

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

**Diseño de prototipo de Aplicación para visualización de
productos en Realidad Aumentada.**

Javier Santiago Cáceres Sánchez

Ingeniería en Sistemas

Trabajo de integración curricular presentado como requisito
para la obtención del título de
Ingeniero en Sistemas

Quito, 19 de diciembre de 2019

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ
COLEGIO CIENCIAS E INGENIERÍAS

**HOJA DE CALIFICACIÓN
DE TRABAJO DE INTEGRACIÓN CURRICULAR**

**Diseño de prototipo de Aplicación para visualización de
productos en Realidad Aumentada.**

Javier Santiago Cáceres Sánchez

Calificación:

Nombre del profesor, Título académico:

Daniel Riofrío, Doctor of Philosophy
in Computer Science

Firma del profesor:

Quito, 19 de diciembre de 2019

Derechos de Autor

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma del estudiante: _____

Nombres y apellidos: Javier Santiago Cáceres Sánchez

Código: 00121011

Cédula de identidad: 1804421004

Lugar y fecha: Quito, 19 de diciembre de 2019

RESUMEN

El presente documento describe el desarrollo de un prototipo de aplicación móvil multiplataforma que tiene como objetivos permitir al usuario la visualización de productos e interacción con estos en Realidad Aumentada. El modelo desarrollado utiliza técnicas de detección de superficies (SLAM) para la superposición de elementos interactivos en el mundo real. Incluye además un sistema de recolección de datos de uso de aplicación para el entendimiento del comportamiento del usuario y la generación de información útil para la mercadotecnia del producto.

Palabras clave: Realidad Aumentada, SLAM, móvil, análisis de datos, markerless AR.

ABSTRACT

This document describes the development of a multiplatform mobile application prototype that allows users to visualize products and interact with them in Augmented Reality. The developed model uses surface detection techniques (SLAM) for the superposition of interactive elements in the real world. It also includes an application usage data collection system for the understanding of user behavior and the generation of useful information for product marketing.

Keywords: Augmented Reality, SLAM, mobile, data analysis, markerless AR.

TABLA DE CONTENIDO

INTRODUCCIÓN	8
DESARROLLO DEL TEMA	11
ANÁLISIS DE REQUERIMIENTOS	11
DISEÑO	13
Desarrollo nativo vs plataforma.....	13
Motor de Realidad Aumentada	14
Interfaz de Usuario, Experiencia de Usuario	14
Recolección de datos.....	15
Compartir Experiencia en Redes Sociales	15
IMPLEMENTACION	17
Carga de escenas con Operaciones Asíncronas	17
Pantalla de selección del producto	19
Integración Realidad Aumentada y SLAM.....	20
Recolección de datos de uso de aplicación	23
VERIFICACIÓN	24
CONCLUSIONES Y TRABAJO FUTURO	25
CONCLUSIONES	25
TRABAJO A FUTURO	26
REFERENCIAS BIBLIOGRÁFICAS	27

ÍNDICE DE FIGURAS

Figura 1. Casos de Uso	12
Figura 2. Prototipo de flujo de pantalla para carga de pantalla de Realidad Aumentada desarrollado en Adobe XD.....	13
Figura 3. Diagrama de Componentes de Software	16
Figura 4. Pantalla de carga asíncrona con información estratégica.....	18
Figura 5. Pantalla de carga de aplicación asíncrona con logo de marca.....	18
Figura 6. Pantalla de selección de modelo de auto para su visualización en Realidad Aumentada	19
Figura 7. A: Indicador de Detección de Superficies Planas sobre el ambiente. B: Objeto ubicado sobre superficie plana detectada.....	21
Figura 8. Extracto de código para la integración del sistema de detección de superficies en Realidad Aumentada.....	22
Figura 9. Extracto de código para el envío de eventos personalizados al servidor de Unity Analytics	23
Figura 10. A: Captura aplicación en dispositivo iOS: iPad Pro. B: Captura aplicación en dispositivo Android: Huawei p30	24

ÍNDICE DE TABLAS

Tabla 1. Descripción de las herramientas utilizadas.....	16
--	----

INTRODUCCIÓN

La Realidad Aumentada en la última década se ha presentado como una tecnología disruptiva, en constante desarrollo y evolución por la demanda tanto del consumidor como la industria con el objetivo de cambiar de manera dramática la manera en que resolvemos problemas e interactuamos con nuestro alrededor en áreas como la educación, el comercio electrónico, la medicina, entre otros. Esta tecnología consiste en la integración de información digital como sonidos, objetos generados por computadora, texto y efectos para intensificar o “aumentar” la experiencia del usuario en el mundo real (Kanuganti: 2019).

