

**Instituto politécnico Nacional
Escuela Superior de Cómputo**

**Trabajo terminal
2017-B062: B-Side
Prototipo de sistema de red social
utilizando realidad aumentada**

David Vega Ramirez y Victor Hugo Jiménez Duhart

8 de mayo de 2018

“Aceptamos la realidad del mundo que nos presentan.”

El show de Truman.

Instituto politécnico Nacional

Abstract

Escuela Superior de Cómputo

Por David Vega Ramirez y Victor Hugo Jiménez Duhart

En este documento se describe el sistema B-Side el cual es un prototipo de red social , el cual hace uso de realidad aumentada (RA) para las interacciones entre sus usuarios.

Índice general

Abstract	II
Lista de figuras	VI
Lista de tablas	VIII
Abreviaciones	X
1. Introducción	1
1.1. Objetivo	2
2. Marco Teórico	3
2.1. Red social	3
2.1.1. Definición	3
2.1.2. Forma y contenido	4
2.1.3. Comunidades de práctica	4
2.1.4. Interacción Humano-Computadora: Computación Ubicua	5
2.2. Realidad aumentada	6
2.2.1. Definición	6
2.2.2. Ejemplos	6
2.2.3. Visualización	10
2.2.3.1. Métodos de aumentación	11
2.2.4. Seguimiento	12
2.2.4.1. Seguimiento de marcador	13
2.2.4.2. Seguimiento sin marcadores	14
2.3. Geolocalización	15
2.4. Frameworks	15
2.4.1. ARCore	15
2.4.2. ARKit	16
2.4.3. Comparación	17
2.5. Servicios en la nube	17
2.5.1. Azure	17
2.5.2. Amazon Web Services	18
2.5.3. Google Cloud Platform	19
2.5.4. Comparación	19
2.6. Gestión de datos	20

2.6.1.	Hadoop	20
2.6.1.1.	Hive	20
2.6.2.	SQL Server	20
2.6.2.1.	T-SQL	21
2.6.3.	SQLite	21
2.7.	Telemetría	22
2.7.1.	Application Insights	22
2.7.2.	Google Analytics para aplicaciones mobiles	23
2.7.3.	Comparación	23
2.8.	PowerBi	24
2.9.	Material Design	24
2.10.	Lenguajes de programación	26
2.10.1.	C#	26
2.10.1.1.	Xamarin	27
2.10.2.	Java	28
3.	Análisis Factibilidad	29
3.1.	Análisis de mercado	29
3.2.	Viabilidad técnica	32
3.2.1.	Usabilidad	32
3.2.2.	Despliegue	32
3.2.3.	Soporte	33
3.2.4.	Características operacionales	33
3.2.5.	Escalabilidad	34
3.3.	Análisis financiero	35
3.3.1.	Costo de tecnología	35
3.3.2.	Roles requeridos	36
3.3.3.	Espacio a utilizar	37
4.	Reporte Técnico	38
4.1.	Metodología	38
4.2.	Requerimientos funcionales	39
4.3.	Requerimientos no funcionales	39
4.4.	Reglas del negocio	40
4.5.	Arquitectura del sistema	41
4.5.1.	Componentes	41
4.5.1.1.	Aplicación Móvil	41
4.5.1.2.	Componentes en Azure	42
4.6.	Clases del sistema	42
4.6.1.	Servidor	42
4.6.1.1.	DataManager	42
4.6.1.2.	Telemetry	43
4.6.1.3.	Communication	43
4.6.2.	Aplicación móvil	45
4.6.2.1.	Polygon	46
4.6.2.2.	Model	46
4.6.2.3.	Geolocation	47

4.6.2.4. Visualization	47
4.6.2.5. Creation	49
4.7. Descripción de actores	49
4.8. Características de la herramienta	50
4.8.1. C1. Identificación con la herramienta	50
4.8.2. C2. Interacción con aumentaciones	50
4.8.3. C3. Creación de aumentaciones	50
4.8.4. C4. Visualización de mapa	50
4.9. Datos de la herramienta	51
4.9.1. Identificación del usuario	51
4.9.2. Interacción con aumentaciones	51
4.9.3. Creación de aumentaciones	51
4.10. Máquina de Estados	52
4.11. Modelo de casos de uso	53
4.12. Descripción de casos de uso	55
4.12.1. CU1. Identificar al usuario	55
4.12.2. CU2. Registrar al usuario	56
4.12.3. CU3. Visualizar aumentaciones	58
4.12.4. CU4. Interactuar con aumentación	59
4.12.5. CU5. Crear aumentación	60
4.12.6. CU6. Visualizar mapa	61
4.13. Diagramas de secuencia	61
4.13.1. DS1. Identificar al usuario	62
4.13.2. DS2. Registrar al usuario	62
4.13.3. DS3. Visualizar aumentaciones	63
4.13.4. DS4. Interactuar con aumentación	63
4.13.5. DS5. Crear aumentación	64
4.14. Interfaces	65
4.14.1. P1: Login	65
4.14.2. P2: Registro	66
4.14.3. P3: Confirmación	67
4.14.4. P4: Principal	68
4.14.5. P5: Interacción	69
4.14.6. P6: Colocar	70
4.14.7. P7: Creación	71
4.14.8. P8: Mapa	72
4.15. Modelo de datos	73
4.15.1. Tablas	73

Índice de figuras

2.1. Campaña publicitaria usando RA en la Estación Victoria de Londres [1]	7
2.2. Promoción de documental de National Geographic usando RA [2]	7
2.3. Prototipo para mejorar el turismo en la ciudad de Roma usando RA [3]	7
2.4. Herramienta para visualizar productos de Lego en sus tiendas [4]	8
2.5. Aplicación que hace uso de RA para poder dibujar en imágenes del mundo real [5]	8
2.6. Periférico y aplicación para hacer que la Ps-Vita pueda hacer uso de RA [6]	9
2.7. Periférico para poder interactuar con el mundo haciendo uso de RA [7]	9
2.8. Aplicación que hace uso de RA y geolocalización para poder interactuar con 'Pokemones' [8]	10
2.9. OT utiliza un elemento óptico(Ej. un espejo) para combinar la visión del usuario con imágenes generadas por computadora [9]	11
2.10. VT captura el mundo real a través de una cámara, modifica la imagen tomada añadiendo modelos de RA y posteriormente se la muestra al usuario en algún tipo de monitor [9]	12
2.11. Modelos de RA proyectadas sobre una superficie [9]	12
2.12. Ejemplo de un marcador fiducial [9]	13
2.13. Flujo de datos de telemetría en Application Insights [10]	23
2.14. Ciclo de vida de los datos hasta ser utilizados por medio de PowerBi [11]	24
2.15. Ejemplo de una aplicación de tipo red social que utiliza Material Design	25
2.16. Ejemplo de una aplicación de calendario que utiliza Material Design	26
2.17. Ciclo de vida de una aplicación creada por medio de Xamarin [12]	27
3.1. Modelos de teléfonos compatibles con ARCore [13]	30
3.2. Pronostico de crecimiento de el mercado de RA en billones de dolares [14]	31
4.1. Diagrama de componentes de B-Side	41
4.2. Diagrama de clases de el servidor	42
4.3. Diagrama de clases de la aplicación móvil	45
4.4. Máquina de estados	53
4.5. Modelo de Casos de Uso	54
4.6. DS1. Identificar al usuario	62
4.7. DS2. Registrar al usuario	62
4.8. DS3. Visualizar aumentaciones	63
4.9. DS4. Interactuar con aumentación	63
4.10. DS5. Crear aumentación	64
4.11. Login de la aplicación	65
4.12. Registro de un nuevo usuario	66

4.13. Confirmación del registro	67
4.14. Interfaz principal	68
4.15. Interacción con un modelo de RA	69
4.16. Colocar una nuevo modelo de RA	70
4.17. Menú de creación	71
4.18. Mapa de modelos de RA populares en el área cercana	72
4.19. Modelo de datos del sistema	73

Índice de cuadros

3.1. Costos al desarrollar	35
3.2. Costos al publicar aplicación	35
4.1. Matriz de Casos de Uso	55

Lista de Códigos

2.1. Ejemplo de un script de Hive	20
2.2. Ejemplo de un script de T-SQL	21
2.3. Hola mundo, utilizando Xamarin y C#	27
2.4. Hola mundo, utilizando el Android SDK y Java	28

Abreviaciones

RA	Realidad Aumentada
AWS	Amazon Web Services
SQL	Structured Query Language
GPS	Global Positioning System
GSM	Global System for Mobile communication
SDK	Software Development Kit
JVM	Java Virtual Machine

Página vacía. . .

Capítulo 1

Introducción

La realidad aumentada ha permitido crear nuevas maneras en las que se puede interactuar con la tecnología y con la vida real al mismo tiempo, a través de objetos virtuales que interactúan con el mundo a nuestro alrededor, rompiendo así con una barrera entre el mundo digital y real.

En la actualidad las aplicaciones que utilizan Realidad Aumentada solo han permitido interactuar localmente a los usuarios, lo que se propone es eliminar esta barrera que existe y permitir la creación de un espacio por medio de una aplicación donde todos sus usuarios puedan convivir simultáneamente mediante el uso de geolocalización, un usuario podrá insertar una aumentación por medio de la aplicación en una ubicación geográfica y cuando un usuario distinto llegue a esta misma ubicación podrá observar lo que otro usuario ha creado en ese lugar.

La aplicación está diseñada para ser compatible con todos los teléfonos Android que soporten el framework de realidad aumentada ARCore y versiones de Android superiores a la versión 7.1., de igual manera la aplicación seguirá los estándares de Material Design para mantener una experiencia de usuario adecuada para una aplicación de Android. Se ha escogido Android como la plataforma para el desarrollo de la aplicación debido a la disponibilidad de tecnología y los costos, comparando Android contra IOS.

La escalabilidad ha sido un factor muy importante al momento de diseñar la arquitectura del sistema por lo que se hará uso de cómputo en la nube, más específicamente los servicios que proporciona Azure en cuanto a sistemas de bases de datos, servidores, almacenamiento y telemetría, esto debido a que gracias a Azure es posible escalar el sistema dependiendo de las necesidades que se presenten, por ejemplo, número de peticiones por segundo, espacio en disco duro o número de servidores. De igual manera es pertinente

saber cuando es que habrá que realizar cambios con respecto a la escalabilidad del sistema o su usabilidad en general, por lo que se ha contemplado el uso de telemetría lo cual permitirá recabar información de uso del sistema y con ella poder hacer estimaciones, estadísticas y en su momento cambios con base en datos.

No se utilizará Android nativo para el desarrollo de la aplicación si no que se realizara por medio de Xamarin, el cual es un software que encapsula C# hacia Android o IOS permitiendo hacer un desarrollo multiplataforma con un solo código(aunque en este caso solo haremos uso del encapsulado hacia android), se ha escogido Xamarin y por consecuente C# debido a que por medio de este lenguaje se permite un desarrollo mas homogéneo con las tecnologías de Azure.

1.1. Objetivo

Diseñar y crear una aplicación que le permitirá a sus usuarios crear e interactuar con objetos creados por sí mismos u otros usuarios haciendo uso de realidad aumentada y geolocalización. Por medio de la aplicación los usuarios podrán crear diferentes objetos por medio de polígonos. Un usuario de la aplicación podrá interactuar con los objetos de otros usuarios dependiendo de la ubicación geográfica donde fueron creados los objetos.

Capítulo 2

Marco Teórico

2.1. Red social

2.1.1. Definición

Las Redes Sociales pueden definirse como un conjunto bien delimitado de actores - individuos, grupos, organizaciones, comunidades, sociedades globales, etc.- vinculados unos a otros a través de una relación o un conjunto de relaciones sociales. Mitchell (1969:2) añade que las características de estos lazos en tanto que totalidad pueden ser usados para interpretar los comportamientos sociales de las personas implicadas, pero parece más bien un objetivo genérico que un criterio específico de definición. Otras definiciones son más instrumentales o más centradas en el aparato metodológico, como la de Freeman (1992:12): «colección más o menos precisa de conceptos y procedimientos analíticos y metodológicos que facilita la recogida de datos y el estudio sistemático de pautas de relaciones sociales entre la gente». El rasgo más característico de las redes sociales consiste en que requieren «conceptos, definiciones y procesos en los que las unidades sociales aparecen vinculadas unas a otras a través de diversas relaciones» (S. Wasserman y K. Faust, 1994:6). Se pueden dar diversas maneras de formalizar y medir los datos y el análisis de las redes sociales, las dos más importantes: la teoría de los grafos, operando a partir de productos cartesianos con los grafos como representación, y la teoría matricial, a partir de las sociomatrizes como matriz de datos inicial.

En las redes sociales podemos encontrar actores sociales, lazos relacionales, díadas, tríadas, grupos y subgrupos. Los actores sociales son entidades sociales sujetos de los vínculos de las redes sociales y los unen diversos tipos de vínculos como lo pueden ser: personales, transferencias de recursos, asociaciones, comportamiento, entre otros.

2.1.2. Forma y contenido

Cabe hacer en las redes una distinción entre forma y contenido. La aproximación de las redes, como muchos otros métodos en sociología, es un procedimiento de análisis formal, evidentemente sobre una base sustantiva de relaciones. Se recoge la información, se transforma en dato, se trata formalmente y se interpreta sustantivamente. Ambas caras, forma y contenido, son indisociables en el análisis de redes.

El contenido es la materia, la sustancia relacional -afecto, información, dinero, etc.- que «fluye» a través de las unidades por medio de las relaciones que se dan entre ellas a partir del intercambio de dicho contenido. Constituye la materialidad sociológica de la relación. Evidentemente implica, y es lo que realmente interesa, un tipo de comportamiento o acción, e incluso percepción de tal relación, entre dos unidades. Estas relaciones tienen interés para el sociólogo ya que han sido elegidas como problemática de investigación: una unidad tiene a la otra como objeto o efecto directo o indirecto y/o recíproco.

Por forma de la red se entienden la expresión abstracta de la relación y las propiedades de la configuración global o de algunas de las partes, es decir, lo que se suele describir como pautas, modelo o estructura de la red.

2.1.3. Comunidades de práctica

Las comunidades de práctica están formadas por personas que se comprometen en un proceso de aprendizaje colectivo en un dominio compartido del esfuerzo humano: una tribu aprendiendo a sobrevivir, una banda de artistas buscando nuevas formas de expresión, un grupo de ingenieros trabajando en problemas similares, una pandilla de amigos definiendo su identidad en la escuela, una red de cirujanos explorando las mejores técnicas. En resumen:

Las comunidades de práctica son grupos de personas que comparten una ocupación o pasión por algo que ellos hacen y aprenden cómo hacerlo mejor mientras interactúan regularmente. Esta definición nos deja, pero no asume, intencionalmente: el aprendizaje puede ser la razón por la que una comunidad se reúne o incidentalmente propiciar las interacciones de sus miembros. No toda comunidad es una comunidad de práctica. Un vecindario por ejemplo, es a menudo llamado una comunidad, pero no es una comunidad de práctica. Tres características son cruciales:

- Dominio: La comunidad de práctica se caracteriza por dominar ciertos temas, es lo que los distingue.

- Comunidad: Las personas construyen relaciones que les permiten aprender de todos. Realizan actividades que les permiten discutir sobre el tema que dominan. Un sitio web no es una comunidad de práctica. Tener el mismo empleo o el mismo título no implica que haya una comunidad de práctica al menos que interactúen y aprendan juntos. Los miembros de una escuela religiosa pueden tener mucho en común, incluso aprender juntos, pero eso no los hacen una comunidad de práctica. Los Impresionistas, por ejemplo, solían reunirse en cafés y estudios para discutir el estilo de las pinturas que estaban haciendo juntos. Estas interacciones fueron esenciales para hacerlos una comunidad de práctica, a pesar de que pintaban solos.
- Práctica: Los miembros no sólo comparten intereses en común como un grupo de personas que les gustan cierto tipo de películas. Desarrollan un repertorio de recursos: experiencias, anécdotas, herramientas, maneras de resolver problemas frecuentes: una práctica compartida.

2.1.4. Interacción Humano-Computadora: Computación UbiCua

La interacción utilizando computación ubicua será la forma de interactuar con el mundo físico. Esto nos ha llevado a una experiencia que resulta en un variedad de cambios importantes hacia la entrada, salida y las interacciones. Primero, la entrada ha removido lo explícito de interactuar por medio de teclados o texto hacia maneras más implícitas. Reuniendo información de los sensores.

Segundo, la integración de las capacidades de la computación ubicua también requiere nuevas tecnologías de salida y técnicas. No necesitamos más los monitores para mostrar la información. Dos importantes tendencias han emergido: coordinación entre múltiples dispositivos de visualización y demanda de menos atención. Estas tendencias han llevado a explorar una nueva clase de dispositivos: llamados ambiente. El ambiente hace que se necesite menos esfuerzo mental. Estos dispositivos no tiene que ser visuales, también pueden ser motrices.

Tercero, un importante factor de la Computación UbiCua es que intenta unir los artefactos computacionales con los artefactos del mundo real. Sobreponiendo la información digital en el mundo real, así es producida la realidad aumentada.

Para entender la interacción es necesario abordar a lo que hacen lo diseñaros lo cual es tomar los modelos que consideran la naturaleza de la relación entre el proceso cognitivo y el mundo exterior. El enfoque en el diseño está tratando de encontrar el balance en esta relación, usando principalmente tres teorías:

Teoría de la actividad: considera los objetivos, metas y operaciones del mundo real. Ambos acciones y objetivos fluyen en el estado físico del mundo en lugar de estar fijos, planes a priori. Debido al cambio en circunstancias, una operación requiere más atención de lo normal. También se hace énfasis en las propiedades de los artefactos que pueden contener implícitamente conocimiento y tradiciones.

Acciones situadas y cognición distribuida: las AC enfatizan en los aspectos de mejora del comportamiento humano y des-enfatiza planes a priori ejecutados por un solo individuo. La CD des-enfatiza en la cognición individual, transforma la perspectiva a un sistema donde los humanos son solo parte de algo más grande.

