

---

Implementación de modelos tridimensionales mediante realidad aumentada para ofrecer una experiencia inmersiva en una aplicación móvil para Android con tecnología ARCore en el Parque Arqueológico Kaminaljuyú

---

Carlos Edgardo López Barrera





UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Implementación de modelos tridimensionales mediante  
realidad aumentada para ofrecer una experiencia inmersiva en  
una aplicación móvil para Android con tecnología ARCore en  
el Parque Arqueológico Kaminaljuyú**

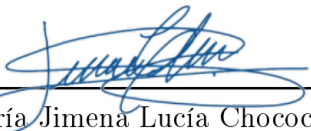
Trabajo de graduación presentado por Carlos Edgardo López Barrera  
para optar al grado académico de Licenciado en Ingeniería en Ciencias  
de la Computación y Tecnologías de la Información

Guatemala,

2025

Vo.Bo.:

(f)   
Ing. Dulce María Chacón Muñoz

(f)   
Lcda. María Jimena Lucía Chocochic Arriaga

Fecha de aprobación: Guatemala, 18 de noviembre de 2025.

Este trabajo de graduación fue elaborado como parte del proyecto AR Tour Kaminaljuyú, una aplicación móvil diseñada para integrar modelos tridimensionales en realidad aumentada (RA) y ofrecer una experiencia inmersiva a los visitantes del Parque Arqueológico Kaminaljuyú. El desarrollo técnico de la aplicación incluyó componentes de visualización, interacción, navegación y contextualización cultural, con el fin de contribuir a la preservación y difusión del patrimonio arqueológico guatemalteco mediante el uso de tecnologías móviles.

El proyecto fue realizado por un equipo multidisciplinario conformado por seis integrantes: Josué Morales estuvo a cargo del diseño de interfaz y experiencia de usuario (UX/UI), Brian Carrillo trabajó en la integración del sistema de geolocalización, Marco Ramírez desarrolló los modelos tridimensionales de las estructuras arqueológicas, Luz Coronado coordinó el análisis y la gestión de costos y Claudia Velásquez aportó la información arqueológica clave para asegurar el rigor histórico del contenido. La implementación técnica relacionada con la integración, renderizado e interacción de los modelos 3D dentro del entorno aumentado correspondió al autor de este documento.

Agradezco profundamente a mis compañeros de equipo por su compromiso, disposición y profesionalismo a lo largo de todo el desarrollo del proyecto. La colaboración constante entre áreas técnicas y conceptuales fue fundamental para consolidar una solución integral y funcional.

Asimismo, se reconoce y agradece la orientación de Dulce Chacón, quien brindó acompañamiento durante todas las etapas del proceso, así como el apoyo técnico y académico proporcionado por María Chocochic, cuya asesoría fue muy importante.

Este esfuerzo no habría sido posible sin el apoyo incondicional de mi familia. Agradezco profundamente a mi padre Axel López, a mi madre Elma Barrera y a mis hermanos Axel López y Carmen López, por su respaldo constante, motivación y confianza a lo largo de toda mi formación académica.

El presente documento representa no solo una solución tecnológica aplicada al patrimonio, sino también el resultado de un proceso colectivo marcado por la dedicación, el aprendizaje interdisciplinario y el compromiso con el desarrollo profesional y cultural.

<b>Prefacio</b>		<b>III</b>
<b>Lista de figuras</b>		<b>VII</b>
<b>Resumen</b>		<b>VIII</b>
<b>Abstract</b>		<b>IX</b>
<b>I. Introducción</b>		<b>1</b>
<b>II. Antecedentes</b>		<b>3</b>
<b>III. Justificación</b>		<b>5</b>
<b>IV. Objetivos</b>		<b>7</b>
A. Objetivo general . . . . .		7
B. Objetivos específicos . . . . .		7
<b>V. Alcance</b>		<b>8</b>
A. Definición del alcance técnico . . . . .		8
1. Funcionalidades implementadas . . . . .		8
2. Implementación y arquitectura técnica . . . . .		9
3. Exclusiones del alcance . . . . .		9
B. Limitaciones del estudio . . . . .		9
1. Limitaciones técnicas de ARCore . . . . .		9
2. Restricciones del <i>hardware</i> móvil . . . . .		10
3. Limitaciones geográficas . . . . .		10
<b>VI. Marco teórico</b>		<b>11</b>
A. Geometría proyectiva y transformaciones espaciales . . . . .		11
B. Representación de rotaciones mediante cuaterniones . . . . .		12
C. Detección de superficies y anclaje espacial . . . . .		12
D. Modelos tridimensionales y su representación . . . . .		13
E. Principios de optimización para entornos móviles . . . . .		13

F.	Interacción táctil y manipulación de objetos virtuales . . . . .	14
<b>VII.</b>	<b>Metodología</b>	<b>15</b>
A.	Herramientas, tecnologías y componentes utilizados . . . . .	15
B.	Fases de planificación y preparación . . . . .	16
1.	Análisis del entorno técnico y planificación . . . . .	16
2.	Revisión y preparación de modelos 3D . . . . .	16
C.	Desarrollo e implementación técnica . . . . .	16
1.	Carga y renderizado en ARCore . . . . .	16
2.	Optimización de recursos gráficos . . . . .	17
D.	Implementación de interacción y navegación . . . . .	17
1.	Interacción táctil y retroalimentación . . . . .	17
2.	Integración de señalización visual . . . . .	17
E.	Validación y finalización . . . . .	18
1.	Validación funcional y pruebas de rendimiento . . . . .	18
2.	Corrección de errores y ajuste final . . . . .	18
3.	Documentación técnica y empaquetado . . . . .	18
<b>VIII.</b>	<b>Resultados y Discusión</b>	<b>20</b>
A.	Visualización de modelos tridimensionales en el entorno real . . . . .	20
B.	Pruebas de manipulación tridimensional (gestos táctiles) . . . . .	22
C.	Evaluación de la experiencia mediante prueba de usabilidad . . . . .	23
D.	Información emergente y retroalimentación contextual . . . . .	25
E.	Visualización de la flecha de navegación en RA . . . . .	26
F.	Síntesis y alcances de la implementación . . . . .	29
<b>IX.</b>	<b>Conclusiones</b>	<b>30</b>
<b>X.</b>	<b>Recomendaciones</b>	<b>31</b>
A.	Recomendaciones técnicas . . . . .	31
1.	Optimización de formatos de modelos 3D . . . . .	31
2.	Validación automática de recursos . . . . .	31
B.	Recomendaciones de evaluación y experiencia de usuario . . . . .	31
1.	Pruebas en condiciones reales . . . . .	31
2.	Mejoras en la retroalimentación visual . . . . .	32
3.	Sistema de analítica y métricas . . . . .	32
C.	Recomendaciones metodológicas . . . . .	32
1.	Ampliación de pruebas de validación . . . . .	32
2.	Documentación técnica integral . . . . .	32
D.	Líneas de investigación y trabajo futuro . . . . .	32
1.	Expansión a otros sitios arqueológicos . . . . .	32
2.	Integración con otras tecnologías emergentes . . . . .	33
<b>XI.</b>	<b>Bibliografía</b>	<b>34</b>

<b>XII. Anexos</b>	<b>36</b>
A. Recursos digitales complementarios . . . . .	36
1. Repositorio de código fuente del proyecto . . . . .	36
2. Encuesta de validación técnica y de interacción . . . . .	36
B. Registro fotográfico de las pruebas de usabilidad . . . . .	36
<b>XIII. Glosario</b>	<b>38</b>

---

## Lista de figuras

---

1.	Montículo C-II-12 renderizado sobre el entorno natural . . . . .	21
2.	Vista frontal del Montículo C-II-13 integrado . . . . .	21
3.	Modelo en estado inicial . . . . .	22
4.	Modelo escalado ( <i>zoom</i> ) . . . . .	22
5.	Modelo rotado . . . . .	22
6.	Criterios técnicos de interacción . . . . .	24
7.	Complejidad de tareas asignadas . . . . .	24
8.	Tarjeta de información del Montículo C-II-13 . . . . .	25
9.	Detalle arqueológico mostrado junto al modelo 3D . . . . .	26
10.	Flecha de navegación señalando el montículo C-II-3 . . . . .	27
11.	Orientación de la flecha hacia el siguiente destino . . . . .	28
12.	Usuario durante la prueba de usabilidad . . . . .	37
13.	Usuarios realizando la encuesta de validación . . . . .	37
14.	Usuarios durante la prueba de usabilidad . . . . .	37

El presente trabajo tuvo como objetivo principal la integración e implementación de modelos tridimensionales mediante tecnología de realidad aumentada (RA) en una aplicación móvil para Android, orientada a la exploración del Parque Arqueológico Kaminaljuyá. A través de la visualización de objetos virtuales sobrepuestos al entorno físico, se buscó enriquecer la experiencia de los visitantes mediante una interfaz inmersiva y dinámica.

La aplicación fue desarrollada utilizando la plataforma ARCore, permitiendo anclar modelos 3D sobre planos reales detectados por la cámara del dispositivo. Los modelos, proporcionados por un colaborador del equipo, fueron integrados en formato .obj y optimizados para garantizar un rendimiento fluido en dispositivos móviles. Se implementaron transformaciones espaciales, renderizado en tiempo real, y un sistema de interacción táctil que permitió escalar, rotar y repositionar los objetos virtuales en función de gestos del usuario.

Durante el proceso, se aplicaron técnicas de optimización de recursos gráficos, retroalimentación háptica para reforzar la interacción, y validaciones funcionales en distintos entornos. Asimismo, se documentó detalladamente cada etapa del desarrollo técnico, desde la configuración inicial hasta la generación del archivo instalable .apk para pruebas.

Como resultado, se obtuvo una herramienta tecnológica funcional que aporta valor educativo y cultural, sirviendo como base replicable para futuras aplicaciones similares en otros sitios patrimoniales de Guatemala.

