

Control de documento

Nombre del proyecto	LensFrame by Swiftcode
Cierre de iteración	I-06 20/10/23
Generado por	Ivan Osmar Vasquez Flores Marcos Eduardo Solis Ceniceros
Aprobado por	Oswaldo Alfonso Hernandez Campos Carlos Francisco Babún Ravelo
Alcance de la distribución del documento	Control interno para todo el proyecto.

## **Índice**

<b>Sobre este documento</b>	<b>3</b>
<b>Resumen de la Iteración.</b>	<b>4</b>
Identificación.	4
Hitos especiales	5
Artefactos y evaluación	8
Riesgos y problemas.	10
Notas y observaciones	12
<b>Asignación de recursos</b>	<b>13</b>
Anexos	14
Glosario de términos	20
Significado de los elementos de la notación gráfica	20

## **Sobre este documento**

La calidad se logra por medio de la revisión constante de las actividades que conducen desde la idea al producto. Al momento del cierre de una iteración es buen momento para hacer un alto, y evaluar lo logrado, los problemas encontrados y los retos a enfrentar.

El presente documento marca el final de la iteración [\[I - 06\]](#), y contiene una evaluación de los artefactos y actividades realizadas durante la misma.

Se recogen también las impresiones y observaciones hechas durante el desarrollo de la iteración, así como el esfuerzo invertido en cada una de las disciplinas involucradas.

Selección de la plataforma en la cual el desarrollo del proyecto se está llevando a cabo.

## Resumen de la Iteración.

### Identificación.

Código de la iteración	Fase a la que pertenece	Fecha de inicio	Fecha de cierre	Comentarios
I - 06	INICIO	16 oct 2023	20 oct 2023	<p>Selección de Plataforma:</p> <p>Se decide en qué plataforma deseas lanzar la aplicación. Las opciones comunes incluyen aplicaciones móviles para smartphones y tablets, aplicaciones web basadas en AR.</p> <p>Desarrollo de Software:</p> <p>Elige una plataforma de desarrollo adecuada, como: Unity.</p>

## Hitos especiales

1. **Selección de Plataforma Confirmada:** Marca la finalización de la decisión sobre la plataforma en la que se lanzará la aplicación. Puede ser una aplicación móvil para smartphones, tablets o una aplicación web basada en AR. Esto implica una evaluación detallada de las necesidades y preferencias de los usuarios finales. Con el fin de optimizar los tiempos de desarrollo, se concede prioridad al desarrollo de la aplicación para dispositivos móviles con sistema operativo Android.
2. **Configuración de Entorno de Desarrollo en Unity:** Implica la configuración exitosa del entorno de desarrollo en Unity, lo que incluye la instalación de Unity y otras herramientas necesarias para el desarrollo de aplicaciones de realidad aumentada. Es importante destacar que la decisión de utilizar Unity como plataforma de desarrollo está sujeta a cambios según las condiciones y necesidades del desarrollo.
3. **Evaluación de Recursos y Pruebas de Integración en la Plataforma Elegida:** Contempla la evaluación de los recursos necesarios para el desarrollo en la plataforma seleccionada, lo cual incluye considerar el hardware específico, el software adicional, y el personal de desarrollo requeridos. Además, implica la realización de pruebas de integración para asegurarse de que la aplicación funcione de manera efectiva en la plataforma seleccionada, ya sea en dispositivos móviles o en un navegador web basado en AR.
4. **Compatibilidad con Hardware y Optimización de Rendimiento:** Abarca el avance en la implementación de una API de cámara unificada y el ajuste de la aplicación para adaptarse a una variedad de dispositivos, incluidas las cámaras integradas en teléfonos móviles. Además, se incluye el desarrollo de controladores y bibliotecas universales para probar una integración fluida en múltiples plataformas y sistemas operativos. Asimismo, este hito marca la fase de optimización para garantizar que la aplicación funcione de manera eficiente y sin problemas en la plataforma seleccionada.
5. **Desarrollo y Evolución del Prototipo Inicial:** Engloba la creación del prototipo inicial en Unity para visualizar la funcionalidad básica de la aplicación en la plataforma seleccionada. Además, incluye la realización de pruebas iniciales de funcionalidad y las subsiguientes fases de ajustes y mejoras con el fin de verificar que el prototipo cumple con los requisitos básicos, las expectativas y se adapta a las necesidades del proyecto.

## Artefactos y evaluación

Artefacto	Aspecto a evaluar	Evaluación %	Comentarios
ICF-21	Informe de Selección de Plataforma	90%	Llevar a cabo el proceso de selección de plataforma para la aplicación, con un énfasis en la prioridad de desarrollo para dispositivos móviles con sistema operativo Android. Se ha recopilado información preliminar sobre las necesidades y preferencias de los usuarios finales, y se menciona la consideración de utilizar Unity como plataforma de desarrollo. La decisión sobre la plataforma está sujeta a cambios según las condiciones y necesidades del desarrollo.