Entendimiento de la práctica humana: Recolectar las costumbres de una población para el desarrollo de nuevos productos en la era digital. [15]

2.2. Realidad aumentada

2.2.1. Definición

La realidad aumentada (RA) es el término que se usa para definir la visión de un entorno físico del mundo real, a través de un dispositivo tecnológico, es decir, los elementos físicos tangibles se combinan con elementos virtuales, logrando de esta manera crear una realidad aumentada en tiempo real. Consiste en un conjunto de dispositivos que añaden información virtual a la información física ya existente, es decir, añadir una parte sintética virtual a la real. [16]

La realidad aumentada es diferente de la realidad virtual porque sobre la realidad material del mundo físico monta una realidad visual generada por la tecnología, en la que el usuario percibe una mezcla de las dos realidades; en cambio, en la realidad virtual el usuario se aísla de la realidad material del mundo físico para sumergirse en un escenario o entorno totalmente virtual. [16]

2.2.2. Ejemplos

1. Victoria Station

En Marzo de 2011 la empresa Axe (Lynx en Inglaterra) situó una pantalla inmensa en la estación Victoria de Londres en la que los transeúntes podían interactuar mediante RA con ángeles(Los cuales eran parte de la publicidad de la compañía) cuando caminaban alrededor de la estación. [1]



FIGURA 2.1: Campaña publicitaria usando RA en la Estación Victoria de Londres [1]

2. National Geographic

Desde Noviembre de 2011 National Geographic ha coordinado una campaña con centros comerciales en Europa por la que mediante realidad aumentada y pantallas las personas que se encuentran en el centro comercial pueden interactuar con animales y dinosaurios a su alrededor . [2]



FIGURA 2.2: Promoción de documental de National Geographic usando RA [2]

3. Roma Aumentada

Por medio del uso de realidad aumentada la región italiana de Lazio se encuentra trabajando en una propuesta para incrementar el turismo en la región especialmente en la ciudad de Roma por medio de una aplicación la cual muestra información relevante de los monumentos de la ciudad y de la región. [3]



FIGURA 2.3: Prototipo para mejorar el turismo en la ciudad de Roma usando RA [3]

4. Lego Digital Box

En algunas de sus tiendas en Estados Unidos Lego a implementado una nueva herramienta para poder interactuar con su productos sin tener que sacar los juegos de piezas fuera de la caja, con tan solo escanear un código QR los visitantes de la tienda pueden interactuar con el producto armado antes de comprarlo en la tienda. [4]



FIGURA 2.4: Herramienta para visualizar productos de Lego en sus tiendas [4]

5. Stiktu

Stiktu es una aplicación de realidad aumentada para Iphone y Android por medio de la cual sus usuarios pueden escanear y dibujar lo que quieran al tomar una imagen de sus alrededores, además de eso incorpora una pequeña red social en la que la cual los usuarios pueden compartir sus creaciones y estas pueden ser valoradas por otros usuarios. [5]



FIGURA 2.5: Aplicación que hace uso de RA para poder dibujar en imágenes del mundo real [5]

6. PS VITA - Wide Area Augmented Reality (WAAR)

En 2011 Sony lanzó WAAR el cual fue una nueva adición para la consola Playstation Vita, usando la cámara de la consola y tarjetas, se podía reconocer una ‘área de juego’ para posteriormente poder jugar mediante la Playstation Vita. [6]



FIGURA 2.6: Periférico y aplicación para hacer que la Ps-Vita pueda hacer uso de RA [6]

7. Google Glass

Glass es un dispositivo de visualización de tipo óptico transparente desarrollado por Google el cual fue anunciado en 2013 y posteriormente cancelado en 2017. El propósito de Glass era mostrar información disponible para los usuarios de teléfonos inteligentes sin utilizar las manos, permitiendo también el acceso a Internet mediante órdenes de voz, de manera comparable a lo que Google Now ofrece en dispositivos Android.⁷ El proyecto Glass es parte de la sección Google X de la compañía, que ha trabajado en otras tecnologías futuristas, como vehículos autónomos. [7]

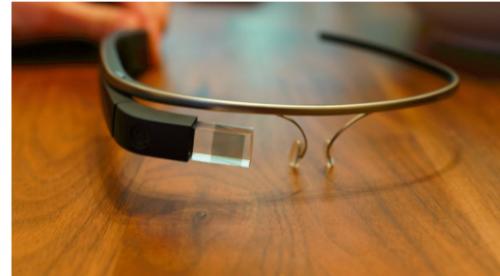


FIGURA 2.7: Periférico para poder interactuar con el mundo haciendo uso de RA [7]

8. Pokemon Go

Pokémon GO es un videojuego de realidad aumentada basado en geolocalización, desarrollado por Niantic, Inc. para dispositivos IOS y Android. Es un videojuego gratuito pero contiene micro-transacciones. El juego consiste en buscar y capturar personajes de la saga Pokémon escondidos en ubicaciones del mundo real y luchar con ellos, lo que implica desplazarse físicamente por las calles de la ciudad para progresar. La aplicación incorpora elementos de interacción social, ya que promueve reuniones físicas de los usuarios en distintas ubicaciones de sus poblaciones. Hasta la fecha es la aplicación de realidad aumentada con mayor popularidad en el planeta. [8]



FIGURA 2.8: Aplicación que hace uso de RA y geolocalización para poder interactuar con 'Pokemones' [8]

2.2.3. Visualización

El sentido de la vista es uno de los sentidos más desarrollados en los humanos, así mismo es el responsable de entregar 70 % de la información que el cerebro recibe acerca de nuestro ambiente, por consecuencia la realidad aumentada se ha centrado en proveer aumentación a las capacidades visuales de las personas. [9]

El campo de vista de un humano promedio es de 200° a 220° horizontalmente, dependiendo de la forma de la cabeza y la posición de los ojos de la persona, mientras que la fovea (Área donde los humanos tenemos agudeza/claridad visual) cubre solo 1° a 2° de la vista de una persona, los humanos corregimos este problema moviendo los ojos. Por lo que para tener una aplicación de realidad aumentada de buena calidad se requiere un dispositivo de visualización con suficiente resolución en el área de agudeza visual de las personas [9]

El ojo humano también puede adaptarse a diferentes entornos de luz ajustando el tamaño de la pupila, por lo que hay un rango de 10^{10} diferentes intensidades de luz que el ojo humano puede percibir, por lo que un aplicación de realidad aumentada versátil tiene que ser capaz de adaptarse a los diferentes cambios de luz. [9]

Un sistema de ideal de RA tiene que tener la capacidad de crear modelos de RA en 3D que puedan popular espacios físicos actuales, ya sean parecidas a la realidad o no, dependiendo de la aplicación.

2.2.3.1. Métodos de aumentación

De las propiedades de la visión humana y de los objetivos para llevar acabo un sistema de realidad aumentada se derivan varios requisitos para el medio de visualización que se utilizará en el sistema de realidad aumentada y los cuales hacen la diferencia entre dispositivos de visualización de imágenes generadas por computadora típicos y uno adecuado para utilizar realidad aumentada. En la actualidad podemos derivar 3 tipos de dispositivos que pueden ser utilizados para este fin. Óptico transparente, vídeo transparente y basado en proyecciones. [9]

1. óptico transparente(OT)

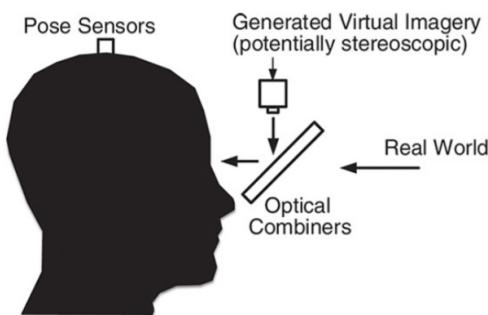


FIGURA 2.9: OT utiliza un elemento óptico(Ej. un espejo) para combinar la visión del usuario con imágenes generadas por computadora [9]

Este tipo de dispositivo utiliza principalmente un elemento parcialmente transmisor y parcialmente reflejante para poder conseguir mezclar los elementos virtuales con la realidad. Por ejemplo un espejo semi-transparente o plateado cumple con este objetivo. El espejo deja pasar la suficiente cantidad de luz del mundo real pasar, de tal manera que podamos seguir observando la realidad directamente y al mismo tiempo imágenes computarizadas son superpuestas en el espejo efectivamente combinando el mundo real con el virtual.

2. vídeo transparente (VT)

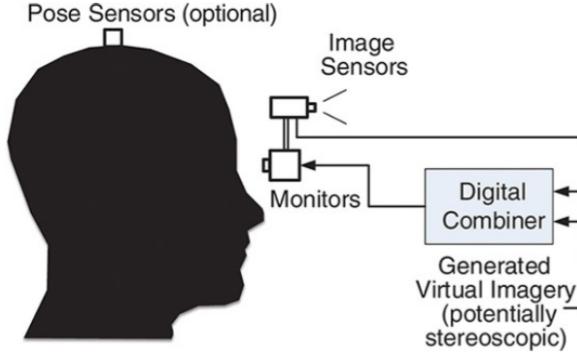


FIGURA 2.10: VT captura el mundo real a través de una cámara, modifica la imagen tomada añadiendo modelos de RA y posteriormente se la muestra al usuario en algún tipo de monitor [9]

El tipo vídeo transparente utiliza una cámara para obtener las imágenes del mundo real, después sobre un buffer de vídeo se agregan los modelos de RA, súperponiéndolos a las imágenes reales para después mostrar todo junto en un display común.

3. proyecciones

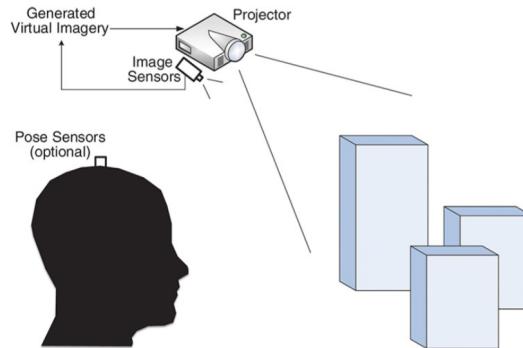


FIGURA 2.11: Modelos de RA proyectadas sobre una superficie [9]

A diferencia de los tipos anteriores, usar una proyección no requiere de un display o combinación especial para observar los modelos de RA, en este caso los modelos de RA son directamente proyectados sobre los objetos en el mundo real.

2.2.4. Seguimiento

El seguimiento se refiere a la búsqueda dinámica de propiedades espaciales de un objeto en tiempo de ejecución. Comúnmente, en el contexto de realidad aumentada, el seguimiento de una entidad significa que la posición y orientación de la entidad se miden continuamente. Se pueden rastrear diferentes entidades: la cabeza, los ojos o las extremidades de un usuario, por ejemplo, o algún dispositivo de realidad aumentada como una cámara o pantalla, o cualquier objeto que este poblando nuestra escena de RA. [9]

En el campo de la Realidad Aumentada, hay dos principales modos distintos de seguimiento, conocidos como seguimiento de marcador y sin marcador.

2.2.4.1. Seguimiento de marcador

Un marcador en el contexto de realidad aumentada se refiere a lo que generalmente se conoce como marcador fiducial. Un marcador fiducial es cualquier objeto que se puede colocar en una escena para proporcionar un punto fijo de referencia de posición o escala. En realidad aumentada, estos marcadores pueden proporcionar una interfaz entre el mundo físico y el contenido de realidad aumentada, como modelos 3D o videos. En su núcleo, estos marcadores permiten que el dispositivo que genera el contenido de RA calcule la posición y orientación (conocidas colectivamente como pose) de su cámara. Cuando este cálculo se realiza en tiempo real, este proceso se conoce como seguimiento de marcador. [9]

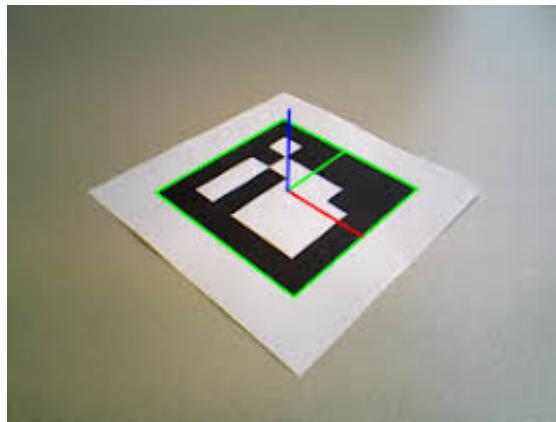


FIGURA 2.12: Ejemplo de un marcador fiducial [9]

El seguimiento de marcadores se puede lograr utilizando solo información visual. En la práctica, esto significa que el seguimiento de marcador se puede lograr con solo una cámara y una computadora; no se necesitan sensores adicionales, como un giroscopio. En contraste, ciertas soluciones sin marcador requieren información de un giroscopio. [9]

Para realizar el cálculo con marcador primero hay que obtener la pose de la cámara lo cual se logra a través de un proceso de dos pasos. Primero se detecta y se confirma un marcador. Luego, usando esta información, se calcula la pose de la cámara.

La detección del marcador ocurre en algunas etapas. En primer lugar, para facilitar la detección, las imágenes de la cámara se preprocesan. Un elemento importante en este paso es convertir la imagen de la cámara a escala de grises. Esto se hace para el aumento comparativo en la velocidad de procesamiento de las imágenes en escala de grises sobre

las imágenes en color ya que por lo general los marcadores solo utilizan blanco y negro por lo que el resto de los colores no son necesarios. [9]

Se necesitan cuatro puntos fijos para determinar la pose de la cámara. Primero tomamos los datos de la imagen de la cámara y calculamos los puntos característicos de todos los marcadores potenciales. Luego, podemos comparar estos marcadores potenciales con los datos pre-programados (El seguimiento por marcador tiene pre-programados todos los posibles marcadores) y verificar si un marcador potencial coincide con alguno de los marcadores que el dispositivo ha estado buscando. Una vez que se ha confirmado un marcador, se utilizan las coordenadas de sus puntos característicos para calcular matemáticamente la posición de la cámara. [9]

2.2.4.2. Seguimiento sin marcadores

En el sentido más general, el seguimiento sin marcadores se utiliza para denotar aquellos sistemas o aplicaciones de Realidad Aumentada, que no requieren ningún preconocimiento del entorno del usuario para colocar el contenido virtual o modelos de RA en la escena 3D. Esto contrasta con el seguimiento con marcador el cual es más frecuente. [9]

El seguimiento sin marcadores permite conocer el campo de visión y la perspectiva del usuario, permitiendo que el entorno virtual o la escena de realidad aumentada reaccione con modelos de RA de acuerdo con objetos reales. Para una experiencia completa de seguimiento de movimiento, el sistema de seguimiento necesita medir el movimiento en seis grados de libertad. [9]

Si bien los métodos de seguimiento de movimiento basados en marcadores utilizan marcadores fiduciales específicos, el seguimiento sin marcadores no los requiere, por lo que es un método más flexible. También evita la necesidad de un entorno preparado en el que se coloquen marcadores de referencia de antemano, por ejemplo. Contrariamente al seguimiento con marcadores, un enfoque sin marcadores permite al usuario caminar libremente en una habitación o un nuevo entorno y aún así recibir retroalimentación posicional, expandiendo el rango de aplicabilidad. [9]

El seguimiento sin marcadores solo utiliza lo que los sensores del dispositivo pueden observar en el entorno para calcular la posición y la orientación de la cámara. El método depende de características naturales en lugar de marcadores específicos, y puede usar un enfoque basado en modelos o procesamiento de imágenes para detectar características que proporcionan datos para determinar la posición y la orientación. [9]

Para aplicaciones de realidad aumentada, se puede usar un enfoque basado en modelos para determinar la ubicación de los objetos virtuales con los reales. El modelo del objeto

real se puede codificar como un diseño asistido por computadora (modelo CAD). El seguimiento de la realidad aumentada sin marcadores, por lo tanto, buscaría y compararía continuamente la imagen que recibe con el modelo 3D conocido. [9]

Si bien el seguimiento sin marcador es una tecnología que se espera que mejore la aplicación de realidad virtual y realidad aumentada, especialmente las aplicaciones móviles, las limitaciones tecnológicas actuales aún requieren una compensación entre precisión y eficiencia.

2.3. Geolocalización

La geolocalización es la capacidad para obtener la ubicación geográfica real de un objeto, como un radar, un teléfono móvil o un ordenador conectado a Internet. La geolocalización puede referirse a la consulta de la ubicación, o bien para la consulta real de la ubicación. El término geolocalización está estrechamente relacionado con el uso de sistemas de posicionamiento, pero puede distinguirse de estos por un mayor énfasis en la determinación de una posición significativa (por ejemplo, una dirección de una calle) y no sólo por un conjunto de coordenadas geográficas. [17]

En el caso de los teléfonos, la geolocalización puede ocurrir ya sea a través de la multilateración de señales de radio entre (varias) torres de telefonía de la red y el teléfono, a través de GPS o GSM. Para ubicar un teléfono móvil utilizando la multilateración de señales de radio, se debe emitir al menos la señal de roaming para contactar a la próxima torre de antena cercana, pero el proceso no requiere una llamada activa. GSM se basa en la intensidad de la señal del teléfono a las torres de antena cercanas. [17] [18]

El posicionamiento móvil puede incluir servicios basados en la ubicación que revelen las coordenadas reales de un teléfono móvil, que es una tecnología utilizada por las compañías de telecomunicaciones u otras compañías, para aproximarse a la ubicación de un teléfono móvil, y por lo tanto también a su usuario. [17]

2.4. Frameworks

2.4.1. ARCore

Es una plataforma para el desarrollo de aplicaciones de realidad aumentada en Android. ARCore utiliza tres tecnologías clave para integrar el contenido virtual con el mundo real a través de la cámara del teléfono:

- Rastreo de movimiento
- Entendimiento del entorno
- Estimación de la luz

¿Cómo funciona?

Fundamentalmente, hace dos cosas: rastrear posición del teléfono mientras éste se mueve, y construir su propio entendimiento del mundo del mundo real.

ARCore utiliza conceptos conocidos ampliamente en la comunidad de la RA para poder brindar un framework que sirva como una herramienta para múltiples usos. Se juntan estos conceptos para hacer que los objetos realmente aparezcan en las superficies. [13]

Seguimiento: Detectar visualmente distintas características en la imagen capturada desde la cámara llamados **puntos resaltados** y utiliza estos puntos para procesar el cambio en su localización. la información visual se combina con las medidas iniciales del dispositivo IMU para estimar su pose (posición y orientación) de la cámara relativa al mundo a lo largo del tiempo.

Comprensión del entorno: Se buscan pedazos o trozos del entorno que aparecen sobre superficies como escritorios o mesas, y hace que estén disponibles como **planos**.

Estimación de la luz: Saber la intensidad de la luz para poder darle más realismo a los objetos que se muestran.

Interacción con el usuario: Cada gesto que el usuario realice con el teléfono es detectado como un evento. Esto permite a los usuarios interactuar con los objetos.

Puntos orientados: Permiten colocar objetos sobre superficies no-horizontales. ARCore calcula el ángulo de inclinación y regresa una pose para que sea más realista.

Anclajes y rastreables: Se pueden definir anclajes para asegurar que se dará seguimiento a la posición del objeto a lo largo del tiempo. Los puntos y los planos son un tipo de objeto llamado rastreable.

2.4.2. ARKit

Es un kit de desarrollo liberado recientemente (2017) por Apple el cual sirve para crear experiencias de realidad aumentada utilizando la cámara del dispositivo y un sistema de rastreo.

En todas las experiencias de realidad aumentada, ARKit utiliza el mundo real y la cámara para coordinar los sistemas siguiendo una convención diestra: los puntos del eje-Y hacia arriba y, cuando es relevante, los puntos de eje-Z hacia el espectador y los puntos del eje-X hacia la derecha del espectador.

Las configuraciones de la sesión pueden cambiar el origen y la orientación del sistema de coordinación con respecto al mundo real. Cada anclaje en una sesión de realidad aumentada define su propio sistema de coordinación, también siguiendo la convención diestra, z-hacia-espectador; por ejemplo, la clase ARFaceAnchor define un sistema para localizar las características faciales.

Para crear una relación entre los espacios reales y virtuales, ARKit usa una técnica llamada odometría visual-inercial. Este proceso combina información del hardware de detección de movimiento con análisis de visión por computadora de la escena visible a la cámara del teléfono. ARKit reconoce características notable en la escena de la imagen, rastrea las diferencias en las posiciones de aquellas características a través de los frames de vídeo, y comprar esa información con los datos de movimiento. El resultado es un modelo altamente preciso de la posición y movimiento del dispositivo.

El rastreo del mundo también analiza y entiende los contenidos de una escena. Usa métodos de hit-testing para encontrar las superficies del mundo real correspondientes a un punto en la imagen de la cámara. Si se habilita el ajuste de detección de plano en tu sesión de configuración. ARKit detecta las superficies planas en la imagen de la cámara y reporta su tamaño y posición. Se puede utilizar los resultados del hit-test o detectar planos para posicionar o interactuar con el contenido virtual en la escena. [19]

2.4.3. Comparación

Framework	Sistemas Operativos Disponibles	Licencia	Soporte
ARCore	Android	Abierta	Sí
ARKit	iOS	Abierta	Sí

2.5. Servicios en la nube

2.5.1. Azure

Microsoft Azure (anteriormente Windows Azure) es un servicio de cómputo en la nube creado por Microsoft para crear, probar, implementar y administrar aplicaciones y

servicios a través de una red global de centros de datos administrados por Microsoft. Proporciona software como servicio , plataforma como servicio e infraestructura como servicio y admite muchos lenguajes de programación, herramientas y marcos de trabajo diferentes, incluidos software y sistemas específicos de Microsoft y de terceros. [20]

Microsoft Azure utiliza un sistema operativo especializado, llamado Microsoft Azure, para ejecutar su "capa de tejido": un cluster alojado en los centros de datos de Microsoft que administra los recursos informáticos y de almacenamiento de las computadoras y apropria los recursos (o un subconjunto de ellos) a las aplicaciones que se ejecutan en la parte superior de Microsoft Azure. [20]

La escalabilidad y la fiabilidad están controladas por Microsoft Azure Fabric Controller, que garantiza que los servicios y el entorno no fallen si uno o más de los servidores fallan dentro del centro de datos de Microsoft y que también proporciona la administración de la aplicación web del usuario como la asignación de memoria y el equilibrio de carga [21]

Azure proporciona una API basada en REST, HTTP y XML que permite a los desarrolladores interactuar con los servicios provistos por Microsoft Azure. Microsoft también proporciona una biblioteca de clase administrada del lado del cliente que encapsula las funciones de interacción con los servicios. También se integra con Microsoft Visual Studio, Git y Eclipse. [20] [21]

2.5.2. Amazon Web Services

Amazon Web Services es una subsidiaria de Amazon.com que proporciona plataformas de computación en la nube a individuos, compañías y gobiernos, en base a una suscripción de pago. La tecnología permite a los suscriptores tener a su disposición un clúster virtual de computadoras completo, disponible todo el tiempo, a través de Internet. La versión de AWS de las computadoras virtuales tiene la mayoría de los atributos de una computadora real, incluido el hardware (CPU y GPU para procesamiento, memoria local / RAM, almacenamiento en disco duro / SSD); una elección de sistemas operativos; redes; y software de aplicación precargado como servidores web, bases de datos, etc. [22]

Cada sistema AWS también virtualiza su entrada y salida de consola (teclado, pantalla y mouse), permitiendo a los suscriptores de AWS conectarse a su sistema AWS usando un navegador moderno. El navegador actúa como una ventana en la computadora virtual, lo que permite a los suscriptores iniciar sesión, configurar y usar sus sistemas virtuales tal como lo harían con una computadora física real. Se puede optar por implementar sus sistemas AWS para proporcionar servicios basados en Internet para su propio beneficio y el de sus clientes. [22]

La tecnología de AWS se implementa en granjas de servidores en todo el mundo y es mantenida por la filial de Amazon. El costo de la suscripción se basa en una combinación de uso, las características de hardware / SO / software / red elegidas por el suscriptor, disponibilidad requerida, redundancia, seguridad y opciones de servicio. Según lo que el suscriptor necesita y paga, puede reservar una única computadora virtual AWS, un grupo de computadoras virtuales, una computadora física (real) dedicada para su uso exclusivo o incluso un grupo de computadoras físicas dedicadas. Como parte del acuerdo de suscripción, Amazon gestiona, actualiza y proporciona seguridad estándar de la industria para el sistema de cada suscriptor. [22] [23]

2.5.3. Google Cloud Platform

Google Cloud Platform, ofrecida por Google, es un conjunto de servicios de computación en la nube que se ejecuta en la misma infraestructura que Google usa internamente para sus productos de usuario final, como la Búsqueda de Google y YouTube. También incluye un conjunto de herramientas de gestión y proporciona una serie de servicios modulares en la nube que incluyen procesamiento, almacenamiento de datos, análisis de datos y aprendizaje automático. El registro requiere una tarjeta de crédito o detalles de la cuenta bancaria. [24] [25]

2.5.4. Comparación

Servicio	Costo Aproximado	Escalabilidad
Azure	750 MXN por mes	Si
Amazon Web Services	850 MXN por mes	Si
Google Cloud	700 MXN por mes	Si

Dentro de esta comparación se buscaron los servicios básicos que requeriría la aplicación, como lo son:

- Tiempo de máquina virtual para el servidor.
- Gestor de base de datos y espacio en disco.
- Servicio de Telemetría

En este caso Google Cloud presenta los servicios mas baratos, pero las funcionalidades de sus servicios son limitadas a comparación de Azure o AWS por lo que se ha seleccionado Azure, que presenta mejor funcionalidad y a un costo mas reducido que AWS. [26] [27] [28]

2.6. Gestión de datos

2.6.1. Hadoop

Hadoop es una colección de utilidades de software de código abierto que facilita el uso de una red de muchas computadoras para resolver problemas que involucran cantidades masivas de datos y computación. Proporciona un framework de software para el almacenamiento distribuido y el procesamiento de big data utilizando el modelo de programación MapReduce. Originalmente diseñado para clusters de computadoras construidos desde hardware básico de uso común, también es utilizado en clusters de hardware de alta gama. Todos los módulos en Hadoop están diseñados con una suposición fundamental de que las fallas de hardware son ocurrencias comunes y deben ser manejadas automáticamente por el framework.

