevar un seguimiento de avances y resolución de dudas.

4.1.1 Fase de Investigación

En la fase de investigación, se llevó a cabo un análisis integral que abarcó tanto la tecnología subyacente como el diseño visual, orientándose hacia la construcción de una aplicación robusta y atractiva.

Se realizaron pruebas de concepto exhaustivas para evaluar diversas tecnologías de reconocimiento óptico de caracteres (OCR) y herramientas, considerando factores como precisión y velocidad. Simultáneamente, se exploraron artículos científicos y técnicos para obtener una comprensión profunda de las metodologías y enfoques más recientes en el ámbito de OCR.

En paralelo, se dedicó una atención especial al "look and feel" de la aplicación. Se llevó a cabo una evaluación minuciosa del tema de colometría, consultando estudios que abordaban la psicología del color y su influencia en la respuesta emocional de los usuarios. Esta investigación contribuyó a la selección de una paleta de colores que no solo reflejara la identidad visual de la aplicación, sino que también generara las sensaciones deseadas para cautivar al usuario.

El análisis del "look and feel" se extendió a la revisión de artículos sobre diseño de interfaz y estrategias centradas en el usuario. Se buscó optimizar la usabilidad mediante la selección adecuada de elementos visuales, como tipografía y diseño de interfaz, con el objetivo de garantizar una experiencia de usuario intuitiva y envolvente.

En conjunto, esta investigación integral sienta las bases para una toma de decisiones informada y una implementación eficiente en las etapas posteriores del desarrollo. La combinación de la evaluación técnica y estética garantiza que la aplicación no solo cumpla con los estándares de precisión y rendimiento, sino que también ofrezca una experiencia visual y funcionalmente cautivadora para el usuario final.

4.1.2 Fase de Planificación

Esta etapa del proyecto fue vital para la estructuración del proyecto. Se tuvo inicialmente un plan de definición de requisitos general, en el que se plantearon historias de usuario, casos de uso, se identificaron los requisitos del sistema y se estableció un plan de trabajo detallado. Seguido a esto, se desarrolló y diseñó la arquitectura del proyecto con base en el modelo de arquitectura C4.

- **Análisis de Requisitos:** El análisis de requisitos es una fase crucial para el desarrollo de cualquier solución, ya que establece las bases sobre las cuales se construirá el sistema. En el caso de nuestra aplicación móvil para la gestión de ingresos y gastos, este análisis se llevó a cabo considerando varios factores clave:
 - **Estudio del Estado del Arte:** Basándonos en el análisis exhaustivo realizado sobre las aplicaciones financieras existentes (como se describió en la sección REFERENCIAR), identificamos funcionalidades esenciales y características deseables. Las aplicaciones líderes en el mercado ofrecen características como seguimiento de gastos, presupuestos, sincronización en la nube, categorización de transacciones, entre otras. Asegurando que nuestra solución no solo incorporara estas funcionalidades, sino que también introdujera innovaciones y mejoras.
 - **Perfil de Usuarios:** Consideramos a los usuarios potenciales de la aplicación: personas que buscan una herramienta fácil de usar para gestionar sus finanzas personales o empresariales. A través de investigaciones y encuestas, identificamos sus necesidades, preferencias y posibles desafíos al usar aplicaciones similares. Esto ayudó a diseñar una solución adaptada específicamente a sus necesidades.
 - **Contexto de Uso:** Reconocemos que la aplicación será utilizada en diversos contextos: desde la rápida revisión de gastos durante un viaje hasta la planificación detallada del presupuesto mensual en casa. Por lo tanto, la solución debe ser versátil, permitiendo una fácil entrada y categorización de datos, y ofreciendo análisis detallados cuando se necesiten.

Con este análisis detallado, establecimos un conjunto claro de requisitos funcionales y no funcionales para la aplicación. Estos requisitos sirvieron como guía durante las fases de diseño e implementación, asegurando que la solución final cumpliera con las expectativas y necesidades de los usuarios.

- **Análisis de Calidad:**

En el contexto del diseño y desarrollo de la aplicación, se ha llevado a cabo una exhaustiva comparativa de atributos esenciales que abarcan tanto aspectos tecnológicos como de experiencia del usuario. La siguiente tabla presenta un análisis detallado de las relaciones entre diferentes atributos, destacando las prioridades y consideraciones clave en la toma de decisiones. La ponderación cuidadosa entre elementos como usabilidad, seguridad, rendimiento y otros, ha sido fundamental para establecer un equilibrio que garantice tanto la eficiencia técnica como una experiencia de usuario optimizada. Este análisis informado sienta las bases para una implementación coherente y eficaz, alineada con los objetivos y requisitos específicos del proyecto.

Atributo (Fila)	Atributo (Columna)	Explicación
Usabilidad	Seguridad	La usabilidad se prefiere sobre la seguridad porque si una aplicación no es fácil de usar, los usuarios pueden no querer adoptarla en primer lugar. Sin embargo, la seguridad sigue siendo una alta prioridad dado que la información financiera está involucrada.
Usabilidad	Rendimiento	La usabilidad es más prioritaria que el rendimiento porque una experiencia de usuario intuitiva y amigable es esencial para la adopción de la aplicación. Aunque el rendimiento es crucial, no será relevante si los usuarios encuentran la aplicación difícil de usar.
Usabilidad	Capacidad	La usabilidad prevalece sobre la capacidad ya que una interfaz de usuario eficaz es esencial para garantizar que los usuarios aprovechen al máximo la capacidad de la aplicación.
Usabilidad	Testability	La usabilidad es más importante ya que directamente afecta la experiencia del usuario final, mientras que la testability, aunque esencial para el desarrollo, no es visible para el usuario.
Usabilidad	Interoperabilidad	La usabilidad se prioriza ya que una buena experiencia de usuario es esencial para el uso de la aplicación, mientras que la interoperabilidad, aunque crucial, es una característica más técnica y oculta.
Usabilidad	Mantenibilidad	La usabilidad es primordial para la adopción por parte del usuario, mientras que la mantenibilidad, aunque crucial para la longevidad de la aplicación, no afecta directamente al usuario.
Usabilidad	Portabilidad	La usabilidad es más importante porque afecta directamente cómo los usuarios interactúan con la aplicación, mientras que la portabilidad es una característica más técnica que permite a la aplicación operar en diferentes plataformas.