Artefacto	Aspecto a evaluar	Evaluación %	Comentarios
ICF-22	Compatibilidad con hardware	80%	Asegurarse de que la aplicación sea compatible con una amplia gama de dispositivos esto es esencial para llegar a la mayor audiencia posible. Esto implica optimizar la aplicación para funcionar en dispositivos móviles en este caso el principal es Android.

Artefacto	Aspecto a evaluar	Evaluación %	Comentarios
ICF-23	Informe de Optimización de Rendimiento	75%	Realizar la optimización de rendimiento de la aplicación. Se están tomando medidas iniciales para mejorar la eficiencia y el rendimiento en la plataforma seleccionada. Se planifican pruebas de rendimiento adicionales.

Artefacto	Aspecto a evaluar	Evaluación %	Comentarios
ICF-24	Prototipo Inicial en Unity	50%	Comenzar con el desarrollo del prototipo inicial en Unity. Se espera lograr la realización de las funcionalidades básicas del software y se realizarán pruebas iniciales. Existe disposición general de realizar ajustes y mejoras.

## Riesgos y problemas.

**ICF-21** La compatibilidad con hardware en una aplicación se refiere a la capacidad de la aplicación para funcionar correctamente en diferentes dispositivos utilizados por los clientes.

Algunos riesgos y problemas que podrían surgir si la compatibilidad con hardware no se maneja adecuadamente:

- **1. Funcionamiento inconsistente:** Si la aplicación no está optimizada para diferentes tipos de dispositivos y sistemas operativos, podría experimentar problemas de rendimiento o funcionamiento inconsistente. Esto puede llevar a una experiencia de usuario deficiente y la posible pérdida de clientes.
- **2. Problemas de visualización y resolución:** Diferentes dispositivos tienen pantallas con resoluciones y tamaños variados. Si la aplicación no está diseñada para adaptarse correctamente a estas diferencias, podría haber problemas de visualización, como texto o imágenes cortadas o desalineadas.
- **3. Rendimiento lento:** Los dispositivos más antiguos o menos potentes pueden experimentar problemas de rendimiento si la aplicación es demasiado exigente en términos de recursos. Esto puede resultar en tiempos de carga lentos o incluso bloqueos de la aplicación.
- **4. Falta de acceso a funciones importantes:** Si la aplicación está diseñada para aprovechar características específicas de ciertos dispositivos (por ejemplo, el escaneo de códigos de

barras con la cámara de un teléfono), los usuarios que no tienen acceso a esas características pueden perder funcionalidades importantes.

**Riesgo Asociado: RIE-18(Problemas de rendimiento/Escalabilidad)**

**RIE-18:** Este riesgo implica que la solución actual puede no ser capaz de manejar la carga de trabajo o el volumen de datos que se espera o requerido en un momento dado.

Puede resultar en tiempos de respuesta lentos, caídas del sistema o una disminución significativa en la eficiencia operativa. Es esencial abordar este riesgo para garantizar que el sistema pueda crecer y funcionar de manera eficaz a medida que las demandas aumentan.

**ICF-22:** Cuando se trata de la compatibilidad con diferentes dispositivos para una aplicación de óptica, pueden surgir varios riesgos y problemas potenciales como lo pueden ser los siguiente que son los más probables o comunes que sucedan:

- 1. Diferentes tamaños y resoluciones de pantalla: Las aplicaciones deben adaptarse a una variedad de tamaños de pantalla y resoluciones para garantizar una experiencia de usuario consistente.
- 2. Rendimiento del hardware: Dispositivos con hardware más antiguo o menos potente pueden tener dificultades para ejecutar la aplicación de manera eficiente, lo que puede resultar en una experiencia de usuario lenta o entrecortada.
- 3. Compatibilidad con versiones de software antiguas: La aplicación debe ser compatible con versiones anteriores de software para llegar a una audiencia más amplia, pero esto puede presentar desafíos al aprovechar las últimas características y tecnologías.
- 4. Actualizaciones y parches de seguridad: Garantizar que la aplicación reciba actualizaciones y parches de seguridad a tiempo en todos los dispositivos es fundamental para mantener la seguridad y la funcionalidad.

Es importante llevar a cabo un proceso de desarrollo y prueba cuidadoso para garantizar que la aplicación funcione de manera óptima en una amplia variedad de dispositivos y escenarios.

