



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**Desarrollo de una aplicación móvil de realidad aumentada con el framework
XAMARIN; caso práctico: Reconocimiento y obtención de información de señales de
tránsito**

Gualpa Yanchatuña, Kevin Omar

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Trabajo de Unidad de Integración Curricular, previo a la obtención del Título de Ingeniero
de Software

Msc. Montaluisa Yugla, Franklin Javier

01 de marzo de 2024

Latacunga - Ecuador



Plagiarism report

Gualpa_Kevin_Testis_NEW.docx

Scan details

Scan Date: February 29th, 2024 at 20:27 UTC	Total Pages: 62	Total Words: 15438
--	--------------------	-----------------------

Plagiarism Detection



3.3%

Types of plagiarism		Words
Identical	1%	149
Minor Changes	0.7%	103
Paraphrased	1.5%	233
Omitted Words	6%	929

AI Content Detection



N/A

Text coverage

AI text

Human text

Plagiarism Results: (17)

- ¿Qué es Xamarin? - Xamarin | Microsoft Learn**

<https://learn.microsoft.com/es-es/xamarin/get-started/what-is-xamarin>

Ir al contenido principal Este explorador ya no se admite. Actualice a Microsoft Edge para aprovechar L...

1.5%
- Clase digital 6. Desarrollo de Aplicaciones - Recursos Educativos Abiertos**

<https://blogs.ugto.mx/rea/clase-digital-6-desarrollo-de-aplicaciones/>

Saltar al contenido Recursos Educativos Abiertos Clase digital 6. Desarrollo de Aplicaciones 11 de abril de 202230 de marzo de ...

0.9%
- T-3859.pdf?sequence=1&isAllowed=y**

<https://repositorio.ursea.bo/bitstream/handle/123456789/29644/T-3859.pdf?sequence=1&isAllowed=y>

daniel orias yamaguchi

UNIVERSIDAD MAYOR DE SAN ANDRÉS FACULTAD DE CIENCIAS PURAS Y NATURALES CARRERA DE INFORMATICA TESIS DE GRADO "EL USO DE REPRODUCCIÓN DE ...

0.6%
- 14838.pdf**

<https://repositorio.unan.edu.ni/14838/2/14838.pdf>

Carlos Nuñez

Universidad Nacional Autónoma de Nicaragua Recinto Universitario Rubén Darío Facultad de Ciencias e Ingeniería Departamento de Computación...

0.6%

Certified by



About this report

help.copyleaks.com

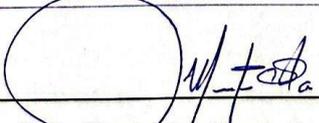
copyleaks.com



MSc. Yugla, Franklin

Javier

C.C.: 0502166796



MSc. Montaluisa Yugla, Franklin
Javier

C.C.: 0502166796



Departamento de Ciencias de la Computación

Carrera de Software

Certificación

Certifico que el trabajo de Unidad de integración curricular: "Desarrollo de una aplicación móvil de realidad aumentada con el framework XAMARIN; caso práctico: Reconocimiento y obtención de información de señales de tránsito" fue realizado por el señor Gualpa Yanchatuña, Kevin Omar el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 01 de marzo del 2024



MSc. Montaluisa Yugla, Franklin Javier
C.C.: 0502166796



Departamento de Ciencias de la Computación

Carrera de Software

Responsabilidad de autoría

Yo, **Gualpa Yanchatuña, Kevin Omar**, con cédula de ciudadanía n° 1804413076, declaro que el contenido, ideas y criterios del trabajo de Unidad de integración curricular: **"Desarrollo de una aplicación móvil de realidad aumentada con el framework XAMARIN; caso práctico: Reconocimiento y obtención de información de señales de tránsito"** es demi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 01 de marzo del 2024

Gualpa Yanchatuña, Kevin Omar

C.C.: 1804413076



Departamento de Ciencias de la Computación

Carrera de Software

Autorización de publicación

Yo, **Gualpa Yanchatuña, Kevin Omar**, con cédula de ciudadanía n° 1804413076, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de Unidad de integración curricular: **"Desarrollo de una aplicación móvil de realidad aumentada con el framework XAMARIN; caso práctico: Reconocimiento y obtención de información de señales de tránsito"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, 01 de marzo del 2024

Gualpa Yanchatuña, Kevin Omar

C.C.: 1804413076

Dedicatoria

Con infinito amor y gratitud dedico este trabajo, a mis padres, Gustavo Gualpa y Gladys Yanchatuña, quienes con su cariño, sacrificio y sabiduría han sido pilar fundamental para guiar mi camino. Este momento especial en mi vida académica es el fruto de su incansable apoyo, trabajo y fe depositada en mí. A ellos, y a toda mi familia, que ha sido parte fundamental de mi proceso de vida, les dedico este trabajo, como un pequeño gesto de gratitud por todo su apoyo y cariño desde mis primeros años de vida.

A mi querida hermana, Edith Gualpa, le dedico un espacio muy especial en mi corazón y en estas líneas. Edith, tu alegría, inocencia y tus abrazos han sido mi refugio y alivio es mis mayores momentos de estrés y cansancio. Eres un recordatorio diario de la importancia de la familia y del amor incondicional entre hermanos, este logro es también un homenaje a ti y un recordatorio de que, aunque el camino es largo y difícil de recorrer con perseverancia y esfuerzo tú vas a lograr llegar lejos en la vida.

A mis amigos, Stalin, Angie, Darío, Bryan, y Elías, quienes han estado a mi lado en los momentos más desafiantes y alegres. Ustedes han formado parte de incontables recuerdos felices en mi vida, también han sido apoyo y soporte , quiero darles las gracias por las tardes de estudio, las conversaciones profundas y las distracciones sanas que hemos compartido juntos, esperando que nuestra amistad perdure en el tiempo y la distancia.

Finalmente, este logro no es solo mío, sino de todos quienes, de una u otra forma, han contribuido a mi crecimiento personal y académico. A todos, mi más sincero agradecimiento.

Gualpa Yanchatuña, Kevin Omar

Agradecimiento

Agradezco a Dios que me ha brindado salud, vida y fuerzas que me han permitido alcanzar un nuevo logro en mi vida. A esta altura pienso y reflexiono sobre las personas que han hecho posible alcanzar este logro, a ellos les expreso mi total gratitud.

A mis padres y hermana, cuya inagotable fuente de amor y apoyo ha sido mi motivación a lo largo de este viaje, a mis abuelos, cuya sabiduría y valores me han guiado en cada paso del camino, y a mis tíos, por su generosidad y por estar siempre dispuestos a ayudar. Cada uno de ustedes ha desempeñado un papel crucial en mi desarrollo personal y académico.

además, Quiero expresar mi más profundo agradecimiento a la Universidad de las Fuerzas Armadas, por brindarme un ambiente académico de excelencia y oportunidades inigualables para mi crecimiento intelectual y profesional. Mi gratitud también a mi tutor, por su guía, paciencia y apoyo a lo largo de este proyecto. También, agradezco sinceramente a mis compañeros de universidad, por el apoyo mutuo y los momentos compartidos. Cada uno de ustedes ha contribuido a hacer de este proceso académico más memorable.

En un momento de reflexión, también quiero tomar este espacio para agradecerme por la perseverancia frente a los obstáculos, por la dedicación y el esfuerzo, y por nunca perder de vista mis objetivos incluso cuando el camino era difícil. Por cumplir el sueño de un adolescente de estudiar y graduarse en la universidad de las fuerzas armadas, que fue su propósito desde que termino el colegio.

Gualpa Yanchatuña, Kevin Omar

ÍNDICE DE CONTENIDOS

Carátula	1
Reporte de Verificación de Contenido.....	2
Certificación	3
Responsabilidad de autoría	4
Autorización de publicación	5
Dedicatoria	6
Agradecimiento.....	7
Índice de contenidos	8
Índice de figuras	11
Índice de tablas	13
Resumen.....	14
Capítulo I: Introducción.....	16
Propósito y contexto del proyecto.....	17
Justificación de la investigación.....	18
Objetivos Generales y específicos.....	18
<i>Objetivo General</i>	<i>18</i>
<i>Objetivo específico</i>	<i>18</i>
Metodología de la investigación.....	19
Capítulo II: Marco teórico	21
Realidad Aumentada	21
<i>Definición y Características</i>	<i>21</i>
<i>Tipos de realidad aumentada.....</i>	<i>22</i>