This project focused on the integration and implementation of three-dimensional models using Augmented Reality (AR) technology within a mobile application for Android, aimed at enhancing the exploration of the Kaminaljuyú Archaeological Park. By overlaying virtual objects onto the physical environment, the application sought to enrich the visitor experience through an immersive and interactive interface.

The application was developed using the ARCore platform, enabling the anchoring of 3D models onto real-world surfaces detected by the device's camera. The models, provided by a team collaborator in .obj format, were optimized to ensure smooth performance on mobile devices. Spatial transformations, real-time rendering, and touch-based interactions were implemented, allowing users to scale, rotate, and reposition virtual objects through familiar gestures.

Optimization techniques were applied to minimize graphical resource consumption, and haptic feedback was incorporated to reinforce the user's interactions. Functional tests were conducted in diverse environmental conditions, and the entire development process—from initial setup to the generation of the installable .apk for testing—was thoroughly documented.

As a result, a functional technological tool was developed, contributing educational and cultural value while establishing a replicable foundation for future applications in other heritage sites across Guatemala.

El avance de la tecnología en las últimas décadas ha transformado de manera profunda la forma en que las personas interactúan con la historia y el patrimonio cultural. Entre las innovaciones más relevantes en este ámbito, la realidad aumentada (AR) se ha consolidado como una herramienta fundamental para mejorar la accesibilidad, visualización y comprensión de sitios históricos, tales como museos y lugares arqueológicos. En este contexto, el presente trabajo aborda la implementación de ARCore para la integración de modelos tridimensionales (3D) en una aplicación de recorridos virtuales del Parque Arqueológico Kaminaljuyú, con el propósito de proporcionar una experiencia inmersiva e interactiva que contribuya a la divulgación y preservación del patrimonio cultural guatemalteco.

Kaminaljuyú constituye uno de los sitios arqueológicos más representativos de la civilización maya en Guatemala y enfrenta el reto de conservar sus estructuras sin limitar el acceso al conocimiento histórico que resguarda. La aplicación de tecnologías basadas en realidad aumentada permite ofrecer alternativas innovadoras de visualización, que complementan la visita presencial y promueven el interés del público mediante la recreación digital de elementos arqueológicos. De esta forma, la tecnología se convierte en un medio eficaz para difundir información histórica de manera segura, interactiva y accesible.

El desarrollo del proyecto implicó diversos desafíos técnicos, entre ellos la optimización de modelos 3D para dispositivos móviles, la incorporación de interacciones táctiles y gestuales para la manipulación de objetos virtuales, y la calibración espacial de los modelos dentro del entorno físico. Para resolver estos retos, se utilizó ARCore, el *framework* de realidad aumentada de Google, el cual proporciona herramientas de reconocimiento de superficies, anclaje de modelos y procesamiento en tiempo real. La metodología adoptada se basó en un enfoque iterativo de diseño y desarrollo, con fases de prototipado, pruebas de rendimiento y evaluación de usabilidad.

Los resultados obtenidos evidencian que la implementación de modelos tridimensionales en realidad aumentada mejora de manera significativa la comprensión del contexto arqueológico y la interacción del usuario con los elementos patrimoniales. Las pruebas realizadas demostraron un rendimiento estable y una alta aceptación por parte de los usuarios, quie-

nes valoraron positivamente la facilidad de uso y el aporte educativo de la herramienta. Se comprobó, además, que la realidad aumentada es un medio eficaz para fortalecer el vínculo entre el visitante y el patrimonio cultural, fomentando la apreciación del legado histórico de Kaminaljuyú.

Este documento presenta el proceso de integración de los modelos 3D con ARCore, los aspectos técnicos involucrados, y las pruebas de usabilidad aplicadas para evaluar la efectividad de la experiencia digital. Finalmente, se plantean las conclusiones derivadas del desarrollo del proyecto, orientadas a promover la aplicación de tecnologías inmersivas como estrategias de preservación y difusión del patrimonio cultural guatemalteco.

La realidad aumentada (RA) se ha consolidado como una tecnología emergente con aplicaciones significativas en el ámbito de la arqueología y la museología. Su capacidad para superponer modelos tridimensionales e información contextual sobre el entorno físico permite enriquecer la experiencia del visitante, mejorar los procesos educativos y contribuir a la conservación del patrimonio cultural sin intervención física directa en los restos arqueológicos.

A nivel internacional, diversos proyectos han evidenciado el potencial de la RA para reinterpretar sitios históricos. Uno de los primeros casos relevantes fue ARCHEOGUIDE, una guía aumentada implementada en el sitio arqueológico de Olimpia (Grecia), la cual integraba reconstrucciones virtuales de templos y escenas de la vida antigua mediante dispositivos móviles y tecnologías de rastreo [1]. De manera similar, en el sitio de Pompeya (Italia), se desarrolló una experiencia de RA que permite visualizar escenas cotidianas de la vida romana y estructuras desaparecidas a través de personajes animados y objetos digitales superpuestos al entorno real [2].

Estas experiencias han evolucionado con el avance de plataformas como ARKit y ARCore, que han facilitado el desarrollo de aplicaciones móviles para turismo cultural. Por ejemplo, Voinea et al. [3] compararon la eficacia de estas plataformas al implementar recorridos virtuales en patrimonio europeo, confirmando su utilidad para mejorar el aprendizaje, la interacción y la percepción del visitante.

En América Latina, Muñoz-Sajama et al. [4] desarrollaron una aplicación móvil con RA para visualizar la aldea de San Lorenzo en Chile, un sitio patrimonial precolombino. Este caso demostró cómo tecnologías accesibles como Unity y Vuforia pueden acercar el patrimonio a públicos más amplios y facilitar su valoración cultural, incluso en contextos con limitaciones de infraestructura.

En cuanto a las tendencias más recientes, el uso de RA se ha consolidado como una herramienta no invasiva para la documentación, conservación y difusión de sitios históricos. Bekele et al. [5] presentan una revisión sistemática de más de 200 aplicaciones inmersivas

en patrimonio cultural, concluyendo que la RA contribuye significativamente a mejorar la comprensión, accesibilidad y preservación de dichos espacios.

Asimismo, Turkoğlu y Çakıcı Alp [6] presentan el caso de Kaisareia-AR, una aplicación que permite al visitante explorar distintas capas históricas del Castillo de Kayseri (Turquía), visualizando los cambios arquitectónicos de la estructura a lo largo de diferentes épocas. Esta propuesta evidencia cómo la RA puede ser una herramienta pedagógica y de conservación a largo plazo.

Estas experiencias internacionales respaldan la pertinencia de desarrollar herramientas tecnológicas como la propuesta en este trabajo, que busca aplicar realidad aumentada para reconstruir digitalmente estructuras del Parque Arqueológico Kaminaljuyú. La integración de RA no solo responde a tendencias globales de digitalización del patrimonio, sino que representa una estrategia viable para fortalecer la educación, el turismo cultural y la conservación patrimonial en Guatemala.

---

## Justificación

---

El Parque Arqueológico Kaminaljuyú constituye uno de los vestigios más relevantes de la civilización mesoamericana en el Valle de Guatemala. Sin embargo, al igual que otros sitios históricos ubicados en entornos urbanos y con limitado acceso a herramientas digitales, la experiencia de visita presenta diversas limitaciones. Entre ellas destacan la escasa disponibilidad de información interactiva, el deterioro de estructuras originales y la falta de motivación del público para realizar recorridos presenciales. La mayor parte de los montículos se encuentra erosionada o sin excavar, y la información disponible se reduce a señalización textual, lo cual dificulta la transmisión efectiva del valor histórico y cultural del sitio, especialmente entre públicos jóvenes o internacionales acostumbrados a recursos visuales y tecnológicos.

Ante esta situación, el proyecto plantea una alternativa innovadora para modernizar y enriquecer la experiencia de exploración de Kaminaljuyú mediante el uso de realidad aumentada (RA). La propuesta consiste en el desarrollo de una aplicación móvil para dispositivos Android, basada en la plataforma ARCore, que integra modelos y objetos tridimensionales anclados con precisión en el entorno físico del parque. Esta herramienta busca ofrecer una experiencia inmersiva, educativa y visualmente atractiva que complemente el recorrido tradicional con elementos virtuales interactivos y contenidos informativos contextualizados sobre cada estructura arqueológica. Con ello, se promueve una comprensión más profunda del patrimonio y una conexión significativa entre el visitante y el entorno histórico.

El proyecto aplica conocimientos y competencias propias de la ingeniería en ciencia de la computación y tecnologías de la información, entre ellas: desarrollo de sistemas móviles, gestión de recursos gráficos, posicionamiento espacial, programación orientada a objetos, integración de SDKs especializados como ARCore, optimización de rendimiento y principios de arquitectura de *software*. Además, fomenta habilidades esenciales como la resolución de problemas técnicos, la innovación aplicada, el pensamiento crítico y el uso ético de tecnologías emergentes.

El impacto potencial del proyecto trasciende el ámbito académico y técnico, ya que la aplicación puede convertirse en una herramienta valiosa para fortalecer el valor cultural, edu-

cativo y turístico del parque. Su diseño escalable y adaptable abre la posibilidad de replicar la solución en otros museos y sitios arqueológicos del país, contribuyendo a la modernización de la divulgación patrimonial mediante tecnologías accesibles.

Desde el perfil profesional del egresado, este trabajo se alinea con dos áreas clave de formación:

- Desarrollo de sistemas informáticos y soluciones tecnológicas, mediante la implementación de una aplicación que responde a una necesidad concreta en un contexto patrimonial; y
- Innovación y emprendimiento tecnológico, al proponer una herramienta con potencial de aplicación institucional o comercial que fomente la valorización del patrimonio cultural guatemalteco.

En conjunto, el proyecto refuerza la formación del egresado como un profesional capaz de diseñar e implementar soluciones tecnológicas con impacto social, evidenciando que el conocimiento adquirido puede materializarse en iniciativas útiles, escalables y relevantes para la sociedad.

### A. Objetivo general

Implementar modelos tridimensionales mediante realidad aumentada para ofrecer una experiencia inmersiva en una aplicación móvil para Android con tecnología ARCore en el Parque Arqueológico Kaminaljuyá.