Microsoft presenta una opción por medio de Azure para desplegar un cluster de Hadoop, Azure HDInsight. Azure HDInsight es un servicio en la nube totalmente administrado que hace que procesar grandes cantidades de datos sea fácil, rápido y rentable. HDInsight utiliza frameworks populares de código abierto como Hadoop, Spark, Hive, LLAP, Kafka, Storm, etc. Azure HDInsight permite una amplia gama de escenarios como por ejemplo almacenamiento y procesamiento de datos a gran escala y machine learning.

2.6.1.1. Hive

Hive es una herramienta para interactuar con la infraestructura de almacenamiento de datos y el procesamiento de datos estructurados en Hadoop. La sintaxis de Hive es similar a la de SQL.

```
SELECT [ALL | DISTINCT] select_expr, select_expr, ...
FROM table_reference
[WHERE where_condition]
[GROUP BY col_list]
[HAVING having_condition]
[ORDER BY col_list]]
[LIMIT number];
```

CÓDIGO 2.1: Ejemplo de un script de Hive

2.6.2. SQL Server

Microsoft SQL Server es un sistema de administración de bases de datos relacionales desarrollado por Microsoft. Como servidor de base de datos, es un producto de software

con la función principal de almacenar y recuperar datos según lo soliciten otras aplicaciones de software, que pueden ejecutarse en la misma computadora o en otra computadora a través de una red (incluido Internet). Se puede interactuar con las bases de datos de SQL server a través de T-SQL

2.6.2.1. T-SQL

Transact-SQL (T-SQL) es la extensión patentada de Microsoft para SQL. T-SQL amplía el estándar SQL para incluir programación de procedimientos, variables locales, varias funciones de soporte para procesamiento de cadenas, procesamiento de fechas, matemáticas, etc. Transact-SQL es fundamental para usar Microsoft SQL Server. Todas las aplicaciones que se comunican con una instancia de SQL Server lo hacen mediante el envío de instrucciones de Transact-SQL al servidor, independientemente de la interfaz de usuario de la aplicación. [29]

```

DECLARE @var1 NVARCHAR(30)
SET @var1 = 'Some Name'
SELECT @var1 = Name
    FROM Sales.Store
    WHERE CustomerID = 100

DELETE u
    FROM users AS u
    INNER JOIN user_flags AS f
        ON u.id = f.id
    WHERE f.name = 'idle'

```

CÓDIGO 2.2: Ejemplo de un script de T-SQL

2.6.3. SQLite

SQLite es un sistema de gestión de bases de datos relacionales contenido en una biblioteca de programación C. A diferencia de muchos otros sistemas de administración de bases de datos, SQLite no es un motor de base de datos de cliente-servidor. Por el contrario, está integrado en el programa final.

SQLite es una opción popular como software de base de datos incrustado para el almacenamiento local / cliente en aplicaciones de software como navegadores web. Podría decirse que es el motor de base de datos más utilizado, ya que es utilizado hoy en día por varios navegadores generalizados, sistemas operativos y sistemas integrados (como teléfonos móviles), entre otros. [30]

2.7. Telemetría

La telemetría es un proceso automatizado de comunicación mediante el cual las mediciones y otros datos se recopilan en puntos remotos y se transmiten a equipos receptores para su monitoreo.

Aunque el término comúnmente se refiere a mecanismos de transferencia de datos inalámbricos (por ejemplo, usando sistemas de radio, ultrasonidos o infrarrojos), también abarca datos transferidos a otros medios tales como redes de computadoras. [31] [32]

En software, la telemetría se usa para recopilar datos sobre el uso de aplicaciones y componentes de aplicaciones, por ejemplo con qué frecuencia se utilizan ciertas características, mediciones del tiempo de inicio y el tiempo de procesamiento, hardware, bloqueos de aplicaciones y estadísticas generales de uso. [32]

2.7.1. Application Insights

Application Insights es un servicio de monitoreo de rendimiento de aplicaciones para desarrolladores en múltiples plataformas. Se utiliza para monitorear aplicaciones en vivo así como detectar automáticamente anomalías de rendimiento. Incluye herramientas de análisis para ayudar a diagnosticar problemas y comprender lo que los usuarios realmente hacen dentro de la aplicación. [10]

Application Insights está diseñado para ayudar a mejorar continuamente el rendimiento y la usabilidad. Funciona para aplicaciones en una amplia variedad de plataformas, incluidas .NET, Node.js y J2EE, alojadas en físico o en la nube. De igual manera se puede utilizar para monitorear y analizar la telemetría desde aplicaciones móviles al integrarse con Visual Studio. [10]

Cabe recalcar que Application Insights es gratis en todas sus plataformas sin ningún límite de uso.

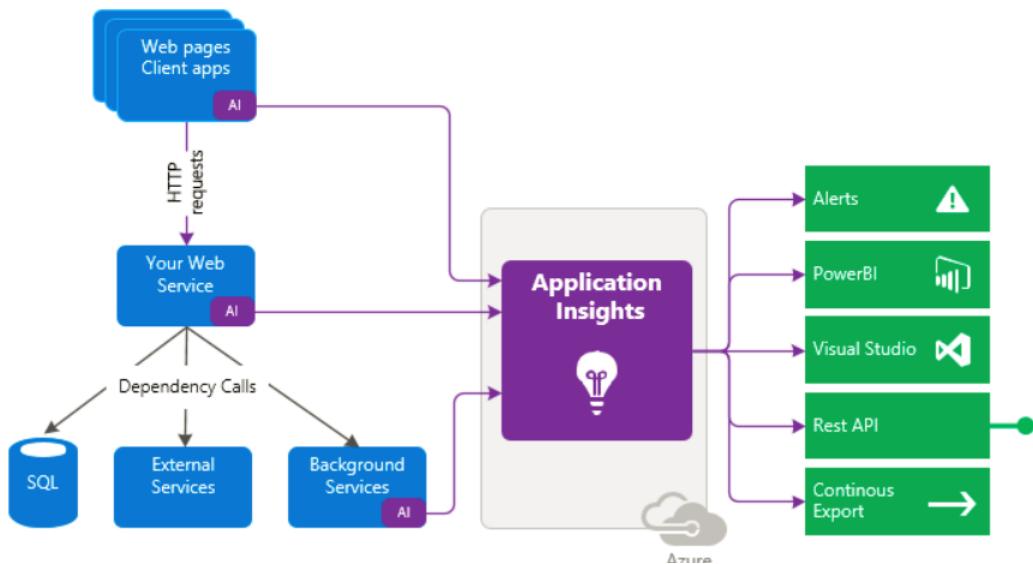


FIGURA 2.13: Flujo de datos de telemetría en Application Insights [10]

2.7.2. Google Analytics para aplicaciones móviles

Google Analytics es un servicio de análisis web de freemium ofrecido por Google que rastrea e informa el tráfico del sitio web. Google Analytics es ahora el servicio de análisis web más utilizado en Internet, también se ofrece en dos versiones adicionales: la suscripción Google Analytics 360, previamente Google Analytics Premium, dirigida a usuarios empresariales, y Google Analytics para aplicaciones móviles, un SDK que permite recolectar datos de uso de aplicaciones iOS y Android [33] [34]

El servicio es considerado freemium ya que un desarrollador puede hacer uso de él mientras se hagan menos de 10 millones de peticiones en un mes, además de ello toda la información capturada mediante Google Analytics podrá ser usada por Google. Mientras que el servicio empresarial remueve estas restricciones el costo de esta suscripción puede llegar a ser hasta 150000 dólares al año. [35]

2.7.3. Comparación

Servicio	Costo Aproximado	Límite de uso
Application Insights	Gratis	No
Google Analytics	Gratis	10 millones de peticiones

Debido a que Application Insights no recae en ningún modelo freemium o tiene un límite de uso, se ha considerado que es la mejor opción en cuanto a servicios de telemetría disponibles.

2.8. PowerBi

Power BI es un producto para el análisis y visualización de información o datos creado por Microsoft el cual proporciona visualizaciones interactivas para poder crear informes y dashboards de una manera eficiente.[\[11\]](#)

Power BI permite combinar datos desde diferentes servicios web, servidores y bases de datos lo que cual permite concentrar la visualización el análisis de datos en un punto común. [\[11\]](#) [\[36\]](#)

El costo de el servicio de Power Bi es gratis mientras se utilice hasta un 1GB de información por mes, en caso de requerir mas espacio el costo asciende a 10 dolares por mes sin limite de utilización de datos. [\[37\]](#)

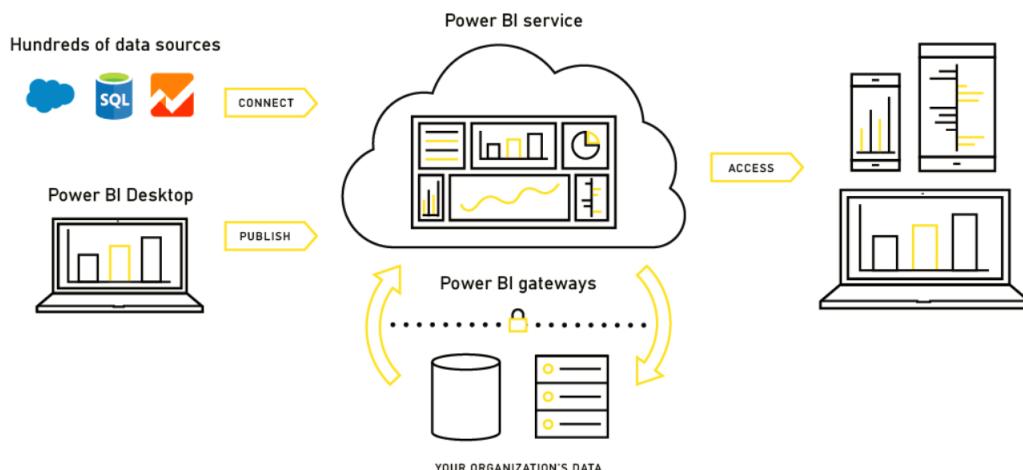


FIGURA 2.14: Ciclo de vida de los datos hasta ser utilizados por medio de PowerBi
[\[11\]](#)

2.9. Material Design

Material Design es un lenguaje de diseño desarrollado en 2014 por Google. Ampliando los motivos de "tarjeta" que se estrenaron en Google Now, Material Design hace un uso más liberal de diseños basados en cuadrículas, animaciones y transiciones receptivas, relleno y efectos de profundidad como iluminación y sombras. Material Design se puede

usar en todas las versiones compatibles de Android, o en API Nivel 21 (Android 5.0) y más recientes (o para versiones anteriores a través de la biblioteca v7 appcompat), que se utiliza en prácticamente todos los dispositivos Android fabricados después de 2009. Material Design se extenderá gradualmente a toda la gama de productos web y móviles de Google, proporcionando una experiencia consistente en todas las plataformas y aplicaciones. El objetivo principal de material design es la creación de un nuevo lenguaje visual que combine principios de buen diseño con innovación técnica y científica. [38]

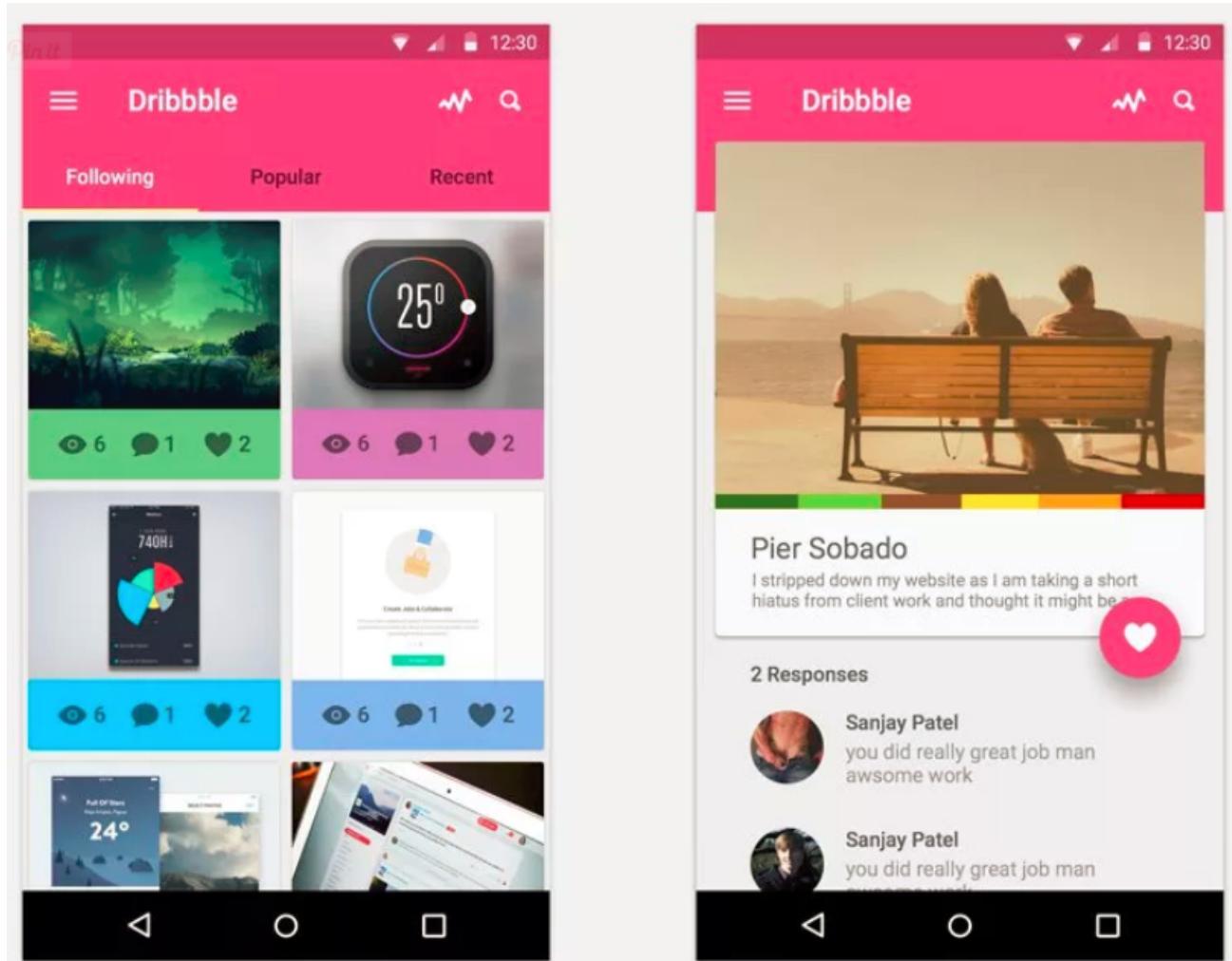


FIGURA 2.15: Ejemplo de una aplicación de tipo red social que utiliza Material Design



FIGURA 2.16: Ejemplo de una aplicación de calendario que utiliza Material Design

2.10. Lenguajes de programación

2.10.1. C#

C# es un lenguaje de programación multi-paradigma que abarca una tipificación fuerte, imperativa, declarativa, funcional, genérica, orientada a objetos (basada en clases) y disciplinas de programación orientadas a componentes. Fue desarrollado por Microsoft dentro de su iniciativa .NET y luego aprobado como estándar. C# es uno de los lenguajes de programación diseñados para Common Language Infrastructure, la cual es una especificación diseñada por Microsoft para que un código escrito en un lenguaje pueda ser ejecutado en diferentes plataformas o arquitecturas sin que tenga que ser cambiado. [39] [40] C# es un lenguaje de programación orientado a objetos de propósito general. La versión más reciente es C# 7.2, que se lanzó en 2017. [39]

2.10.1.1. Xamarin

Xamarin es una implementación multiplataforma de Common Language Infrastructure (CLI). Por lo que un código implementado en C# puede generar aplicaciones nativas de Android, IOS o Windows, lo cual permite un código unificado entre plataformas. [12]

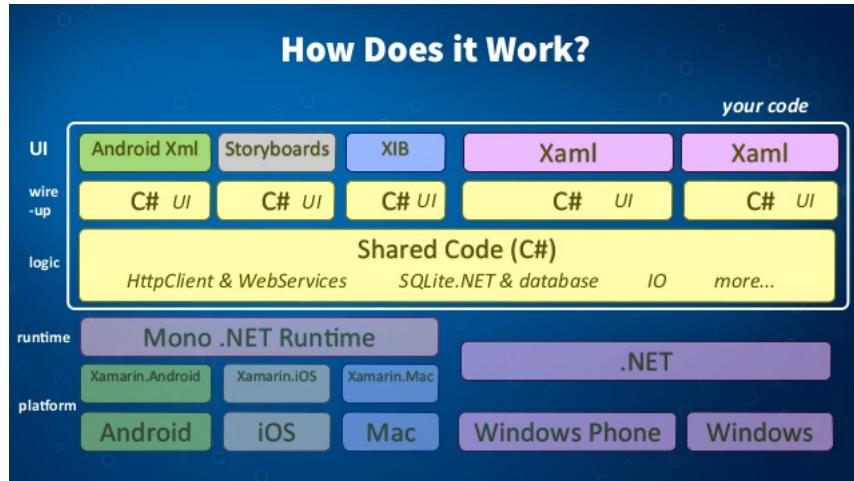


FIGURA 2.17: Ciclo de vida de una aplicación creada por medio de Xamarin [12]

```
using System;

using Android.App;
using Android.Content;
using Android.OS;

namespace Mono.Samples.HelloWorld
{
    [Activity (Label = "Hello World Demo", MainLauncher = true)]
    [IntentFilter(new[] { Intent.ActionView },
        Categories = new[] {
            Intent.CategoryDefault,
            Intent.CategoryBrowsable
        },
        DataScheme = "http",
        DataHost = "example.com"
    )]
    public class HelloAndroid : Activity
    {
        protected override void OnCreate (Bundle savedInstanceState)
        {
            base.OnCreate (savedInstanceState);

            SetContentView (Resource.Layout.main);
        }
    }
}
```