Según proyecciones, el mercado para la Realidad Aumentada sólo en la industria de la compra y venta de productos alcanzará 1.6 billones de dólares en 2025, en base a ese gran potencial de mercado, gigantes de la tecnología han invertido en el desarrollo y despliegue de herramientas de desarrollo orientados a la Realidad Aumentada como el ARkit de Apple para dispositivos iOS, ARCore de Google para dispositivos Android, además de planes de desarrollo de lentes con Realidad Aumentada (Towers: 2018). Estas herramientas y la tecnología actual brindan a los desarrolladores la oportunidad de crear experiencias en Realidad Aumentada con más facilidad y alta disponibilidad para los usuarios de dispositivos móviles sin necesidad de otros dispositivos y sensores especializados.

Según la revista Forbes, y en base a las proyecciones del mercado mencionadas anteriormente, los negocios de compra y venta de productos deben adoptar e incorporar la Realidad Aumentada dado que esta tecnología permitirá su crecimiento y el incremento de ganancias. La implementación de Realidad Aumentada y Realidad Virtual ayuda tanto a vendedores como consumidores a tener una mejor experiencia de compraventa, disminuyendo

costos operacionales y ofreciendo experiencias inmersivas y atractivas respectivamente. Como Kanuganti explica, usando realidad aumentada los consumidores pueden conocer el tamaño y la apariencia del producto de forma realista antes de realizar la compra en línea, incluyendo el cómo se verá y funcionará el producto en la ubicación deseada por el comprador. Por ejemplo, con la aplicación Dulux Visualiser, los consumidores pueden probar diferentes tonos y colores de pintura para su habitación antes de comprarla. Usando la cámara de un teléfono inteligente, se escanea la habitación y se “pinta” con cualquier color (2019).

Otro ejemplo destacado es lo que hace Macy's al ofrecer muestreo cosmético en Realidad Aumentada frente a la dificultad del consumidor de encontrar el tono correcto entre las docenas de tonos de cosméticos disponibles en una tienda. Las muestras pueden ayudar, pero las tiendas no siempre tienen muestras para cada color o producto. Sin embargo, Macy's ayuda a las personas a probar el maquillaje con el uso de realidad aumentada, ya que pueden hacerlo a través de una aplicación móvil que proporciona más de 1,000 tonos. Este enfoque permite a las personas comprender qué opciones son más adecuadas para ellos, incluso cuando están en casa y prefieren comprar cosas en línea. Sin esta opción, las personas pueden decidir comprar más tarde o no comprarlas, pero, ver las posibilidades de sombra en tiempo real estimula la necesidad de comprar (Matthews, 2019).

Entre los beneficios de la Realidad Aumentada para la industria de compraventa de productos resaltan los siguientes (Naveen, 2019):

- Refuerza la confianza del consumidor.
- Provee experiencias personalizadas.

- Incrementa la satisfacción del consumidor.
- Habilita nuevas metodologías de mercadeo y publicidad.

El presente documento describe el desarrollo de un prototipo de aplicación móvil multiplataforma que permite a una compañía o marca de autos presentar al consumidor un catálogo con la característica de visualizar cada modelo de auto e interactuar con el mismo en Realidad Aumentada (cambiando el color, tamaño, variaciones, etc). Además, el prototipo descrito incluye técnicas de recolección de datos de uso de aplicación para obtener retroalimentación del comportamiento del usuario en la aplicación, información útil, por ejemplo, para conocer los productos más explorados dentro de la aplicación. Se presenta también un breve análisis de las opciones disponibles hasta la fecha en cuanto al diseño de una aplicación de esta naturaleza como los motores de Realidad Aumentada, su capacidad, así como los ambientes y lenguajes de programación. Con el propósito de documentar el proceso de desarrollo del prototipo el siguiente capítulo se apega al modelo clásico: análisis de requerimientos, diseño, implementación y verificación.

DESARROLLO DEL TEMA

ANÁLISIS DE REQUERIMIENTOS

Con el objetivo de obtener el listado de requerimientos para la aplicación se planifican reuniones con el cliente o representante de la empresa cliente. Estas reuniones son de vital importancia dado que los requerimientos deben reflejar de manera clara el propósito del producto final, las expectativas del cliente en cuanto a su capacidad, disponibilidad, tiempo deseado de publicación, costos aproximados, limitaciones, entre otros aspectos. Para el prototipo desarrollado los requerimientos son los siguientes:

- Compatibilidad con celulares inteligentes con sistemas operativos Android y iOS.
- Visualización en Realidad Aumentada sin uso de marcadores de 3 modelos de autos.
- La aplicación debe contar con información relevante de cada modelo de auto.
- Sistema de recolección de datos de uso para reconocimiento de los modelos de auto más explorados por los usuarios.