### B. Objetivos específicos

- Integrar objetos y modelos tridimensionales en una aplicación móvil para Android, con el fin de permitir su visualización inmersiva en el entorno físico, mediante el uso de la tecnología de realidad aumentada ARCore.
- Desarrollar e implementar interacciones funcionales entre los modelos 3D y el usuario, para mejorar la experiencia de exploración del Parque Kaminaljuyá, mediante técnicas de posicionamiento espacial, gestos táctiles y retroalimentación visual y/o táctil.
- Establecer anclajes espaciales precisos para la correcta proyección de los modelos 3D en el entorno físico del Parque Kaminaljuyá, con el fin de lograr una alineación coherente entre la realidad aumentada y el espacio real, mediante el uso de funcionalidades avanzadas de ARCore.
- Evaluar el rendimiento, la estabilidad y la correcta ejecución de la aplicación móvil en diferentes dispositivos Android, con el fin de asegurar su funcionamiento óptimo en condiciones reales de uso, mediante pruebas técnicas de carga, compatibilidad y comportamiento en campo utilizando la plataforma ARCore.

## A. Definición del alcance técnico

El presente proyecto se enfocó en la implementación de una solución técnica para la visualización de modelos tridimensionales en un entorno de realidad aumentada, utilizando la plataforma ARCore sobre dispositivos Android. Su objetivo específico fue integrar, manipular y renderizar modelos 3D representativos del Parque Arqueológico Kaminaljuyú, con el fin de ofrecer una experiencia inmersiva que contribuya a la exploración y divulgación del patrimonio cultural guatemalteco.

El alcance del trabajo se limitó a los aspectos directamente relacionados con el procesamiento y despliegue visual de dichos modelos, sin involucrar tareas de modelado original, desarrollo de la interfaz gráfica de usuario, ni gestión de geolocalización o persistencia en la nube. Los modelos fueron proporcionados por un colaborador del equipo y se entregaron en formato .obj. El trabajo se centró en la integración de estos recursos dentro del entorno RA, garantizando su correcta carga, posicionamiento espacial, orientación y visualización en tiempo real.

### 1. Funcionalidades implementadas

Las funcionalidades desarrolladas como parte de este alcance técnico incluyeron:

- Carga asincrónica de modelos 3D en formato .obj, con asignación automática de materiales y texturas compatibles.
- Aplicación de transformaciones espaciales (traslación, rotación, escala) utilizando estructuras homogéneas  $4 \times 4$ , vinculadas a eventos de interacción táctil.
- Optimización de recursos gráficos para asegurar un desempeño estable en dispositivos móviles, mediante reducción de complejidad geométrica, agrupación de materiales y

control del tamaño de texturas.

- Renderizado en tiempo real de los modelos integrados en el contexto del mundo físico detectado por la cámara del dispositivo, sobre planos identificados por ARCore.
- Manipulación directa de los modelos mediante gestos multitáctiles, incluyendo escalado, rotación y desplazamiento, así como retroalimentación háptica en eventos clave para reforzar la interacción del usuario.

## 2. Implementación y arquitectura técnica

Todas las funcionalidades desarrolladas fueron encapsuladas en componentes modulares como `ModelController.kt`, `ModelPart.kt` y `HapticFeedback.kt`, integrados dentro del ciclo de vida de la sesión RA mediante `ARCoreSessionLifecycleHelper.kt`. La aplicación resultante fue probada en entornos reales con condiciones lumínicas y espaciales variables, y se comprobó que los modelos se comportaban de manera estable, manteniendo su alineación espacial con el entorno y respondiendo correctamente a la interacción del usuario.

## 3. Exclusiones del alcance

Este alcance técnico no incluye el diseño del flujo completo de usuario, la navegación por rutas dentro del parque, la incorporación de contenido educativo contextual, ni la implementación de mecanismos de almacenamiento persistente o colaboración multiusuario. Estas funcionalidades fueron abordadas por otros miembros del equipo o reservadas para etapas posteriores del desarrollo del proyecto.

## B. Limitaciones del estudio

El presente trabajo presenta ciertas limitaciones que deben considerarse al interpretar los resultados obtenidos y la aplicabilidad de la solución desarrollada:

### 1. Limitaciones técnicas de ARCore

La implementación depende directamente de las capacidades y restricciones del *framework* ARCore de Google. Estas incluyen limitaciones en la detección de superficies bajo condiciones de iluminación deficiente, dificultades para el *tracking* en superficies muy reflectantes o con patrones repetitivos, y dependencia de características visuales distintivas en el entorno para mantener el anclaje espacial de los modelos. Adicionalmente, ARCore requiere dispositivos compatibles con un conjunto específico de sensores (acelerómetro, giroscopio, cámara), lo que limita la disponibilidad de la aplicación a una fracción del parque instalado de dispositivos Android.

## **2. Restricciones del *hardware* móvil**

El rendimiento de la aplicación está condicionado por las capacidades de procesamiento gráfico y memoria de los dispositivos móviles objetivo. Los modelos tridimensionales complejos pueden generar carga computacional significativa, especialmente en dispositivos de gama media o baja, resultando en reducción del *framerate*, latencia en la respuesta táctil, o calentamiento excesivo del dispositivo. La capacidad limitada de almacenamiento también restringe el número y tamaño de los modelos que pueden integrarse en una sola aplicación.

## **3. Limitaciones geográficas**

El desarrollo se enfocó específicamente en el contexto del Parque Arqueológico Kaminaljuyú, con modelos y contenido cultural adaptados a este sitio particular. La aplicación no fue diseñada como una solución genérica para otros sitios arqueológicos, lo que limita su transferibilidad directa sin adaptaciones significativas en el contenido, los modelos tridimensionales, y potencialmente en la lógica de geolocalización y navegación.

La RA (realidad aumentada) se refiere a la superposición de contenido digital como modelos tridimensionales, imágenes o datos sobre el entorno físico en tiempo real. A través de esta tecnología se logra una experiencia híbrida donde lo real y lo virtual se integran espacialmente, sin eliminar la percepción del entorno físico [7]. Su aplicación ha sido particularmente útil en contextos culturales y educativos por su capacidad para representar elementos históricos de forma visual e interactiva. Los aspectos perceptuales y de interacción en RA presentan desafíos específicos relacionados con la alineación visual, la latencia y la coherencia espacial entre elementos reales y virtuales [8]. Para su implementación técnica en entornos móviles, la RA (realidad aumentada) requiere una base teórica sólida en áreas como geometría proyectiva, gráficos por computadora, interacción humano-computadora y representación espacial.

## A. Geometría proyectiva y transformaciones espaciales

La integración coherente de objetos virtuales en una escena aumentada depende del uso correcto de sistemas de coordenadas y transformaciones espaciales. En entornos tridimensionales, los objetos se representan en un sistema de coordenadas local que debe transformarse al sistema de coordenadas del mundo virtual, luego a coordenadas de cámara, y finalmente a coordenadas de pantalla. Para ello se utilizan matrices de transformación homogéneas de  $4 \times 4$ , las cuales permiten combinar operaciones de traslación, rotación, escala y proyección de forma eficiente [9].

Cada vértice de un modelo se transforma mediante la siguiente ecuación matricial:

$$p_{clip} = P \cdot V \cdot M \cdot p_{local} \quad (1)$$

donde:

- $M$ : matriz homogénea  $4 \times 4$  de modelo, transforma del espacio del objeto al mundo virtual.
- $V$ : matriz homogénea  $4 \times 4$  de vista, transforma del mundo al sistema de cámara.
- $P$ : matriz homogénea  $4 \times 4$  de proyección, convierte a coordenadas de pantalla con perspectiva.

Estas matrices trabajan en conjunto para mantener la coherencia geométrica entre el objeto virtual y el entorno físico percibido por la cámara del dispositivo. La utilización de transformación espacial homogéneas también permite encadenar múltiples operaciones en una sola matriz compuesta, lo cual es fundamental en motores gráficos modernos y sistemas como ARCore.

## B. Representación de rotaciones mediante cuaterniones

La representación de rotaciones tridimensionales es un componente esencial en RA. Aunque es posible utilizar ángulos de Euler o matrices de rotación  $3 \times 3$ , estas opciones presentan problemas como la ambigüedad de interpolación y el bloqueo de *gimbal*. En cambio, los cuaterniones ofrecen una forma más estable y eficiente para representar y componer rotaciones en 3D [10].

Un cuaternión unitario puede expresarse como:

$$q = w + xi + yj + zk \quad (2)$$

donde  $w$  representa la componente escalar y  $x, y, z$  forman el vector imaginario. Esta representación permite interpolaciones suaves entre orientaciones (*slerp*) y rotaciones compuestas sin distorsión.

En entornos como ARCore, las orientaciones tanto de la cámara como de los objetos anclados se representan internamente mediante cuaterniones. Esta estructura permite transformar los objetos virtuales con precisión cuando el usuario rota el dispositivo o manipula el objeto directamente, como ocurre en los gestos de rotación integrados en la aplicación.

## C. Detección de superficies y anclaje espacial

Para ubicar modelos tridimensionales de forma coherente sobre el entorno físico, la RA (realidad aumentada) móvil debe comprender su contexto espacial. ARCore logra esto a través de una combinación de visión por computadora y sensor inercial inerciales, lo cual le permite identificar características visuales del entorno y generar mapas locales del espacio [11]. La implementación de estas funcionalidades requiere el uso de lenguajes como Kotlin, que facilita el desarrollo nativo en Android [12], así como el manejo adecuado de sensores y eventos táctiles a través de las API (*application programming interface*) oficiales de Android [13].

Una vez detectadas suficientes características, ARCore puede identificar planos horizontales o verticales. Estos planos sirven como base para anclar modelos tridimensionales mediante estructuras conocidas como *anchor* (ARCore). Un *anchor* (ARCore) almacena una transformación espacial fija respecto al mundo real detectado, permitiendo que el contenido digital permanezca en una ubicación consistente aunque el usuario se desplace o rote su dispositivo.

