DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

OPCION I.- TESIS

TRABAJO PROFESIONAL

"Desarrollo de funcionalidades de marcadores del sistema generador de aplicaciones de realidad aumentada (ARAPP Builder)"

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA:
SERGIO ANTONIO CORNU TINOCO

DIRECTOR DE TESIS:
M.R.T. Ignacio López Martínez

CODIRECTOR DE TESIS:
Mtro. LUIS ANGEL REYES HERNÁNDEZ

ORIZABA, VER. MÉXICO

2020

Contenido

Índice de Figuras

Índice de tablas

Resumen

En la actualidad los *smartphones* (teléfonos inteligentes) han puesto al alcance del usuario promedio la tecnología de realidad aumentada, gracias que estos en tiempos recientes aumentaron la capacidad de procesamiento y han disminuido sus costos, haciendo que estos sean una plataforma viable para la ejecución de aplicaciones de realidad aumentada.

La plataforma web *ARAPP Builder* permite al usuario crear sus propias aplicaciones de realidad aumentada sin la necesidad de contratar un programador o grupo de estos, ya que con esta plataforma el usuario puede desarrollar una aplicación personalizada de realidad aumentada para el sistema operativo *Android*.

ARAPP Builder ofrece la creación de aplicaciones de realidad aumentada que hacen uso de marcadores que desencadenan los elementos a aumentar; sin embargo, existen nuevas tecnologías como *markless* (sin marca de la traducción del inglés), GPS (*Global Positioning System* por sus siglas en inglés) y NFC (*Near Field Communication* por sus siglas en inglés); para ampliar las características de generación de ARAPP *Builder* se integraran las tecnologías previamente mencionadas.

Introducción

En la actualidad el uso de aplicaciones de realidad aumentada ha ido en aumento gracias a que la tecnología se ha puesto al alcance del usuario promedio a través del uso de los smartphones (teléfonos inteligentes, comúnmente conocidos como celulares) que han tenido un aumento de capacidad de procesamiento y una disminución en su precio, esto ha permitido que si una persona cuenta con un smartphone promedio ya cuenta con un dispositivo capaz de hacer uso de la realidad aumentada.

Para el desarrollo de las aplicaciones de realidad aumenta se necesita contratar a un equipo de desarrollo que se haga cargo de la codificación, pruebas y correcciones de las aplicaciones, esto genera un gran costo que solo es permisivo para las grandes empresas y PYMEs desarrolladoras de aplicaciones, relegando al usuario promedio ya que este no cuenta normalmente con los recursos necesarios para el desarrollo.

Como solución a la problemática del usuario promedio al tratar de generar sus propias aplicaciones de realidad aumentada, se desarrolló la plataforma web ARAPP Builder, esta plataforma le permite al usuario promedio realizar sus propias aplicaciones de realidad aumentada sin la necesidad de contratar personal, lo que representa un gran ahorro para este.

ARAPP Builder tiene la característica de generar aplicaciones para el sistema operativo *Android*, estas aplicaciones han demostrado ser efectivas para acercar aún más al usuario final a la tecnología de realidad aumentada, es así como para darle más opciones de uso y con la intención de aumentar las características en las aplicaciones generadas se propone la integración de nuevas tecnologías como *markless*, GPS y NFC.

El presente trabajo se divide en 4 capítulos que consisten en: capítulo 1 se presentan los antecedentes del proyecto, así como el marco teórico y el planteamiento del problema; capítulo 2 se presenta el estado del arte sobre la realidad aumentada con base en artículos que hablen sobre esta y se presenta también un análisis comparativo; capítulo 3 en este capítulo se presenta la aplicación de la metodología seleccionada para el desarrollo del proyecto; capítulo 4 se presentan los resultados de los casos de estudio seleccionados y por último punto del documento se presentan las conclusiones de este.

Capítulo 1 Antecedentes.

1.1 Marco Teórico

En el presente capítulo se detallan los conceptos más importantes para el desarrollo del proyecto de tesis, ofreciendo así una mejor comprensión sobre el mismo.

1.1.1 Realidad Aumentada

Como lo indica [1], la realidad aumentada (RA) se describe cómo “mezclar objetos con entes generados virtualmente en el mismo ambiente”, estos entes generados virtualmente pueden llegar a presentarse en la forma de imágenes, texto, sonidos o videos, para que estos sean observados por los usuarios utilizando algún dispositivo capaz de ejecutar las aplicaciones de RA. Entre las áreas de aplicación que tiene la RA se encuentran: el sector salud, en el área de mantenimiento industrial [2], en el sector automovilístico [3], en el área de ingeniería civil [4], así como también en el área de ventas [5].

En años recientes las aplicaciones de RA han presentado un aumento en su desarrollo y en su uso por parte del público en general gracias al aumento del uso de los dispositivos móviles [6], ya sean teléfonos inteligentes, tablets o consolas de juegos portátiles.

La RA se divide en dos grandes exponentes los cuales son la RA basada en imágenes (marcadores o códigos especiales diseñados) que es el medio más utilizado para hacer uso de esta y el segundo exponente que utiliza la geolocalización para mostrar información al usuario sobre el punto donde se encuentra [7].

1.1.2 Marcador

Un marcador es todo elemento visual que será reconocido por la aplicación de RA y ayudará a disparar la acción correspondiente a éste, como lo puede ser colocar un objeto 3D en la escena, agregar una imagen, reproducir un sonido o video. Estos marcadores se dividen en diferentes tipos como los códigos QR (*Quick Response Code* en inglés) [8], imágenes con más trabajo visual como lo es el color, contraste; también es posible utilizar objetos del mundo real como marcadores para las aplicaciones de RA.

1.1.3 Markless

Tecnología utilizada en las aplicaciones de RA para dotar a las aplicaciones de reconocimiento de patrones [5], [8] para así disparar un evento dentro de la aplicación y de esta manera no usar la tecnología de marcadores tradicional, para hacer uso del *markless* se utilizan diversas técnicas de visión por computadora que

permiten analizar las imágenes capturadas mediante una cámara conectada a un dispositivo inteligente.

Haciendo uso del *markless* en las aplicaciones se les dota de una nueva perspectiva, ya que estas no están limitadas al uso de marcadores que tengan que seguir algún patrón específico para su detección y rastreo dentro de la aplicación, ya que se le da la opción al usuario de utilizar imágenes que tenga a su disposición o elementos que se encuentren en su entorno y san capturados por el dispositivo que esté empleando en ese momento.

1.1.4 NFC

Near Field Communication (NFC por sus siglas en inglés) es una tecnología de comunicación inalámbrica de corto alcance basada en radiofrecuencia que trabaja en la frecuencia base de 13.56 MHz [9], que permite la transmisión de datos entre dispositivos; entre los usos que se le dan a esta tecnología está el uso en transacciones monetarias, la compartición de tarjetas de presentación y sistemas de acceso. La tecnología NFC ha cobrado auge en años recientes con el uso de los smartphones que la utilizan para diversos fines, ya sea como medio de transferencia de archivos, llaves de acceso a diversos sitios y como medio de pago electrónico.

1.1.5 GPS

GPS (*Global Positioning System* por sus siglas en inglés) es un sistema de radionavegación desarrollado por los Estados Unidos de América, proporciona servicios de posicionamiento, navegación y cronometría, el sistema se compone por tres elementos los cuales son: satélites en órbita, estaciones terrestres de seguimiento y control y por último el receptor GPS en manos del usuario [10].

1.1.6 API

Una API (*Application Programming Interface* por sus siglas en inglés) es un conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar software, esto simplifica el desarrollo aplicaciones permitiendo ahorrar tiempo y dinero. Las API otorgan flexibilidad; simplifican el diseño y el uso de aplicaciones [11], el api también se pueden presentar en forma de SDK (*Software Development Kit* por sus siglas en inglés).

1.1.7 ARToolKitX

Es un SDK que consiste de bibliotecas y utilerías que ayudan a los desarrolladores implementar aplicaciones de RA y realidad mixta, el SDK incluye aplicaciones de ejemplo para demostrar las capacidades de ARToolKitX [12], la última versión desarrollada es 1.0.6.1 a la fecha de escribir este documento.

1.1.8 ARCore

ARCore de Google provee un SDK para los ambientes de desarrollo más populares, este provee una serie de API's para las partes esenciales de la realidad aumentada como lo es el rastreo de movimiento, entendimiento del ambiente y estimación de luz; con todo lo ya mencionado es posible crear aplicaciones de RA para los dispositivos más populares en el mercado [13].

1.1.9 Android

Es un sistema operativo para dispositivos móviles, basado en Linux que permite la creación de aplicaciones de uso general y juegos, este sistema operativo móvil está siendo mantenido por Google siendo la versión 10 la más actual al momento de estar redactando este documento, las aplicaciones que se realizan para *Android* se empaquetan en un archivo con extensión .apk, estos archivos contienen los contenidos de la aplicación y es el archivo que se utiliza para instalar dichas aplicaciones [14].

Cada aplicación de *Android* está protegida por las siguientes características:

- El sistema le asigna a cada aplicación un usuario de Linux único.
- Cada proceso se ejecuta en su propia máquina virtual.
- Cada aplicación ejecuta su propio proceso de Linux, el sistema se encarga de gestionar dicho proceso.

Android SDK permite escribir las aplicaciones haciendo uso del lenguaje C y C++, así como también en Kotlin (lenguaje de código abierto) y el uso de la tecnología JAVA, y una vez que se compilan el código fuente de la aplicación se empaqueta como se mencionó previamente y se procede a la instalación en el dispositivo objetivo.

1.1.10 Lenguaje de desarrollo web

Un lenguaje de programación se define como un lenguaje entendible por los humanos que se utiliza para darle instrucciones a una computadora, dicho de esta manera un lenguaje de programación web es un lenguaje que se utiliza para darle instrucciones a una computadora dentro del ambiente web.

1.1.10.1 PHP

PHP (acrónimo recursivo de PHP: *Hypertext Preprocessor*) es un lenguaje de código abierto desarrollado en el año de 1995 por Ramus Lerford, su versión más reciente a la fecha de escritura de este documento es la 7.4.6 este lenguaje de programación es adecuado para el desarrollo web ya que este permite el diseño y construcción de sitios web dinámicos, entre las ventajas que este lenguaje presenta esta que puede ser incrustado en el código HTML de una página web y ser interpretado sin problemas por la misma, los códigos PHP son ejecutados en el servidor web generando HTML y enviándolo al cliente, entre las características que ofrece este lenguaje es su facilidad de uso para usuarios principiantes y características mucho más avanzadas para programadores profesionales[15].

1.1.10.1 Laravel

Framework de código abierto para el desarrollo de aplicaciones web con lenguaje PHP, fue creado en el año 2011, este *framework* (marco de trabajo en español) hace énfasis en la calidad del código generado, la facilidad de mantenimiento y la escalabilidad de los proyectos, con esto último permite su uso en proyecto de gran tamaño o de un tamaño reducido. La comunidad de desarrolladores de Laravel se mantiene en constante desarrollo de mejoras y complementos para este, haciendo que sea cada vez más versátil para el desarrollo web con PHP [16].

1.2 Planteamiento del problema

La aplicación ARAPP *Builder* tiene la funcionalidad de construcción de aplicaciones de RA para usuarios finales, dicha aplicación está realizada de una manera modular y consiste en tres capas que se son la capa de vista que es la parte con la cual interactúa el usuario, el controlador que se encarga de manejar las acciones que le soliciten al sistema y la última capa que es la capa de modelo esta capa es la encarga de interactuar con los datos

El módulo de captura del sistema de generación de aplicaciones de realidad aumentada tiene una gran importancia, ya que está encargado de la detección y desencadenamiento de los procesos por aumentar, aunque el módulo es plenamente funcional permite mejorarse realizando una reingeniería para incorporar nuevas tecnologías eliminando la limitación del uso de marcadores visuales.

Tomando en cuenta lo anterior, se pretende reconstruir el módulo de captura del sistema para integrar nuevas tecnologías como lo es el uso del GPS, *markless* y NFC, haciendo así que el usuario final tenga una gama de opciones al momento de realizar su aplicación de RA.

1.3 Objetivo general y específicos

En esta sección se presentan el objetivo general y los específicos del proyecto, los cuales son metas bien definidas para la realización de este.

1.3.1 Objetivo general

Incrementar la funcionalidad del sistema generador de aplicaciones de realidad aumentada, agregando tecnologías NFC, *Markless* y GPS como diversos marcadores.

1.3.2 Objetivos específicos

- Analizar el estado del arte de los diferentes tipos de marcadores de RA.
- Reconocer y detallar las condiciones que debe tener el módulo de captura del sistema generador de aplicaciones de RA
- Investigar las diferentes tecnologías que se puedan usar como mejora de marcadores de RA.
- Desarrollar el módulo de captura, dependiendo de la plataforma y su caracterización.
- Probar los casos de estudio definidos

1.4 Justificación

La aplicación ARAPP Builder ha demostrado ser efectiva para que los usuarios finales hagan sus propias aplicaciones de RA sin necesidad de recurrir a un programador, ARAPP Builder está conceptualizado para que sea escalable con nuevas características que se agreguen conforme avance el tiempo, así estas mejoras solo se deben aplicar a la base del sistema para evitar rehacer la aplicación entera nuevamente.

La tecnología relacionada con la RA ha evolucionado bastante en los últimos años, en gran medida gracias a la reducción de costos de los equipos y al aumento de procesamiento, como es el caso de los smartphones que actualmente son relativamente baratos y con la misma capacidad de procesamiento de muchos equipos de cómputo portátil.

En el campo de los smartphones se tiene una gran cantidad de tecnologías combinadas como lo son la comunicación móvil, el GPS (*Global Positioning System* por sus siglas en inglés), cámaras de video con buena resolución y la inclusión de NFC (*Near Field Communication* por sus siglas en inglés), por mencionar algunos. La combinación de estas y otras tecnologías en un dispositivo móvil lo dota de la posibilidad de uso combinado con aplicaciones de RA. Dicha combinación permite ampliar el uso de la RA haciéndola más agradable y útil para el usuario promedio.

El sistema de ARAPP *Builder* genera aplicaciones que se dividen en tres secciones de la cuales el presente trabajo se centrará en el módulo de captura, ya que este

módulo tiene la función de detectar los marcadores que desencadenan los elementos a mostrar al usuario debido a esta característica del módulo se ha decidido que este es el elemento por modificar para añadir funcionalidades utilizando diferentes tecnologías. Entre las mejoras que se pueden realizar al sistema está el mejoramiento del módulo de captura, ya que este módulo cumple una función crítica dentro de la aplicación base, debido a que este módulo es el encargado de la detección y desencadenamiento de los elementos a aumentar mediante las aplicaciones; entre las mejoras que se pueden implementar en este módulo se encuentran la adición del uso de *markless* (sin marcadores), mejorar la implementación de NFC con la que ya cuenta la aplicación, y por último, la utilización del GPS para desarrollar cercas geográficas.

Todas estas tecnologías se planean integrar en la medida que cada sistema operativo de smartphone lo permita, ya que cada uno de los dos grandes exponentes en el mercado (IOS y *Android*) tienen sus respectivas restricciones.

Capítulo 2 Capítulo 2. Estado de la práctica

Para ofrecer un panorama más amplio sobre el estado de las aplicaciones de RA y sobre las aplicaciones que estas tienen se analizaron varios artículos de los cuales se realizó un análisis para extraer la información importante y mostrarla en esta sección.