Atributo (Fila)	Atributo (Columna)	Explicación
Usabilidad	Adaptabilidad	La usabilidad se prefiere ya que determina cómo los usuarios experimentan la aplicación. Aunque la adaptabilidad es importante, la experiencia del usuario es primordial.
Seguridad	Rendimiento	La seguridad se prioriza sobre el rendimiento porque proteger la información del usuario es crucial. El rendimiento es importante, pero no a expensas de comprometer la seguridad.
Seguridad	Capacidad	La seguridad es más crucial que la capacidad ya que garantizar la protección de los datos de los usuarios es esencial, independientemente del volumen de datos que la aplicación pueda manejar.
Seguridad	Testability	La seguridad es primordial porque garantiza la protección de los datos del usuario. La testability es esencial para el proceso de desarrollo, pero no se prioriza sobre la seguridad.
Seguridad	Interoperabilidad	La seguridad prevalece porque garantiza la integridad y protección de la información del usuario. La interoperabilidad, aunque esencial para integrar con otros sistemas, no se prioriza sobre la seguridad.
Seguridad	Mantenibilidad	La seguridad es más importante porque protege la información del usuario. La mantenibilidad es crucial para el ciclo de vida del software, pero la protección de datos es primordial.
Seguridad	Portabilidad	La seguridad se prioriza ya que proteger los datos del usuario es esencial, mientras que la portabilidad permite que la aplicación funcione en diferentes plataformas.
Seguridad	Adaptabilidad	La seguridad es primordial para proteger los datos del usuario. Aunque la adaptabilidad permite que la aplicación se ajuste a diferentes contextos, la seguridad es más crítica.
Rendimiento	Capacidad	El rendimiento es más importante que la capacidad porque garantizar que la aplicación funcione de manera fluida y rápida es esencial para la experiencia del usuario. Aunque la capacidad determina cuánto puede manejar la aplicación, un buen rendimiento es crucial para mantener a los usuarios satisfechos.
Rendimiento	Testability	El rendimiento prevalece sobre la testability ya que una aplicación rápida y eficiente es lo que el usuario final experimenta. Aunque la testability es esencial para garantizar la calidad del software, no es directamente visible para el usuario.

Atributo (Fila)	Atributo (Columna)	Explicación
Rendimiento	Interoperabilidad	El rendimiento es prioritario porque garantiza que la aplicación funcione de manera óptima. La interoperabilidad, aunque importante para trabajar con otros sistemas, es secundaria en comparación con la experiencia del usuario directo.
Rendimiento	Mantenibilidad	El rendimiento es más importante porque afecta directamente cómo los usuarios experimentan la aplicación. La mantenibilidad es crucial para las actualizaciones y el ciclo de vida del software, pero la experiencia del usuario es primordial.
Rendimiento	Portabilidad	El rendimiento se prioriza ya que una aplicación rápida es esencial para la experiencia del usuario. La portabilidad, aunque permite que la aplicación funcione en diferentes plataformas, no se prioriza sobre una experiencia de usuario óptima.
Rendimiento	Adaptabilidad	El rendimiento es primordial para garantizar una experiencia de usuario fluida. Aunque la adaptabilidad es importante para ajustarse a diferentes contextos y cambios, no se prioriza sobre el rendimiento.
Capacidad	Testability	La capacidad, que determina cuánto puede manejar la aplicación, se prefiere sobre la testabilidad. Aunque garantizar que el software sea testable es importante, la capacidad tiene un impacto más directo en la experiencia del usuario.
Capacidad	Interoperabilidad	La capacidad se prioriza ya que determina el volumen de datos que la aplicación puede manejar. La interoperabilidad es crucial para la integración con otros sistemas, pero la capacidad tiene un impacto directo en la funcionalidad.
Capacidad	Mantenibilidad	Aunque la capacidad es crucial para determinar cuánto puede manejar la aplicación, la mantenibilidad, que asegura que la aplicación pueda ser actualizada y corregida con facilidad, es igualmente importante. Ambos son vitales, pero la capacidad tiene un impacto más directo en la experiencia del usuario.
Capacidad	Portabilidad	La capacidad es más importante que la portabilidad porque determina el volumen de datos que la aplicación puede manejar. Aunque la portabilidad es esencial para operar en diferentes plataformas, la capacidad tiene un impacto más directo en la funcionalidad.

Atributo (Fila)	Atributo (Columna)	Explicación
Capacidad	Adaptabilidad	La capacidad, que determina cuánto puede procesar la aplicación, se prefiere sobre la adaptabilidad. Aunque adaptarse a diferentes contextos es valioso, manejar grandes volúmenes de datos es esencial para la funcionalidad principal.
Testability	Interoperabilidad	La testabilidad se prioriza sobre la interoperabilidad ya que garantizar que la aplicación sea testeable es crucial para su calidad y funcionamiento. Aunque la interoperabilidad es esencial para la integración con otros sistemas, asegurarse de que el software funcione como se espera es fundamental.
Testability	Mantenibilidad	La testabilidad y la mantenibilidad están estrechamente relacionadas, pero la testabilidad es ligeramente más importante. Si una aplicación es fácilmente testeable, es probable que también sea mantenible. Sin embargo, la prioridad es garantizar que la aplicación funcione correctamente antes de considerar cómo se mantendrá en el futuro.
Testability	Portabilidad	La testabilidad se prefiere sobre la portabilidad. Es crucial garantizar que la aplicación funcione correctamente en su plataforma principal antes de considerar cómo se desempeñará en otras plataformas.
Testability	Adaptabilidad	Asegurarse de que la aplicación sea testeable es más importante que su adaptabilidad. Aunque adaptarse a diferentes contextos y cambios es valioso, es esencial garantizar que la aplicación funcione como se espera.
Interoperabilidad	Mantenibilidad	La interoperabilidad, que asegura que la aplicación pueda trabajar con otros sistemas, es ligeramente más prioritaria que la mantenibilidad. Sin embargo, ambas son esenciales para garantizar que la aplicación tenga un ciclo de vida largo y sea útil en un ecosistema más amplio.
Interoperabilidad	Portabilidad	La interoperabilidad se prioriza sobre la portabilidad ya que trabajar con otros sistemas y plataformas es esencial. Aunque funcionar en diferentes plataformas es valioso, integrarse con otros sistemas es más crítico para el ecosistema general.

Atributo (Fila)	Atributo (Columna)	Explicación
Interoperabilidad	Adaptabilidad	La interoperabilidad es más importante que la adaptabilidad. Asegurarse de que la aplicación pueda interactuar con otros sistemas y servicios es esencial, mientras que adaptarse a diferentes contextos es una característica secundaria.
Mantenibilidad	Portabilidad	La mantenibilidad, que asegura que la aplicación pueda ser fácilmente actualizada y corregida, es más prioritaria que la portabilidad. Aunque funcionar en diferentes plataformas es valioso, garantizar un ciclo de vida largo para la aplicación es más crucial.
Mantenibilidad	Adaptabilidad	La mantenibilidad se prefiere sobre la adaptabilidad. Aunque adaptarse a diferentes contextos y cambios es valioso, es más esencial asegurarse de que la aplicación pueda ser mantenida y actualizada con facilidad.
Portabilidad	Adaptabilidad	La portabilidad, que permite que la aplicación funcione en diferentes plataformas, es ligeramente más importante que la adaptabilidad. Aunque adaptarse a diferentes contextos es valioso, garantizar que la aplicación sea accesible en diferentes dispositivos y sistemas es crucial.

Tabla 4: Cruce de Atributos CashFlow

En la tabla 5, las marcas de verificación (✓) indican que el atributo correspondiente a la fila es considerado más importante que el atributo de la columna respectiva. Por otro lado, las cruces (×) señalan que el atributo de la columna tiene una mayor prioridad que el atributo de la fila. Cuando ambas celdas contienen un guion (-), denota que ambos atributos son percibidos como igualmente relevantes en términos de prioridad. Esta representación visual facilita la comprensión de las relaciones de importancia relativa entre diferentes atributos, proporcionando una guía rápida para la toma de decisiones y la asignación de recursos en el diseño y desarrollo del sistema correspondiente.