Esta técnica garantiza que, una vez colocado un modelo por ejemplo, una estructura maya reconstruida, este permanezca “pegado” al entorno físico, ofreciendo una experiencia de exploración estable y creíble.

## D. Modelos tridimensionales y su representación

Los modelos 3D que se integran en sistemas RA (realidad aumentada) deben cumplir con ciertos requisitos estructurales: contener geometría limpia, textura (*texture mapping*) optimizadas y materiales compatibles con el motor de renderizado. Los modelos en formato .obj (*wavefront object file format*), son un estándar ampliamente utilizado por su simplicidad, portabilidad y compatibilidad con múltiples plataformas de diseño y renderizado [14].

El archivo .obj (*wavefront object file format*) permite describir la geometría de un objeto en términos de vértices, normales y coordenadas UV, separados de sus materiales (.mtl). Su estructura basada en texto facilita la depuración, modificación y adaptación a distintos entornos de ejecución. En aplicaciones móviles, este formato es especialmente útil cuando se requiere flexibilidad en la integración, aunque puede requerir optimización gráfica adicionales para alcanzar un rendimiento óptimo.

## E. Principios de optimización para entornos móviles

La visualización de contenido 3D en tiempo real desde dispositivos móviles implica retos técnicos debido a las limitaciones de procesamiento, memoria y energía. Por ello, es necesario aplicar estrategias de optimización para asegurar la fluidez del sistema sin comprometer la calidad visual del modelo.

Entre las prácticas más efectivas se encuentran:

- **Reducción de polígonos:** disminuye la complejidad geométrica mediante simplificación de mallas, manteniendo las formas esenciales del objeto.
- **Compresión y reducción de texturas:** las textura (*texture mapping*) se ajustan a resoluciones adecuadas (potencias de dos) y se comprimen en formatos compatibles como ETC2.
- **Agrupación de materiales:** reducir la cantidad de materiales aplicados disminuye el número de llamadas de dibujo (*draw calls*), mejorando el desempeño gráfico.

- **Eliminación de geometría no visible:** eliminar partes internas u ocultas del modelo evita renderizados innecesarios.

Estas técnicas permiten mantener la tasa de cuadros por segundo (FPS (*frames per second*)) estable, reducen el uso de recursos de GPU y aumentan la duración de batería, elementos fundamentales para una experiencia RA (realidad aumentada) fluida y funcional [15]. La optimización gráfica de modelos 3D incluye también el uso de herramientas especializadas como Blender para preparar geometrías limpias y textura (*texture mapping*) eficientes [16].

## F. Interacción táctil y manipulación de objetos virtuales

La interacción en entornos RA (realidad aumentada) se apoya fuertemente en gestos táctiles que permiten al usuario manipular los modelos 3D de forma directa e intuitiva. Este tipo de interacción responde a patrones ya conocidos en interfaces móviles, como:

- **Toque para colocar:** detecta el plano donde se colocará el modelo.
- *Pinch-to-zoom:* permite escalar el modelo en tiempo real.
- **Giro con dos dedos:** aplica rotación sobre el eje vertical.
- **Arrastre con un dedo:** desplaza el objeto dentro del plano anclado.

Estas interacciones se traducen internamente en transformación espacial aplicadas a la matriz homogénea  $4 \times 4$  de modelo del objeto, y deben calcularse en tiempo real con alta precisión para que el comportamiento virtual sea coherente con el movimiento del usuario.

El desarrollo de la solución tecnológica para la visualización de modelos tridimensionales en RA (realidad aumentada) se estructuró en varias fases, siguiendo un enfoque iterativo y modular. A continuación, se describen detalladamente las etapas ejecutadas durante el proceso, considerando exclusivamente los componentes asociados a la integración, manipulación, renderizado y validación de modelos 3D mediante la plataforma ARCore para Android.

## A. Herramientas, tecnologías y componentes utilizados

La implementación técnica del proyecto se realizó utilizando Android Studio Hedgehog (2023.1.1 Patch 2), junto con el sistema de construcción Gradle 8.1.1 y el SDK de Android desde el nivel 26 (Android 8.0), asegurando compatibilidad con dispositivos certificados por Google ARCore. El lenguaje de programación principal fue Kotlin (1.9.x), aprovechando su interoperabilidad con Java y su modelo de concurrencia mediante corrutina para tareas asíncronas como la carga de modelos.

El sistema de renderizado fue construido directamente sobre la API (*application programming interface*) de bajo nivel de ARCore, utilizando componentes personalizados como `ModelController.kt`, `ModelPart.kt` y `HelloGeoRenderer.kt`. Estos módulos gestionaron la carga de modelos en formato `.obj` (*wavefront object file format*), el posicionamiento relativo en el espacio tridimensional y la transformación espacial de sus matriz homogénea  $4 \times 4$  homogéneas  $4 \times 4$  para rotación, traslación y escalado. El anclaje espacial se manejó mediante la creación de instancias de *anchor* (ARCore) vinculadas a detecciones de planos proporcionadas por la sesión activa de ARCore (`ARCoreSessionLifecycleHelper.kt`).

La interacción del usuario se implementó mediante los controladores nativos de Android: `GestureDetectorCompat` para gestos de deslizamiento y rotación, y `ScaleGestureDetector`

para escalamiento. Las acciones se tradujeron directamente en transformación espacial espaciales sobre la malla del modelo. Se añadió retroalimentación háptica mediante la clase (*class*) *vibrator* (Android) encapsulada en `HapticFeedback.kt`, activada tras eventos clave como colocación o reposicionamiento del modelo.

Los modelos 3D fueron verificados en Blender 3.6 LTS, confirmando orientación, escala y limpieza de geometría. Durante la ejecución, se utilizaron herramientas de perfilado integradas en Android Studio para evaluar el uso de CPU, memoria y la estabilidad de la tasa de cuadros por segundo (FPS (*frames per second*)). El paquete final fue compilado como .apk (Android *package kit*) en modo *release*, utilizando Proguard y R8 para minimizar el tamaño del binario y ofuscar el código fuente.

## B. Fases de planificación y preparación

### 1. Análisis del entorno técnico y planificación

El proceso se inició con la identificación de los requisitos técnicos necesarios para integrar modelos 3D en una aplicación de RA funcional sobre dispositivos móviles. Se definieron criterios de compatibilidad (mínimo Android 8.0 con soporte para ARCore), recursos computacionales deseables (procesador de 64 bits y GPU con soporte para OpenGL ES 3.0), así como consideraciones de rendimiento objetivo (fluidez mínima de 30 FPS). Simultáneamente, se configuró el entorno de desarrollo en Android Studio, integrando el ARCore SDK y librerías auxiliares para renderizado y control de gestos.

### 2. Revisión y preparación de modelos 3D

Los modelos tridimensionales fueron proporcionados por un colaborador del equipo en formato .obj. Se procedió a validar su integridad geométrica, asegurando que las mallas estuvieran cerradas, optimizadas y con asignación correcta de coordenadas de textura. A través de herramientas como Blender, se ajustaron las escalas relativas, se orientaron correctamente los ejes de referencia y se eliminaron vértices redundantes. Aunque no se realizó modelado desde cero, fue necesario preparar los archivos para su incorporación eficiente en el entorno móvil, cuidando el peso de los archivos y la complejidad de las estructuras.

## C. Desarrollo e implementación técnica

### 1. Carga y renderizado en ARCore

Una vez preparados, los modelos se integraron en el entorno de desarrollo mediante una arquitectura que permitiera su carga asincrónica. Se empleó una estructura modular basada en las clases `ModelController.kt` y `ModelPart.kt`, encargadas de la gestión de transformaciones espaciales, materiales y control del ciclo de vida de cada modelo. Se implementó renderizado en tiempo real sobre planos horizontales detectados por ARCore, con soporte

para mapeo de texturas, iluminación básica y configuración dinámica de escala. La renderización fue adaptada para mantener una tasa de cuadros por segundo estable, incluso en condiciones de uso prolongado.

## 2. Optimización de recursos gráficos

Dado el entorno móvil de ejecución, se aplicaron diversas estrategias de optimización. Se redujo el conteo de polígonos en los modelos, se comprimieron texturas para minimizar el uso de memoria, y se consolidaron materiales para disminuir las llamadas al renderizador. Además, se eliminó geometría interna no visible, se verificó la validez del sistema de coordenadas y se limitó el número de modelos activos simultáneamente para evitar sobrecargas. Estas acciones contribuyeron a garantizar una experiencia fluida en dispositivos de gama media y alta.

## D. Implementación de interacción y navegación

### 1. Interacción táctil y retroalimentación

Se diseñaron e implementaron gestos multitáctiles para permitir la manipulación directa de los modelos dentro del espacio RA. Las interacciones incluyeron:

- Escala proporcional mediante gesto de pinzamiento.
- Rotación sobre eje Y a través de desplazamiento lateral con dos dedos.
- Reubicación del modelo sobre el plano detectado mediante arrastre simple.

Para reforzar la experiencia sensorial, se integró retroalimentación háptica mediante la clase `HapticFeedback.kt`, activando una vibración leve en eventos clave como la colocación exitosa del modelo o el bloqueo de un gesto. Las interacciones fueron limitadas por umbrales lógicos de escala y rotación para evitar comportamientos visuales erráticos o inconsistentes.

### 2. Integración de señalización visual

Durante el desarrollo de la interfaz aumentada, se integró un componente visual de señalización orientado a guiar al usuario en tiempo real hacia el siguiente punto de interés. Para ello, se implementó una flecha tridimensional persistente en el *head-up display* (HUD) de la aplicación, la cual calcula dinámicamente la dirección en la que se encuentra el próximo modelo en la secuencia establecida.

Este indicador se actualiza en función de la posición relativa entre el usuario y el objeto virtual siguiente, y se orienta automáticamente en tiempo real conforme el usuario se desplaza dentro del entorno físico. A diferencia de soluciones con múltiples flechas por modelo, se optó por una única flecha de navegación continua para evitar saturación visual y mantener

la claridad de la escena. Esta flecha permanece visible durante todo el recorrido y actúa como un sistema de referencia para facilitar la exploración del entorno aumentado.