A continuación, se presentan los artículos más significativos que se revisaron.

2.1 Trabajos Relacionados

López, D et al. inician en [17] con una comparativa entre la RA y RV(Realidad Virtual), mostrando cómo cada una es diferente, dentro del documento se describe la situación actual de los sistemas de autor y sobre la búsqueda realizada para encontrar artículos que hablen sobre estos, la RA y la RA en dispositivos móviles; una vez obtenido el estado del arte procede a describir la solución propuesta que da origen a ARAPP Builder.

La solución propuesta es un sistema generador de aplicaciones de RA que hace uso del patrón de diseño MVC(Modelo-Vista-Controlador) y haciendo uso del marco de trabajo JSF y PrimeFaces para la interfaz de usuario; como API de RA se utilizó ARToolKit, una vez definas las herramientas y marcos de trabajo se procedió a la realización de un prototipo para comprobar la correcta integración de las tecnologías empleadas.

En [2], mencionaron la creación de una aplicación de RA adaptativa para mejorar la eficiencia de las tareas de mantenimiento industrial, para realizar esta mejora se ha desarrollado la aplicación ARAUM (Augmented Reality Authoring for Maintenance en inglés), la cual permite crear su propio contenido de RA.

Para la creación del contenido de RA de la aplicación se utilizó el framework Vuforia para hacer uso de la función clave de point (*markless*) tracking para detectar el objeto real y así sobreponer sobre el modelo 3D generado en la aplicación. Para la parte de experimentación y pruebas de la aplicación se utilizó un grupo de personas con experiencias distintas en el uso de aplicaciones de RA.

Los resultados arrojados por estas pruebas y experimentos demostraron que los usuarios que utilizaron la aplicación pueden crear contenido de RA para la misma de manera más rápida y que el uso de este mismo contenido generado ayuda a reducir el tiempo de mantenimiento ya sea que se tenga experiencia en esta área o no.

Como se describió en [3], la utilización de aplicaciones de RA en el sector automovilístico tiene grandes usos ya sea en casos donde el usuario está dentro

del vehículo o fuera de este, ya que estas aplicaciones pueden estar en smartphones, tabletas o en visualizadores montados en la cabeza (head-mounted displays en inglés). Entre las acciones que se pueden listar que se benefician de dicha aplicación de la RA están el entrenamiento mecánico, soporte técnico y la ayuda al usuario a identificar componentes del tablero. Dentro de los desafíos que presenta el estar realizando las aplicaciones de RA está el hecho de que no pueden emplear marcadores como tal para hacer uso de estos en las aplicaciones ya que pueden afectar la estética o el desempeño del automóvil (esto último cuando se necesita inspeccionar el motor con la aplicación). Para dar solución a dicho problema se propuso el desarrollo de una aplicación que utiliza el reconocimiento de patrones y así hacer caso omiso de los marcadores estándar.

La aplicación desarrollada se probó en el Volkswagen Tracking Challenge que desde el año 2014 forma parte de la (ISMAR) “International Symposium on Mixed and Augmented Reality”.

Kim, H.S et al. expuso en [4] que las peticiones por diseños en 3D se han incrementado en los proyectos de construcción actuales debido a que la forma de los edificios se ha hecho más elaboradas con el tiempo, aun así, varios ingenieros de campo siguen prefiriendo los dibujos en 2D, la principal razón de esto es que la información en el sitio puede ser diferente a la mostrada por el modelo 3D.

Aunque ya se han hecho progresos en la representación de los modelos en un ambiente puramente de RV (Realidad Virtual) esta representación no muestra las condiciones del mundo real, en cambio una representación utilizando RA muestra objetos generados por computadora en el mundo real, lo que mejora el entendimiento de las condiciones de trabajo. Haciendo uso de la tecnología de marcadores los ingenieros logran observar información detallada de los modelos al momento de inspeccionar los dibujos 2D con la ayuda de una aplicación de RA.

W. Fang et. al. describe en [18] que son las WAR (Wearable Augmented Reality en inglés, realidad aumentada usable) y sobre cómo estas se utilizan para visualizar elementos virtuales en tiempo real utilizando dispositivos móviles, todo esto gracias a que las WAR proveen de una interfaz humano-computadora.

Dentro de este artículo se discute cómo en tiempos recientes el uso de las WAR se ha visto en aumento gracias al decremento del costo de la visión por computadora y cómo esta reducción en costos que el uso de markless sea muy atractivo para emplearse en las WAR.

En [19] realizaron un estudio de la aplicación de la RA para incrementar la experiencia del usuario al momento de estar de estar viendo televisión, para realizar dicho realizado en conjunto por la British Broadcasting Corporation Research & Development department (BBC R&D) y la University of Manchester School of Computer Science, se realizó un experimento en donde se evaluaba la viabilidad de

aplicación de la tecnología de RA en un programa documental de la BBC para dicho experimento se empleó una televisión y un equipo de RA Microsoft HoloLens.

El contenido de RA era observado en por los participantes del experimento de tal manera que se veían dos pantallas al lado de la televisión mostrando información y retroalimentación al usuario, así como también la figura de una persona que observara fijamente al usuario cuando este la observaba, los resultados del estudio fueron satisfactorios mostrando diversos aspectos sobre este, como el hecho de que varios usuarios pausaron la reproducción del video para observar el contenido de RA, otros no observaron contenidos debido a estar prestando atención a la grabación en la televisión o por estar viendo algún otro elemento.

Por la parte del trabajo a futuro mencionan que se debe de investigar qué tipo de efecto que tiene cada tipo de elemento de RA en la experiencia del usuario, experimentar con diferentes géneros de programas de televisión para entender como la RA afecta la experiencia del usuario, por último, mencionan que, aunque la combinación de la RA con la televisión puede dar nuevas experiencias aún hay riesgos y problemas que se deben de investigar más a fondo.

P. Verma et al. abordan en [20] la creación de una aplicación para dispositivo móvil que facilite la navegación en interiores, ya que las personas encuentran el interior de los edificios confusos debido a la arquitectura que estos tienen y a la falta de puntos de fácil reconocimiento por parte de los usuarios, para dar una solución a esta problemática se hace uso de teléfonos inteligentes los cuales pueden mostrar al usuario con información sobre el complejo donde se encuentre.

La experiencia del usuario es importante para los sistemas de navegación en interiores y la RA es una tecnología que se puede utilizar para emplear una experiencia más conveniente y atractiva para los usuarios, ya que con el uso de la RA las instrucciones para llegar a un determinado lugar pueden ser mostradas en tiempo real en la pantalla de un dispositivo.

Para dar solución a la problemática expuesta en este artículo se presenta la creación de una aplicación para dispositivo móvil que muestra en tiempo real las instrucciones para llegar a determinados lugares dentro de un edificio, esta aplicación se desarrolló para dispositivos móviles de la marca Apple usando ARKit como motor de RA y Unity para convertir los dibujos 2D del edificio a un modelo 3D. El funcionamiento de la aplicación es que la ubicación del usuario se registra en tiempo real e intercambiar información con un servidor utilizando la red de internet inalámbrica del edificio, la información que el servidor le devuelve al dispositivo son las instrucciones para llegar al lugar que el usuario selecciono.

Por último, se mencionan los resultados obtenidos del trabajo de investigación los cuales arrojan resultados favorables al momento de convertir el mapa 2D una escena 3D, por la parte de la aplicación muestra que las flechas de navegación se muestran de manera correcta y permiten una interacción con el usuario correcta.

En [21] se propone la combinación de la RA y la TAR (*Tangible Augmented Reality* por sus siglas en inglés) para el entrenamiento de los usuarios en la tarea de ensamble manual, se propone la combinación de estas tecnologías ya que estas le dan una mejor retroalimentación al usuario, para comprobar la viabilidad de esta combinación se realizó un estudio donde aún grupo de personas de pruebas se les dio la tarea de ensamblar de manera parcial una *motherborad* (tarjeta madre).

El grupo estaba conformado por diez personas con experiencia en el manejo de la RA y quince personas sin experiencia alguna con la RA, el experimento fue realizado con un dispositivo móvil Google Pixel XL con VR Box, Unity y Vuforia, para la prueba se utilizaron piezas de Ilustración echas de cartón que fueron revestidas con una imagen para que sobre de estas se sobrepusiera la imagen de la pieza a emplear. Una vez que el usuario agarraba una pieza se le daban ciertas instrucciones como girar la pieza para familiarizarse con ella y por último colocar la pieza en correspondiente lugar dentro de la *motherboard*.

Los resultados del experimento demostraron que mediante el uso de esta técnica los usuarios sienten la manipulación de los objetos virtuales se siente natural, intuitiva y fácil, así también arrojo resultado que muestran que ciertos usuarios se sintieron desconcertados por el uso de la tecnología ya que esta les mostraba una ubicación diferente de dónde colocar el objeto debido al ángulo de visión que tenían.

En [22] mencionan que con la proliferación de tiendas en línea y proyecto de hágalo usted mismo los productos son generalmente divididos en partes pequeñas para su envió, permitiendo así que los usuarios puedan armar los productos por ellos mismos, siguiendo las instrucciones que vienen en manuales impresos que no ofrecen ninguna retroalimentación sobre el ensamble del objeto.

Para lograr este objetivo se realizó una técnica que permitiera detectar la forma de objetos y determinar su posición y mediante el empleo de algoritmos mostrar las instrucciones para el ensamble que son mostradas en una transmisión de video que el usuario observa en tiempo real, para probar este desarrollo se realizo un experimento con sujetos de prueba donde se les pidió que armaran un objeto usando las instrucciones que les daba la aplicación. Los resultados de este experimento demostraron que a los usuarios les fue de agrado el sistema por que les ayudo al ensamble del objeto de prueba.

La aplicación de esta nueva técnica de rastreo de markless realizada puede ser mejorada haciendo uso de la computación paralela lo cual mejoraría los tiempos de respuesta, así también se recomienda hacer un análisis sobre los pros y contras de utilizar esta nueva técnica. Los autores de este articulo piensan que la técnica desarrollada es aplicable a aplicaciones que hagan uso de un dispositivo visualizador como es el caso de oculus.

Viyanon et al. describen en [23] el desarrollo de una aplicación para dispositivos móviles que pretende ayudar a los usuarios al momento de escoger la decoración

o muebles de su hogar, ya que los usuarios al momento de estar escogiendo estos tienen dudas como si el mueble combinara con los demás muebles que ya tiene o si este entrara en la habitación donde lo planea poner.

Para dar soluciona esta problemática se desarrolló la aplicación “AR Funiture” la cual utiliza marcadores para mostrar información en pantalla sobre los objetos que el usuario quiere conocer y un uso *markless* para colocar modelos 3D a escala real de estos objetos sobre la habitación donde el usuario planea poner estos, para el desarrollo de esta aplicación se empleó el *framework* Kudan SDK como motor de RA. Para evaluar la usabilidad de la aplicación se usó un cuestionario donde se evaluaba la aplicación como tal, el resultado de este cuestionario demostró que la aplicación tiene un uso aceptable por parte de los usuarios.

Así también como parte de los resultados obtenidos de los experimentos realizados con la aplicación demostró que la aplicación puede ser mejorada en su rendimiento general, que esta detecte las paredes y los techos de manera correcta para mostrar los modelos 3D de manera correcta.

2.2 Análisis Comparativo

A continuación, se muestra un análisis comparativo de los artículos mostrados en la sección 2.1, los resultados del análisis se muestran en la tabla 2.1, 2.2 y 2.3.

Articulo	Problema	Objetivo	Resultado	Estado	Tecnología
[2]	El tiempo que tardan los empleados encargados de mantenimiento es excesivo	Mejorar el tiempo que tarda un operario en dar mantenimiento	Creación de aplicación ARAUM para mejorar el mantenimiento	Terminado	Vuforia framework
[3]	Dentro de los desafíos que presenta el estar realizando las aplicaciones de RA está el hecho de que no pueden emplear marcadores como tal para hacer uso de estos en las aplicaciones, ya que afectan la estética o el desempeño del automóvil.	Implementación de markless para realizar actividades de RA en automóviles.	Los resultados de las pruebas de la aplicación desarrollada fueron satisfactorios y expuestos en el ISMAR.	Terminado.	C++ OpenCV, Point Cloud Library (PCL), OpenNI, Microsoft Kinect SDK and Developer Toolkit and videoInput

Tabla 2.1 Análisis comparativo de artículos relacionados con la RA.

Articulo	Problema	Objetivo	Resultado	Estado	Tecnología
[4]	Incremento en los diseños de edificios utilizando tecnología 3D, los ingenieros siguen prefiriendo los dibujos 2D.	Desarrollar una aplicación de RA que muestre los modelos 3D de los edificios sobre los dibujos 2D de los ingenieros para mostrar información detallada sobre estos	Haciendo uso de marcadores en los dibujos los modelos se muestran mostrando información detallada, así como una vista de la construcción terminada.	Terminado.	No indica.
[17]	No existe un generador de aplicaciones de RA que le permita al usuario crear sus propias aplicaciones.	Creación de un sistema de autor para la creación de aplicaciones de RA.	Creación de ARAPPBuilder	Terminado.	ARToolKit Primefaces JavaServerFaces
[18]	Propuesta de un dispositivo de RA utilizable que haga uso de multisensores y tenga seis grados de libertad	Desarrollar un sensor de RA usable que utilice seis grados de libertad	Se propone la creación de un multisensor para la RA usable	Terminado	Cámara Monocular Unidad de medición Inercial Sensor de profundidad SAMSUNG S6
[19]	Falta de un estudio que demuestre los resultados de la combinación de la tecnología de RA con la acción de mirar programas televisivos	Investigar la combinación de la RA con la acción de observar programas de televisión.	Estudio sobre la aceptación de la combinación de ambas tecnologías.	Terminado.	Microsoft HoloLens televisión
[20]	La navegación en interiores presenta dificultades para los usuarios debido a lo complejo y confuso del diseño de los edificios	Desarrollar una aplicación que permita mejorar la navegación en interiores	Desarrollo de Indoar, aplicación para navegación en interiores	Terminado	ARKit Unity XCode Three.js Firebase

Tabla 2.2 Análisis comparativo de artículos relacionados con la RA.

Articulo	Problema	Objetivo	Resultado	Estado	Tecnología
[21]	Falta de estudio sobre la combinación de la RA y la TAR	Crear un estudio que demuestre los beneficios de la combinación de la RA y la TAR	Estudio que muestra los resultados de aplicar la combinación de la RA y TAR en un ensamble manual	Terminado	Google Pixel XL VR Box Unity Vuforia framework
[22]	El ensamble de productos se hace siguiendo manuales impresos que no ofrecen ayuda alguna de este tipo de medio	Desarrollar una aplicación que ayude con el ensamble de objetos usando *markless*	Desarrollo de una técnica que permite hace uso de *markless* para detectar objetos y permitir su emsamble	Trabajo a futuro	PC con un procesador quad-core a 3.4 GHz y 12Gb de ram ASUS Xtion Pro Live
[23]	Los usuarios no saben si la decoración o muebles que quieren comprar combinaran con su decoración o si estos entraran en la habitación donde los quieren colocar	Crear una aplicación que permita visualizar muebles y decoraciones en habitaciones antes de la compra	Desarrollo de AR Furniture	Terminado	Unity Kudan SDK Android 4.4.2(kitkat)

Tabla 2.3 Análisis comparativo de artículos relacionados con la RA.