Señales de tránsito.....	23
<i>Importancia y función en la vía pública.....</i>	<i>23</i>
<i>Clasificación de las señales de tránsito.....</i>	<i>24</i>
Aplicaciones móviles y tecnologías asociadas	25
<i>Evolución de las aplicaciones móviles</i>	<i>25</i>
<i>Xamarin como framework de Desarrollo.....</i>	<i>26</i>
<i>Ventajas y desafíos de Xamarin en el desarrollo móvil</i>	<i>28</i>
Capítulo III: Estado del arte	30
Aplicaciones existentes de Realidad Aumentada para Señales de Tránsito	30
Aplicaciones desarrolladas con Xamarin en Realidad Aumentada	31
Tecnologías Utilizadas en Aplicaciones de Realidad Aumentada con Xamarin	32
Identificación de Brechas y Oportunidades en Soluciones Actuales.....	33
Capítulo IV: Diseño de la aplicación	35
Arquitectura General del sistema.....	36
Flujo de la aplicación	37
Interfaz de usuario y experiencia de usuario	39
Proceso de Reconocimiento de Señales de Tránsito en Realidad Aumentada	43
Capítulo V: Desarrollo e Implementación.....	45
Roles de la metodología Scrum	48
Planteamiento de las historias de usuario	49
Producto backlog del proyecto	50
Selección de la plataforma de desarrollo	51
Programación de la funcionalidad de reconocimiento de señales de tránsito	52

<i>Sprint 01: Generación de conjunto de datos (DataSet) de las señaléticas de tránsito</i>	53
<i>H.U.01: Generación del DataSet de las señales de tránsito.....</i>	53
<i>H.U.02: Desarrollo del algoritmo de detección de señaléticas de tránsito.</i>	56
<i>Sprint 02. Desarrollo de la funcionalidad de mostrar información de las señaléticas de tránsito vigentes</i>	64
<i>H.U.03: Desarrollo de la opción de tipos de señales.....</i>	64
<i>H.U.04: Desarrollo de la opción ley de tránsito de la aplicación</i>	70
<i>Sprint 3. Desarrollo de un sistema web para la administración de la información de las señaléticas de tránsito.</i>	73
<i>H.U.05: Desarrollo de sistema web para la gestión de la API.....</i>	73
Capítulo VI: Evaluación y Resultados	80
Validación del sistema	80
Marco de Validación.....	80
Resultados Obtenidos y Análisis	82
Capítulo VII: Conclusiones y Recomendaciones.....	88
Conclusiones.....	88
Recomendaciones.....	89
Bibliografía	90
Anexos.....	94

ÍNDICE DE FIGURAS

Figura 1 <i>Formas de las señales de tránsito</i>	25
Figura 2 <i>Funcionamiento de Xamarin</i>	27
Figura 3 <i>Arquitectura cliente - servidor</i>	36
Figura 4 <i>Arquitectura mvvm</i>	37
Figura 5 <i>Flujo del sistema</i>	39
Figura 6 <i>Pantalla de inicio app</i>	40
Figura 7 <i>Pantalla de opciones</i>	40
Figura 8 <i>Funcionalidad de la opción captura</i>	41
Figura 9 <i>Funcionalidad opción tipos de señales</i>	42
Figura 10 <i>Funcionalidad opción ley de tránsito</i>	42
Figura 11 <i>Ciclo de vida de Scrum</i>	47
Figura 12 <i>Clasificación de las señaléticas</i>	56
Figura 13 <i>Arquitectura BackEnd</i>	61
Figura 14 <i>Comprobación de funcionamiento de BackEnd</i>	62
Figura 15 <i>Aplicación móvil con implementación de detección de señales</i>	63
Figura 16 <i>Arquitectura de la API de información</i>	69
Figura 17 <i>Página de información de señales</i>	70
Figura 18 <i>Diseño móvil de vistas de descarga de manual</i>	72
Figura 19 <i>Arquitectura web</i>	75
Figura 20 <i>Página de inicio de sistema web</i>	76
Figura 21 <i>Página de formulario de registro</i>	76
Figura 22 <i>Página de registro web</i>	77
Figura 23 <i>Página principal del sistema web</i>	77
Figura 24 <i>Formulario para subir nueva información</i>	78
Figura 25 <i>Opciones de editar y eliminar una señalética de tránsito</i>	78
Figura 26 <i>formulario de editar una señal</i>	79

Figura 27 <i>Página de información de servicios de emergencia</i>	79
Figura 28 <i>Escenario de validación 1</i>	81
Figura 29 <i>Escenario de validación 2</i>	81
Figura 30 <i>Escenario de validación 3</i>	82
Figura 31 <i>Resultados escenario 1</i>	83
Figura 32 <i>Resultado de escenario 2</i>	85
Figura 33 <i>Resultado de validación 3</i>	86

ÍNDICE DE TABLAS

Tabla 1 <i>Roles en la metodología SCRUM</i>	46
Tabla 2 <i>Equipos de desarrollo Scrum</i>	48
Tabla 3 <i>Historia de usuario 01</i>	49
Tabla 4 <i>Product Backlog 1</i>	50
Tabla 5 <i>Herramientas y plataformas de desarrollo</i>	52
Tabla 6 <i>Historia de usuario 01 detallada</i>	54
Tabla 7 <i>Sprint Backlog 2</i>	55
Tabla 8 <i>Historia de usuario 2</i>	57
Tabla 9 <i>Sprint Backlog 3</i>	58
Tabla 10 <i>pseudocódigo servidor de reconocimiento de señales de tránsito</i>	59
Tabla 11 <i>historia de usuario 3</i>	64
Tabla 12 <i>Sprint Backlog 4</i>	65
Tabla 13 <i>algoritmo de gestión de información de señalización vial</i>	66
Tabla 14 <i>Historia de Usuario 4</i>	70
Tabla 15 <i>Sprint Backlog 5</i>	71
Tabla 16 <i>Historia de Usuario 5</i>	73
Tabla 17 <i>Sprint Backlog 6</i>	74
Tabla 18 <i>Lista de chequeo 1</i>	83
Tabla 19 <i>Lista de chequeo 2</i>	85

Resumen

En el siguiente proyecto se propone desarrollar una aplicación móvil de realidad aumentada con el uso del framework Xamarin, esta herramienta nos permitirá aprovechar sus capacidades de desarrollo multiplataforma para lograr un alto rendimiento en distintos entornos. Esta aplicación tiene un enfoque en el reconocimiento y la obtención de información detallada de las señales de tránsito, empleando sofisticadas técnicas de aprendizaje automático para poder identificar características de las señaléticas como colores y texturas. Su objetivo principal es servir como una herramienta que ofrece a los usuarios una experiencia interactiva, permitiendo visualizar datos e información relevante sobre la señal de tránsito que se enfoque con la cámara integrada de los dispositivos móviles, además esta aplicación podrá brindar información respecto de las normativas y señales actuales en el país, también nos permitirá descargar dicha normativa en nuestros dispositivos siendo una herramienta útil para cualquier tipo de usuario. Adicional con el objetivo de implementar, un desarrollo ágil y efectivo del proyecto, se optó por implementar como metodología de desarrollo a Scrum durante todo el ciclo de vida del software, este enfoque ágil proporcionará la flexibilidad necesaria para adaptar la aplicación a posibles cambios durante su desarrollo y garantiza una entrega continua de avances al proyecto.

Palabras clave: aplicación móvil, realidad aumentada, xamarin, reconocimiento de señales de tránsito.

Abstract

In the following project we propose to develop an augmented reality mobile application with the use of the Xamarin framework, this tool will allow us to take advantage of its cross-platform development capabilities to achieve high performance in different environments. This application has a focus on the recognition and obtaining detailed information of traffic signs, using sophisticated machine learning techniques to identify features of the signs such as colors and textures. Its main objective is to serve as a tool that offers users an interactive experience, allowing to visualize data and relevant information about the traffic sign that is focused with the integrated camera of the mobile devices, also this application can provide information regarding the current regulations and signs in the country, it will also allow us to download these regulations in our devices being a useful tool for any type of user. Additionally, with the objective of implementing an agile and effective development of the project, we chose to implement Scrum as a development methodology throughout the life cycle of the software, this agile approach will provide the necessary flexibility to adapt the application to possible changes during its development and ensures a continuous delivery of progress to the project.

Keywords: mobile application, augmented reality, xamarin, traffic sign recognition

Capítulo I

Introducción

Actualmente existe un rápido crecimiento urbano y, por consiguiente, un incremento en la movilidad vehicular lo que resulta en un entorno vial más complejo. Como medida de control se implementó el uso de símbolos de tránsito, a partir de ello surgió un desafío que es especialmente evidente, las decenas de mensajes diferentes que transmite cada señal de tránsito («Criteria for the Design and Evaluation of Traffic Sign Symbols», 1990). Por lo que interpretar de forma correcta cada señalética es crucial para garantizar la seguridad y eficiencia del tráfico.

La Realidad Aumentada (AR, por sus siglas en inglés, Augmented Reality) tiene como objetivo simplificar la vida del usuario, al proporcionar información virtual que enriquece su percepción y la interacción con el mundo real. La tecnología AR aumenta el sentido de la realidad al superponer objetos virtuales y señales sobre el mundo real en tiempo real (Carmigniani & Furht, 2011). Por su versatilidad, la Ar está siendo aprovechada en una amplia gama de campos, desde la educación hasta la medicina y la industria.