Atributo	U	S	R	C	T	I	M	P	A
Usabilidad (U)	-	✓	✓	✓	✓	✓	✓	✓	✓
Seguridad (S)	×	-	✓	✓	✓	✓	✓	✓	✓
Rendimiento (R)	×	×	-	✓	✓	✓	✓	✓	✓
Capacidad (C)	×	×	×	-	✓	✓	✓	✓	✓
Testability (T)	×	×	×	×	-	✓	✓	✓	✓
Interoperabilidad (I)	×	×	×	×	×	-	✓	✓	✓
Mantenibilidad (M)	×	×	×	×	×	×	-	✓	✓
Portabilidad (P)	×	×	×	×	×	×	×	-	✓
Adaptabilidad (A)	×	×	×	×	×	×	×	×	-

Tabla 5: Tabla de Atributos con Marcas de Verificación y Cruces

- Diagramas de bases de datos:

- Arquitectura con base en el modelo C4: Considerando que se tuvo como base el modelo C4 para el diseño de la arquitectura del sistema, se definieron en esta actividad los diagramas de contexto, contenedores y componentes. Para el cuarto nivel del modelo (código), se decidió hacer la construcción de la documentación en formato expuesto por Apple con la herramienta xcode.

4.1.3 Fase de Implementación

En esta fase, se tuvo en cuenta las habilidades de cada estudiante para hacer una división de tareas. Fue bastante importante la comunicación constante entre los miembros del proyecto, esto debido a la etapa posterior de Integración. Se aclara que hubo varias iteraciones de entregas continuas debido a modificaciones o mejoras que se debían implementar, sin embargo, se describe el desarrollo para llegar a la solución final.

4.2 Lenguajes y Herramientas

El proyecto se desarrollará utilizando Swift como lenguaje y herramienta de desarrollo de aplicaciones móviles. Esta plataforma nos brindará un entorno de desarrollo robusto y versátil para la creación de interfaces de usuario interactivas y funcionales.

Además, para la gestión de datos y servicios en la nube, utilizaremos Firebase, una plataforma de desarrollo de aplicaciones móviles respaldada por Google. Firebase nos permitirá implementar autenticación de usuarios, almacenamiento de datos en tiempo real, notificaciones push y muchas otras características útiles para mejorar la experiencia del usuario y la funcionalidad de la aplicación.

En cuanto a las herramientas de Google, aprovecharemos el ecosistema de servicios y herramientas que ofrece para potenciar nuestro proyecto. Esto puede incluir Google Analytics para el seguimiento y análisis de datos, Google Maps para la integración de mapas y geolocalización, y Google Cloud Platform para el despliegue y administración de recursos en la nube.

Con la combinación de Swift, Firebase y las herramientas de Google, estaremos equipados con un conjunto sólido y completo de tecnologías para desarrollar una aplicación móvil moderna, eficiente y altamente funcional. Estas herramientas nos permitirán aprovechar al máximo las capacidades de los dispositivos móviles y brindar una experiencia de usuario excepcional a nuestros clientes.

Aspecto	Flutter	Swift
Lenguaje	Dart	Swift
Desarrollo	Framework de desarrollo multiplataforma	Lenguaje y framework específico para desarrollo iOS/macOS
UI	Basado en widgets personalizados	Utiliza Interface Builder y Storyboards para diseño de interfaces
Rendimiento	Alto rendimiento y velocidad de ejecución	Optimizado para dispositivos Apple, proporcionando un rendimiento excelente
Compatibilidad	Compatible con iOS y Android	Enfocado exclusivamente en plataformas Apple (iOS/macOS)
Documentación	Documentación extensa y actualizada	Amplia documentación disponible, con un enfoque específico en plataformas Apple
Curva de aprendizaje	Curva de aprendizaje empinada, especialmente si no estás familiarizado con Dart	Curva de aprendizaje moderada, especialmente si ya tienes experiencia en Swift
Comunidad	Comunidad en crecimiento con soporte activo	Gran comunidad y amplio soporte de la comunidad, especialmente para desarrollo iOS
Personalización	Permite una mayor personalización y diseño creativo	Proporciona opciones de personalización, pero con una estructura más orientada a estándares de Apple
Ecosistema	Menos maduro en comparación con Swift	Amplio ecosistema con muchas bibliotecas y herramientas disponibles para desarrollo iOS
Costo	Gratuito y de código abierto	Gratuito y de código abierto

Tabla 6: Comparación entre Flutter y Swift

Aspecto	Firestore	Otras bases de datos
Tipo	Base de datos en tiempo real y plataforma de desarrollo de aplicaciones móviles	Bases de datos relacionales y no relacionales
Escalabilidad	Automática y flexible	Depende de la configuración y hardware
Integración	Integrado con otros servicios de Firestore	Requiere configuración y administración separada
Almacenamiento	Almacenamiento de datos en memoria o en la nube	Almacenamiento en memoria, en disco o en la nube
Actualizaciones	Actualizaciones en tiempo real	Actualizaciones dependen del enfoque de la base de datos
Seguridad	Autenticación y seguridad integradas	Requiere implementación de seguridad personalizada
Costo	Modelo de precios basado en el uso	Costos varían según el proveedor y la configuración
Soporte	Soporte activo de Firestore y comunidad	Soporte varía según la base de datos elegida
Funcionalidades adicionales	Ofrece características como autenticación de usuarios, notificaciones push, almacenamiento en la nube, análisis de datos, etc.	Funcionalidades adicionales pueden variar según la base de datos

Tabla 7: Comparación entre Firestore y Otras BD

4.2.1 Planeación y Diseño

Para definir los criterios de aceptación entre el cliente y el equipo de desarrollo se plantea lo siguiente:

Objetivo del Plan de Aceptación: El objetivo del plan de aceptación es asegurar que la aplicación de finanzas personales cumpla con los requisitos definidos y las expectativas del cliente, brindando una funcionalidad completa, confiable y de alta calidad.

Procedimiento de Aceptación:

1. Definición de criterios de aceptación:

- Se establecerán criterios claros y medibles para evaluar el éxito y la satisfacción del proyecto.
- Los criterios pueden incluir la funcionalidad, el rendimiento, la usabilidad, la seguridad y la precisión de los algoritmos de clasificación con IA y OCR.
- Estos criterios serán compartidos con los stakeholders clave y se acordarán mutuamente.

2. Planificación de pruebas de aceptación:

- Se diseñará un conjunto exhaustivo de pruebas de aceptación para validar el cumplimiento de los criterios definidos.
- Las pruebas pueden incluir casos de uso, escenarios de prueba y pruebas funcionales para todas las funcionalidades de la aplicación.
- Se definirán los datos de prueba necesarios y se prepararán entornos de prueba adecuados para simular situaciones reales.

3. Ejecución de pruebas de aceptación:

- Se realizarán pruebas de aceptación en función del conjunto de pruebas definido previamente.
- Se registrarán y documentarán los resultados de las pruebas, incluyendo cualquier problema o desviación identificada.
- Se realizarán pruebas de regresión para asegurar que las correcciones o mejoras no afecten negativamente las funcionalidades existentes.

