



INSTITUTO MEXICANO DEL TRANSPORTE

Implementación de algoritmos de realidad aumentada para la visualización de indicadores en el autotransporte

Cesar Jaime Montiel Moctezuma
Miguel Gastón Cedillo Campos
Bernardo Hernández Sánchez

Publicación Técnica No. 706
Sanfandila, Qro.
2022

ISSN 0188-7297

Esta investigación fue realizada dentro del Laboratorio Nacional Conacyt en Sistemas de Transporte y Logística de la Coordinación de Transporte Integrado y Logística del Instituto Mexicano del Transporte (IMT), por el Dr. Cesar Jaime Montiel Moctezuma, el Dr. Miguel Gastón Cedillo Campos y el Mtro. en C. Bernardo Hernández Sánchez.

Esta investigación es el producto final del proyecto de investigación interna TI 17/20: *“Implementación de algoritmos de realidad aumentada para la visualización de indicadores en el autotransporte”*.

Se agradece la revisión y aportaciones del Dr. Carlos Martner Peyrelongue, coordinador de Transporte Integrado y Logística del IMT, cuyas observaciones mejoraron la calidad de este documento.

Contenido

	Página
Sinopsis.....	vii
Abstract.....	ix
Resumen ejecutivo.....	xi
Introducción.....	1
1. Antecedentes.....	3
1.1 Tecnologías inmersivas.....	3
1.1.1 Realidad aumentada.....	3
1.1.2 Realidad virtual.....	4
1.1.3 Realidad mixta.....	4
1.2 Impacto y beneficios de la tecnología de realidad aumentada en logística y transporte.....	5
1.2.1 Realidad aumentada en el transporte, una vista al pasado.....	5
1.2.2 Proyectos e instituciones que trabajan con realidad aumentada.....	8
2. Tecnología de Realidad Aumentada.....	13
2.1 Algoritmos.....	13
2.1.1 Simultaneous Localization and Mapping.....	13
2.1.2 Sistemas sensoriales.....	16
2.1.3 Filtros.....	16
2.1.4 Puntos de Referencia.....	17
2.2 Dispositivos Existentes.....	18
2.2.1 Google Glass.....	18
2.2.2 Microsoft Hololens 2.....	19
2.2.3 Vuzix Blade.....	20
2.2.4 Moverio BT-300.....	21
2.2.5 DreamGlass.....	22

2.2.6	Magic Leap	23
2.3	Dispositivo utilizado.....	24
3.	Implementación de realidad aumentada	25
3.1	Sistema de Comunicación.....	25
3.1.1	Servicio Web.....	26
3.1.2	Arquitectura orientada a servicios.....	29
3.2	Herramientas para Realidad Aumentada	32
3.2.1	ARKit 4	32
3.2.2	ARCore.....	36
3.2.3	ZapWorks	39
3.2.4	DroidAR.....	40
3.2.5	D'Fusion Mobile	41
3.2.6	Vuforia	42
4.	Desarrollo y resultados.....	45
4.1	Desarrollo de Aplicaciones en Realidad Aumentada	45
4.1.1	Desarrollo Nativo en Vuzix	45
4.1.2	Desarrollo Híbrido en Vuzix	46
4.2	Servicios Web	49
4.2.1	Google Maps Platform	49
4.2.2	OpenWeather API.....	51
4.2.3	Mapa Digital.....	55
4.3	Aplicaciones desarrolladas.....	57
	Conclusiones.....	61
	Bibliografía	63

Índice de figuras

	Página
Figura 1.1 Espectro de tecnologías.....	4
Figura 3.1 Arquitectura de Servicios Web	27
Figura 3.2 Pila de protocolos para servicios web	28
Figura 3.3 Elementos de Arquitectura SOA.....	30
Figura 3.4 Representación de comunicación de las API's con otros sistemas.....	32
Figura 3.5 Logo ARKit Apple	32
Figura 3.6 Logo RealityKit Apple	34
Figura 3.7 Logo Reality Composer.....	35
Figura 3.8 Logo <i>Reality Converter</i>	36
Figura 3.9 Logo <i>Reality ARCore</i>	36
Figura 3.10 Logo <i>ZapWorks</i>	39
Figura 3.11 Logo <i>Vuforia</i>	42
Figura 4.1 Formato de peticiones a Google Places API.....	50
Figura 4.2 Resultado obtenido de una petición al servicio web de Google <i>Geocoding</i> API	51
Figura 4.3 Formato de peticiones a <i>OpenWeather</i> API.....	53
Figura 4.4 Resultado obtenido de una petición al servicio web de <i>OpenWeather</i> API	54
Figura 4.5 Formato de peticiones al Mapa Digital	56
Figura 4.6 Resultado obtenido de una petición al servicio web de Mapa Digital API	56
Figura 4.7 Resultado de la petición a la API interna del sistema operativo	57
Figura 4.8 Resultados de las peticiones a los servicios web de <i>GoogleMaps</i> y <i>OpenWeather</i>	58
Figura 4.9 Pantalla Principal de la aplicación desarrollada con <i>Vuforia</i>	58
Figura 4.10 Pantalla con identificación de alguna imagen u objeto.....	59
Figura 4.11 Interfaz con los resultados obtenidos de los servicios web	59

Índice de tablas

Página

Tabla 1.1 Proyectos de Instituciones que desarrollan realidad aumentada para el sector transporte	10
Tabla 2.1 Especificaciones técnicas Google Glass Enterprise.....	18
Tabla 2.2 Especificaciones técnicas Microsoft Hololens	20
Tabla 2.3 Especificaciones técnicas Vuzix Blade.....	21
Tabla 2.4 Especificaciones técnicas Moverio BT-300	22
Tabla 2.5 Especificaciones técnicas DreamGlass.....	23
Tabla 2.6 Especificaciones técnicas Magic Leap	24

Sinopsis

Este artículo se centra en el análisis de nuevas tecnologías inmersivas y sus aplicaciones en varios problemas de transporte, en cómo funciona la realidad aumentada y qué *frameworks* se utilizan para desarrollar aplicaciones que implementan esta tecnología. Además, se describe una revisión del paradigma de desarrollo de servicios web para comprender cómo se implementan los protocolos y estándares para optimizar la comunicación entre sistemas y cómo esta idea se puede combinar con la realidad aumentada para mostrar información importante a los tomadores de decisiones. En Internet, se pueden encontrar servicios web de centros de investigación que ofrecen información relevante para el transporte como el clima y el tráfico de los vehículos, estos fueron utilizados en las aplicaciones desarrolladas para complementar los datos mostrados al usuario. Se desarrollaron dos aplicaciones para entender cómo ambas tecnologías se pueden implementar juntas, la versión de Android empleada fue 5.1 debido a que el dispositivo de realidad aumentada utilizado para los experimentos tiene esta versión. El objetivo de esta investigación es identificar las oportunidades de implementar nuevas tecnologías para resolver problemas de toma de decisiones en el sector transporte.

Abstract

This paper focuses on the analysis of new immersive technologies and their applications on several transport problems, how the augmented reality works and which frameworks are used to develop applications that implement this technology. Further, a review of web services development paradigm is described to understand how the protocols and standards are implemented to optimize the communication between systems and how this idea can be combined with augmented reality to show important information to decision makers. Web services of global research centers are available to get information on transport aspects, such as weather and vehicle traffic, these data were implemented in developed applications for this research. Two apps were developed understand how both technologies can be implemented together, the android version was 5.1 due to the augmented reality device used for experiments has this version. The goal of this research is identifying the opportunities of implement new technologies to solve decision making problems in the transport sector.

Resumen ejecutivo

Antecedentes

La realidad aumentada ha estado en crecimiento en los últimos años. Sobre todo, debido al incremento en la capacidad computacional y a la interacción con otras tecnologías como la inteligencia artificial, la realidad virtual, el Internet de las cosas y el Big Data, tecnologías que han mejorado las actividades cotidianas de las personas.

Dado su potencial para facilitar las tareas en diferentes sectores, y lejos de la predisposición inicial a ser usada en aplicaciones enfocadas en videojuegos, se ha extendido su influencia en el sector industrial, el turismo, el transporte, la medicina, la logística, entre otros.

De manera general, la tecnología inmersiva es toda aquella que tiene como objetivo incorporar realidades artificiales simuladas en un entorno real, dando al usuario una experiencia en donde tiene una interacción a través de diferentes sentidos, con algo inexistente o simulado virtualmente. La tecnología no solo es aprovechada a través del sentido de la vista, también existen algunos dispositivos que incorporan estímulos sensoriales a través de otros sentidos.

Para la finalidad de esta investigación, solo se abordarán tecnologías inmersivas relacionadas con la vista, principalmente la realidad aumentada.

Tecnología de Realidad Aumentada

Para entender mejor la tecnología relacionada para el desarrollo de sistemas que implementen realidad aumentada, es necesario dividirlo en dos aspectos, la parte tecnológica con un enfoque de “*software*” y otro con un enfoque desde el “*hardware*”. El primer aspecto hace referencia a todos los algoritmos internos que se ejecutan para analizar la información que se recolecta a través de un dispositivo como una cámara y aquellos algoritmos utilizados para predecir la ubicación y que generan información virtual a través de estos dispositivos. El segundo aspecto que se debe conocer se enfoca en las características técnicas que tienen los dispositivos que ofrecen la experiencia de realidad aumentada.

En este capítulo se tratarán estos dos aspectos fundamentales, se explicarán más detalladamente cada uno de los algoritmos que se ven involucrados en todos los aspectos de un sistema de realidad aumentada, y se explicarán a detalle algunos de los dispositivos más actuales que han desarrollado varias empresas tecnológicas y que nos pueden ofrecer esta experiencia. Finalmente, se realizará una comparación de los dispositivos existentes para identificar las diferencias entre

estos desde un enfoque económico, y la justificación del dispositivo utilizado en esta investigación.

Implementación de Realidad Aumentada

Para identificar y entender la creación de las aplicaciones que implementan realidad aumentada, es necesario conocer aquellos “*Frameworks*” existentes que facilitan o agilizan el desarrollo, ya que, al existir una gran variedad, es necesario seleccionar la mejor alternativa que sea compatible con el dispositivo utilizado para esta investigación: el *Vuzix Blade*.

Por otro lado, el sistema de realidad aumentada no es la única parte de esta investigación, también se incluye la explicación del sistema de comunicación que fue desarrollado para generar la información relevante a mostrar en el sistema. Igualmente, se explica de manera general los cambios que se realizaron en el sistema web del Mapa Digital.

El Mapa Digital es un proyecto desarrollado en el IMT-Laboratorio Nacional Conacyt en Sistemas de Transporte y Logística que permite monitorear la fluidez corredores de transporte de carga por carretera, así como el valor logístico de la infraestructura, favoreciendo la mejor toma de decisiones en la planeación de la infraestructura carretera al igual que la mejora de la operación, productividad y eficiencia logística de empresas del autotransporte de carga. Este proyecto es un sistema web o aplicación web que ofrece a través de indicadores las capacidades de análisis y visualización de datos masivos del transporte y logística en México, los resultados se pueden observar a través de la interacción directa con un mapa del país, dividido en varias secciones los cuales nos ofrecen información en diferentes capas, y donde cada una de las capas nos ofrece información relevante por tramos carreteros identificados por la Red Nacional de Caminos.

Como se mencionó anteriormente, el Mapa Digital es una aplicación web que necesita interacción humana para la visualización de los indicadores y la información que requiere el usuario. Por otra parte, es necesario adaptar este sistema para que pueda existir una comunicación con otros sistemas, es decir, una comunicación entre máquinas para la transferencia de información es por eso por lo que se recurrirá a la definición de los servicios web para esta implementación.

Desarrollo y Resultados

Para comprobar la funcionalidad y la importancia de la tecnología de realidad aumentada para ofrecer soluciones innovadoras a los sistemas de transporte, se realizó una implementación piloto de dicha tecnología. La aplicación sirvió para experimentar con la tecnología e incluir información relevante obtenida de sistemas informáticos desarrollados en otros proyectos. También se desarrolló un servicio web que obtenía información del sistema de Mapa Digital y también se incorporó la petición a otros servicios web existentes como *Google Maps API* y *OpenWeatherAPI*.

Para desarrollar aplicaciones con esta tecnología, primero se necesita definir cuáles serán las herramientas para utilizar, ya que de esto dependen cuáles serán las estrategias y metodologías para la implementación en diferentes dispositivos existentes. Para empezar, hay que considerar que para el desarrollo de cualquier aplicación para el dispositivo seleccionado (*Vuzix Blade*) se tienen dos opciones, el desarrollo nativo para aplicaciones Android (debido al sistema operativo del dispositivo) o el desarrollo híbrido que puede basarse en *frameworks* que faciliten el desarrollo y puedan ejecutar tales como los que se presentaron en el capítulo anterior.

Conclusiones

Tecnologías tales como Inteligencia Artificial, Big Data, IoT, Realidad Inmersiva, son solo algunas que actualmente se han desarrollado y han sido adoptadas por industria, academia y gobierno para facilitar actividades que pueden ser automatizadas. Dentro de estas, la realidad inmersiva ha sido explotada para la agilización de capacitación en muchas organizaciones debido a la facilidad de simular situaciones reales en entornos controlados. De su lado, la realidad aumentada ha sido utilizada ampliamente en dispositivos móviles con muchos propósitos, pero actualmente, debido a los lentes inteligentes que soportan esta tecnología, se ha incrementado su uso en la industria para facilitar actividades sin tener conocimientos previos, ya que muestra de manera virtual toda la información necesaria para que personas sin conocimiento alguno, puedan realizar dichas actividades.

El uso de realidad aumentada para identificar información importante tanto para los tomadores de decisiones de las organizaciones, así como para los transportistas, ayudarían a identificar datos de manera rápida y efectiva a un pequeño costo de los dispositivos o lentes inteligentes. De manera particular, las aplicaciones realizadas sirvieron para comprobar que la realidad aumentada tiene un gran potencial en el sector transporte y logística para tomadores de decisiones.

Introducción

El autotransporte de carga representa el 3.34% del PIB en México según datos reportados por el Inegi (Instituto Nacional de Estadística y Geografía [INEGI], 2019) y de acuerdo a las estadísticas reportadas por parte de la SCT (Secretaría de Comunicaciones y Transportes [SCT], 2019a). Este modo de transporte cubre el 82% del servicio en México. Él traslada 556 millones de toneladas al año, siendo el primer lugar en el traslado de carga a nivel nacional. Se puede considerar un modo de transporte transversal, debido a que participa en una gran diversidad de ramas de la industria, desde el traslado de materias primas a las fábricas para su producción, hasta la distribución del producto terminado a centros de distribución y usuarios finales.

El peso económico que tiene la producción y distribución de mercancías hace que los líderes en la industria y en gobierno, desarrollen estrategias que impulsen la confiabilidad del sector logístico para alcanzar la eficiencia económica tanto de las empresas, como del país. Se trata de un pilar importante para México, por lo tanto, se vuelve necesario comprender el panorama actual del país en el contexto de la operación de las cadenas de suministro.

Entre los retos para volverse más competitivo que enfrenta este modo de transporte, se encuentran el marco jurídico (Secretaría de Comunicaciones y Transportes [SCT], 2019b), así como las interrupciones a su operación generadas por la actividad criminal (Cámara Nacional de Autotransporte [CANACAR], 2019) en ciertas zonas de la república. Es bajo este panorama que, se requieren indicadores adecuados como apoyo a la toma de decisiones.

Como parte del marco de referencia del presente trabajo, se tomó en cuenta el Proyecto de Nación 2018-2024, documento en donde se hace mención sobre la importancia de la (Secretaría de Comunicaciones y Transportes, 2020):

“Construcción, modernización y conservación de la infraestructura carretera, que permita, por una parte, incrementar la competitividad del país en el contexto internacional y, simultáneamente, abatir los sobrecostos de transporte, así como construir, modernizar, reconstruir y conservar la red de caminos alimentadores y rurales, de tal manera que permita la comunicación en cualquier época del año, en especial a las comunidades que presentan los índices más altos de marginación y de población indígena.”

Bajo este contexto, el objetivo de este proyecto ha sido identificar la tecnología más adecuada para que los tomadores de decisiones en el autotransporte puedan acceder ágilmente a la información que actualmente, diversas plataformas publican. Es decir, contar con información a la mano, sin necesidad de una interacción directa

con los sistemas. Es así que en el presente documento se analiza la tecnología de realidad aumentada. Entre sus ventajas está, el proveer la posibilidad de mostrar los indicadores en tiempo real a través de dispositivos (gafas/lentes) que no interfieran con las actividades operativas de las personas que consultan la información.

Actualmente la realidad aumentada (RA) es una tecnología que se utiliza día a día en diferentes industrias para optimizar las tareas diarias de los colaboradores. Por ejemplo, a través de esta tecnología, se puede contar con una visualización detallada de cada uno de los pasos a seguir para realizar un proceso o actividad, a través de dispositivos como son las *Tablet's* o los *Smartphone's*.

A diferencia de la realidad virtual, en la cual se puede simular diversas operaciones, la realidad aumentada depende totalmente de la interacción con el mundo real, y esto es una gran ventaja para el sector del autotransporte.

La realidad aumentada permite acceder a los indicadores más relevantes para los transportistas operando sus vehículos, pero sin obligarlos a desarrollar tareas que pongan en riesgo su actividad (como soltar el volante para tomar un dispositivo y consultar la información). La realidad aumentada, facilita el acceso ágil a información que permita resolver situaciones en tiempo real y/o cumplir de mejor manera la normatividad vigente (Diario Oficial de la Federación [DOF], 2018).

Este proyecto tiene como objetivo desarrollar tecnología de realidad aumentada para comunicar la información del proyecto de Mapa Digital del Laboratorio Nacional en Sistemas de Transporte y Logística.

En cuanto al alcance del proyecto, se contempla el desarrollo e implementación de de una aplicación de realidad aumentada, que muestre información y resultados de uno de los proyectos más importantes del Laboratorio y algunos servicios web ya existentes en internet desarrollados por otros centros de investigación de reconocimiento mundial.

La metodología para esta investigación se inicia con el método inductivo, mediante el estudio de la información referente a las tecnologías inmersivas: antecedentes, características, ventajas y desventajas de estas. Posteriormente, el análisis de diferentes *frameworks* y herramientas que permitan la implementación y comunicación con los sistemas actuales. Finalmente, el desarrollo de una aplicación que implemente la tecnología de realidad aumentada y la comunicación con servicios web.

1. Antecedentes

La realidad aumentada ha estado en crecimiento en los últimos años. Sobre todo, debido al incremento en la capacidad computacional y a la interacción con otras tecnologías como la inteligencia artificial, la realidad virtual, el Internet de las cosas y el Big Data, tecnologías que han mejorado las actividades cotidianas de las personas.

Dado su potencial para facilitar las tareas en diferentes sectores, y lejos de la predisposición inicial a ser usada en aplicaciones enfocadas en videojuegos, se ha extendido su influencia en el sector industrial, el turismo, el transporte, la medicina, la logística, entre otros.

1.1 Tecnologías inmersivas

De manera general, la tecnología inmersiva es toda aquella que tiene como objetivo incorporar realidades artificiales simuladas en un entorno real, dando al usuario una experiencia en donde tiene una interacción a través de diferentes sentidos, con algo inexistente o simulado virtualmente. La tecnología no solo es aprovechada a través del sentido de la vista, también existen algunos dispositivos que incorporan estímulos sensoriales a través de otros sentidos.

Para la finalidad de esta investigación, solo se abordarán tecnologías inmersivas relacionadas con la vista, principalmente la realidad aumentada.

1.1.1 Realidad aumentada

La realidad aumentada es una técnica de integración que aprovecha procesamiento de imágenes, cómputo en tiempo real, seguimiento de movimiento, reconocimiento de patrones, proyección de imágenes y extracción de características para generar capas superpuestas digitales en el mundo real. En otras palabras, este tipo de sistemas combinan los objetos reales con objetos virtuales en ambientes reales para registrar su interacción en tiempo real (Behzadan et al., 2015).

Al inicio de la creación de esta tecnología, se buscaba mostrar vistas tridimensionales a través de pantallas que se podían montar en la cabeza, pero los primeros prototipos solo eran capaces de renderizar¹ algunos objetos con pocas líneas (Azuma et al., 2001). En la actualidad, la investigación en esta área ha

¹ El término renderización es un anglicismo usado para referirse al proceso de generar imagen a partir de un modelo 2D o 3D por medio de programas informáticos.

incrementado drásticamente en los últimos años, por lo que es posible visualizar objetos virtuales complejos en ambientes aumentados, pero, además, con capacidad para realizar actividades de interacción con objetos reales.

1.1.2 Realidad virtual

Por otro lado, la realidad virtual es una realidad simulada por computadora que replica un ambiente físico o un mundo imaginario a través de tecnología inmersiva. Es decir, reemplaza el mundo físico del usuario con un ambiente completamente virtual lo cual aísla sus receptores sensoriales visuales y audibles del mundo real. De manera general la realidad virtual es observada a través de sistemas que muestran estos objetos y pueden tener interactividad con el usuario, creando así una experiencia virtual (Nomura & Sawada, 1999).