## **E. Validación y finalización**

### **1. Validación funcional y pruebas de rendimiento**

Se llevaron a cabo pruebas funcionales en distintos entornos simulados (interiores, exteriores, iluminación variable) con dispositivos de especificaciones medias y altas. Se evaluaron los siguientes aspectos:

- Estabilidad de anclajes sobre planos horizontales detectados por ARCore.
- Tiempo de carga promedio por modelo (inferior a 3 segundos).
- Respuesta a gestos táctiles y fluidez de la animación.
- Consumo de memoria en ejecución y tasa de cuadros por segundo (FPS).

Se documentaron comportamientos anómalos como fluctuaciones en la iluminación ambiental y pérdida ocasional de *tracking* en condiciones de baja textura visual, lo cual fue considerado dentro de los límites esperados del SDK.

### **2. Corrección de errores y ajuste final**

A partir de las pruebas realizadas, se corrigieron errores relacionados con escalas incorrectas, orientación invertida en algunos modelos y conflictos de textura al aplicar ciertos materiales. Se ajustaron los límites mínimos y máximos de *zoom*, y se mejoró la interpolación de rotación para asegurar una experiencia más fluida. También se ajustó la lógica de inicialización para prevenir cargas simultáneas de múltiples modelos, evitando caídas de rendimiento.

### **3. Documentación técnica y empaquetado**

Se documentó exhaustivamente el proceso técnico, incluyendo:

- Configuración del entorno de desarrollo.
- Estructura de clases y componentes desarrollados.
- Esquemas de interacción y flujo de eventos.
- Bitácoras de prueba y análisis de rendimiento.

Se generó un archivo .apk de la aplicación con fines de prueba y distribución interna. Este archivo fue utilizado para instalar la aplicación en distintos dispositivos y facilitar su evaluación funcional. No se realizó el despliegue oficial en Google Play Store, ya que esta etapa se reservó para una fase posterior del proyecto.

#### **A. Visualización de modelos tridimensionales en el entorno real**

Durante las pruebas realizadas en el Parque Arqueológico Kaminaljuyú, se verificó el comportamiento de la aplicación en condiciones reales, evaluando el posicionamiento, estabilidad y renderizado de los modelos tridimensionales. Cada montículo modelado fue cargado dinámicamente desde archivos en formato .obj, anclado sobre planos horizontales detectados en tiempo real por ARCore, y renderizado mediante estructuras programadas en Kotlin.

Los modelos virtuales se mantuvieron visualmente estables durante el desplazamiento del usuario, conservando la escala, proporción y orientación establecidas durante su inicialización. Las pruebas incluyeron distintos escenarios de iluminación y tipos de superficie natural (césped, tierra, áreas sombreadas), observándose una tasa de cuadros promedio sostenida de 45 FPS, sin caídas de rendimiento perceptibles en dispositivos de gama media.



Figura 1: Montículo C-II-12 renderizado sobre el entorno natural

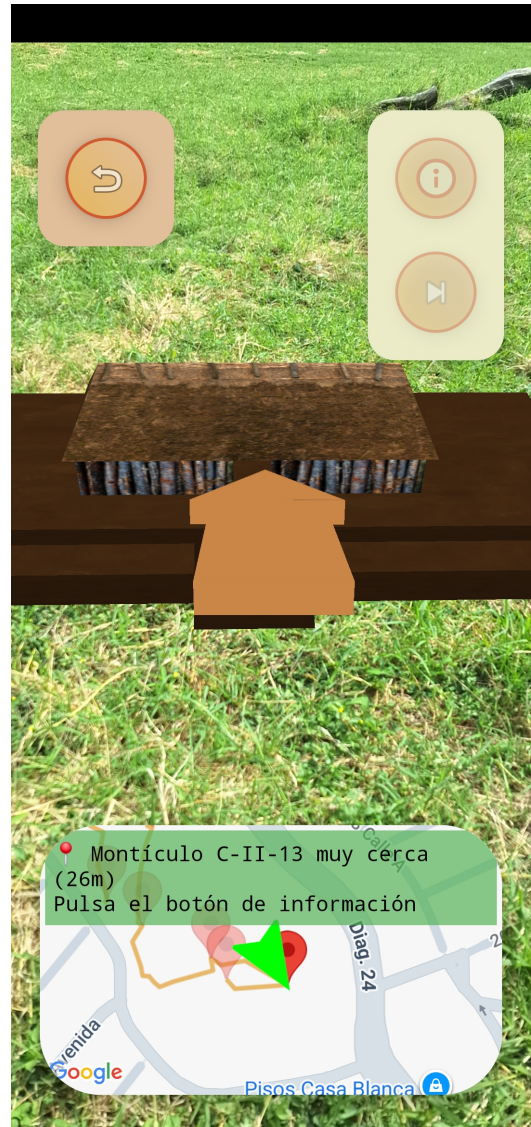


Figura 2: Vista frontal del Montículo C-II-13 integrado

Los resultados observados confirman que los anclajes generados mediante la API de ARCore se comportaron de forma coherente bajo condiciones ambientales variables. A pesar de que algunas áreas presentaban superficies irregulares o iluminación desigual, los anclajes se mantuvieron estáticos una vez colocados, sin experimentar deslizamientos ni rotaciones involuntarias. Esto sugiere que el algoritmo de detección de planos fue suficientemente robusto para mantener referencias estables.

El rendimiento del motor de visualización 3D sobre ARCore, implementado directamente en Kotlin sin depender de motores gráficos externos, demostró eficiencia notable. El manejo asíncrono de entrada/salida con corrutinas y el desacoplamiento del hilo de renderizado del hilo principal permitieron mantener una experiencia fluida. La tasa de cuadros sostenida de 45 FPS en dispositivos de gama media valida la arquitectura modular implementada mediante `ModelController.kt`.

## B. Pruebas de manipulación tridimensional (gestos táctiles)

Uno de los objetivos principales fue implementar un sistema de interacción táctil que permitiera al usuario manipular los modelos tridimensionales mediante gestos nativos de Android. Para validar esta funcionalidad, se realizaron pruebas específicas que incluyeron:

- Escalamiento mediante gesto de pinza (*zoom in / zoom out*)
- Rotación mediante gesto de giro
- Traslación mediante gesto de arrastre

Se capturaron secuencias visuales en las que se mostró un mismo modelo en estado inicial, luego escalado y posteriormente rotado, confirmando la correcta aplicación de las matrices de transformación homogénea sobre el objeto anclado.



Figura 3: Modelo en estado inicial



Figura 4: Modelo escalado (*zoom*)

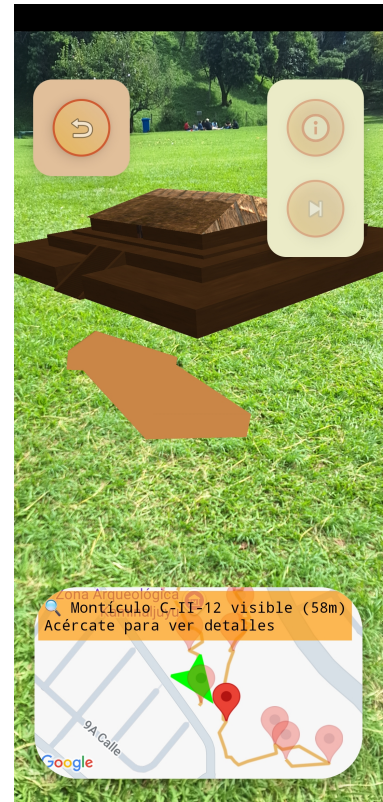


Figura 5: Modelo rotado

Estas interacciones no solo respondieron de manera precisa y sin latencia, sino que también conservaron el anclaje espacial del modelo en el entorno físico detectado. La respuesta del sistema fue inmediata ante los cambios de escala y orientación, manteniendo la fidelidad visual del objeto en todo momento.

La manipulación de los modelos tridimensionales mediante gestos táctiles multitáctiles demostró alta fidelidad en la captura de interacciones. El sistema respondió con latencia prácticamente imperceptible, indicando que las transformaciones espaciales fueron aplicadas correctamente mediante matrices homogéneas y respetaron los límites definidos para cada nodo anclado.

La retroalimentación háptica reforzó significativamente la percepción de respuesta por parte del usuario, especialmente en eventos como los cambios de escala, aportando una dimensión sensorial adicional a la experiencia. Incluso en escenarios de manipulación intensiva (rotaciones múltiples, escalas extremas), no se observó pérdida de integridad geométrica ni bloqueos del sistema, validando la robustez de la implementación.

### C. Evaluación de la experiencia mediante prueba de usabilidad

Con el fin de validar el comportamiento de la aplicación en condiciones controladas, se organizó una prueba de usabilidad simulada dentro del campus universitario. Se colocaron modelos virtuales en puntos georreferenciados distribuidos a lo largo de un recorrido, y se invitó a los participantes a interactuar con ellos como si estuvieran en el parque arqueológico.

Se encuestaron a 17 usuarios con un formulario digital, centrado en criterios de interacción directa y experiencia general. Los resultados mostraron:

- 82.4% afirmó que pudo hacer *zoom* y rotación de forma precisa y predecible.
- 82.4% consideró que los modelos se mostraron en el lugar y momento esperado.
- 94.1% manifestó que la respuesta gestual fue fluida, sin presencia de *lag* perceptible.
- 88.2% indicó haber comprendido las acciones disponibles y haberlas ejecutado con éxito.