4. Revisión y validación:

- Se realizará una revisión exhaustiva de los resultados de las pruebas de aceptación para evaluar el cumplimiento de los criterios definidos.
- Se involucrarán a los stakeholders clave, como los usuarios finales y los expertos en finanzas, para validar las funcionalidades y proporcionar retroalimentación adicional.
- Se llevarán a cabo revisiones técnicas y funcionales para asegurar la calidad y la coherencia del sistema en general.

5. Resolución de problemas y mejoras:

- Se abordarán cualquier problema o desviación identificada durante las pruebas de aceptación.
- Se realizarán correcciones y mejoras según sea necesario para cumplir con los criterios de aceptación y las expectativas del cliente.
- Se repetirán las pruebas de aceptación en los casos relevantes para verificar que las soluciones implementadas sean efectivas.

6. Aprobación y entrega:

- Una vez que se hayan cumplido los criterios de aceptación y se haya abordado cualquier problema identificado, se solicitará la aprobación final del cliente y de los stakeholders clave.
- Se preparará la entrega del producto final, incluyendo la documentación relevante, los manuales de usuario y los recursos necesarios.

7. Seguimiento post-implementación:

- Se realizará un seguimiento después de la implementación para garantizar que la aplicación esté funcionando según lo esperado y satisfaciendo las necesidades de los usuarios.
- Se recopilará retroalimentación adicional y se realizarán ajustes o mejoras adicionales según sea necesario.

4.2.2 Desarrollo

La herramientas de desarrollo en el proyecto:

1. **GitHub:** GitHub es una empresa que le ofrece a sus usuarios un servicio de repositorio Git en el que almacena los datos en la nube. Su principal objetivo es facilitar el trabajo colaborativo y, por tanto, el control de versiones. [9]
2. **IntelliJ:** Este es un entorno de desarrollo integrado (IDE) diseñado para programar en Java y Kotlin. [17]
3. **Xcode:** Este es un entorno de desarrollo integrado (IDE) diseñado para programar en Swift y las tecnologías relacionadas. [18]

Así mismo, los lenguajes seleccionados se pueden observar a continuación.

- **UML:** En este lenguaje se realizará el modelado visual. Este es un lenguaje de modelado visual de propósito general que se utiliza para especificar, visualizar, construir y documentar los artefactos de un sistema de software. Captura decisiones y conocimiento sobre sistemas que deben ser comprendidos. [11]
- **Python:** En este lenguaje se realizará el modelo de detección de ruido con ayuda de la librería Tensorflow. Este es un lenguaje de programación orientado a objetos, flexible y de alto nivel. Es un lenguaje de propósito general, por lo tanto, se pueden crear desde modelos de aprendizaje automático hasta proyectos mucho más avanzados que utilicen la inteligencia artificial.
- **Swift:** Swift es un lenguaje de programación desarrollado por Apple Inc. y diseñado para el desarrollo de aplicaciones para sus plataformas, incluyendo iOS, macOS, watchOS y tvOS. Fue presentado por primera vez en la Conferencia Mundial de Desarrolladores de Apple (WWDC) en 2014 y se ha convertido en el principal lenguaje de programación para el desarrollo de aplicaciones en el ecosistema de Apple.
- **Java (Spring Boot):** En este lenguaje se realizará la implementación del *backend* del servicio de gestión de usuarios y el *API*. Este es un marco que tiene sus bases en el lenguaje de programación Java. Este permite a los usuarios desarrollar aplicaciones Spring independientes y que sea posible ejecutarlas.

4.3 Plan de Aceptación del Producto

Para definir los criterios de aceptación entre el cliente y el equipo de desarrollo se plantea lo siguiente:

Objetivo del Plan de Aceptación: El objetivo del plan de aceptación es asegurar que la aplicación de finanzas personales cumpla con los requisitos definidos y las expectativas del cliente, brindando una funcionalidad completa, confiable y de alta calidad.

Procedimiento de Aceptación:

1. Definición de criterios de aceptación:

- Se establecerán criterios claros y medibles para evaluar el éxito y la satisfacción del proyecto.
- Los criterios pueden incluir la funcionalidad, el rendimiento, la usabilidad, la seguridad y la precisión de los algoritmos de clasificación con IA y OCR.
- Estos criterios serán compartidos con los stakeholders clave y se acordarán mutuamente.

2. Planificación de pruebas de aceptación:

- Se diseñará un conjunto exhaustivo de pruebas de aceptación para validar el cumplimiento de los criterios definidos.
- Las pruebas pueden incluir casos de uso, escenarios de prueba y pruebas funcionales para todas las funcionalidades de la aplicación.
- Se definirán los datos de prueba necesarios y se prepararán entornos de prueba adecuados para simular situaciones reales.

3. Ejecución de pruebas de aceptación:

- Se realizarán pruebas de aceptación en función del conjunto de pruebas definido previamente.
- Se registrarán y documentarán los resultados de las pruebas, incluyendo cualquier problema o desviación identificada.
- Se realizarán pruebas de regresión para asegurar que las correcciones o mejoras no afecten negativamente las funcionalidades existentes.

4. Revisión y validación:

- Se realizará una revisión exhaustiva de los resultados de las pruebas de aceptación para evaluar el cumplimiento de los criterios definidos.
- Se involucrarán a los stakeholders clave, como los usuarios finales y los expertos en finanzas, para validar las funcionalidades y proporcionar retroalimentación adicional.
- Se llevarán a cabo revisiones técnicas y funcionales para asegurar la calidad y la coherencia del sistema en general.

5. Resolución de problemas y mejoras:

- Se abordarán cualquier problema o desviación identificada durante las pruebas de aceptación.
- Se realizarán correcciones y mejoras según sea necesario para cumplir con los criterios de aceptación y las expectativas del cliente.
- Se repetirán las pruebas de aceptación en los casos relevantes para verificar que las soluciones implementadas sean efectivas.

6. Aprobación y entrega:

- Una vez que se hayan cumplido los criterios de aceptación y se haya abordado cualquier problema identificado, se solicitará la aprobación final del cliente y de los stakeholders clave.
- Se preparará la entrega del producto final, incluyendo la documentación relevante, los manuales de usuario y los recursos necesarios.

7. Seguimiento post-implementación:

- Se realizará un seguimiento después de la implementación para garantizar que la aplicación esté funcionando según lo esperado y satisfaciendo las necesidades de los usuarios.
- Se recopilará retroalimentación adicional y se realizarán ajustes o mejoras adicionales según sea necesario.

4.4 Organización del Proyecto y Comunicación

4.4.1 Interfaces Externas

Dentro de las interfaces externas se consideran las siguientes:

1. Proveedores de servicios: Aquellos proveedores externos que suministran recursos, tecnología, materiales u otros elementos necesarios para la ejecución del proyecto tales como AWS, Google Cloud, entre otros.
2. Sistemas de terceros: Otros sistemas o software con los cuales el proyecto necesita integrarse o interactuar para el intercambio de datos o la interoperabilidad, dentro de estos se incluyen los necesarios para el funcionamiento de OCR dentro del aplicativo.
3. Regulaciones y estándares de la industria: Normativas, leyes o estándares específicos de la industria que deben ser cumplidos o considerados en el desarrollo y ejecución del proyecto. Tales como la protección de datos, la autenticación y el uso del software del dispositivo.
4. Usuarios finales o clientes: Aquellos individuos o entidades que utilizarán o se beneficiarán del producto o servicio final del proyecto.
5. Organismos reguladores: Entidades o instituciones gubernamentales u organizaciones externas que tienen la autoridad para supervisar o regular aspectos relacionados con el proyecto.