CÓDIGO 2.3: Hola mundo, utilizando Xamarin y C#

2.10.2. Java

Java es un lenguaje de programación de computadora de propósito general fuertemente tipado que es concurrente, basado en clases, orientado a objetos, y específicamente diseñado para tener la menor cantidad de dependencias de implementación posible. Su objetivo es permitir a los desarrolladores de aplicaciones escribir una vez, ejecutar en cualquier lugar-[41], lo que significa que el código compilado de Java se puede ejecutar en todas las plataformas que admitan Java sin la necesidad de recompilarlo. Las aplicaciones creadas en Java generalmente se compilan de tal manera que se puede ejecutar en cualquier máquina virtual Java (JVM) independientemente de la arquitectura de la computadora. [41] Java es un pilar clave en Android debido a que el SDK de Android usa el lenguaje Java como base para las aplicaciones del mismo, aunque una aplicación desarrollada por medio del SDK no puede ser ejecutada en cualquier JVM. [42] Android no proporciona la biblioteca estándar de Java SE completa, aunque el SDK de Android incluye una implementación independiente de un gran subconjunto de la misma. Es compatible con Java 6 y algunas características de Java 7. [42]

```
package com.example.helloworld;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

CÓDIGO 2.4: Hola mundo, utilizando el Android SDK y Java

Capítulo 3

Análisis Factibilidad

3.1. Análisis de mercado

El mercado de la realidad aumentada en la actualidad es muy pequeño comparado a otras industrias, esto debido a dos principales razones, el tipo de dispositivos que pueden hacer uso de realidad aumentada en la actualidad y el número de aplicaciones disponibles.

En el caso de los dispositivos disponibles, actualmente solo teléfonos de gama alta que hayan sido manufacturados desde el 2017 a la fecha pueden hacer uso de los principales frameworks de realidad aumentada, ARCore de Google para los teléfonos que hacen uso de Android como sistema operativo y ARkit de Apple para los teléfonos que hacen uso de IOS. [13] [19]

En el caso de IOS, actualmente solo el Iphone X es compatible con ARKit por lo que de todos los usuarios de IOS activos en el mundo a la fecha que son alrededor de 700 millones de usuarios [43] solo hay 30 millones usuarios que cuentan con un Iphone X, lo cual corresponde entre 4% y 5% de los usuarios totales. [44]

En el caso de Android la cantidad de modelos de teléfonos que puedan soportar ARCore es considerable, como se observa en la figura siguiente.

Name	Model
Asus Zenfone AR	ASUS_A002, JP_ASUS_A002, VZW_ASUS_A002A
Asus ZenFone Ares	ASUS_A002_2
Google Pixel 2	All models
Google Pixel 2 XL	All models
Google Pixel	All models
Google Pixel XL	All models
Huawei P20	EML-L29, EML-L09
Moto Z2 Force	hash
OnePlus 5	ONEPLUS A5000
Samsung A5	SM-A520F, SM-A520X, SM-A520F, SM-A520W, SM-A520K, SM-A520L, SM-A520S
Samsung A7	SM-A720F, SM-A720F, SM-A720X, SM-A720S
Samsung A8	SM-A530W, SM-A530X, SM-A530F, SM-A530N
Samsung A8+	SM-A730F, SM-A730X
Samsung Note8	SM-N9508, SM-N950F, SM-N950N, SM-N950U, SM-N950W, SM-SC01K, SM-SCV37
Samsung S7	SM-G930A, SM-G930F, SM-G930K, SM-G930L, SM-G930P, SM-G930R SM-G930S, SM-G930T, SM-G930U, SM-G930V, SM-G930W
Samsung S7 Edge	SM-G935A, SM-G935F, SM-G935K, SM-G935L, SM-G935P, SM-G935R, SM-G935S, SM-G935T, SM-G935U, SM-G935V, SM-G935W, SM-SCV33, SM-SC02H
Samsung S8	SM-G950U, SM-G950N, SM-G950F, SM-G950W, SM-SCV36, SM-SC02J
Samsung S8+	SM-G955F, SM-G955N, SM-G955U, SM-G955W, SM-SC03J, SM-SCV35

FIGURA 3.1: Modelos de teléfonos compatibles con ARCore [13]

De acuerdo a cifras de ventas al mes de Febrero de este año, se han vendido alrededor de 100 millones de dispositivos Android que son compatibles con ARCore [45], actualmente se estima que hay alrededor de 2 mil millones de usuarios de Android [46] por lo cual el porcentaje de usuarios de Android que puede hacer uso de ARCore es alrededor del 5% de los usuarios totales de Android.

Si bien los porcentajes de cada plataforma que pueden hacer uso de realidad aumentada, son pequeños, estos siguen representando a millones de usuarios potenciales solamente en la actualidad.

En cuanto a las aplicaciones de realidad aumentada disponibles actualmente, de acuerdo a datos de Apple hay cerca de 1000 aplicaciones que utilizan ARkit [47] mientras que solo hay 85 aplicaciones en la app store de Android que utilizan ARCore [48], por lo que las aplicaciones de realidad aumentada que utilizan ARKit a la fecha solamente representan el 0.047% de el número total de aplicaciones disponibles para IOS. mientras que en el caso de Android solo representan el 0.0022%, considerando que hay cerca de 2 millones de aplicaciones disponibles en IOS y 3 millones de aplicaciones disponibles para Android.

[49] [50]

La principal razón de estos números son las fechas en que las herramientas para poder realizar el desarrollo de aplicaciones de realidad aumentada fueron liberadas para los desarrolladores, en el caso de ARkit la versión 1.0 fue liberada en Septiembre del 2017 mientras que en el caso de ARCore fue hace apenas unos meses, en Febrero de 2018.

[13] [19]

Si bien en actualidad la industria de la realidad aumentada apenas esta comenzando a dar sus primeros pasos en cuanto a aplicaciones disponibles y dispositivos que pueden hacer uso de las mismas, no quiere decir que esto se mantendrá por siempre si no que se prevé que abarque una gran cantidad del mercado de usuarios de tecnología en un futuro no tan lejano como se puede observar en la siguiente gráfica.

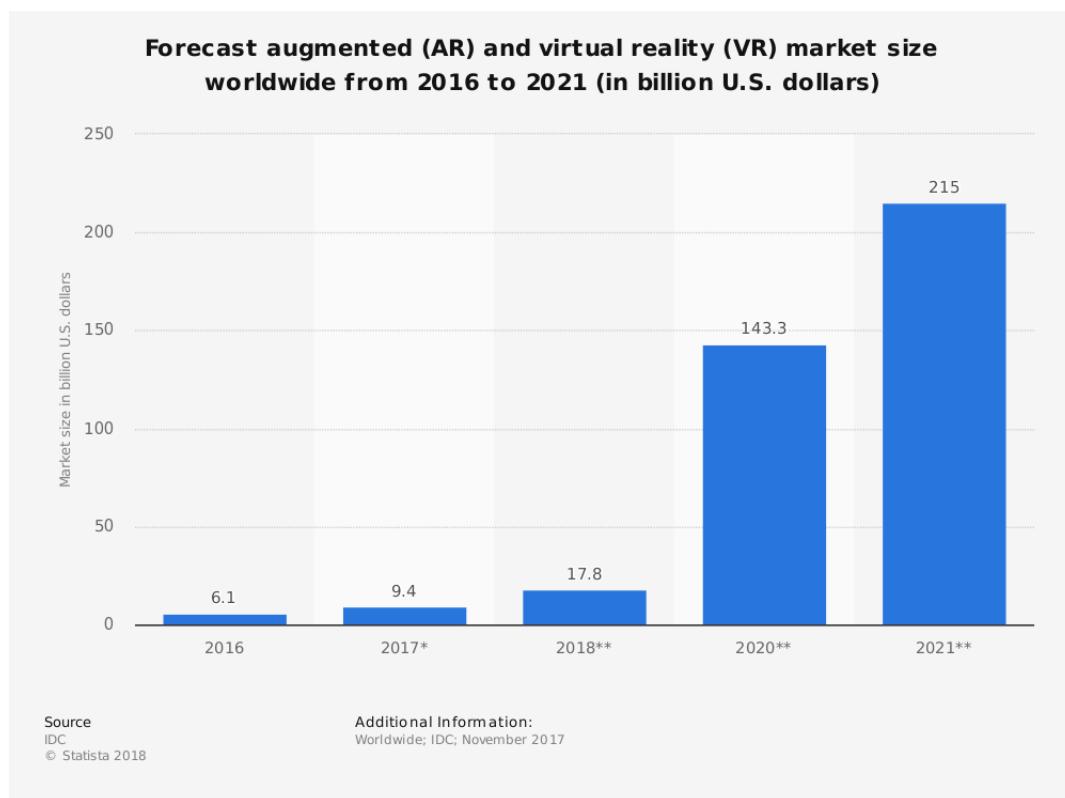


FIGURA 3.2: Pronostico de crecimiento de el mercado de RA en billones de dolares [14]

Se prevé que el mercado de la industria del cine y la televisión alcancé un mercado de 321 billones de dolares para 2021 [51], para el mismo año el mercado de la realidad aumentada alcanzara dos terceras partes del tamaño de este mercado, lo cual resulta una interesante comparación debido a que el mercado de la industria de la realidad aumentada a comenzado hace apenas unos años mientras que la industria del cine y de la televisión llevan décadas como una industria establecida. [14]

3.2. Viabilidad técnica

3.2.1. Usabilidad

1. Compatibilidad - Debido a que el desarrollo se ha centrado a dispositivos Android que puedan utilizar ARCore y utilicen alguna versión de Android superior a la 7.0 no habrá ningún problema de compatibilidad con el sistema propuesto ya que desde un principio se han delimitado los modelos compatibles debido al soporte de realidad aumentada.
2. Aspecto visual - Actualmente los estándares de Material Design son respetados por todas las versiones de Android superiores a la 5.0 [38] por lo que todas las interfaces serán congruentes para nuestros usuarios objetivo.
3. Navegación - La navegación de la aplicación se a centrado en ser minimalista, permitiendo navegar entre actividades mediante una sola acción, manteniendo una ruta clara para los usuarios.
4. Contenido - En cada actividad que el usuario puede realizar dentro de la aplicación se ha mantenido un contenido claro y conciso para que el usuario pueda llevar a cabo cada una de las actividades justo con lo que puede observar dentro de las interfaces.

3.2.2. Despliegue

El sistema cuenta con dos puntos de despliegue la Play Store de Google para la aplicación como tal y Azure para el despliegue de los servicios requeridos por el sistema.

Para realizar el despliegue de una aplicación hacia la Play Store es necesario contar con una licencia de desarrollo de Google, contar con iconos de la aplicación, capturas de pantalla y que la APK sea validada por los servicios de Google. [52] Durante la etapa de desarrollo se ha contemplado pagar la licencia de la Play Store y generar los gráficos necesarios para cubrir los requisitos de la misma para publicar la aplicación, en cuanto a la validación de la APK debido a que no se ha contemplado manejar transacciones monetarias o interactuar con los dispositivos de los usuarios mas allá de lo generado dentro de la aplicación, no se espera que haya ningún contratiempo en cuanto a la validación de la APK.

En cuanto a los servicios de Azure solo es necesario contar con una cuenta de desarrollo y créditos de Azure para contratar el espacio en la nube de los servicios, ya que se cuenta con una licencia de estudiante y créditos ilimitados para los servicios de Azure que son requeridos, el despliegue en Azure no presentara problema alguno.

Es importante mencionar que es posible realizar el despliegue hacia la Play Store y Azure desde el entorno de desarrollo integrado que ha sido seleccionado, en este caso Visual Studio, lo cual vuelve los puntos de despliegue hasta cierto punto homogéneos.

3.2.3. Soporte

Se ha contemplado que el sistema utilizara dos stacks de tecnología principalmente, Xamarin y Azure, y un lenguaje visual para el front end del sistema, Material Design.

Xamarin engloba las tecnologías móviles de la aplicación, esto incluye ARCore y los elementos de Android que serán utilizados para llevar a cabo esta sección del sistema, toda la documentación requerida se encuentra en la web oficial de Xamarin [12] o en su defecto en la web oficial de Google Developers [52]/

Las tecnologías y los servicios en la nube a utilizar pertenecen al stack de tecnologías de Azure, La documentación para todos los servicios de Azure se encuentra en la web de documentación principal de Microsoft [53].

Material design es un lenguaje propiedad de Google, toda su documentación y sus estándares pueden ser encontrados en su web oficial [38].

Toda la documentación de las tecnologías o lenguajes a utilizar es constantemente actualizada por su compañía propietaria, en este caso Microsoft o Google, lo cual garantiza un nivel de documentación aceptable y un constante mejoramiento de la misma en caso de requerirlo.