En la actualidad, existe una gran cantidad de dispositivos de realidad virtual en el mercado, y aunque han ganado gran popularidad en la industria de los videojuegos, existen líneas de investigación que se enfocan en utilizar esta tecnología para otras áreas como la salud, capacitación de personal para diferentes industrias, o para la simulación de actividades de control de vehículos pesados o herramientas que no son tan fáciles de manejar.

1.1.3 Realidad mixta

Los desarrollos de tecnologías como realidad aumentada y realidad virtual han impulsado a las empresas a desarrollar dispositivos holográficos que se benefician la tecnología mixta, la cual puede interpretarse como una mezcla de una realidad híbrida en la cual objetos físicos y digitales coexisten e interactúan entre sí en tiempo real. El término de realidad mixta fue originalmente introducido por un artículo de Milgram y Kishino en 1994 titulado como: “A Taxonomy of Mixed Reality Visual Displays” (Milgram, 2011). Los autores describen un espectro que explica las relaciones entre ambas tecnologías el cual se puede visualizar en la Figura 1.1.



Fuente: A Taxonomy of Mixed Reality Visual Displays (Milgram, 2011)

Figura 1.1 Espectro de tecnologías

1.2 Impacto y beneficios de la tecnología de realidad aumentada en logística y transporte

El uso de la realidad aumentada ha aumentado en los últimos años debido a la capacidad de incorporar ambientes simulados para pruebas de proyectos y prototipos tecnológicos, pero no trabaja como una implementación independiente. Al contrario, funciona mejor en conjunto con otras tecnologías como inteligencia artificial, Internet de las cosas, Big Data.

1.2.1 Realidad aumentada en el transporte, una vista al pasado

La realidad aumentada ha sido implementada en diferentes modos de transporte, así como para optimizar la toma de decisiones en el sector logístico. Existe un gran número de proyectos los cuales siguen trabajando con esta tecnología o intentan incorporarla en los procesos y actividades rutinarias, debido a esto, existe un gran número de empresas dedicadas a ofrecerla como un producto o servicio que cubre las necesidades de los usuarios finales. Por otro lado, hay muchas otras instituciones y organizaciones que continúan con un desarrollo independiente y no buscan contratar o implementar esta tecnología de manera comercial. La información recopilada a continuación, no clasifica a los autores entre quienes usan o no de manera comercial dispositivos de realidad aumentada, por el contrario, se explica de manera muy general cuáles son los objetivos generales de dichos proyectos y se tratan de clasificar dependiendo del modo de transporte y la perspectiva que manejan.

1.2.1.1 Aplicación en vehículos terrestres

La realidad aumentada, principalmente en vehículos terrestres, se maneja de dos maneras: aquellos dispositivos que un usuario humano utiliza para mejorar la toma de decisiones a través de la visualización de datos importantes al manejar, o aquellos desde una perspectiva de máquinas, la cual se utiliza principalmente en vehículos autónomos.

Usuarios Humanos. - Desde una perspectiva en la cual se desea ofrecer la información necesaria para optimizar el manejo de vehículos de transporte terrestre, se han enfocado principalmente los proyectos en el uso de *HUD's (Head-up displays)*. Estos son pantallas transparentes que presentan información a los usuarios de tal manera que estos no tengan que cambiar su perspectiva para ver dicha información, desde un enfoque de realidad aumentada, dichos dispositivos se han utilizado para proveer información que reduzca o prevenga accidentes de tráfico (Takeda et al., 2018), mejorando la conciencia y optimizando la toma de decisiones de los conductores de estos vehículos (Gregoriades & Sutcliffe, 2018). También se han desarrollado sistemas que cuantifican la cantidad de distracciones potenciales que podrían tener los conductores al manejar y que podrían provocar un accidente en zonas muy cercanas a peatones (H. Kim & Gabbard, 2018) proporcionando la información necesaria del entorno para enfocarse en mejorar la atención en el

camino (Eyraud et al., 2015). Además, se han incluido algunos sistemas que generan la información espacial del entorno del conductor, no solo en zona peatonal, sino en cualquier autopista o carretera en donde se encuentre el conductor (de Oliveira et al., 2019).

Todo ello, generando información suficiente para la implementación de sistemas cooperativos, con el objetivo de brindar información a los conductores para realizar maniobras de cambios de carril seguros, visualizando la información a través de realidad aumentada de los puntos ciegos de los espejos retrovisores (Calvi et al., 2020a),(Jolovic et al., 2016) y dando un seguimiento continuo de los vehículos que se encuentran alrededor del automóvil en un determinado radio (Zimmermann et al., 2018). Para aquellos vehículos pesados que circulan por carreteras con dos carriles, con un carril para cada sentido, se han desarrollado sistemas de asistencia para que los conductores puedan visualizar la información que se obstaculiza al encontrarse detrás de estos vehículos pesados. Estos sistemas vuelven “*invisibles*” a los vehículos pesados a través de cámaras que analizan y transforman la información visible y generan un sistema de realidad aumentada en la parte trasera de los vehículos para ser mostrada a los demás conductores que circulan detrás del vehículo pesado (Gomes et al., 2012).

Por otro lado, se han desarrollado sistemas basados en modelos híbridos que apoyan a la mejor reacción de los conductores a través del incremento en la sensibilidad de los sentidos, es decir, alertas que afectan la percepción en situaciones peligrosas a través de sistemas visuales. Además de sistemas basados en sensores que evalúan y analizan el comportamiento de estos conductores (S. Kim & Dey, 2015), y sistemas que simulan diferentes situaciones reales para identificar la reacciones a estos riesgos (Hussain et al., 2013).

Aparte de estos proyectos con una perspectiva vehicular, existen proyectos que dan prioridad a la información recopilada o generada en el entorno desde una perspectiva peatonal. Es decir, proporcionan datos relevantes que incrementan la seguridad de los peatones (Tong & Jia, 2019) y ayudan a identificar la distancia y el tiempo restante para que el vehículo analizado llegue a cruces peatonales (Calvi et al., 2020b).

Por otro lado, la realidad aumentada también ha sido utilizada en los sistemas de transporte no motorizados, donde se han desarrollado plataformas para mejorar la calidad del viaje para ciclistas. De este modo, vuelven más atractivo el uso de las bicicletas y aumentan la seguridad del entorno (Ginters, 2019).

Usuarios Máquina.- Una de las aplicaciones principales de la realidad aumentada es la inclusión de información representativa en los automóviles autónomos(Ajorloo et al., 2020). En la actualidad, este tipo de vehículos ha ganado una gran importancia y ha llamado la atención de muchos centros de investigación a nivel mundial que buscan acelerar su desarrollo. La tecnología principalmente se ha enfocado en el desarrollo de técnicas para disminuir el riesgo de accidentalidad al detectar la información de los peatones que se encuentren en el camino o ruta del

vehículo (Hartmann et al., 2018), también se ha enfocado el uso de esta tecnología en la identificación y generación automática de información digital que represente el entorno a través de los otros vehículos que se encuentran alrededor (Li & Nashashibi, 2011)(Zhou et al., 2020).

Por otro lado, otra de las aplicaciones principales es la simulación de pruebas al generar situaciones de riesgo en las cuales podrían estar involucrados dichos vehículos(Feng et al., 2020). Igualmente para generar situaciones realistas que den soporte a los sistemas para entrenar y mejorar el rendimiento de estos automóviles (Koduri et al., 2018). Cabe destacar que la mayoría de los proyectos que hasta hoy han implementado realidad aumentada para vehículos autónomos, se enfocan en entrenamiento y experimentación de los sistemas de inteligencia artificial que los controlan, y solo son un apoyo para la simulación, es decir, no son la base principal de dichos sistemas.

1.2.1.2 Perspectiva de infraestructura

La infraestructura en el transporte juega un rol importante (Secretaria de Comunicaciones y Transportes [SCT], 2013). Al ser un factor para el desarrollo económico; brinda comunicación entre los centros de población regionales, centros de producción y consumo; además de ser un factor para elevar la competitividad, reduciendo costos y tiempos de transporte, facilitar el acceso a mercados e integrar cadenas productivas, facilitar el acceso a servicios de educación y salud; entre otras características importantes.

Desde esta perspectiva también se han desarrollado proyectos importantes que muestran avances tecnológicos para mejorar la infraestructura, proyectos para identificar nueva infraestructura y proyectos para determinar daños en la misma o el estado en el que se encuentran. Estos proyectos tienen como objetivo la generación de sistemas que se basan en realidad aumentada para integrar información digital para identificar y optimizar los análisis que se tienen que realizar, un ejemplo de estos proyectos es una aplicación de celular para identificar los defectos en zonas o estructuras carreteras basándose en la ubicación geoespacial del teléfono y analizando la información a través de visión computacional. Dichos aparatos, posteriormente se conectan a un sistema que se encuentra en un servidor, el cual recopila la información para la generación de reportes y el monitoreo en tiempo real de la infraestructura, esta información es analizada por los tomadores de decisiones que se encargan del mantenimiento carretero (Chang et al., 2012), aunque también puede ser utilizada para dar un seguimiento geoespacial a estos detalles [27][28].

Por otro lado se han desarrollado sistemas que apoyan a la identificación de zonas, previo a la construcción de alguna autopista o carretera, dicho sistema proporciona datos a través de una aplicación celular y muestra los posibles resultados finales de las futuras construcciones antes de comenzarlas (Mallela et al., 2020). A pesar de que estos sistemas tienen problemas de precisión, es un gran avance para la integración de esta tecnología ya que generan información con pocos recursos. También se han desarrollado aplicaciones que optimizan el manejo y control del

equipamiento a utilizar en la construcción esta infraestructura (B. Kim et al., 2012) ya que a través de la realidad aumentada, se puede capacitar, monitorear o dar instrucciones de manera interactiva. Además, existen algunos otros proyectos enfocados en la reconstrucción de carreteras, es decir, realizar análisis de imágenes y videos para reconstrucción tridimensional de las carreteras que se identifiquen en estos tipos de archivos (Golparvar-Fard et al., 2015).

1.2.1.3 Sistemas de navegación

La realidad aumentada ha sido una tecnología bien aceptada por diferentes áreas en el sector transporte, ya sea simulando situaciones para identificar el comportamiento de vehículos autónomos o a través de sistemas capaces de aportar información a conductores o peatones para mejorar la toma de decisiones, la realidad aumentada no se ha desarrollado solo para el transporte terrestre. En los sistemas de navegación marítima ha aportado soluciones que mejoran los sistemas de seguridad y apoyan a evitar accidentes marítimos (OH et al., 2016). También ha servido para el desarrollo de sistemas de monitoreo en tiempo real para mejorar el rendimiento de los operadores a través de dispositivos llamados WIAR (*Wearable Immersive Augmented Reality*), los cuales proporcionan información del entorno, analiza los datos, y genera recomendaciones a los operadores que manejan los sistemas de seguridad críticos (Grabowski et al., 2018). Por otro lado, además de la gran variedad de desarrollos civiles, la realidad aumentada también ha sido una tecnología empleada para ofrecer simuladores y prototipos de prueba para sistemas militares (Wickens et al., 2018).

Las ubicaciones se han vuelto un factor importante para los sistemas inteligentes debido a la información que se puede generar partiendo de estos lugares, es por eso por lo que existen diferentes tecnologías para censar las ubicaciones tales como el GPS, el cual se ha vuelto el sistema más importante para posicionamiento y navegación, aunque el problema más grande que tiene es el posicionamiento en interiores. Para esto se ha utilizado otro tipo de tecnologías como RFID, UWB, Bluetooth y WiFi. Pero en la actualidad, gracias a la importancia que se le ha dado a los sistemas de realidad aumentada, se puede utilizar como una herramienta de soporte para esta tarea (Xu et al., 2020).

1.2.2 Proyectos e instituciones que trabajan con realidad aumentada

En la actualidad, existen varios proyectos en diferentes universidades y centros de investigación a nivel mundial que se encargan del desarrollo de la tecnología de realidad aumentada enfocada en el futuro del transporte. Dichas instituciones han obtenido presupuestos de diferentes programas y han tenido colaboración con otras instituciones con un enfoque industrial para implementar dichas soluciones. En la **¡Error! No se encuentra el origen de la referencia.** se muestran algunos de estos proyectos que se han desarrollado en los últimos años y algunos que continúan en desarrollo, pero cabe aclarar que estos son solo algunos de los proyectos que están utilizando esta tecnología como una herramienta para optimizar las operaciones

logísticas y de transporte. Algunos otros son solo una exploración y presentación de la tecnología, mientras que otros ya son aplicaciones muy específicas para alguna actividad o proceso.

Como se ha comentado a lo largo de este capítulo, la realidad aumentada no se utiliza por si sola para la solución de los problemas. Se utiliza como un complemento tecnológico y se combina con algunas otras tecnologías como realidad virtual, internet de las cosas, inteligencia artificial, visión computacional, entre otras, para ofrecer los mejores resultados posibles.

Tabla 1. 1 Proyectos de Instituciones que desarrollan realidad aumentada para el sector transporte

Proyecto	Instituciones	Inicio	Fin	Objetivo
Next generation test methods for active safety function (Grante, 2012)	Volkvo Technology AB (Research Center)	2014	2014	Desarrollo de un framework que modele situaciones de prueba para evaluar sistemas de seguridad de automóviles en diferentes ambientes.
Deep-Learning Based Trajectory Forecast for Safety of Intersections with Multimodal Traffic (Fan, 2017)	Universidad de Texas	2017	2019	Explorar la realidad aumentada con procesamiento de imágenes basados en Deep-Learning para la estimación de la trayectoria de los conductores en una intersección.
CAV Testing Scenario Design and Implementation using Naturalistic Driving Data and Augmented Reality (Y. Feng & Liu, 2017)	Instituto de Investigación en Transporte de la Universidad de Michigan	2017	2020	Implementación de un sistema de realidad aumentada para evaluar la eficiencia del manejo seguro de los vehículos autónomos.
Augmented Reality (AR) in Life-Cycle Management of Transportation Infrastructure Projects (Szary & Kumapley, 2018)	Centro de Infraestructura y Transporte Avanzados (Research Center)	2018	2019	Desarrollar una estrategia para presentar la tecnología de realidad aumentada como una herramienta que beneficie los sistemas de infraestructura del transporte.
Development of An Augmented Reality Environment for Connected and Automated Vehicle Testing (Liu et al., 2019)	Instituto de Investigación en Transporte de la Universidad de Michigan	2018	2019	Integración de un sistema de realidad aumentada para las pruebas en <i>McCity</i> y los vehículos autónomos que se encuentran desarrollando en ese lugar.
Visualizing Sea Level Rise Impacts in Transportation Planning(Wolshon, 2018b)	Centro de Investigación de la Costa del Golfo para la Resiliencia en Evacuación y Transporte	2018	2019	Evaluación de nuevas tecnologías para ayudar a residentes de comunidades en el sur de Florida para entender mejor los impactos del nivel del mar en la infraestructura de transporte y en la misma comunidad.
Assessing Alternative Approaches for Conveying Automated Vehicle Intentions (Doerzaph, 2018)	Instituto de Transporte de Virginia Tech	2018	2020	Desarrollo y evaluación de una interfaz de realidad aumentada para aumentar la conciencia situacional del sistema de conducción de vehículos terrestres y su entorno.
Engaging the Business and Tourism Industry in Visualizing Sea Level Rise Impacts to Transportation Infrastructure in Waikiki, Hawaii(Wolshon, 2018a)	Universidad del Estado de Louisiana	2018	2020	Basado en el proyecto desarrollado para el sur de Florida(Wolshon, 2018b), este proyecto refleja la investigación en la zona de Waikiki, Honolulu, Hawaii
Virtual Tour(VT), Informational Modeling(IM), and Augmented Reality(AR) for Visual Inspections(VI) and Structural Health Monitoring(SHM)(Szary et al., 2018)	Universidad de Princeton	2018	2020	Crear un método y un software asociado para la integración, documentación, acceso y visualización de inspecciones y monitoreo en tiempo real de la infraestructura.
Augmented Reality Enhancing the Inspections of Transportation Infrastructure: Research, Education, and Industry Implementation (Moreau et al., 2019)	Universidad de Nuevo Mexico	2019	2019	Uso de interfaces de realidad aumentada para crear escenarios que mejoren las operaciones diarias o decisiones gerenciales para el desarrollo de infraestructura de transporte.
Augmented Reality Testing Environment Development – Phase II (Liu & Feng, 2019)	Instituto de Investigación en Transporte de la Universidad de Michigan	2019	2020	Integración de un sistema de realidad aumentada para las pruebas en <i>McCity</i> y los vehículos autónomos que se encuentran desarrollando en ese lugar, fase 2(Liu et al., 2019).

Proyecto	Instituciones	Inicio	Fin	Objetivo
<i>Augmented Weather Interfaces Project (AWIP)</i> (Caldwell et al., 2019)	Universidad Purdue, Escuela de Ingeniería Industrial, Instituto de Tecnología de Florida, Universidad del Estado de Iowa	2019	2021	Desarrollo de una App que provea información para notificar a los pilotos de vehículos aéreos la situación climatológica.
<i>SPR-4424: Evaluation of Current Technologies for: Training, Web Apps and New Technologies</i> (Iyer & Dunlop, 2019) <i>Augmented Reality Transit Dispatcher Interface</i> (Stanton, 2019)	Universidad Purdue / Departamento de Transporte de Indiana JHRP MACRO, Ross Baruzzini (Research Center)	2019	2021	Desarrollo de sistemas para entrenamiento a través de diferentes tecnologías para el sector transporte y de infraestructura Implementación de una interfaz para el control y seguridad del tránsito en sistemas ferroviarios.
<i>Augmented Bridge Inspection with Augmented Reality and Haptics-based Aerial Manipulation</i> (Oh, n.d.)	Universidad de Nevada	2020	2021	Probar, evaluar, verificar y validar, el mantenimiento y las inspecciones de puentes a través de vehículos aéreos basados en sistemas hápticos y realidad aumentada para ofrecer nuevas capacidades.
<i>DeepScenario: City-Scale Scenario Generation for Automated Driving System Testing & Evaluation</i> (Liu et al., 2020)	Instituto de Investigación en Transporte de la Universidad de Michigan/Universidad de Michigan, Ann Arbor	2020	2021	Se pretende generar una plataforma de simulación que genere escenarios reales con la escala de una ciudad para probar y evaluar el comportamiento de vehículos autónomos.
<i>Interfacing Synchrono and NADS for Virtual Simulation of Conventional & Connected and Autonomous Vehicles</i> (Schwarz et al., 2020)	Universidad de Iowa, Universidad de Wisconsin	2020	2021	Establecer una simulación que genere escenarios de tráfico con varios vehículos autónomos para evaluar el ambiente y el comportamiento de dichos vehículos.
<i>Smart Glasses for Improving Mobility of Low Vision People</i> (Cai, 2021)	Universidad de Carnegie Mellon	2020	2021	Desarrollar un prototipo para lentes inteligentes que implementen realidad aumentada para mejorar la visión de personas con problemas visuales y mejorar su conducción.
<i>Bridge Cracks Monitoring: Detection, Measurement, and Comparison using Augmented Reality</i> (Mousa, 2020)	Universidad de Nuevo México	2020	2022	Desarrollo de un método preciso para la recolección de datos a través de herramientas de realidad aumentada, datos que servirán para la identificación y el análisis del estado de los puentes.
<i>Using Augmented Reality to Help Older Adults Make Safe Road-Crossing Decisions</i> (Rector et al., n.d.)	Universidad de Iowa	2020	2022	Desarrollo de un sistema basado en realidad aumentada para indicar visualmente que espacios entre vehículos son seguros o inseguros para cruzar, y apoyar en esa decisión a adultos de edad avanzada.
<i>Fatigue Crack Inspection Using Computer Vision and Augmented Reality</i> (Jawed, 2021)	Universidad de Kansas	2021	Activo	Desarrollo de sistema de visión computacional con realidad aumentada para detección de grietas por fatiga para puentes vehiculares.

Fuente: Elaboración propia

2. Tecnología de Realidad Aumentada

Para entender mejor la tecnología relacionada para el desarrollo de sistemas que implementen realidad aumentada, es necesario dividirlo en dos aspectos, la parte tecnológica con un enfoque de “*software*” y otro con un enfoque desde el “*hardware*”. El primer aspecto hace referencia a todos los algoritmos internos que se ejecutan para analizar la información que se recolecta a través de un dispositivo como una cámara y aquellos algoritmos utilizados para predecir la ubicación y que generan información virtual a través de estos dispositivos. El segundo aspecto que se debe conocer se enfoca en las características técnicas que tienen los dispositivos que ofrecen la experiencia de realidad aumentada.

En este capítulo se tratarán estos dos aspectos fundamentales, se explicarán más detalladamente cada uno de los algoritmos que se ven involucrados en todos los aspectos de un sistema de realidad aumentada, y se explicarán a detalle algunos de los dispositivos más actuales que han desarrollado varias empresas tecnológicas y que nos pueden ofrecer esta experiencia. Finalmente, se realizará una comparación de los dispositivos existentes para identificar las diferencias entre estos desde un enfoque económico, y la justificación del dispositivo utilizado en esta investigación.