4.4.2 Organigrama y Descripción de Roles

A continuación, se presenta el organigrama del proyecto.



Figura 2: Organigrama del Equipo

Rol	Encargado	Descripción
Scrum Master	Angie Tatiana Peña	Un Scrum Master es el facilitador del proceso Scrum en un equipo ágil. Sus responsabilidades incluyen garantizar que el equipo comprenda y siga las prácticas de Scrum, eliminar obstáculos que puedan afectar la productividad, apoyar al equipo en la adopción de prácticas ágiles, promover la mejora continua y proteger al equipo de interferencias externas. Su objetivo principal es ayudar al equipo a trabajar de manera efectiva y alcanzar sus metas.
Product Owner	Santiago Vasquez Sánchez	Un Product Owner es el responsable de maximizar el valor del producto y representar los intereses de los stakeholders. Sus responsabilidades incluyen definir y priorizar las funcionalidades del producto, gestionar el backlog, colaborar con el equipo de desarrollo, tomar decisiones sobre el producto y garantizar su éxito comercial. Su objetivo principal es asegurar que el producto cumpla con las necesidades y expectativas de los usuarios y genere valor para el negocio.
Solution Lead	Cristobal Balcázar Castrillon	Un Solution Lead es el líder de soluciones en un proyecto o equipo. Sus responsabilidades incluyen proporcionar dirección técnica y estratégica, diseñar y arquitectar soluciones, colaborar con los stakeholders para comprender sus necesidades, coordinar y supervisar la implementación de soluciones, y asegurar la calidad y el cumplimiento de los requisitos técnicos. Su objetivo principal es liderar el equipo técnico y garantizar la entrega exitosa de soluciones eficientes y efectivas.
Developers	Angie Tatiana Peña, Cristobal Castrillon Balcázar y Santiago Vasquez Sánchez	Un desarrollador es un profesional encargado de escribir, probar y mantener el código de software. Sus responsabilidades incluyen comprender los requisitos del proyecto, diseñar soluciones técnicas, escribir código limpio y eficiente, realizar pruebas y depuración, colaborar con otros miembros del equipo y seguir las mejores prácticas de desarrollo. Su objetivo principal es contribuir a la creación de software funcional y de alta calidad que cumpla con los requisitos del proyecto.

Tabla 8: Descripción de los roles

5 Administración del Proyecto

5.1 Métodos y Herramientas de Estimación

Teniendo en cuenta la división de los requisitos en historias con calificación cuantitativa en tiempo de ejecución y desarrollo, se proponen las siguientes herramientas para su estimación:

1. **Planning Poker:** Herramienta basada en la participación colaborativa del equipo, fomentando la discusión y la toma de decisiones conjuntas. Evalúa el tamaño relativo de las tareas o historias en lugar de estimaciones precisas en horas, promoviendo un enfoque ágil y flexible.
2. **Estimación en T-shirt sizes:** Proporciona una estimación rápida y visualmente intuitiva del tamaño relativo de las tareas utilizando tamaños de camiseta en lugar de valores numéricos. Útil para la planificación inicial sin invertir demasiado tiempo en detalles precisos.
3. **Estimación en puntos de historia:** Se basa en la estimación relativa, comparando las tareas con una historia de referencia conocida asignada a un número determinado de puntos de historia. Facilita la creación de un lenguaje común de estimación en el equipo.
4. **Estimación en horas ideales:** Permite estimar el esfuerzo requerido para cada tarea en un entorno ideal sin interrupciones, proporcionando una comprensión detallada del esfuerzo y la planificación inicial del proyecto.
5. **Estimación basada en expertos:** Basada en el conocimiento y experiencia de expertos en el campo del proyecto, utilizando la técnica Delphi para obtener estimaciones imparciales y precisas.

5.2 Inicio del Proyecto

5.2.1 Entrenamiento del Personal

El objetivo es capacitar al personal en el uso de las herramientas tecnológicas necesarias. Se proponen las siguientes actividades:

1. Identificación de las herramientas tecnológicas, como Android Studio, TensorFlow, servicios en la nube, Git, JIRA, Slack, entre otros.
2. Evaluación de las necesidades de entrenamiento, identificando las habilidades técnicas actuales, brechas de conocimiento y roles específicos que requieren capacitación.
3. Diseño del programa de entrenamiento, definiendo objetivos de aprendizaje, creando módulos prácticos y considerando opciones de capacitación.
4. Planificación de sesiones de entrenamiento, estableciendo un calendario, asignando instructores con experiencia y proporcionando materiales de capacitación.
5. Implementación del entrenamiento, llevando a cabo las sesiones y asegurando la participación activa.
6. Práctica y aplicación de conocimientos, proporcionando oportunidades para aplicar los conocimientos adquiridos en proyectos reales.
7. Evaluación del entrenamiento, realizando evaluaciones periódicas y ajustando el programa según sea necesario.

5.2.2 Infraestructura

Para la infraestructura del proyecto se propone:

1. **Servidores y almacenamiento:** Configuración de servidores físicos o virtuales, implementación de arquitectura escalable, configuración de herramientas de monitoreo y uso de servicios de almacenamiento en la nube.
2. **Plataformas de desarrollo:** Configuración de entornos de desarrollo integrados (IDE), implementación de sistemas de control de versiones.
3. **Bases de datos:** Elección y configuración de una base de datos adecuada, implementación de sistemas de respaldo y recuperación ante desastres.
4. **Infraestructura en la nube:** Utilización de servicios en la nube como AWS, GCP o Azure, configuración de instancias virtuales, implementación de servicios de escalado automático.
5. **Seguridad:** Configuración de firewalls, certificados SSL/TLS, implementación de políticas de seguridad y acceso basadas en roles.
6. **Monitoreo y gestión:** Configuración de herramientas de monitoreo y gestión de logs, implementación de soluciones de gestión de incidentes y notificaciones.

5.3 Planes de Trabajo del Proyecto

5.3.1 Descomposición de Actividades

La descomposición de actividades en tareas más pequeñas facilita la gestión y el seguimiento del proyecto. Las tareas manejables permiten un mejor control y estimación de los recursos y el tiempo requeridos.

1. **Identificación de las actividades principales:** Identificar las actividades necesarias para completar el proyecto, como análisis de requisitos, diseño de interfaz, desarrollo del módulo de IA, integración de OCR, pruebas y despliegue.
2. **Desglose de actividades:** Descomponer cada actividad principal en tareas más pequeñas y manejables, por ejemplo, entrevistas con usuarios, recopilación de requisitos, elaboración de casos de uso, etc.
3. **Secuenciación de actividades:** Determinar la secuencia lógica de las actividades, identificando dependencias entre ellas para establecer un orden adecuado.
4. **Asignación de recursos:** Evaluar los recursos necesarios para cada tarea y asignar los recursos adecuados en función de la experiencia y disponibilidad del personal.
5. **Estimación de duración:** Estimar la duración de cada tarea en función de la complejidad, esfuerzo y disponibilidad de recursos.
6. **Priorización de actividades:** Identificar las actividades críticas y priorizarlas para asegurar un enfoque adecuado y asignación de recursos.
7. **Creación del diagrama de Gantt:** Utilizar un diagrama de Gantt para visualizar las actividades, su secuencia y duración, facilitando la coordinación y comunicación del plan del proyecto.