3.2.4. Características operacionales

Operacionalmente se espera que la aplicación esté disponible 99.99 % del tiempo, esto debido a que Azure garantiza que cada uno de nuestros servicios tendrá este nivel de disponibilidad, lo cual ha sido estipulado en los de contratos de cada uno de los servicios que se planean utilizar dentro de Azure.

- App Services: “*Garantizamos que las Aplicaciones que se ejecuten en una suscripción de cliente estarán disponibles durante el 99,95 % del tiempo.*” - **Contrato de servicio para los App Services** [54]

- Almacenamiento: “*Garantizamos que, como mínimo, durante el 99.99 % (99,9 % para Nivel de Acceso Esporádico) del tiempo procesaremos las solicitudes de lectura de datos de las Cuentas de Almacenamiento con Redundancia Geográfica con Acceso de Lectura (RA-GRS), siempre y cuando los intentos erróneos de lectura de datos de la región principal se vuelvan a intentar en la región secundaria.*”” - **Contrato de servicio para el almacenamiento** [55]

- SQL Server: “*Garantizamos que, como mínimo, durante el 99,99 % del tiempo los clientes dispondrán de conectividad entre los niveles Basic, Standard o Premium individuales o elásticos de su Base de Datos SQL de Microsoft Azure y nuestra puerta de enlace a Internet.*” - **Contrato de servicio para SQL Server** [56]
- Application Insights “*Garantizamos que la latencia de datos del servicio Application Insights no excederá las dos horas el 99,9 % del tiempo.*” - **Contrato de servicio para Application Insights** [57]

3.2.5. Escalabilidad

Una de las principales razones por las que se ha incluido el cómputo en la nube (Azure) dentro del diseño del sistema es para poder enfrentar los desafíos de escalabilidad de una manera mas eficiente y redituable, esto debido a que todos los servicios contemplados a utilizar dentro de Azure, pueden ser escalados dinámicamente dependiendo de las necesidades futuras del sistema. En términos generales el escalamiento de Azure funciona de dos maneras, escalamiento hacia afuera y escalamiento hacia arriba, el escalamiento hacia afuera instancia nuestros servicios en mas máquinas virtuales o servidores mientras que el escalamiento hacia arriba incrementa el tamaño de la instancia(memoria, procesamiento, etc..). [58]

3.3. Análisis financiero

3.3.1. Costo de tecnología

Servicio	Costo Aproximado (MXN)
Equipo de computo Thinkpad T550	16,000.00
Licencia de Visual Studio	Sin costo
OnePlus 5	10,500.00
SQL Server	Sin costo
Espacio en disco	500.00
Hosting	Sin costo
Application Insights	Sin costo
ARCore	Sin costo
PowerBi	Sin costo

CUADRO 3.1: Costos al desarrollar

Durante el desarrollo del sistema se ha considerado utilizar una licencia de estudiante de Azure lo cual vuelve inexistentes los costos de sus servicios por el momento.

Servicio	Costo Aproximado (MXN)
SQL Server	14,000.00 (por mes)
Espacio en disco	500.00 (Por Terabyte)
Hosting	1,000.00 (por mes)
Application Insights	Sin costo
PowerBi	200.00 (por mes)
Licencia de Play Store	500.00

CUADRO 3.2: Costos al publicar aplicación

Una vez que haya terminado la etapa de desarrollo y se considere que es pertinente liberar la aplicación será inválido utilizar una licencia de estudiante, por lo que los costos aumentaran considerablemente.

3.3.2. Roles requeridos

- Administrador de base de datos
 - Cualificaciones:
 - Conocimiento de T-SQL y Hive.
 - Experiencia usando SQL Server y Hadoop.
 - Experiencia en diseño y migración de bases de datos.
 - Experiencia en taxonomía de eventos para telemetría.
 - Experiencia en aplicaciones de herramientas de inteligencia de negocios.
 - Salario: 30,000.00 pesos al mes
 - Actividades a realizar
 - Diseño de bases de datos.
 - Manejo de herramientas de inteligencia de negocios.
 - Manejo de SQL-server y Hadoop.
 - Diseño de taxonomía de eventos para inteligencia de negocios.
- Desarrollador especializado en aplicaciones móviles
 - Cualificaciones:
 - Experiencia desarrollando aplicaciones utilizando Xamarin.
 - Conocimiento de Java(Android) y C#.
 - Experiencia utilizando ARCore.
 - Experiencia con GIT.
 - Conocimiento de Material Design.
 - Salario: 25,000.00 pesos al mes.
 - Actividades a realizar
 - Desarrollo de aplicación para móviles Android utilizando Xamarin.
 - Implementación de patrones de diseño.
 - Implementación del framework ARCore.
 - Desarrollo de documentación
- Arquitecto de software
 - Cualificaciones:
 - Experiencia en escalabilidad de sistemas.
 - Experiencia en diseño de arquitecturas de servidores y aplicaciones móviles.

- Experiencia en ciclo de vida del desarrollo de software.
- Experiencia en pruebas unitarias.
- Experiencia en integración y ensamblado de software.
- Salario: 40,000.00 pesos al mes
- Actividades a realizar
 - Diseño de arquitectura para aplicación móvil y servidor (Cliente-servidor)
 - Diseño e implementación de scripts para integración.
 - Despliegue e integración continua.
- Especialista en Cloud Computing
 - Cualificaciones:
 - Experiencia en servicios en la nube(Azure, AWS, GCL)
 - Experiencia en implementación y escalabilidad de servidores en la nube.
 - Experiencia en servicios de telemetría.
 - Experiencia construyendo data pipelines.
 - Salario: 35,000.00
 - Actividades a realizar
 - Implementación, despliegue y mantenimiento de servidores en la nube
 - Creación de pipeline de datos de telemetría hacia un sistema gestor de base de datos.
 - Orquestación de servicios en la nube.

3.3.3. Espacio a utilizar

Para realizar el sistema propuesto es requerido un espacio de trabajo con al menos las siguientes características:

- Ubicación: Ciudad de México.
- Tamaño: 5 metros cuadrados.
- Facilidades:
 - Escritorio.
 - Silla de oficina.
 - Internet de al menos 10mbps.
 - Electricidad.

Capítulo 4

Reporte Técnico

4.1. Metodología

Para el desarrollo de este trabajo terminal se seguirá la metodología de prototipos para poder generar un prototipo del sistema y probarlo en el menor tiempo posible, y también porque reduce el riesgo de que no se satisfagan todos los requisitos planteados debido a que es posible realizar pruebas constantemente. Además estas metodología nos permitirá seguir los siguientes pasos que se muestran a continuación:

- Recolectar y refinar los requisitos. Esta primera fase es de suma importancia para definir el sistema ya que en todo momento se pueden agregar y limpiar los requisitos, y es necesario que se pueda definir lo que puede ser posible y lo que no en todo momento.
- Diseño rápido. Es necesario bosquejar las propuestas de solución antes de ir directamente a la implementación para dejar más claro lo que se hará y no perder tiempo procesando de nuevo los requisitos.
- Construcción del prototipo. Utilizando las herramientas adecuadas se construirá de la mejor manera posible un prototipo que refleje lo que se haya diseñado previamente, cuidando los detalles y, agregando nuevos en caso de que se tenga que adaptar alguna tecnología.
- Evaluación del prototipo. Se realizarán pruebas de uso con los usuarios interesados que permitan saber qué tan lejos o cerca está el prototipo de ser viable, y si realmente es o no una opción para los usuarios.

- Refinamiento del prototipo. Una vez creado el prototipo se pondrá a prueba cada característica que sea fundamental para cumplir el objetivo de esta manera ir filtrando lo que sirve y lo que necesita cambios.
- Producto de ingeniería. Se repetirá cada vez que sea necesario la evaluación y el refinamiento para encontrar la mejor solución que cumpla con lo planeado desde un inicio.

4.2. Requerimientos funcionales

Los requerimientos funcionales de un sistema describen lo que el sistema debe hacer. Son declaraciones de los servicios que debe de proporcionar el sistema, de la manera en que ese debe de reaccionar a entradas particulares y como se debe de comportar en situaciones particulares.

RF1. Visualización de modelos de RA.

La herramienta mostrará los modelos que haya en el entorno haciendo uso de la geolocalización y la posición del teléfono.

RF2. Creación de modelos de RA.

El usuario podrá crear un modelo al seleccionar polígonos tridimensionales de una lista pre-definida.

RF3. Mapa.

A través de la herramienta se mostrara un mapa con los modelos de RA mas populares en un área de 100m a la redonda.

RF4. Interacción con modelos de RA.

Al hacer click en un modelo, se podrá interactuar con esta.

RF5. Notificaciones.

El usuario recibirá una notificación una vez que este cerca de un modelo popular.

4.3. Requerimientos no funcionales

Los requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste.

1. Escalabilidad El sistema sera escalado de acuerdo a las necesidades generadas por las demandas de los usuarios, como puede ser el uso de memoria y la velocidad de respuesta.

2. Usabilidad La aplicación sera intuitiva en el sentido de que los usuarios podrán hacer uso de la misma sin seguir ningún tipo de guía.
3. Disponibilidad El sistema estará disponible un 99.99 % del tiempo para los usuarios.
4. Compatibilidad La aplicación solo sera compatible con aquellos modelos de teléfonos que puedan hacer uso de ARCore.
5. Geolocalización La Geolocalización utilizada en la aplicación solo podrá ser tan precisa como el GPS del teléfono lo permita.

4.4. Reglas del negocio

1. Edición de modelos de RA. Una vez que una aumentación sea colocada no podrá editarse solo eliminarse.
2. Varios modelos de RA en una misma ubicación. En caso de que varias modelos tengan las mismas coordenadas, se irán rotando empezando por la mas popular y así sucesivamente.
3. Limites de visibilidad. La aplicación solo mostrará un modelo cuando el usuario este a 5 metros de distancia.
4. Eliminación de modelos de RA. Un modelo será eliminado si su creador lo elimina o si la popularidad de el mismo es baja.