2.1 Algoritmos

2.1.1 Simultaneous Localization and Mapping

Localización y Mapeo Simultáneos o por sus siglas en inglés: “*Simultaneous Localization and Mapping (SLAM)*”, más que una tecnología es un problema en geometría computacional y robótica, el cual representa la construcción y actualización de un mapa que describe un ambiente desconocido, mientras se hace el seguimiento a un agente que se encuentra ubicado en ese ambiente (Durrant-whyte & Bailey, 2006). La razón de que se incluyera esta información es debido a que los algoritmos que se han desarrollado a lo largo de los años para resolver este problema han sido ampliamente utilizados para mejorar la implementación de realidad aumentada en diferentes tipos de dispositivos, donde cada dispositivo se identifica como el agente, y el entorno es toda aquella información que rodea a dicho agente. Para identificar la tecnología que se utiliza actualmente para generar la información que los algoritmos necesitan para resolver el problema, es necesario conocer más a detalle de *SLAM* e identificar las estrategias de solución existentes.

La idea de este problema surgió como el proceso mediante el cual un robot móvil puede construir un mapa de un entorno y, al mismo tiempo, utilizar este mapa para

deducir su ubicación, pero tiene la particularidad de que el mapa y la posición de este robot son desconocidos. Otra característica viene dada por el hecho de que este agente tiene un modelo cinemático conocido y se desplaza por un entorno desconocido poblado de puntos de referencia artificiales o naturales. Para lograr el objetivo se requiere una estimación simultánea de las ubicaciones del robot y del punto de referencia. El problema del SLAM consiste en encontrar una representación adecuada para los modelos de observación y de movimiento de dicho agente, y para lograr esto, el robot debe estar equipado con un sistema sensorial capaz de tomar medidas de la ubicación relativa entre los puntos de referencia y el propio agente (Durrant-whyte & Bailey, 2006).

En la realidad aumentada, el robot se ve representado como un dispositivo que va a contener la aplicación y este va a realizar ciertos movimientos dependiendo de lo que el usuario desee, entonces, se tiene que interpretar la posición y orientación para que se puedan generar objetos virtuales en el ambiente y se ajusten de acuerdo a los movimientos del dispositivo. El dispositivo cuenta con una cámara con la cual se puede apreciar el entorno, y la aplicación de realidad aumentada, incorpora los objetos virtuales generados de acuerdo con lo que se muestre en el entorno, existiendo una interacción entre la realidad y lo virtual.

2.1.1.1 Metodología

Para solucionar el problema de *SLAM* se sigue un patrón que se repite a lo largo del proceso de construcción del mapa, esta serie de pasos comienza al final de cada movimiento que realiza el dispositivo (Naminski, 2013). Los pasos son descritos a continuación:

- 1. Lecturas de odometría y predicción de la ubicación.** - El primer paso es el proceso de recolectar información sobre la posición actual del dispositivo para almacenarla en un vector de control. Este vector es una matriz que contiene la información de cómo afectan los movimientos a la posición. La información proporcionada detalla los cambios de orientación y de distancia recorrida por el dispositivo desde el último estado registrado, y esta información se utiliza para generar una predicción sobre la ubicación dentro del mapa y la dirección en la que se encuentra.
- 2. Lecturas de los sensores y asociación de datos.** - El siguiente paso es recolectar la información de los puntos de referencia visibles desde la posición actual del robot, esta información representa la distancia y el rumbo de estos puntos con respecto al robot. Los valores se utilizan junto con las estimaciones de posición almacenadas de los puntos de referencia para estimar la posición del robot mediante triangulación. En este paso se genera el proceso de asociación de datos, donde se asocia la información de los puntos de referencia actualmente visibles con los que ya han sido observados. Este paso es importante ya que, si el robot no puede identificar

un punto de referencia visible con otro ya observado, no puede utilizar la información almacenada para predecir su ubicación.

3. **Correlación de la posición.** - Este paso es el encargado de estimar una nueva posición, para eso se utiliza una distribución de probabilidad para su ubicación. La estrategia de cálculo es realizar una combinación de la ubicación por triangulación a partir de los puntos de referencia con la información estimada del robot de acuerdo con su posición y orientación, la fórmula para esto depende mucho del algoritmo utilizado.
4. **Actualización de los puntos de referencia.** - El paso siguiente es actualizar las ubicaciones estimadas de los puntos de referencia, para eso se utiliza la información de distancia y rumbo recolectados por los sensores y la estimación de la posición actual del robot. Las correlaciones entre las ubicaciones de estos puntos también se actualizan, al igual que la información del mapa que representa el entorno.
5. **Agregar nuevos puntos de referencia.** - El último paso es añadir puntos de referencia basándose en la nueva posición del robot. Al determinar y ubicar estos nuevos puntos de referencia, se añaden a la lista, posteriormente se generan y almacenan las correlaciones entre los puntos nuevos con los ya existentes.

Además de los pasos anteriormente descritos, un aspecto crucial para resolver con éxito el problema de *SLAM* es la definición de los puntos de referencia, los cuales se obtienen del entorno mediante el proceso de extracción de características. Los algoritmos utilizados para este proceso de manera general obtienen muchas características de la información las cuales no siempre son las más adecuadas, entonces es necesario seleccionar de ese conjunto de datos las más representativas. Para este proceso existen algoritmos como la extracción de puntos de referencia (*Spike Landmark Extraction*) y el consenso de muestreo aleatorio (*RANSAC-Random Sampling Consensus*) (Naminski, 2013).

Spike Landmark Extraction. - Este método se enfoca en extraer los puntos de referencia a través de los sensores de láser y los datos de distancia que recolectan, esto permite identificar y analizar los datos de áreas con cambios extremos, con el objetivo de identificar puntos de referencia como esquinas o aquellos objetos que puedan servir de referencia como patas de una mesa o silla. Este método tiene el problema de fallar en situaciones o ambientes con poco cambio, ya que, al no identificar diferencias en el ambiente, lo maneja como un mismo objeto o superficie.

Random Sampling Consensus. - Es el proceso de extracción de líneas formadas en el entorno para usarlas como puntos de referencia y es útil para extraer cambios en superficies como paredes diferentes. Estas líneas se usan como puntos de referencia extrayendo en cada posición el punto de la línea más cercana al robot

para realizar un análisis de alcance y rumbo para las actualizaciones de las ubicaciones posteriores.

2.1.2 Sistemas sensoriales

Muchos grupos de investigación han trabajado en resolver el problema de *SLAM*, y los sensores más comunes utilizados en estas investigaciones se pueden categorizar en tres tipos: i) sensores basados en láser; ii) sensores basados en sonares; y iii) sistemas basados en visión (Aulinas et al., 2008). Además, se han utilizado otras tecnologías como el GPS (*Global Position System*). Sin embargo, todas estas tecnologías de diferentes sensores tienen ciertos errores de medición o también conocidos como ruido, y además tienen limitantes para la navegación en el ambiente generado, como lo es la luminosidad o que el ruido no puede penetrar ciertas paredes de manera adecuada. A continuación, se describen con mayor detalle:

- **Sistemas basados en láser.** - estos sistemas tienen sensores láser de gran precisión, de manera general, su funcionamiento se basa en enviar un pulso láser en un haz estrecho hacia el objetivo y volver al punto emisor. Un ejemplo muy claro de este tipo de sistemas son los LIDAR (*Light Detection and Ranging* o *Laser Imaging Detection and Ranging*).
- **Sistemas basados en sonares.** - estos sistemas son rápidos y proporcionan mediciones y reconocimiento con información casi tan precisa como los sistemas basados en visión, pero con la falta de datos en cuanto apariencia se refiere. Sin embargo, dependen de los sensores inerciales como los odómetros, y esto provoca que un error repercuta en grandes efectos sobre el cálculo de las posiciones posteriores.
- **Sistemas basados en visión.** - este tipo de sistemas tienen un gran alcance y alta resolución, pero el costo computacional es relativamente alto y las características visuales detalladas son más difíciles de extraer y analizar. Estos sistemas se enfocan en el cálculo de la estructura 3D, y con respecto a robótica, la ubicación de las características, la posición y la orientación del robot.

2.1.3 Filtros

El estudio para la solución del problema *SLAM* ha tenido muchas aproximaciones desde que empezó el auge de la robótica, y desde los años 90's se han estudiado enfoques probabilísticos para resolverlo (Thrun, 2003). Los ejemplos más comunes utilizados han sido los Filtros de Kalman (*KF-Kalman Filters*), Filtros de Partículas (*PF-Particle Filters*) y Maximización de esperanza (o valor esperado) (*EM-Expectation Maximization*). Estas tres técnicas son derivaciones matemáticas de la regla recursiva de Bayes, y se han vuelto dominantes debido a que el mapeado del robot es caracterizado por la incertidumbre y el ruido de los sensores, y estos algoritmos probabilísticos toman en cuenta el modelado de diferentes fuentes de

ruido y sus respectivos efectos en las mediciones finales. **Los Filtros de Kalman.**- son filtros de Bayes que representan las ubicaciones posteriores a través de distribuciones multivariantes unimodales (Gaussianas) que pueden representarse de forma compacta mediante una pequeña cantidad de parámetros (Davison & Murray, 2002). La idea de este algoritmo se basa en la suposición de que las funciones de transición de los estados y mediciones de los robots son lineales con un ruido gaussiano añadido, y las posiciones posteriores también tienen una distribución gaussiana.

- **Los Filtros de Partículas.**- también son conocidos como método secuencial de Monte Carlo (SMC) y son un filtro de Bayes recursivo que se implementa en simulaciones de Monte Carlo (Montemerlo et al., 2002). Se ejecuta a través de un conjunto de agrupaciones de puntos aleatorios, llamados partículas, que representan las ubicaciones posteriores. A diferencia de los Filtros de Kalman, este método representa la distribución mediante un conjunto de muestras extraídas de esta distribución, lo que la hace capaz de manejar sensores no lineales y con ruido no gaussiano. El problema recae en el crecimiento de la complejidad computacional dependiente de la cantidad de nuevos puntos de referencia que se van detectando, por lo que no es adecuado para aplicaciones en tiempo real. Por otro lado, este método se ha aplicado de manera exitosa en el problema de la localización del robot con un enfoque de posición y orientación, pero no a la construcción de mapas ni en la determinación de posición y orientación de los puntos de referencia.
- **Maximización de Esperanza.**- La estimación a través de EM es un algoritmo que se desarrolló en el contexto del método de máxima verosimilitud, y ofrece una solución óptima ideal para la construcción de mapas cuando se conoce la orientación y posición del robot (Burgard et al., 1999). El algoritmo itera en dos pasos, un paso de esperanza (o valor esperado) el cual calcula la posición posterior del robot en un mapa dado, y el siguiente paso es calcular el mapa más probable en función de las esperanzas de la posición y orientación del robot. El resultado final de estos pasos es una serie de mapas cada vez más precisos. Una ventaja de este algoritmo con respecto a los demás, es que aborda el problema de la asociación de datos ya que constantemente localiza al robot en relación con el mapa, que se refuerza con las iteraciones repetitivas. El problema es que el procesar los mismos datos varias veces no es eficiente para aplicaciones en tiempo real debido a la complejidad.

2.1.4 Puntos de Referencia

Los puntos de referencia son puntos reconocibles en el entorno que rodea al dispositivo que se utilizan para la localización, un ejemplo son las paredes o las mesas. La correlación que existe entre estos puntos es crucial para generar buenos resultados en *SLAM*. La información generada por los sensores es poco confiable como medida para determinar la ubicación de los puntos de referencia desde la

posición y la orientación estimada del robot, sin embargo, la información de la distancia entre dos puntos de referencia es más fiable, debido a que mientras más avanza el robot tomando en cuenta estos puntos de referencia y sus distancias relativas desde diferentes posiciones en el mapa, la correlación entre éstos converge (BAILEY & DURRANT-WHYTE, 2006).

2.2 Dispositivos Existentes

En la actualidad hay muchos equipos de diferentes compañías que han logrado implementar con éxito la realidad aumentada, cada uno de estos dispositivos tiene sus propias características técnicas que lo hacen únicos con respecto a los demás en el mercado. En este capítulo se explican algunos de los productos sobresalientes del mercado, pero eso no indica de ninguna manera que sean las únicas marcas que hayan desarrollado dispositivos de realidad aumentada.

2.2.1 Google Glass

Este dispositivo de visualización de realidad aumentada fue desarrollado por *Google* y presentado en el congreso I/O de 2012. La idea original era mostrar información disponible en teléfonos inteligentes sin utilizar las manos, permitiendo el acceso a través de comandos de voz. Aunque *Google* ha dejado de producir este tipo de dispositivos directamente, tiene una colaboración con un conjunto de empresas que se encargan de la venta y soporte. Las aplicaciones desarrolladas para *Google Glass Enterprise* (Google, 2020e) dependen exclusivamente de servidores externos y son manejados y desarrollados por terceros. Las especificaciones técnicas más actuales de estos dispositivos son mostradas en la Tabla 2.1.

Tabla 2.1 Especificaciones técnicas Google Glass Enterprise

HW/SW	Información
Procesador	Qualcomm Snapdragon XR1
Sistema operativo	Android Oreo 8.1
Memoria	3GB LPDDR4 32GB eMMC Flash
Wi-Fi	IEEE 802.11a/g/b/n/ac, dual-band
Bluetooth	Bluetooth 5.0
Cámara	8 Megapixel con sensor de colores Campo de visión de 83° Apertura f/2.4 Enfocado a 0.6m Hasta 1080p30 de video
Display	640 pixeles x 360 pixeles RGB
Salida de audio	Mono Speaker Audio USB Bluetooth (HFP)
Micrófono	3 micrófonos formadores de haz de campo cercano

HW/SW	Información
Touch	Panel táctil de gestos multitáctiles
Carga de datos	Compatible con USB PD 2.0 (carga rápida de hasta 1,5 A a 5 V) Transferencia de datos USB 2.0 Conector compatible con USB-C
LED	Privacidad (cámara) LED verde, alimentación (trasera) LED blanco
Batería	800 mA-h (2880 C)
Sensores inerciales	Acelerómetro de 3 ejes Giroscopio de 3 ejes Magnetómetro de 3 ejes
Materiales	Nylon de resina
Rugosidad	IP53 (resistente al agua pulverizada y al ingreso de polvo limitado)
Temperatura de funcionamiento	0° C a 35° C (32° F a 95° F)
Temperatura de almacenamiento	-20° C a 45° C (-4° F a 113° F)
Humedad relativa	5% a 95% sin condensación
Peso sin marco	46 g
Dimensiones sin marco	212 mm x 57 mm x 29 mm (desplegado) 182 x 55 mm x 29 mm (plegado)

Fuente: Elaboración propia basado en Google Glass Enterprise (Google, 2020f)

2.2.2 Microsoft Hololens 2

Después de varios años de haber entregado las primeras versiones de Microsoft Hololens (Microsoft, 2020a), finalmente en 2019 se presentó la nueva generación de estos dispositivos, los *Microsoft Hololens 2*, durante el Congreso Mundial de Móviles (*MWC*²), uno de los congresos considerados más importantes a nivel mundial en este sector.

Hololens 2 ofrece una experiencia de realidad mixta de manera cómoda e inmersivas, con la confiabilidad, seguridad y escalabilidad de la nube y los servicios de IA de *Microsoft*. Tiene un campo de visión amplio donde se pueden observar textos y detalles de imágenes en 3D de manera cómoda, tiene un mecanismo de ajuste para no quitarse completamente las gafas, e incluye una funcionalidad para tocar agarrar o mover los hologramas, además también incluye comandos de voz que tienen un filtro para ruidos industriales e incluyen micrófonos inteligentes que tienen un procesamiento de lenguaje natural a través de la tecnología de *Microsoft*. Las especificaciones técnicas más actuales de estos dispositivos se exponen en la Tabla 2.2.

² *MWC-Mobile World Congress* (Delson Group Inc., 2019).- Es un congreso anual enfocado al mundo de la comunicación móvil. Es considerado el más importante del mundo en este sector. Suele ser un espacio donde se presentan avances en comunicaciones inalámbricas y móviles.

Tabla 2.2 Especificaciones técnicas Microsoft Hololens

HW/SW		Información
Pantalla	Óptica	Lentes holográficos transparentes (guías de ondas)
	Resolución	2000 motores de luz 3:2
	Densidad holográfica	>2500 radiantes (puntos de luz por radián)
	Representación basada en la vista	Muestra optimización para la posición de los ojos en 3D
Sensores	Seguimiento de la cabeza	4 cámaras ligeras visibles
	Seguimiento ocular	2 cámaras de IR
	Profundidad	Sensor de profundidad de tiempo transcurrido de 1MP
	IMU	Acelerómetro, giroscopio y magnetómetro
	Cámara	Fotos de 8MP, video de 1080p30
Audio y voz	Matriz de micrófono	5 canales
	Bocinas	Sonido espacial integrado
Humano	Seguimiento de la mano	Modelo articulado a dos manos, manipulación directa
	Seguimiento ocular	Seguimiento en tiempo real
	Voz	Comando y control en el dispositivo, lenguaje natural con conectividad a Internet
Entorno	Seguimiento 6DoF	Seguimiento de posición de escala mundial
	Cartografía espacial	Malla de entorno en tiempo real
	Captura de realidad mixta	Holograma mixto y fotos y videos del entorno físico
	Procesador	Qualcomm Snapdragon 850
Cálculo y Conectividad	HPU	Unidad de procesamiento holográfico personalizado de segunda generación
	Memoria	DRAM LPDDR4x 4 GB
	Almacenamiento	UFS 2.1 de 64GB
	Wi-Fi	802.11ac 2x2
	Bluetooth	Bluetooth 5.0
	USB	USB Type-C
Físico	Peso	566

Fuente: Elaboración propia basado en Microsoft Hololens (Microsoft, 2020b)

2.2.3 Vuzix Blade

Las gafas inteligentes *Vuzix Blade* ofrecen una conexión manos libres del mundo digital con el mundo real, proporcionando acceso a información de ubicación, recolección de datos, comunicación remota con audio y video, entre otros beneficios. Las *Blade Upgraded Smart Glasses* (Vuzix, 2020a) son una versión mejorada de estos dispositivos que incluyen características como una cámara de 8 megapíxeles con enfoque automático, altavoces estéreo integrados y control de voz avanzado.

La óptica transparente *Waveguide* fusiona las instrucciones digitales con las tareas del mundo real, eliminando las distracciones y la oclusión de la visión y reduciendo así las tasas de error. Al mismo tiempo, el dispositivo inalámbrico es una herramienta de comunicación audiovisual para aplicaciones de asistencia remota. La empresa Vuzix desarrolladora de este dispositivo cuenta con varias familias de productos con diferentes enfoques tecnológicos, pero todos con la finalidad de incorporar esta idea de realidad aumentada como gafas inteligentes (*Vuzix Blade*) o complementos de visualización (*M-Series Smart Glasses*). En esta investigación se explicará a detalle el dispositivo *Vuzix Blade*, ya que es el más adecuado para las comparaciones con las demás tecnologías. Las especificaciones técnicas más actuales de estos dispositivos son:

Tabla 2.3 Especificaciones técnicas Vuzix Blade

W/SW	Información
Óptica y electrónica	Óptica transparente basada en guías de ondas
	Pantalla vibrante a todo color
	Monocular para el ojo derecho
	CPU ARM de cuatro núcleos
	Cámara de 8 megapíxeles (con enfoque automático)
	Sistema operativo Android 5.1
	Lentes con protección UV total y certificaciones de seguridad
	1 GB de RAM
	8 GB de Memoria Flash
Controles	Control por voz - multilingüe
	Panel táctil con gestos
	Rastreadores de movimiento de la cabeza
	Aplicación complementaria para dispositivos Android e iOS
Audio	Funcionalidad BT completa
	Micrófonos con cancelación de ruido
	Audio incorporado
Batería	Baterías internas recargables LiPo
Conectividad	Ranura de almacenamiento micro SD de 8G
	Wi-Fi de 2,4 GHz y Bluetooth inalámbrico
	Micro USB

Fuente: Elaboración propia basado en Vuzix Blade Smartglasses (Vuzix, n.d.)