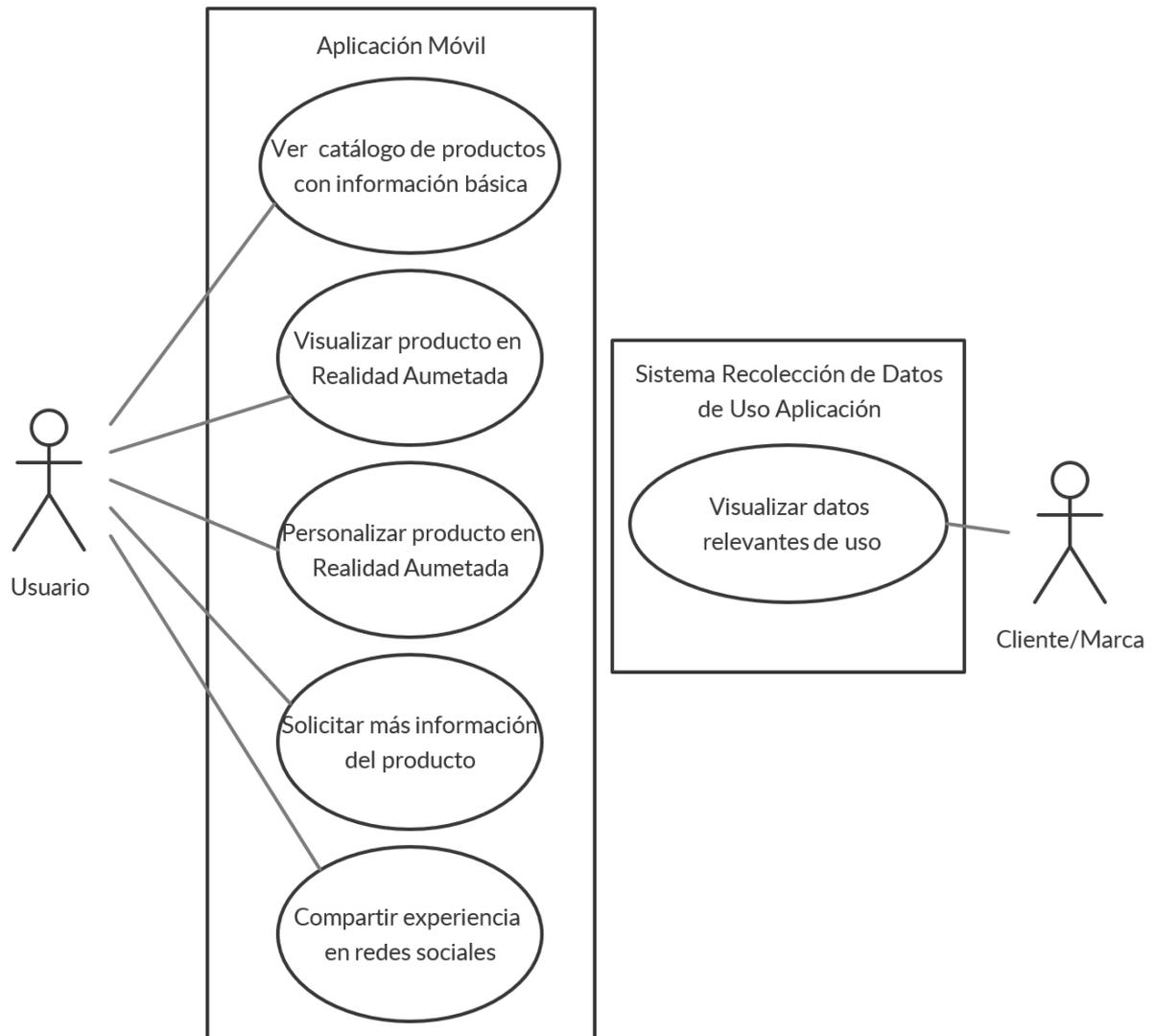


Figura 1. Casos de Uso

En base a los requerimientos se determinan los casos de uso descritos en la Figura 1 y se genera una propuesta del flujo de pantallas usando la herramienta Adobe XD que permite además simular interacciones y animaciones entre las pantallas de la aplicación, facilitando la percepción del cliente de cómo la aplicación se vería y como funcionaría una vez desarrollada (Adobe.com, 2019). El objetivo de este prototipado rápido es permitir al cliente aprobar la propuesta o exigir cambios de manera temprana. Un ejemplo del prototipado se muestra en la

Figura 2 que muestra un extracto del flujo de pantallas de la propuesta inicial para el cliente: una pantalla de carga con información relevante de los modelos de autos previa a la pantalla de visualización e interacción con los modelos en realidad aumentada.