Uno de los campos donde se está dando grandes avances en la realidad aumentada es en el desarrollo de aplicaciones móviles esto debido a varios factores como, el extenso mercado de los dispositivos móviles y las capacidades técnicas avanzadas de estos aparatos, actualmente los dispositivos móviles están equipados con sensores, cámaras de alta resolución y poseen la suficiente potencia de procesamiento para utilizar tecnologías de realidad aumentada (Mota et al., 2018). Existen varios ejemplos de la implementación de AR en aplicaciones móviles, desde aplicaciones como Snapchat hasta Juegos móviles como Pokémon GO. Además, la Ar también se puede complementar con otras tecnologías como la visión por computadora, que consiste en el registro y reconocimiento de imágenes utilizando diferentes métodos de visión por computadora, estos métodos suelen constar de dos etapas: el seguimiento, donde en las imágenes de la cámara se detectan elementos

visuales u puntos de interés, luego está la reconstrucción o reconocimiento de dichos elementos (Carmigniani & Furht, 2011). Estas Tecnologías juntas abren una puerta a infinitos campos de aplicación permitiendo explorar nuevas posibilidades.

Con el objetivo de aprovechar estas innovaciones y brindar un apoyo en el reconocimiento de la extensa gama de señales de tránsito, surge la necesidad de desarrollar una aplicación móvil de realidad aumentada con el objetivo de reconocer y brindar información acerca de las señales de tránsito. Estas señales se detectan mediante el uso de la cámara y tecnología de procesamiento de imágenes digitales, Luego, la aplicación proporciona información sobre esa señal de tráfico en específico (Karunathilaka et al., 2015).

Propósito y contexto del proyecto

El propósito principal de este proyecto de trabajo de integración curricular es desarrollar una aplicación móvil de realidad aumentada para el reconocimiento y obtención de información de señales de tránsito utilizando el framework Xamarin, que sirva de apoyo en la comprensión y cumplimiento de las normas de tránsito. Esta aplicación busca proporcionar una experiencia interactiva y novedosa al interactuar con las señales de tránsito mediante el uso de tecnologías emergentes de realidad aumentada.

La seguridad vial es una preocupación en el Ecuador por lo que las señales de tránsito se utilizan para ayudar al movimiento seguro y ordenado del tránsito de peatones y vehículos (Gonzales, s. f.). Por lo que la correcta interpretación de las señales de tránsito es primordial para la seguridad de los conductores y peatones. La realidad aumentada ofrece la posibilidad de mejorar la comprensión e interacción con las señales de tránsito proporcionando información en tiempo real. Este proyecto se sitúa en el contexto de la movilidad urbana y la tecnología de realidad aumentada, con el objetivo de ayudar la interpretación efectiva de señales de tránsito. A través de la implementación de una

aplicación móvil utilizando Xamarin, se pretende explorar y demostrar la viabilidad de esta tecnología para mejorar la seguridad y la eficiencia en el ámbito vial.

Justificación de la investigación

La necesidad de realizar una aplicación móvil de realidad aumentada haciendo uso del framework Xamarin para el reconocimiento y obtención de información de señales de tránsito, se da como una respuesta ante la confusión en la interpretación del extenso catálogo de señaléticas de tránsito ubicadas en las vías, esto como medida de prevención de accidentes de tránsito fruto del constante desarrollo urbano e incremento del tráfico.

El desconocimiento de estas señales implica una alta probabilidad de sufrir accidentes de tránsito, esta aplicación ofrece ayudar a los usuarios a facilitar la interpretación de las señales de tránsito brindando información relevante de cada una de ellas, beneficiando al público que desconoce sobre la normativa y el significado que cada una de las señaléticas nos ofrece, asegurando así la educación y seguridad vial.

Objetivos Generales y específicos

Objetivo General

Desarrollar una aplicación móvil de realidad aumentada utilizando el framework Xamarin, enfocado en el reconocimiento y la obtención de información de señales de tránsito, contribuyendo así a la seguridad vial y proporcionando una experiencia de usuario innovadora y eficiente.

Objetivo específico

- Investigar las tecnologías más adecuadas en el ámbito de reconocimiento de imágenes y realidad aumentada para establecer una base robusta de información para el desarrollo de la aplicación.

- Desarrollar una aplicación móvil utilizando Xamarin, con un enfoque en la usabilidad de sus interfaces para garantizar accesibilidad y facilidad de uso para los usuarios
- Implementar un sistema de reconocimiento de señales de tránsito con un enfoque de realidad aumentada, con el objetivo de garantizar una detección precisa de las señaléticas viales
- Llevar a cabo pruebas en la aplicación para validar su desempeño y exactitud en condiciones reales de uso, además, realizar modificaciones conforme a lo requerido para asegurar su óptimo desempeño.

Metodología de la investigación

La metodología de investigación para el desarrollo de la aplicación móvil de reconocimiento de señales de tránsito con realidad aumentada se encuentra dividida en distintas etapas claves que abarcan desde la investigación preliminar hasta el desarrollo de la aplicación móvil. El objetivo principal es lograr el reconocimiento de señales de tránsito a través del uso de cámaras de dispositivos móviles. Para cumplir con el objetivo de identificar y clasificar señales de tránsito en imágenes que genera la cámara, se debe implementar el uso de algoritmos sofisticados de visión por computadora y técnicas de aprendizaje profundo.

Por ellos, un punto importante es la investigación en las distintas bases de datos científicos IEEE, Scopus que nos permitirá contar con información relevante y consolidada acerca de la implementación de las tecnologías antes mencionadas en nuestro tema de investigación. Luego se investigará y analizará el proceso del aprendizaje profundo y generación del DataSet (Conjunto de datos), por consiguiente, se generará el DataSet que contendrá varias imágenes de señales de tránsito ubicadas en diversas zonas como ciudades, carreteras, parques, zonas turísticas, barrios, etc. Este DataSet servirá como apoyo en el entrenamiento del modelo de aprendizaje profundo, posteriormente se evaluará la predicción del modelo a partir de enviar información de una señalética a la vez, con esto

se podrá verificar que la información devuelta por el modelo sea la correcta, en caso de tener predicciones incorrectas se realizará ajustes al modelo hasta obtener los resultados esperados.

Una vez que logremos que el modelo sea lo suficientemente robusto analizaremos e investigaremos técnicas para lograr la implementación a dispositivos móviles, para este proceso se utilizara el framework Xamarin, con el modelo de reconocimiento de señale de tránsito entrenado, para cumplir con el objetivo principal de nuestro proyecto.

Capítulo II

Marco Teórico

Realidad Aumentada

La Realidad Aumentada (AR) es una tecnología innovadora que amplía el mundo físico del usuario añadiendo elementos virtuales en el mundo real, Cualquier sistema de realidad aumentada debe respetar estas tres reglas: vista 3D, debe combinar lo real y lo virtual (Bouaziz et al., 2020). Para observar la superposición de objetos virtuales en el mundo real comúnmente se utilizan dispositivos multimedia la más comunes son las pantallas digitales.

Los sistemas AR para garantizar una superposición consistente de objetos, deben estimar, en tiempo real, la posición del objeto uno de los métodos más comunes para realizar la estimación es utilizar marcadores, mismos que serán identificados por cámaras, sin embargo, en algunos casos, el uso de marcadores no es factible, y es necesario aplicar otros métodos que se detallaran más adelante (de Souza Cardoso et al., 2020).

Como podemos darnos cuenta, AR Es una tecnología muy novedosa, y debido al desarrollo de computadoras más rápidas, cámaras avanzadas y algoritmos novedosos se ha convertido en un campo de investigación multidisciplinario popular en las últimas décadas lo que motiva los investigadores a ampliar las áreas de aplicación de la RA (Makhataeva & Varol, 2020).

Definición y Características

La realidad aumentada es una tecnología que permite la combinación del entorno físico y real con información del entorno virtual, esto con el fin de modificar la percepción física del usuario (*La realidad aumentada*, 2018).

Varios componentes están involucrados en el funcionamiento de la realidad aumentada, los cuales incluyen:

- Dispositivo con cámara que toma las imágenes reales que observan los usuarios.
- Dispositivo multimedia en el cual se proyecta la combinación de las imágenes reales con las virtuales.
- Dispositivo de procesamiento para interpretar y procesar la información.

La RA se puede clasificar según los dispositivos que se utilizan, y son tres categorías las más principales y utilizadas:

La primera, utilizar pantallas montadas en la cabeza (HMD), hardware portátil, como gafas o cascos. Estos dispositivos son categorizados en óptica transparente y video transparente. Segundo tenemos el uso de pantallas, como las incorporadas en teléfonos inteligentes y tabletas también monitores, para mostrar objetos virtuales. Como tercer tipo tenemos el uso de realidad aumentada espacial que consiste en un conjunto de proyectores para proyectar la información virtual sobre objetos reales (de Souza Cardoso et al., 2020).

De acuerdo a lo revisado anteriormente tenemos que las características principales de la RA son, la combinación de elementos del mundo real con elementos virtuales, La superposición de estos elementos en tiempo real, La superposición de los objetos virtuales se pueden observar a través de pantallas, monitores o con el uso de gafas.

Tipos de realidad aumentada

Existen diversos tipos de realidad aumentada, cada uno con características y contextos diferentes de uso e implementación, tenemos los siguientes enfoques de realidad aumentada:

- Realidad aumentada basada en el reconocimiento de patrones o marcas: utiliza marcadores o activadores físicos que el software reconoce y es cuando superpone algún modelo 3D. “El software de Realidad Aumentada realiza un seguimiento (tracking) del patrón o marcador, permitiendo que se ajuste la posición del

modelo 3D que aparece en la pantalla cuando le movemos o giramos” (Salas Alvarez et al, 2016) .