2.2.4 Moverio BT-300

Las gafas Moverio BT-300 (Epson, 2020a) cuentan con la tecnología más avanzada de Epson de pantalla digital OLED (diodo orgánico emisor de luz) con base de silicio, lo que convierte al dispositivo en las *smart glasses* transparentes más ligeras con pantalla OLED2. Cuenta con un CPU de cuádruple núcleo Intel Atom x5 1,44 GHz, además de una gran variedad de sensores incluyendo un sensor de movimiento de visor con controlador, GPS, micrófono y cámara, entre otros sensores para crear la experiencia de realidad aumentada. Estos dispositivos destacan además por ser los primeros que permiten acceder a una vista en primera persona en las tareas de pilotaje de drones, mejorando la forma en la que se graban vídeos y se toman

imágenes. Además del control táctil que incorporan las propias gafas, las Moverio ofrecen a los usuarios un mando externo que puede ser útil a la hora de desarrollar determinadas tareas a distancia. Las especificaciones técnicas más actuales de estos dispositivos son:

Tabla 2.4 Especificaciones técnicas Moverio BT-300

HW/SW		Información
Aspecto	Tipo de dispositivo	Si-OLED (Silicon - Organic Light-Emitting Diode)
	Método conductor	Mono Crystalline Silicon Active Matrix
	Tamaño de pantalla	0,43 pulgada panel ancho (16:9)
	Número de píxeles	921.600 píxeles (1.280x 720) x RGB
	Campo de visión	aprox. 23 °
	Tamaño de pantalla (distancia de proyección)	80 pulgadas en 5 m - 320 pulgadas en 20 m
	Reproducción del color	24 bits color (16,77 millones de colores)
	Frecuencia de actualización	30 Hz
Sistema Operativo		Moverio OS
Sensores		GPS, Brújula, Giroscopio, Acelerómetro, Sensor de iluminación
Conectividad	LAN inalámbrica	IEEE 802.11a/b/g/n/ac Wi-Fi Direct, Wi-Fi Miracast (Source/Sink)
	Bluetooth	Bluetooth Smart Ready
	Micro USB	USB 2.0
CPU y Memoria	CPU	Intel® Atom™ x5, 1.44GHz Quad Core
	RAM	2 GB
	Memoria interna	16 GB
	Memoria externa	microSD (máximo 2 GB), microSDHC (máximo 32 GB)
Formatos Compatibles	Vídeo	MP4 (MPEG4+AAC), MPEG2 (H.264+AAC)
	Audio	AAC, MP3, WAV
	Compatible con 3D	Side-by-side
General	Temperatura operativa	5 °C - 35 °C, 41 ° F - 95 ° F, 20 % - 80 % Humedad
	Duración de la batería	6 horas (Modo vídeo con Sistema operativo a 25°)

Fuente: Elaboración propia basado en Moverio BT-300 (Epson, 2020b)

2.2.5 DreamGlass

Son unas gafas para realidad aumentada orientadas más a desarrolladores de software que a su uso en el sector industrial, y son diferentes a las demás ya que son las únicas que basan su uso en un *smartphone* (DreamGlass, 2020a). Ofrecen un campo de visión de 90° de campo de visión en una resolución de 2.5k y un seguimiento de la cabeza con 3 grados de libertad, además de trabajar con celulares tiene una compatibilidad directa con PC. Ofrece para el desarrollo de aplicaciones en esta plataforma, un entorno de desarrollo llamado *DreamWorld SDK*, basado en *Unity*. Las especificaciones técnicas más actuales de estos dispositivos son expuestas en la Tabla 2.5.

Tabla 2.5 Especificaciones técnicas DreamGlass

HW/SW		Información
Display	Campo de visión	90° en RA, equivalente a 3 metros
	Resolución	4K (3840*1080)
	Píxeles por pulgada	868
	Soporte para 3D	Si
	Relación de aspecto	18:9
Audio	Salida	Sonido espacial integrado, entrada 3.5mm y Bluetooth
	Micrófono	No
Sensor	Rastreo	1000Hz para rastrear la cabeza
	Teclado	10 botones
Desarrollo	Puerto externo para sensor	USB 3.0 en el Headset
	SDK	Unity
CPU y Memoria	CPU	1.8GHz Quad Core AI Chip
	GPU	Mali-T864
	Almacenamiento	64GB
	Memoria	4GB
Sistema Operativo		Android
Batería		8000 mAh Batería de Litio
Tamaño		269*170*43mm, con una banda ajustable
Peso		185 sin cable

Fuente: Elaboración propia basado en Dreamglass 4K/4k Plus (DreamGlass, 2020b)

2.2.6 Magic Leap

El dispositivo que se ofrece para el desarrollo de realidad aumentada, más que eso, es una implementación tecnológica que se mezcla con otros dispositivos para ofrecer una experiencia inmersiva desde un enfoque de realidad mixta, es decir, además de incluir la pantalla de visualización frontal como los demás, también incluye otros dispositivos que se conectan de manera inalámbrica que permiten interactuar con toda la información virtual que se muestra en dicha pantalla, además, se pueden realizar conexiones en tiempo real con otros dispositivos *Magic Leap* para interactuar con otras personas y con objetos compartidos para todos los integrantes de la sesión (Magic Leap, 2020). Las especificaciones técnicas más actuales de estos dispositivos son expuestas en la Tabla 2.6.

Tabla 2.6 Especificaciones técnicas Magic Leap

HW/SW	Información
Frecuencia de actualización	120Hz
Sistema operativo	Lumin Os
HW/SW	Información
Memoria	8Gb
Wi-Fi	IEEE 802.11a/g/b/n/ac, dual-band
Bluetooth	Bluetooth 4.2
Cámara	2 Megapíxeles con sensor de color Campo de visión de 76.9° con foco fijo Video de 1080p30
Display	1280x960 pixeles RGB
Salida de audio	Built-in Stereo Speakers
Carga de datos	USB-C PD 2.0
Batería	3.5hrs
Sensores inerciales	Acelerómetro de 2x3-ejes Magnetómetro de 3-ejes
Peso	316g

Fuente: Elaboración propia basado en Magic Leap 2 Overview (Magic Leap, 2020)

2.3 Dispositivo utilizado

Aunque los dispositivos explicados anteriormente solo son algunos de los existentes en el mercado, el objetivo principal de esta investigación no es realizar la comparación entre estos. El principal objetivo es analizar de manera general cómo trabaja la realidad aumentada para visualizar información de otros sistemas informáticos, y para eso, cualquiera de los dispositivos antes mencionados puede cumplir esta funcionalidad.

El dispositivo utilizado en nuestra investigación fue el *Vuzix Blade*, el cual fue proporcionado por el IMT-Laboratorio Nacional Conacyt en Sistemas de Transporte y Logística para realizar pruebas de nuevas tecnologías en proyectos de logística.

Además de las especificaciones técnicas que ya se mencionaron (ver sección **¡Error! No se encuentra el origen de la referencia.**), cuenta con algunas otras ventajas, como lo son la gran variedad de aplicaciones que se han desarrollado para este dispositivo específicamente. Además cuenta con una aplicación *Blade Companion* (Vuzix, 2020b)(Vuzix, 2020c), la cual ofrece la capacidad de administrar y monitorear el dispositivo a través de la conectividad Bluetooth. También cuenta con una aplicación dedicada a la configuración de *Amazon Alexa*, el asistente personal basado en órdenes de voz, con el cual podemos administrar la mayoría de las funcionalidades del dispositivo. Finalmente, entre otra de las muchas peculiaridades del dispositivo, se pueden recibir notificaciones de mensajes y llamadas del celular pareado con el dispositivo, se pueden contestar los mensajes sin problemas, pero aún no tienen implementado el contestar las llamadas telefónicas.

3. Implementación de realidad aumentada

Para identificar y entender la creación de las aplicaciones que implementan realidad aumentada, es necesario conocer aquellos “*Frameworks*” existentes que facilitan o agilizan el desarrollo, ya que, al existir una gran variedad, es necesario seleccionar la mejor alternativa que sea compatible con el dispositivo utilizado para esta investigación: el *Vuzix Blade*.

Por otro lado, el sistema de realidad aumentada no es la única parte de esta investigación, también se incluye la explicación del sistema de comunicación que fue desarrollado para generar la información relevante a mostrar en el sistema. Igualmente, se explica de manera general los cambios que se realizaron en el sistema web del Mapa Digital.

3.1 Sistema de Comunicación

El Mapa Digital es un proyecto desarrollado en el IMT-Laboratorio Nacional Conacyt en Sistemas de Transporte y Logística que permite monitorear la fluidez corredores de transporte de carga por carretera, así como el valor logístico de la infraestructura, favoreciendo la mejor toma de decisiones en la planeación de la infraestructura carretera al igual que la mejora de la operación, productividad y eficiencia logística de empresas del autotransporte de carga. Este proyecto es un sistema web o aplicación web que ofrece a través de indicadores las capacidades de análisis y visualización de datos masivos del transporte y logística en México, los resultados se pueden observar a través de la interacción directa con un mapa del país, dividido en varias secciones los cuales nos ofrecen información en diferentes capas, y donde cada una de las capas nos ofrece información relevante por tramos carreteros identificados por la Red Nacional de Caminos³.

Como se mencionó anteriormente, el Mapa Digital es una aplicación web que necesita interacción humana para la visualización de los indicadores y la información que requiere el usuario. Por otra parte, es necesario adaptar este sistema para que pueda existir una comunicación con otros sistemas, es decir, una comunicación entre máquinas para la transferencia de información es por eso por lo que se recurrirá a la definición de los servicios web para esta implementación.

³ La Red Nacional de Caminos es la representación cartográfica digital y georreferenciada de la infraestructura vial del país con alta precisión y escala de gran detalle; modelada y estructurada con el fin de facilitar el cálculo de rutas está conformada bajo estándares internacionales y el riguroso marco normativo aplicable del Sistema Nacional de Información Estadística y Geográfica (SNIEG), la RNC fue desarrollado en la colaboración entre la SCT-IMT y el Inegi.

Para entender más acerca de esto es necesario explicar algunas definiciones de informática y desarrollo de servicios web.

3.1.1 Servicio Web

Para entender la diferencia entre una aplicación web y un servicio web es necesario entender de mejor manera cuales son las características y definiciones de cada término. Por definición una aplicación web es una herramienta la cual puede ser accedida desde un servidor web a través de internet o de una intranet mediante un navegador. En otras palabras, es un programa que se codifica en un lenguaje interpretable por los navegadores web, los cuales tienen la ventaja de ser independientes del sistema operativo del usuario, así como la facilidad para ser actualizados y mantenidos sin necesidades de instalar nuevo software con los usuarios finales. Estas aplicaciones web tienen elementos que permiten la comunicación activa entre el usuario y la información, esto permite que puedan acceder a los datos de manera interactiva, ya que el sistema responderá de acuerdo con las acciones que realice el usuario.

Por otro lado, los servicios web, son una tecnología que se utiliza para intercambiar datos entre distintas aplicaciones basándose en protocolos y estándares que definen la estructura de la comunicación. Estas aplicaciones pueden ser desarrolladas en lenguajes de programación diferentes, y ejecutadas en cualquier plataforma. A través de estos servicios, se puede tener una comunicación efectiva e intercambiar datos en internet o en cualquier red. Las organizaciones mundiales OASIS⁴ (OASIS, 2020a) y W3C⁵ (World Wide Web Consortium [W3C], 2020c) son aquellas responsables de la arquitectura y la reglamentación de los servicios web.

El W3C define un servicio web como: *“Un servicio web es un sistema de software diseñado para soportar la interacción máquina-a-máquina, a través de una red, de forma interoperable. Cuenta con una interfaz descrita en un formato WSDL procesable por un equipo informático, a través de la que es posible interactuar con el mismo mediante el intercambio de mensajes SOAP, típicamente transmitidos usando serialización XML sobre HTTP conjuntamente con otros estándares web”* (World Wide Web Consortium [W3C], 2020d).

3.1.1.1 Arquitectura

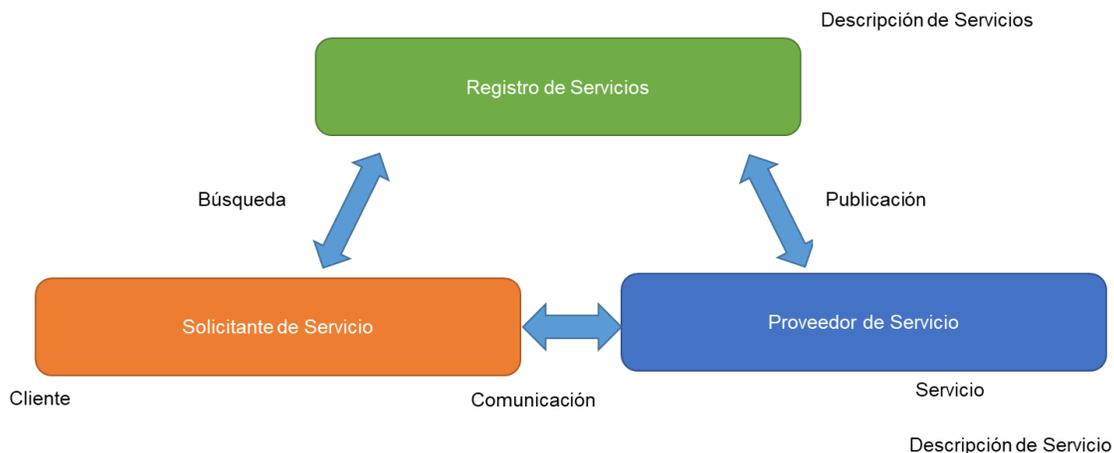
Un servicio web es una noción abstracta que debe ser implementada por un agente concreto, este agente representa la pieza concreta de software o hardware que envía y recibe mensajes, mientras que el servicio es el recurso caracterizado por el

⁴ OASIS (Organization for the Advancement of Structured Information Standards). - es un consorcio internacional sin fines de lucro que se orienta al desarrollo, la convergencia y la adopción de los estándares de comercio electrónico y servicios web.

⁵ W3C (World Wide Web Consortium).- es un consorcio internacional que genera recomendaciones y estándares que aseguran el crecimiento de la *World Wide Web*.

conjunto abstracto de funcionalidades que se proporcionan. Para entender mejor, se puede implementar un servicio *web* concreto utilizando un agente desarrollado en un lenguaje de programación en un momento dado, posteriormente, se puede desarrollar otro agente escrito en un lenguaje de programación diferente con la misma funcionalidad. Aunque el agente haya cambiado, el servicio *web* sigue siendo el mismo y teniendo la misma funcionalidad.

De manera general, en esta arquitectura se pueden identificar tres partes principales (World Wide Web Consortium [W3C], 2020e), tal como se muestra en la Figura 1.1, lo que es el proveedor de servicios *web*, el solicitante de los servicios, y un publicador que tiene el registro de los servicios existentes. El proveedor actualiza al publicador la definición de sus servicios *web* a través del envío de un documento *WSDL*, posteriormente el solicitador pide la información al publicador referencia a un servicio que desee realizar, y este último le proporciona la información necesaria para su comunicación. Posteriormente el agente solicitante contacta con el agente proveedor para realizar una petición a través de un protocolo definido, y finalmente el proveedor valida la petición y envía el dato de manera estructurada en un formato de *XML* utilizando el protocolo respectivo. Este fichero transferido en la petición se valida a través de *XSD*.



Fuente: Elaboración propia

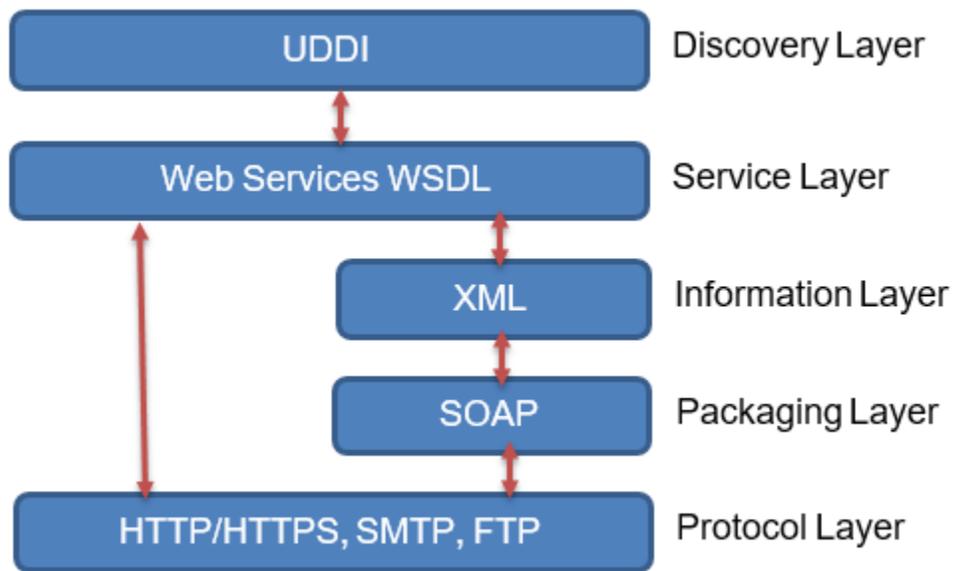
Figura 3.1 Arquitectura de Servicios Web

3.1.1.2 Protocolos

Algunos de los estándares principales definidos en la arquitectura de servicios *web* son explicados a continuación:

- **Web Services Protocol Stack.** Es una colección de protocolos y estándares para redes de computadoras que son utilizados para definir, localizar, implementar y hacer que un servicio *web* interactúe con otro (HezareMoghadam et al., 2008), tal como se muestra en Figura 3.2. Cuenta

con cuatro partes: a) Servicio de transporte- responsable del transporte de mensajes a través de protocolos como *HTTP*, *SMTP* y *FTP*; b) Mensajería *XML*- responsable por la codificación de mensajes en un formato *XML* para que puedan ser entendidos en cualquier extremo de la conexión, incluye protocolos *XML-RPC*, *SOAP*, *REST*; c) Descripción del servicio- se encarga de describir la interfaz pública de un servicio web a través de un formato de interfaz *WSDL*; d) Descubrimiento de servicios- centraliza los servicios en un registro común para que los servicios puedan publicar su ubicación y descripción y los haga disponibles en la red. La *API UDDI* se utiliza normalmente para esta tarea.



Fuente: Elaboración propia basado en (HezareMoghadam et al., 2008)

Figura 3.2 Pila de protocolos para servicios web

- **XML (eXtensible Markup Language).** Es un metalenguaje que permite definir lenguajes de marcas desarrollado por W3C utilizado para almacenar datos en forma legible(World Wide Web Consortium [W3C], 2020h).
- **SOAP (Simple Object Access Protocol).** es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML(World Wide Web Consortium [W3C], 2020b).
- **XML-RPC (Remote Procedure Call).** es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes(World Wide Web Consortium [W3C], 2020g). Es un protocolo muy simple ya que solo define unos cuantos tipos de datos y comandos útiles, además de una descripción completa de corta extensión.

- **WSDL (Web Services Description Language)**. es el lenguaje de la interfaz pública para los servicios web(World Wide Web Consortium [W3C], 2020f). Es una descripción basada en *XML* de los requisitos funcionales necesarios para establecer una comunicación con los servicios web.
- **UDDI (Universal Description, Discovery and Integration)**. protocolo para publicar la información de los servicios web(OASIS, 2020b). Permite comprobar qué servicios web están disponibles.
- **REST (Representational State Transfer)**. arquitectura que se basa en el uso del protocolo *HTTP* y proporciona una API que utiliza cada uno de los métodos (*GET, POST, PUT, DELETE*) para poder realizar diferentes operaciones entre la aplicación que ofrece el servicio web y el cliente(Fielding, 2000).

3.1.2 Arquitectura orientada a servicios

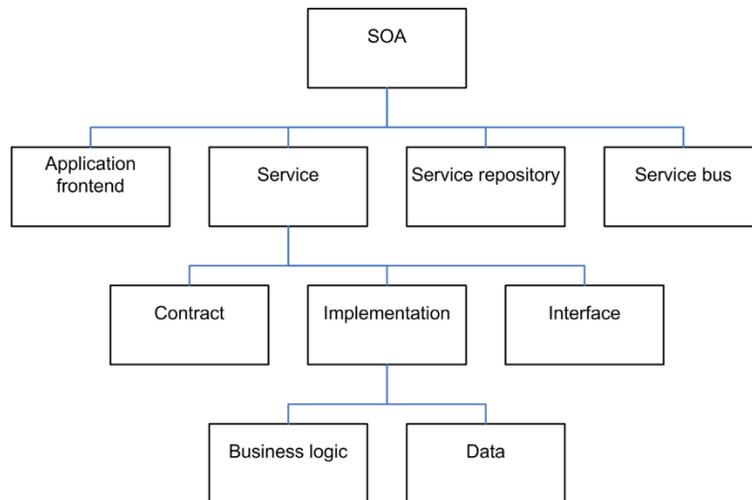
La arquitectura orientada a servicios (*SOA-Service Oriented Architecture*) es una arquitectura que se apoya en orientar las capacidades de un sistema a una idea basada de servicios, donde un servicio es la representación lógica de una actividad de negocio que tiene un resultado de negocio específico. Por ejemplo, considerando un sistema bancario basado en esta arquitectura, algunas actividades de negocio serían: a) la recuperación de los datos de un cliente, b) revisión de su crédito, c) reporte de ingresos y gastos en la cuenta, d) validación de la información del cliente, entre otras (González Quiroga, 2011). Se puede definir un servicio como un paquete funcional de software al cual se puede acceder a través de una infraestructura de red. Los servicios son autónomos, auto-contenidos y tienen la capacidad de controlarse y validarse automáticamente. Estos servicios tienen tres elementos fundamentales (Niknejad et al., 2020):

- **Contrato:** El uso de la funcionalidad que provee un servicio es gobernado por un contrato, el cual especifica el propósito, la funcionalidad, las restricciones y el modo de uso del servicio y es definido por el negocio.
- **Implementación:** La funcionalidad en sí misma que provee el servicio, puede ser realizada utilizando cualquier tecnología.
- **Interfaces:** Las interfaces proveen la forma de acceder a la funcionalidad de acuerdo con el contrato. Un servicio puede ofrecer múltiples interfaces para permitir su consumo de diferentes maneras.