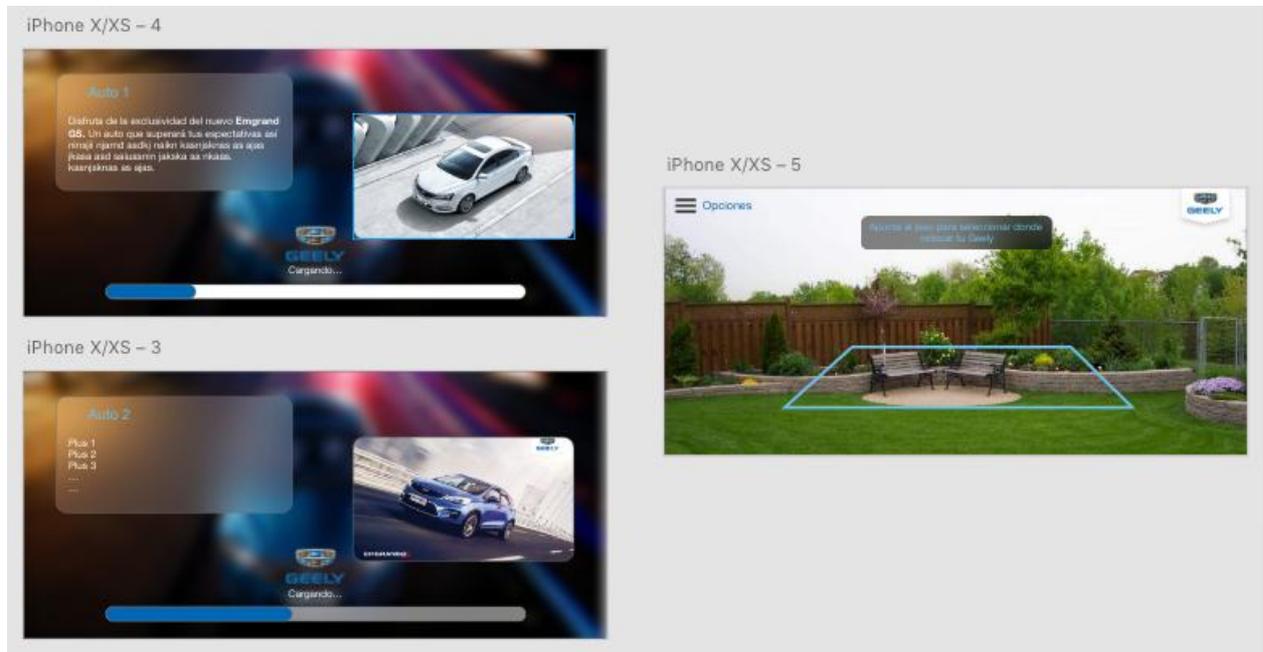


Figura 2. Prototipo de flujo de pantalla para carga de pantalla de Realidad Aumentada desarrollado en Adobe XD

DISEÑO

Desarrollo nativo vs plataforma

La aplicación, bajo los requerimientos del cliente, debe ser compatible con dispositivos Android y iOS para asegurar su disponibilidad a la mayor cantidad de usuarios, especialmente si el producto forma parte de una campaña de marketing de la empresa cliente. Se consideraron 2 opciones: realizar el desarrollo nativo a cada plataforma o realizar el desarrollo multiplataforma, es escogió esta última opción como la adecuada dado el reducido tiempo de producción. Se utilizó Unity3D como software de desarrollo multiplataforma dada su compatibilidad con motores de realidad aumentada siendo C# el lenguaje de programación.

Motor de Realidad Aumentada

Para la realidad aumentada se consideraron distintos motores: Vuforia, Unity AR Foundation, Wikitude, cada uno de estos ofrece un API para la implementación de realidad aumentada con SLAM, sistema que permite la detección de planos y superficies haciendo uso de la cámara y sensores de profundidad de dispositivos móviles. El costo aproximado por aplicación de Vuforia y Wikitude es de \$500, por otro lado, Unity AR Foundation es gratuito. Luego de realizar pruebas rápidas se consideró a Unity AR Foundation como la mejor opción en términos de rendimiento y compatibilidad con dispositivos Android y iOS. Esto se debe a que Unity AR Foundation está construido por subsistemas implementados sobre ARCore y ARKit que son tecnologías de realidad aumentada nativas a Android y iOS respectivamente (Unity, 2019).

Interfaz de Usuario, Experiencia de Usuario

El diseño de la Interfaz de Usuario (UI) va de la mano del diseño de Experiencia de Usuario (UX), ésta última determina el flujo de pantallas y los casos de uso del software que se va a desarrollar. Existen recomendaciones que aseguran una adecuada interacción y determinan una buena experiencia de usuario, por ejemplo: contenido legible, navegación intuitiva, elementos claramente visibles, entre otros (Chapman, 2019). Cabe recalcar que, al momento de diseñar la experiencia de usuario, es muy importante resaltar los elementos de interés del cliente y la finalidad de la aplicación, por ejemplo, si se trata de una aplicación que permite la visualización de autos deportivos en realidad aumentada para aumentar las ventas de la marca, el diseño de UX y UI debe atraer la atención del usuario a la funcionalidad de la aplicación que permite interactuar con los autos en realidad aumentada y permitir agendar un “test drive” o una cita con un agente de la marca con facilidad.

Recolección de datos

En la actualidad se ha estandarizado la inclusión de sistemas de recolección de datos de uso de la aplicación para generar información sobre el comportamiento del usuario, permitiendo implementar mejoras además de obtener retroalimentación sobre la experiencia de usuario de manera indirecta. Para el prototipo de aplicación cuyo objetivo es dar a conocer y promover la venta de autos de una marca específica, la recolección de datos de uso puede ser efectiva para identificar el auto con el que el usuario interactúa más veces o durante un mayor tiempo, ayudando de esta manera a la empresa a promocionar a ese auto como prioridad en un nicho de mercado específico.