5.3.2 Calendarización

Para la calendarización se propone:

1. **Identificación de las actividades y duraciones estimadas:** Revisar la descomposición de actividades y estimaciones de duración para tener una lista completa de tareas y tiempos estimados.
2. **Secuenciación de actividades:** Establecer la secuencia de actividades teniendo en cuenta las dependencias y la posibilidad de realizar tareas simultáneamente.
3. **Creación del calendario:** Utilizar una herramienta de gestión de proyectos o hoja de cálculo para asignar fechas de inicio y finalización estimadas a cada actividad.
4. **Ajuste de fechas:** Considerar días no laborables y ajustar las fechas de las actividades según las limitaciones de recursos y los plazos del proyecto.
5. **Establecimiento de hitos:** Identificar hitos importantes en el proyecto y asignar fechas específicas en el calendario para monitorear el progreso.
6. **Revisión y ajuste:** Revisar y ajustar el calendario según sea necesario, manteniendo un equilibrio entre la planificación y la flexibilidad.
7. **Comunicación y seguimiento:** Compartir el calendario con el equipo y otras partes interesadas, realizar un seguimiento regular del progreso y realizar ajustes según sea necesario.

5.3.3 Asignación de Recursos

La asignación de recursos debe ser flexible y ajustarse según sea necesario. Se proponen las siguientes actividades:

1. **Identificación de los recursos necesarios:** Revisar actividades y requisitos del proyecto para identificar los recursos necesarios, como personal, equipos, herramientas, software, etc.
2. **Evaluación de la disponibilidad de recursos:** Evaluar la disponibilidad de recursos necesarios y determinar si es necesario adquirirlos o asignarlos.
3. **Asignación de personal:** Identificar habilidades necesarias y asignar personal considerando sus habilidades, experiencia y disponibilidad.
4. **Adquisición de herramientas y software:** Identificar herramientas y software necesarios, adquirirlos mediante compra o configuración según sea necesario.
5. **Establecimiento de fechas de asignación:** Establecer fechas de inicio y finalización para la asignación de recursos, considerando las fechas de inicio de actividades y dependencias.
6. **Gestión de recursos externos:** Identificar recursos externos o proveedores de servicios, establecer acuerdos o contratos y coordinar su participación en el proyecto.
7. **Seguimiento y ajuste:** Realizar un seguimiento continuo de la asignación de recursos, ajustar según sea necesario considerando cambios en requisitos, disponibilidad y restricciones presupuestarias.

5.4 Asignación de Presupuesto y Justificación

1. **Identificación de los costos del proyecto:** Revisar todas las actividades y recursos necesarios para determinar los costos asociados, incluyendo salarios, adquisiciones, contratación de recursos externos, infraestructura, entre otros.
2. **Estimación de los costos:** Realizar una estimación detallada de los costos para cada actividad y recurso identificado, utilizando información histórica, cotizaciones y métodos de investigación.
3. **Consolidación de los costos:** Sumar todos los costos estimados para obtener el costo total del proyecto, incluyendo un margen adicional para imprevistos.
4. **Asignación de presupuesto a nivel de actividades:** Asignar un presupuesto específico para cada actividad considerando la prioridad y la importancia de las tareas.
5. **Asignación de presupuesto a nivel de recursos:** Asignar presupuesto a cada recurso identificado, asegurando suficientes fondos para cubrir los costos asociados durante todo el proyecto.

5.4.1 Justificación de la Asignación de Presupuesto

La asignación de presupuesto se justifica en base a los siguientes aspectos:

1. **Cumplimiento de los objetivos del proyecto:** El presupuesto asignado garantiza recursos para llevar a cabo todas las actividades del proyecto y lograr los objetivos establecidos.
2. **Calidad y eficiencia:** Un presupuesto adecuado permite la adquisición de herramientas, software y recursos externos de calidad, contribuyendo a la eficiencia en la ejecución del proyecto.
3. **Gestión de riesgos:** La asignación de un margen adicional en el presupuesto ayuda a mitigar posibles riesgos y contingencias, asegurando recursos para situaciones imprevistas.
4. **Competitividad y posicionamiento en el mercado:** La inversión en tecnologías como la IA y el OCR puede proporcionar ventajas competitivas, justificando la asignación de presupuesto.
5. **Rentabilidad a largo plazo:** La asignación adecuada de presupuesto garantiza una ejecución efectiva, generando beneficios económicos y financieros a largo plazo.

6 Monitoreo y Control del Proyecto

6.1 Administración de Requerimientos

La administración de los requerimientos es un proceso iterativo y continuo. A medida que el proyecto avanza, es posible que surjan nuevos requerimientos o cambios en los existentes. Por lo tanto, es crucial gestionar estos cambios de manera efectiva y asegurarse de que los requerimientos se mantengan alineados con las necesidades del proyecto y los stakeholders.

1. **Identificación de Requerimientos:** En colaboración con los stakeholders relevantes, como usuarios, clientes y expertos en el dominio, identificar y comprender completamente los requerimientos del proyecto utilizando técnicas como entrevistas, cuestionarios y talleres.
2. **Recopilación de Información:** Utilizar herramientas como entrevistas, cuestionarios y talleres para recopilar información y capturar los requerimientos de manera clara y concisa.
3. **Registro y Documentación:** Registrar y documentar todos los requerimientos identificados en un formato estructurado y comprensible.
4. **Herramientas de Documentación:** Utilizar herramientas como una matriz de requerimientos, una lista de verificación o un documento de especificaciones funcionales para capturar detalles de cada requerimiento.
5. **Análisis de Viabilidad:** Evaluar y analizar los requerimientos identificados para comprender su viabilidad técnica y su alineación con los objetivos del proyecto.
6. **Priorización:** Priorizar los requerimientos según su importancia, urgencia y valor para el cliente o usuario final.

6.1.1 Gestión de Cambios en Requerimientos

1. **Establecimiento de Proceso:** Establecer un proceso para gestionar los cambios en los requerimientos a lo largo del proyecto.
2. **Control de Cambios:** Implementar un sistema de control de cambios para evaluar y aprobar cualquier cambio propuesto, asegurando que estén alineados con los objetivos y restricciones del proyecto.

6.1.2 Comunicación y Seguimiento

1. **Comunicación Efectiva:** Comunicar de manera efectiva los requerimientos a todos los miembros del equipo del proyecto y a los stakeholders relevantes.
2. **Registro Actualizado:** Mantener un registro actualizado de los requerimientos y su estado, asegurando que todos los cambios y actualizaciones sean comunicados y entendidos por todos los involucrados.

6.1.3 Validación y Verificación de Requerimientos

1. **Actividades de Validación:** Realizar actividades de validación para asegurar que los requerimientos capturados cumplen con las expectativas de los stakeholders.
2. **Técnicas de Verificación:** Utilizar técnicas como revisiones técnicas, pruebas de aceptación y demostraciones del producto para validar y verificar que los requerimientos se han implementado correctamente.