Hay que tener en cuenta que la arquitectura de un Servicio Web no es lo mismo que la implementación de una arquitectura *SOA*. Un Servicio Web engloba varias tecnologías, incluyendo *XML, SOAP, WSDL, UDDI*, tal como se explicó en el capítulo 3.1.1.1, los cuales permiten construir soluciones de programación para mensajes específicos y para problemas de integración de aplicaciones. En cambio, *SOA* es una arquitectura de aplicación en la cual todas las funciones están definidas como servicios independientes con interfaces que pueden ser llamadas en secuencias bien definidas para formar los procesos de negocio. En *SOA* la clave

está en la interfaz ya que define los parámetros requeridos y la naturaleza del servicio y no la tecnología utilizada.

Los elementos principales(Krafzig et al., 2005) de esta arquitectura se pueden observar en Figura 3.3:



Fuente: Elaboración propia basado en (Krafzig et al., 2005)

Figura 3.3 Elementos de Arquitectura SOA

En una arquitectura *SOA*, los nodos de la red hacen disponibles sus recursos a otros participantes como servicios independientes a los que tienen acceso de un modo estandarizado. La mayoría de las definiciones de *SOA* identifican el uso de servicios web (empleando *SOAP* o *REST*) en su implementación, aunque no es necesario, ya que se pueden implementar utilizando cualquier tecnología basada en servicios.

3.1.2.1 SOAP vs REST

Los servicios *SOAP* se basan en el protocolo *SOAP*, esto quiere decir que su comunicación se realiza a través de *XML* y a través del protocolo *HTTP*, pero no se limita a este protocolo, también puede trabajar junto *FTP*, *POP3*, *TCP*, *JMS*, *MQ*, entre otros. Algunas características de *SOAP* son que permite un tipado mucho más fuerte, permite agregar metadatos mediante los atributos, permite definir espacios de nombre, evitando la ambigüedad. Pero esta también se puede ver como una desventaja, ya que *SOAP* es un formato más pesado, tanto en tamaño como en procesamiento, pues los *XML* tiene que ser parseados a un árbol *DOM*, y resolver los espacios de nombre antes de poder empezar a procesar el documento (World Wide Web Consortium [W3C], 2020b).

Por otro lado, los servicios *REST* son una tecnología mucho más flexible que transporta datos por medio del protocolo *HTTP*, pero éste permite utilizar los diversos métodos que proporciona para comunicarse, como lo son *GET*, *POST*, *PUT*, *DELETE*, *PATCH*, además se apoya en los códigos de respuesta nativos de *HTTP*(404,200,204,409)(World Wide Web Consortium [W3C], 2020a). *REST* es tan

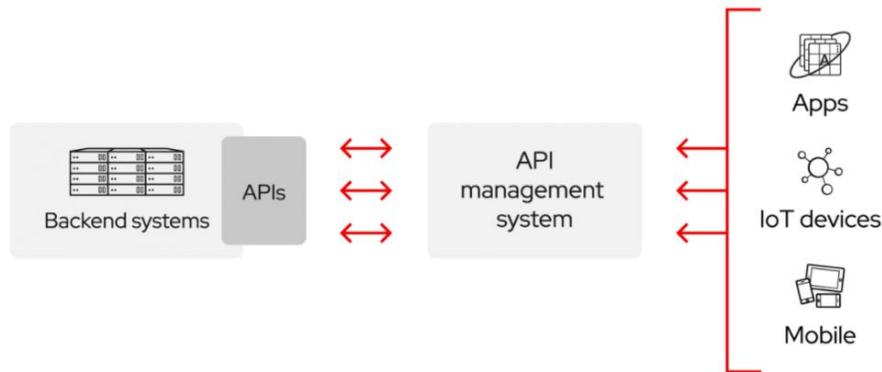
flexible que permite transmitir cualquier tipo de datos, ya que se definen por el *Header Content-Type*, lo que nos permite mandar documentos *XML*, *JSON*, Binarios como imágenes o documentos, texto, lo cual tiene como ventaja contra *SOAP* que solo permite la transmisión de datos en formato *XML*. A pesar de la gran variedad de tipos de datos que se pueden mandar con *REST*, se usa de manera muy particular el formato *JSON* por un motivo muy importante, *JSON* es interpretado de forma natural por *JavaScript*. Otra gran ventaja que presenta *JSON* es que son formatos mucho más livianos y mucho más rápidos de ser procesados que un formato *XML*(Fielding, 2000).

3.1.2.2 API REST

Tal y como se ha explicado a lo largo de este capítulo, una forma de implementar la arquitectura *SOA* es a través de tecnologías como *SOAP* y *REST*, y cada una de estas puede ser considerada tomando en cuenta las características del sistema donde se quiere desarrollar. Aunque, por otro lado, *REST* ha tomado bastante fuerza debido a su rendimiento comparado contra *SOAP*, y debido al incremento del uso tecnológico de Internet de las Cosas que, por la misma característica de conectar cada vez más dispositivos a Internet, *REST* ha tomado la delantera en cuanto a implementaciones de estos servicios. Es por que se ha seleccionado esta tecnología como el ecosistema de comunicación del Mapa Digital con el sistema de realidad aumentada y para poder lograr este objetivo, entra la metodología del desarrollo de *API*'s.

Una *API* (*Application Programming Interface*) es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones (Ofoeda et al., 2019)(Meng et al., 2018). Las *API* permiten que los servicios se comuniquen con otros sin necesidad de saber cómo se encuentran implementados. Esto simplifica el desarrollo de las aplicaciones y otorgan flexibilidad ya que simplifican el diseño, la administración y el uso de las aplicaciones(Woody et al., 2020). Las *API* también se pueden considerar como contratos, ya que a través de cierta documentación se representa un acuerdo entre dos partes, una parte que manda una solicitud remota con cierta estructura en particular, y la otra parte está encargada de responder siguiendo esa misma estructura(Sagdeo, 2019).

Las *API* son un medio simplificado para conectar infraestructura a través del desarrollo de aplicaciones nativas de la nube, pero también permiten compartir la misma información con clientes y otros usuarios externos(Red Hat, 2020), tal como se puede mostrar en Figura 3.4.



Fuente: Elaboración propia basado en (Red Hat, 2020)

Figura 3.4 Representación de comunicación de las API's con otros sistemas

3.2 Herramientas para Realidad Aumentada

Existen numerosos *frameworks* para el desarrollo de la tecnología de realidad aumentada (RA), cada uno de estos tiene sus particularidades y tienen algunas ventajas competitivas contra los demás como lo es el cliente final. Es decir, el tipo de plataforma que usará el dispositivo para la ejecución de las aplicaciones de RA. Debido a que cada uno se enfoca en diferentes metas, es necesario conocer las características de los cada uno de los *frameworks* y la infraestructura existente para el desarrollo de aplicaciones en esta tecnología. Algunos de los *frameworks* más relevantes para el desarrollo de aplicaciones AR se describen a continuación.

3.2.1 ARKit 4

El principal enfoque de *ARKit 4* (Apple, 2020a) es el desarrollo de aplicaciones para sistemas operativos *iOS* e *iPadOS*, de manera particular, para el desarrollo de aplicaciones trabaja en conjunto con otros *frameworks* de software como lo son *RealityKit*, *Reality Composer* y *Reality Converter*, lo cual lo hacen una opción muy fuerte para el desarrollo de RA. Pero limita a la infraestructura desarrollada por Apple, lo cual deja a un lado la compatibilidad con otros dispositivos que no tengan esta tecnología.



Fuente: Web page ARKit (Apple, 2020a)

Figura 3.5 Logo ARKit Apple

ARKit presenta una *Depth API* completamente nueva, que incorpora una nueva forma de acceder a la información profunda detalladamente a través de la recopilación de datos por el escáner LIDAR de estos dispositivos. Incorpora la comunicación con el sistema de mapas de Apple para aprovechar la información de alta resolución del módulo *Location Anchors* y mejorar la experiencia de RA en puntos específicos del planeta. Además, tiene compatibilidad para el sistema de *Face Tracking* a través del motor neuronal incorporado en algunos dispositivos de Apple. Para entender mejor la capacidad técnica de este *framework* es necesario revisar cada una de las características, las cuales son:

- **Depth API.** - Las capacidades avanzadas de comprensión de escenas integradas en el escáner LIDAR permiten que esta *API* utilice información de profundidad por píxel sobre el entorno circundante. Cuando se combina con los datos de malla 3D generados por *Scene Geometry*, esta información de profundidad hace que la oclusión de objetos virtuales sea aún más realista al permitir la ubicación instantánea de objetos virtuales y combinarlos a la perfección con su entorno físico.
- **Location Anchors.** - Permite anclar las creaciones AR en coordenadas específicas de latitud, longitud y altitud. Los usuarios pueden moverse alrededor de objetos virtuales y verlos desde diferentes perspectivas, exactamente como los objetos reales se ven a través de la lente de una cámara.
- **Face Tracking.** - La compatibilidad del seguimiento facial se extiende a la cámara frontal, y permite rastrear hasta tres rostros a la vez con la cámara la *TrueDepth*⁶ para potenciar experiencias de cámara frontal con otras aplicaciones.
- **Scene Geometry.** - Permite crear mapas topológicos del espacio con etiquetas que identifiquen diferentes objetos como pisos, paredes, techos, ventanas, puertas y asientos. Esta comprensión profunda del mundo real desbloquea la oclusión de objetos y la física del mundo real para objetos virtuales, y también brinda más información para potenciar flujos de trabajo de RA.
- **Oclusión.** - El contenido de RA pasa de manera realista detrás y delante de las personas en el mundo real, haciendo la RA más envolvente y, al mismo tiempo, permitir efectos de estilo de pantalla verde en casi cualquier entorno.
- **Cámaras simultáneas.** - Se puede hacer seguimiento facial y geoespacial en cámaras frontales y traseras.
- **Sesiones colaborativas.** - El sistema tiene la capacidad de poder realizar sesiones colaborativas entre diferentes usuarios, facilitando el desarrollo de las aplicaciones y dándoles la capacidad de experiencias compartidas entre usuarios finales.

⁶ Cámara desarrollada por Apple que permite capturar datos faciales precisos proyectando y analizando más de 30 000 puntos invisibles para crear un mapa de profundidad de la cara y permite capturar una imagen infrarroja de ésta (Apple, 2020b).

- **RA instantáneo.** - El escáner LIDAR con el que cuentan los dispositivos permite una lectura y detección ágil del ambiente, y con esto, se pueden realizar superposiciones de objetos de RA sin realizar un escaneo completo.
- **Captura de movimiento.** - A través de la tecnología con la que cuentan los dispositivos se puede realizar una captura en tiempo real de los movimientos de los usuarios con una simple cámara, esto permite generar un patrón del cuerpo de los usuarios e incorporar esta información como entrada a las aplicaciones de realidad aumentada.
- **Múltiple Face Tracking.** - Este *framework* permite dar seguimiento a varias caras a la vez.

3.2.1.1 Reality Kit

Reality Kit es un framework de la misma compañía que *ARKit*, pero se enfoca en otras metas, como lo son la renderizada foto realista, efectos de la cámara, animaciones y la física de los objetos en sí. Ambos frameworks trabajan como una integración para ofrecer mejores resultados al implementar la realidad aumentada. Al trabajar de manera nativa ambos frameworks, se pueden obtener renderizados realistas para identificar el esqueleto humano y hacer transformaciones y animaciones de éste, así como identificar información de un cuerpo físico, e incluir información de videos en las aplicaciones.



Fuente: Web page ARKit (Apple, 2020a)

Figura 3.6 Logo RealityKit Apple

- **Textura de video.** - Este *framework* añade la capacidad de incluir videos en escenas de realidad aumentada, añadiendo objetos, superficies e incluso personajes a la vida a través de la animación virtual de estos elementos.
- **Mejora de la oclusión de los objetos a través de renderizado.** - Combinando la información del LIDAR con la detección de bordes de RealityKit, se puede implementar una interacción entre los objetos reales con los objetos generados virtualmente, como lo es, poner objetos sobre mesas, o que estos no puedan traspasar paredes, tal y como si estos objetos virtuales interactuaran de manera física con el entorno real.
- **Swift API.** - El lenguaje de desarrollo de este framework es Swift, el cual nos ofrece una biblioteca con las características necesarias para la implementación adecuada y el desarrollo rápido de aplicaciones de RA.

- **Capacidad de escalamiento.** - Se pueden desarrollar sistemas de realidad aumentada utilizando la cache del CPU de manera adecuada y utilizando los múltiples núcleos para mejorar el rendimiento, y de manera general, tiene la capacidad de mejorar estas características de rendimiento de manera automática.

3.2.1.2 Reality Composer y Reality Converter

Estas dos herramientas en conjunto pertenecen a *AR Creation Tools*, las cuales ofrecen la capacidad de convertir modelos en 3D en objetos que posteriormente pueden ser utilizados en las aplicaciones de realidad aumentada.

Reality Composer facilita la construcción, pruebas, afinamiento y simulación de las aplicaciones que se desean desarrollar, ofrece una variedad de herramientas para hacer uso de los diseños y modelos virtuales existentes, o para la creación de éstos, los cuales posteriormente pueden ser exportados a las aplicaciones de realidad aumentada a través de la herramienta de Apple *Xcode*⁷.



Fuente: Web page ARKit (Apple, 2020a)

Figura 3.7 Logo Reality Composer

- **Biblioteca para compilación de RA.** - Se tiene la capacidad para importar y exportar los archivos USDZ⁸ o realizar búsquedas de objetos virtuales ya existentes en esta biblioteca, además permite la personalización de las características de estos objetos como lo son el tamaño o el estilo.
- **Animaciones y audio.** - Esta herramienta permite la capacidad de mover, escalar y agregar énfasis a los objetos virtuales, las cuales son acciones que se realizan cuando el usuario interactúa con la aplicación, además se puede incluir información auditiva como una capa extra de realidad aumentada.

⁷ Es un entorno de desarrollado creado por Apple para el desarrollo de sus aplicaciones(Apple, 2020c).

⁸ *Universal Scene Description* es un marco para el intercambio de datos de gráficos por computadora en 3D.

- **Record and Play.** - Esta librería tiene la capacidad de guardar la información de la cámara y los sensores en las ubicaciones donde se tienen las experiencias de realidad aumentada.

Por otro lado, *Reality Converter* es una aplicación que se enfoca particularmente en mejorar la implementación de los modelos 3D; nos permite convertir, visualizar y personalizar los objetos en un formato de USDZ, a pesar de que se encuentra en su fase beta, ya está disponible para los desarrolladores de esta plataforma.



Fuente: Web page ARKit (Apple, 2020a)

Figura 3.8 Logo *Reality Converter*

3.2.2 ARCore

ARCore (Google, 2020a) es la plataforma de desarrollo de aplicaciones de realidad aumentada desarrollada por Google, se basa en el uso de diferentes *API's* y permite a los teléfonos celulares detectar el ambiente, entendiendo cómo el mundo interactúa con la información. Las *API's* utilizadas permiten el desarrollo en plataformas como Android y iOS.



Fuente: Web page (Google, 2020a)

Figura 3.9 Logo *Reality ARCore*

Se basa en tres capacidades principales que permiten la integración de contenido virtual a través de las cámaras de los teléfonos, la primera es permitir al teléfono entender y rastrear su posición relativa en el mundo. Es decir, de todo a su alrededor, la segunda capacidad permite detectar la ubicación y el tamaño de todas las superficies, ya sea en horizontal, vertical o en diferentes ángulos, finalmente permite al teléfono estimar las condiciones de iluminación del ambiente.

De manera muy general, este *framework* realiza dos actividades, la primera es rastrear la posición del dispositivo mientras se mueve, y la segunda actividad es construir el mundo que lo rodea desde un punto de vista del dispositivo.

Para entender mejor este *framework* es necesario explicar más detalladamente cada uno de los módulos que lo componen (Google, 2020b), los cuales serán descritos a continuación.

3.2.2.1 Motion tracking

A medida que el teléfono se mueve, *ARCore* utiliza el algoritmo localización y mapeo simultáneo (SLAM) para comprender dónde se encuentra en relación con su entorno, este algoritmo se ha explicado en el capítulo anterior. *ARCore* detecta características visualmente distintas en la imagen capturada por la cámara, estas características son llamadas puntos de características, y utiliza estos puntos para calcular su cambio de ubicación. La información visual se combina con mediciones inerciales (*IMU*⁹) del dispositivo para calcular la posición y orientación de la cámara en relación con su entorno a través del tiempo.

Al alinear tanto la posición como la orientación de la cámara virtual que contiene la información 3D con la orientación y posición de la cámara del dispositivo, los desarrolladores pueden renderizar contenido virtual desde la perspectiva correcta. La imagen virtual renderizada se puede superponer sobre la imagen obtenida de la cámara del dispositivo, haciendo que parezca que el contenido virtual es parte del mundo real.

3.2.2.2 Entendimiento del ambiente

ARCore busca grupos de puntos de características que parecen estar en superficies horizontales o verticales comunes, como mesas o paredes, y pone estas superficies a disposición de su aplicación como planos. *ARCore* también puede determinar los límites de cada plano y poner esa información a disposición de la aplicación para que se puedan colocar objetos virtuales apoyados en superficies planas.

Debido a que *ARCore* utiliza puntos de características para detectar planos, es posible que las superficies planas sin textura, como una pared blanca, no se detecten correctamente.

3.2.2.3 Entendimiento de la profundidad

ARCore puede crear mapas de profundidad, imágenes que contienen datos sobre la distancia entre superficies desde un punto dado, utilizando la cámara RGB de un dispositivo. La información proporcionada por un mapa de profundidad permite al usuario generar experiencias inmersivas y realistas, como hacer que los objetos

⁹ *Inertial Measurement Unit* se obtiene a través de los acelerómetros y giroscopios del dispositivo.

virtuales choquen con precisión con las superficies observadas o hacer que aparezcan delante o detrás de objetos del mundo real.

3.2.2.4 Light estimation

El *framework* detecta información sobre la iluminación de su entorno y proporcionar la intensidad media y la corrección de color de una imagen, esta información permite iluminar objetos virtuales en las mismas condiciones que el entorno que los rodea, aumentando la sensación de realismo.

3.2.2.5 Interacción con el usuario

Para realizar una interacción con el usuario de manera eficaz, este *framework* utiliza la pantalla para identificar la coordenada (x, y) correspondiente y proyecta un rayo en el punto de vista de la cámara correspondiente a su entorno, devolviendo cualquier plano o puntos característicos que el rayo cruza, considerando la posición y orientación del teléfono como punto de referencia para la intersección en el espacio real. Esto permite a los usuarios seleccionar o interactuar con objetos en el entorno.

3.2.2.6 Puntos de orientación

Los puntos de orientación apoyan a los desarrolladores a colocar objetos virtuales en superficies tomando en cuenta características como su ángulo. Al realizar cualquier interacción con la aplicación, *ARCore* devuelve la información de los puntos de referencia, y posteriormente, a través de los puntos características cercanos intenta calcular el ángulo de la superficie.

3.2.2.7 Anclajes y rastreables

Las posiciones y orientaciones pueden cambiar a medida que el *framework* mejora su comprensión de la posición del dispositivo y del entorno. Para colocar un objeto virtual se necesita un ancla para asegurarse de que el *framework* pueda monitorearlo a través del tiempo, esto quiere decir que *ARCore* tiene la capacidad de actualizar la posición de los objetos virtuales dependiendo de los cambios de posición existentes con el tiempo en los planos físicos reales identificados, esto permite que al realizar un movimiento físico del dispositivo, se le pueda dar un seguimiento correcto de su posición y orientación a los objetos virtuales anclados en el entorno.

3.2.2.8 Imágenes de realidad aumentada

Esta función permite que las aplicaciones desarrolladas puedan responder a imágenes en 2D específicas, lo cual permite, que, al visualizar estas imágenes en la aplicación, se conviertan en desencadenadores de ciertas acciones, como podría ser el mostrar un objeto en 3D al visualizar dichas imágenes. Una ventaja de este *framework* es que puede rastrear estas imágenes a pesar del movimiento.

De manera particular, para agregar imágenes hay dos alternativas la primera es generar una base de datos de imágenes que puedan ser reconocidas por el sistema, o actualizar las imágenes individuales en tiempo real desde el dispositivo. Una vez registradas, *ARCore* detectará estas imágenes, los límites de éstas y devolverá la posición y orientación del objeto correspondiente.

3.2.2.9 Multiplataforma

Este framework tiene una *API* llamada *ARCore Cloud Anchor* la cual proporciona la capacidad de que los puntos de características generados y la base de datos de la información referente a los modelos 2D y 3D, puedan ser compartidos entre usuarios que utilicen diferentes dispositivos, esto es porque, esta *API* almacena toda esta información, y posteriormente puede ser accesada por cualquier dispositivo sin importar la plataforma que haga la petición.