Para implementar la recolección de datos de uso de aplicación se disponen de distintas herramientas, cada una consiste en un servidor el cual recibe los datos de uso cuando se activa un evento desde los dispositivos de los usuarios al usar la aplicación. Entre otros, las dos opciones consideradas para la aplicación son Firebase de Google y Unity Analytics. Se opta por Unity Analytics dada su facilidad de integración y su gratuidad frente a Firebase, además, Unity Analytics ofrece el acceso web a un panel administrativo de la aplicación que permite la visualización de los datos de uso de una manera organizada y personalizable.

Compartir Experiencia en Redes Sociales

Uno de los requerimientos del prototipo para la visualización de autos en realidad aumentada es la capacidad de compartir la experiencia del usuario al usar la aplicación en redes sociales. Para ello es necesaria la implementación de un sistema de captura de pantalla que muestre el modelo de auto en el mundo real para poder compartir la "foto" en redes sociales. Un sistema como éste permite al usuario dar a conocer a sus amistades o familiares una posible compra de un auto o discutir sobre qué color o qué modelo es el más adecuado. Para la

implementación de este sistema se utiliza el paquete NativeShare compatible con Unity que permite compartir contenido de manera nativa en dispositivos Android y iOS.

Como resultado de las decisiones de Diseño respecto a la arquitectura de la aplicación para el cumplimiento de los requerimientos mencionados anteriormente se generó un diagrama de componentes de software ilustrado en la Figura 3.

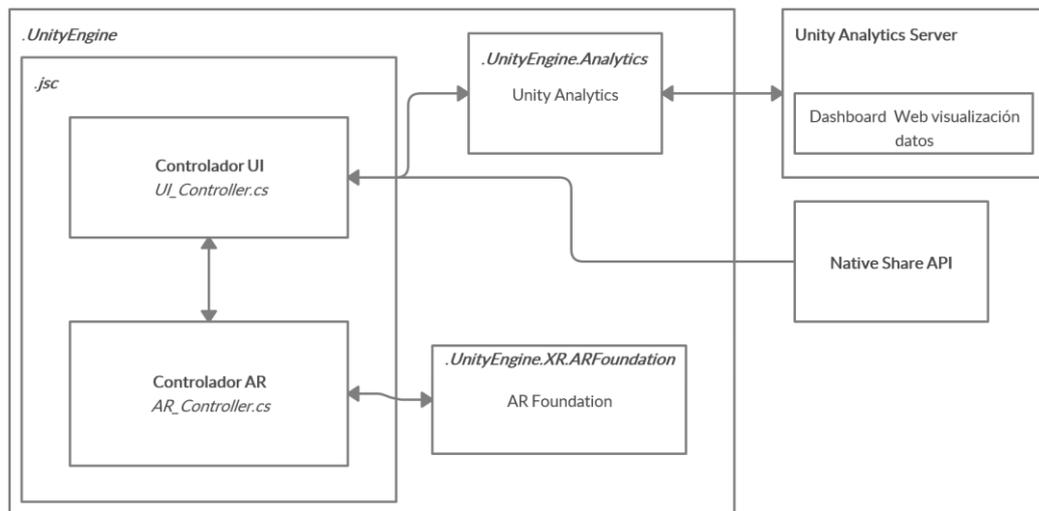


Figura 3. Diagrama de Componentes de Software

Herramienta	Versión	Descripción
Unity3D	2018	Ambiente de programación, permite el desarrollo de aplicaciones compatibles con iOS, Android, entre otros.
AR Foundation	2.2.0-preview	API de realidad aumentada construido sobre ArKit y ArCore para su uso adecuado en la plataforma correspondiente.
Unity Analytics	2019	API de Unity3D que permite almacenar en el Servidor de Unity datos de cada aplicación desarrollada, ofrece un dashboard en su plataforma web.
Native Share	1.14	API compatible con Unity que permite compartir archivos de manera nativa en iOS y Android

Tabla 1. Descripción de las herramientas utilizadas para la implementación del prototipo.

IMPLEMENTACION

La aplicación prototipo se basó en el siguiente flujo de pantallas, cuya implementación se describe en esta sección:

1. Pantalla de carga de aplicación con logo de la marca
2. Pantalla selección del producto y visualización de información.
3. Pantalla de carga de realidad aumentada.
4. Pantalla de realidad aumentada del producto seleccionado.