- Realidad aumentada basada en el reconocimiento de imágenes: utiliza imágenes del entorno como elementos activadores para colocar contenido virtual sobre ellas, lo importante y esencial es que las imágenes que se proporcionan deben ser suficientemente detalladas, de lo contrario la detección puede ser defectuosa (Salas Alvarez *et al*, 2016).
- Realidad aumentada basada en la geolocalización: Utiliza dispositivos móviles y software para mostrar puntos de interés (POIs) en ubicaciones específicas. Los usuarios pueden localizar POIs en sus dispositivos y acceder a información como descripciones, imágenes y videos (Salas Alvarez *et al*, 2016).
- Realidad aumentada basada en el reconocimiento de objetos: Permite usar objetos físicos como activadores, aumentando el realismo de los sistemas y aportando nuevas posibilidades (Salas Alvarez *et al*, 2016).

Señales de tránsito

Las señales de tráfico son los signos visuales colocados en las vías y que son utilizados para ofrecerle información a los conductores y peatones que transitan por un camino, estas señales se clasifican de acuerdo a diferentes características como forma, colore y símbolo (Cruz Sanchez, 2020). Por lo que cada uno tiene un significado y una normativa diferente, con el propósito de regular el tráfico y garantizar la seguridad vial tanto de peatones como de conductores.

Importancia y función en la vía pública

Las señales de tránsito surgieron en base a la necesidad de crear un sistema que pudiera guiar al usuario en su recorrido, la principal función de la señalización de las calles y carreteras es la de suministrar a los conductores información necesaria o útil en el momento y lugar que la precisen (Rivero et al., 2019). Enfocado al objetivo de garantizar la seguridad

vial, ayudando a prevenir accidentes, dirigir el flujo vehicular y alertar sobre alguna condición específica de la vía.

Clasificación de las señales de tránsito

La categorización de las señales de tránsito se encuentra definida conforme a lo establecido en la Normativa RTE INEN004, en base con la Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial de Ecuador. Según el reglamento, se identifican ocho tipos distintos de señales para su clasificación, incluyendo:

Señales Regulatorias (Código R), estas señales regulan la circulación vehicular y señalan la aplicación de requisitos legales.

Señales Preventivas (Código P), Advierten a los usuarios sobre condiciones inesperadas o peligrosas en las vías.

Señales de Información (Código I), Informan las direcciones, distancias, destinos, rutas y puntos de interés.

Señales Especiales Delineadoras (Código D), Informan a los usuarios sobre la aproximación de un lugar con cambios bruscos de altura, ancho y dirección de una vía.

Señales para Trabajos en la Vía y Propósitos Especiales (Código T), proporcionan información y orientan a los usuarios de las vías para transitar con seguridad en áreas de trabajo en carretera. También advierten sobre otras condiciones temporales y peligrosas.

Señales para Zonas Escolares (Código E), información de la aproximación a un centro educativo.

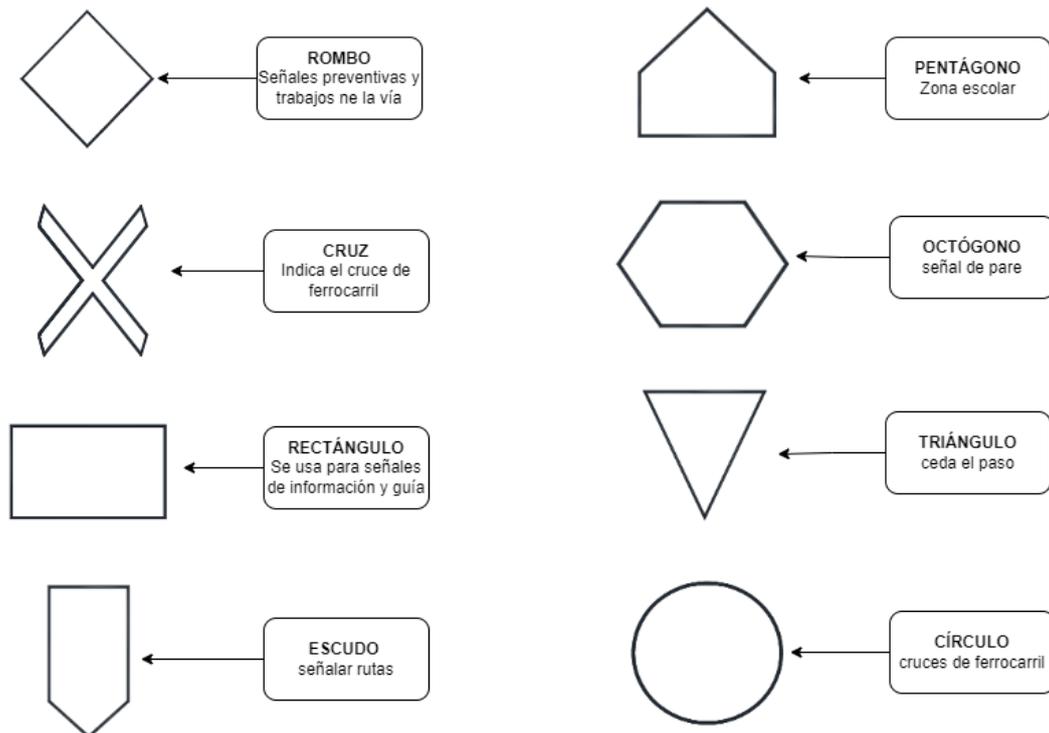
Señales Turísticas y de Servicios (Código ER), información sobre direcciones, sitios de interés, destinos turísticos y accesos a servicios públicos.

Señales y Dispositivos para Zonas de Riesgo (Código SGR), Información acerca de posibles amenazas por fenómenos de origen natural.

También se clasifican de acuerdo, a su forma esto con el propósito de facilitar la identificación de la señal por parte del conductor. A continuación, en la figura 1 se detalla las principales formas que diferencian a las señales de tráfico.

Figura 1

Formas de las señales de tránsito



Aplicaciones móviles y tecnologías asociadas

Evolución de las aplicaciones móviles

Las aplicaciones móviles transformaron la manera en que interactuamos con la tecnología, específicamente con los dispositivos móviles que año tras año adquieren mejores características técnicas como mayor poder de procesamiento, mejores cámaras, etc. Estas mejoras técnicas han impulsado la constante evolución de las aplicaciones móviles mejorando funcionalidades y experiencias de usuario.

Por los años 90, aplicaciones sencillas como agenda y calendario fueron las primeras en estar disponibles en dispositivos móviles haciendo uso de las capacidades técnicas de esa época. Por el año 2007 el lanzamiento del iPhone y un año después el lanzamiento de la app store permitieron la creación y comercialización de aplicaciones móviles.

La evolución de las aplicaciones móviles se ha visto influenciada por factores externos como la creación del Internet de alta velocidad, mejoras técnicas a nivel de hardware, computación en la nube y cámaras de mayor calidad que facilitó el surgimiento de aplicaciones de streaming, juegos en línea, aplicaciones de realidad aumentada y realidad virtual (*The History of Mobile Apps: A Comprehensive Examination*, s. f.).

Xamarin como framework de Desarrollo

Los desarrolladores móviles buscan que sus aplicaciones estén disponibles en todas las plataformas disponibles por ejemplo IOS y Android, pero dado que cada plataforma funciona de forma diferente a veces esta tarea resulta difícil para los desarrolladores (Prajapati et al., 2016). Debido a esta dificultad se crearon nuevas tecnologías de desarrollo multiplataforma que brinda grandes ventajas en comparación con la programación nativa para cada plataforma, como por ejemplo el hecho de escribir el código una vez y utilizarlo en varias plataformas simplificando y optimizando el desarrollo de aplicaciones móviles.

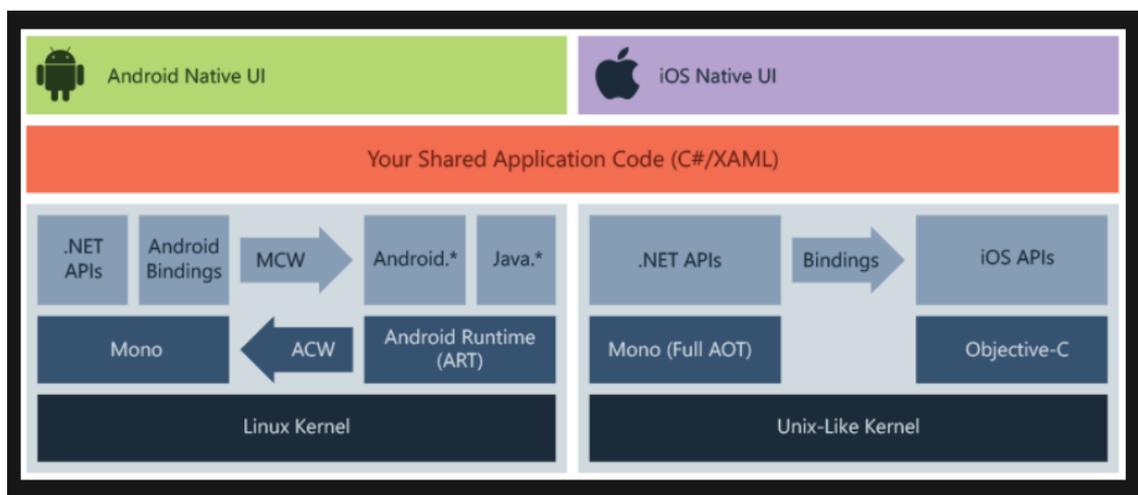
En esta ocasión nos centraremos en Xamarin, un Framework de desarrollo multiplataforma. Permite a los desarrolladores usar C# para escribir sus códigos, que luego se compilan para cada plataforma compatible, proporciona rendimiento nativo y la apariencia es similar a la de las aplicaciones nativas, en el año 2011 Microsoft adquirió Xamarin luego lo integro a su propio IDE, Visual Studio y lo convirtió en código abierto, este framework admite patrones de diseño como MVC y MVVM (Vishal & Kushwaha, 2018).