**Riesgo Asociado:RIE-18(Problemas de rendimiento/Escalabilidad), RIE-13(Dispositivo obsoleto del cliente) RIE-18:**

Este riesgo implica que la solución actual puede no ser capaz de manejar la carga de trabajo o el volumen de datos que se espera o requerido en un momento dado. Puede



resultar en tiempos de respuesta lentos, caídas del sistema o una disminución significativa en la eficiencia operativa. Es esencial abordar este riesgo para garantizar que el sistema pueda crecer y funcionar de manera eficaz a medida que las demandas aumenten.

**RIE-13:** Los dispositivos obsoletos pueden carecer de las actualizaciones de software o hardware necesarias para garantizar la compatibilidad con las últimas tecnologías y funcionalidades. Esto puede resultar en una experiencia de usuario subóptima o en la imposibilidad de utilizar ciertas características o servicios. Es importante tener en cuenta este riesgo al diseñar aplicaciones o servicios para asegurarse de que sean accesibles para una amplia gama de usuarios.

**ICF-23:** La optimización de rendimiento de una aplicación es un proceso crucial para garantizar que funcione de manera eficiente y satisfaga las expectativas de los usuarios.

- Problemas técnicos inesperados: Durante la optimización del rendimiento, pueden surgir problemas técnicos inesperados que afecten el funcionamiento de la aplicación, lo que podría llevar a caídas del sistema o fallos en el rendimiento.
- Incompatibilidad con dispositivos: A medida que se mejora el rendimiento, es posible que la aplicación no sea compatible con todos los dispositivos y sistemas operativos, lo que podría limitar su alcance y audiencia.
- Reacciones de los usuarios: Los usuarios actuales de la aplicación pueden experimentar cambios en el rendimiento que no les guste, lo que podría llevar a reacciones negativas o la pérdida de usuarios.
- Comunicación inadecuada: Si no se comunica claramente a los usuarios las actualizaciones y cambios en la aplicación, esto podría generar confusión y frustración, lo que afectaría la percepción del usuario sobre la aplicación.

**Riesgo Asociado: RIE-19(Comunicación deficiente)/RIE-13(Dispositivo obsoleto del cliente).**

**RIE-19:** La comunicación deficiente en el desarrollo de una aplicación de probador de lentes virtuales puede dar lugar a malentendidos internos, retrasos en las actualizaciones, respuestas inadecuadas a los usuarios, dificultades en la resolución de problemas y la falta de alineación con la visión del producto. Para mitigar estos riesgos, es crucial establecer canales de comunicación

efectivos entre los equipos internos y con los usuarios, definir procesos de gestión de proyectos que fomenten la colaboración y la transparencia, y responder de manera oportuna a los comentarios y preocupaciones de los usuarios. La comunicación clara y eficaz es esencial para el éxito sostenible de la aplicación.

**ICF-13:** La presencia de clientes que utilizan dispositivos obsoletos en la aplicación puede conllevar riesgos de incompatibilidad de hardware y software, una experiencia deficiente para el usuario y costos adicionales de desarrollo. Para mitigar estos riesgos, es esencial llevar a cabo pruebas exhaustivas en una amplia gama de dispositivos y sistemas operativos, identificar y abordar problemas de incompatibilidad, y establecer requisitos de hardware y software mínimos, comunicando claramente a los usuarios para gestionar expectativas y asegurar una experiencia satisfactoria.

**ICF-24:** Se crea un nuevo proyecto en Unity, se diseñan las escenas que simulan la experiencia de uso de los lentes virtuales y se desarrollan las funcionalidades básicas, como la interacción con los lentes y la visualización en 3D. Tras pruebas iniciales, se realizan ajustes y mejoras con base en la retroalimentación de los miembros del equipo, y se documenta el proceso para una colaboración efectiva. El desarrollo es un proceso iterativo, y se recomienda mantener una comunicación constante con el público objetivo para lograr una experiencia de prueba de lentes virtuales efectiva y satisfactoria.

- Documentación: A medida que avanzas en el desarrollo, asegúrate de documentar adecuadamente el código y los procesos para facilitar la colaboración con otros desarrolladores y para futuras actualizaciones.
- Diseño de Interfaz de Usuario (UI): Crea una interfaz de usuario simple pero funcional que permita a los usuarios cargar una foto o activar la cámara en tiempo real para probar los lentes virtuales. La interfaz debe ser intuitiva y fácil de usar.
- Pruebas Iniciales: Realiza pruebas iniciales para verificar que las funcionalidades básicas funcionen como se espera. Esto implica probar la aplicación en diferentes dispositivos y

condiciones de iluminación para asegurarse de que la superposición de los lentes sea precisa.