Una vez realizado el análisis de los artículos expuestos con anterioridad se demuestra que la RA y sobre como está siendo abordada por diferentes frentes, así como también la combinación que tiene con diversas tecnologías como el GPS y el uso de *markles*.

2.3 solución Propuesta

El sistema de autor ARAPP *Builder* hace uso de las tecnologías ARToolkit como motor de realidad aumentada para las aplicaciones desarrolladas, JSF como motor de la plataforma web, MySQL como gestor de base de datos de código abierto y licencia libre, el sistema se desarrolló utilizando el IDE NetBeans y el componente final es el uso de Gradlew para compilar la aplicación final.

Como ya se ha mencionado ARAPP *Builder* es una plataforma web, dicha plataforma hace uso del modelo de desarrollo MVC (modelo-vista-controlador) como se muestra en la figura 2.1

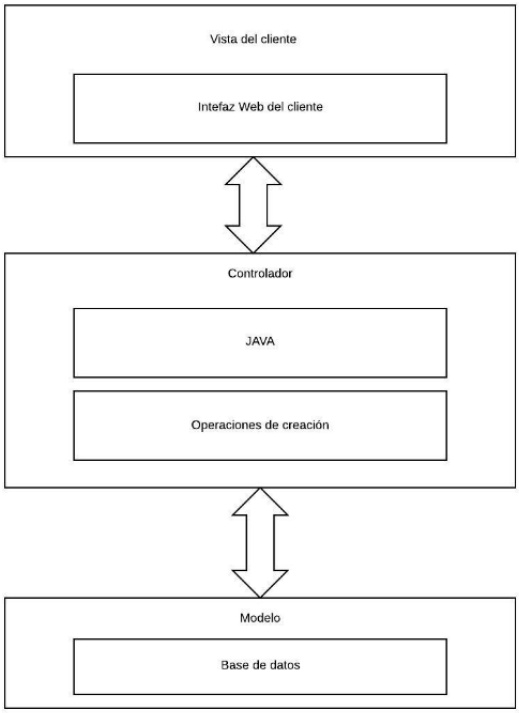

Figura 2.1 Modelo MVC de ARAPP Builder.

Debido a los cambios tecnológicos que han ocurrido desde la realización de ARAPP *Builder* se han realizado cambios a la plataforma y al proceso que esta utiliza para generar las aplicaciones, entre los cambios que se realizaron están el uso de ARCore como motor de RA para las aplicaciones y el desarrollo de la plataforma utilizando el lenguaje de programación PHP empleando el *framework* Laravel, dichos cambios han afectado en lo mínimo el modelo MVC de ARAPP *Builder*.

La plataforma web funciona dividendo sus componentes entre las tres capas del modelo MVC, la primera capa que es la vista interactúa el usuario haciendo sus operaciones normales, en la capa de control es donde el controlador de PHP interactúa con sus métodos para realizar las operaciones necesarias e interactuar con la tercera y última capa que es la capa del modelo, esta última interactúa con la base de datos del sistema.

Capítulo 3 Capítulo 3 Aplicación de la metodología

En el presente capítulo se detalla el proceso de la metodología de desarrollo ágil XP para dar seguimiento a la problemática expuesta en capítulos anteriores, dicha metodología ya mencionada se divide en cinco fases de las cuales cuatro se realizan de manera cíclica, estas fases iniciales son: planificación, diseño, codificación y pruebas; última fase del desarrollo es la liberación de la aplicación terminada.

La metodología de desarrollo XP es una de las llamadas metodologías de desarrollo ágil, esta metodología pone más énfasis en la adaptabilidad que en la prevención, gracias a esta característica permite adaptarse a los cambios que puedan surgir durante el desarrollo del proyecto, para dar solución a los cambios que surgen la metodología hace uso de un desarrollo cíclico en la que por cada iteración del desarrollo se incrementa la funcionalidad del sistema.

3.1 Planificación del proyecto

Como parte de la planificación del proyecto se presentan los diferentes recursos utilizados para apoyar al desarrollo, entre los elementos que se presentan se encuentran las historias de usuario, asignación de roles, planificación de lanzamientos y diagramas.

3.1.1 Historias de usuario

Las historias de usuario tienen la misma función que los casos de uso en otras metodologías y son una representación de un requerimiento del software descrito por el usuario utilizando lenguaje coloquial, como se puede apreciar en la tabla 3.1 que es la historia de usuario número 1.

Historia de usuario	
Número: 1	Nombre: Uso de *markless*
Prioridad cliente: Alta	Prioridad en desarrollo: Alta
descripción: Se necesita que la aplicación pueda hacer uso de imágenes, para ya no tener que hacer códigos especiales para la RA.	
Observaciones: • Cambio de la API de ARToolKit	Iteración asignada: • 1

Tabla 3.1 Historia de usuario 1.

Como se aprecia en la tabla 3.2 se muestra la historia de usuario número 2 que hace referencia al uso del GPS dentro de las aplicaciones de RA realizadas con la plataforma ARAPP *Builder*.

Historia de usuario	
Número: 2	Nombre: Uso de GPS.
Prioridad cliente: Alta	Prioridad en desarrollo: Alta
Descripción: Se quiere emplear el GPS para mostrar objetos en un lugar especifico	
Observaciones: • Se necesita investigar el uso de GPS. • Probar combinación de GPS y RA.	Iteración asignada: 2

Tabla 3.2 Historia de usuario 2.

En la tabla 3.3 se ilustra la historia de usuario número 3, en esta historia se aborda el tema relacionado con el uso de NFC dentro de las aplicaciones generadas con la plataforma web.

Historia de usuario	
número: 3	nombre: Uso de NFC
Prioridad cliente: Alta	prioridad en desarrollo: Alta
descripción: Se quiere emplear NFC para aumentar elementos usando etiquetas NFC.	
observaciones: • Se necesita investigar el uso de NFC. • Probar combinación de NFC y RA.	Iteración asignada: • 3

Tabla 3.3 Historia de usuario 3.

Como se ha presentado, estas son las historias de usuario que se han realizado para el desarrollo del proyecto siguiendo las fases de la metodología XP, cada historia presenta una característica que se ha de desarrollar para integrar en la plataforma.

3.1.2 Asignación de Roles

En la asignación de roles se definen las responsabilidades que cada miembro del equipo de trabajo cumplirá durante el desarrollo del sistema, dicha asignación se muestra en la tabla 3.4.

Rol	**Asignado**
Programador	Sergio Antonio Cornu Tinoco
Cliente	Ignacio López Martínez

Tester	Sergio Antonio Cornu Tinoco

Tabla 3.4 Asignación de roles del proyecto.

3.1.3 Release Planning (Planificación de Lanzamiento)

El siguiente paso de la metodología es realizar un calendario de trabajo donde se especifican las actividades a realizar durante el desarrollo del proyecto, este calendario se realiza con base en las historias de usuario creadas con anterioridad. Cada historia de usuario se realizará en una interacción del proyecto teniendo en cuenta si prioridad la representación gráfica del calendario se muestra en la tabla 3.5.

Historia	Interacción	Prioridad	Fecha inicio	Fecha fin
Historia 1	1	Alta	06/01/2020	31/01/2020
Historia 2	2	Alta	03/02/2020	28/02/2020
Historia 3	3	Alta	02/03/2020	03/04/2020

Tabla 3.5 Plan de lanzamiento del proyecto.

3.1.3 Diagrama de Clases

Como parte de la planeación del desarrollo del proyecto se realizó un diagrama de clases que muestra las clases del sistema y cómo se relacionan estas dentro del sistema esto se aprecia en la figura 3.1.

En la figura 3.1 se muestran las clases que conforman la plataforma web y como estas se relacionan entre sí, las clases mostradas en la figura trabajan a nivel de controlador según el modelo de desarrollo MVC, ya que estas solo acceden al modelo mediante el uso del ORM (*Object-Relational Mapping* por sus siglas en inglés) *Eloquent* que incluye el marco de trabajo Laravel.

La función de cada clase de la plataforma web es la siguiente "CrearAppAndroidController" es la clase principal con la cual el usuario interactúa ya que esta es la encarga de crear la aplicación y de redirigir al usuario a la página de edición, así también maneja la opción de compilación de la aplicación y las opciones de descarga. El resto de las clases de la plataforma web "AumentarTextoController", "AumentarVideoController", "ImagenAumentarControllder"y por último "Modelo3dController" son las clases encargadas del manejo de los contenidos de las aplicaciones de RA, estas clases evalúan la calidad del marcador a utilizar y realizan las operaciones necesarias para manejar los contenidos a aumentar.

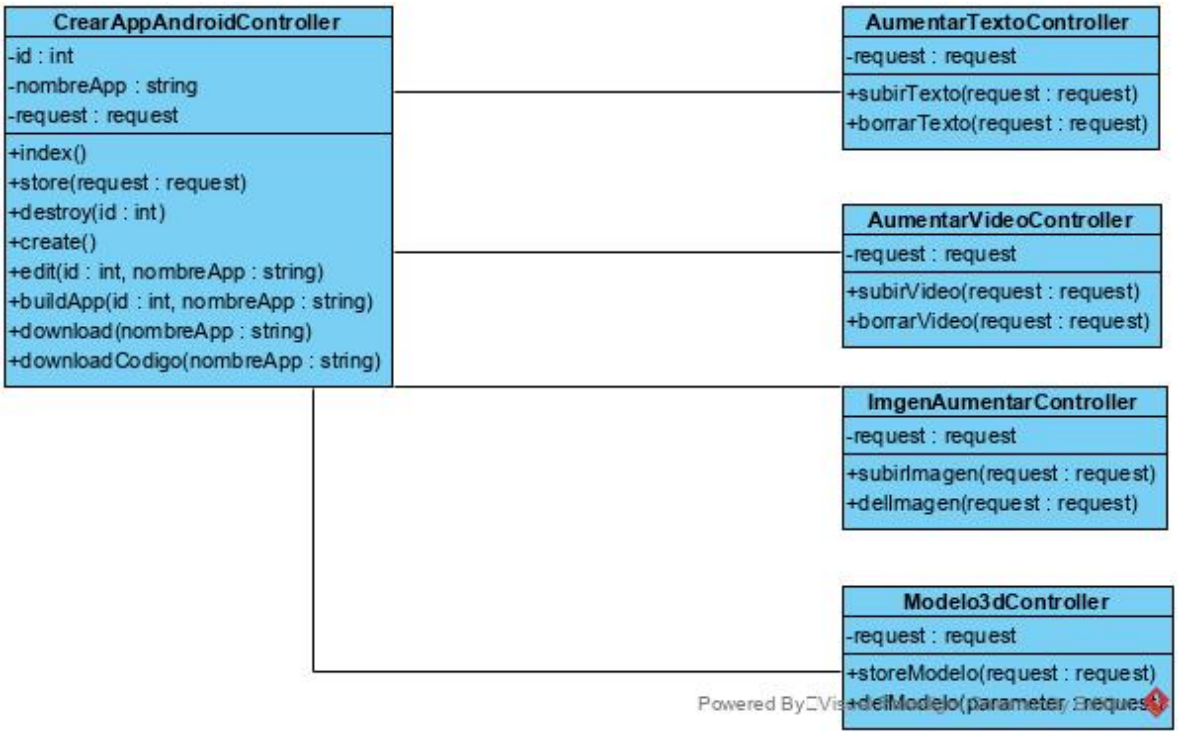

Figura 3.1 Diagrama de clases de ARAPP Builder.

Así también se realizó un diagrama de clases de la aplicación Android base y sobre cómo las clases de esta se relacionan entre sí, el diagrama de la interacción de estas clases *Android* se puede apreciar en la figura 3.2.

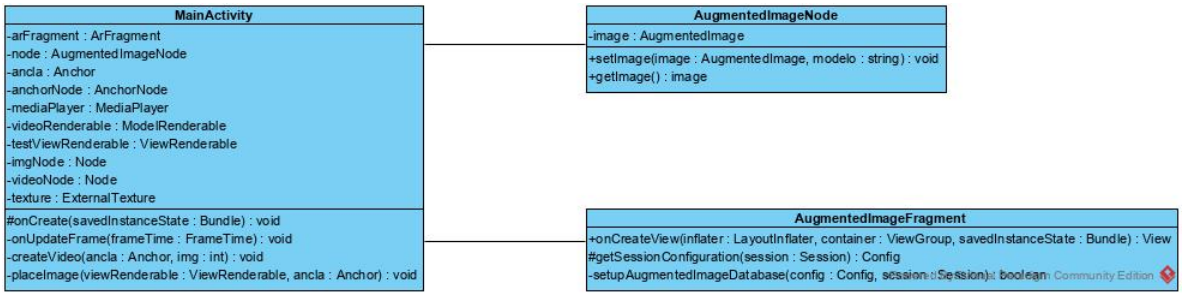

Figura 3.2 Diagrama de clases de la aplicación base.

3.1.4 Diagramas de secuencias

Los diagramas de secuencia tienen el objetivo de describir de una manera gráfica la interacción entre objetos de una aplicación a través del tiempo haciendo énfasis en la secuencia de los mensajes intercambiados por los objetos.

3.1.4.1 Diagrama de Secuencia de Creación de App

En la figura 3.3 se observa el diagrama de secuencia de creación de una aplicación de RA haciendo uso de la plataforma web.

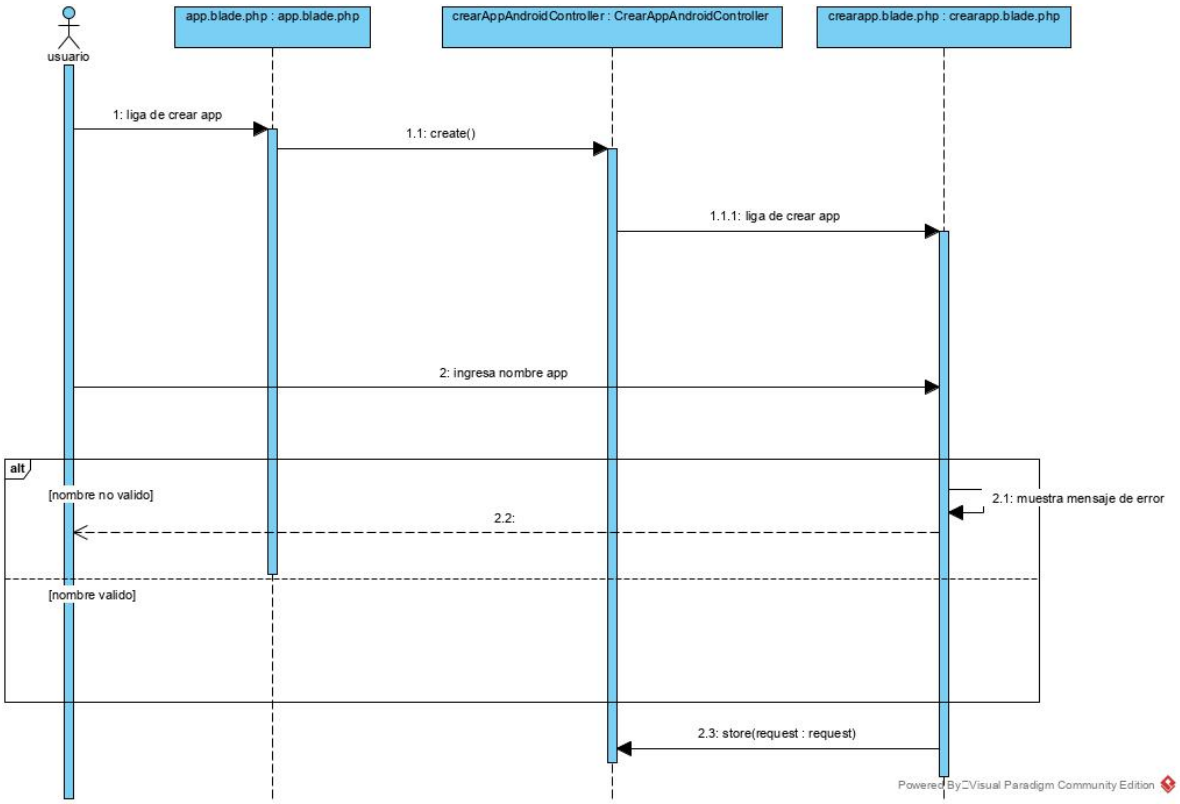

Figura 3.3 Diagrama de secuencia de creación de aplicación.