Xamarin es perfecto para desarrollar aplicaciones móviles multiplataforma en las que no desea muchas características nativas para todas y cada una de las plataformas. Con este Framework, los desarrolladores pueden crear diseños de interfaz de usuario nativos que se pueden compartir entre las aplicaciones de Android, iOS (Prajapati et al., 2016). Se asegura hasta un 80% de código compartido en aplicaciones que requieren menos funcionalidades específicas y nativas de cada plataforma.

Para el diseño de interfaz de usuario, se utiliza el lenguaje de marcado XAML. Una vez que diseñe el diseño, conecte todos los controles con su código de BackEnd compartido y obtenga aplicaciones totalmente nativas para Android, iOS y Windows Phone. Durante el tiempo de ejecución, cada control se asigna a elementos de interfaz de usuario nativos específicos de la plataforma, como Textbox en Windows, UITextView para iOS y EditText para Android. En la figura 2 se puede observar con mayor detalle cómo es la interacción de Xamarin con los sistemas operativos más populares.

Figura 2

Funcionamiento de Xamarin



En el desarrollo de aplicaciones móviles multiplataforma con el uso del framework Xamarin tenemos varios componentes que interactúan entre sí para lograr la programación multiplataforma. Primero la interfaz de Usuario Nativa las dos barras representan las interfaces de usuario nativas para Android e iOS. Esto significa que Xamarin permite a los

desarrolladores diseñar interfaces que se ajusten a cada plataforma, utilizando los elementos de UI nativos de cada una.

En medio, tenemos el código de aplicación compartido escrito en C# y XAML, donde los desarrolladores escriben la lógica de negocios y la interfaz de usuario que se comparte entre plataformas.

Después tenemos la Capa de Abstracción de cada Plataforma, por un lado, tenemos las aplicaciones Xamarin.Android, que se compilan desde C# en lenguaje intermedio (IL) que, después, se compila Just-in-Time (JIT) en un ensamblado nativo cuando se inicia la aplicación. Las aplicaciones Xamarin.Android se ejecutan en el entorno de ejecución Mono, en paralelo con la máquina virtual en tiempo de ejecución de Android (ART). Xamarin proporciona enlaces .NET a los espacios de nombres Android.* y Java.*. El entorno de ejecución Mono llama a estos espacios de nombres a través de contenedores administrados que se pueden llamar (MCW) y proporciona a ART contenedores que se pueden llamar de Android (ACW) , lo que permite que ambos entornos invoquen código entre sí (profexorgeek, 2023).

Por otro lado, tenemos a Las aplicaciones Xamarin.iOS se compilan Ahead-of-Time (AOT) de forma completa desde código de C# en código de ensamblado de ARM nativo. Xamarin usa selectores para exponer Objective-C en código de C# administrado y Registrars para exponer código de C# administrado en Objective-C. Los selectores y Registrars se denominan colectivamente "enlaces" y permiten la comunicación entre Objective-C y C# (profexorgeek, 2023).

Ventajas y desafíos de Xamarin en el desarrollo móvil

Una de las ventajas de Xamarin es que permite escribir y compilar aplicaciones 100% nativas para iOS, Android y Mac compartiendo el mismo código base escrito completamente en el lenguaje C#. Lo que ahorra tiempo y energía para los desarrolladores y sin comprometer los costos. Permite desarrollar rápidamente versiones beta que abarcan

todos los sistemas operativos móviles principales. También permite pruebas unitarias, de integración y de sistema gracias a su integración con el .NET framework, lo que es crucial dada la presión actual por lanzar productos rápidamente Avdic, D. (2019). React native vs xamarin-mobile for industry.

Los desafíos de Xamarin incluyen el tamaño grande de los archivos de instalación debido a la necesidad de incluir enlaces y referencias para todos los sistemas operativos móviles, lo que puede aumentar los tiempos de descarga e inicialización de la aplicación, También el desarrollo UI puede ser uno de los aspectos más consumidores de tiempo al usar Framework, por otro lado la integración con las últimas actualizaciones y características de los marcos de trabajo móviles de Google y Apple puede retrasarse, afectando a los desarrolladores que trabajan con Xamarin cuando los usuarios experimentan nuevas características con aplicaciones nativas que no están disponibles en aplicaciones desarrolladas con Xamarin (Avdic, D. (2019). React native vs xamarin-mobile for industry).

Capítulo III

Estado del arte

Aplicaciones existentes de Realidad Aumentada para Señales de Tránsito

La realidad aumentada (AR) es una tecnología que se ha implementado en varios campos, en esta sección nos enfocaremos en la revisión de estudios previos sobre AR implementada en reconocimiento de señales de tránsito, conducción y educación vial. A continuación, se presentan algunas de las contribuciones más notables en este campo, extraídas de la literatura revisada.

Algunos estudios resaltan que este sistema aprovecha la capacidad de procesamiento de los dispositivos móviles modernos para identificar señales de tránsito y proporcionar información relevante al usuario a través de una interfaz de RA, mejorando así la interacción y la comprensión de las señales de tránsito en entornos urbanos (Agudelo & Borda, 2012).

También se estudia la RA en una aplicación capaz de controlar el tráfico mediante la generación y superposición de elementos virtuales en imágenes reales del tráfico, permitiendo el control de situaciones de tráfico diversas en tiempo real como la invasión de vehículos en zonas prohibidas, el monitoreo de la velocidad, el control de estacionamiento, y la gestión de entradas y salidas de vehículos en áreas determinadas, todo desde cámaras estáticas. Se destaca el uso de tecnologías como OpenCV para visión artificial (Rodríguez Ventura, 2016).

Existe otro tipo de enfoque basado en RA con el objetivo de aumentar la seguridad en la conducción detectando y superponiendo señales de tráfico virtuales sobre la imagen real captada por la cámara del vehículo esto puede asistir eficazmente a los conductores, mejorando la interpretación de las señales y la respuesta a las condiciones del tráfico (Abdi et al., 2015)

Estas aplicaciones demuestran el potencial de la RA para mejorar la seguridad y eficiencia del tráfico vehicular, a través de la superposición de información útil sobre el mundo real, facilitando así una mejor interpretación de las señales de tránsito y la toma de decisiones en tiempo real por parte de los conductores, promoviendo la seguridad vial con el uso de nueva tecnología.

Aplicaciones desarrolladas con Xamarin en Realidad Aumentada

Luego de una ardua investigación no se logró recopilar información muy relevante acerca del uso de xamarin para el desarrollo de aplicaciones con realidad aumentada, solo existen libros que abarcan el tema pero con ejemplos sencillos, como me menciona en el libro “.NET Developer's Guide to Augmented Reality in iOS: Building Immersive Apps Using Xamarin, ARKit, and C#” (Englestone, 2021). Se detalla un ejemplo de cómo utilizar la herramienta de ARKit para el desarrollo de aplicaciones móviles de realidad aumentada, sin embargo, solo se abarca esta sección con ejemplos sencillos, y también se menciona que esta herramienta solo es compatible con el sistema operativo ios, siendo esta una gran desventaja ya que el objetivo de nuestro proyecto es implementarlo en multiplataforma para aprovechar esta característica del framework xamarin.

Por otro lado en el trabajo realizado por (Čarapina, s. f.), destaca las ventajas de xamarin en el desarrollo de interfaces ya que gran porcentaje del código se pudo escribir una vez y ser compilada tanto en los dispositivos Android como iOS, sin embargo que xamarin tiene acceso solo a funcionalidades básicas del dispositivo por lo que para acceder a funcionalidades más completas se debe programar por separado para cada sistema operativo ya que para implementar la realidad aumentada se requiere del uso de varios sensores y se debe tener acceso a funcionalidades más completas, esta situación es contraproducente para un desarrollador y es una gran punto en contra del framework Xamarin. .