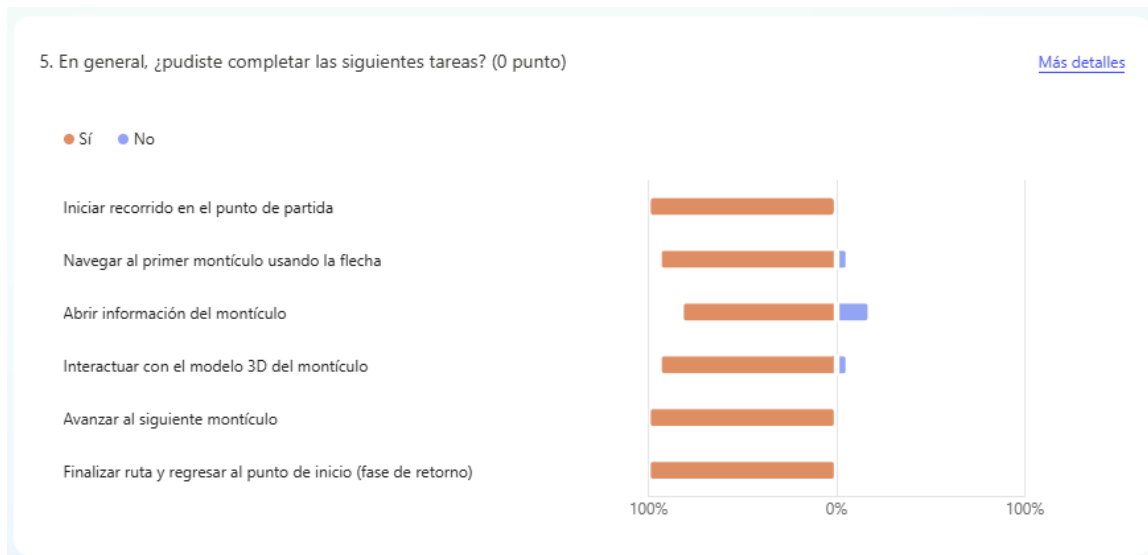


Figura 6: Criterios técnicos de interacción



Figura 7: Completitud de tareas asignadas

Durante las pruebas de usabilidad, los participantes interactuaron directamente con la aplicación en tiempo real, evaluando tanto la respuesta del sistema como la intuitividad de los gestos implementados.

Los datos recolectados en la prueba de usabilidad controlada ( $n = 17$ ) respaldan la efectividad del diseño de interacción implementado. El hecho de que más del 88 % de los participantes reportara haber comprendido intuitivamente las acciones disponibles sugiere que los patrones de interacción fueron suficientemente naturales como para prescindir de una curva de entrenamiento explícita.

Los porcentajes superiores al 82 % en precisión de gestos y estabilidad de visualización

indican que la arquitectura técnica cumplió con los requisitos de usabilidad esperados para aplicaciones de RA. La fluidez reportada (94.1% sin *lag* perceptible) valida la optimización de recursos y el manejo eficiente del ciclo de renderizado, representando un avance importante en términos de diseño centrado en el usuario dentro de entornos de realidad aumentada aplicados al patrimonio cultural.

## D. Información emergente y retroalimentación contextual

En las pruebas realizadas tanto en sitio como en entorno simulado, se validó el correcto despliegue de información arqueológica contextual al interactuar con cada modelo. El sistema desplegó una tarjeta emergente (InfoDialog) con hallazgos, periodos culturales y datos técnicos sobre el montículo correspondiente, anclada al botón de información visible en la interfaz RA.

Si bien este componente no fue parte directa del desarrollo asignado, sí formó parte de las pruebas integrales, evidenciando que la interacción entre el modelo y su información contextual fue fluida y comprensible para los usuarios.

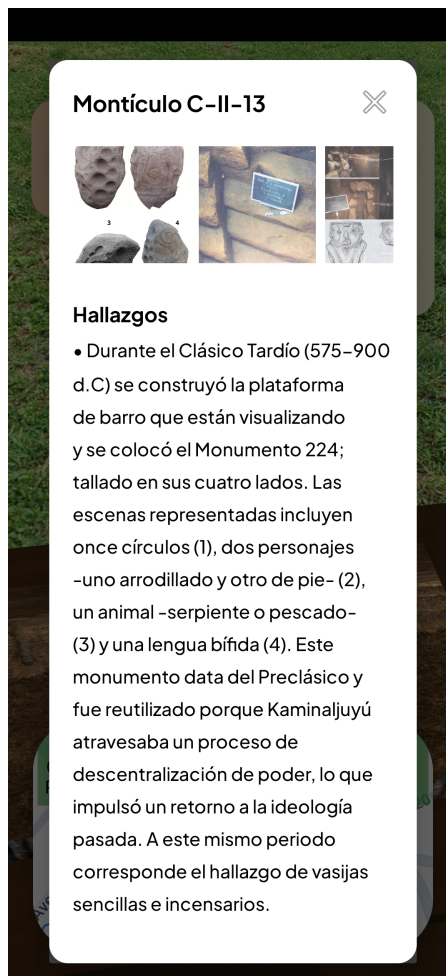


Figura 8: Tarjeta de información del Montículo C-II-13

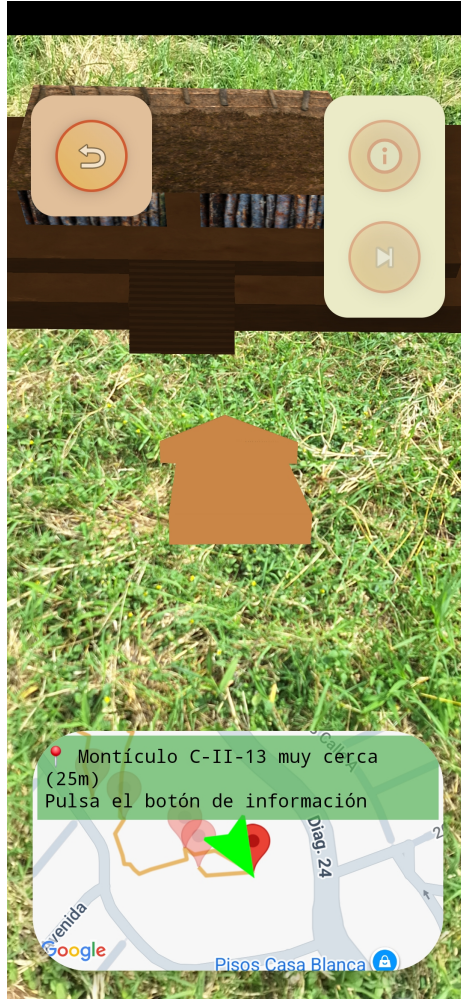


Figura 9: Detalle arqueológico mostrado junto al modelo 3D

La validación de la integración contextual mediante el despliegue de información arqueológica vinculada a cada modelo representa un aporte significativo en términos de experiencia inmersiva. La visualización simultánea de elementos gráficos (modelo + tarjeta informativa) dentro del mismo campo visual demuestra que el sistema permitió la sincronización entre el modelo desplegado y su contenido histórico, lo cual es esencial en proyectos que buscan fines educativos y de divulgación cultural.

Esta integración confirma que la arquitectura desarrollada es extensible y capaz de soportar componentes adicionales sin comprometer el rendimiento o la estabilidad del sistema de visualización 3D principal.

## E. Visualización de la flecha de navegación en RA

Como parte del flujo de navegación asistida, se implementó un componente tridimensional adicional en forma de flecha direccional. Esta flecha tuvo como propósito orientar al

usuario hacia el siguiente punto de interés dentro del recorrido, indicando visualmente la dirección aproximada que debía seguir. El objeto fue renderizado en RA sobre el entorno físico, de forma que mantuviera su orientación relativa en todo momento con respecto al montículo objetivo.

El modelo de flecha se cargó mediante el mismo sistema de renderizado utilizado para los montículos, y fue anclado dinámicamente en función del avance del usuario. Se aplicaron transformaciones de rotación basadas en vectores de dirección entre la posición actual del dispositivo y la ubicación del próximo punto de interés, asegurando que la orientación fuese coherente con el mapa en pantalla.

Durante las pruebas realizadas en sitio, se confirmó que el modelo respondía correctamente al avance del usuario, reorientándose cuando se alcanzaban nuevos puntos de navegación. La flecha se mantuvo visible y estable sobre el plano del suelo, facilitando la interpretación espacial sin interrumpir la visualización de los modelos patrimoniales.

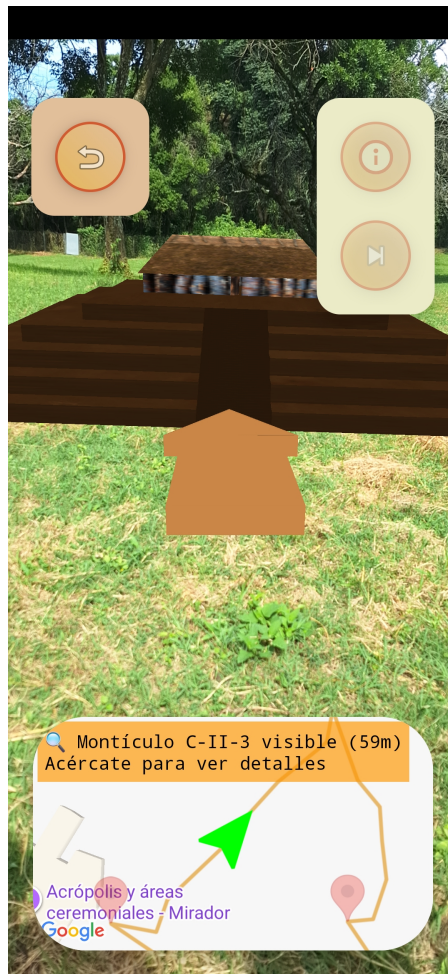


Figura 10: Flecha de navegación señalando el montículo C-II-3

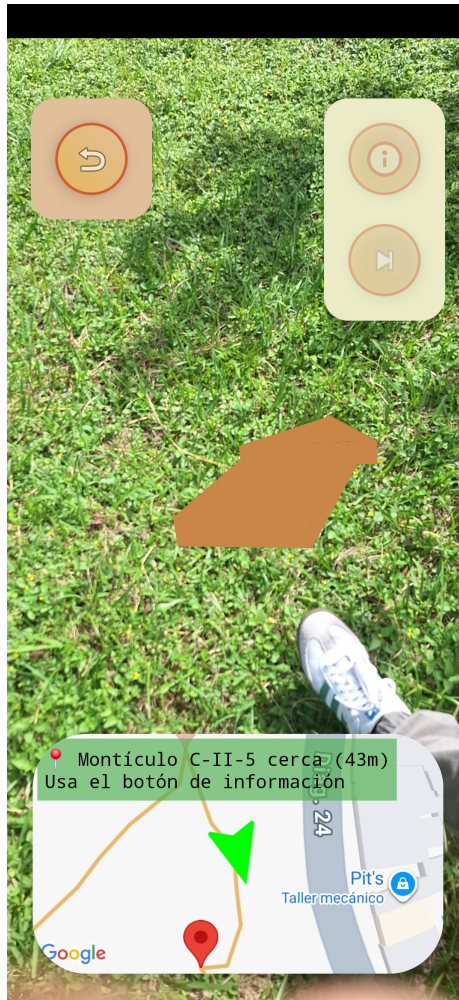


Figura 11: Orientación de la flecha hacia el siguiente destino

Esta visualización tridimensional complementó el sistema de navegación en pantalla, aportando un refuerzo visual dentro del entorno RA sin necesidad de recurrir únicamente a elementos planos como el mapa. Su presencia ayudó a guiar al usuario de forma natural dentro del recorrido simulado.