6.1.4 Mantenimiento de Trazabilidad

Establecer y mantener la trazabilidad de los requerimientos a lo largo del ciclo de vida del proyecto.

6.2 Monitoreo y Control de Progreso

Para el monitoreo y control del proyecto, se proponen las siguientes actividades:

1. **Definición de Métricas y Criterios de Desempeño:** Garantizar la calidad del producto mediante la definición de métricas y criterios de desempeño.
2. **Recopilación de Datos y Seguimiento del Progreso:** Recopilar datos y realizar un seguimiento del progreso para identificar fallas en tiempo oportuno.
3. **Comparación con Objetivos Establecidos:** Comparar los resultados con los objetivos establecidos para identificar fallas, oportunidades, amenazas y oportunidades.
4. **Análisis de Desviaciones:** Analizar las desviaciones identificadas y evaluar su impacto en el proyecto.
5. **Acciones Correctivas y Preventivas:** Tomar acciones correctivas y preventivas según sea necesario para abordar desviaciones y garantizar el logro de los objetivos.
6. **Comunicación y Reporte Regular:** Comunicar y reportar regularmente el progreso a todos los interesados, asegurando una transparencia adecuada.
7. **Actualización y Ajuste del Plan:** Actualizar y ajustar el plan del proyecto según sea necesario en función de los cambios en el entorno y los resultados obtenidos.

6.3 Toma de Decisiones con Análisis DOFA

Para la toma de decisiones, se propone el uso del análisis DOFA, que se evidencia en la figura 3. El análisis DOFA involucra la identificación de Fortalezas, Oportunidades, Debilidades y Amenazas para informar decisiones estratégicas y tácticas del proyecto.

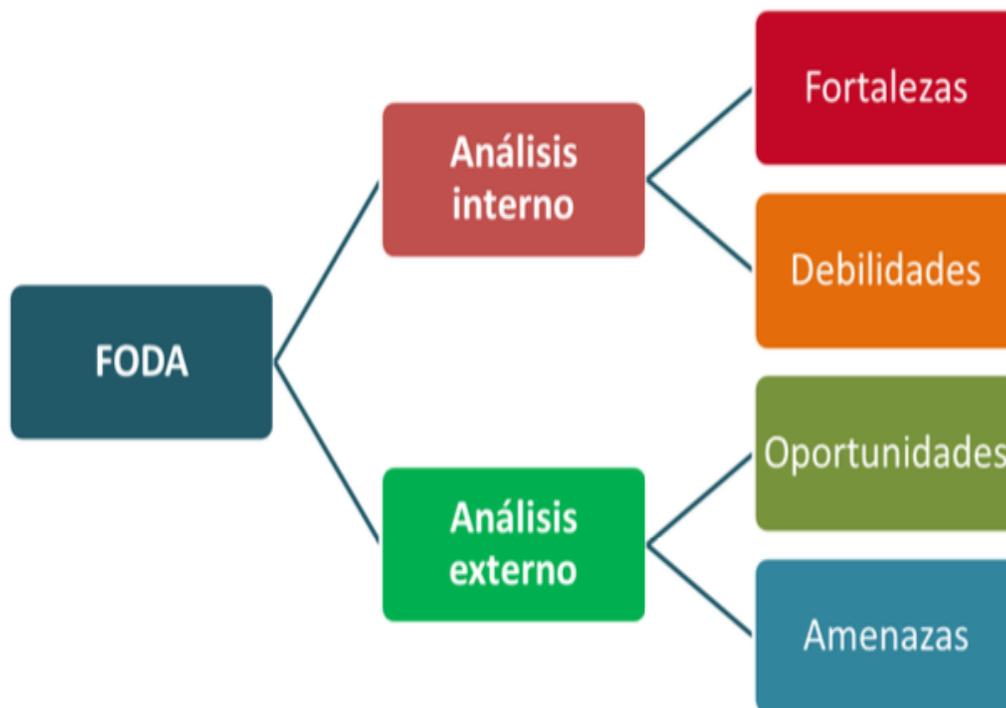


Figura 3: Matriz DOFA

6.3.1 Cierre del Proyecto

Para finalizar el proyecto, es necesario que todos los miembros del equipo entreguen la última versión de los documentos generados a lo largo del proyecto, así como el repositorio de GitHub con el código fuente final tanto del modelo como del prototipo móvil funcional y su integración con el modelo.

7 Entrega del Producto

7.1 Producto Mínimo Viable (MVP)

7.1.1 Entrega del Producto Mínimo Viable (MVP)

Para la entrega del Producto Mínimo Viable (MVP), se definen los siguientes aspectos:

1. **Versión funcional del MVP:** Entrega de la versión funcional del MVP en una fecha específica al final del semestre.
2. **Equipo de desarrollo y proyecto:** Identificación del equipo responsable y del proyecto.

Artefacto a Entregar	Entrega	Responsable
Versión funcional del MVP	Fecha específica al final del semestre	Equipo de desarrollo y proyecto

Tabla 9: Entrega del Producto Mínimo Viable (MVP)

7.1.2 Documentación del MVP

1. **Artefacto a Entregar:** Documentación del MVP que incluye descripción de funcionalidades, guía de usuario y requisitos técnicos.
2. **Entrega:** Antes de la fecha de entrega del MVP.
3. **Responsable:** Equipo de desarrollo y proyecto.

Artefacto a Entregar	Entrega	Responsable
Documentación del MVP	Antes de la fecha de entrega del MVP	Equipo de desarrollo y proyecto

Tabla 10: Documentación del MVP

7.1.3 Capacitación y Transición

Artefacto a Entregar	Entrega	Responsable
Programa de capacitación y transición	Antes de la fecha de implementación del sistema	Equipo de capacitación y soporte

Tabla 11: Capacitación y Transición

7.1.4 Entrega de Manuales

Artefacto a Entregar	Entrega	Responsable
Manuales de usuario y manuales técnicos del sistema	Junto con la implementación del sistema	Equipo de documentación y soporte

Tabla 12: Entrega de Manuales

8 Soporte

8.1 Ambiente de Trabajo

Las reglas generales que el equipo tendrá en cuenta son:

Normas	Descripción
Puntualidad en reuniones	Cumplir con las fechas y horas establecidas previamente. Si no se puede asistir a una reunión, anunciar con antelación las razones para su falta.
Comunicación adecuada con los demás compañeros	Usar un lenguaje adecuado y respetuoso con el grupo y usar los medios de comunicación establecidos.
Cumplir con fechas programadas	Entregar a tiempo los trabajos asignados y asistir a las reuniones programadas.
Democracia en las decisiones	Tener en cuenta todas las opiniones de los integrantes y realizar votaciones cuando son necesarias.
Expresar inconformidades y necesidades al grupo	Comunicarse de manera asertiva y respetuosa con los integrantes, debatiendo y llegando a un acuerdo con las inconformidades expresadas.