Carga de escenas con Operaciones Asíncronas

En Unity la carga de una escena, la cual es el equivalente a una actividad en el desarrollo nativo Android, puede tomar algunos segundos dependiendo de las características y complejidad de los objetos que conforman la misma. Por ejemplo, una escena con un arreglo de elementos en tres dimensiones, texturas, iluminación específica, etc., tardará un tiempo significativo mayor a una escena que consta de pocos elementos en dos dimensiones (imágenes, botones, texto). Por esta razón, y para evitar que la carga se desarrolle mientras se muestra una pantalla negra que afecta una buena experiencia de usuario, se opta por realizar la carga de manera asíncrona, es decir, se muestra una escena simple que permite al usuario conocer el estado de carga o el avance de la carga con cierta animación como se muestra en la Figura 4. La ventaja del uso de estas escenas intermedias durante carga de escenas complejas es que ofrecen la oportunidad de ubicar información estratégica que apoya el propósito de la aplicación, por ejemplo, para el caso de los autos, se incluye la marca de la compañía animada (Figura 3) en la carga inicial de la aplicación y las características mediante imágenes y texto animados relacionadas al modelo de auto seleccionado para la interacción en Realidad Aumentada (Figura 4).



Figura 4. Pantalla de carga asíncrona con información estratégica.



Figura 5. Pantalla de carga de aplicación asíncrona con logo de marca.

Pantalla de selección del producto

La pantalla de selección debe contar con una vista previa del producto y una descripción con información relevante. Para el prototipo se implementó una vista en tres dimensiones a manera de carrusel, con información dinámica del producto seleccionado en un panel flotante. Para la selección de los productos se implementaron dos opciones para el usuario: botones a manera de flechas en los costados de la pantalla y detección de acciones sobre la pantalla táctil: deslizar hacia la derecha o hacia la izquierda para cambiar el producto, una captura de esta pantalla de selección se muestra en la Figura 6.



Figura 6. Pantalla de selección de modelo de auto para su visualización en Realidad

Aumentada

Una vez seleccionado el producto, se confirma la selección con un botón que activa la carga de la escena de realidad aumentada. Dado que no es posible la transmisión directa de datos entre escenas en Unity3D, en este caso el índice del producto que se debe mostrar en realidad aumentada se opta por utilizar el sistema de preferencias de usuario que ofrece Unity3D. Este sistema consiste en guardar en memoria persistente del dispositivo un diccionario de datos, de este modo en la escena de selección de producto guardamos en el diccionario la clave "productoAR" con su índice como valor, al cual accedemos en la escena posterior.

Integración Realidad Aumentada y SLAM

Para la integración del sistema de mapeo y localización simultáneo (SLAM) se utilizó la librería AR Foundation de Unity. Para implementar la detección de la superficie en el mundo real donde se desea ubicar el producto en realidad aumentada se necesita de los siguientes elementos de la librería en la escena:

- **AR Session:** Controla el ciclo de vida de una experiencia AR, habilitando o deshabilitando AR en la plataforma destino. Además, permite conocer el estado de la experiencia AR como la compatibilidad con el dispositivo, estado de inicialización, etc (Unity, 2018).
- **AR Session Origin:** Su propósito es transformar las entidades rastreables (como superficies planas y puntos específicos en el ambiente) en su posición final, orientación y escala adecuada en la escena de Unity (Unity, 2018). A esta entidad debe estar relacionada la entidad **AR Camera**, que permite retornar los cuadros de imagen desde la cámara del dispositivo móvil para su análisis.

Una vez agregados estos elementos es necesaria la integración del sistema de reconocimiento de planos para la ubicación del producto en realidad aumentada, para esto es necesario

superponer sobre el espacio un indicador de un plano detectado como se muestra en la Figura 7 el cual permite al usuario saber que la aplicación está lista para ubicar un objeto sobre la superficie detectada. Cada cuadro de imagen es analizado mediante “raycasts” que son rayos que retornan booleanos indicando su colisión o no con ciertos elementos, en este caso colisión con los planos detectados en el ambiente real mediante la entidad **ARRaycastManager** provisto por la librería mencionada anteriormente. Si los raycasts detectan colisiones se confirma la detección de un plano en ese cuadro de imagen y se activa la visualización del indicador de plano como se aprecia en la figura 7. En los cuadros de imagen posteriores se calibra la posición del indicador o se desactiva su visualización si no se detecta una superficie. Cuando el indicador se encuentra activo el usuario puede ubicar el objeto en realidad aumentada tocando la pantalla.