3.1.4.2 Diagrama de Secuencia de Modificación de Aplicación Agregar Modelo 3D

En la figura 3.4 se observa el diagrama de secuencia para agregar modelos 3D a las aplicaciones de RA.

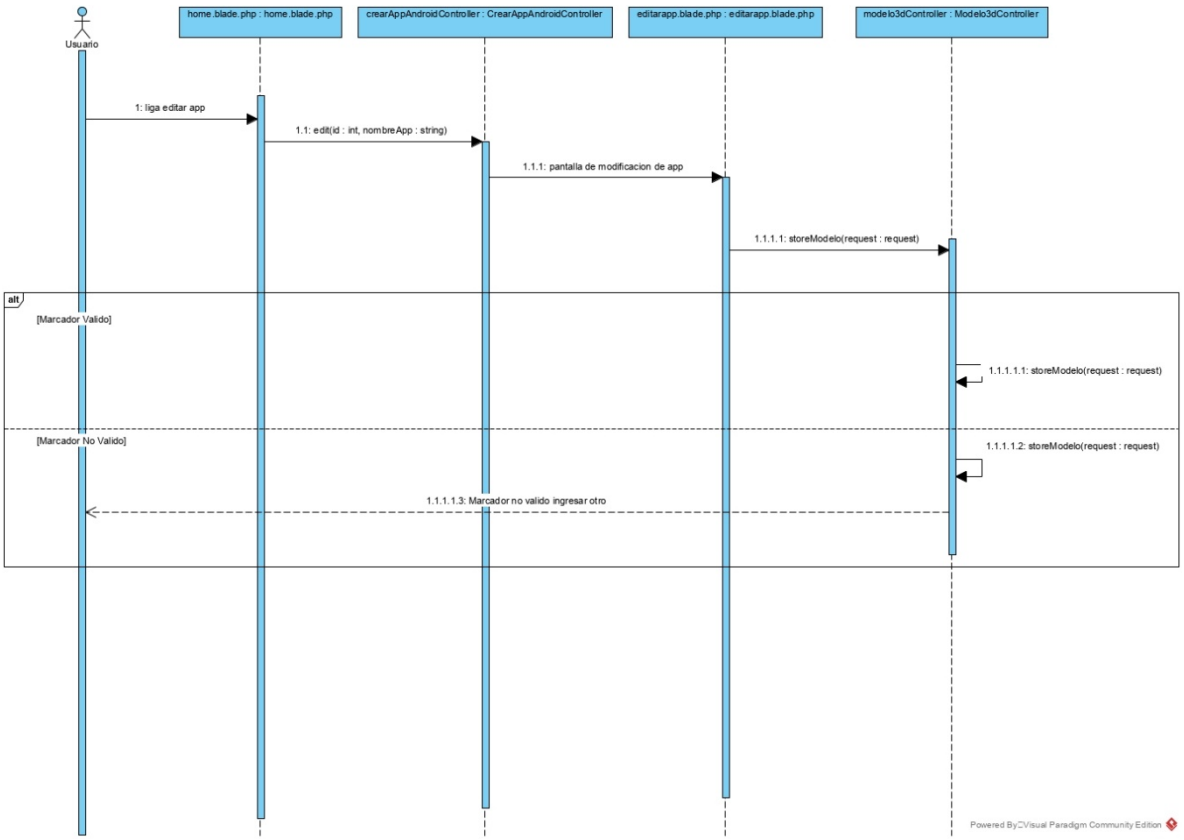

Figura 3.4 Diagrama de Modificación de Aplicación Agregar Modelo 3D.

3.1.4.3 Diagrama de Secuencia de Modificación de Aplicación Agregar Imagen

En la figura 3.5 se muestra el diagrama de creación de modificación de aplicación especifico al caso de agregar una imagen como contenido para aumentar dentro de una aplicación de RA.

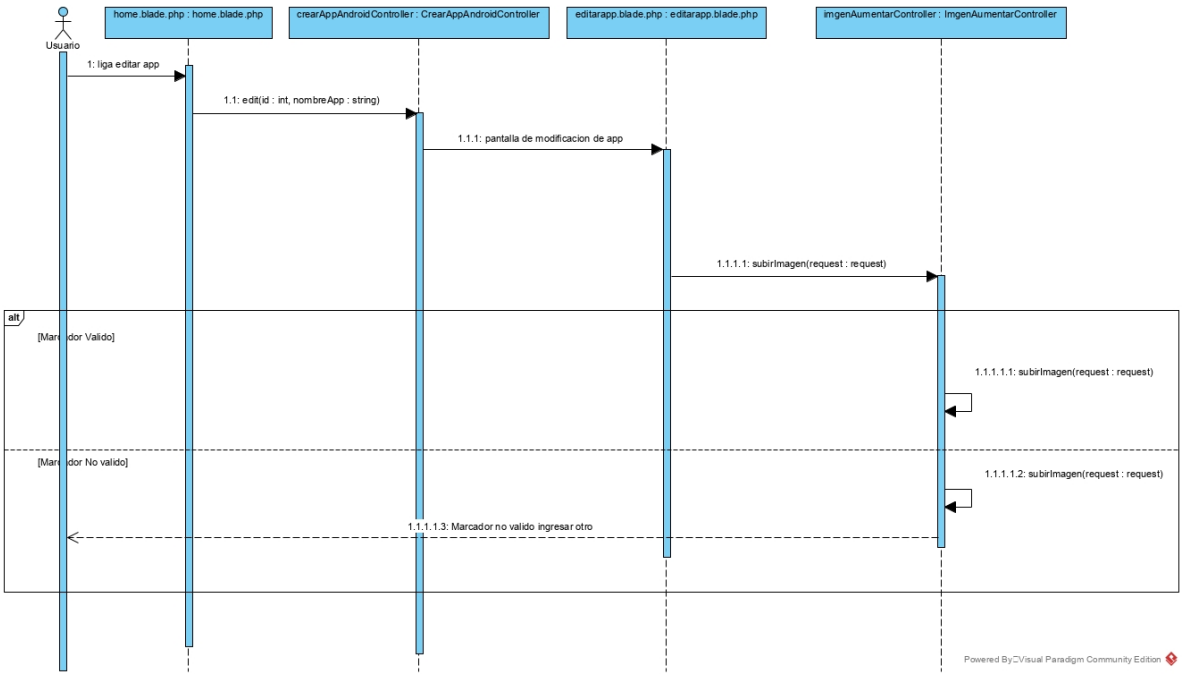

Figura 3.5 Diagrama de Secuencia de Modificación de Aplicación Agregar Imagen.

3.1.4.5 Diagrama de Secuencia de Modificación de Aplicación Agregar Texto

En la figura 3.6 se muestra el diagrama que muestra la secuencia correspondiente a la secuencia para agregar un texto para aumentar dentro de una aplicación de RA.

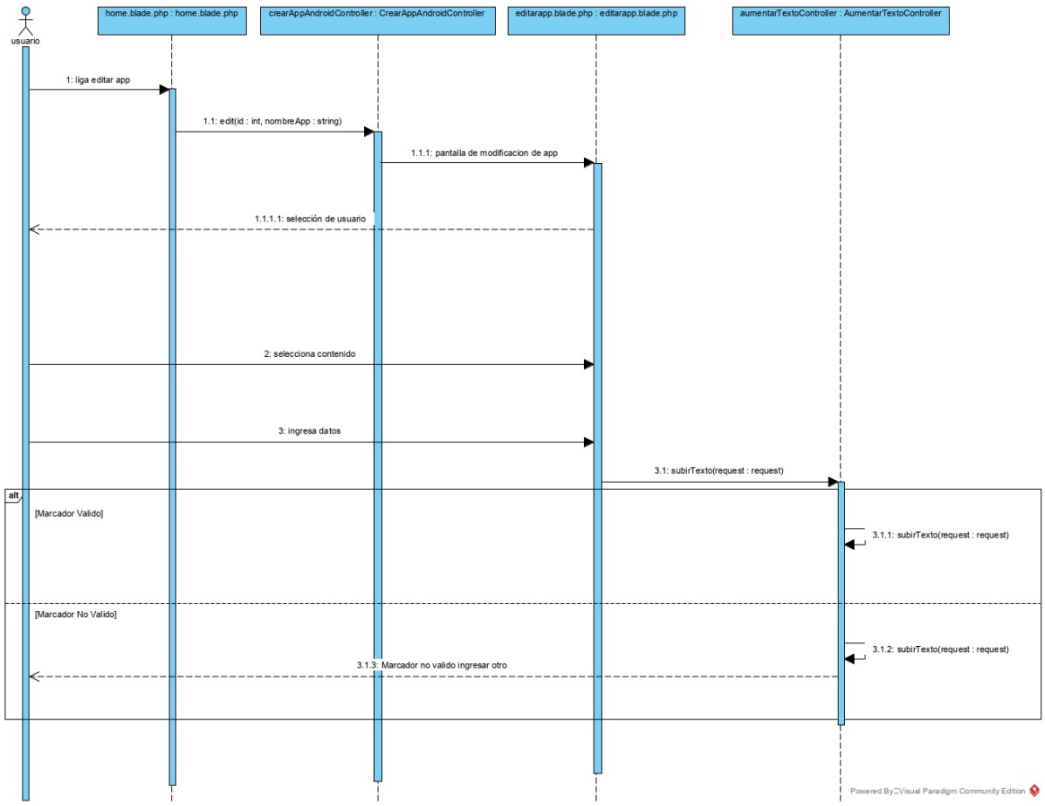

Figura 3.6 Diagrama de Secuencia de Modificación de Aplicación Agregar Texto

3.1.4.6 Diagrama de Secuencia de Modificación de Aplicación Agregar Video

Prosiguiendo con los diagramas de secuencia se muestra en la figura 3.7 el diagrama de secuencia para agregar un video a una aplicación de RA.

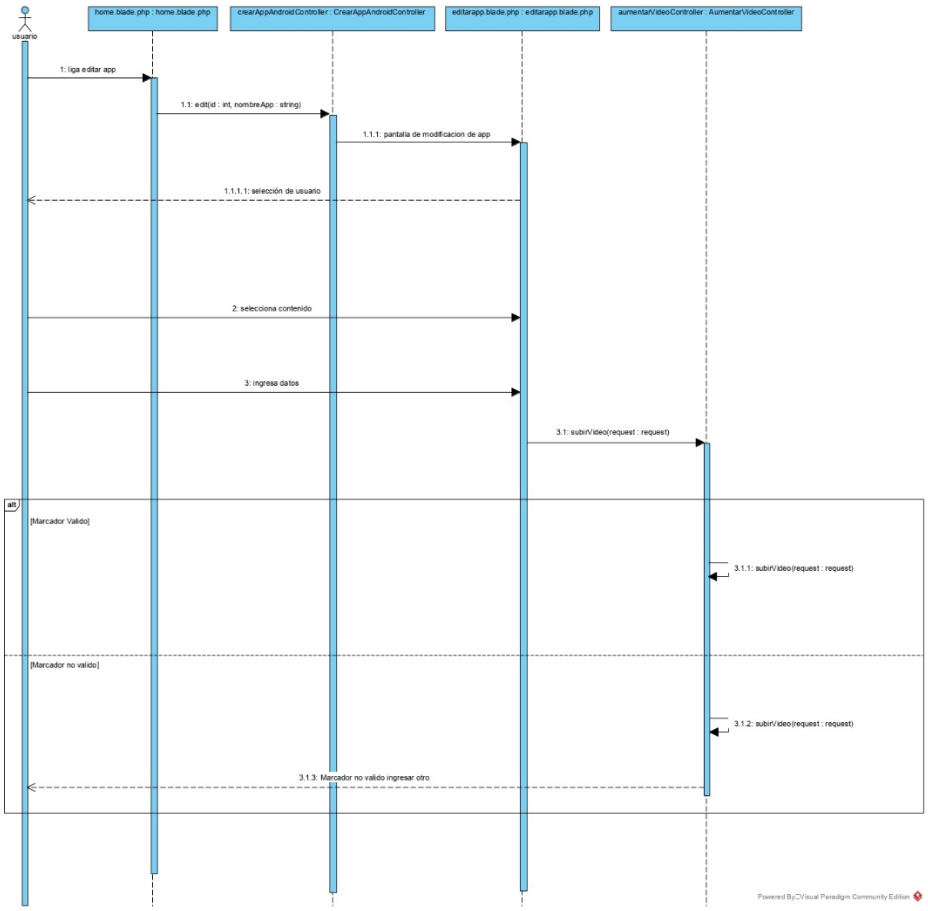

Figura 3.7 Diagrama de Secuencia de Modificación de Aplicación Agregar Video.

3.2 Diseño

En la segunda parte de la metodología XP se aborda el diseño del sistema de manera sencilla apoyándose en bocetos realizados por el cliente mostrando cómo espera que sea la aplicación al término de su desarrollo, estos diseños son lo más sencillos posibles para facilitar una buena interpretación y su posterior traslado a la aplicación.

3.2.1 Bocetos del cliente

En esta sección se observan los bocetos que realizó el cliente sobre cómo espera que la interfaz se vea en el navegador web de los usuarios, estos bocetos están realizados a mano por el mismo cliente y lo que este dibuja en ellos se representara en la plataforma o se indicara que no es posible realizar dicho diseño.

En la figura 3.8 se observa el boceto que realizó el cliente para la página de inicio de la plataforma web.

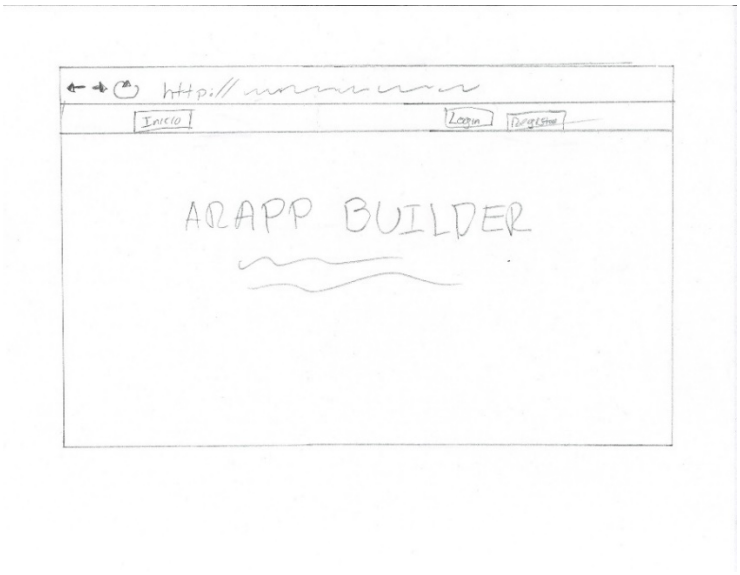

Figura 3.8 Boceto de pantalla principal de la plataforma web.