Tecnologías Utilizadas en Aplicaciones de Realidad Aumentada con Xamarin

Para desarrollar e implementar la tecnología de realidad aumentada en xamarin, existen algunas alternativas de acuerdo al modo en el que vayamos a trabajar nuestro sistema, algunas de las herramientas que se pueden utilizar son las siguientes:

Wikitude: En el año 2012 la empresa lanzó su SDK para el desarrollo de aplicaciones AR que incluye funciones como el reconocimiento, seguimiento y geolocalización de objetos, es uno de los SDK más extensos y es compatible con varias plataformas como Android, ios, Windows, unity, Cordova, Xamarin, Titanium y React Native (Bañon Rubio, 2019). Sin embargo, estas ventajas lo convierten en el SDK más caro con un pago anual de 2490 euros, a pesar de tener una versión gratuita esta incluye una marca de agua y funcionalidades limitadas.

ARKit: En 2017, Apple presentó al mundo su marco de realidad aumentada ARKit y lo ha mejorado cada año, esta herramienta utiliza la cámara, el giroscopio y el acelerómetro de su dispositivo para determinar la dirección en la que apunta su dispositivo, existe la posibilidad de implementar ARKit con xamarin y es fantástico permitir a los desarrolladores de .NET crear aplicaciones iOS de realidad aumentada usando C#, aunque no hay mucha gente que lo haga todavía (Englestone, 2021). Es importante aclarar que con ARKit solo podremos crear aplicaciones de realidad aumentada solo para el sistema operativo ios.

ARCore: Es un SDK desarrollado por Google y fue presentado en marzo de 2018. Por tratarse de Google este SDK rápidamente ganó popularidad. Fue la respuesta de Google a su competencia anteriormente mencionada, ARCore tiene una gran comunidad de la que es posible aprender fácilmente. El anuncio anterior La ventaja y las buenas descripciones en la API ayudan al principiante (Bañon Rubio, 2019).

Xamarin.Essentials: A pesar de no ser un SDK específicamente para el desarrollo de Realidad aumentada, Xamarin.Essentials es una librería que nos proporciona una API multiplataforma para acceder a características nativas del dispositivo como, sistemas de

archivos, cámara, sensores (Jacome Leal, 2023). Esta podría ser una herramienta a tomar en cuenta en el desarrollo de nuestro proyecto.

Identificación de Brechas y Oportunidades en Soluciones Actuales

Las aplicaciones móviles de realidad aumentada para el reconocimiento de señales de tránsito es una tecnología básicamente nuevas, pero se posiciona como una herramienta eficaz para mejorar la seguridad vial, aunque hay varios desafíos en lograr implementar este tipo de aplicaciones debido a varios factores, como por ejemplo la amplia gama de señales de tránsito que existe actualmente complica que los modelos de inteligencia artificial sea capaz de detectar cada una de las señales sin cometer falsos positivos. Además este tipo de aplicaciones de reconocimiento de imágenes requieren de grandes volúmenes de datos para ser entrenado, por lo que con la tecnología actual hace casi inviable implementar todo en una aplicación por lo que obligatoriamente estas aplicaciones requieren el acceso al internet permanente para funcionar correctamente.

Otro inconveniente es que uno de los principales factores de accidentes de tránsito es el uso de los teléfonos móviles mientras se conduce, por lo que es difícil que un conductor utilice la aplicación de reconocimiento de señales mientras conduce, además de ello el desarrollo y mantenimiento de este tipo de aplicaciones es alto y aun que es una buena alternativa utilizar la realidad aumentada para reconocer las señales de tránsito aún no existe un modelo de negocio que haga a estas aplicaciones autosostenibles. Además las entidades de control vial actualmente no toman este tipo de tecnología como una alternativa para mejorar la seguridad vial.

Sin embargo, a futuro existen varias posibilidades para mejorar la seguridad vial mediante la realidad aumentada ya que los vehículos de transporte inteligentes equipados con inteligencia, incluidos los vehículos sin conductor, requieren detección y reconocimiento de señales de tránsito para evitar accidentes (Alam & Jaffery, 2020). Por lo que es posible que en años posteriores la tecnología de identificación de señales de tránsito baya

mejorando y tome otro rumbo enfocado en mejorar la seguridad vial y en la sofisticación de los autos autónomos que años tras año van ampliando su mercado mundial.

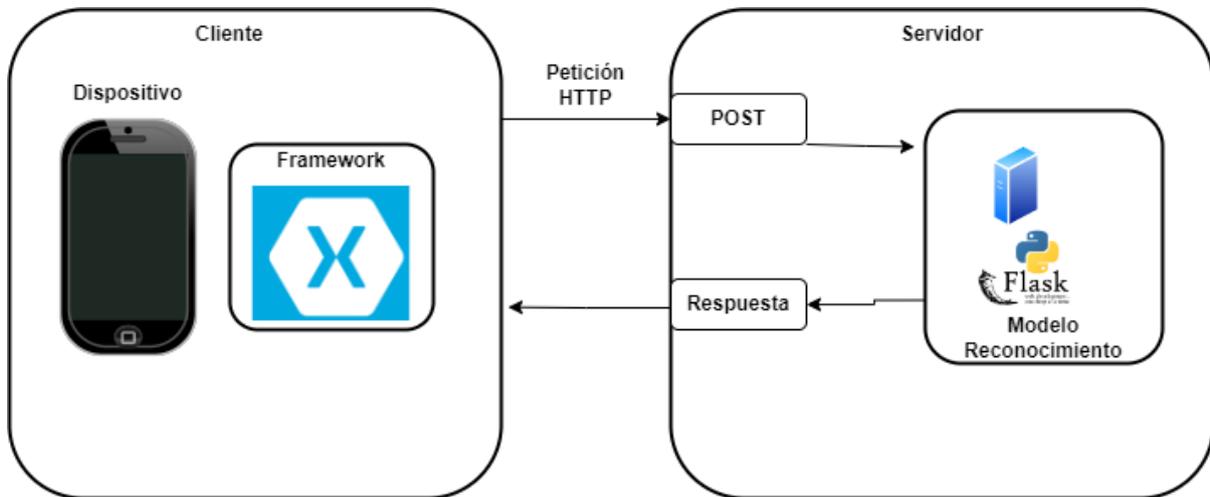
Capítulo IV

Diseño de la aplicación

En esta sección se especifica el proceso de desarrollo de la aplicación móvil con un enfoque de realidad aumentada, utilizando técnicas de visión por computadora y algoritmos de aprendizaje profundo, detallando la arquitectura que se emplea en la aplicación, el flujo de la aplicación, interfaz de usuario, etc.

Para que nuestra aplicación realice el reconocimiento de las señales de tránsito, la entrada principal deberán ser imágenes, que se obtienen a través de la cámara del dispositivo. En este contexto la aplicación deberá ser capaz de reconocer las señaléticas que se encuentran en todo el territorio nacional, estas podrán ser procesadas por un modelo de inteligencia artificial de reconocimiento de imágenes , y se mostrara la información adecuada de cada señal detectada. Cumpliendo así con el propósito de brindar la información a los usuarios de la aplicación.

De forma global el sistema seguirá una arquitectura cliente – servidor, donde el dispositivo móvil, con la aplicación desarrollada en xamarin serán el cliente, la misma que se comunicara con un servidor externo en el que estará alojado nuestro modelo, el servidor va ser capaz de manejar las solicitudes generadas por el cliente, actividad que es fundamental para que las señales de tránsito puedan ser reconocidas, esta arquitectura se detalla de mejor manera en la figura 3.

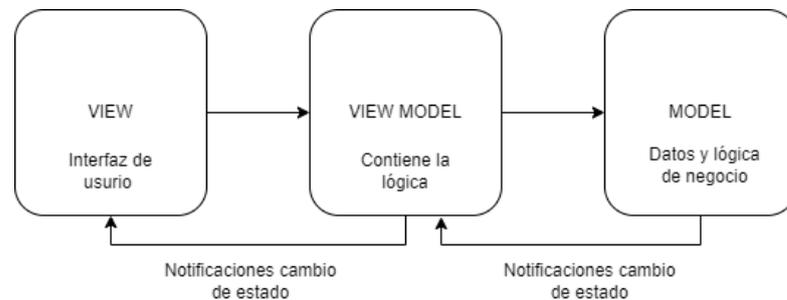
Figura 3*Arquitectura cliente - servidor***Arquitectura General del sistema**

La arquitectura general de la aplicación móvil presenta el marco organizativo que detalla la funcionalidad del sistema. Como se mencionó anteriormente nuestro sistema se basa en un modelo cliente-servidor, diseñado para optimizar el procesamiento de imágenes y la gestión de datos. En el núcleo se encuentra nuestra aplicación móvil, dentro de la cual se empleó la arquitectura Modelo-Vista-VistaModelo, que nos permite separar la lógica de la interfaz de usuario (UI) de la lógica de negocios y el manejo de datos.

Esta separación de responsabilidades facilita el desarrollo, el mantenimiento y la prueba de la aplicación, además de permitir una experiencia de usuario más fluida y reactiva. En la figura 4 se muestra gráficamente la arquitectura MVVM, con las responsabilidades de cada componente.