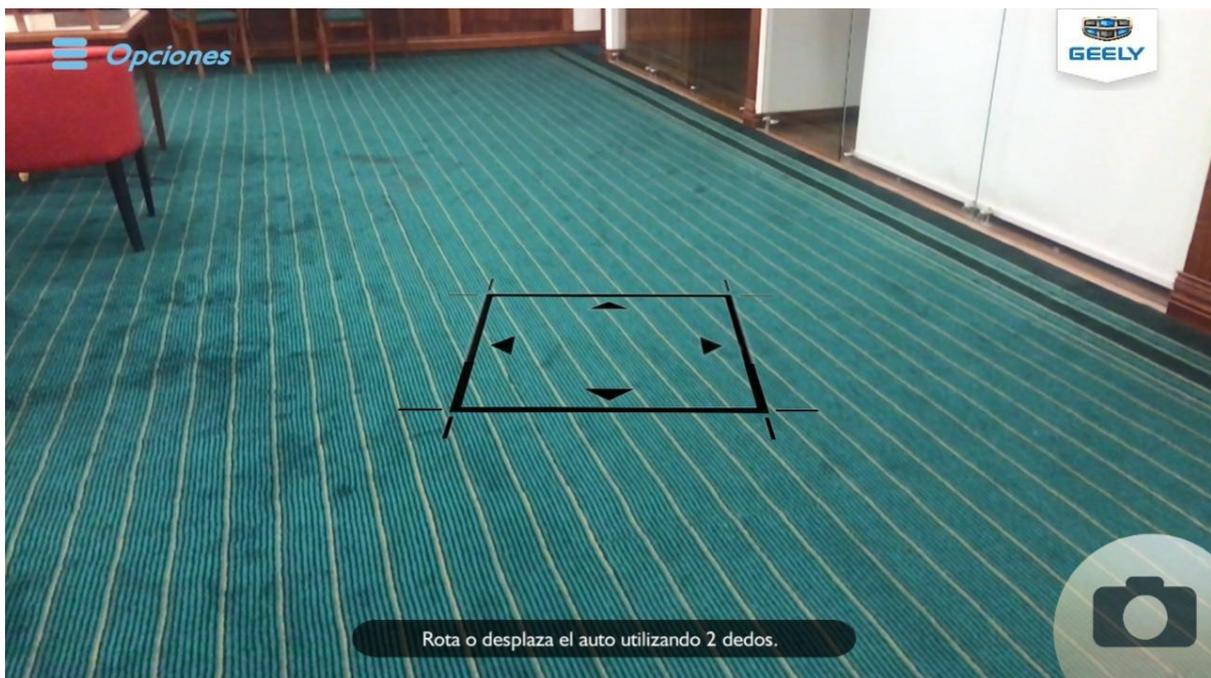


Figura 7. A: Indicador de Detección de Superficies Planas sobre el ambiente. **B:** Objeto ubicado sobre superficie plana detectada.

La figura 8 muestra un extracto de código con la implementación descrita, las funciones `Start` y `Update` son *callbacks* de la clase *MonoBehaviour* que es la clase base de la cual todo script Unity se deriva, **Start()** es llamado una sola vez en el primer frame cuando se carga una escena por lo cual es la función ideal para implementar la inicialización y en este caso encontrar la referencia a objetos como **ARRaycastManager** y otros necesarios; **Update()** por otro lado es llamado en cada cuadro, lo que permite actualizar la posición y calibrar el sistema de coordenadas en el que se encuentra el objeto en realidad aumentada respecto al entorno real, de igual manera esta función nos permite determinar si el usuario ha tocado la pantalla en un cuadro determinado para ubicar el producto que desea visualizar.

```

void Start()
{
    raycastManager = FindObjectOfType<ARRaycastManager>();
    m_GraphicRayCaster = FindObjectOfType<GraphicRaycaster>();
    m_EventSystem = FindObjectOfType<EventSystem>();

    toastInicial.SetTrigger("toast");
}

void Update()
{
    UpdatePlacementPose();
    UpdatePlacementIndicator();
    if(!IsCanvasButtonPressed())
    if (placementPoseIsValid && Input.touchCount > 0 && Input.GetTouch(0).phase ==
TouchPhase.Began)
    {
        if(!placed)
        PlaceObject();
    }
}

private void PlaceObject()
{
    placed = true;
    ObjectToPlace.transform.position = placementPose.position;
    ObjectToPlace.transform.rotation = placementPose.rotation;
    buttonTrash.SetActive(true);
    toastPlacement.SetTrigger("toast");
}

```

Figura 8. Extracto de código para la integración del sistema de detección de superficies en

Recolección de datos de uso de aplicación

La implementación de la recolección de datos consiste en métodos definidos programáticamente ejecutados cuando ocurren eventos, en el extracto de código que se muestra en la figura 9 se demuestra un ejemplo para la recolección de datos, en este caso se envía el tiempo total que el usuario interactuó con un modelo de auto específico en realidad aumentada, la función es llamada cuando ocurren ciertos eventos: al cerrar la pantalla o escena de realidad aumentada ya sea volviendo a la pantalla de selección de productos o cerrando la aplicación:

```
public void Send_AR_AnalyticsData()
{
    AnalyticsEvent.Custom("car_model_AR_interaction", new Dictionary<string,
object>{
        { "car_id", car_Id },
        { "time_viewing_"+car_Id, Time.timeSinceLevelLoad }
    });
}
```

Figura 9. Extracto de código para el envío de eventos personalizados al servidor de Unity Analytics

En el extracto de código de la figura 9 se envía al servidor de UnityAnalytics el evento “*car_model_AR_interaction*” a manera de diccionario con el identificador del producto: “*car-id*” y el tiempo que el usuario ha utilizado la escena de Realidad Aumentada con dicho producto: “*time_viewing_<<car_Id>>*”.