3.2.3 ZapWorks

ZapWorks (Zappar, 2020) es una herramienta para realidad aumentada bastante robusta enfocada en la creación, publicación y análisis de experiencias RA pero desde la perspectiva de un cliente, es decir, facilita la integración e implementación de esta tecnología en campañas de marketing.



Fuente: Web page ZapWorks (Zappar, 2020)

Figura 3.10 Logo ZapWorks

Por otro lado, *ZapWorks* ofrece integración con otras tecnologías para agilizar la integración de su tecnología con aplicaciones para celulares o páginas web y en las cuales incluyen características como seguimiento de caras o imágenes, soporte para 3D, soporte para videos, desarrollo en 360°, identificación de iluminación dinámica, y publicación instantánea de los resultados. Las herramientas que nos ofrecen serán descritas a continuación.

3.2.3.1 WebAR

La primera herramienta es *WebAR*, la cual se enfoca completamente en los algoritmos utilizados para realidad aumentada y buscan que esta tecnología sea accesible a cualquiera que la necesite, lo cual logran a través de una aplicación web que incorpora todas las características fundamentales para ofrecer una experiencia de RA. Además, incluyen *triggers* que se ejecutan al detectar códigos de QR para visualizar de manera inmediata los enlaces asociados, también incorporan reconocimiento de imágenes a través de *URL's* personalizados.

3.2.3.2 Zapalytics

Esta herramienta se ofrece como un panel automático de control de datos donde se pueden generar reportes e informes del uso de los productos que utilicen cualquier otra herramienta de *ZapWorks*, principalmente se utilizan para generar y visualizar métricas de rendimiento. De manera autónoma identifica qué funciona bien y qué no al realizar llamadas a las funciones internas del sistema, identificando cada objeto a través de un etiquetado personalizado, y generando información en tiempo real de los horarios en los cuales se interactúa con dichos objetos o productos.

3.2.3.3 ZapWorks Studio

Esta herramienta funciona como plataforma de diseño y desarrollo de aplicaciones en realidad aumentada y realidad virtual, y de la cual se ofrecen tutoriales para identificar de manera ágil la forma de crear una aplicación desde cero, documentación suficiente para facilitar el manejo de sus librerías con ejemplos incluidos, y existe un foro donde se puede tener interacción con otros usuarios y expertos en el uso de toda la tecnología de *ZapWorks*. Esta herramienta puede considerarse un IDE completo el cual puede interactuar con otras librerías externas, incluyendo algoritmos de rastreo y seguimiento del entorno a través de las funciones incorporadas por *ARKit* y *ARCore*.

Por otro lado, también se ofrece un SDK para realidad aumentada a través de librerías que incluyen los algoritmos necesarios para la implementación de la tecnología, y que es compatible con diferentes lenguajes y plataforma de desarrollo, tales como *ThreeJS*¹⁰(Cabello, 2013), *A-Frame*¹¹(Supermedium et al., 2015), *JavaScript*, *Unity* y *React*¹²(Facebook, 2013).

3.2.4 DroidAR

DroidAR se enfoca completamente en el desarrollo de realidad aumentada para Android (Heinen, 2010), tiene la capacidad de incluir ubicaciones y marcas dentro de las aplicaciones. Puede utilizarse para diferentes escenarios y esto se puede observar en todas las aplicaciones disponibles en el repositorio de

¹⁰ *Three.js* es una biblioteca liviana escrita en *JavaScript* para crear y mostrar gráficos animados por computadora en 3D en un navegador Web y puede ser utilizada en conjunción con el elemento canvas de *HTML5*, *SVG* o *WebGL*.

¹¹ *A-Frame* es un framework web de código abierto para crear experiencias de realidad virtual. Es una estructura de sistema de componente de entidad para *Three.js* donde los desarrolladores pueden crear escenas 3D y *WebVR* usando *HTML*.

¹² *React* es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página.

*GitHub*¹³(Microsoft, 2008), ya que se encuentra bajo la licencia de GNU GPL v3¹⁴(Free Software Foundation Inc., 2007) bajo la licencia comercial.

Una característica de este *framework* es que incluye el reconocimiento de pisadas, pueden incorporarse características de maniobrabilidad con ubicaciones físicas y tiene compatibilidad con *frameworks* de *crowdsourcing*. Al ser uno de los primeros SDK's desarrollados para realidad aumentada, el prototipo inicial fue desarrollado en el año 2010, fue muy utilizado en su época, pero a lo largo de los años y con la incorporación del mercado de muchas empresas que ofrecían su propia tecnología de realidad aumentada, poco a poco se fue olvidando este *framework*, y en la actualidad puede ser utilizado sin problema pero ya no tiene el apoyo de la comunidad como antes, y por lo tanto no se han desarrollado nuevas funcionalidades.

3.2.5 D'Fusion Mobile

La solución *D'Fusion Mobile* es una herramienta multiplataforma que soporta principalmente plataformas Web, iPhone y Android. Sus características principales son: permite construir todo el escenario a presentar en realidad aumentada a través de la GUI; la inteligencia de escenarios se hace con LUA; tiene su propio IDE para desarrollo; la codificación es independiente del sistema operativo, por lo tanto se puede desarrollar para Android y iOS a la vez; tiene compatibilidad con exportaciones de *Blender*¹⁵ y *Maya*¹⁶; soporta multi-tag y *Face-Tracking*; soporta Unity 3D, para Android JNI¹⁷ y iPhone Lib ya están pre-compilados los sistemas operativos.

A pesar de que existen muchos sistemas desarrollados con este framework, la empresa *Total Immersion* ya no se encuentra trabajando en este proyecto, y no existe ninguna comunidad para darle soporte a estos sistemas, la documentación

¹³ Es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Git es un software de control de versiones pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

¹⁴ La Licencia Pública General de GNU es una licencia libre y con derecho a copia para el software y otros tipos de obras. Está pensada para garantizar su libertad de compartir y cambiar todas las versiones de un programa, para asegurar que siga siendo software libre para todos sus usuarios.

¹⁵ *Blender* - es un programa informático multiplataforma, dedicado especialmente al modelado, iluminación, renderizado, la animación y creación de gráficos tridimensionales.

¹⁶ *Autodesk Maya* - es un programa informático dedicado al desarrollo de gráficos 3D por ordenador, efectos especiales, animación y de dibujo.

¹⁷ JNI – es la interfaz nativa de Java para desarrollo de aplicaciones Android.

ya es muy escasa, y no existen tutoriales como tal para el desarrollo de las App's, solo algunos ejemplos muy sencillos.

3.2.6 Vuforia

Vuforia es un kit de desarrollo de software (SDK) (PTC Inc., 2020) para realidad aumentada específicamente para la creación de aplicaciones en dispositivos móviles, se basa en el uso de la tecnología de visión computacional para reconocer y rastrear imágenes y objetos en 3D dentro del mundo real, lo que permite que los desarrolladores puedan orientar y posicionar cualquier tipo de objetos virtuales como lo son los modelos en 3D.



Fuente: Web page Vuforia (PTC Inc., 2020)

Figura 3.11 Logo Vuforia

Soporta una gran variedad de formatos tanto en 2D como en 3D para mejorar la experiencia en realidad aumentada, además puede incluir objetivos de imagen sin marcadores, objetivos de modelos 3D y marcadores de referencia para identificar la posición y orientación desde la perspectiva del usuario. Otras características incluidas en la SDK son la localización de dispositivos con seis grados de libertad en el espacio, la detección de oclusión localizada mediante botones virtuales, la selección de imágenes objetivo en tiempo de ejecución y la capacidad de crear y reconfigurar conjuntos de objetivos mediante programación en tiempo de ejecución.

Por otro lado, *Vuforia* también ofrece una API para el desarrollo de aplicaciones en lenguajes como C++, Java, Objective-C++ y .Net a través de un motor de desarrollo de videojuegos: Unity. Por esta razón, la compatibilidad de las aplicaciones desarrolladas nativamente se incrementa y pueden ser ejecutadas en diferentes sistemas operativos como iOS y Android sin mucho problema, algunas de las características implementadas en *Vuforia* en los últimos años son explicadas a continuación:

- Simplificación de modelos: Incluyen un nuevo proceso de simplificación dentro del generador de modelos objetivo, el cual reduce el recuento de polígonos y el número de partes para optimizar el tamaño de la operación de rastreo de modelos en los dispositivos, es decir, los modelos 3D son cargados en una nube de Vuforia, y el proceso de identificación de este modelo se realiza en ese servicio y no en el dispositivo que corre la aplicación.

- Generador de áreas objetivo: Su generador soporta escaneos de sistemas de cartografía móviles tales como *NavVis M6* y *VLX*¹⁸.
- Aplicación de creación de áreas objetivo: Tienen actualizada la funcionalidad para que dentro de las aplicaciones desarrolladas se tenga una característica que permita a los usuarios borrar conjuntos de áreas objetivo o compartirlos con otros usuarios.
- Actualización de documentación: Cada cierto tiempo recopilan datos de los usuarios que utilizan estos dispositivos para definir nuevas funcionalidades tales como la mejora que permite combinar múltiples áreas objetivo o para crear nuevas imágenes a partir de un conjunto de áreas. El soporte que se le sigue dando al sistema sigue vigente hasta la fecha.

¹⁸ *NavVis M6* y *VLX*- son sistema de mapeo móvil portátiles que permite un escaneo rápido con una interrupción mínima en entornos comerciales e industriales en tiempo real y capturar información del ambiente en alta calidad.

4. Desarrollo y resultados

Para comprobar la funcionalidad y la importancia de la tecnología de realidad aumentada para ofrecer soluciones innovadoras a los sistemas de transporte, se realizó una implementación piloto de dicha tecnología. La aplicación sirvió para experimentar con la tecnología e incluir información relevante obtenida de sistemas informáticos desarrollados en otros proyectos. También se desarrolló un servicio web que obtenía información del sistema de Mapa Digital y también se incorporó la petición a otros servicios web existentes como *Google Maps API* y *OpenWeatherAPI*.

4.1 Desarrollo de Aplicaciones en Realidad Aumentada

Para desarrollar aplicaciones con esta tecnología, primero se necesita definir cuáles serán las herramientas para utilizar, ya que de esto dependen cuáles serán las estrategias y metodologías para la implementación en diferentes dispositivos existentes. Para empezar, hay que considerar que para el desarrollo de cualquier aplicación para el dispositivo seleccionado (*Vuzix Blade*) se tienen dos opciones, el desarrollo nativo para aplicaciones Android (debido al sistema operativo del dispositivo) o el desarrollo híbrido que puede basarse en *frameworks* que faciliten el desarrollo y puedan ejecutar tales como los que se presentaron en el capítulo anterior (ver sección 3.2).

4.1.1 Desarrollo Nativo en Vuzix

El desarrollo móvil nativo consiste en implementar en lenguajes de programación específicos a las plataformas donde se desea desarrollar por ejemplo plataformas de dispositivos móviles como Android o iOS, este desarrollo se fundamenta en el lenguaje base de dichos sistemas operativos como lo son Java y Kotlin para Android, y lenguajes como Objective-C y Swift para iOS.

El detalle de las aplicaciones nativas es que el proceso de desarrollo debe dividirse dependiendo a la cantidad de plataformas donde se desea implementar, en otras palabras, si se desea realizar una App que pueda ejecutarse tanto para iOS como para Android, serán dos proyectos diferentes debido a que se desarrollarán en lenguajes diferentes, aunque la finalidad de la aplicación sea la misma. Una ventaja dentro del desarrollo nativo es que se tiene completo acceso a los recursos del sistema y a las prestaciones del sistema operativo.

Para este caso particular de realidad aumentada para Vuzix, el desarrollo es tan sencillo como cualquier otro software para dispositivos celulares, ya que se pueden

utilizar las *API's* de *Android* y la plataforma de *Android Studio* para su fácil implementación, aunque al final, existen algunas diferencias inherentes, que se deben tomar en cuenta ya que las interacciones que se realizan con unos lentes de realidad aumentada y con un Smartphone son completamente diferentes.

El dispositivo incluye una pequeña pantalla del lado derecho de los lentes, donde se visualiza el sistema operativo y las aplicaciones que se instalen, lo cual refleja el primer paso a la tecnología que se desea implementar, además cuenta con otras particularidades de interacción con el usuario que mejoran esta experiencia.

Dentro de la plataforma web nos ofrece Vuzix se pueden encontrar varios tutoriales (Vuzix, 2018), los cuales brindan información necesaria para el desarrollo de una primera aplicación de realidad aumentada, con ejemplos claros que nos muestran los primeros pasos en la instalación de las herramientas necesarias, hasta proyectos completos que ya se hayan desarrollado en estos dispositivos. De manera general los tutoriales se basan en los siguientes temas de desarrollo:

- Uso de los paquetes y recursos del HUD
- Revisión de recursos y clases incorporados
- Estilos de diseño a seguir para la creación de la interfaz de usuario
- Configuración y control de la conectividad con la aplicación *Blade Companion*
- Control de temas y estilos para el HUD
- Menú de acción
- Configuración de la pantalla de inicio y sus respectivos *Widgets*
- Aplicación del sistema de reconocimiento de frases para interpretar y responder a comandos de voz.
- Uso y configuración de la cámara integrada

Para nuestro proyecto, cabe resaltar que cada uno de los tutoriales fueron desarrollados sin ningún problema y los resultados que se muestran en el sitio fueron replicados con éxito. Posteriormente a la realización de estos tutoriales, se debió cubrir dos funcionalidades extras dentro del desarrollo, la primera fue la geolocalización del dispositivo, para identificar la posición y la posible infraestructura de transporte que se encontrara cercana a dicha ubicación, y la segunda funcionalidad fue la implementación del código que generará peticiones a los servicios Web y analizará los datos de la respuesta.

4.1.2 Desarrollo Híbrido en Vuzix

Por otro lado, el desarrollo de aplicaciones para diferentes dispositivos con diferentes sistemas operativos se puede realizar a través del desarrollo híbrido, donde podemos encontrar diferentes tipos de plataformas que nos apoyan en la reducción de tiempo de desarrollo, debido a que ya existen herramientas que disminuyen el tiempo de codificación. Además, algunas aplicaciones pueden ser desarrolladas a través de los motores de videojuegos, ya que la mayoría de éstos se enfoca en desarrollo multiplataforma.

El desarrollo de aplicaciones híbridas es un enfoque de programación para dispositivos móviles. Ellas combinan las fortalezas de la programación nativa con otras tecnologías para desarrollar aplicaciones multiplataforma, las cuales, al final serán compiladas y ejecutadas de forma nativa sin importar el sistema operativo del dispositivo (Malavolta et al., 2015).

De manera general, existen muchos *frameworks* que nos proporcionan las herramientas necesarias para esta tarea de desarrollo de aplicaciones multiplataforma y son bastante versátiles para que se adapten a las necesidades de cada proyecto. Sin embargo, la desventaja es que en la mayoría de las aplicaciones desarrolladas el rendimiento puede variar de acuerdo con el dispositivo en donde se instalan. Existen muchos *frameworks* diferentes donde se pueden desarrollar dichas aplicaciones, y cada uno de estos frameworks tiene sus propias capacidades de desarrollo, ya que al final se hace una traducción de cualquiera de estos códigos a código nativo para que pueda ser ejecutado en cualquier dispositivo, algunos de los *frameworks* más usuales para el desarrollo híbrido son *Xamarin* (Microsoft, 2020c), *Titanium* (Axway, 2020) y *PhoneGap* (Adobe Systems Incorporated, 2020), aunque no son las únicas plataformas para cumplir este cometido.

Para el desarrollo de la aplicación de ejemplo con esta tecnología la cual implementa realidad aumentada, se utilizó la herramienta Unity, el cual es un motor de videojuegos que tiene la capacidad de implementar librerías que nos apoyan a esta tarea del desarrollo de realidad aumentada, e incorpora funcionalidades que compilan las aplicaciones finales en diferentes plataformas, incluyendo dispositivos móviles. Unity utiliza *OpenGL* (Windows, Mac, Linux), *Direct3D*(Windows), *OpenGL ES* (Android, IOS) e interfaces propietarias(Wii). Tiene soporte para el mapeado de relieves, de reflejos, oclusión ambiental, sombras dinámicas, render a texturas y efectos de pos-procesamiento de pantalla. Por otro lado, incluye un mecanismo para el desarrollo de scripting que se basa en la implementación de código abierto de .NET. Debido a todas estas características, y muchas otras más, el desarrollo de cualquier aplicación en Unity puede ser enfocado a un gran número de plataformas, tanto para plataformas PC y dispositivos móviles, así como plataformas web, Smart TV, consolas, y dispositivos de realidad extendida.

Para el desarrollo de la aplicación para este proyecto se utilizó Unity con Vuforia, además se siguió la misma base de la aplicación desarrollada nativamente, es decir, las secciones que muestran los resultados de los tres servicios web, *GoogleMaps*, *OpenWeather* y el Mapa Digital. Además, el desarrollo de la aplicación de ejemplo se basó en los tutoriales (Vuforia, 2020)(Denys Gamers Tutorial, 2017) ofrecidos para Vuforia, para mejorar el entendimiento de la tecnología y la implementación de mejores resultados para realidad aumentada. Para desarrollar cualquier aplicación con Vuforia es necesario tener en cuenta 3 pasos fundamentales:

- Ajuste del Motor de Vuforia en Unity

- Inserción de *assets*¹⁹
- Configuración de elementos de Vuforia

Para el primer paso, es necesario descargar el motor de videojuegos de Vuforia y cargarlo en la plataforma de Unity para poder utilizarlo posteriormente en el desarrollo, además se necesita configurar una cuenta en la página de Vuforia para generar una clave de licencia que posteriormente será agregada en nuestra aplicación para poder utilizar todas las funcionalidades de Vuforia, existen varias licencias con diferentes características, para este proyecto se utilizó la licencia de desarrollador, que a pesar de la limitación de algunas características, para los resultados de este proyecto, no afecta dicha limitación, además, no es necesario realizar ningún pago siempre y cuando la aplicación desarrollada no genere beneficios económicos.

Para el segundo paso, se puede realizar de dos maneras, se pueden incluir los elementos independientes dentro de la aplicación de Unity, o se puede generar una base de datos de imágenes en la página de Vuforia, para este último caso, los formatos de los elementos objetivo pueden ser de tipos diferentes: objetivos de imagen, objetivos cilíndricos, objetivos de objeto 2D, objetivos de objeto 3D, objetivos de *VuMarks* y multi-objetivos; cada uno de estos objetivos dependerá de cuáles son los algoritmos que utilizará el framework para rastrearlos mientras la aplicación se encuentre activa. Vuforia nos ofrece 3 maneras de utilizar las bases de datos de los elementos que queremos incluir en nuestras aplicaciones: base de datos en la nube, base de datos para dispositivos y base de datos *VuMark*; el primer caso es utilizado para hacer peticiones a su servidor y recuperar esta lista imágenes almacenadas para dicho proyecto, los otros dos casos son para bases de datos que se guardarán de manera local en el dispositivo donde se instalará la aplicación, la diferencia entre estos dos, es que el primero se enfoca en la necesidad de hacer un reconocimiento lo más rápido posible y no es necesario el cambio del conjunto de imágenes frecuentemente, y se recomienda el uso de los tipos de objetivos más sencillos (imagen, objetivos cilíndricos y objetivos 2D) para que los algoritmos decodifiquen de manera rápida y efectiva dichos elementos.

Finalmente, para el último caso, la configuración de los elementos de Vuforia, es necesario hacer un cambio de los elementos predeterminados que se ajustan en Unity para un nuevo proyecto, y ajustar los nuevos elementos que nos ofrece el motor de Vuforia, tales como la cámara para realidad aumentada. Dentro de esta actividad se ajusta la base de datos elegida y se importa para que pueda ser accesada dentro de nuestra aplicación.

Para los 3 pasos descritos anteriormente, se puede visualizar la configuración de cada uno de estos en el portal de desarrollador de Vuforia (Vuforia, 2020), donde

¹⁹ *Asset*. - Es una representación de cualquier ítem que puede utilizado en el proyecto, son archivos que representan modelos en 3D, audio, imágenes, o cualquier tipo de archivo que soporte Unity para incluir en las aplicaciones

se incluyen imágenes de la configuración de Unity, así como todas las opciones que se pueden configurar en el administrador de objetivos de la página oficial, donde se crean y gestionan las bases de datos para las aplicaciones.

4.2 Servicios Web

Para realizar las pruebas correspondientes se utilizaron tres servicios web, dos ya existentes en internet y muy utilizados de manera global, y otro desarrollado en el instituto que hace referencia a un poco de la información que se encuentra disponible en el Mapa Digital.

4.2.1 Google Maps Platform

En la actualidad, la empresa *Google* cuenta con muchas *API's* enfocadas en diferentes áreas informáticas, las cuales permiten la comunicación e integración de los servicios que ofrece con otros servicios web (Google, 2020c). De estas *API's* se pueden obtener funcionalidades de análisis. *Machine Learning*, acceso a datos de usuarios (cuentas *Google*), información geoespacial, entre otros. En este caso se hará referencia a las *API's* enfocadas a mapas, ya que serán utilizadas para implementarlas en la aplicación de realidad aumentada.