La flecha de navegación tridimensional demostró ser eficaz como elemento de guía dentro del recorrido de RA. Su orientación dinámica basada en el vector de dirección entre el usuario y el siguiente montículo permitió que la navegación se mantuviera intuitiva sin depender exclusivamente del mapa bidimensional. El modelo se mantuvo en todo momento sobre el plano físico, orientado correctamente a lo largo del trayecto, y actualizándose en tiempo real conforme el usuario se aproximaba a nuevos objetivos.

Esta implementación sugiere que la inclusión de indicadores visuales tridimensionales puede complementar e incluso sustituir a interfaces tradicionales de navegación, especialmente en contextos de exploración patrimonial. La persistencia y estabilidad de la flecha durante todo el recorrido contribuyó significativamente a reducir la carga cognitiva del usuario, permitiendo una experiencia más fluida y natural.

## F. Síntesis y alcances de la implementación

Los resultados observados en este proyecto demuestran que es viable desarrollar sistemas de realidad aumentada para entornos arqueológicos utilizando tecnologías abiertas y accesibles. La implementación de visualización tridimensional en tiempo real mediante ARCore permitió validar múltiples aspectos funcionales y técnicos asociados al despliegue, manipulación e integración de modelos en dispositivos móviles.

El uso de Android Studio junto con Kotlin y ARCore proporcionó una base sólida para el desarrollo, manejando cuidadosamente el ciclo de vida de las sesiones y el control de recursos. Las pruebas de rendimiento realizadas a través del *profiler* de Android Studio evidenciaron que el uso de CPU y GPU se mantuvo dentro de márgenes aceptables durante sesiones prolongadas, lo cual es indicativo de una arquitectura bien optimizada para dispositivos móviles.

La combinación de anclajes espaciales precisos, modelos optimizados, interacción gestual fluida y despliegue contextual de información permite crear experiencias inmersivas que pueden aplicarse tanto en campo como en entornos educativos o museográficos. El hecho de implementar el renderizado de modelos en formato `.obj` de forma directa sin convertirlos a formatos binarios optimizados implicó retos en cuanto a gestión de memoria, tiempos de carga y manejo de texturas, pero el uso de estructuras específicas como `ModelController.kt` permitió lograr tiempos de carga inferiores a dos segundos por modelo.

Estos hallazgos pueden servir como base para proyectos futuros que busquen expandir las capacidades de la realidad aumentada en el ámbito del patrimonio, incorporando nuevos elementos como rutas guiadas, narración interactiva o visualización multicapas. La validación exitosa de la arquitectura técnica, la usabilidad del sistema de interacción, y la estabilidad del renderizado en condiciones reales confirman que las tecnologías móviles actuales son suficientemente maduras para soportar aplicaciones de RA con propósitos educativos y de divulgación cultural.

- Se logró integrar correctamente modelos tridimensionales en la aplicación móvil desarrollada para Android, utilizando la plataforma ARCore. Los modelos en formato .obj fueron cargados y renderizados exitosamente sobre el entorno físico, con representación visual coherente y posicionamiento espacial estable, cumpliendo el objetivo de visualización inmersiva mediante realidad aumentada.
- Se implementaron interacciones funcionales que permitieron al usuario manipular los modelos 3D mediante gestos táctiles intuitivos, incluyendo escalado, rotación y desplazamiento. Estas interacciones respondieron de forma fluida y fueron validadas mediante pruebas de usabilidad, que reflejaron una alta comprensión por parte de los usuarios. Además, se incorporó retroalimentación háptica para reforzar la experiencia de exploración.
- Los modelos fueron anclados mediante mecanismos de posicionamiento espacial precisos proporcionados por ARCore. Se emplearon anclajes estáticos adaptados a planos físicos detectados en el entorno real, logrando una alineación coherente entre los objetos virtuales y el contexto arqueológico, tanto en pruebas simuladas como en campo.
- La aplicación fue evaluada en distintos dispositivos Android, confirmando su estabilidad, compatibilidad y rendimiento técnico. Se alcanzaron tasas de renderizado superiores a 45 FPS en equipos de gama media, sin interrupciones críticas ni fallos en la carga de modelos, validando el funcionamiento óptimo del sistema en condiciones reales de uso.

## **A. Recomendaciones técnicas**

### **1. Optimización de formatos de modelos 3D**

Se recomienda utilizar modelos tridimensionales en formatos más optimizados para entornos móviles, como `.glb` o `.usdz`, que permiten compresión binaria, soporte directo de materiales y menor carga en tiempo de ejecución. Esto podría reducir los tiempos de carga inicial y mejorar el desempeño en dispositivos de gama baja.

### **2. Validación automática de recursos**

Es aconsejable incorporar una capa de validación automática al cargar modelos 3D, que verifique aspectos como orientación, escala, integridad geométrica y materiales faltantes antes de renderizarlos. Este mecanismo podría prevenir errores visuales durante la ejecución, especialmente en implementaciones que incluyan modelos nuevos o generados por terceros.

## **B. Recomendaciones de evaluación y experiencia de usuario**

### **1. Pruebas en condiciones reales**

Se sugiere realizar pruebas de usabilidad en condiciones más cercanas al entorno real, incluyendo variaciones de luz, clima y superficies poco texturizadas, a fin de validar la robustez del anclaje y la capacidad de detección de planos por parte de ARCore en situaciones de campo más exigentes.

## **2. Mejoras en la retroalimentación visual**

Es recomendable ampliar el sistema de retroalimentación al usuario mediante componentes visuales complementarios como indicadores de carga, confirmaciones visuales de anclaje o animaciones al finalizar gestos. Esto contribuiría a una mejor comprensión de la interacción, especialmente para usuarios no familiarizados con entornos de realidad aumentada.

## **3. Sistema de analítica y métricas**

Se sugiere instrumentar un sistema de analítica *in-app* que registre eventos clave como tiempo de visualización, número de gestos ejecutados o frecuencia de uso por modelo. Esta información permitiría cuantificar la interacción de los usuarios y tomar decisiones informadas para optimizar la experiencia futura.

## **C. Recomendaciones metodológicas**

### **1. Ampliación de pruebas de validación**

Desde el punto de vista metodológico, se recomienda estructurar una batería de pruebas más amplia que incluya métricas de rendimiento detalladas, pruebas con usuarios diversos por perfil demográfico y escenarios comparativos con aplicaciones similares, para fortalecer la validez externa de los resultados obtenidos.

### **2. Documentación técnica integral**

Se aconseja documentar exhaustivamente cada módulo técnico implementado mediante diagramas de arquitectura, descripciones funcionales y bitácoras de cambios. Esta documentación favorecerá la mantenibilidad del sistema y facilitará la transferencia de conocimiento en futuras fases del proyecto o en iniciativas derivadas.

## **D. Líneas de investigación y trabajo futuro**

El desarrollo realizado en este trabajo abre múltiples oportunidades para investigación y expansión de las capacidades de realidad aumentada aplicadas al patrimonio cultural. Las siguientes líneas de trabajo representan extensiones naturales que podrían fortalecer significativamente el impacto y alcance de la solución implementada:

### **1. Expansión a otros sitios arqueológicos**

La metodología y arquitectura técnica desarrolladas pueden adaptarse para su implementación en otros sitios arqueológicos de Guatemala y la región mesoamericana. Sitios

como Tikal, El Mirador, Quiriguá, Yaxhá, y Copán presentan características arqueológicas y geográficas que permitirían aprovechar la misma base tecnológica, con las adaptaciones correspondientes en contenido cultural, modelos tridimensionales específicos, y sistemas de geolocalización ajustados a cada contexto particular. Esta expansión requeriría colaboración interdisciplinaria con arqueólogos especializados en cada sitio, así como estudios de viabilidad técnica y cultural.

## **2. Integración con otras tecnologías emergentes**

La convergencia de la realidad aumentada con tecnologías como realidad virtual (VR), internet de las cosas (IoT), e inteligencia artificial ofrece oportunidades para crear experiencias más inmersivas y contextualmente inteligentes. La integración con sensores IoT podría proporcionar datos ambientales en tiempo real (temperatura, humedad, calidad del aire) que complementen la experiencia educativa. Los sistemas de IA podrían personalizar el contenido según el perfil del visitante, nivel de conocimiento previo, o intereses específicos, mientras que la realidad virtual permitiría recrear construcciones arqueológicas desaparecidas o inaccesibles físicamente.