Figura 4

Arquitectura mvvm



El cliente en nuestra arquitectura es una aplicación móvil desarrollada sobre el framework Xamarin. La aplicación se encarga de capturar fotos utilizando la cámara del dispositivo. A continuación, envía las imágenes al servidor, que las procesa y envía los resultados al cliente. Además, la aplicación móvil está conectada a una base de datos que se utiliza para recuperar información de cada panel y mostrarla en la interfaz de usuario.

Por otro lado, el servidor, junto con el modelo pre-entrenado, realiza la función de procesar las imágenes captadas por la aplicación móvil. Las imágenes son procesadas mediante un modelo de inteligencia artificial capaz de identificar y clasificar las señales de tráfico. Cuando el modelo ha terminado de analizar la imagen, el servidor se encarga de enviar el resultado de la predicción a la aplicación móvil, estructurándose la respuesta en formato JSON para su correcto procesamiento por parte de la aplicación. Al separar las responsabilidades de captura y procesamiento de imágenes, conseguimos una aplicación móvil ligera a la vez que aprovechamos la enorme capacidad de procesamiento de los servidores modernos.

Flujo de la aplicación

A continuación, se detalla la secuencia y el proceso a través del cual la información fluye dentro de la aplicación, así como la forma en que el usuario interactuará con la misma. Este recorrido explica cómo las entradas del usuario se traducen en acciones concretas y

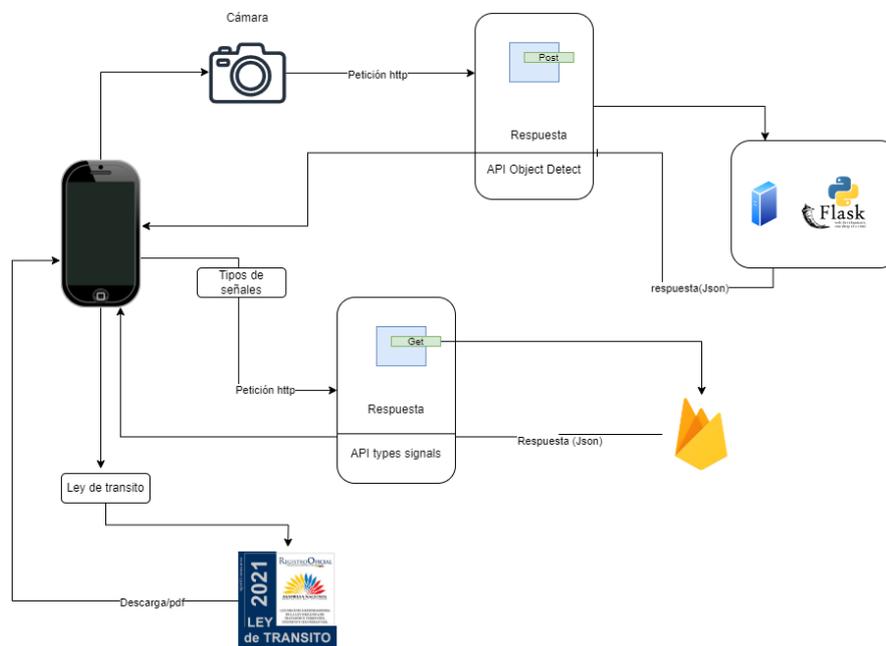
cómo la aplicación responde a estas interacciones para ofrecer una experiencia fluida y coherente

El flujo de la aplicación empieza cuando el usuario accede a la aplicación desarrollada en el framework xamarin, luego tiene acceso a tres distintas opciones de uso que son captura, Tipos de señales, ley de tránsito.

La opción de captura se encarga del reconocimiento de señales de tránsito, este se apoya del módulo de cámara a la cual se tiene acceso a través la librería xamarin.essential, desventajosamente esta librería solo nos da acceso a funcionalidad básica de la cámara nativa lo que complica implementar el enfoque de realidad aumentada, después el usuario puede tomar una foto y enviar mediante solicitud HTTP a un servidor implementado con Python y Flask. En este servidor se encuentra alojado un modelo de inteligencia artificial para identificar señales de tránsito. El mismo que retorna una respuesta en formato JSON que es procesada y tratada para mostrarla al usuario.

Por otro lado, la opción tipos de señales nos muestra información actualizada de todas las señales de tránsito vigentes a la actualidad, de la misma forma en esta opción la aplicación se comunica con un servidor mediante solicitudes HTTP, aun servidor de Firebase que contiene una base de datos que almacena la información de cada una de las señales de tránsito basado en los manuales gubernamentales disponibles.

Finalmente, la última opción de ley de tránsito permite acceder al documento oficial vigente, con el objetivo de conocer más detalles acerca de las normativas actuales. La figura 5 muestra gráficamente el flujo completo de nuestra aplicación.

Figura 5*Flujo del sistema*

Nota. Se representa el flujo de la aplicación móvil con todas sus funcionalidades disponibles y se detalla las tecnologías implicadas.

Interfaz de usuario y experiencia de usuario

La generación de las interfaces de usuario (UI) y la experiencia de usuario (UX) de nuestra aplicación se centra en asegurar tres pilares fundamentales de experiencia de usuario como lo es la accesibilidad, usabilidad y la eficacia. Con el propósito de brindar una experiencia de usuario fluida e intuitiva con la aplicación. A continuación, se detallará el funcionamiento en conjunto de las interfaces del sistema.

La pantalla principal ha sido diseñada de forma minimalista y con un diseño amigable donde destacan información respecto de la funcionalidad de la aplicación con un logo llamativo junto con un botón “Iniciar” ubicado en una zona de fácil acceso para el usuario que anima al usuario a utilizar la aplicación.

Figura 6

Pantalla de inicio app



Luego de iniciar, tenemos la pantalla de menú de opciones, siguiendo el mismo patrón de la pantalla principal contamos con un diseño minimalista y limpia, y cada opción cuenta con su respectivo botón y un icono que trata de comunicar la funcionalidad de esa opción, cada botón esta nombrado de forma que el usuario sepa a qué funcionalidad va a acceder, al elegir una opción accederemos a más funcionalidades. La figura 7 nos muestra la interfaz con las opciones disponibles en la aplicación.

Figura 7

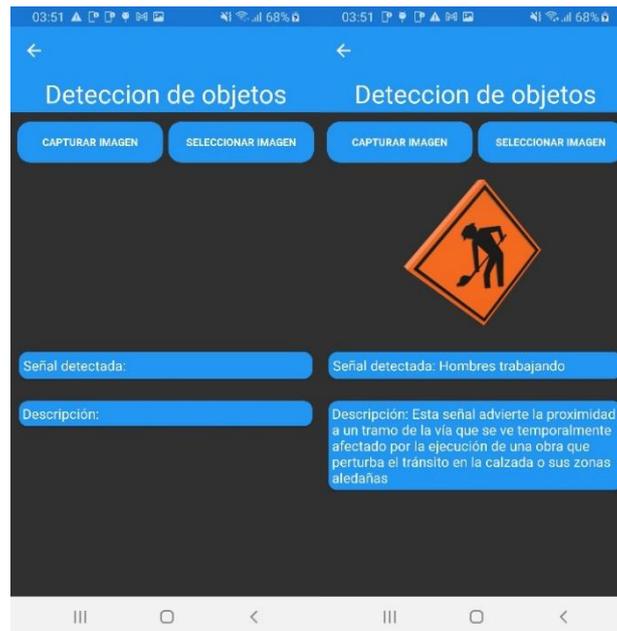
Pantalla de opciones



La primera opción “Captura”, muestra dos botones que nos permiten acceder a la cámara y al almacenamiento del dispositivo ya sea para capturar una foto o elegir una imagen de nuestra galería así poder enviar la imagen de las señaléticas al servidor.

Figura 8

Funcionalidad de la opción captura



La segunda opción “Tipos de señales”, muestra de forma ordenada cada tipo de señal de tránsito, con etiquetas de textos detallando las categorías y unas pilas de imágenes, con el propósito de no sobre cargar de información esta pantalla. Cada señal cuenta con la opción de ser seleccionada, con el propósito de brindar más información. La siguiente pantalla muestra de forma detallada la información de la señal de tránsito seleccionada, tratando de mostrar toda la información posible sin sobre cargar la pantalla.

Figura 9

Funcionalidad opción tipos de señales



La tercera opción “Ley de tránsito”, sigue el mismo patrón minimalista y limpio con un texto una imagen claros que muestran el archivo que contiene, con un botón claro y llamativo que motiva al usuario a descargar el archivo que esta pantalla contiene, que en este caso es la ley de tránsito.

Figura 10

Funcionalidad opción ley de tránsito



Como se puede observar en las imágenes, el diseño de la interfaz de usuario se a adaptado para ser lo más sencilla y limpia posible y también tiene un enfoque en ser una

aplicación con baja complejidad de uso para que sea fácil de usar y navegar, es decir reduciendo el número de pasos para acceder a cada funcionalidad de la aplicación. Se ha prestado atención en los colores implementado unos colores solidos que transmiten seriedad.