Dentro de los servicios web que nos ofrece *Google* para el uso y presentación de mapas (Google, 2020g), se encuentra una clasificación de tres áreas:

- **Google Maps API:** Visualización de mapas estáticos y dinámicos, descarga de imágenes de *Street View* y vistas en 360°.
 - **Funciones:** *Maps* y *Street Views*
- **Google Routes API:** Obtención de indicaciones de calidad y actualizaciones de tráfico en tiempo real para determinar las mejores rutas a recorrer con diferentes medios de transporte.
 - **Funciones:** *Directions*, *Distance Matrix* y *Roads*
- **Google Places API:** Obtención de información de más de 100 millones de sitios ubicados alrededor del mundo, se pueden encontrar las ubicaciones exactas a través de datos como números de teléfono, direcciones y señales en tiempo real.
 - **Funciones:** *Details*, *Current Place*, *Find Place*, *Autocomplete*, *Geocoding*, *Geolocation* y *Timezone*.

Cada una de las *API's* que se ofrecen tiene su propia documentación para el uso de estas. Para el caso particular de este proyecto se utilizará *Google Places*, para obtener información de la ubicación en la que se encuentre el dispositivo de realidad aumentada a través de su posición latitud/longitud. Para hacer las peticiones en esta *API* (Google, 2020d), el formato es el siguiente:

```
https://maps.googleapis.com/maps/api/geocode/Formato?Parámetros
```

Fuente: Elaboración propia basado en la API de Google (Google, 2020d)

Figura 4.1 Formato de peticiones a Google Places API

donde **Formato** indica el tipo de archivo que nos regresará la petición a este servicio web, la cual nosotros podemos configurar con cualquiera de dos opciones: *JSON* o *XML*. Los **Parámetros** son aquellos que describen de una mejor específica cómo se hará la petición y cuáles son los resultados que queremos obtener a través de esta, en este caso hay parámetros obligatorios y hay parámetros opcionales. Para el caso específico de este proyecto, tal como se explicaba anteriormente, se realizan peticiones con los valores de latitud y longitud obtenidas del dispositivo, y se obtiene la dirección que se encuentre en dicha ubicación.

Dentro de los parámetros obligatorios se encuentran **latlng** y **key** en los cuales tenemos que ingresar la información de la ubicación y una cadena de texto generada por *Google Platform* la cual es una llave que valida al usuario que está realizando dichas peticiones. En el caso de los parámetros opcionales podemos encontrar **language** que define el lenguaje en el que queremos los resultados, **result_type** que se utiliza como un filtro más específico del lugar, y podemos incluir búsquedas cercana de esa geoposición con información de calles, rutas, intersecciones, países, localidades, vecindarios, construcciones, código postal, puntos de interés, entre otras opciones; y finalmente **location_type** que en caso de que el parámetro anterior mande varias ubicaciones y se generen varias direcciones, se puede filtrar aún más a través de los valores dados en este parámetro.

Por otro lado, un ejemplo de los resultados obtenidos de esta API se puede observar en la Figura 4.2, donde podemos observar los resultados obtenidos de una petición sin ajustar los parámetros opcionales, solo los parámetros obligatorios, esta cadena de texto representa la información en formato JSON, pero para este proyecto solo se tomará la clave **formatted_address**, la cual representa los datos más relevantes de la petición.

```

{
  "results" : [
    {
      "address_components" : [
        {
          "long_name" : "277",
          "short_name" : "277",
          "types" : [ "street_number" ]
        },
        {
          "long_name" : "Bedford Avenue",
          "short_name" : "Bedford Ave",
          "types" : [ "route" ]
        },
        {
          "long_name" : "Williamsburg",
          "short_name" : "Williamsburg",
          "types" : [ "neighborhood", "political" ]
        },
        ... ]
      "formatted_address" : "277 Bedford Avenue, Brooklyn, NY 11211, USA",
      "geometry" : {
        "location" : {
          "lat" : 40.714232,
          "lng" : -73.9612889
        },
        "location_type" : "ROOFTOP",
        "viewport" : {
          "northeast" : {
            "lat" : 40.7155809802915,
            "lng" : -73.9599399197085
          },
          "southwest" : {
            "lat" : 40.7128830197085,
            "lng" : -73.96263788029151
          }
        }
      },
      "place_id" : "ChIJd8B1Q2BZwokRAFUEcm_grcA",
      "types" : [ "street_address" ]
    },
    ... Additional results[] ...
  ]
}

```

Fuente: Elaboración propia basado en la API de Google (Google, 2020d)

Figura 4.2 Resultado obtenido de una petición al servicio web de Google *Geocoding* API

4.2.2 OpenWeather API

Esta API está desarrollada por un grupo de científicos de datos ubicados en Londres, Reino Unido (OpenWeather, 2020) la cual nos ofrece información histórica, información en tiempo real e información pronosticada del clima para cualquier lugar del mundo. Los productos que nos ofrecen tienen una gran cobertura con información y análisis de los parámetros climatológicos esenciales tales como la temperatura, precipitación, probabilidad de precipitación, humedad, presión, abundancia de nubes, viento, entre otros aspectos, pero además ofrece información como riesgo en los caminos o alertas climatológicas, toda esta información que ofrecen proviene de un gran conjunto de sistemas nacionales climatológicos alrededor del mundo.

Algo que incluyen en los análisis que realizan para ofrecer la información en casi cualquier ubicación del mundo son redes neuronales convolucionales que recolectan la información de diferentes fuentes de datos y la procesan para cubrir cualquier ubicación y considerar matices locales del clima. Entre las fuentes de datos que alimentan sus sistemas se encuentran 82 mil estaciones meteorológicas alrededor del mundo en las cuales se incluyen agencias meteorológicas nacionales como NOAA, *Environment Canada*, *Met Office*, etc., radares, y satélites meteorológicos.

De manera general *OpenWeather* ha clasificado en cuatro partes los servicios que ofrece, cada una de estas clasificaciones se enfoca en diferentes análisis que realizan, y aunque existe una versión gratuita para utilizar todas las *API's* que ofrecen, es necesario pagar una suscripción para quitar la limitante de la cantidad de peticiones a realizar por día. Dentro de las cuatro clasificaciones y sus respectivas *API's* podemos encontrar:

- Recopilación de datos meteorológicos actuales y previos
 - Datos meteorológicos actuales
 - Pronóstico por horas-4 días
 - *API* de una llamada
 - Pronóstico diario-16 días
 - Previsión climática 30 días
 - Descarga masiva
 - *API* de radiación solar
 - Alertas meteorológicas globales-Notificaciones *push*
 - Pronóstico de 5 días / 3 horas
 - *API* Riesgo en el camino
- Recopilación de datos meteorológicos históricos
 - *API* de tiempo histórico
 - *API* de datos meteorológicos estadísticos
 - Parámetros acumulados
 - Descarga Masiva
 - Datos meteorológicos históricos por estado para todos los códigos postales (solo algunos países)
- Colección de mapas
 - Mapas del tiempo 2.0
 - Mapa global de precipitaciones
 - Mapas en relieve
 - Mapas meteorológicos 1.0
- Colección de otras *API* meteorológicas
 - *API* de contaminación del aire
 - *API* de codificación geográfica
 - Estaciones meteorológicas
 - Índice UV
 - Disparadores meteorológicos

En el caso de este proyecto, la API utilizada dentro del sistema de realidad aumentada fue la que proporciona el clima dependiendo de la ubicación geoespacial del dispositivo, la cual puede ser llamada con el nombre de la ubicación, o la latitud y longitud.

Existen cuatro opciones para la petición del servicio web, la primera opción es una llamada por el nombre de la ciudad a buscar, para este caso los parámetros obligatorios solicitados en la API son **q**-nombre de la ciudad, **appid**-la clave única de la API, y los parámetros opcionales son **mode**- el formato de respuesta el cual puede ser *XML* o *JSON*, el valor predeterminado es *JSON*, **units**-unidades de medida con opciones de *standard*, *metric* e *imperial*, el valor predeterminado es *standard*, finalmente **lang**- el lenguaje en el cual va a venir representada la respuesta. En la segunda opción solo cambia un parámetro, el **id**-que representa un identificador de ubicación en el sistema. La tercera opción es la que se identifica por las coordenadas geográficas y cambia por dos parámetros, **lat**-latitud del lugar, y **lon**-longitud. Finalmente, la última opción es una búsqueda a través del código postal pero solo puede ser utilizado para algunos países, y el parámetro es **zip- {zip code}, {country}** el cual tiene que incluir el código del país en donde se desea buscar. El formato para dichas solicitudes es el siguiente:

```
https://api.openweathermap.org/data/2.5/weather?Parámetros
```

Fuente: Elaboración propia basado en la API de OpenWeather (OpenWeather, 2020)

Figura 4.3 Formato de peticiones a OpenWeather API

Finalmente, el resultado que se obtendrá de cualquier de las peticiones explicadas anteriormente,

```
{
  "coord": {
    "lon": -122.08,
    "lat": 37.39
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "clear sky",
      "icon": "01d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 282.55,
    "feels_like": 281.86,
    "temp_min": 280.37,
    "temp_max": 284.26,
    "pressure": 1023,
    "humidity": 100
  },
  "visibility": 16093,
  "wind": {
    "speed": 1.5,
    "deg": 350
  },
  "clouds": {
    "all": 1
  },
  "dt": 1560350645,
  "sys": {
    "type": 1,
    "id": 5122,
    "message": 0.0139,
    "country": "US",
    "sunrise": 1560343627,
    "sunset": 1560396563
  },
  "timezone": -25200,
  "id": 420006353,
  "name": "Mountain View",
  "cod": 200
}
```

Fuente: Elaboración propia basado en la API de OpenWeather (OpenWeather, 2020)

Figura 4.4 Resultado obtenido de una petición al servicio web de *OpenWeather* API

4.2.3 Mapa Digital

Es un proyecto para el desarrollo de capacidades para el análisis y visualización de datos masivos del transporte y la logística desarrollado en el Instituto Mexicano del Transporte, permite monitorear la fluidez y desempeño de las cadenas de suministro, así como el valor logístico de la infraestructura, favoreciendo la mejor toma de decisiones tanto para la planeación e inversión en infraestructura, como para mejorar la operación, productividad y eficiencia logística de México. Este proyecto surgió de la oportunidad de mejora de la condición de la congestión de tráfico a nivel nacional; la Subsecretaría de Transporte de la Secretaría de Comunicaciones y Transportes, a través del análisis anual de Estadística Básica del Autotransporte Federal 2020 (Secretaría de Comunicaciones y Transportes [SCT], 2020), ha identificado que el 56.9 % del total del transporte de carga a nivel nacional es representado por el autotransporte de carga, y hace falta un análisis específico del flujo de carga en todo el país. Además, en México se encuentra la ciudad más afectada por congestiones a nivel mundial. Ciudad de México presenta un 66% de congestión con respecto al tiempo de viaje con flujo libre, esto representa hasta 59 minutos de tiempo extra pro congestiones, de acuerdo al índice de tráfico mostrado por Tom-Tom en su página web (Tom, 2020). México no cuenta con una entidad que se encargue de realizar análisis de grandes bases de datos enfocados en el transporte y la logística para monitorear permanentemente la fluidez y desempeño de las cadenas de suministro en México, o que proporcione a empresas y gobierno información de inteligencia oportuna y estratégica para la toma de decisiones tanto del futuro inmediato, como de largo plazo. Es por eso que la finalidad de este proyecto es generar capacidades de análisis y diagnósticos periódicos basado en indicadores logísticos clave con relevancia para el sector gubernamental y empresarial.

El Mapa Digital es un proyecto aún en desarrollo, y la finalidad de la implementación del proyecto propuesto en este documento, es mejorar la infraestructura tecnológica que sirva como base para el Mapa Digital y otros proyectos actuales en el Laboratorio Nacional en Sistemas de Transporte y Logística, para mejorar los resultados que se ofrecen a las instituciones y organizaciones vinculadas, además de ofrecer una oportunidad de mejora de los servicios computacionales ofrecidos como IMT.

El servicio web desarrollado se basa en servicios REST que permiten la manipulación de la información del Mapa Digital. Para la prueba de las aplicaciones a desarrollar se generó un servicio que se enfocará solo en la base de datos de las estaciones nacionales de combustible, y a través de dicho dispositivo se puede experimentar con la inserción, actualización o eliminación de estas estaciones; además a este servicio solo se puede acceder de manera local debido a las funcionalidades que se desarrollaron ya que aún no tiene implementado un sistema de seguridad confiable para evitar operaciones no permitidas por externos.

```
https://192.168.43.1:8000/MapaDigital/estaciones/{Operaciones}
```

Fuente: Elaboración propia

Figura 4.5 Formato de peticiones al Mapa Digital

Para las operaciones desarrolladas en esta API, se tomaron en cuenta las más básicas que se pueden realizar en una base de datos:

- **agregar.-** envía un objeto Estación que lo inserta en la BD
- **actualizar.-** recibe un objeto Estación que actualiza la información
- **eliminar.-** elimina de la BD el objeto enviado en la petición
- **buscar.-** recibe la información latitud y longitud y devuelve la cantidad de estaciones de combustible que se encuentren cercanas.

Para los propósitos de esta investigación se utilizó la funcionalidad de **buscar**, y es la que se muestra en las aplicaciones desarrolladas, además los parámetros que se necesitan en este servicio web se tienen que enviar como información dentro del cuerpo de la petición, es decir, no puede visualizarse la información que se manipula en la URL de la petición. En el caso del resultado JSON de la petición **buscar**, el servidor nos devuelve una lista de las estaciones que se encuentran en la zona, y, por lo tanto, la manipulación de esta información se codifica en el lado cliente (la aplicación de RA), entonces solo se muestra la cantidad total de estaciones, pero se pueden realizar otros análisis para mostrar en nuestra aplicación final.

```
[
  {
    "id": "mv_estacion_combustible.fid--154202bf_1628bf81afc_-69e5",
    "calirepr": "Definida",
    "clase": "Estación de Abastecimiento de Combustible",
    "fecha_act": "2017-06-23",
    "gid": "9899",
    "id_pi": "10205",
    "nombre": "SoniGas",
    "nombre_alt": "N/D",
    "subclase": "Estación de Carburación",
    "longitud": -100.104513,
    "latitud": 20.468325
  },
  ...
]
```

Fuente: Elaboración propia

Figura 4.6 Resultado obtenido de una petición al servicio web de Mapa Digital API

4.3 Aplicaciones desarrolladas

Como ejemplo se desarrollaron dos pequeñas aplicaciones, la primera se desarrolló para Android 5.1 la cual recolectaba la información de los servicios web, y mostraba los resultados en forma de texto, esta funcionalidad en el sistema es la base para la comunicación entre diferentes sistemas informáticos desarrollados con diferentes tecnologías y que puedan comunicarse a través de esta arquitectura.

Dentro de esta aplicación se incluyeron tres funcionalidades, la primera es la solicitud automática de la posición geoespacial del dispositivo a través de una petición a la *API* interna del sistema operativo, la cual nos brindaría la latitud y la longitud en tiempo real, tal como se muestra en la Figura 4.7:



Fuente: Elaboración propia

Figura 4.7 Resultado de la petición a la API interna del sistema operativo

Partiendo de esta petición de la ubicación, ya se pueden realizar peticiones a cualquier servicio web que se proponga, en este caso, en la aplicación incluimos dos servicios web, la primera petición es a *Google Maps* (ver capítulo 4.2.1), y la segunda es a *OpenWeather* (ver capítulo 4.2.2). Los resultados para cada una de estas peticiones fueron limitados a solo ciertos atributos de la respuesta *JSON*, con el objetivo de no llenar de texto la pantalla, ya que la pantalla del dispositivo de realidad aumentada es demasiado pequeña y se necesita ser muy conciso en los datos a mostrar. Dichos resultados se pueden visualizar en la aplicación tal como se muestra en la Figura 4.8:



Fuente: Elaboración propia

Figura 4.8 Resultados de las peticiones a los servicios web de *GoogleMaps* y *OpenWeather*

Tal como se explicó en el capítulo anterior, el texto mostrado en la aplicación es muy poca información comparada a lo que realmente las peticiones a dichos servicios nos pueden ofrecer. La idea principal de este desarrollo es ejemplificar los resultados de las peticiones a los servicios web, es por eso que los resultados *JSON* de cada una de las peticiones se guarda en variables internas del sistema para un uso posterior dependiente de las funcionalidades que se requieran desarrollar, y en este caso solo se muestra uno de los atributos, en el caso de *GoogleMaps* es la dirección en forma de texto representada por la latitud y la longitud, y en *OpenWeather* es el tipo de clima que se encuentra en dicha ubicación descrita en una cadena de texto.

Para la segunda aplicación se desarrolló en Unity, está dividida en dos partes, la primera se conforma de una interfaz que realizará el rastreo de las imágenes implementadas en la base de datos, dicha interfaz se puede observar en la Figura 4.9.



Fuente: Elaboración propia

Figura 4.9 Pantalla Principal de la aplicación desarrollada con *Vuforia*

La segunda parte de la aplicación hace referencia a una interfaz que se activará cuando los algoritmos dentro del framework de Vuforia identifiquen algunas de las imágenes que se encuentran guardadas en la base de datos de imágenes objetivo, tal como se muestra en la Figura 4.10. La interfaz cambiará dependiendo de la imagen que identifique, para y generará un objeto en 3D de manera virtual; además también se incluirá una caja de texto con información relevante para la aplicación y un botón de actualizar para resolver el problema de *delay* del dispositivo.



Fuente: Elaboración propia

Figura 4.10 Pantalla con identificación de alguna imagen u objeto

Finalmente, al presionar en el botón de actualizar, se modificará el cuadro de texto con la información geoespacial del dispositivo, y dependiendo de dicha latitud y longitud, realizará peticiones a los tres servicios web y mostrará el resultado respectivo, tal como se muestra en la Figura 4.11. Tal como se realizó en la primera aplicación, solo se muestra uno de los atributos obtenidos del resultado JSON de cada petición, con el objetivo de mostrar la funcionalidad.



Fuente: Elaboración propia

Figura 4.11 Interfaz con los resultados obtenidos de los servicios web

Conclusiones

En las últimas décadas, el avance científico ha logrado generar un gran número de tecnologías que pueden ser aplicadas a diferentes sectores y que se han convertido en herramientas clave para la solución y optimización de muchos problemas diarios en estos sectores.

Tecnologías tales como Inteligencia Artificial, Big Data, IoT, Realidad Inmersiva, son solo algunas que actualmente se han desarrollado y han sido adoptadas por industria, academia y gobierno para facilitar actividades que pueden ser automatizadas. Dentro de estas, la realidad inmersiva ha sido explotada para la agilización de capacitación en muchas organizaciones debido a la facilidad de simular situaciones reales en entornos controlados. De su lado, la realidad aumentada ha sido utilizada ampliamente en dispositivos móviles con muchos propósitos, pero actualmente, debido a los lentes inteligentes que soportan esta tecnología, se ha incrementado su uso en la industria para facilitar actividades sin tener conocimientos previos, ya que muestra de manera virtual toda la información necesaria para que personas sin conocimiento alguno, puedan realizar dichas actividades.

Por otro lado, los servicios web han aumentado su popularidad debido a la capacidad de permitir la comunicación entre sistemas digitales independizando la tecnología con la cual se desarrollen, y actualmente, con todo el enfoque digital que se le está dando a todas las actividades cotidianas en el mundo. Su uso es necesario para optimizar y automatizar operaciones entre sistemas. La implementación de un sistema web es sencilla y actualmente existen muchas plataformas que agilizan el desarrollo de cualquier sistema informático sobre este paradigma; además existen arquitecturas que tienen como base esta metodología, tal como lo es la arquitectura de microservicios, la cual agiliza el desarrollo de sistemas computacionales. Esta última modulariza sistemas complejos en componentes mucho más sencillos, autónomos y fáciles de desarrollar. Los resultados en el desarrollo de este proyecto mostraron la importante capacidad y oportunidad que la implementación de estas tecnologías puede atraer al sector transporte y logística.

El uso de realidad aumentada para identificar información importante tanto para los tomadores de decisiones de las organizaciones, así como para los transportistas, ayudarían a identificar datos de manera rápida y efectiva a un pequeño costo de los dispositivos o lentes inteligentes. De manera particular, las aplicaciones realizadas sirvieron para comprobar que la realidad aumentada tiene un gran potencial en el sector transporte y logística para tomadores de decisiones.

También las aplicaciones desarrolladas han servido como experimento de la aplicación de los servicios web existentes y como la aplicación de un servicio web interno del IMT-Laboratorio Nacional Conacyt en Sistemas de Transporte y Logística, esta primera aplicación servirá como base para que el desarrollo de un gran número de los proyectos ya existentes, puedan ser migrados a un sistema de este estilo y puedan implementarse otros paradigmas tales como una arquitectura de microservicios para optimizar la comunicación entre proyectos.