Tabla 13: Reglas de Trabajo

Las faltas que se tendrán en cuenta serán:

Normas	Descripción
Incumplimiento con el trabajo asignado. (Se determina en <i>Notion</i> - Leve)	Aumentar la carga de trabajo en la siguiente entrega. Para esto se manejará un <i>Notion</i> con las distintas actividades que deberá llevar a cabo cada integrante.
Incumplimiento frecuente con las entregas. (Se determina por las actas de trabajo - Grave)	Se procederá a hablar directamente con la profesora María Emilia.
Comportamiento inadecuado con los compañeros. (Grave)	Programar una reunión con los compañeros para realizar nuevos acuerdos, en caso tal de continuar con el comportamiento se hablará directamente con la profesora María Emilia.

Tabla 14: Faltas y Sanciones

Finalmente, los causales de expulsión que se tendrán en cuenta son:

- Comportamiento inadecuado con los demás compañeros de forma persistente. Reincidencia luego de haber hablado con las profesoras.
- Incumplimiento a todas las actividades asignadas sin justa causa.
- Comunicación nula con los demás integrantes del equipo sin justa causa.

8.2 Análisis y Administración de Riesgos

Actualmente, se han logrado identificar los siguientes riesgos:

Al ser estos los riesgos principalmente identificados, estos son los riesgos más importantes que actualmente el proyecto posee.

Riesgo	Impacto	Prevención	Prioridad
Insuficiencia de datos para entrenar el modelo	En caso tal de no tener los suficientes datos, el modelo no va a poder ser entrenado, por ende, todo el proyecto se estancaría.	Asegurarse de que el ente investigativo tenga los suficientes datos. Adicional a esto, buscar más datos en repositorios encontrados en Internet.	Alta
Poco tiempo de entrenamiento	Debido a que el entrenamiento puede llegar a durar hasta más de una semana, puede que el proyecto no se logre completar en su totalidad.	Desarrollar adecuadamente la calendarización y el equipo de trabajo debe ser cumplido con las fechas de entrega	Alta

Tabla 15: Faltas y Sanciones

8.3 Métricas y Proceso de Medición

El plan de análisis y administración de riesgos tiene como objetivo identificar y evaluar proactivamente los posibles riesgos que pueden afectar al proyecto. Se realizará una evaluación exhaustiva de los riesgos, priorizando según su nivel e impacto, y desarrollando planes de mitigación.

8.3.1 Administración de Configuración y Documentación

El plan de administración de configuración y documentación busca establecer un proceso eficiente y estructurado para gestionar la configuración del proyecto y garantizar la integridad y trazabilidad de los elementos y documentos relacionados. Se implementará un sistema de control de versiones y se establecerán roles y responsabilidades claros.

8.3.2 Métricas y Procesos de Medición

Se definirán métricas claras y objetivas para cuantificar el avance, el cumplimiento de plazos, la satisfacción del cliente y la calidad del producto. Se utilizarán herramientas y técnicas apropiadas para recopilar y analizar datos, generando informes periódicos.

8.4 Control de Calidad

El plan de control de calidad tiene como objetivo garantizar que los entregables y productos del proyecto cumplan con los estándares de calidad establecidos. Se llevarán a cabo actividades de inspección, pruebas y acciones correctivas y preventivas para mejorar la calidad y evitar problemas recurrentes. La mejora continua será fundamental, con la participación activa de todo el equipo.

Proceso	Momentos	Responsables
Revisión en el centro de escritura de calidad de la documentación del proyecto	Cuando se finalice un documento a entregar	Todos los integrantes del equipo
Validación y aprobación de código a implementar	Al finalizar un requerimiento asignado.	Todos los integrantes del equipo

Tabla 16: Métricas y proceso de medición

9 Referencias

- [1] Andreou, P. C., & Anyfantaki, S. (2021). Financial literacy and its influence on internet banking behavior. *European Management Journal*, 39(5), 658-674. Recuperado de: <https://doi.org/10.1016/j.emj.2020.12.001>
- [2] Angel, S. (2018). Smart tools? A randomized controlled trial on the impact of three different media tools on personal finance. *Journal of Behavioral and Experimental Economics*, 74, 104-111. [Online] Recuperado de: <https://doi.org/10.1016/j.socec.2018.04.002>
- [3] GAYOSO ROJAS, Y. J., MARTINEZ OLIVA, R. A. O., MUNDACA BENAVIDES, E. J., VELASQUEZ, P., ROLLER, J., REYES CASTRO, G. A., SANCHEZ HERRERA, J., & SANTISTEBAN QUIROZ, J. P. (s. f.). EL FUTURO DE LA INTELIGENCIA ARTIFICIAL. Mahapatra, P., & Singh, S. K. (2021). *Artificial Intelligence and Machine Learning: Discovering New Ways of Doing Banking Business*. CRC Press. Recuperado de: <https://doi.org/10.1201/9781003125129-4>
- [4] Memon, J., Sami, M., Khan, R. A., & Uddin, M. (2020). Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR). *IEEE Access*, 8, 142642-142668. Recuperado de: <https://doi.org/10.1109/ACCESS.2020.3012542>
- [5] Valenzuela, C. G. (2022, December 24). *¿Qué es latex y cómo funciona esta útil herramienta para crear documentos?* *Computer Hoy*. Retrieved March 13, 2023, from <https://computerhoy.com/tecnologia/latex-como-funciona-util-herramienta-crear-documentos-1165366>
- [6] “Todo lo que necesitas saber sobre Microsoft TEAMS,” *Ibermatica*.
- [7] R. Orellana, “Qué es WhatsApp, para qué sirve y por qué es tan popular,” Nov. 12, 2021. <https://es.digitaltrends.com/celular/que-es-whatsapp/> (accessed Oct. 27, 2022).
- [8] “What Is GitHub? A Beginner’s Introduction to GitHub,” Sep. 2022. <https://kinsta.com/knowledgebase/what-is-github/> (accessed Oct. 27, 2022).
- [9] F. Flores, “Qué es Visual Studio Code y qué ventajas ofrece,” *OpenWebinars*, Jul. 22, 2022. <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/> (accessed Oct. 27, 2022).
- [10] “¿Qué es el lenguaje unificado de modelado (UML)?,” *Lucidchart*. <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml> (accessed Oct. 27, 2022).
- [11] “¿Qué es Python?,” *Datademia*. <https://datademia.es/blog/que-es-python> (accessed Oct. 27, 2022).
- [12] “Spring Boot - Introduction,” *Tutorials point*. https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm (accessed Oct. 27, 2022).
- [13] “Angular.” <https://desarrolloweb.com/home/angular> (accessed Oct. 27, 2022).
- [14] M. F. Aguirre, “Planning Poker: metodología scrum para una estimación ágil,” Nov. 18, 2020. <https://www.appvizer.es/revista/organizacion-planificacion/gestion-proyectos/planning-poker> (accessed Nov. 12, 2022).
- [15] University of Guadalajara, “Tipos de cronogramas,” *Course Hero*. <https://www.coursehero.com/file/137968934/Tipos-de-cronogramasdocx/> (accessed Nov. 12, 2022).

- [16] *UA.* (n.d.). Retrieved March 14, 2023, from <https://economicas.ua.es/es/documentos/miro-pizarra-colaborativa.pdf>
- [17] *Funcionalidades - IntelliJ idea. JetBrains.* (n.d.). Retrieved March 13, 2023, from <https://www.jetbrains.com/es-es/idea/features/>
- [18] *Xcode - Apple.* (s.f.). Recuperado de <https://developer.apple.com/xcode/>