**Riesgo Asociado: RIE-29(Problema en navegación y usabilidad de la app)/RIE-19(Comunicación deficiente).**

**RIE 29:** El riesgo de problemas en la navegación y usabilidad de la aplicación conlleva la posibilidad de confusión de los usuarios, dificultades para encontrar funciones clave, flujos de usuario complicados y problemas de accesibilidad. Para mitigar este riesgo, es fundamental realizar pruebas de usabilidad, diseñar la interfaz pensando en el usuario, mapear un flujo de usuario claro, utilizar pruebas A/B, recopilar retroalimentación constante y garantizar la accesibilidad para todos los usuarios. La iteración continua y la atención a la retroalimentación del usuario son esenciales para crear una experiencia de usuario efectiva y satisfactoria.

**RIE-13:** El riesgo de una comunicación deficiente con el equipo de desarrollo puede resultar en una serie de problemas, como malentendidos, retrasos en el proyecto y la entrega de un producto final insatisfactorio. Para mitigar este riesgo, es fundamental establecer canales de comunicación claros y regulares, definir roles y responsabilidades dentro del equipo, utilizar herramientas de gestión de proyectos eficaces, fomentar la retroalimentación abierta y asegurarse de que todos los miembros del equipo tengan acceso a la información y los recursos necesarios para realizar su trabajo de manera efectiva. La comunicación constante y efectiva es esencial para mantener el proyecto en el camino correcto y lograr resultados exitosos.

## Notas y observaciones

Para asegurar una buena compatibilidad de hardware en una aplicación de tienda óptica que utiliza realidad aumentada, es esencial considerar las siguientes notas y consideraciones:

1. **Requisitos mínimos de hardware:** Define claramente los requisitos mínimos de hardware para que la aplicación funcione de manera óptima. Esto incluye aspectos como la potencia de procesamiento, capacidad de la GPU, memoria RAM y características de la cámara.
2. **Optimización de rendimiento:** Ajusta la aplicación para que funcione sin problemas incluso en dispositivos menos potentes. Esto puede implicar la reducción de gráficos o efectos visuales en configuraciones de hardware más modestas.
3. **Resolución y orientación de pantalla:** Diseña la interfaz de usuario para que se adapte a diferentes resoluciones de pantalla y orientaciones (horizontal y vertical). Asegúrate de que la aplicación se vea bien tanto en teléfonos como en tabletas.
4. **Interfaz de usuario intuitiva y adaptable:** Crea una interfaz de usuario intuitiva que sea fácil de navegar y que se adapte a diferentes métodos de entrada, como pantallas táctiles y ratones. Asegúrate de que los controles y botones sean lo suficientemente grandes y espaciados para ser seleccionados fácilmente en pantallas táctiles.
5. **Soporte al cliente:** Proporciona canales de comunicación claros para que los usuarios informen sobre problemas de compatibilidad. Ofrece asistencia y soluciones cuando los usuarios experimentan problemas en sus dispositivos.

Al prestar atención a estos puntos en una variedad de dispositivos, puedes garantizar una experiencia de usuario óptima y una buena compatibilidad con hardware en tu aplicación de tienda óptica con realidad aumentada.

## Asignación de recursos

Rol	Horas-Hombre	Desempeñado por	Observaciones
Analista/Investigador	6hrs	Marcos E. Solis Cenicerros	Análisis de requerimientos necesarios para el desarrollo del proyecto e investigación general que abre el panorama de desarrollo.
Programación	3 hrs	Oswaldo A. Hernandez Campos	Desarrollo continuo de la aplicación,
Investigador	1hrs	Iván O. Vásquez Flores	Investigación general que abre el panorama de desarrollo.
Analista	1hrs	Alejandro García Alférez	Investigación general que abre el panorama de desarrollo.
Equipo de desarrollo/ Investigador	4 hrs	Carlos Francisco Babún Ravelo	Crear/programar las funciones del producto e investigación general que abre

			el panorama de desarrollo.
--	--	--	----------------------------

## Anexos

### Anexo 1 (ICF-21): Informe de Selección de Plataforma.

Este informe se centra en el proceso de **selección de plataforma** para la aplicación. El principal enfoque recae en el desarrollo para dispositivos móviles con sistema operativo Android. La selección de la plataforma es un paso crucial en el proyecto y se basa en la recopilación de información preliminar sobre las necesidades y preferencias de los usuarios finales. Además, se considera la utilización de Unity como plataforma de desarrollo. Cabe destacar que la decisión sobre la plataforma está sujeta a cambios según las condiciones y necesidades del desarrollo.