## Referencias bibliográficas

- [1] V. Vlahakis et al., “ARCHEOGUIDE: An augmented reality guide for archaeological sites,” *IEEE Computer Graphics and Applications*, vol. 22, n.º 5, págs. 52-60, 2002. DOI: 10.1109/MCG.2002.1028726
- [2] G. Papagiannakis y N. Magnenat-Thalmann, “Mobile augmented heritage: Enabling human life in ancient Pompeii,” *International Journal of Architectural Computing*, vol. 5, n.º 2, págs. 396-415, 2007. DOI: 10.1260/1478-0771.5.2.396
- [3] G.-D. Voinea, F. Gîrbacia, C. C. Postelnicu y A. Marto, “Exploring cultural heritage using augmented reality through Google’s Project Tango and ARCore,” en *VR Technologies in Cultural Heritage*, Springer, 2018, págs. 93-106. DOI: 10.1007/978-3-030-05819-7\_8
- [4] M. Muñoz-Sajama, D. Aracena-Pizarro, R. Cornejo-Mejías y M. Navarrete-Álvarez, “Una aplicación de Realidad Aumentada para recorrer el sitio patrimonial ‘Aldea de San Lorenzo’,” *Ingeniare. Revista Chilena de Ingeniería*, vol. 26, n.º s1, págs. 133-142, 2018. DOI: 10.4067/S0718-33052018000500065
- [5] M. K. Bekele, R. Pierdicca, E. Frontoni, E. S. Malinverni y J. Gain, “A survey of augmented, virtual, and mixed reality for cultural heritage,” *Journal on Computing and Cultural Heritage*, vol. 11, n.º 2, págs. 1-36, 2018. DOI: 10.1145/3145534
- [6] H. D. Turkoğlu y N. Çakıcı Alp, “Evaluating cultural heritage preservation through augmented reality: Insights from the Kaisareia-AR application,” *Architecture*, vol. 5, n.º 3, pág. 59, 2025. DOI: 10.3390/architecture5030059
- [7] R. T. Azuma, “A survey of augmented reality,” *Presence: Teleoperators and Virtual Environments*, vol. 6, n.º 4, págs. 355-385, 1997. DOI: 10.1162/pres.1997.6.4.355
- [8] E. Kruijff, J. E. Swan y S. Feiner, “Perceptual issues in augmented reality revisited,” en *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, IEEE, 2010, págs. 3-12. DOI: 10.1109/ISMAR.2010.5643530

- [9] D. Schmalstieg y T. Hollerer, *Augmented Reality: Principles and Practice*. Boston, MA: Addison-Wesley Professional, 2016.
- [10] K. Shoemake, “Animating rotation with quaternion curves,” *ACM SIGGRAPH Computer Graphics*, vol. 19, n.º 3, págs. 245-254, 1985. DOI: 10.1145/325165.325242
- [14] T. Möller, E. Haines y N. Hoffman, *Real-Time Rendering*, 4.ª ed. Boca Raton, FL: CRC Press, 2018.

## Referencias de internet

- [11] Google Developers. “ARCore overview,” visitado 15 de oct. de 2025. dirección: <https://developers.google.com/ar>
- [12] JetBrains. “Kotlin documentation,” visitado 18 de oct. de 2025. dirección: <https://kotlinlang.org/docs/home.html>
- [13] Android Developers. “SensorManager and gesture detection documentation,” visitado 25 de oct. de 2025. dirección: <https://developer.android.com/reference/android>
- [15] Khronos Group. “glTF 2.0 Overview,” visitado 20 de oct. de 2025. dirección: <https://www.khronos.org/glTF>
- [16] Blender Foundation. “Blender User Manual,” visitado 22 de oct. de 2025. dirección: <https://docs.blender.org/manual/en/latest/>

## **A. Recursos digitales complementarios**

Este anexo recopila enlaces a recursos en línea utilizados o generados durante el desarrollo del proyecto. Su finalidad es facilitar la consulta externa del código fuente y encuestas aplicadas como parte del proceso de validación.

### **1. Repositorio de código fuente del proyecto**

**URL:** <https://github.com/mvrcentes/AR-Tour-Kaminaljuyu.git>

Contiene el código desarrollado en Kotlin, documentación técnica por módulo y estructura del proyecto en Android Studio.

### **2. Encuesta de validación técnica y de interacción**

**URL:** <https://forms.cloud.microsoft/r/W1j0F4GZdv>

Formulario aplicado a los participantes durante la prueba de campo simulada en la universidad.

## **B. Registro fotográfico de las pruebas de usabilidad**

Como parte del proceso de validación, se documentó fotográficamente la interacción de los usuarios con la aplicación durante las pruebas de usabilidad realizadas en el campus universitario. Las siguientes imágenes muestran a los participantes utilizando la aplicación

AR Tour Kaminaljuyú, evaluando aspectos como la facilidad de uso, la respuesta gestual del sistema y la experiencia general de navegación en realidad aumentada.

Estas pruebas fueron fundamentales para obtener retroalimentación directa sobre el comportamiento de la aplicación en condiciones reales de uso, permitiendo validar tanto la funcionalidad técnica como la experiencia de usuario del sistema desarrollado.

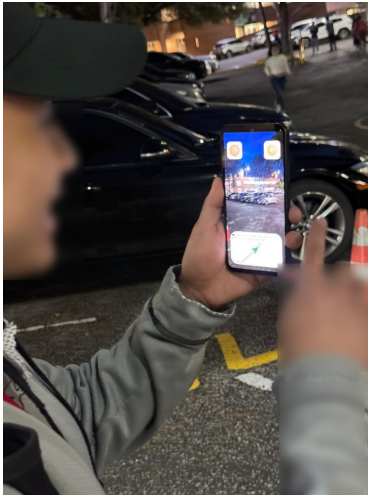


Figura 12: Usuario durante la prueba de usabilidad



Figura 13: Usuarios realizando la encuesta de validación



Figura 14: Usuarios durante la prueba de usabilidad

- .apk** (*Android package kit*) Archivo ejecutable que contiene todos los componentes necesarios para instalar una aplicación en dispositivos Android, incluyendo código compilado, recursos gráficos y archivos de manifiesto. 15..
- .obj** (*wavefront object file format*) Formato de archivo estándar para representar geometría tridimensional. Describe vértices, caras, normales y coordenadas UV mediante texto plano. Es ampliamente soportado por motores gráficos y *software* de modelado como Blender. 12, 14..
- anchor** (**ARCore**) Elemento que representa una posición fija en el espacio físico. Permite anclar objetos virtuales a coordenadas persistentes detectadas por ARCore, conservando su alineación incluso cuando el usuario se mueve. 12, 14..
- API** (*application programming interface*) Conjunto de funciones, clases o métodos predefinidos que permiten a los desarrolladores interactuar con sistemas o bibliotecas sin conocer su implementación interna. 11, 14..
- ARCore** Plataforma de desarrollo de realidad aumentada creada por Google para dispositivos Android. Ofrece funciones como rastreo de movimiento, detección de planos y anclaje de objetos virtuales en el entorno. 11, 12, 14..
- Blender** *Software* de código abierto utilizado para modelado, animación, texturizado y renderizado de objetos tridimensionales. Es compatible con múltiples formatos como .obj y .fbx. 13, 15..
- clase** (*class*) Unidad estructural en programación orientada a objetos. Define un tipo de dato personalizado compuesto por atributos (estado) y métodos (comportamiento). En Kotlin, se declaran con la palabra clave `class`. 15..
- corrutina** Función asincrónica ligera utilizada en Kotlin para ejecutar tareas sin bloquear el hilo principal. Permite operaciones como carga de archivos o espera por eventos. 14..

**FPS (*frames per second*)** Unidad que mide la cantidad de cuadros renderizados por segundo en una aplicación gráfica. Una tasa mayor a 30 FPS se considera fluida en contextos móviles. 13, 15..

**GestureDetectorCompat** Clase de Android que detecta gestos del usuario como toques simples, arrastres o giros. Se utiliza para capturar interacciones táctiles de forma compatible con distintas versiones de Android. 14..

**HelloGeoRenderer.kt** Clase personalizada en el proyecto que gestiona el ciclo de vida del renderizado de objetos 3D en el espacio RA. Se encarga de mantener la escena gráfica y actualizar sus elementos. 14..

**Kotlin** Lenguaje de programación moderno, conciso y seguro, compatible con la plataforma Android. Permite programación orientada a objetos y funcional, y es recomendado oficialmente por Google para desarrollo Android. 11, 14..

**matriz homogénea  $4 \times 4$**  Estructura matemática utilizada para realizar transformaciones tridimensionales como rotación, escala y traslación en gráficos 3D. Permite combinar múltiples transformaciones en una sola operación. 11, 13, 14..

**ModelController.kt** Clase responsable de cargar, transformar y renderizar modelos 3D en el entorno aumentado. Maneja operaciones como rotación, escalado, posicionamiento y asociación con anclas espaciales. 14..

**optimización gráfica** Proceso de reducir la complejidad de un modelo o escena 3D para mejorar el rendimiento, sin comprometer significativamente su calidad visual. Incluye técnicas como simplificación de mallas, compresión de texturas y reducción de *draw calls*. 12, 13..

**RA (realidad aumentada)** Tecnología que permite integrar contenido digital interactivo sobre el entorno físico a través de un dispositivo con cámara, sensores y procesador gráfico. 10–14..

**renderizado** Proceso mediante el cual un objeto tridimensional es transformado en una imagen visible en pantalla, con perspectiva, luz y texturas aplicadas. 12, 14..

**sensor inercial** Dispositivo dentro del móvil que mide aceleración, orientación y movimiento. Es utilizado por ARCore para combinar información visual con datos físicos del entorno y mejorar el rastreo. 11..

**textura (*texture mapping*)** Imagen 2D aplicada sobre la superficie de un modelo 3D para simular materiales reales como piedra, madera o metal. 12, 13..

**transformación espacial** Operación matemática que modifica la posición, rotación o escala de un objeto en un sistema de coordenadas tridimensional. Se representa mediante matrices homogéneas. 11–15..

**vibrator (Android)** Clase del sistema Android que permite activar el motor de vibración del dispositivo. Se utiliza para generar retroalimentación háptica durante la interacción. 15..