Como parte de los bocetos realizados en la figura 3.9 se observa el boceto realizado para el registro de nuevos usuarios dentro de la plataforma web.

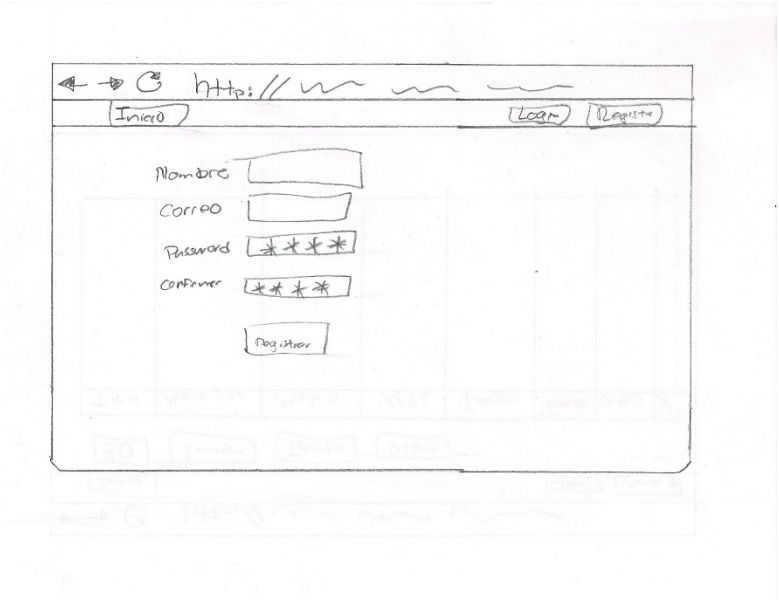

Figura 3.9 Boceto para el registro de nuevos usuarios en la plataforma.

En la figura 3.10 se observa el boceto para la página principal del usuario.

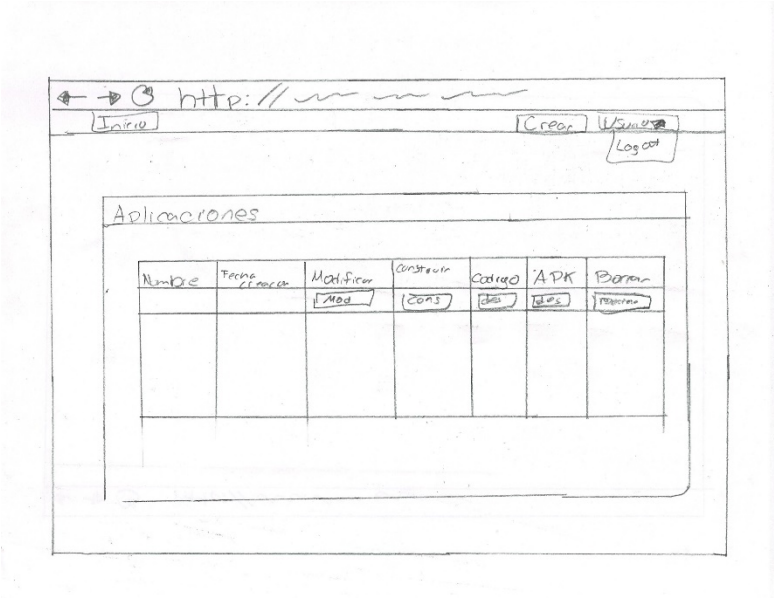

Figura 3.10 Boceto de la página principal del usuario.

En la figura 3.11 se observa el boceto creado para la creación de una nueva aplicación por parte del usuario.

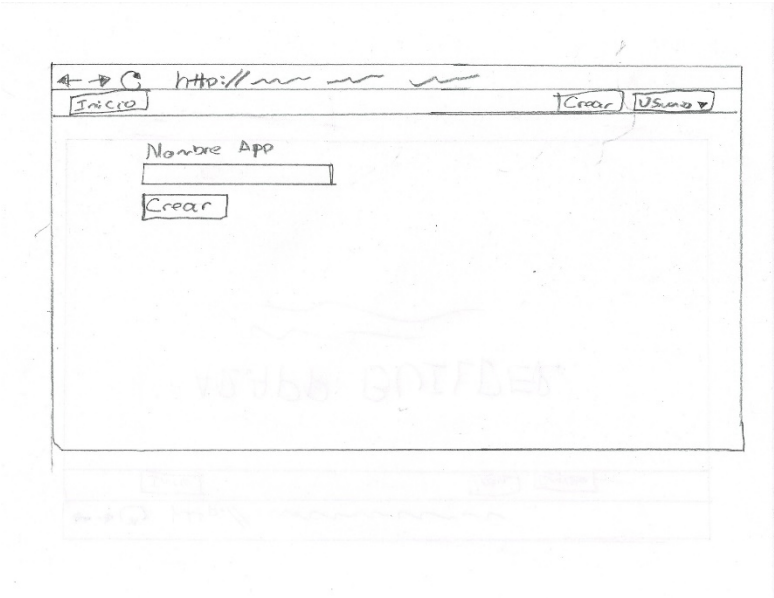

Figura 3.11 Boceto para la creación de una nueva aplicación.

En la figura 3.2.1.6 se aprecia el boceto realizado para la modificación de las aplicaciones creadas por el usuario usando la plataforma web.

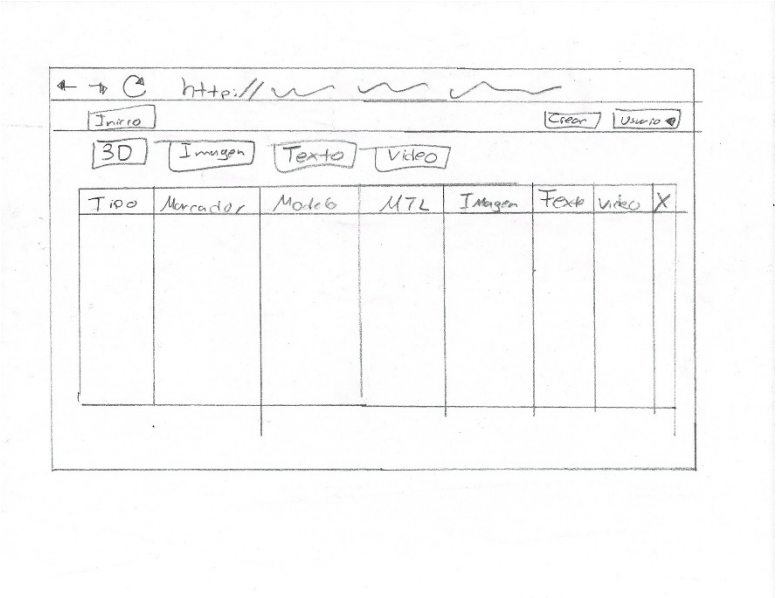

Figura 3.12 Boceto para la modificación de una aplicación.

3.2.2 Tarjetas CRC (Clase-Responsabilidad-Colaboración)

Las tarjetas CRC son una herramienta de desarrollo que ayuda a los programadores a centrarse en los elementos a desarrollar del sistema sin las distracciones que conlleva utilizar un enfoque tradicional para el desarrollo, por cada clase que se realice para el sistema se realiza una tarjeta CRC que describe sus funciones y las colaboraciones con otras clases que tenga.

En la tabla 3.6 se aprecian las responsabilidades de la clase de la plataforma web CrearAppAndroidController y las colaboraciones que esta tiene con otras clases dentro de la plataforma.

CrearAppAndroidController	
Responsabilidad	Colaboración
Crear Aplicación Dar nombre aplicación Borrar aplicación Descargar código Descargar apk	Aplicacion_androids Contenidos_apps Users

Tabla 3.6 Tarjeta CRC de CrearAppAndroidController.

La tabla 3.7 muestra las funciones de la clase AumentarTextoController y las colaboraciones que esta tiene con las demás clases de la plataforma.

AumentarTextoController	
Responsabilidad	Colaboración
Colocar elemento markless en el servidor Recibir texto a aumentar Borrar elemento markless y texto aumentado	Contenido_apps users

Tabla 3.7Tarjeta CRC de AumentarTextoController.

En la tabla 3.8 se muestran las responsabilidades de la clase AumentarVideoController de la plataforma web, como se aprecia en la tabla esta clase se encarga de procesar el contenido de tipo video que se utilizará en las aplicaciones de RA.

AumentarVideoController	
Responsabilidad	Colaboración
Colocar elemento markless en el servidor Recibir video a aumentar Convertir video a ".mp4" usando códec h264 Borrar elemento markless y video	Contenido_apps users

Tabla 3.8 Tarjeta CRC de AumentarVideoController.

En la tabla 3.9 se muestran las responsabilidades de la clase ImagenAumentarController, esta clase se encarga del manejo de las imágenes que se utilizarán como elemento a aumentar en las aplicaciones generadas.

ImagenAumentarController	
Responsabilidad	Colaboración
Colocar elemento markless en el servidor Recibir imagen a aumentar Borrar elemento markless e imagen aumentada	Contenido_apps users

Tabla 3.9 Tarjeta CRC de ImagenAumentarController.

Como se muestra en la tabla 3.10 se muestran las responsabilidades de la clase Modelo3DController que se encarga de procesar los modelos 3D que se mostrarán en las aplicaciones de RA generadas.

Modelo3DController	
Responsabilidad	Colaboración
Colocar elemento markless en el servidor Recibir modelo 3d Borrar modelo 3d y elemento markless	Contenido_apps users

Tabla 3.10 Tarjeta CRC de Modelo3DController.

3.3 Codificación

En esta sección del documento se presenta la codificación realizada durante el proyecto mostrando parte de los códigos realizados.

3.3.1 Desarrollo aplicación base *Android*

Para el desarrollo de la aplicación base Android que se emplea en el proceso de generación automática dentro de la aplicación web se utilizó el IDE de desarrollo *Android Studio* ya que este es el ambiente de desarrollo nativo para aplicaciones *Android*.

Para que las aplicaciones de *Android* hagan uso de la tecnología de RA es necesario modificar el archivo “AndroidManifest.xml” agregando las líneas <uses-permission android:name="android.permission.CAMERA" />, <uses-feature android:name="android.hardware.camera.ar" android:required="true"/> y <meta-data android:name="com.google.ar.core" android:value="required" /> como se muestra en la siguiente figura 3.13.

AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
        <manifest xmlns:android="http://schemas.android.com/apk/res/android"
            package="ito.depi.prueba">

            <uses-permission android:name="android.permission.CAMERA" />

            <uses-feature android:name="android.hardware.camera.ar" android:required="true"/>

            <uses-permission android:name="android.permission.INTERNET" />

            <application
                android:allowBackup="true"
                android:icon="@mipmap/ic_launcher"
                android:label="@string/app_name"
                android:roundIcon="@mipmap/ic_launcher_round"
                android:supportsRtl="true"
                android:theme="@style/AppTheme"
                android:screenOrientation="locked">
                <activity android:name=".MainActivity">
                    <intent-filter>
                        <action android:name="android.intent.action.MAIN" />

                        <category android:name="android.intent.category.LAUNCHER" />
                    </intent-filter>
                </activity>
                <meta-data android:name="com.google.ar.core" android:value="required" />
            </application>

        </manifest>
```

Figura 3.13 Archivo AndroidManifest modificado para permitir uso de RA.

Siguiendo con el desarrollo de la aplicación base se prosigue a escribir la clase java que sirve de punto de entrada para la aplicación esta clase es “MainActiviti.java”, en esta clase se realiza la carga de las clases complementarias y la acción de aumentar los elementos de RA para la aplicación.

El método “onCreate” como se muestra en la figura 3.14 se encarga de instanciar clases de la API de ARCore necesarias para el funcionamiento de la RA, así como también de indicar cual es la vista principal de la aplicación.

```java
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        arFragment = (ArFragment) getSupportFragmentManager().findFragmentById(R.id.ux_fragment);
        arFragment.getArSceneView().getScene().addOnUpdateListener(this::onUpdateFrame);

        texture = new ExternalTexture();
    }
```

Figura 3.14 Método onCreate e la clase MainActivity.

El método de onUpdateFrame de la clase MainActivity se encarga de instanciar las clases responsables del manejo de la base de datos de marcadores de RA y la carga y creación de los diferentes elementos a aumentar en la aplicación, así también se encarga de analizar cuadro por cuadro capturado con la cámara del

dispositivo para buscar algún elemento de RA que desencadene una acción esta acción se observa en la figura 3.15.

```
53      private void onUpdateFrame(FrameTime frameTime) {
54
55          Frame frame = arFragment.getArSceneView().getArFrame();
56
57          // If there is no frame, just return.
58          if (frame == null) {
59              return;
60          }
61
62          Collection<AugmentedImage> updatedAugmentedImages = frame.getUpdatedTrackables(AugmentedImage.class);
63
64          for (AugmentedImage augmentedImage : updatedAugmentedImages) {
65              switch (augmentedImage.getTrackingState()) {
66                  case PAUSED:
67                      Toast toast = Toast.makeText(this, "Imagen detectada: " + augmentedImage.getName(),
68                              Toast.LENGTH_SHORT);
69                      toast.show();
70                      break;
71
72                  case TRACKING:
73                      switch (augmentedImage.getIndex()) {
74                      }
75                      break;
76
77                  case STOPPED:
78                      System.out.println("!!!STOPPED!!!");
79                      // augmentedImageMap.remove(augmentedImage);
80                      break;
81              }
82          }
83
84      }
```

Figura 3.15 Método onUpdateFrame de la clase MainActivity.

Siguiendo con el desarrollo del proyecto base se escribió la clase AugmentedImageFragment dicha clase se encarga de configurar la aplicación y de cargar la base de datos de marcadores a utilizar por la aplicación, este proceso de configuración y de carga de base de datos lo realiza al momento de que se carga la vista principal en la aplicación; a continuación, se detallan los métodos contenidos en esta clase.

Como se observa en la figura 3.16 el método “onCreateView” se encarga de configurar la vista principal de la aplicación, ocultando elementos que no son necesarios para la aplicación.

```
35      @Override
36      public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container,
37              @Nullable Bundle savedInstanceState) {
38          View view = super.onCreateView(inflater, container, savedInstanceState);
39          getPlaneDiscoveryController().hide();
40          getPlaneDiscoveryController().setInstructionView(null);
41          getArSceneView().getPlaneRenderer().setEnabled(false);
42          return view;
43      }
```

Figura 3.16 Método onCreateView de la clase AugmentedImageFragment.

El método mostrado en la figura 3.17 es el encargado de configurar la base de datos donde se almacena la información de los elementos que desencadenan una operación de RA.

```
45      @Override
46      protected Config getSessionConfiguration(Session session) {
47          Config config = super.getSessionConfiguration(session);
48          config.setFocusMode(Config.FocusMode.AUTO);
49          if (!setupAugmentedImageDatabase(config, session)) {
50              Toast toast = Toast.makeText(getContext(), "Error con la base de datos imagenes",
51                      Toast.LENGTH_SHORT);
52              toast.show();
53          }
54          return config;
55      }
56
```

Figura 3.17 Método getSessionConfiguration de la clase AugmentedImageFragment.

En la figura 3.18 se aprecia el método setupAugmentedImageDatabase que se encarga de cargar la base de datos de los elementos disparadores de RA y pasarla a la configuración global de la aplicación, una vez que se carga la base de datos en la configuración global de la aplicación se puede hacer uso de esta.