Otra de las futuras aplicaciones que se pueden desarrollar es implementar un sistema que sea capaz de identificar todos los señalamientos carreteros. Por lo tanto, se podría anidar información importante en las imágenes de esta base de datos, para proporcionar a los conductores otra información que no pueda ser visualizada solo con los señalamientos. Entre ellos serían riesgo de velocidad máxima en cada tramo carretero, riesgo de accidentalidad, confiabilidad de tiempos de tránsito, entre otros indicadores que solo se encuentran en una plataforma web, pero no pueden ser visualizados en todo momento por los conductores, y que serviría para mejorar los patrones de conducción a nivel nacional.

Bibliografía

- Adobe Systems Incorporated. (2020). *PhoneGap*. <https://build.phonegap.com/>
- Ajorloo, H., Sreenan, C., Bomfin, R., Danneberg, M., & Fettweis, G. (2020). MinDFul: Using double links for stabilizing mmWave wireless channels for application to autonomous vehicles and augmented reality. *Procedia Computer Science*, 175(1), 365–372. <https://doi.org/10.1016/j.procs.2020.07.052>
- American Society of Civil Engineers. (2017). *Computing in Civil Engineering 2017: Sensing, Simulation, and Visualization*. <https://doi.org/10.1061/9780784480830>
- American Society of Civil Engineers. (2019). *Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation*. <https://doi.org/10.1061/9780784482421>
- Apple. (2020a). *ARKit Apple*. <https://developer.apple.com/augmented-reality/arkit/>
- Apple. (2020b). *FACE ID*. <https://support.apple.com/es-lamr/HT208108#:~:text=La cámara TrueDepth captura datos,una imagen infrarroja de esta.>
- Apple. (2020c). *Xcode*. <https://developer.apple.com/xcode/>
- Aulinas, J., Petillot, Y., Salvi, J. & Llado, X. (2008). The SLAM problem: a survey. *Conference: Artificial Intelligence Research and Development*, 1(Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence), 22–24. <https://doi.org/10.3233/978-1-58603-925-7-363>
- Axway. (2020). *Titanium*. <https://www.appcelerator.com/Titanium/>
- Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S. & MacIntyre, B. (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6), 34–47. <https://doi.org/10.1109/38.963459>
- Bailey, T. & Durrant-Whyte, H. (2006). Simultaneous Localization and Mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3). <https://doi.org/10.1109/MRA.2006.1678144>
- Behzadan, A., Dong, S. & Kamat, V. (2015). Augmented reality visualization: A review of civil infrastructure system applications. *Advanced Engineering Informatics*, 29(2), 252–267. <https://doi.org/https://doi.org/10.1016/j.aei.2015.03.005>

- Burgard, W., Fox, D. & Jans, H. (1999). Sonar-Based Mapping With Mobile Robots Using EM. *CiteSeer*.
https://www.researchgate.net/publication/2582919_Sonar-Based_Mapping_With_Mobile_Robots_Using_EM
- Cabello, R. (2013). *Three.js*. <https://threejs.org/>
- Calvi, A., D'Amico, F., Ferrante, C. & Ciampoli, L. (2020a). Evaluation of augmented reality cues to improve the safety of left-turn maneuvers in a connected environment: A driving simulator study. *Accident Analysis & Prevention*, 148. <https://doi.org/10.1016/j.aap.2020.105793>
- Calvi, A., D'Amico, F., Ferrante, C. & Ciampoli, L. B. (2020b). Effectiveness of augmented reality warnings on driving behaviour whilst approaching pedestrian crossings: A driving simulator study. *Accident Analysis & Prevention*, 147. <https://doi.org/10.1016/j.aap.2020.105760>
- Cámara Nacional de Autotransporte [CANACAR]. (2019). *Robo Autotransporte*.
- Chang, J., Hsu, H. & Chao, S. (2012). Development of a Road Monitoring and Reporting System Based on Location-Based Services and Augmented-Reality Technologies. *Journal of Performance of Constructed Facilities*, 26(6), 812–823. [https://doi.org/10.1061/\(ASCE\)CF.1943-5509.0000272](https://doi.org/10.1061/(ASCE)CF.1943-5509.0000272)
- Davison, A. & Murray, D. (2002). Simultaneous Localization and Map-Building Using Active Vision. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 24(7). <https://doi.org/10.1109/TPAMI.2002.1017615>
- de Oliveira, N., Kandil, D. & Gabbard, J. (2019). *Augmented reality head-up displays effect on drivers' spatial knowledge acquisition* (Vol. 63, Issue 1, pp. 1486–1487). <https://doi.org/10.1177/1071181319631287>
- Delson Group Inc. (2019). *Mobile World Congress*. <https://www.mwcbarcelona.com/>
- Denys Gamers Tutorial. (2017). *Vuforia Tutorial Video*. <https://www.youtube.com/watch?v=n9xXPndegK0>
- DreamGlass. (2020a). *DreamGlass 4K/4K Plus*. <https://www.dreamworldvision.com/>
- DreamGlass. (2020b). *DreamGlass 4K/4K Plus Especificaciones*. <https://www.dreamworldvision.com/dream-glass-4k-4k-plus>
- Durrant-Whyte, H. & Bailey, T. (2006). Simultaneous localization and mapping: Part I. *IEEE Robotics & Automation Magazine*, 13(2), 99–110. <https://doi.org/10.1109/MRA.2006.1638022>
- Epson. (2020a). *Moverio BT-300*. <https://www.epson.es/products/see-through->

mobile-viewer/gafas-moverio-bt-300

Epson. (2020b). *Moverio BT-300 Especificaciones*.
<https://www.epson.es/products/see-through-mobile-viewer/gafas-moverio-bt-300#specifications>

Eyraud, R., Zibetti, E. & Baccino, T. (2015). Allocation of visual attention while driving with simulated augmented reality. *Transportation Research Part F: Traffic Psychology and Behaviour*, 32, 46–55. <https://doi.org/10.1016/j.trf.2015.04.011>

Facebook. (2013). *React*. <https://github.com/facebook/react>

Feng, S., Feng, Y., Yan, X., Shen, S., Xu, S. & Liu, H. (2020). Safety Assessment of Highly Automated Driving Systems in Test Tracks: A New Framework. *Accident Analysis & Prevention*, 144. <https://doi.org/10.1016/j.aap.2020.105664>

Fielding, T. (2000). *Architectural Styles and the Design of Network-based Software Architectures* [UNIVERSITY OF CALIFORNIA].
<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

Free Software Foundation Inc. (2007). *GNU General Public License*.
<http://www.gnu.org/licenses/gpl-3.0.html>

Ginters, E. (2019). Augmented reality use for cycling quality improvement. *Procedia Computer Science*, 149, 167–176. <https://doi.org/10.1016/j.procs.2019.01.120>

Golparvar-Fard, M., Balali, V., & de la Garza, J. M. (2015). Segmentation and Recognition of Highway Assets Using Image-Based 3D Point Clouds and Semantic Texton Forests. *Journal of Computing in Civil Engineering*, 29(1), 401–402–403. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000283](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000283)

Gomes, P., Olaverri-Monreal, C. & Ferreira, M. (2012). Making Vehicles Transparent Through V2V Video Streaming. *IEEE Transactions on Intelligent Transportation Systems*, 13(2), 930–938. <https://doi.org/10.1109/tits.2012.2188289>

González Quiroga, M. (2011). *ESTUDIO DE ARQUITECTURAS DE REDES ORIENTADAS A SERVICIO* [Universitat Politècnica de Catalunya].
https://upcommons.upc.edu/bitstream/handle/2099.1/12312/ESTUDIO_DE_ARQUITECTURAS_DE_REDES_ORIENTADAS_A_SERVICIO.pdf?sequence=1

Google. (2020a). *ARCore*. <https://developers.google.com/ar/discover>

Google. (2020b). *ARCore Fundamentals*.
<https://developers.google.com/ar/discover/concepts>

Google. (2020c). *Google Developers*. <https://developers.google.com/>

- Google. (2020d). *Google Geocoding API*. <https://developers.google.com/maps/documentation/geocoding/overview?hl=es#Geocoding>
- Google. (2020e). *Google Glass Enterprise*. <https://www.google.com/glass/start/>
- Google. (2020f). *Google Glass Enterprise Especificaciones*. <https://www.google.com/glass/tech-specs/>
- Google. (2020g). *Google Maps Platform*. <https://developers.google.com/maps/documentation?hl=es>
- Grabowski, M., Rowen, A. & Rancy, J. (2018). Evaluation of Wearable Immersive Augmented Reality Technology in Safety-Critical Systems. *Safety Science*, 103(1), 23–32. <https://doi.org/10.1016/j.ssci.2017.11.013>
- Gregoriades, A. & Sutcliffe, A. (2018). Simulation-based evaluation of an in-vehicle smart situation awareness enhancement system. *Ergonomics*, 61(7), 947–965. <https://doi.org/10.1080/00140139.2018.1427803>
- Hartmann, M., Viehweger, M., Stolz, M., Watzenig, D., Spitzer, M. & Desmet, W. (2018). Pedestrian in the Loop: An Approach Using Augmented Reality. *SAE International Journal of Engines*, 11(6), 1053–2018. <https://doi.org/10.4271/2018-01-1053>
- Heinen, S. (2010). *DroidAR*. <https://bitstars.github.io/droidar/>
- HezareMoghadam, N., Khoshbakhtian, M., Yeganeh, H. & Darvishan, A. (2008). WEB Service Interoperability Analysis and Introduction of a Design Method to reduce non Interoperability Effects. *IJCSNS*, 8, 149–153.
- Hussain, K., Radwan, E. & Moussa, G. (2013). Augmented Reality Experiment: Drivers' Behavior at an Unsignalized Intersection. *IEEE Transactions on Intelligent Transportation Systems*, 14(2), 608–617. <https://doi.org/10.1109/TITS.2012.2226239>
- Instituto Nacional de Estadística y Geografía [INEGI]. (2019). *Transportes*. <https://www.inegi.org.mx/temas/transportes/>
- Jolovic, D., Martin, P. & Stevanovic, A. (2016). Left-Turn Phasing—State-of-the-Art Review and the Outlook for Future Improvements. *16th COTA International Conference of Transportation*, 1624–1644. <https://doi.org/10.1061/9780784479896.149>
- Kim, B., Kim, C. & Kim, H. (2012). Interactive Modeler for Construction Equipment Operation Using Augmented Reality. *Journal of Computing in Civil Engineering*, 26(3), 331–341. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000137](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000137)

- Kim, H. & Gabbard, J. (2018). Quantifying Distraction Potential of Augmented Reality Head-Up Displays for Vehicle Drivers. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 62(1), 1923. <https://doi.org/10.1177/1541931218621436>
- Kim, S. & Dey, A. (2015). Augmenting Human Senses to Improve the User Experience in Cars: Applying Augmented Reality and Haptics Approaches to Reduce Cognitive Distances. *Multimedia Tools and Applications*, 1–21. <https://doi.org/10.1007/s11042-015-2712-4>
- Koduri, T., Bogdoll, D., Paudel, S. & Sholingar, G. (2018). AUREATE: An Augmented Reality Test Environment for Realistic Simulations. *SAE International Journal of Engines*, 11(6). <https://doi.org/10.4271/2018-01-1080>
- Krafzig, D., Banke, K. & Slama, D. (2005). *Enterprise SOA: Service-oriented Architecture Best Practices*.
- Li, H. & Nashashibi, F. (2011). Multi-vehicle cooperative perception and augmented reality for driver assistance: A possibility to 'see' through front vehicle. *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 242–247. <https://doi.org/10.1109/ITSC.2011.6083061>
- Magic Leap. (2020). *Magic Leap Especificaciones*. <https://www.magicleap.com/magic-leap-2>
- Malavolta, I., Ruberto, S., Soru, T. & Terragni, V. (2015). Hybrid Mobile Apps in the Google Play Store: An Exploratory Investigation. *2nd ACM International Conference on Mobile Software Engineering and Systems*. <https://doi.org/10.1109/MobileSoft.2015.15>
- Mallela, J., Gilson, K., Goodrum, P. & Steen, J. (2020). *Leveraging Augmented Reality for Highway Construction*. <https://www.fhwa.dot.gov/publications/research/infrastructure/pavements/20038/20038.pdf>
- Meng, M., Steinhardt, S. & Schubert, A. (2018). Application programming interface documentation: What do software developers want? *Journal of Technical Writing and Communication*, 48(3), 295–330. <https://doi.org/10.1177/0047281617721853>
- Microsoft. (2008). *GitHub*. <https://github.com/>
- Microsoft. (2020a). *Microsoft HoloLens*. <https://www.microsoft.com/es-es/hololens>
- Microsoft. (2020b). *Microsoft HoloLens Especificaciones*. <https://www.microsoft.com/es-mx/hololens/hardware>
- Microsoft. (2020c). *Xamarin*. <https://dotnet.microsoft.com/apps/xamarin>

- Milgram, P. (2011). a Taxonomy of Mixed Reality Visual Displays. *Industrial Engineering*, 12, 1–14.
- Montemerlo, M., Thrun, S., Koller, D. & Wegbreit, B. (2002). FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. *Proceedings of the AAAI National Conference on Artificial Intelligence*.
- Naminski, M. (2013). *An Analysis of Simultaneous Localization and Mapping (SLAM) Algorithms* [Macalester].
https://digitalcommons.macalester.edu/cgi/viewcontent.cgi?article=1030&context=mathcs_honors
- Niknejad, N., Ismail, W., Ghani, I., Nazari, B., Bahari, M. & Hussin, A. (2020). Understanding Service-Oriented Architecture (SOA): A systematic literature review and directions for further investigation. *Information Systems*, 91, 101491.
<https://doi.org/10.1016/j.is.2020.101491>
- Nomura, J. & Sawada, K. (1999). Virtual reality technology and its industrial applications. In *Control Engineering Practice*, 7 (11).
[https://doi.org/10.1016/S0967-0661\(99\)00114-8](https://doi.org/10.1016/S0967-0661(99)00114-8)
- OASIS. (2020a). OASIS. <https://www.oasis-open.org/>
- OASIS. (2020b). UDDI. <https://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm>
- Ofoeda, J., Boateng, R., & Effah, J. (2019). Application programming interface (API) research: A review of the past to inform the future. *International Journal of Enterprise Information Systems*, 15(3), 76–95.
<https://doi.org/10.4018/IJEIS.2019070105>
- OH, J., Park, S. & Kwon, O.-S. (2016). Advanced Navigation Aids System based on Augmented Reality. *International Journal of E-Navigation and Maritime Economy*, 5, 21–31. <https://doi.org/10.1016/j.enavi.2016.12.002>
- OpenWeather. (2020). *OpenWeather*. <https://openweathermap.org/>
- PTC Inc. (2020). *Vuforia*. <https://developer.vuforia.com/>
- Red Hat. (2020). ¿QUÉ ES UNA API? <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>
- Sagdeo, P. (2019). Harvard Journal of Law & Technology Volume 33 , Number 1 Fall 2019 M EDICAL AI AND C ONTEXTUAL B IAS Harvard Journal of Law & Technology. *Harvard Journal of Law & Technology*, 33(1).
- Secretaria de Comunicaciones y Transportes [SCT]. (2020). *Programa Sectorial de Comunicaciones y Transportes 2020-2024*.

- Secretaría de Comunicaciones y Transportes [SCT]. (2018). *NOM-087-SCT-2-2017*. México: Diario Oficial de la Federación https://dof.gob.mx/nota_detalle.php?codigo=5529381&fecha=28/06/2018
- Secretaria de Comunicaciones y Transportes [SCT]. (2013). *Infraestructura de Transporte*. http://www.sct.gob.mx/uploads/media/Presentacion_RMC_Infraestructura_de_Transporte_2013-2018_01.pdf
- Secretaria de Comunicaciones y Transportes [SCT]. (2019a). *Autotransporte Federal*. <http://www.sct.gob.mx/transporte-y-medicina-preventiva/autotransporte-federal/%0A%0A>
- Secretaria de Comunicaciones y Transportes [SCT]. (2019b). *Autotransporte Marco Jurídico*. <http://www.sct.gob.mx/transporte-y-medicina-preventiva/autotransporte-federal/marco-normativo/>
- Secretaria de Comunicaciones y Transportes [SCT]. (2020). *Estadística Básica del Autotransporte Federal 2020*. https://www.sct.gob.mx/fileadmin/DireccionesGrales/DGAF/EST_BASICA/EST_BASICA_2019/Estadística_Básica_del_Autotransporte_Federal_2020_1.pdf
- Supermedium, Google & WebVR. (2015). *A-Frame*. <https://aframe.io/>
- Takeda, K., Ishihara, K. & Kawamorita, T. (2018). A Sense of Distance and Augmented Reality for Stereoscopic Vision. *SAE International Journal of Engines*, 11(6), 1036–2018. <https://doi.org/10.4271/2018-01-1036>
- Thrun, S. (2003). Robotic mapping: A survey. In *Exploring artificial intelligence in the new millennium*. <https://doi.org/10.5555/779343.779345>
- Tom, T. (2020). *Traffic Index*. https://www.tomtom.com/en_gb/traffic-index/
- Tong, Y. & Jia, B. (2019). *An Augmented-reality-based Warning Interface for Pedestrians: User Interface Design and Evaluation* (Vol. 63, Issue 1, p. pp 1834-1838). <https://doi.org/10.1177/1071181319631413>
- Vuforia. (2020). *Vuforia Tutorial*. <https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.html>
- Vuzix. (n.d.). *Blade Smart Glasses Especificaciones*. <https://vuzix-website.s3.amazonaws.com/files/Content/pdfs/Vuzix-Blade-Upgraded-Smart-Glasses-d07.pdf>
- Vuzix. (2018). *Vuzix Blade Nativo*. <https://www.vuzix.com/Developer/KnowledgeBase/Detail/63>
- Vuzix. (2020a). *Blade Smart Glasses*. <https://www.vuzix.com/products/blade-smart->

glasses-upgraded

Vuzix. (2020b). *Vuzix Companion* *Android*.
https://play.google.com/store/apps/details?id=com.vuzix.companion&hl=es_MX&gl=US

Vuzix. (2020c). *Vuzix Companion iPhone*. <https://apps.apple.com/us/app/vuzix-companion/id1383316233>

Wickens, C., Dempsey, G., Pringle, A., Kazansky, L. & Hutka, S. (2018). Developing and Evaluating an Augmented Reality Interface to Assist the Joint Tactical Air Controller by Applying Human Performance Models. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 62(1), 686–690. <https://doi.org/10.1177/1541931218621155>

Woody, S., Burdick, D., Lapp, H. & Huang, E. (2020). Application programming interfaces for knowledge transfer and generation in the life sciences and healthcare. *Npj Digital Medicine*, 3(1), 1–5. <https://doi.org/10.1038/s41746-020-0235-5>

World Wide Web Consortium [W3C]. (2020a). *REST*. <https://www.w3.org/2001/sw/wiki/REST>

World Wide Web Consortium [W3C]. (2020b). *SOAP*. <https://www.w3.org/TR/soap/>

World Wide Web Consortium [W3C]. (2020c). *W3C*. <https://www.w3.org/>

World Wide Web Consortium [W3C]. (2020d). *Web Service*. <https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>

World Wide Web Consortium [W3C]. (2020e). *Web Service Architecture*. <https://www.w3.org/DesignIssues/WebServices.html>

World Wide Web Consortium [W3C]. (2020f). *WSDL*. <https://www.w3.org/TR/2001/NOTE-wsdl-20010315>

World Wide Web Consortium [W3C]. (2020g). *XML-RPC*. <http://xmlrpc.com/>

World Wide Web Consortium [W3C]. (2020h). *XML*. <https://www.w3.org/XML/>

Xu, J., Xue, F., Chiaradia, A., Lu, W. & Cao, J. (2020). *Indoor-Outdoor Navigation without Beacons: Compensating Smartphone AR Positioning Errors with 3D Pedestrian Network*. 444–452. <https://doi.org/10.1061/9780784482858.049>

Zappar. (2020). *ZapWorks*. <https://zap.works/>

Zhou, P., Braud, T., Zavodovski, A., Liu, Z., Chen, X., Hui, P. & Kangasharju, J. (2020). Edge-Facilitated Augmented Vision in Vehicle-to-Everything Networks. *IEEE Transactions on Vehicular Technology*, 69(10), pp 12187-12201.

<http://dx.doi.org/10.1109/TVT.2020.3015127>

Zimmermann, M., Schopf, D., Lütteken, N., Liu, Z., Storost, K., Baumann, M., Happee, R. & Bengler, K. (2018). Carrot and stick: A game-theoretic approach to motivate cooperative driving through social interaction. *Transportation Research Part C: Emerging Technologies*, 88, 159–175. <https://doi.org/10.1016/j.trc.2018.01.017>



COMUNICACIONES
SECRETARÍA DE INFRAESTRUCTURA, COMUNICACIONES Y TRANSPORTES



Km 12+000 Carretera Estatal 431 “El Colorado Galindo”
Parque Tecnológico San Fandila, Mpio. Pedro Escobedo,
Querétaro, México. C.P. 76703
Tel: +52 (442) 216 97 77 ext. 2610
Fax: +52 (442) 216 9671

publicaciones@imt.mx

<http://www.imt.mx/>