5. Ubicación. El usuario solo podrá publicar sus modelos de RA en su ubicación geográfica correspondiente.

4.5. Arquitectura del sistema

4.5.1. Componentes

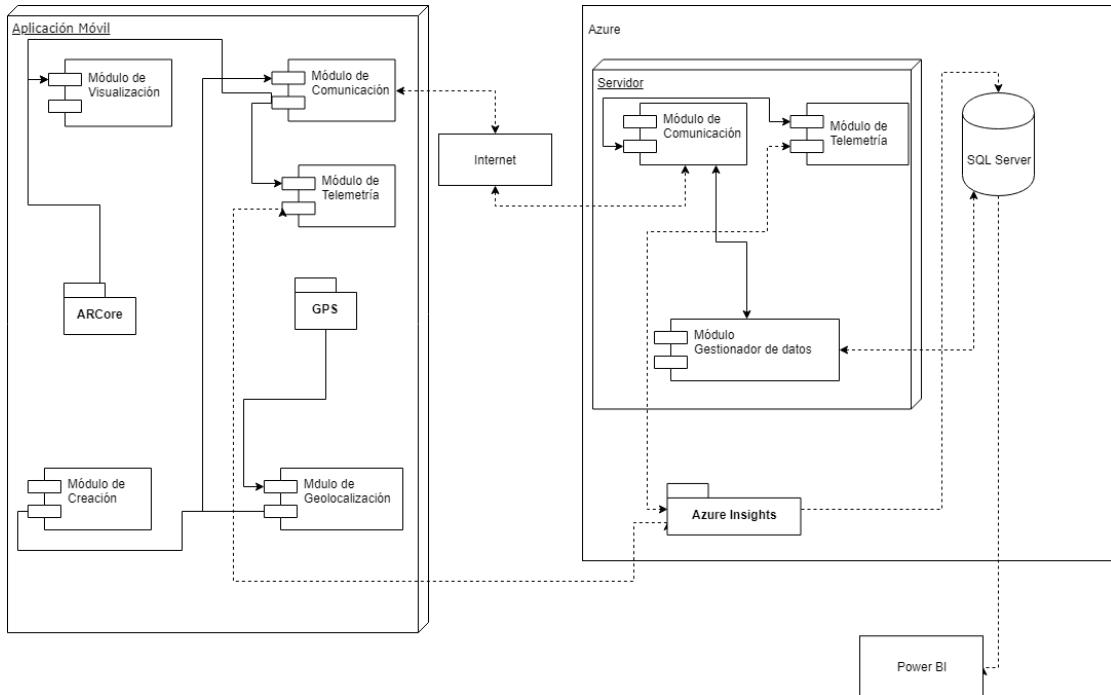


FIGURA 4.1: Diagrama de componentes de B-Side

4.5.1.1. Aplicación Móvil

1. Módulo de Visualización Su propósito controlara la parte visual de la aplicación, sera el encargado de mostrar las interfaces así como los modelos de RA.
2. Módulo de Geolocalización Su propósito es interactuar con el GPS del teléfono para obtener la ubicación actual del usuario.
3. Módulo de Comunicación Su propósito es gestionar las llamadas HTTP al servidor al igual que recibirlas.
4. Módulo de Telemetría La función de este módulo es obtener información acerca de lo que los usuarios están haciendo dentro de la aplicación al igual que las fallas que ocurran dentro de la misma.
5. Módulo de Creación Este módulo gestionara, la creación de modelos utilizando figuras predefinidas.

4.5.1.2. Componentes en Azure

1. Servidor

- a) Módulo de Gestión de Datos El propósito de este módulo sera obtener e insertar los datos obtenidos del cliente en el sistema gestionador de datos.
- b) Módulo de Telemetría Al igual que en el cliente, este módulo obtendrá información acerca del estado del servidor para alertar de fallas y hacer un reporte de uso.
- c) Módulo de Comunicación Por medio de este módulo se gestionaran las llamadas HTTP al servidor provenientes de los clientes al igual que las respuestas hacia los mismos.

4.6. Clases del sistema

4.6.1. Servidor

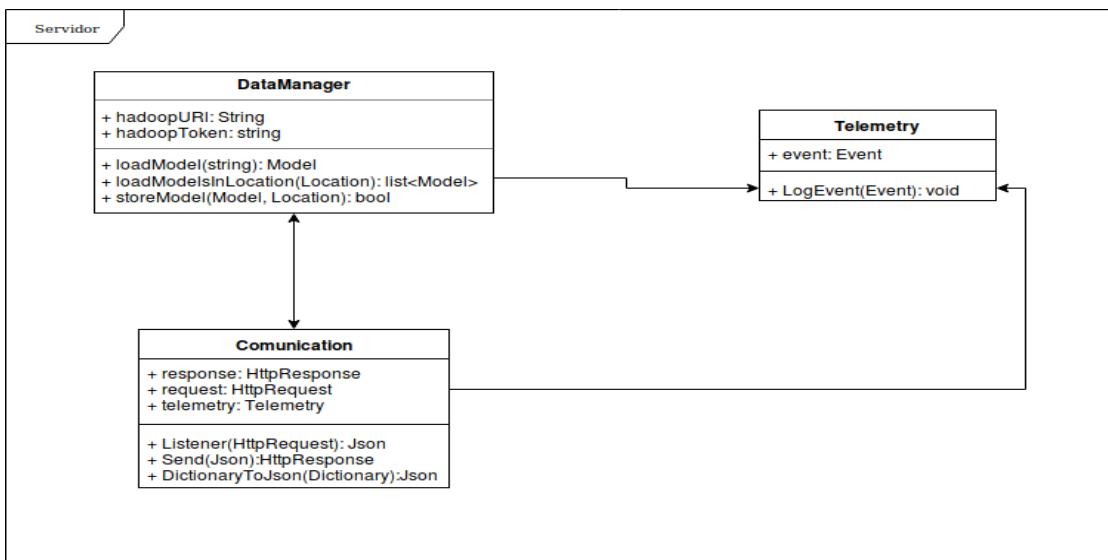


FIGURA 4.2: Diagrama de clases de el servidor

4.6.1.1. DataManager

- Propósito: Gestionar la escritura y la lectura de datos hacia SQL Server.
- Variables
 - serverURI: String que contiene la dirección de el SQL Server.
 - serverToken: String que contiene el token de acceso de el SQL Server.

- Métodos
 - loadModel
 - Propósito: Cargar un modelo desde la base de datos.
 - Parámetros: String que contiene el ID del modelo.
 - Retorno: Un objeto de clase Model.
 - loadModelsInLocation
 - Propósito: Cargar una lista de modelos dada una ubicación geográfica
 - Parámetros: Objeto de tipo Location
 - Retorno: Un objeto de tipo List<Model> que contiene los modelos dada la ubicación.
 - storeModel
 - Propósito: Guardar un modelo en la base de datos dada una ubicación geográfica.
 - Parámetros: Objeto de tipo model y un Objeto de tipo Location.
 - Retorno: vacio.

4.6.1.2. Telemetry

- Propósito: Gestionar las llamadas al servicio de telemetría.
- Variables
 - insightsToken: String que contiene el token de acceso para application Insights.
- Métodos
 - logEvent
 - Propósito: Enviar un evento al servicio de telemetría.
 - Parámetros: Objeto de tipo Event que contiene la información a enviar.
 - Retorno: Vacío.

4.6.1.3. Comunication

- Propósito: Gestionar las llamadas HTTP entre el servidor y la aplicación.
- Variables
 - eventLogger: Objeto de tipo Telemetry para loggear eventos.

- sSocket: Objeto del tipo server socket para recibir y responder peticiones a los usuarios.

- Métodos

- Listener
 - Propósito: Escuchar las peticiones de los usuarios.
 - Parámetros: Objeto del tipo XMLHttpRequest contenido la petición de un usuario.
 - Retorno: Objeto del tipo JSON con la respuesta.
- Send
 - Propósito: Enviar respuesta hacia los usuarios.
 - Parámetros: Objeto de tipo JSON con la información.
 - Retorno: Objeto de tipo HTTPResponse con la respuesta hacia el usuario.

4.6.2. Aplicación móvil

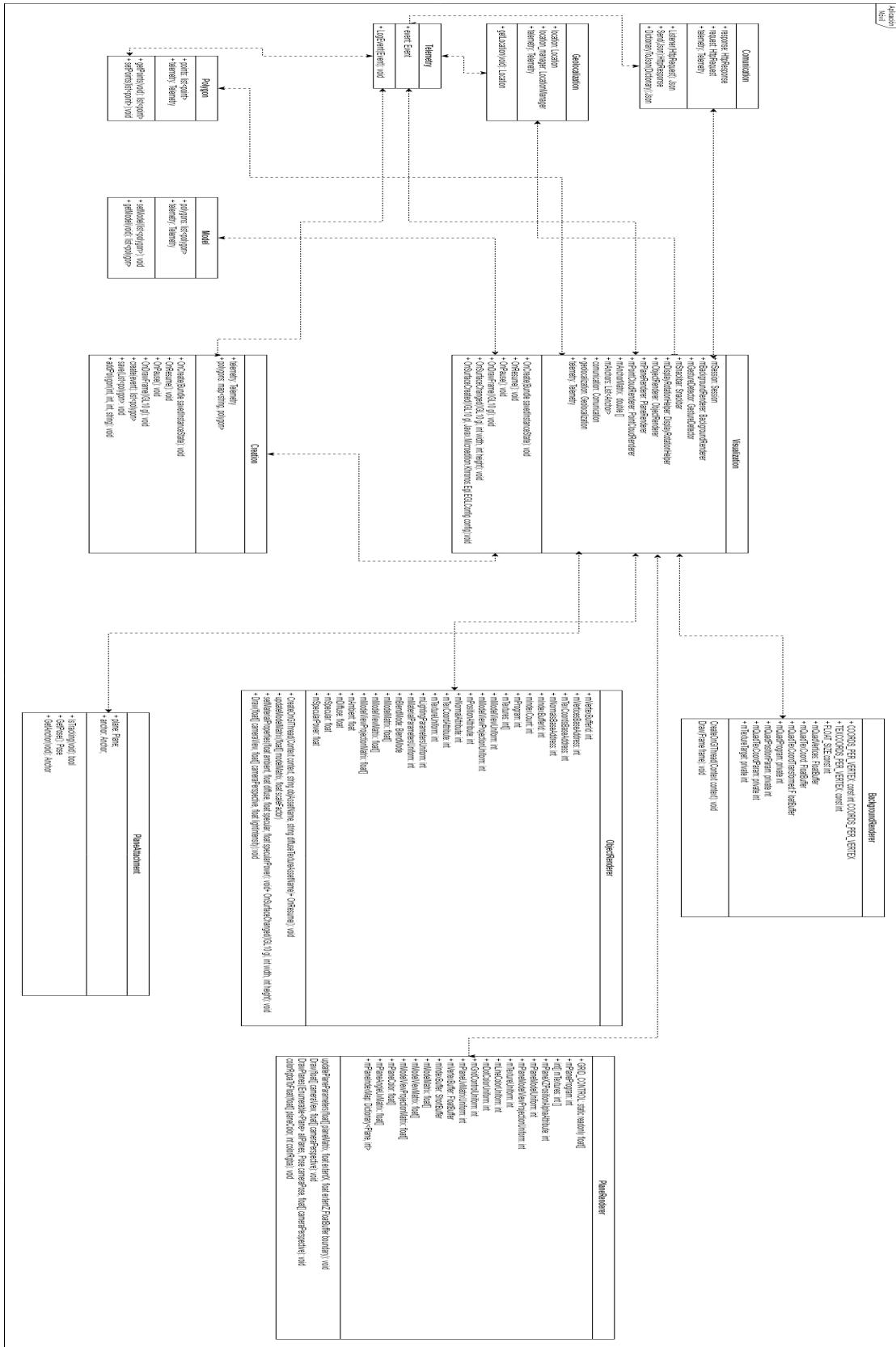


FIGURA 4.3: Diagrama de clases de la aplicación móvil

4.6.2.1. Polygon

- Propósito: Representar un polígono.
- Variables
 - points: Objeto del tipo List<point>.
 - telemetry: Objeto del tipo Telemetry para loggear eventos.
- Método
 - getPoints
 - Propósito: Regresar la lista de puntos.
 - Parámetros: Ninguno.
 - Retorno: Objeto del tipo List<point>de puntos.
 - setPoints
 - Propósito: Ingresar lista de puntos.
 - Parámetros: Objeto del tipo List<point>de puntos.
 - Retorno: Vació.

4.6.2.2. Model

- Propósito: Representar un modelo mediante polígonos.
- Variables
 - polygons: Objeto del tipo lista de polígonos.
 - telemetry: Objeto del tipo Telemetry para loggear eventos.
- Métodos
 - setModel
 - Propósito: Ingresar lista de polígonos.
 - Parámetros: Objeto del tipo List<Polygon>de polígonos.
 - Retorno: Vació.
 - getModel
 - Propósito: Regresar lista de polígonos.
 - Parámetros: Ninguno.
 - Retorno: Objeto del tipo lista de polígonos.

4.6.2.3. Geolocalization

- Propósito: Manejar la interacción de la aplicación con el GPS del teléfono.
- Variables
 - location: Objeto del tipo Location que guarda la ubicación actual.
 - location_manager: Objeto del tipo LocationManager que gestiona la interacción con el GPS.
 - telemetry: objeto del tipo Telemetry para loggear eventos.
- Métodos
 - getLocation
 - Propósito: Obtener ubicación actual del usuario.
 - Parámetros: Ninguno.
 - Retorno: Objeto del tipo Location.

4.6.2.4. Visualization

- Propósito: Manejo de gráficos de la aplicación.
- Variables
 - mSession: Objeto del tipo Session que guarda el estado actual de la aplicación
 - mBackgroundRenderer: Objeto del tipo BackgroundRenderer para renderizar el fondo.
 - mGestureDetector: Objeto del tipo GestureDetector para detectar acciones del usuario.
 - mSnackbar: Objeto del tipo Snackbar para enviar alertas al usuario.
 - mDisplayRotationHelper: Objeto del tipo DisplayRotationHelper para mostrar rotaciones.
 - mObjectRenderer: Objeto del tipo ObjectRenderer para renderizar los modelos.
 - mPlaneRenderer: Objeto del tipo PlaneRenderer para renderizar planos.
 - mPointCloudRenderer: Objeto del tipo PointCloudRenderer para renderizar la nube de puntos.
 - mAnchors: Objeto del tipo List<Anchor>que guarda la información de las anclas de los objetos.

- geolocation: Objeto del tipo Geolocation para la interacción con la ubicación del usuario.
- telemetry: Objeto del tipo Telemetry para el loggeo de eventos.

- Métodos

- OnCreate
 - Propósito: Configuración de la aplicación al iniciar.
 - Parámetros: Objeto del tipo Bundle que guarda la información de la aplicación.
 - Retorno: Vació.
- OnResume
 - Propósito: Configuración de la aplicación al retornar del estado de pausa.
 - Parámetros: Ninguno.
 - Retorno: Vació.
- OnPause
 - Propósito: Configuración de la aplicación al pausar.
 - Parámetros: Ninguno.
 - Retorno: Vació.
- OnDrawFrame
 - Propósito: Manejo de la perspectiva del usuario al mover la cámara.
 - Parámetros: Objeto del tipo IGL10 para el manejo de gráficos mediante OpenGL
 - Retorno: Vació.
- OnSurfaceChanged
 - Propósito: Manejo de las texturas y los objetos al ser puestos sobre un plano.
 - Parámetros: Objeto del tipo IGL10 para el manejo de gráficos mediante OpenGL, entero con el tamaño horizontal y un entero con el tamaño vertical de la superficie.
 - Retorno: Vació.
- OnSurfaceCreated
 - Propósito: Creación de planos una vez que son detectados.
 - Parámetros: Objeto del tipo IGL10 para el manejo de gráficos mediante OpenGL y objeto del tipo EGLConfig para la configuración gráfica.
 - Retorno: Vació.

4.6.2.5. Creation

- Propósito: Manejo de la creación de objetos.
- Variables
 - location: Objeto del tipo Location para la asignación de ubicación al objeto.
 - telemetry: Objeto del tipo Telemetry para el loggeo de eventos.
 - polygons: Objeto del tipo map<string,polygon>para el mapeo de los polígonos disponibles para la creación de modelos.
- Métodos
 - Create
 - Propósito: Creación de un nuevo modelo.
 - Parámetros: Objeto del tipo Event, que indica que se accionado el botón de terminar.
 - Retorno
 - Save
 - Propósito: Guardar un modelo.
 - Parámetros: Objeto del tipo List<Polygon>que contiene los polígonos utilizados para la creación del modelo.
 - Retorno: Vació.
 - addPolygon
 - Propósito: Agregar un polígono al modelo.
 - Parámetros: Tres enteros con las coordenadas X, Y y Z del centroide del polígono y una string que contiene el identificador de el polígono a agregar al modelo.
 - Retorno: Vació.

4.7. Descripción de actores

La herramienta la utilizará cualquier persona y podrá realizar las siguientes tareas que se presentan.

Usuario

Representa a cualquier persona. *Responsabilidades*

- Registrarse
- Identificarse
- Visualizar aumentaciones
- Interactuar con aumentaciones
- Crear aumentaciones

4.8. Características de la herramienta

Para satisfacer los requerimientos de la herramienta, se identificaron tres características.

4.8.1. C1. Identificación con la herramienta

La herramienta permitirá al usuario elegir un nombre de usuario único en caso de estar disponible, correo y contraseña para poder tener una identidad en el sistema. El correo sirve principalmente para recuperar el acceso a la herramienta y activar su cuenta.

4.8.2. C2. Interacción con aumentaciones

El usuario podrá visualizar e interactuar con las aumentaciones que ya estén disponibles cerca de su posición. Desde la interfaz principal de la herramienta le mostrará las aumentaciones más populares. También podrá calificar las aumentaciones que esté visualizando.

4.8.3. C3. Creación de aumentaciones

La herramienta permitirá al usuario en todo momento crear aumentaciones sobre las superficies detectadas. Sin embargo, la herramienta sólo permitirá crearlas en la misma posición en la que se encuentra el usuario en ese momento.

4.8.4. C4. Visualización de mapa

La herramienta mostrará al usuario en qué lugares hay aumentaciones para que esté pueda moverse hacia esa posición y visualizar aumentaciones.

4.9. Datos de la herramienta

Una vez definidas las características de la herramienta, se especifican los datos que la herramienta manejará para permitir el acceso al usuario, interactuar con las aumentaciones y crear aumentaciones.