```
57      private boolean setupAugmentedImageDatabase(Config config, Session session) {
58          AugmentedImageDatabase augmentedImageDatabase;
59          AssetManager assetManager = getContext() != null ? getContext().getAssets() : null;
60          augmentedImageDatabase = new AugmentedImageDatabase(session);
61          Bitmap augmentedImageBitmap;
62          InputStream is;
63          try {
64              is = assetManager.open("test1.jpg");
65              augmentedImageBitmap = BitmapFactory.decodeStream(is);
66              augmentedImageDatabase.addImage("test1", augmentedImageBitmap);
67              is = assetManager.open("foto1.jpeg");
68              augmentedImageBitmap = BitmapFactory.decodeStream(is);
69              augmentedImageDatabase.addImage("foto1.", augmentedImageBitmap);
70          } catch (IOException e) {
71              e.printStackTrace();
72          }
73          config.setAugmentedImageDatabase(augmentedImageDatabase);
74          return true;
75      }
76
77  }
```

Figura 3.18 Método setupAugmentedImageDatabase de la clase AugmentedImageFragment.

La última clase creada para el proyecto base de las aplicaciones *Android* es AugmentedImageNode esta clase es la encargada de cargar a memoria los modelos 3D que se agreguen a la aplicación mediante el uso de la plataforma web como se muestra en la figura 3.19.

```java
@SuppressWarnings({ "AndroidApiChecker" })
public class AugmentedImageNode extends AnchorNode {

    private AugmentedImage image;

    public AugmentedImageNode(Context context) {
        //Proceso de carga de modelos
    }

    @SuppressWarnings({ "AndroidApiChecker", "FutureReturnValueIgnored" })
    public void setImage(AugmentedImage image, String modelo) {
        //Construccion de los modelos
    }

    public AugmentedImage getImage() {
        return image;
    }
}
```

Figura 3.19 Clase AugmentedImageNode.

Como se puede apreciar en la figura anterior esta clase solo cuenta con dos métodos, el primer método se encarga de cargar en memoria los modelos 3d de la aplicación, así como también de las texturas de estos, el segundo método sólo es el encargado de devolver un valor de imagen que utiliza la aplicación.

3.3.2 Desarrollo de la plataforma web

Como parte del desarrollo de la plataforma web se ha utilizado el lenguaje de programación web PHP haciendo uso del marco de trabajo Laravel.

Como primer punto en el desarrollo de la plataforma es definir las rutas que este seguirá para la navegación y acciones de este, para definir las rutas se escriben en el archivo “web.php” en dicho archivo se escribe la ruta de la página, controlador y el método de transporte que este ocupe ya sea post o get, esto se puede observar en la figura 3.20.

```php
Route::get('/', function () {
    return view('welcome');
});

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');

Route::resource('/aplicacion', 'CrearAppAndroidController');
Route::get('/aplicacion/{id}/{nombreapp}/edit', 'CrearAppAndroidController@edit');
Route::get('/aplicacion/construir/{id}/{nobmreapp}','CrearAppAndroidController@buildApp');
Route::get('/aplicacion/download/{nombreapp}', 'CrearAppAndroidController@download');
Route::get('/aplicacion/downloadcodigo/{nombreapp}','CrearAppAndroidController@downloadCodigo');

Route::post('/modelo', 'Modelo3dController@storeModelo');
Route::get('/modelo/{id}/{nombreapp}', 'Modelo3dController@delModelo');

Route::post('/imagen', 'ImgenAumentarController@subirImagen');
Route::get('/imagen/{id}/{nombreapp}', 'ImgenAumentarController@delImagen');

Route::post('/texto', 'AumentarTextoController@subirTexto');
Route::get('/texto/{id}/{nombreapp}', 'AumentarTextoController@borrarTexto');

Route::post('/video', 'AumentarVideoController@subirVideo');
Route::get('/video/{id}/{nombreapp}', 'AumentarVideoController@borrarVideo');

Auth::routes(['verify' => true]);
```

Figura 3.20 Rutas web utilizada por la plataforma.

Como se muestra en la figura 3.20 cada ruta de la plataforma se corresponde con una vista web o un controlador de la plataforma, los controladores son los scripts PHP que brindan su funcionalidad a la plataforma estos controladores se encargan de manejar desde el inicio de sesión en la plataforma hasta la creación de código fuente y la compilación de los archivos APK que el usuario puede descargar. Los controladores se encuentran contenidos en una ruta provista por el *framework* Laravel, los controladores realizados para la plataforma web se muestran en la figura 3.21.

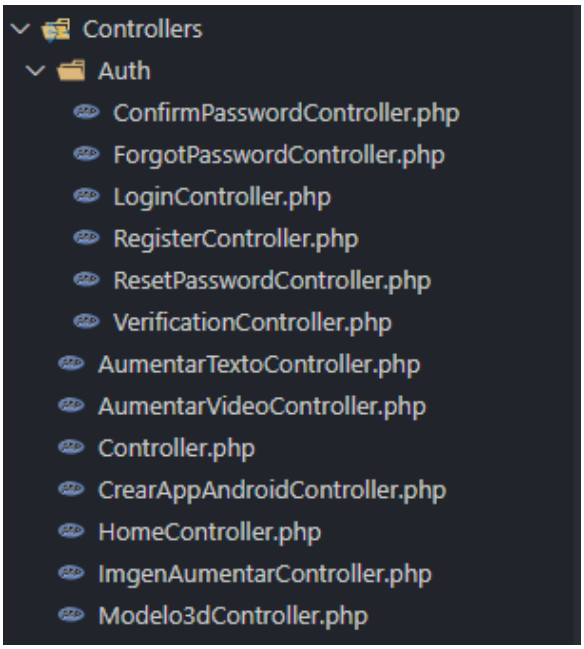

Figura 3.21 Controladores de la plataforma web.

Los controladores que se encuentran contenidos dentro de la carpeta “Auth” son creados por el propio *framework*. Los controladores que se encuentran fuera de esta carpeta son de autoría propia y proveen los métodos necesarios para hacer funcionar al sistema, siguiendo con la explicación de cada controlador se procede a mostrar cada uno dando una explicación de los métodos que lo conforman.

El primer controlador con el que el usuario interactúa al ingresar al sistema es el controlador “HomeController” como se aprecia en la figura 3.22, este controlador se encarga de verificar que el usuario esté autorizado en el sistema este proceso se realiza en el constructor del controlador, el método index() que contiene este controlador se encarga de redirigir al usuario a su página principal junto con una lista de las aplicaciones que este ha realizado utilizando el sistema.

app > Http > Controllers > HomeController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use App\AplicacionAndroid;

class HomeController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware(['auth', 'verified']);
    }

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Contracts\Support\Renderable
     */
    public function index()
    {
        $apps = AplicacionAndroid::where('user_id',Auth::id())->get();
        return view('home', compact('apps'));
    }
}
```

Figura 3.22 Controlador HomeController.

El siguiente controlador que se creó es “CrearAppAndroidController” mostrado en la figura 3.23 este controlador es el encargado de administrar y compilar las aplicaciones *Android* realizadas con el sistema, este controlador es el más extenso tanto en número de líneas como en métodos. Los métodos que abarca este controlador son:

- create (): método que solo se realiza la operación de mostrar la vista para la creación de una nueva aplicación.
- store(Request $request): método que realiza la operación de almacenaje de la información de la aplicación en la base de datos, creación de la estructura de carpetas de la aplicación y la escritura de los primeros archivos necesarios para esta.
- edit($id,$nombreapp): método que se encarga de regresar la vista para la edición de una aplicación ya creada.
- destroy($id): método que realiza la operación de borrado de la aplicación de la base de datos y la destrucción de la estructura de carpetas de la aplicación.
- buildApp($id,$nombreapp): método que realiza la operación de creación de los archivos restantes para la compilación de la aplicación.

- download($nombreapp): método que realiza la operación de compilación de la aplicación para generar el archivo con extensión APK y descarga del mismo.
- downloadCodigo($nombreapp): método con la operación de comprimir el código fuente de la aplicación creada en un archivo zip y descarga de este mismo archivo.

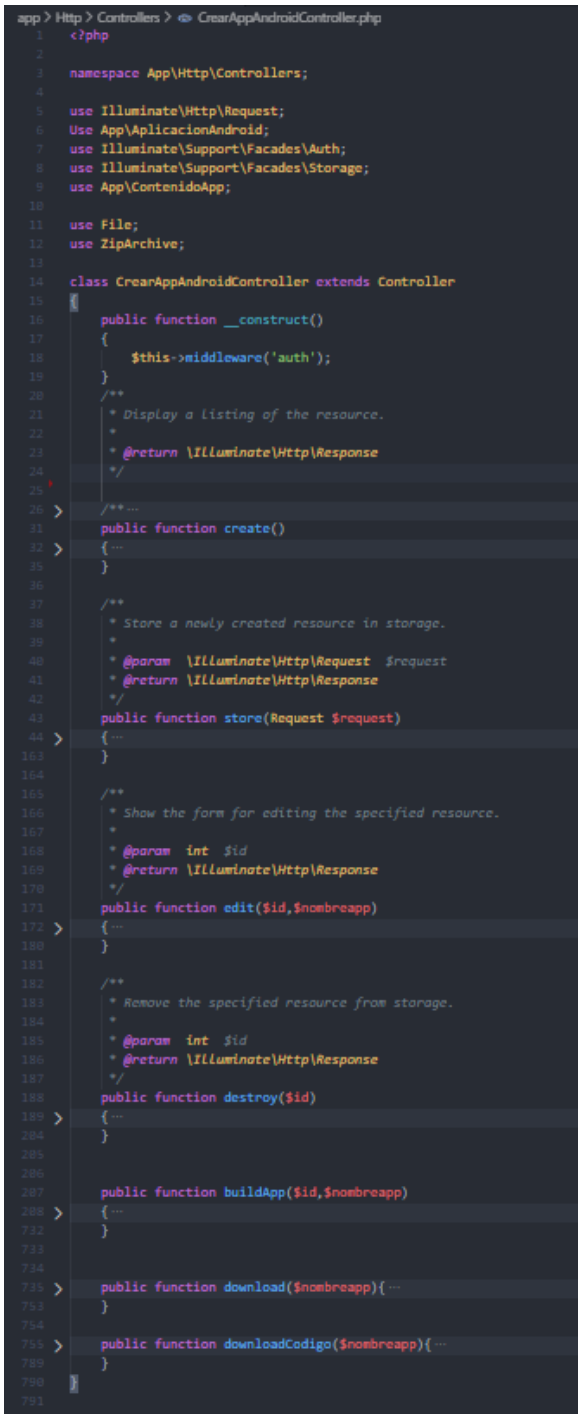

Figura 3.23 Controlador CrearAppAndroidController.

El siguiente controlador escrito es el controlador de “Modelo3dController” que se aprecia en la figura 3.24 este controlador tiene la función de agregar modelos 3D para la aplicación y eliminar los modelos agregados a la aplicación, el controlador solo contiene dos métodos que se describen a continuación.

- storeModelo(Request $request): método que se encarga de almacenar el modelo 3D para la aplicación, este método también tiene la función de invocar la utilidad de “arcoreimg.exe” que se encarga de analizar la calidad de la imagen a utilizar como marcador.
- delModelo(Request $request): método con la función de eliminar el modelo 3D de la aplicación.

app > Http > Controllers > Modelo3dController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\ContenidoApp;
use App\AplicacionAndroid;

class Modelo3dController extends Controller
{
    //

    public function __construct()
    {
        $this->middleware('auth');
    }

    public function storeModelo(Request $request)
    {…
    }

    public function delModelo(Request $request)
    {…
    }
}

```

Figura 3.24 Controlador Modelo3dController.

El resto de los controladores realizados para la plataforma resultan con una estructura similar la mostrada en la figura 3.3.2.5 la cual consiste en el método constructor de la clase PHP, un método para tratar el contenido de la aplicación y

su correspondiente método para eliminar dicho contenido de la aplicación, dichos controladores se muestran en las siguientes figuras que son 3.25, 3.26.

app > Http > Controllers > ImgenAumentarController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\ContenidoApp;
use App\AplicacionAndroid;

class ImgenAumentarController extends Controller
{
    //
    public function __construct()
    {
        $this->middleware('auth');
    }

    public function subirImagen(Request $request)
    {···
    }

    public function delImagen(Request $request){···
    }


}
```

Figura 3.25 Controlador ImgenAumentarController.

app > Http > Controllers > AumentarTextoController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\ContenidoApp;
use App\AplicacionAndroid;

class AumentarTextoController extends Controller
{
    //
    public function __construct()
    {
        $this->middleware('auth');
    }

    public function subirTexto(Request $request){···
    }

    public function borrarTexto(Request $request){···
    }
}
```

Figura 3.26 Controlador AumentarTextoController.

El último controlador escrito fue el controlador encargado de manejar la carga de video para las aplicaciones, este controlador al igual que los demás tiene la función de verificar la calidad de la imagen a utilizar como marcador, así como de también utilizar la herramienta FFMPEG para la conversión del video a formato “mp4” que utiliza el *codec* de video “h.264”, la estructura de este controlador se muestra en la figura 3.27 y las instrucciones para realizar la conversión del video en la figura 3.28.

app > Http > Controllers > AumentarVideoController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\ContenidoApp;
use App\AplicacionAndroid;

class AumentarVideoController extends Controller
{
    //
    public function __construct()
    {
        $this->middleware('auth');
    }

    public function subirVideo(Request $request){ ···
    }

    public function borrarVideo(Request $request){ ···
    }
}
```

Figura 3.27 Controlador AumentarVideoController.

```
$file->move(public_path()."/videos/",$file->getClientOriginalName());
$lowerCaseVIdeo = strtolower($file->getClientOriginalName());
$convertVideo = "ffmpeg -i \"".public_path()."\\videos\\".
$file->getClientOriginalName()."\" \"".public_path()."/appsgeneradas/
$request->nombreapp/app/src/main/res/raw/".substr($lowerCaseVIdeo,0,
-4).".mp4\"";
exec($convertVideo);
```

Figura 3.28 Instrucciones para convertir el video.

Siguiendo con el desarrollo de la plataforma web se crearon las vistas que el navegador web muestra ya que son la parte con la cual el usuario interactúa, las plantillas se crearon utilizando el motor de plantilla propio del *framework* Laravel que lleva por nombre Blade, las plantillas creadas con este motor tienen una estructura de nombre “nombrearchivo.blade.php”, la ventaja de utilizar este motor de plantillas es que permite extender desde una plantilla base evitando así tener que repetir código entre las distintas vistas de la plataforma. Las vistas y la plantilla utilizadas

por la plataforma se encuentran contenidas en la ruta “resources/views” como se muestra en la figura 3.29.