#### Recopilación de Datos:

Para tomar una decisión fundamentada sobre la plataforma de desarrollo, se realizó una evaluación de las necesidades y preferencias de los usuarios finales. A continuación, se presenta la información relevante que se ha recopilado:

- **Preferencia de Dispositivos:** La mayoría de los usuarios finales utilizan dispositivos móviles con sistema operativo Android. Esto indica una alta demanda en esta plataforma, ya que se adapta a sus preferencias y necesidades.
- **Facilidad de Uso:** Se ha observado que la mayoría de los usuarios finales están familiarizados con dispositivos Android y prefieren una aplicación fácil de usar en sus dispositivos móviles. La comodidad y la experiencia del usuario son factores clave.
- **Compatibilidad:** Android ofrece una amplia gama de dispositivos y versiones del sistema operativo, lo que garantiza una mayor compatibilidad y un alcance más amplio. Esto es fundamental para llegar a la mayor audiencia posible.
- **Herramientas de Desarrollo:** La consideración de utilizar Unity como plataforma de desarrollo se basa en su capacidad probada para crear experiencias de realidad aumentada

efectivas. Unity es una herramienta versátil y ampliamente utilizada para este tipo de aplicaciones.

#### **Evaluación de Plataformas:**

Después de analizar la información recopilada, se han evaluado las siguientes plataformas:

- Android: Dada la preferencia de los usuarios finales, la alta compatibilidad y la facilidad de uso, Android se ha seleccionado como la plataforma principal para el desarrollo de la aplicación.
- Unity: La elección de Unity se mantiene como una opción sólida debido a su capacidad para desarrollar contenido de realidad aumentada de alta calidad. Unity será una herramienta fundamental en el proceso de desarrollo.

Con base en la evaluación y la información recopilada, se ha tomado la decisión de que la plataforma principal para el desarrollo de la aplicación será Android. Unity se mantendrá como una herramienta importante para la creación de contenido de realidad aumentada en la plataforma Android.

#### **Anexo 2 (ICF-22): Compatibilidad con hardware.**

La compatibilidad con hardware es un aspecto fundamental en el desarrollo de la aplicación. La capacidad de que la aplicación funcione de manera efectiva en una amplia gama de dispositivos móviles Android es crucial para alcanzar la mayor audiencia posible y garantizar una experiencia de usuario consistente. A continuación, se profundizará en las consideraciones clave para lograr la compatibilidad con hardware.

#### **Diversidad de Dispositivos Android:**

La plataforma Android abarca una diversidad de dispositivos, desde teléfonos inteligentes hasta tabletas, de diversas marcas y modelos. Cada uno de estos dispositivos tiene características de hardware específicas, como capacidades de procesamiento, memoria, cámaras y resoluciones de

pantalla. Asegurarse de que la aplicación funcione en este espectro es esencial para satisfacer las necesidades de los usuarios.

#### **Versiones del Sistema Operativo Android:**

Además de la diversidad de dispositivos, Android tiene múltiples versiones del sistema operativo en uso. Esto significa que la aplicación debe ser compatible con diferentes versiones, desde las más antiguas hasta las más recientes. Esta compatibilidad garantiza que incluso los usuarios con dispositivos más antiguos puedan acceder a la aplicación y disfrutar de sus funcionalidades.

#### **Resoluciones de Pantalla y Adaptabilidad:**

Los dispositivos Android tienen una variedad de resoluciones de pantalla. La aplicación debe adaptarse de manera dinámica a estas resoluciones para que los elementos visuales se muestren correctamente en todos los dispositivos. El diseño responsivo desempeña un papel fundamental en esta adaptabilidad.

#### **Optimización de Rendimiento:**

La aplicación debe ser eficiente en el uso de los recursos del dispositivo. Esto implica optimizar la CPU, la memoria y la capacidad de almacenamiento. Un rendimiento óptimo es esencial para garantizar que la aplicación funcione sin problemas en dispositivos con diversas capacidades de hardware.

#### **Pruebas en Diversos Dispositivos:**

Para garantizar la compatibilidad con hardware, se realizan pruebas exhaustivas en una amplia variedad de dispositivos Android. Estas pruebas incluirán dispositivos con diferentes versiones del sistema operativo, resoluciones de pantalla y capacidades de hardware. Cualquier problema de compatibilidad identificado se abordará y resolverá de manera oportuna.

En conclusión, **la compatibilidad con hardware** es una prioridad clave en el desarrollo de la aplicación. Al abordar las consideraciones mencionadas y seguir estrategias de optimización, se



aseguraré que la aplicación funcione de manera óptima en una amplia gama de dispositivos Android. Esto no solo ampliará el alcance de la aplicación, sino que también garantizará una experiencia de usuario coherente y de alta calidad.

### **Anexo 3 (ICF-23) : Informe de Optimización de Rendimiento.**

La **optimización de rendimiento** es un aspecto crítico en el desarrollo de la aplicación. La eficiencia y el rendimiento de la aplicación son fundamentales para garantizar una experiencia de usuario fluida y satisfactoria. En este informe, se detallan las medidas iniciales tomadas para mejorar el rendimiento y se describe el plan para realizar pruebas adicionales.