4.9.1. Identificación del usuario

Para que el usuario pueda tener acceso a la herramienta son necesarios los siguientes datos:

- **DI1. Usuario**

Observaciones: Alfabético; Nombre de usuario único.

- **DI2. Correo**

Observaciones: Alfabético; Dirección de correo.

- **DI2. Password**

Observaciones: Alfabético; Contraseña.

4.9.2. Interacción con aumentaciones

Para interactuar con las aumentaciones se mostrarán los siguientes datos:

- **DR1. Aumentación**

Observaciones: Gráfico; Gráfico de la aumentación en 3D sobre la superficie en la fue creada.

- **DR2. Usuario**

Observaciones: Alfabético; Usuario que creó la aumentación.

- **DR2. Calificación**

Observaciones: Numérico; Promedio de calificación recibida por los usuarios.

4.9.3. Creación de aumentaciones

Para crear aumentaciones son necesarios los siguientes datos:

- **DC1. Usuario**

Observaciones: Alfabético; Nombre del usuario que está creando la aumentación

- **DC2. Posición**

Observaciones: Número; Coordenadas sobre las que está posicionado el usuario.

- **DC3. Tipo**

Observaciones: Alfabético; Tipo de plantilla de creación.

- **DC4. Polígono**

Observaciones: Figura; Polígono predefinido.

4.10. Máquina de Estados

Para satisfacer las características de la herramienta, se divide en las siguientes fases:

- Identificación
- Visualización de aumentaciones
- Interacción con aumentación
- Creación de aumentación

Para visualizar las fases es necesario un modelo de máquina de estados.

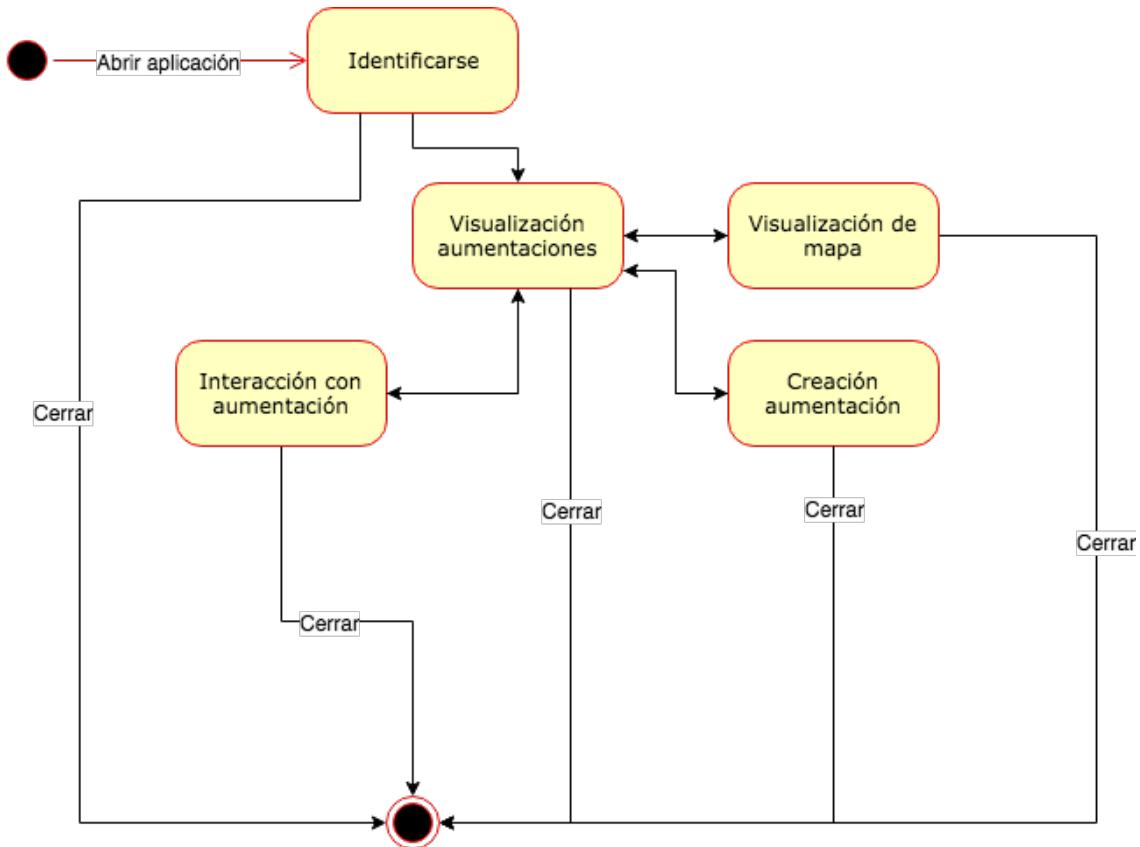


FIGURA 4.4: Máquina de estados

4.11. Modelo de casos de uso

En B-Side se identificaron 5 casos de uso necesarios para cubrir las características del sistema.

- CU1. Identificar al usuario
- CU2. Registrar al usuario
- CU3. Visualizar aumentaciones
- CU4. Interactuar con aumentación
- CU5. Crear aumentación
- CU6. Visualizar mapa

En la figura 4.5 se muestra el modelo de Casos de Uso de la herramienta.

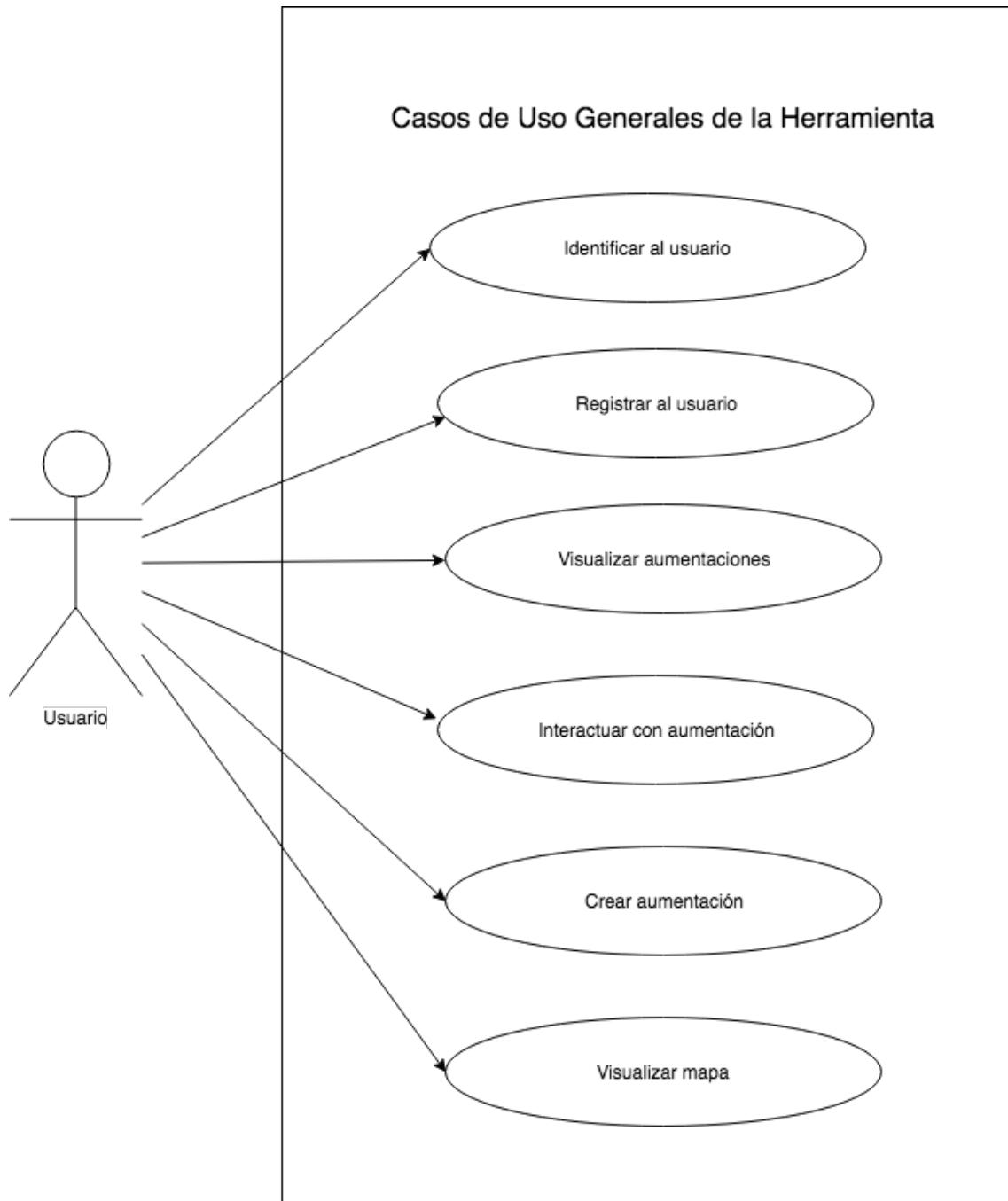


FIGURA 4.5: Modelo de Casos de Uso

En la tabla 4.1 se muestran las características de la sección 4.8 que se satisfacen en cada caso de uso.

Matriz de Casos de Uso		
Caso de Uso	Nombre	Característica de la Herramienta
CU1	Identificar al usuario	C1
CU2	Registrar al usuario	C1
CU3	Visualizar aumentaciones	C2
CU4	Interactuar con aumentación	C2
CU5	Crear aumentación	C3
CU6	Visualizar mapa	C4

CUADRO 4.1: Matriz de Casos de Uso

4.12. Descripción de casos de uso

4.12.1. CU1. Identificar al usuario

CU1. Identificar al usuario	
Actor	Usuario
Propósito	Iniciar una sesión para que pueda crear aumentaciones e interactuar con las mismas
Resumen	El actor iniciará la herramienta y esta mostrará un formulario con los campos requeridos para iniciar sesión o un botón enlace que lo lleve al CU2 (4.12.2)
Entradas	Credenciales del usuario
Precondiciones	No debe estar iniciada una sesión en la herramienta
Postcondiciones	Sesión inicializada en la herramienta para el usuario identificado correctamente
Autor(es)	

Trayectoria Principal

1. El usuario debe ingresar el nombre de usuario y contraseña.
2. La herramienta verifica que sean correctos. [**Trayectoria A**]
3. La herramienta muestra la interfaz "Principal"([4.14](#)).

————— Fin de la Trayectoria

Trayectoria A: Nombre de usuario o contraseña inválido

1. La herramienta muestra el mensaje: "*Nombre de usuario o contraseña inválido*"
2. Regresa al paso **1** de la **Trayectoria Principal**.

————— Fin de la Trayectoria

4.12.2. CU2. Registrar al usuario

CU2. Registrar al usuario	
Actor	Usuario
Propósito	Almacenar la información del usuario necesaria para identificarse
Resumen	El actor deberá elegir la opción "Registrarse" para que lo lleve a la interfaz "Registro" (4.12)
Entradas	Datos de registro
Precondiciones	El usuario y correo no deben haber sido registrados antes en el sistema
Postcondiciones	Se enviará un correo de confirmación para que se pueda finalizar el registro
Autor(es)	

Trayectoria Principal

1. El usuario debe ingresar su correo, nombre de usuario, contraseña y verificar la misma.
2. La herramienta verifica que sean válidos. **[Trayectoria A]**
3. **[Trayectoria B]**
4. **[Trayectoria C]**
5. **[Trayectoria D]**
6. La herramienta muestra la interfaz "Envío de confirmación" (Ref).

————— Fin de la Trayectoria

Trayectoria A: Correo inválido

1. La herramienta muestra el mensaje: "*El correo ingresado es inválido*"

2. Regresa al paso **1** de la **Trayectoria Principal**.

————— Fin de la Trayectoria

Trayectoria B: Usuario inválido

1. La herramienta muestra el mensaje: "*El nombre de usuario ingresado es inválido*"
2. Regresa al paso **1** de la **Trayectoria Principal**.

————— Fin de la Trayectoria

Trayectoria C: Contraseña inválida

1. La herramienta muestra el mensaje: "*La contraseña ingresada no cumple con los requisitos*"
2. Regresa al paso **1** de la **Trayectoria Principal**.

————— Fin de la Trayectoria

Trayectoria D: Contraseñas no son iguales

1. La herramienta muestra el mensaje: "*Las contraseñas ingresadas no son iguales*"
2. Regresa al paso **1** de la **Trayectoria Principal**.

————— Fin de la Trayectoria

4.12.3. CU3. Visualizar aumentaciones

CU3. Visualizar aumentaciones	
Actor	Usuario
Propósito	Visualizar las aumentaciones que se encuentran cerca de la posición
Resumen	La herramienta mostrará la interfaz "Principal" (4.14) con las aumentaciones que se encuentren más cercanas y populares a la posición del usuario
Entradas	Posición del usuario
Precondiciones	El usuario deberá apuntar con la cámara del dispositivo hacia donde quiera visualizar aumentaciones
Postcondiciones	Se mostrarán las aumentaciones más populares y cercanas en un radio de 10 metros
Autor(es)	

Trayectoria Principal

1. El usuario debe apuntar a la zona donde quiera visualizar las aumentaciones.
2. La herramienta buscará las aumentaciones que estén cercanas a la zona en un radio de 10 metro. **[Trayectoria A]**
3. La herramienta muestra las aumentaciones colocadas sobre la superficie donde fueron creadas en la interfaz "Principal" ([4.14](#)).

————— Fin de la Trayectoria

Trayectoria A: No se encontraron aumentaciones

1. La herramienta muestra el mensaje: "**No hay aumentaciones cerca de la zona. ¡Tú puedes ser el primero en crear una!**"

————— Fin de la Trayectoria

4.12.4. CU4. Interactuar con aumentación

CU4. Interactuar con aumentación	
Actor	Usuario
Propósito	Calificar una aumentación.
Resumen	La herramienta mostrará la interfaz "Interacción"(4.15) al seleccionar una aumentación
Entradas	Aumentación seleccionada
Precondiciones	El usuario deberá seleccionar una aumentación mostrada en la interfaz "Principal"(4.14)
Postcondiciones	La herramienta mostrará el menú "Calificar." en el cual el usuario podrá emitir una calificación para la aumentación en un rango de 0 a 5
Autor(es)	

Trayectoria Principal

1. El usuario debe seleccionar una aumentación visible.
2. La herramienta mostrará el menú "Calificar" con figuras prellenadas que representan el promedio que ha recibido por otros usuarios.
3. [Trayectoria A]

————— Fin de la Trayectoria

Trayectoria A: Emitir calificación

1. El usuario emite una calificación en un rango de 0 a 5.

————— Fin de la Trayectoria

4.12.5. CU5. Crear aumentación

CU5. Crear aumentación	
Actor	Usuario
Propósito	Crear un aumentación en la misma zona que se encuentra el usuario
Resumen	La herramienta mostrará la interfaz "Creación" (4.17) decidir con qué tipo de plantilla se comenzará a crear la aumentación.
Entradas	<ul style="list-style-type: none"> ■ Tipo de plantilla ■ Polígono
Precondiciones	El usuario deberá seleccionar la opción "Colocar" de la interfaz "Principal" (4.14) y elegir un tipo de plantilla
Postcondiciones	La herramienta colocará la aumentación en la zona en la que se encontraba el usuario cuando seleccionó la opción "Colocar"
Autor(es)	

Trayectoria Principal

1. El usuario debe seleccionar la opción "Colocar"
2. La herramienta mostrará el menú "Plantillas"
3. El usuario debe elegir una de las plantillas
4. La herramienta mostrará la interfaz "Creación" (4.17)
5. El usuario debe seleccionar un polígono del menú "Polígonos"
6. La herramienta mostrará un vista previa de la aumentación según el polígono y plantilla seleccionados.
7. El usuario dará selecciona la opción "Crear"
8. La herramienta mostrará la aumentación colocada sobre la superficie.

————— Fin de la Trayectoria

4.12.6. CU6. Visualizar mapa

CU6. Visualizar mapa	
Actor	Usuario
Propósito	Mostrar
Resumen	El actor iniciará la herramienta y esta mostrará un formulario con los campos requeridos para iniciar sesión o un botón enlace que lo lleve al CU2 (4.12.2)
Entradas	Credenciales del usuario
Precondiciones	No debe estar iniciada una sesión en la herramienta
Postcondiciones	Sesión inicializada en la herramienta para el usuario identificado correctamente
Autor(es)	

Trayectoria Principal

1. El usuario debe ingresar el nombre de usuario y contraseña.
2. La herramienta verifica que sean correctos. [**Trayectoria A**]
3. La herramienta muestra la interfaz "Principal" ([4.14](#)).

————— Fin de la Trayectoria

Trayectoria A: Nombre de usuario o contraseña inválido

1. La herramienta muestra el mensaje: "**Nombre de usuario o contraseña inválido**"
2. Regresa al paso 1 de la **Trayectoria Principal**.

————— Fin de la Trayectoria

4.13. Diagramas de secuencia

Los siguientes diagramas de secuencia contemplan las acciones realizadas por el usuario.

4.13.1. DS1. Identificar al usuario

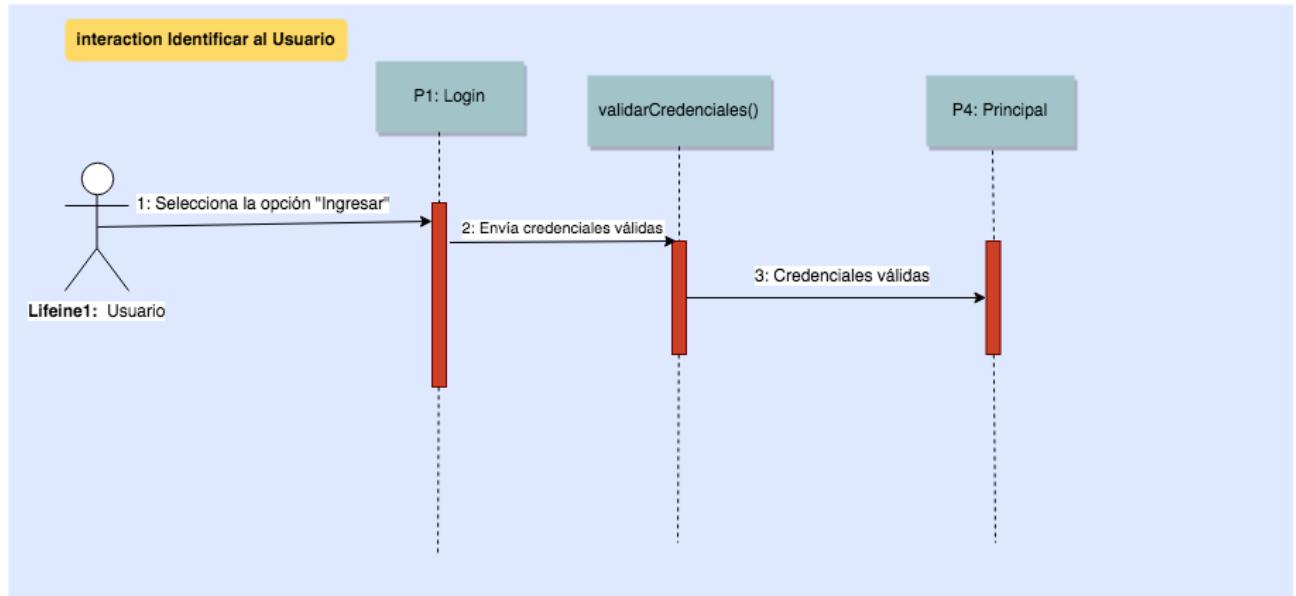


FIGURA 4.6: DS1. Identificar al usuario

4.13.2. DS2. Registrar al usuario

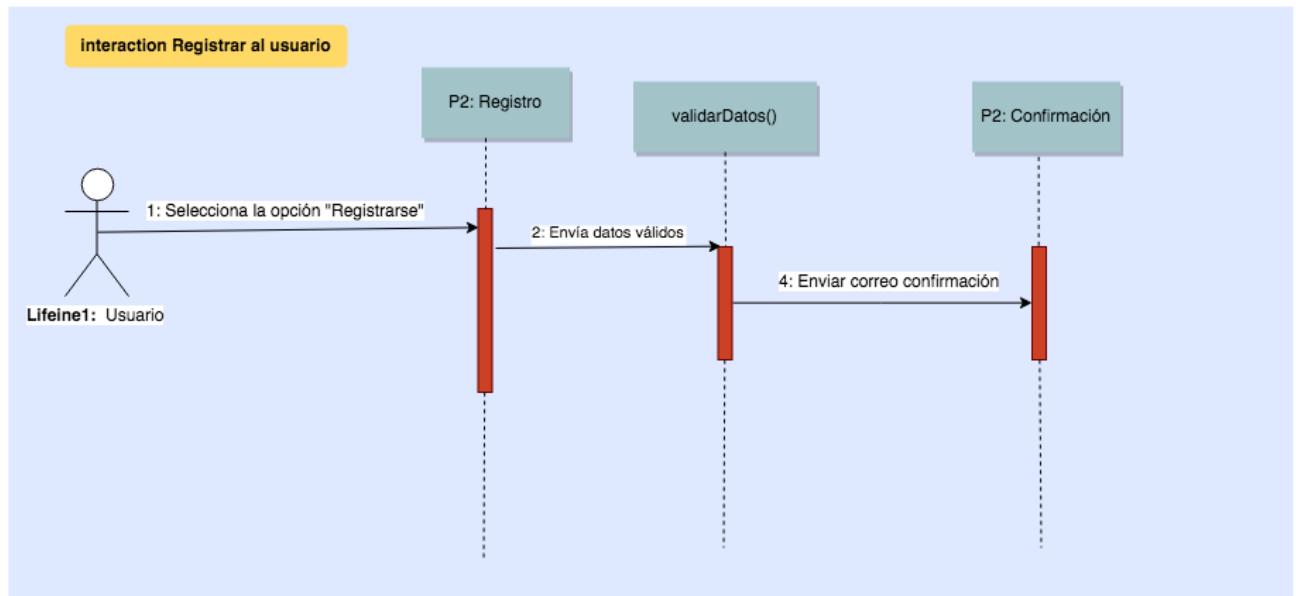


FIGURA 4.7: DS2. Registrar al usuario

4.13.3. DS3. Visualizar aumentaciones

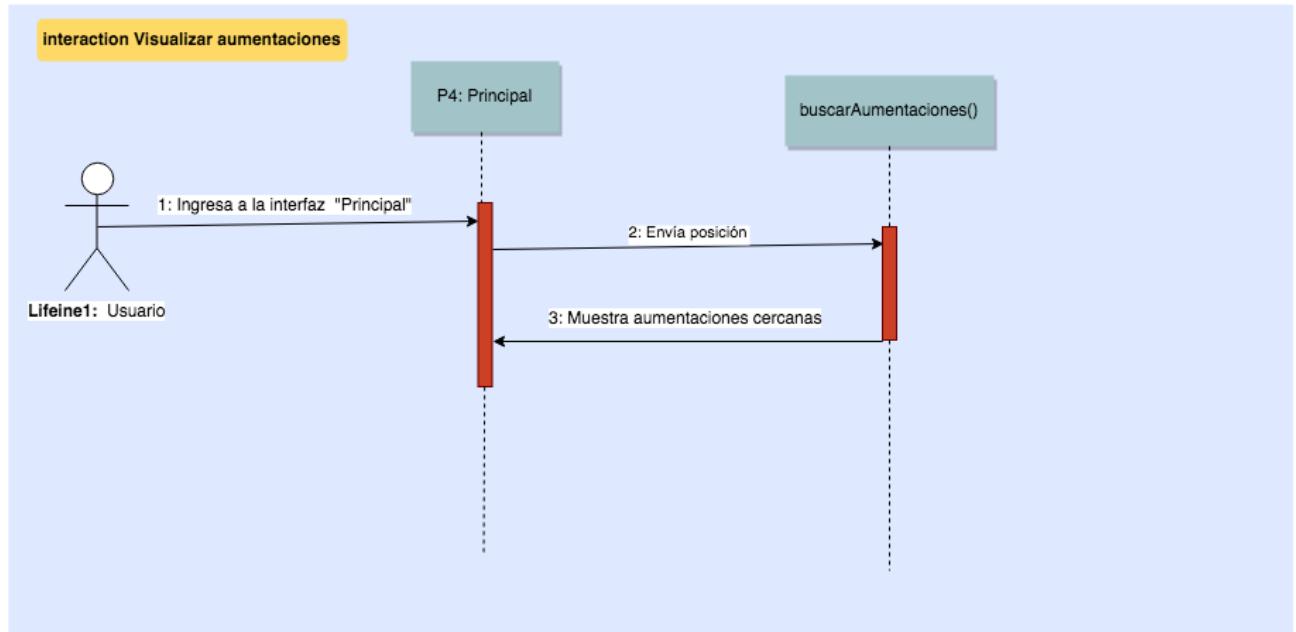