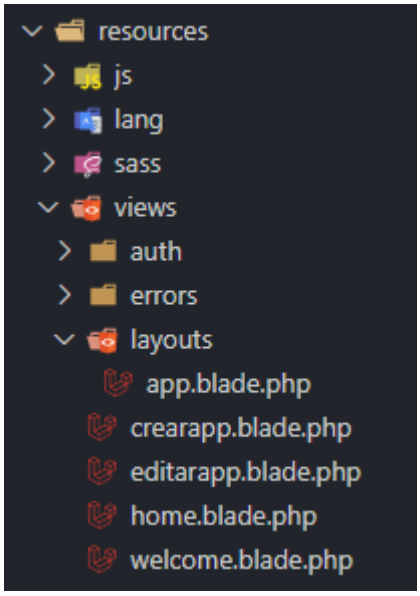

Figura 3.29 Ubicación de la vistas y plantilla de la plataforma.

La plantilla utilizada es una combinación de código PHP y código HTML utilizando la sintaxis del motor Blade, una muestra del código de la plantilla es el uso de etiquetas especiales para modificar la barra de navegación permitiendo así una distinción entre un usuario con sesión iniciada en el sistema o un visitante como se muestra en la figura 3.30.

```
@guest
    <li class="nav-item">
        <a class="nav-link" href="{{ route('login') }}">{{ __('Login') }}</a>
    </li>
    @if (Route::has('register'))
        <li class="nav-item">
            <a class="nav-link" href="{{ route('register') }}">Registro</a>
        </li>
    @endif
@else
    <li>
        <a class="nav-link" href="{{ url('/aplicacion/create') }}">Crear App</a>
    </li>
    <li class="nav-item dropdown">
        <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-toggle="dropdown"
        aria-haspopup="true" aria-expanded="false" v-pre>
            {{ Auth::user()->name }} <span class="caret"></span>
        </a>

        <div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdown">
            <a class="dropdown-item" href="{{ route('logout') }}"
               onclick="event.preventDefault();
                             document.getElementById('logout-form').submit();">
                {{ __('Logout') }}
            </a>

            <form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;">
                @csrf
            </form>
        </div>
    </li>
@endguest
```

Figura 3.30 Barra de navegación con etiquetas Blade.

Las vistas del sistema extienden de la plantilla principal permitiendo así la escritura de un archivo que no tiene la necesidad de estar repitiendo código común entre todas las vistas del sistema, esta reutilización del código de la plantilla se muestra en el archivo de vista utilizando etiquetas especiales de Blade para denotar secciones que pueden ser rellenadas con el código necesario para la vista a utilizar.

En la figura 3.31 se muestra el código de la vista de la página principal del sistema una vez que el usuario ha iniciado sesión el mismo, se observa que el código de etiquetado HTML se encuentra entre las etiquetas de código Blade.

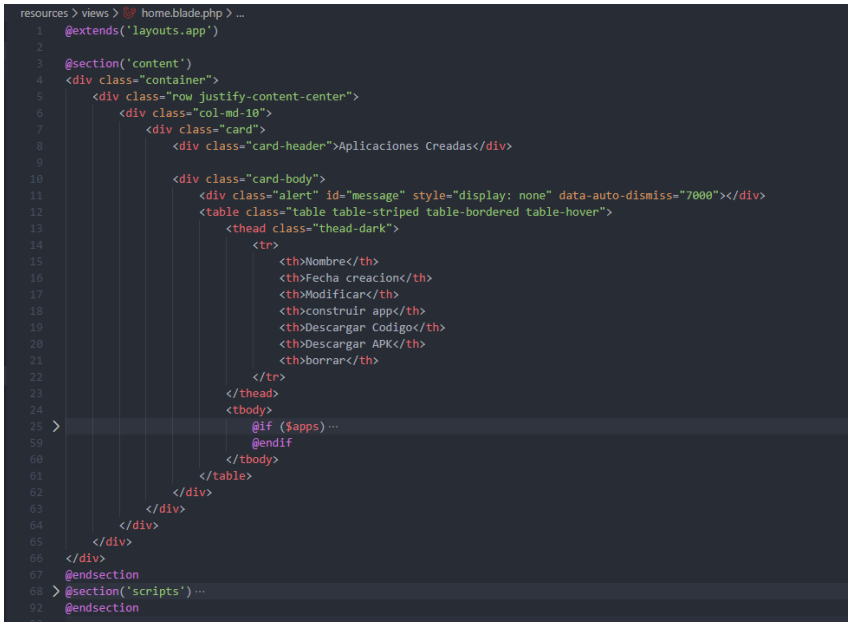

Figura 3.31 Vista del sistema extendiendo de la plantilla.

Capítulo 4 Capítulo 4 Resultados

En la presente sección se muestran los resultados obtenidos de los casos de estudio empleados para la realización del proyecto, estos casos de estudio son *markless*, GPS y NFC.

4.1 Caso de Estudio Markless

Para este caso de estudio se comprobó la viabilidad de la combinación de *markless* dentro de las aplicaciones creadas usando ARAPP *Builder* usando el API ARCore, para probar el caso de estudio aquí expuesto se realizaron diferentes aplicaciones de RA que mostraban diversos contenidos aumentados en la pantalla del dispositivo, entre los contenidos utilizados para realizar pruebas son un modelo 3D, una imagen y un video.

El resultado de las pruebas ha demostrado ser satisfactorio para la integración de *markless* dentro de las aplicaciones, ya que estas han podido mostrar los contenidos seleccionados al momento de detectar la imagen a utilizar como elemento disparador de la RA, como muestra de esto se presenta un fragmento de código realizado por la plataforma ARAPP *Builder* ya con la capacidad de distinguir un elemento disparador de RA y las instrucciones necesarias para desencadenar el elemento a mostrar observado en la figura 4.1.

```
68          for (AugmentedImage augmentedImage : updatedAugmentedImages) {
69              switch (augmentedImage.getTrackingState()) {
70                  case PAUSED:
71                      Toast toast = Toast.makeText(this,
72                              "Imagen detectada: " + augmentedImage.getName(), Toast.LENGTH_SHORT);
73                      toast.show();
74                      break;
75
76                  case TRACKING:
77                      switch (augmentedImage.getIndex()) {
78                          case 0://MODELO 3D
79                              ancla = augmentedImage.createAnchor(augmentedImage.getCenterPose());
80                              createVideo(ancla,augmentedImage.getIndex());
81                              break;
82                      break;
83
84                  case STOPPED:
85                      System.out.println("!!!STOPPED!!!");
86  //                    augmentedImageMap.remove(augmentedImage);
87                      break;
88              }
89          }
90
91      }
```

Figura 4.1 Código generado por ARAPP Builder.

El código mostrado en la figura 4.1 es el código generado por ARAPP *Builder* de una aplicación de RA que al momento de detectar un marcador de tipo *markless*

muestra en la pantalla un video sobre este, esta muestra de código generado ya hace uso del nuevo API de RA empleada para este proyecto permitiendo el correcto funcionamiento de la aplicación generada.

Como resultado de las pruebas realizadas de la integración de *markless* ha sido satisfactoria en las nuevas aplicaciones, así como también dentro de la plataforma de ARAPP *Builder* para demostrar esto último se muestran capturas de pantalla del dispositivo de pruebas mostrando diferentes elementos aumentados como se aprecia en las figuras 4.2 que muestra un modelo 3D sobre una imagen, 4.3 proyecta un video sobre una imagen y 4.4 despliega una imagen sobre la imagen disparadora de RA.

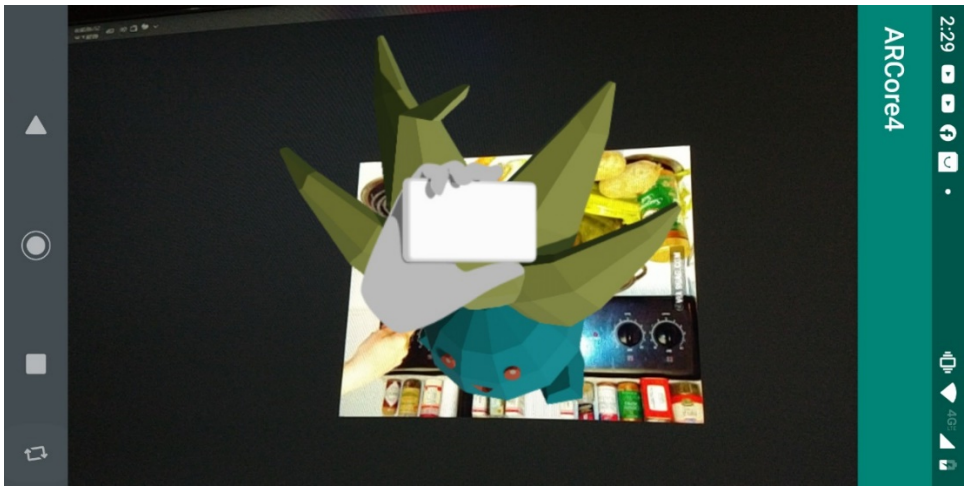

Figura 4.2 Aplicación de RA mostrando un modelo 3d.

Figura 4.3 Aplicación de RA mostrando un video.

Figura 4.4 Aplicación de RA mostrando una imagen 2D.

4.1.1 Rediseño Plataforma

Durante el desarrollo del caso de estudio *markless* también se trabajó el rediseño de la plataforma web debido a los cambios de tecnología usados mencionados con anterioridad en secciones previas del documento, a continuación, se muestra el rediseño de la plataforma ya con la nueva tecnología web utilizada para la misma.

En la figura 4.5 se muestra la pantalla de inicio del sistema mostrando el nuevo diseño que se le dio a la plataforma web.

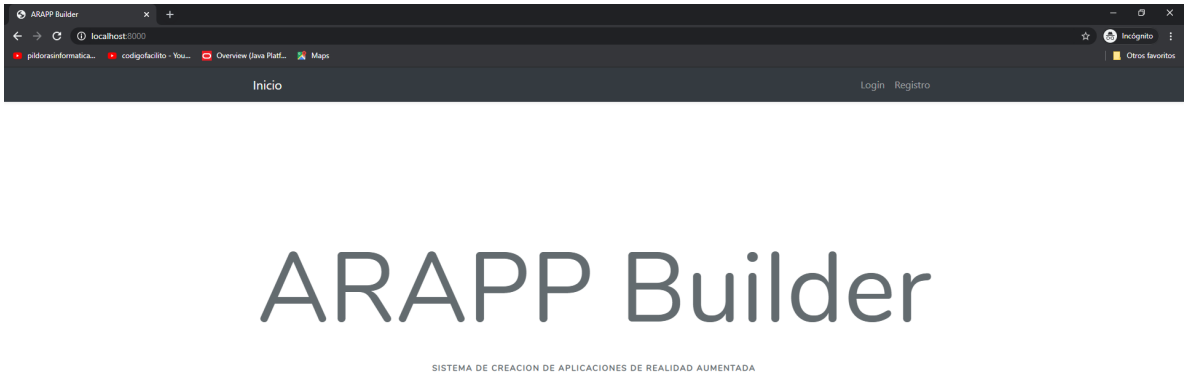

Figura 4.5 Pantalla inicial de la plataforma web.

Siguiendo con la presentación del diseño de la plataforma se muestra el formulario de registro de nuevos usuarios mostrado en la figura 4.6.

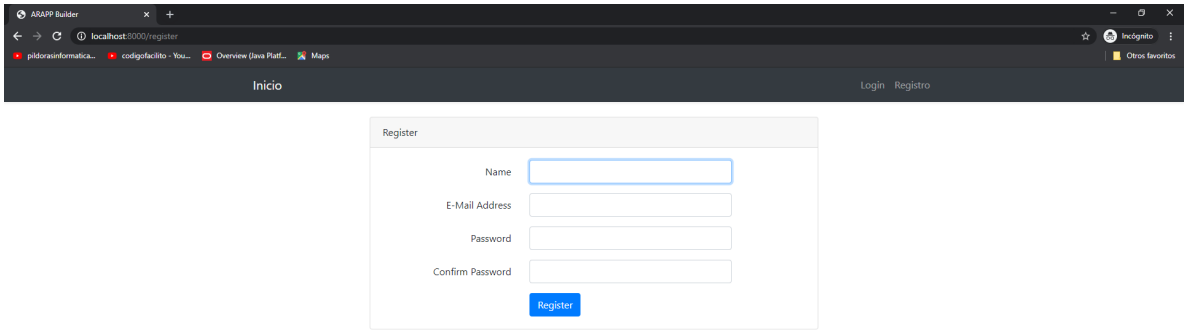

Figura 4.6 Formulario de registro de nuevos usuarios.

En la figura 4.7 se muestra la pantalla de inicio de sesión de la plataforma.

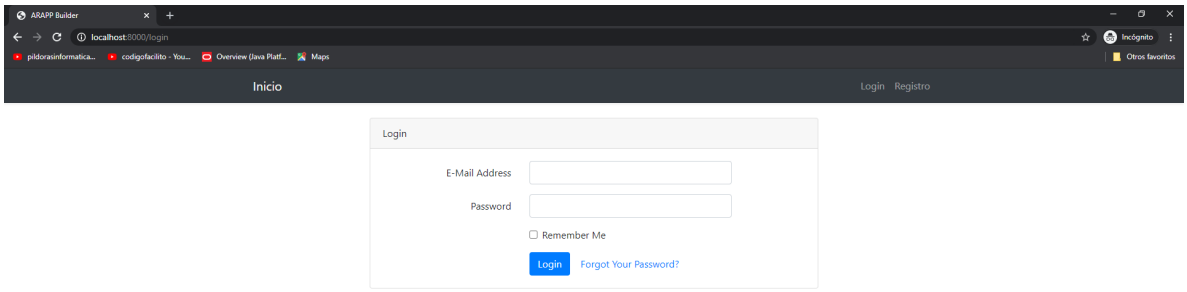

Figura 4.7 Formulario de inicio de sesión.

En la figura 4.8 se muestra la pantalla principal que tienen los usuarios una vez que han ingresado de manera correcta a la plataforma, en esta pantalla se muestran las aplicaciones que el usuario ha creado y las diferentes opciones que estas presentan.

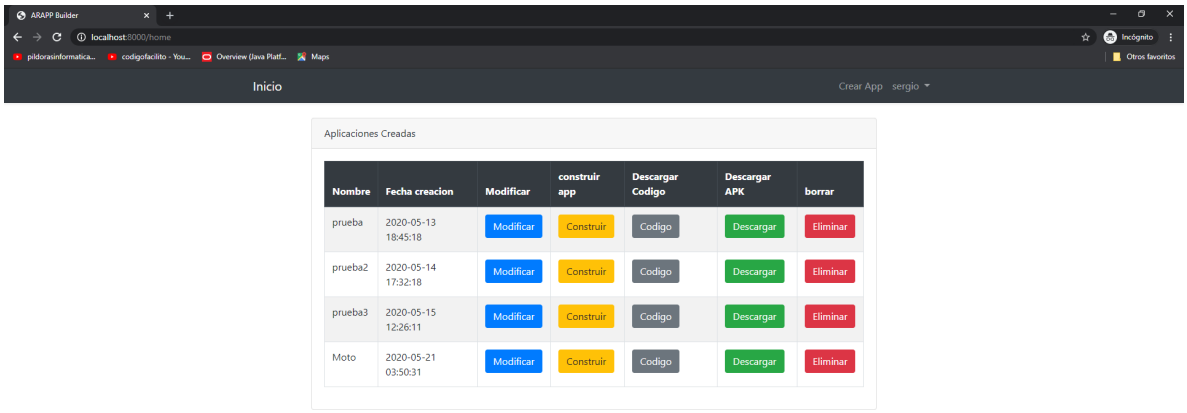

Figura 4.8 Pantalla principal de los usuarios.