#### **Medidas Iniciales de Optimización:**

- **Optimización de Código:** Se ha revisado y optimizado el código de la aplicación para mejorar la eficiencia en el uso de recursos. Se han identificado y corregido fragmentos de código que podrían causar ralentizaciones o consumir recursos innecesariamente.
- **Gestión de Memoria:** Se han implementado estrategias de gestión de memoria para asegurarse de que la aplicación libere recursos no utilizados de manera oportuna. Esto es fundamental para dispositivos con capacidades de memoria limitadas.
- **Carga Diferida de Recursos:** Los recursos de la aplicación, como modelos de lentes y texturas, se cargan de manera diferida según sea necesario. Esto reduce la carga inicial y mejora el tiempo de inicio de la aplicación.
- **Optimización de Gráficos:** Se han realizado ajustes en la calidad de los gráficos y la resolución para equilibrar la calidad visual con el rendimiento. Esto garantiza una experiencia de usuario atractiva sin sacrificar el rendimiento.
- **Caché de Datos:** Se ha implementado un sistema de caché de datos para almacenar localmente datos que no cambian con frecuencia, lo que reduce la necesidad de descargarlos repetidamente.

#### **Plan para Pruebas de Rendimiento Adicionales:**

Se reconoce la importancia de pruebas exhaustivas para evaluar el rendimiento de la aplicación en una variedad de situaciones. El plan para pruebas de rendimiento adicionales incluye:

- **Pruebas de Carga:** Se simularán cargas de usuarios concurrentes para evaluar la capacidad de la aplicación para manejar múltiples usuarios al mismo tiempo. Esto es esencial para situaciones en la tienda física o en momentos de alta demanda.
- **Pruebas de Rendimiento en Diversos Dispositivos:** Se realizarán pruebas en una variedad de dispositivos Android con diferentes capacidades de hardware y versiones del sistema operativo. Esto asegurará la compatibilidad y el rendimiento en una amplia gama de dispositivos.
- **Monitorización Continua:** Se implementará un sistema de monitorización continua para recopilar datos en tiempo real sobre el rendimiento de la aplicación. Esto permitirá identificar problemas de rendimiento de manera proactiva y tomar medidas correctivas.

La optimización de rendimiento es un proceso continuo que garantiza que la aplicación funcione de manera óptima en diversas situaciones. Las medidas iniciales de optimización han mejorado la eficiencia y el rendimiento de la aplicación. Sin embargo, las pruebas adicionales son esenciales para garantizar un rendimiento constante y una experiencia de usuario sin problemas. Se reconoce que el rendimiento es un factor crítico en el éxito de la aplicación, y se están tomando medidas proactivas para abordar este aspecto. La optimización de rendimiento es una prioridad constante a lo largo del desarrollo de la aplicación.

#### **Anexo 4 (ICF-24): Prototipo Inicial en Unity.**

El desarrollo de la aplicación ha alcanzado la etapa crucial de crear un prototipo inicial en Unity. El prototipo servirá como una representación funcional de la aplicación y permitirá probar las funcionalidades básicas. Se detalla el proceso y los objetivos del prototipo inicial.

### **Objetivos del Prototipo Inicial:**

- **Demostrar Funcionalidades Básicas:** El prototipo debe ser capaz de demostrar las funcionalidades esenciales de la aplicación, incluida la prueba virtual de lentes y la visualización de detalles del producto.
- **Validar Interacción del Usuario:** Se busca validar la interacción del usuario con la aplicación, incluida la navegación por el catálogo de lentes y la prueba virtual. Se prestará especial atención a la facilidad de uso y la experiencia del usuario.
- **Realizar Pruebas Iniciales:** El prototipo permitirá realizar pruebas iniciales para identificar posibles problemas o áreas de mejora en las funcionalidades básicas. Esto incluye la detección de errores y la evaluación del rendimiento.
- **Flexibilidad para Ajustes:** Se espera que el prototipo sea lo suficientemente flexible como para permitir ajustes y mejoras según sea necesario. Esto es fundamental para adaptarse a los comentarios y las necesidades en evolución.

### **Proceso de Desarrollo:**

El desarrollo del prototipo inicial se llevará a cabo en las siguientes etapas:

- **Diseño de Interfaz de Usuario:** Se realizará el diseño de la interfaz de usuario en Unity, centrándose en la representación visual de los catálogos de lentes y la funcionalidad de prueba virtual.
- **Implementación de Funcionalidades Básicas:** Se implementarán las funcionalidades básicas, incluida la visualización de detalles del producto y la capacidad de probar virtualmente las lentes utilizando la cámara del dispositivo.
- **Pruebas Iniciales:** Se llevarán a cabo pruebas iniciales para evaluar la funcionalidad y la usabilidad. Esto incluirá la identificación de problemas y la recopilación de comentarios de los usuarios de prueba.
- **Ajustes y mejoras:** Con base en los resultados de las pruebas iniciales, se realizarán ajustes y mejoras en el prototipo. Esto podría implicar correcciones de errores, optimización de rendimiento y mejoras en la interfaz de usuario.