FIGURA 4.8: DS3. Visualizar aumentaciones

4.13.4. DS4. Interactuar con aumentación

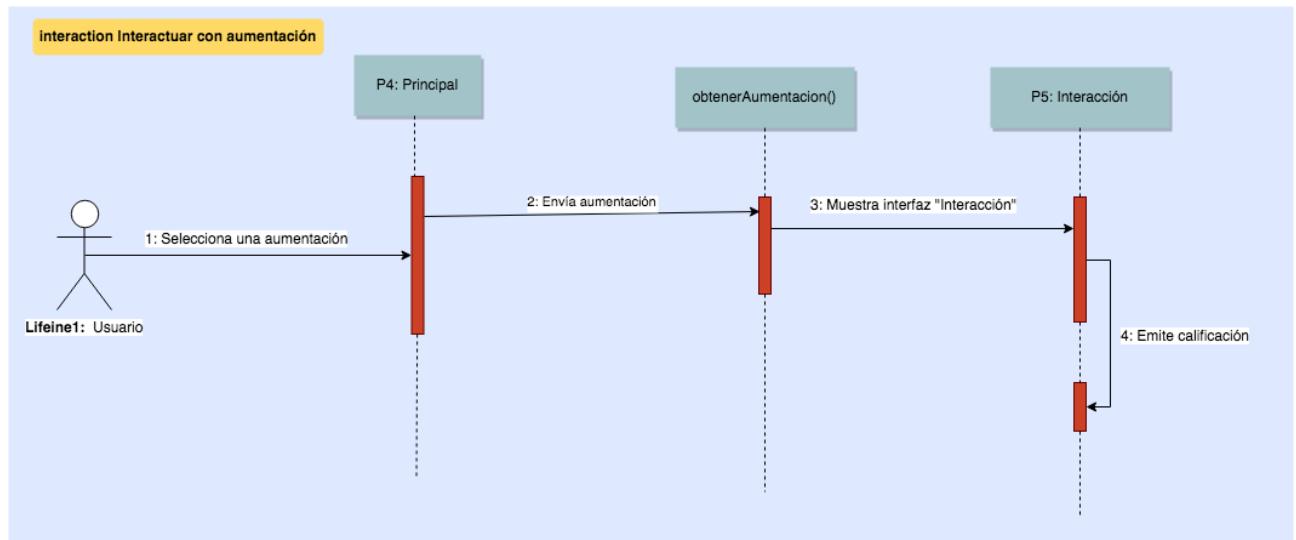


FIGURA 4.9: DS4. Interactuar con aumentación

4.13.5. DS5. Crear aumentación

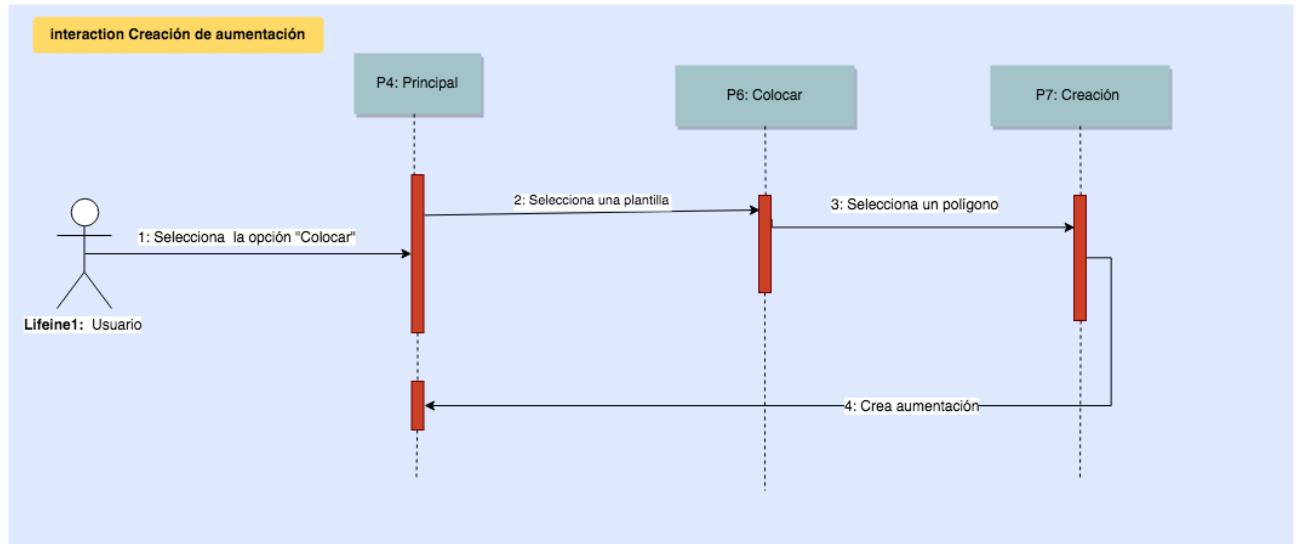


FIGURA 4.10: DS5. Crear aumentación

4.14. Interfaces

4.14.1. P1: Login

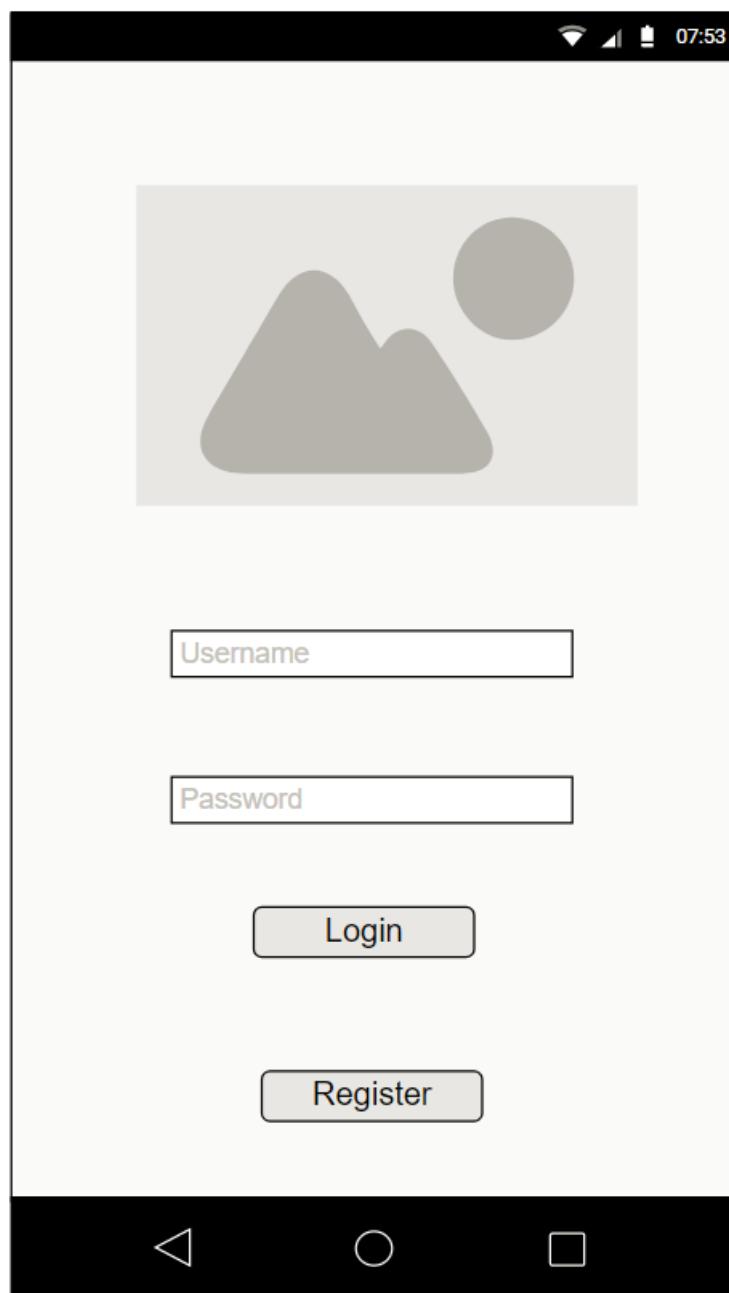


FIGURA 4.11: Login de la aplicación

4.14.2. P2: Registro

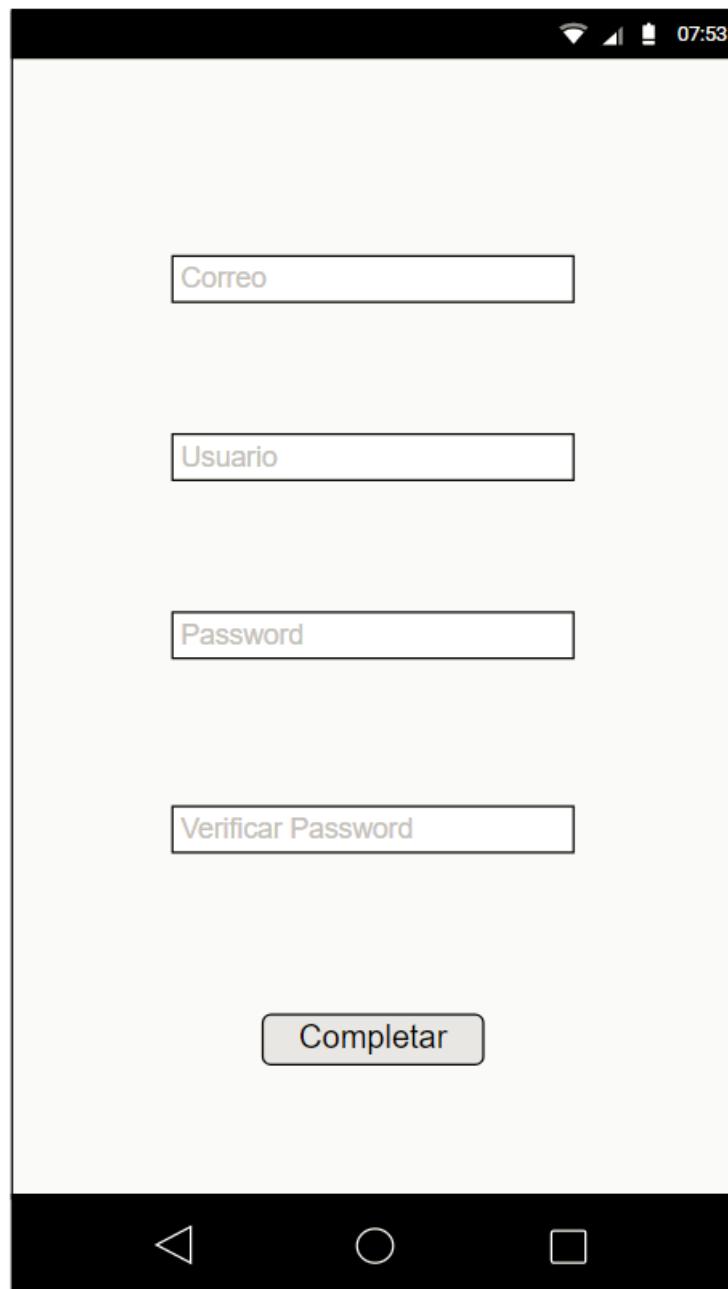


FIGURA 4.12: Registro de un nuevo usuario

4.14.3. P3: Confirmación

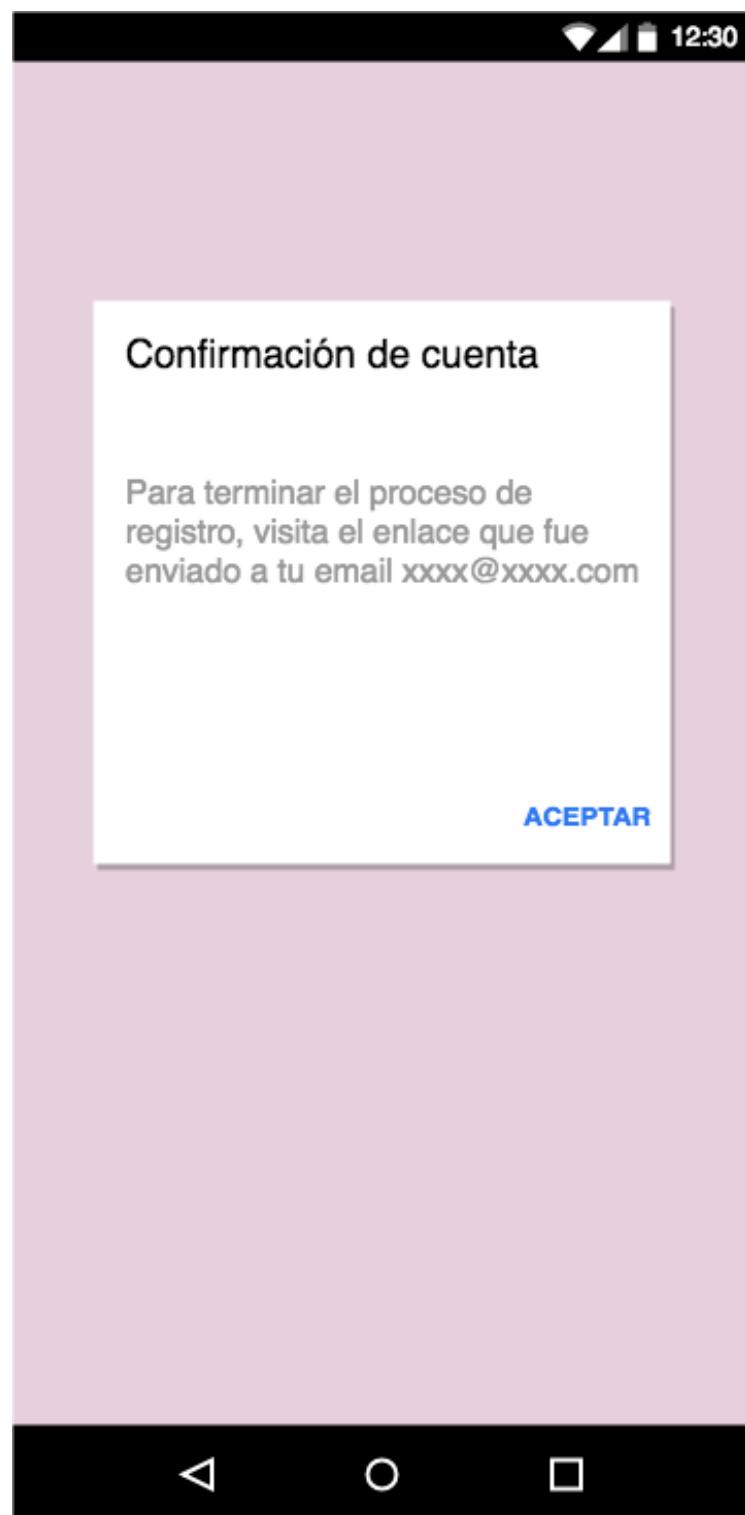


FIGURA 4.13: Confirmación del registro

4.14.4. P4: Principal

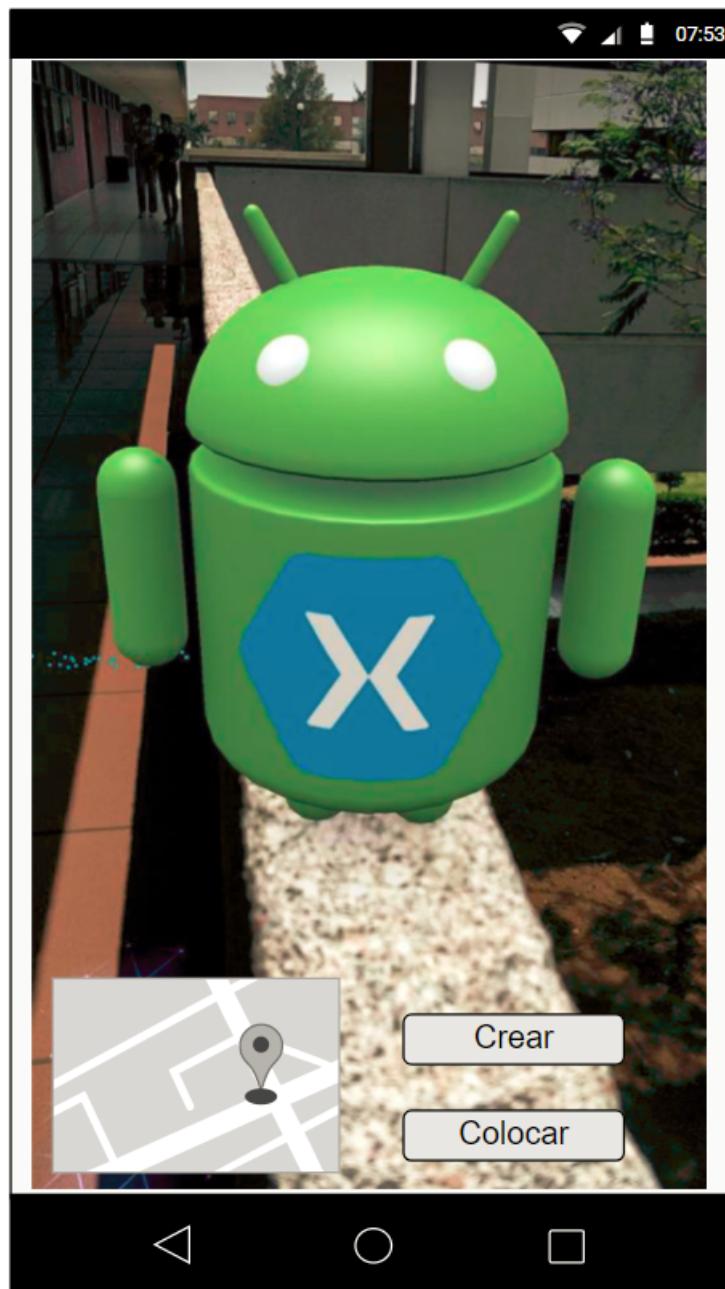


FIGURA 4.14: Interfaz principal

4.14.5. P5: Interacción

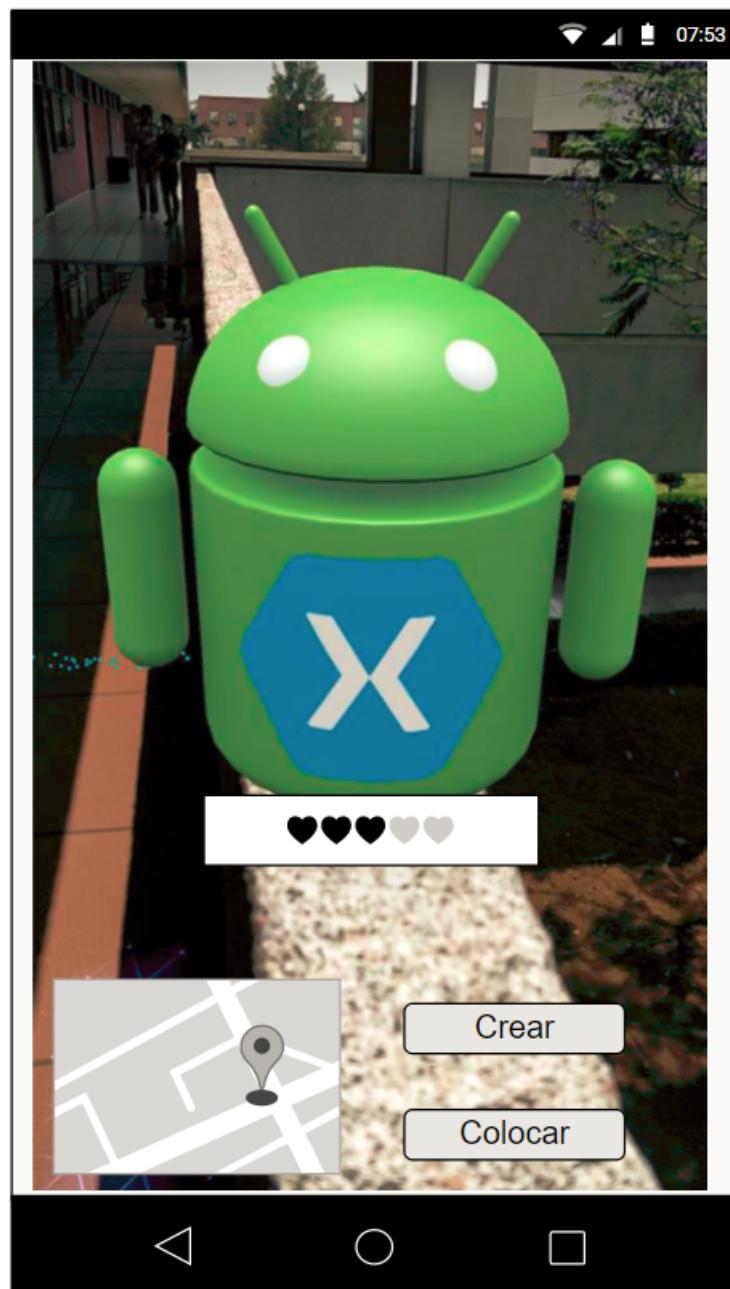


FIGURA 4.15: Interacción con un modelo de RA

4.14.6. P6: Colocar

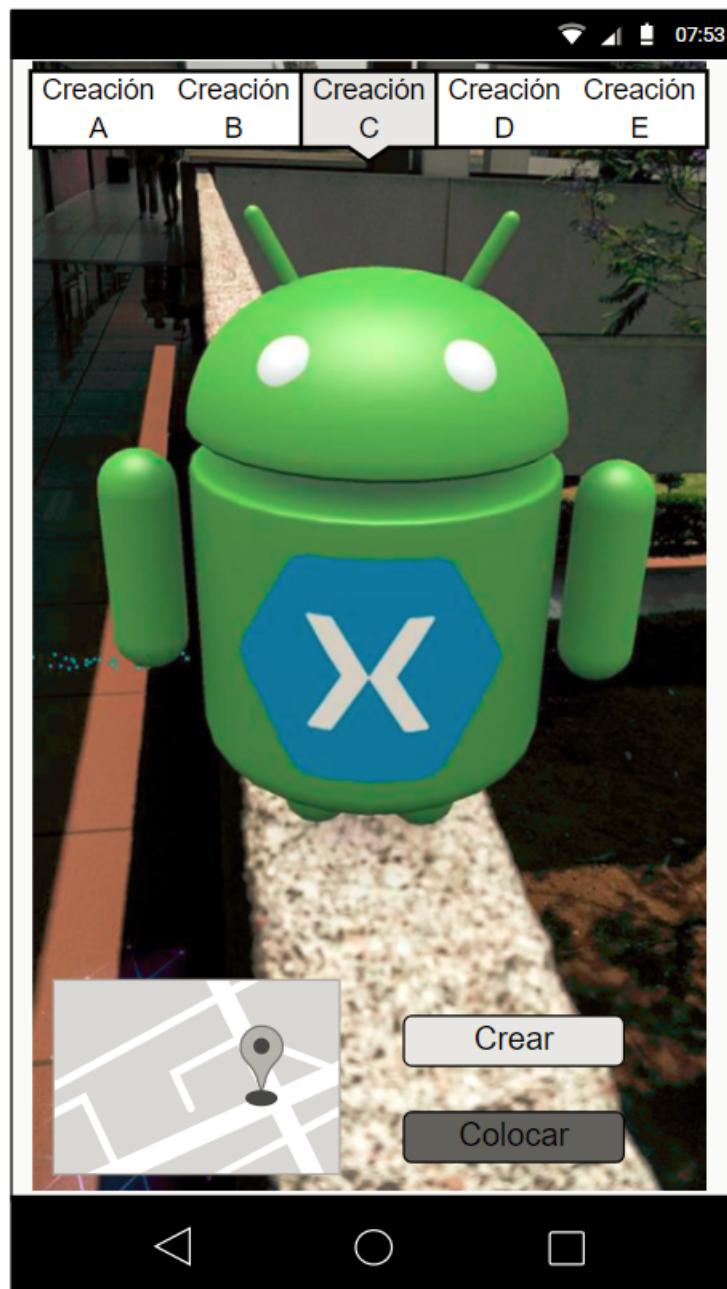


FIGURA 4.16: Colocar una nuevo modelo de RA

4.14.7. P7: Creación



FIGURA 4.17: Menú de creación

4.14.8. P8: Mapa

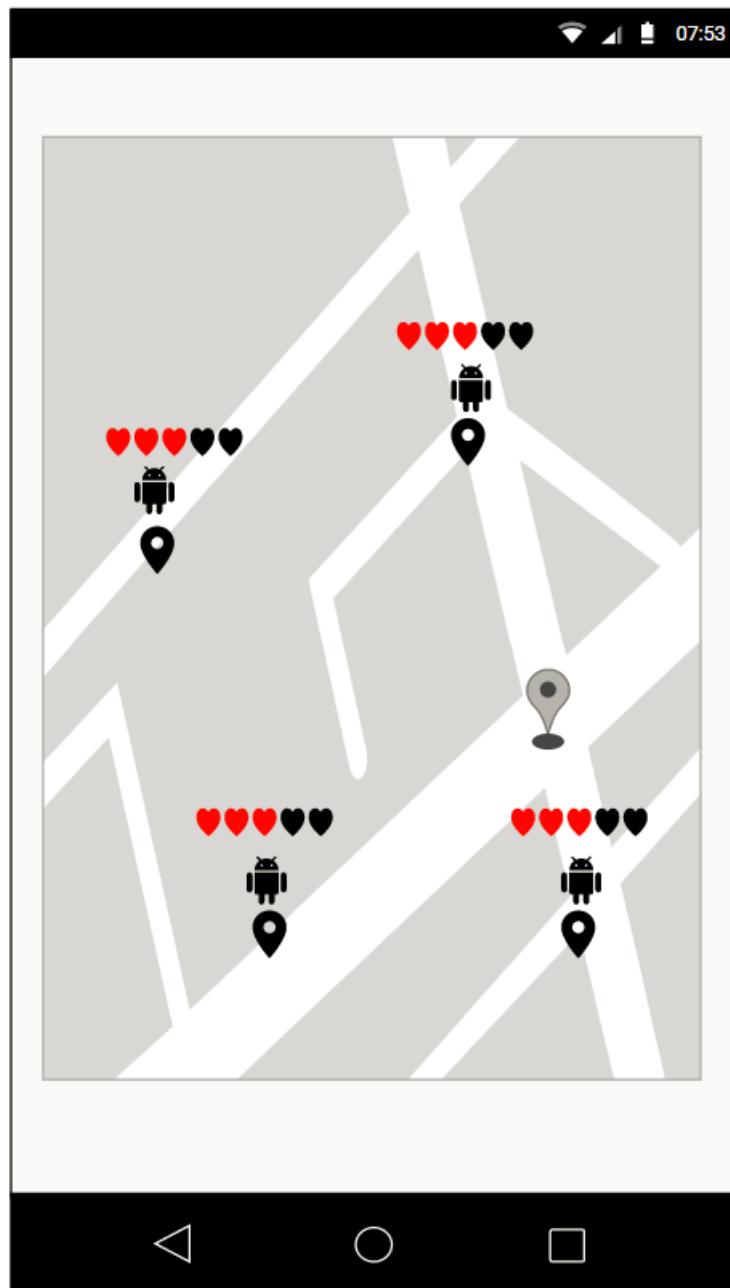


FIGURA 4.18: Mapa de modelos de RA populares en el área cercana

4.15. Modelo de datos

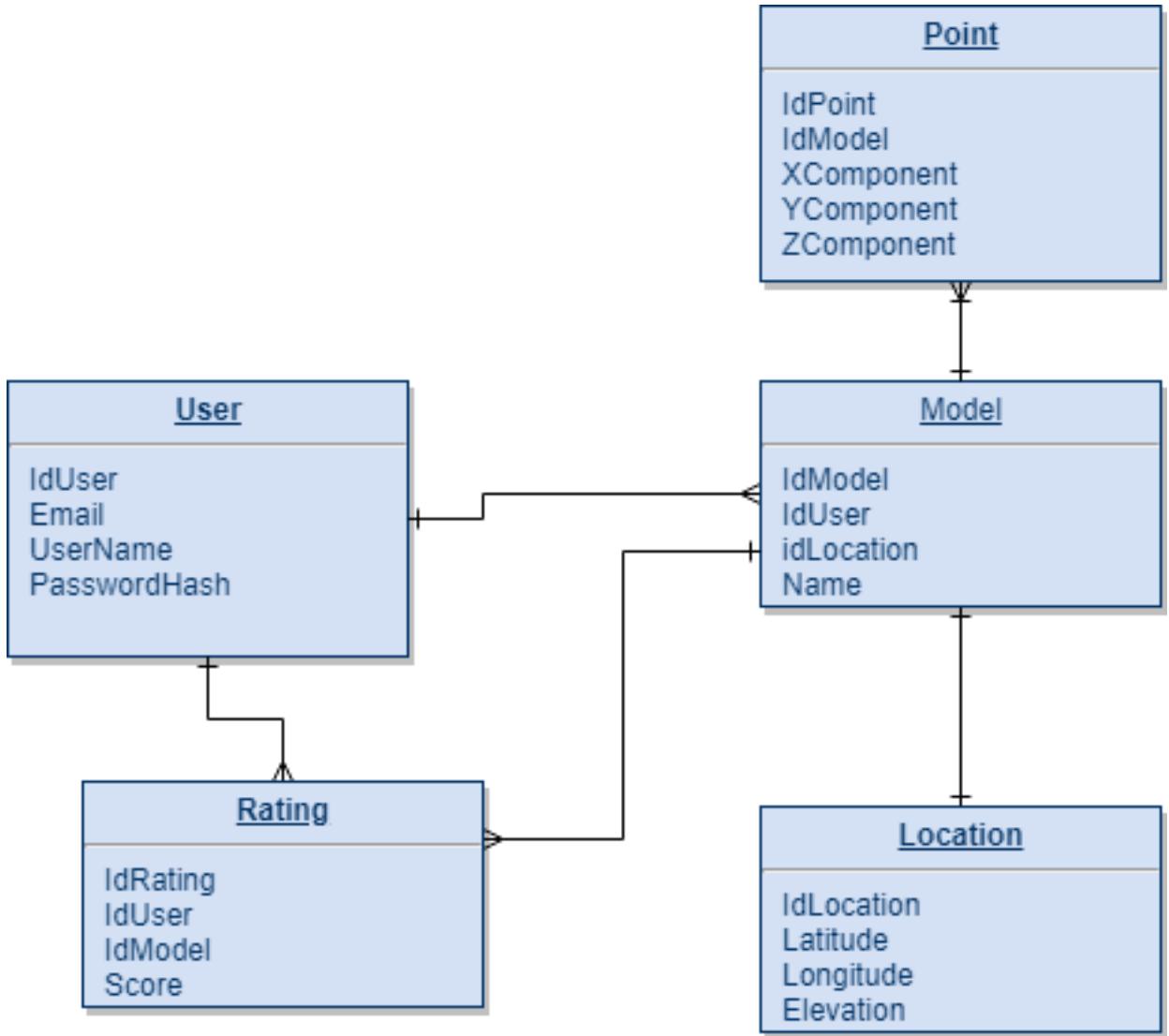


FIGURA 4.19: Modelo de datos del sistema

4.15.1. Tablas

1. User: El propósito de esta tabla es guardar la información de los usuarios, esta ligada principalmente con los modelos y los ratings, un usuario puede tener muchos modelos y muchos ratings de modelos .
2. Rating: Su propósito es guardar el rating que un usuario ha asignado a un modelo, un rating pertenece a un usuario y a un modelo.
3. Model: Su propósito es guardar la información de los modelos, un modelo esta compuesto de puntos por lo que su relación es de uno a muchos, un modelo pertenece

a un usuario, un modelo pertenece a una ubicación y varios ratings pertenecen a un modelo.

4. Point: Su principal uso es guardar la información de los puntos que pertenecen a un modelo.
5. Location: El propósito de esta tabla es guardar la información de la latitud, longitud y elevación donde fue colocado un modelo.

Bibliografía

- [1] Maisi McCabe. Victoria station lynx. <https://www.campaignlive.co.uk/article/lynx-turns-outdoor-augmented-reality/1061042>. Accesado: 2018-05-03.
- [2] David Kiefaber. National geographic dinosaurs. <http://www.adweek.com/creativity/national-geographic-lets-you-pet-dinosaurs-mall-136591/>. Accesado: 2018-05-03.
- [3] Andrew Liszewski. Augmented rome. <https://gizmodo.com/experience-romes-coliseum-like-it-just-opened-through-t-512843401>. Accesado: 2018-05-03.
- [4] Judy Berna. Lego digital box. <https://www.wired.com/2011/11/lego-digital-box-a-clever-new-way-to-sell-lego/>. Accesado: 2018-05-03.
- [5] Chris Cameron. Introducing stiktu. <https://www.layar.com/news/blog/2011/12/20/introducing-stiktu/>, . Accesado: 2018-05-03.
- [6] Chris Cameron. Tgs: Vita takes augmented reality to new level. <http://www.ign.com/articles/2011/09/15/tgs-vita-takes-augmented-reality-to-new-level>, . Accesado: 2018-05-03.
- [7] David Goldman. Google unveils 'project glass' virtual-reality glasses. <http://money.cnn.com/2012/04/04/technology/google-project-glass/>. Accesado: 2018-05-03.
- [8] Niantic. Pokemon go. <https://www.pokemon.com/us/pokemon-video-games/pokemon-go/>. Accesado: 2018-05-03.
- [9] Rodrigo Silva, Jauvane Oliveira, and G A. Giraldi. *Introduction to augmented reality*. 01 2003.
- [10] Application insights overview. <https://docs.microsoft.com/en-us/azure/application-insights/app-insights-overview>. Accesado: 2018-04-23.

- [11] Power bi resumen. <https://powerbi.microsoft.com/en-us/what-is-power-bi/>, . Accesado: 2018-04-23.
- [12] Xamarin documentation. ¿que es xamarin? <https://www.xamarin.com/platform>. Accesado: 2018-04-20.
- [13] Google Developers. Arcore overview. <https://developers.google.com/ar/discover/>, . Accesado: 2018-04-25.
- [14] International Data Corporation. Pronostico del crecimiento del mercado de ra. <https://www.statista.com/statistics/591181/global-augmented-virtual-reality-market-size/>. Accesado: 2018-04-24.
- [15] Alan Dix. Human-computer interaction. In *Encyclopedia of database systems*, pages 1327–1331. Springer, 2009.
- [16] Margaret Rouse. Realidad aumentada. <https://whatis.techtarget.com/definition/augmented-reality-AR>. Accesado: 2018-03-03.
- [17] Geolocation 101: How it works, the apps, and your privacy. <https://www.pcworld.com/article/192803/geolo.html>, . Accesado: 2018-04-23.
- [18] Tracking a suspect by mobile phone. <http://news.bbc.co.uk/2/hi/technology/4738219.stm>, . Accesado: 2018-04-23.
- [19] Apple Developers. Arkit overview. <https://developer.apple.com/arkit/>, . Accesado: 2018-04-25.
- [20] Servicios azure. <https://azure.microsoft.com/es-mx/>, . Accesado: 2018-04-20.
- [21] Precio azure. <https://azure.microsoft.com/en-us/pricing/calculator/>, . Accesado: 2018-04-20.
- [22] Servicios aws. https://aws.amazon.com/?nc2=h_lg, . Accesado: 2018-04-20.
- [23] Precio aws. <https://calculator.s3.amazonaws.com/index.html>, . Accesado: 2018-04-20.
- [24] Servicios google cloud. <https://cloud.google.com/why-google-cloud/>, . Accesado: 2018-04-20.
- [25] Precio google cloud. <https://cloud.google.com/products/calculator/>, . Accesado: 2018-04-20.
- [26] Costo azure. <https://azure.microsoft.com/en-us/pricing/calculator/>, . Accesado: 2018-04-16.

- [27] Costo amazon web services. <https://calculator.s3.amazonaws.com/index.html>, . Accesado: 2018-04-16.
- [28] Costo google cloud. <https://cloud.google.com/products/calculator/#id=180d2994-c046-4ef1-8f6e-2289b52b4928>, . Accesado: 2018-04-16.
- [29] T-sql. <https://msdn.microsoft.com/en-us/library/bb510741.aspx>. Accesado: 2018-04-16.
- [30] Grant Allen and Mike Owens. *The Definitive Guide to SQLite*. Apress, Berkely, CA, USA, 2nd edition, 2010. ISBN 1430232250, 9781430232254.
- [31] Telemetry: Summary of concept and rationale. *NASA STI/Recon Technical Report N*, 89, December 1987.
- [32] What is telemetry? <http://www.sealtag.org/What%20Is%20Telemetry.html>. Accesado: 2018-04-23.
- [33] Google analytics. https://www.google.com/analytics/analytics/#?modal_active=none, . Accesado: 2018-04-23.
- [34] Google. Google analytics for mobile apps. <https://developers.google.com/analytics/solutions/mobile>, . Accesado: 2018-04-23.
- [35] Google analytics limits. <https://developers.google.com/analytics/devguides/collection/analyticsjs/limits-quotas>, . Accesado: 2018-04-23.
- [36] Power bi conceptos. <https://docs.microsoft.com/en-us/power-bi/service-basic-concepts>, . Accesado: 2018-04-23.
- [37] Power bi costo. <https://reviews.financesonline.com/p/microsoft-power-bi/>, . Accesado: 2018-04-23.
- [38] Material design. <https://material.io/guidelines/>. Accesado: 2018-04-20.
- [39] C#. <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/>. Accesado: 2018-04-23.
- [40] Common language infrastructure. <http://www.ecma-international.org/publications/standards/Ecma-335.htm>. Accesado: 2018-04-23.
- [41] James Gosling, Bill Joy, Guy L. Steele, Gilad Bracha, and Alex Buckley. *The Java Language Specification, Java SE 8 Edition*. Addison-Wesley Professional, 1st edition, 2014. ISBN 013390069X, 9780133900699.

- [42] Google developers documentation. Overview android sdk. <https://developer.android.com/studio/command-line/>. Accesado: 2018-05-02.
- [43] Don Reisinger. Número de usuarios de iphone. <https://www.statista.com/statistics/276306/global-apple-iphone-sales-since-fiscal-year-2007/>, . Accesado: 2018-04-25.
- [44] Don Reisinger. Número de iphone x vendidos. <http://fortune.com/2018/01/23/apple-iphone-x-sales-2/>, . Accesado: 2018-04-25.
- [45] Ron Amadeo. Número de teléfonos compatibles con arcore. <https://arstechnica.com/gadgets/2018/02/googles-arcore-hits-version-1-0-brings-augmented-reality-to-100-million-devices/>. Accesado: 2018-04-25.
- [46] Joe Rossignol. Número de usuarios de android. <https://www.macrumors.com/2017/05/17/2-billion-active-android-devices/>. Accesado: 2018-04-25.
- [47] Neil Hughes. Número de aplicaciones que utilizan arkit. <https://appleinsider.com/articles/18/01/03/developer-adoption-of-apples-arkit-tepid-after-ios-11-launch-data-suggests>. Accesado: 2018-04-25.
- [48] Ryan Dawsl. Número de aplicaciones que utilizan arcore. <https://www.developer-tech.com/news/2018/mar/21/arcore-used-over-85-apps-android/>. Accesado: 2018-04-25.
- [49] AppBrain. Número de aplicaciones disponibles para android. <https://www.appbrain.com/stats/number-of-android-apps>. Accesado: 2018-04-25.
- [50] Sam Costello. Número de aplicaciones disponibles para ios. <https://www.lifewire.com/how-many-apps-in-app-store-2000252>. Accesado: 2018-04-25.
- [51] Statista. Pronostico del tamaño del mercado de la industria del entretenimiento. <https://www.statista.com/statistics/259985/global-filmed-entertainment-revenue/>. Accesado: 2018-04-25.
- [52] Google. Google developers. <https://developers.google.com/>, . Accesado: 2018-04-23.
- [53] Microsoft. Microsoft documentation. <https://docs.microsoft.com/>. Accesado: 2018-04-23.
- [54] Azure legal. Azure appservice service level agreement. https://azure.microsoft.com/en-us/support/legal/sla/app-service/v1_4/, . Accesado: 2018-05-03.

- [55] Azure legal. Azure storage service level agreement. https://azure.microsoft.com/en-us/support/legal/sla/storage/v1_3/, . Accesado: 2018-05-03.
- [56] Azure legal. Azure sql server service level agreement. https://azure.microsoft.com/en-us/support/legal/sla/sql-database/v1_1/, . Accesado: 2018-05-03.
- [57] Azure legal. Azure application insights service level agreement. https://azure.microsoft.com/es-mx/support/legal/sla/application-insights/v1_0/, . Accesado: 2018-05-03.
- [58] Tutorialspoint. Azure scalability. https://www.tutorialspoint.com/microsoft_azure/microsoft_azure_scalability.htm. Accesado: 2018-05-03.