VERIFICACIÓN

Finalizada la implementación se realizaron pruebas del funcionamiento integral de la aplicación en plataformas Android y iOS. Este procedimiento permitió el reconocimiento de errores y comportamientos no deseados en cada componente de la aplicación para su inmediata corrección previa la publicación en las tiendas Google Play y AppStore. La figura 10 muestra capturas de pantalla de la aplicación en cada sistema operativo durante la etapa de pruebas.



Figura 10. **A:** Captura aplicación en dispositivo iOS: iPad Pro. **B:** Captura aplicación en dispositivo Android: Huawei p30.

CONCLUSIONES Y TRABAJO FUTURO

CONCLUSIONES

La aplicación prototipo cumple con los requerimientos establecidos inicialmente, el diseño de arquitectura implementado resultó muy adecuado y su rendimiento fue verificado correctamente en las plataformas destino: Android y iOS logrando 60 cuadros por segundo. El principal reto del prototipo fue su optimización a nivel general para disminuir el tamaño de la aplicación, así como los tiempos de carga. Para esto, se tomaron algunas medidas como la reducción de las dimensiones de las texturas de los materiales y reducción de polígonos en los 3 modelos de autos utilizados.

El uso de Unity como ambiente de programación resultó muy positivo especialmente para el control de versiones gracias a su servicio "Unity Collab", el cual permite hacer un seguimiento de los cambios implementados, así como la posibilidad de realizar un "roll back" a versiones anteriores si existe algún fallo o pérdida de contenido durante la implementación de la aplicación. De igual manera el uso de Unity Analytics facilitó la gestión y visualización de los datos de uso de aplicación recolectados desde los dispositivos de los usuarios, evitando la necesidad de implementar un sistema externo que se traduce además en costos adicionales y un mayor tiempo de producción.

La limitación principal de la aplicación prototipo desarrollada es su incompatibilidad con dispositivos de gama baja dado que estos no cuentan con el hardware necesario para soportar la tecnología de detección de superficies SLAM implementada.

TRABAJO A FUTURO

Para solucionar el problema de incompatibilidad con dispositivos de gama baja se puede desarrollar una versión de la aplicación prototipo con un sistema diferente para la realidad aumentada, por ejemplo, haciendo uso de un marcador o utilizando técnicas de “*instant tracking*” en lugar de SLAM. El desarrollo de una aplicación “lite” asegura un alcance más amplio en los usuarios interesados en utilizar la aplicación que no cuentan con dispositivos móviles de alta gama.

Como una posible mejora al prototipo desarrollado se puede implementar un sistema táctil de interacción con el modelo de auto en realidad aumentada adicional al menú con el que cuenta la aplicación, por ejemplo: tocar en las puertas del auto para que se abran, o tocar las luces para que se prendan/apaguen. Un sistema de interacción doble: táctil y con un menú puede resultar más intuitivo y fácil de usar, posibilitando una mejora importante en la experiencia del usuario y por tanto su satisfacción al utilizar la aplicación.

Finalmente, el prototipo desarrollado no cuenta con un sistema dinámico de carga y actualización de productos para su visualización en Realidad Aumentada, en este caso los modelos de los autos están guardados en memoria como parte del paquete de la aplicación. Un sistema de carga dinámica de productos desde web posibilitaría escalabilidad en el modelo planteado de visualización e interacción de productos en realidad aumentada. Además, un sistema dinámico de carga elimina la necesidad de recompilar la aplicación para la inclusión de nuevos productos, reduciendo también el tamaño de la aplicación en la memoria de los dispositivos.

REFERENCIAS BIBLIOGRÁFICAS

- Unity3D (2019). Unity ArFoundation Manual. Obtenido el 27 de noviembre de 2019 de <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@2.2/manual/index.html>
- Chapman, C. (2019). Mobile UX design principles and best practices. Obtenido el 27 de noviembre de 2019 de <https://uxdesign.cc/mobile-ux-design-principles-and-best-practices-594bec53a31e>
- Businesswire (2017). Global Mobile Augmented Reality Market to Showcase a CAGR of 77% Through 2021: Technavio. Obtenido el 27 de noviembre de 2019 de <https://www.businesswire.com/news/home/20170130005409/en/Global-Mobile-Augmented-Reality-Market-Showcase-CAGR>
- Tractica. (2016) Mobile Augmented Reality App Downloads to Reach 1.2 Billion Annually by 2019. Obtenido el 27 de noviembre de 2019 de <https://www.tractica.com/newsroom/press-releases/mobile-augmented-reality-app-downloads-to-reach-1-2-billion-annually-by-2019/>
- Naveen, J. (2019). Retailers Have A Lot To Gain From AR and VR. Obtenido el 27 de noviembre de 2019 de <https://medium.com/@shashvatshukla/framework-vs-library-vs-platform-vs-api-vs-sdk-vs-toolkits-vs-ide-50a9473999db>
- Wayman, R. (2019). Compound Annual Growth Rate: What You Should Know. Obtenido el 27 de noviembre de 2019 de <https://www.investopedia.com/investing/compound-annual-growth-rate-what-you-should-know/>