**Disposición General de Realizar Ajustes y Mejoras:**

Es importante destacar que el prototipo inicial se desarrollará con la disposición de realizar ajustes y mejoras a lo largo del proceso. La flexibilidad es esencial para adaptarse a las necesidades cambiantes y para garantizar que el prototipo represente de manera precisa las funcionalidades esenciales de la aplicación.

El desarrollo del prototipo inicial en Unity permitirá validar las funcionalidades básicas y recopilar comentarios valiosos de los usuarios de prueba. La disposición de realizar ajustes y mejoras garantiza que el prototipo sea una representación precisa y funcional de la aplicación final.

## **Glosario de términos**

### **Términos generales**

1. **Unity:** Una plataforma de desarrollo de juegos y aplicaciones en 2D y 3D.
2. **Android:** Sistema operativo móvil de código abierto desarrollado por Google, ampliamente utilizado en dispositivos como teléfonos y tabletas.
3. **Compatibilidad con Hardware:** Asegurarse de que la aplicación funcione en una amplia variedad de dispositivos y sistemas.
4. **Optimización de Rendimiento:** Mejorar la eficiencia y el rendimiento de la aplicación.
5. **Recopilación de Datos:** Proceso de recolección de información relevante para tomar decisiones informadas.
6. **Interfaz de Usuario Intuitiva y Adaptable:** Diseñar la parte de la aplicación con la que los usuarios interactúan para que sea fácil de usar y se adapte a diferentes dispositivos y métodos de entrada.
7. **Pruebas de Carga:** Simulación de cargas de usuarios concurrentes para evaluar la capacidad de la aplicación.
8. **Monitorización Continua:** Proceso de seguimiento en tiempo real del rendimiento de la aplicación.
9. **Prototipo Inicial:** Una versión temprana y funcional de la aplicación que se utiliza para pruebas y evaluaciones iniciales.
10. **Gestión de Memoria:** Estrategias para administrar eficazmente el uso de la memoria de un dispositivo.
11. **Diversidad de Dispositivos Android:** La variedad de dispositivos Android, incluyendo teléfonos y tabletas de diferentes marcas y modelos.
12. **Resoluciones de Pantalla:** Las dimensiones de la pantalla en términos de píxeles que afectan la calidad visual de la aplicación.
13. **API de Cámara Unificada:** Una interfaz de programación de aplicaciones que permite el acceso y control de la cámara de un dispositivo.

14. **Caché de Datos:** Almacenamiento temporal de información que no cambia con frecuencia para reducir la carga de datos en línea.
15. **Seguridad y Parches de Actualización:** Mantener la aplicación actualizada con correcciones de seguridad y nuevas versiones de software.
16. **Diseño Responsivo:** Diseñar la interfaz de usuario de manera que se adapte automáticamente a diferentes tamaños y orientaciones de pantalla.
17. **Gestión de Recursos:** Administrar eficazmente los recursos del dispositivo, como la CPU, la memoria y el almacenamiento.
18. **Escalabilidad:** La capacidad de un sistema o aplicación para crecer y manejar un mayor volumen de usuarios o datos sin reducir el rendimiento.
19. **Framework:** Una estructura o conjunto de herramientas de desarrollo que proporciona funcionalidades comunes y facilita la creación de aplicaciones.
20. **Integración Continua:** Práctica que implica la fusión regular de cambios de código en un repositorio compartido y la ejecución automática de pruebas para garantizar la estabilidad del proyecto.
21. **Interfaz de Usuario (UI):** La parte de la aplicación con la que interactúa el usuario, que incluye botones, menús y elementos visuales.
22. **SDK (Kit de Desarrollo de Software):** Un conjunto de herramientas y recursos para desarrollar aplicaciones en una plataforma específica, como Android o iOS.

## Significado de los elementos de la notación gráfica

8:11 PM

LTE 93%



**DISPONIBLE**



**DISPONIBLE**



Se trabajó en la interfaz la cual consta de un menú principal junto a su vez con componentes pensados para su uso como lo son los siguientes:

- Menú de búsqueda: este consta con la función de hacer una búsqueda más específica en la que se planifica palabras clave dependiendo de la categoría o similitud que busque el cliente.
- Imágenes sobre los modelos: una imagen clara del modelo de lente específico y a su vez con un mensaje en el que define si el producto está o no disponible tanto para la compra y o venta posible.
- icono de usuario: planificado para que el usuario tenga su cuenta la cual consta de sus datos personales como, nombre, teléfono, entre otras más que se piensan añadir en un punto del posible desarrollo a corto plazo

Este menú como tal no será el final pero sí como un boceto y posible avance en el desarrollo de la aplicación.