El formulario para la creación de una nueva aplicación se muestra en la figura 4.9, en este formulario solo se brinda el nombre de la aplicación a crear, la plataforma identifica si el nombre dado ya está dado de alta y en caso de que ese nombre no se encuentre disponible le notifica al usuario como se aprecia en la figura 4.10.

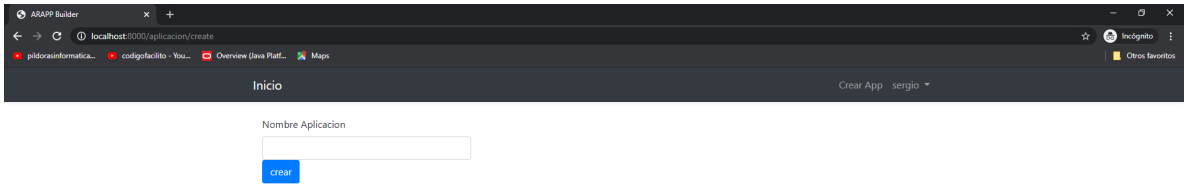

Figura 4.9 Formulario de creación de aplicación.

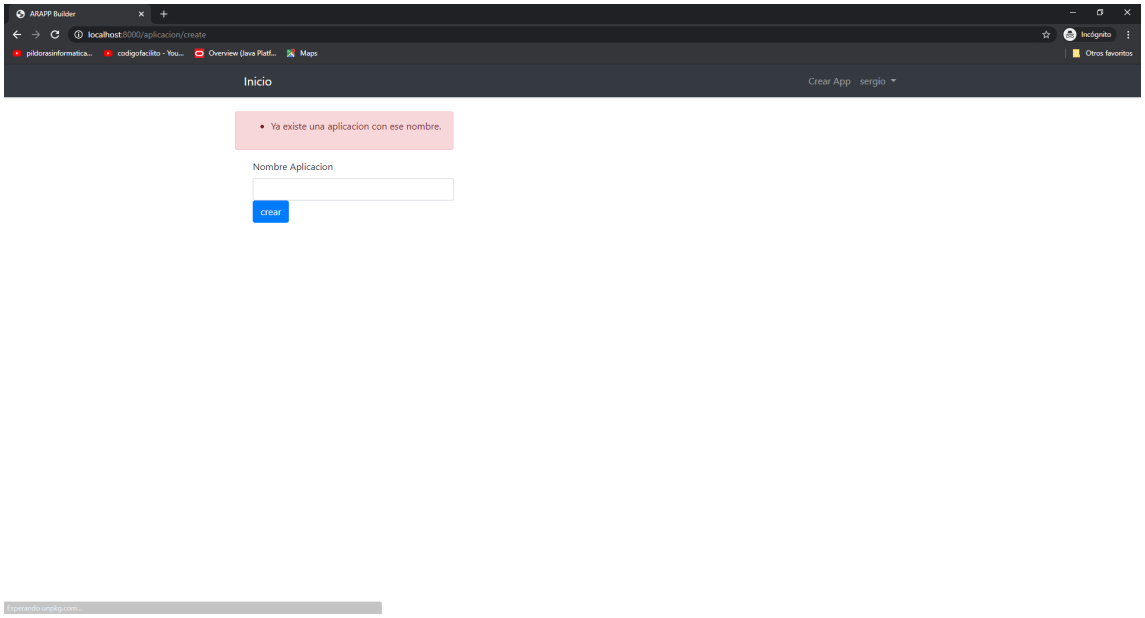

Figura 4.10 Pantalla que indica que ya hay una aplicación con ese nombre.

El formulario para la modificación de las aplicaciones creadas por el usuario se muestra en la figura 4.11, en esta pantalla se aprecia los diferentes contenidos que puede agregar el usuario a sus aplicaciones de RA.

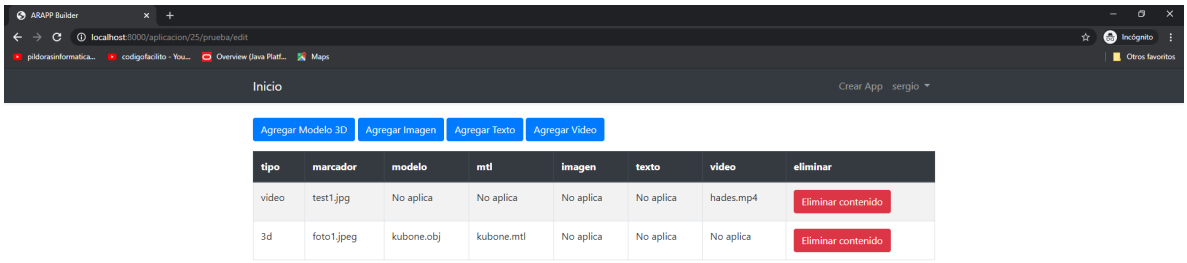

Figura 4.11 Pantalla de modificación de aplicación.

4.2 Caso de estudio GPS

En este segundo caso de estudio se realizó el trabajo necesario para la integración de la tecnología de GPS en las aplicaciones realizadas con la plataforma web ARAPP Builder.

Como parte de la integración de la tecnología de GPS con las aplicaciones de RA se realizó una investigación para determinar si la combinación de dichas tecnologías era posible y los requerimientos que la tecnología de GPS necesita para funcionar de manera correcta en los dispositivos móviles con sistema *Android*.

El resultado de la investigación sobre la combinación de las dos tecnologías fue satisfactorio demostrando que la combinación era posible y ya se ha realizado con anterioridad mostrando varios exponentes de esta combinación de tecnologías como se muestra en la siguiente figura 4.12 donde se muestra un modelo 3D suspendido en el espacio sin la necesidad de un elemento disparador de RA.

Figura 4.12 Elemento de RA localizado en el espacio físico sin marcador.

Como se puede apreciar en la figura 4.12 el modelo 3D se encuentra en suspendido en el espacio físico sin la necesidad de algún elemento disparador, este es un ejemplo de la correcta función de la tecnología de GPS y RA, para replicar este comportamiento en las aplicaciones realizadas con la plataforma web se procedió a realizar varias aplicaciones de prueba que hagan uso de las dos tecnologías.

Los resultados de las aplicaciones de prueba demostraron que en este punto la integración de la API ARCore y la tecnología de GPS es funcional hasta cierto punto, ya que esta presenta ciertas complicaciones al momento de integrar estas en una sola aplicación. Las aplicaciones presentan cierta inestabilidad al momento de mostrar un elemento de RA en la pantalla del dispositivo, además de que el API detecta patrones en determinados lugares.

Las aplicaciones de prueba fueron realizadas haciendo uso del IDE (*Integrated Development Environment* por sus siglas en inglés) *Android Studio*, estas aplicaciones de prueba emplearon la API de RA ARCore y la información que el GPS del dispositivo de pruebas proporcionaba, el funcionamiento de las aplicaciones de prueba consistía en la colocación de las coordenadas de un punto en el espacio del cual se le daba un radio de veinte metros para generar una cerca geográfica donde se definen los puntos donde se vizualizara el contenido

aumentado. El funcionamiento de las aplicaciones de prueba demostró ser inestable al momento de detectar los puntos definidos dentro de la cerca geográfica, los elementos aumentados se mostraban de manera errática de en la patalla del dispositivo de prueba y otro comportamiento que presentaban las aplicaciones de prueba es el mostrado en la figura 4.13 donde las aplicaciones reconocían patrones dentro del cerco definido.

Figura 4.13 Aplicación de RA y GPS detectando patrones en una superficie.

4.3 Caso de estudio NFC

Para verificar si la combinación de NFC y ARCore se procedió al desarrollo de aplicaciones que hagan uso de estas dos tecnologías, estas aplicaciones fueron realizadas empleando el IDE de desarrollo *Android Studio*, el funcionamiento esperado de las aplicaciones realizadas consistía en que cuando la aplicación detectara la proximidad de una etiqueta NFC esta leyera el contenido de esta y mostrara en pantalla el resultado de la lectura.

Los resultados de estas aplicaciones de prueba no fueron satisfactorios ya que estas presentan inestabilidad al momento de funcionamiento. Las aplicaciones de

pruebas al momento de cargar la funcionalidad de AR entran en conflicto con la API para el manejo de NFC provocando inestabilidad y en algunos casos cierre de la aplicación al detectar una etiqueta NFC.

Conclusiones

Como se ha demostrado en este documento el desarrollo de las nuevas funcionalidades para la plataforma ARAPP *Builder* se ha llevado con éxito combinando las nuevas tecnologías mencionadas con anterioridad, el cambio de la tecnología de la plataforma web de utilizar JAVA a PHP ha permitido hacer un desarrollo más veloz gracias al uso del *framework* Laravel, el empleo de este permite automatizar varias tareas del sistema así como una escritura de código necesaria para el funcionamiento de una manera más veloz.

Por su parte la integración del uso de *markless* dentro de las aplicaciones de RA ha resultado ser el mayor éxito del desarrollo, ya que esta característica se logró replicar empleando ARCore como API de RA, esta API ha demostrado ser efectiva para el desarrollo de *markless* ya que ha permitido a las aplicaciones identificar imágenes sin características propias de un marcador de RA como es el uso de un marco especial para la imagen, esta característica añadida a la plataforma web ARAPP *Builder* es de gran utilidad para los usuarios finales ya que les permite utilizar las diferentes imágenes que tienen a su disposición en sus equipos de cómputo o en sus dispositivos móviles.

El uso de GPS con RA ha demostrado ser efectivo para el desarrollo de aplicaciones de prueba, los resultados de obtenidos de estas aplicaciones de prueba han demostrado que la combinación de GPS y RA es posible aun que esta necesita más trabajo a futuro por parte de las compañías desarrolladoras de las APIs para mejorar la integración de estas, en base a esto no se ha descartado que en un futuro sea posible integrar estas tecnologías de mejor manera.

NFC y AR ya han demostrado ser compatibles en trabajos anteriores, pero cuando se combinan las API de NFC y ARCore se presentan inestabilidades en las aplicaciones a la fecha de redacción de este documento, no se descarta la posibilidad que en un futuro estas complicaciones de las tecnologías sean superadas gracias al desarrollo de las plataformas de AR y NFC.

Referencias

[1] F. N. Afif and A. H. Basori, “Orientation Control for Indoor Virtual Landmarks based on Hybrid-based Markerless Augmented Reality,” *Procedia - Soc. Behav. Sci.*, vol. 97, pp. 648–655, 2013, doi: 10.1016/j.sbspro.2013.10.284.

[2] J. A. Erkoyuncu, I. F. del Amo, M. Dalle Mura, R. Roy, and G. Dini, “Improving efficiency of industrial maintenance with context aware adaptive authoring in augmented reality,” *CIRP Ann. - Manuf. Technol.*, vol. 66, no. 1, pp. 465–468, 2017, doi: 10.1016/j.cirp.2017.04.006.

[3] J. Paulo Lima *et al.*, “Markerless tracking system for augmented reality in the automotive industry,” *Expert Syst. Appl.*, vol. 82, pp. 100–114, 2017, doi: 10.1016/j.eswa.2017.03.060.

[4] H. S. Kim, Sangmi-Park, Sunju-Han, and L. S. Kang, “AR-based 4D CAD System Using Marker and Markerless Recognition Method,” *Procedia Eng.*, vol. 196, no. June, pp. 29–35, 2017, doi: 10.1016/j.proeng.2017.07.169.

[5] S. H. Huang, Y. I. Yang, and C. H. Chu, “Human-centric design personalization of 3D glasses frame in markerless augmented reality,” *Adv. Eng. Informatics*, vol. 26, no. 1, pp. 35–45, 2012, doi: 10.1016/j.aei.2011.07.008.

[6] A. Pyae and L. E. Potter, “A player engagement model for an augmented reality game: A case of Pokémon Go,” *Proc. 28th Aust. Comput. Interact. Conf. OzCHI 2016*, pp. 11–15, 2016, doi: 10.1145/3010915.3010960.

[7] E. Duque-bedoya, “Usando realidad aumentada para motivar las competencias informacionales : experiencias en clase .,” *XV Encuentro Int. Virtual Educ.*, pp. 1–11, 2014.

[8] A. Rese, D. Baier, A. Geyer-Schulz, and S. Schreiber, “How augmented reality apps are accepted by consumers: A comparative analysis using scales and opinions,” *Technol. Forecast. Soc. Change*, vol. 124, pp. 306–319, 2017, doi: 10.1016/j.techfore.2016.10.010.

[9] I. Media and T.--synthetic F. Media, “About the Technology,” 2001. http://nfc-forum.org/what-is-nfc/about-the-technology/.

[10] “Sistema de Posicionamiento Global,” 2017. https://www.gps.gov/spanish.php.

[11] Red Hat, “¿Qué es una API?,” *Red Hat*, 2014. https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces.

[12] artoolkitX, “What is artoolkitX?,” 2018. [Online]. Available: https://github.com/artoolkitx/artoolkitx/wiki/Introduction-to-artoolkitX.

[13] Google, “Choose Your Development Environment | ARCore | Google

Developers," *ARcore Documentation*, 2017. https://developers.google.com/ar/develop/.

[14] Android Developers, "Aspectos fundamentales de la aplicación," 2016. https://developer.android.com/guide/components/fundamentals?hl=es-419.

[15] The PHP Group, "¿Qué es PHP?" https://www.php.net/manual/es/intro-whatis.php.

[16] Laravel LLC, "Laravel." https://laravel.com/docs/6.x.

[17] D. López, I. López, L. A. Reyes, C. Romero, and B. A. Olivares, "Arquitectura y procesos para el desarrollo de un Sistema de Autor de aplicaciones móviles con Realidad Aumentada," 2018.

[18] W. Fang, L. Zheng, and X. Wu, "Multi-sensor based real-time 6-DoF pose tracking for wearable augmented reality," *Comput. Ind.*, vol. 92–93, pp. 91–103, 2017, doi: 10.1016/j.compind.2017.06.002.

[19] P. Saeghe *et al.*, "Augmenting television with augmented reality," *TVX 2019 - Proc. 2019 ACM Int. Conf. Interact. Exp. TV Online Video*, pp. 255–261, 2019, doi: 10.1145/3317697.3325129.

[20] P. Verma, K. Agrawal, and V. Sarasvathi, "Indoor Navigation Using Augmented Reality," *Proc. 2020 4th Int. Conf. Virtual Augment. Real. Simulations*, pp. 58–63, 2020, doi: 10.1145/3385378.3385387.

[21] A. C. C. Reyes, N. P. A. Del Gallego, and J. A. P. Deja, "Mixed Reality Guidance System for Motherboard Assembly Using Tangible Augmented Reality," *Proc. 2020 4th Int. Conf. Virtual Augment. Real. Simulations*, pp. 1–6, 2020, doi: 10.1145/3385378.3385379.

[22] L. C. Wu, I. C. Lin, and M. H. Tsai, "Augmented reality instruction for object assembly based on markerless tracking," *Proc. - 20th ACM SIGGRAPH Symp. Interact. 3D Graph. Games, I3D 2016*, pp. 95–102, 2016, doi: 10.1145/2856400.2856416.

[23] W. Viyanon, T. Songsuittipong, P. Piyapaisarn, and S. Sudchid, "AR furniture: Integrating augmented reality technology to enhance interior design using marker and markerless tracking," *ACM Int. Conf. Proceeding Ser.*, vol. Part F131840, 2017, doi: 10.1145/3144789.3144825.