**Informe de Selección de Plataforma:**

Aspecto Esencial	Descripción
Preferencia del Usuario	La preferencia de los usuarios es un factor crucial en la selección de la plataforma. En este contexto, se ha identificado que la mayoría de los usuarios finales prefieren utilizar dispositivos móviles con sistema operativo Android. Esta preferencia se basa en su familiaridad con esta plataforma y la comodidad que les brinda.
Compatibilidad con Múltiples Dispositivos	La compatibilidad con una amplia gama de dispositivos es un aspecto fundamental. La plataforma seleccionada debe garantizar que la aplicación sea compatible con una variedad de dispositivos móviles, incluyendo smartphones y tablets con diferentes tamaños de pantalla, resoluciones y capacidades de hardware.
Herramientas de Desarrollo	La elección de las herramientas de desarrollo es un componente vital en la selección de plataforma. Estas herramientas determinarán cómo se desarrollará la aplicación y qué recursos estarán disponibles para el equipo de desarrollo. La elección de las herramientas debe basarse en su idoneidad para crear una experiencia efectiva de realidad aumentada y satisfacer las necesidades de los usuarios finales.

**Compatibilidad con hardware:**

Aspecto Esencial	Descripción
Requisitos de Hardware Mínimos	Se deben definir claramente los requisitos de hardware mínimos para que la aplicación funcione de manera efectiva. Esto incluye detalles como la velocidad del procesador, la cantidad de RAM, la capacidad de



	almacenamiento y las tarjetas gráficas necesarias.
Mantenimiento y Actualizaciones	La compatibilidad con hardware debe ser sostenible a lo largo del tiempo. Se debe planificar cómo manejar las actualizaciones de hardware, los cambios en las especificaciones y las nuevas versiones de dispositivos para que la aplicación siga siendo funcional.
Pruebas Rigurosas	Realizar pruebas exhaustivas en una variedad de dispositivos y configuraciones de hardware para garantizar que la aplicación funcione de manera coherente en diferentes entornos. Las pruebas beta en el mundo real pueden ser especialmente útiles para evaluar la compatibilidad con hardware.

**Informe de Optimización de Rendimiento:**

Aspecto Esencial	Descripción
Optimización de Código	La optimización de código implica una revisión exhaustiva y mejoras en el código de la aplicación para mejorar su eficiencia y rendimiento. Esto incluye la identificación y corrección de fragmentos de código que podrían causar ralentizaciones o consumir recursos innecesariamente.
Gestión de Memoria	La gestión de memoria es una estrategia que garantiza que la aplicación libere recursos no utilizados de manera oportuna. Esto es fundamental, especialmente en dispositivos con capacidades de memoria limitadas. Implementar una gestión de memoria evita el agotamiento de recursos y asegura un rendimiento óptimo en la aplicación.
Rendimiento de Red	El rendimiento de red se refiere a la velocidad y la eficiencia con la que la aplicación se

	conecta a servicios en línea, descarga datos y envía información. Optimizar el rendimiento de red implica reducir la latencia, minimizar el tiempo de carga de contenidos externos y garantizar una transmisión de datos eficiente.
--	---

**Prototipo Inicial en Unity:**

Aspecto Esencial	Descripción
Consideraciones Técnicas	Aunque el prototipo es una versión simplificada de la aplicación final, se deben tener en cuenta las consideraciones técnicas importantes, como la elección de plataformas de destino (por ejemplo, dispositivos móviles, realidad virtual), gestión de recursos y manejo de colisiones.
Interfaz de Usuario (UI) Básica	Asegurarse de tener una interfaz de usuario básica que permita a los usuarios interactuar con el prototipo. La UI debe ser lo suficientemente clara para que los usuarios comprendan cómo utilizar la aplicación, pero no es necesario que sea altamente pulida en esta etapa.
Pruebas de Usuario Tempranas	Realizar pruebas de usuario con personas que representen a tu público objetivo lo antes posible. Esto te ayudará a recopilar retroalimentación valiosa y realizar ajustes en tu prototipo según las necesidades y expectativas de los usuarios.
Flexibilidad para Iterar	Preparar el prototipo para iterar y realizar cambios en él en función de la retroalimentación que recibas de las pruebas de usuario. La flexibilidad es clave para mejorar el prototipo y llevarlo a una versión más madura.