Proceso de Reconocimiento de Señales de Tránsito en Realidad Aumentada

La funcionalidad principal de la aplicación es el reconocimiento de las señales de tránsito la cual es una característica que brinda una experiencia innovadora al usuario, esta sección ofrece una descripción detallada del proceso mediante el cual se lleva a cabo la identificación y visualización de las señales de tráfico:

Captura y Procesamiento de Imágenes: El proceso se inicia cuando el usuario utiliza la funcionalidad de "Captura" en la aplicación para tomar una fotografía de la escena de tráfico utilizando la cámara integrada del dispositivo móvil. La imagen capturada se procesa utilizando el módulo de cámara, para tener acceso a la cámara se utilizó la librería Xamarin.Essentials, esta librería nos permite acceder a ciertas funcionalidades básica de la cámara nativa del dispositivo.

Transmisión de Imágenes al Servidor: Una vez tomadas las fotos, se envían mediante peticiones HTTP a un servidor local desarrollado con Python y Flask. Este servidor controla la comunicación entre la aplicación móvil y el modelo pre-entrenado utilizado para reconocer las señales de tráfico.

Reconocimiento y Análisis de Señales de Tránsito: Las imágenes recibidas por el servidor son procesadas por un modelo previamente entrenado para identificar y clasificar las señales de tráfico. El modelo utiliza algoritmos de visión por computadora para detectar patrones y características en cada imagen con el fin de identificar las señales de tráfico en la escena.

Retorno de Resultados al Dispositivo Móvil: Una vez completado el proceso de reconocimiento, el servidor devuelve una respuesta en formato JSON que contiene información sobre la señal de tráfico identificada en la imagen. Esta respuesta es procesada por la aplicación móvil y finalmente la información se visualiza en la pantalla de captura.

Capítulo V

Desarrollo e Implementación

En este capítulo se va a detallar todos los pasos a seguir para lograr desarrollar eficientemente y en función de tiempos establecidos el aplicativo móvil de identificación de señales de tránsito basado en el uso de un modelo pre-entrenado de reconocimiento de señale de tránsito.

Anteriormente ya se especificó el flujo y la arquitectura de la aplicación, pero nada de ello tendría sentido si no implementamos una metodología de trabajo que nos permita organizar el flujo de nuestro desarrollo del proyecto, es por este motivo que se implementó el uso de metodologías ágiles las cuales ofrecen un desarrollo más ligero y rápido que han alcanzado un alto nivel de popularidad y éxito en la industria del software más concretamente nos centramos en Scrum, un marco iterativo e incremental que asegura entregas continuas durante los sprint planificados (Scharff & Verma, 2010).

El objetivo de cada sprint es crear una implementación que equivalga a una historia de usuario mismo que se le asigna a un desarrollador, al final del sprint, el equipo revisa el trabajo y el progreso para poder discutir posibles mejoras. Después de eso, el proceso se convierte en un ciclo repetitivo (Bytyqi, 2022). Antes de plantear los sprint es importante conformar los equipos scrum lo cuales deben estar formados por el scrum master, Product Owner, el propietario del producto y el equipo de desarrollo, la existencia de scrum masters busca construir equipos autoorganizados, estos equipos deben contar con las habilidades y cualidades necesarias para conllevar el desarrollo de los proyectos(Ayunda & Budiardjo, 2020). Cada uno de los roles cumple distintas funciones en el equipo a continuación la tabla 1, detalla las funciones y responsabilidades de cada rol y que se aplicaran en nuestro proyecto.

Tabla 1*Roles en la metodología SCRUM*

Rol	Descripción
Scrum Master	El rol del Scrum Master es responsable de asegurar que el equipo tenga un sólido entendimiento y aplicación de Scrum, además de la generación de eventos y artefactos propios de Scrum.
Product Owner	El Product Owner es responsable de definir qué es o no parte del producto a entregar.
Development Team	El equipo está compuesto por profesionales que realizan el trabajo de desarrollar y entregar incrementos del producto software.

Nota. Roles de scrum. Recuperados de (Garcia et al., 2020)

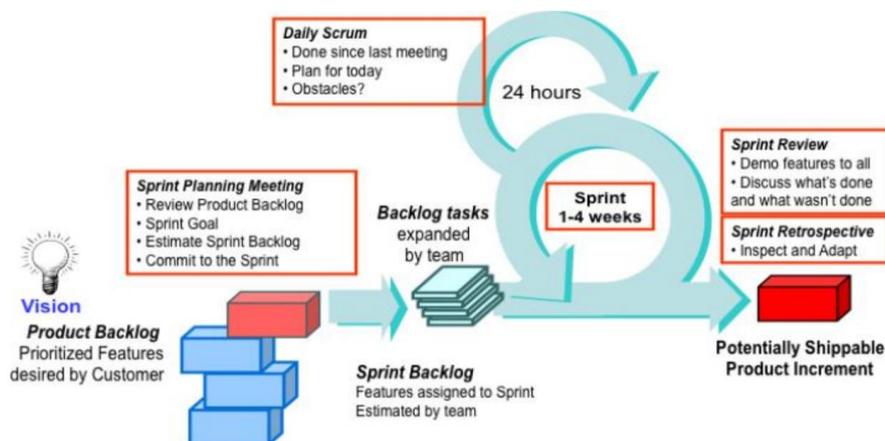
Luego, se procede a la definición del producto backlog, que es una lista de características y prioridades del producto en esta sección se incluye a los clientes o usuarios finales del proyecto (Santos et al., 2015). El trabajo continúa con la planificación del sprint, donde el equipo elige tareas del producto backlog y crea una lista de tareas estimando esfuerzos y tiempos en que deben culminarse, esto da forma al sprint backlog. De esta forma empieza la ejecución del sprint que cuenta con una duración de entre 1 a 4 semanas, en este momento el equipo está concentrado en alcanzar la meta del sprint, por lo que el equipo de desarrollo está protegido de perturbaciones o modificaciones al sprint backlog pero este puede aceptar cualquier cambio para un próximo sprint (Hema et al., 2020).

Es importante mencionar que durante la ejecución del sprint el scrum master debe trabajar en colaboración con el equipo de desarrollo generando reuniones diarias que deben tener una duración de sólo 15 minutos para sincronizar el trabajo, la revisión del sprint es una reunión que involucra la participación entre el equipo Scrum y las partes interesadas después del sprint, el propósito de la revisión del sprint es realizar inspecciones y adaptación de cada incremento para que podamos optimizar las necesidades para el próximo sprint, también el equipo scrum reflexiona sobre lo que ha hecho bien, lo que salió mal y lo que se podría mejorar, así podremos introducir mejoras procesables para el próximo sprint (Ayunda & Budiardjo, 2020).

En la figura 11, se detalla gráficamente el ciclo de vida de scrum durante el desarrollo de proyectos software, lo que nos permite tener una idea más clara de cómo vamos a organizar nuestro equipo para lograr finalizar con esto nuestro desarrollo.

Figura 11

Ciclo de vida de Scrum



Nota. Recuperado de (Hema et al., 2020)

Una vez que se definió la metodología de desarrollo y se especificó los roles y las responsabilidades de cada uno. Dentro de nuestro proyecto enfocado al desarrollo de la aplicación móvil de reconocimiento de señaléticas de tránsito se asigna los roles en base a las habilidades técnicas de cada miembro. En la tabla 2 se detalla los roles de cada uno de

los miembros que abarca este proyecto, los cuales han sido elegidos con el propósito de brindar el aporte necesario en la ejecución del proyecto.

Roles de la metodología Scrum

Tabla 2

Equipos de desarrollo Scrum

N.º	Rol Scrum	Integrantes	Funciones
1	Product Owner	Ing. Javier Montaluisa	Definir los requisitos del sistema Establecer los tiempos de entrega para su evaluación
2	Scrum Master	Kevin Omar Gualpa Yanchatuña	Líder del equipo Desarrollo y cumplimiento de las entregas de producto en el tiempo establecido
3	Equipo de desarrollo	Kevin Omar Gualpa Yanchatuña Unda Reinoso Anthony Josue Chuquitarco Copara Angie Maricela	Implementación y desarrollo de la aplicación móvil con enfoque de realidad aumentada

Nota. Roles de cada miembro en base a Scrum

El proyecto cuenta con un equipo de desarrollo conformado únicamente por una persona que será el encargado de cumplir las tareas enfocadas a la programación del proyecto de aplicación móvil, adicional a esto se asignó roles de desarrolladores al equipo involucrado en el desarrollo de la evaluación y comparativa de los frameworks en lo que se desarrolla la aplicación con un enfoque de realidad aumentada como se establece esto para que de cierta forma tengan el conocimiento de cómo van los avances del proyecto. Luego

de a ver asignado los roles de cada miembro involucrado, procedemos a generar los requisitos específicos del sistema que son de vital importancia para entregar un producto robusto y de acuerdo a las necesidades del usuario final, todas las historias de usuario necesarias para que nuestro sistema funcione correctamente, se resumen de manera detallado en la Tabla 3.

Planteamiento de las historias de usuario

Tabla 3

Historia de usuario 01

ID	Nombre	Rol	Funcionalidad	Resultado
1	HU.1	Como usuario	Como usuario necesito un conjunto de datos (Dataset) que contenga imágenes de señalización de tráfico conforme al manual de la Agencia Nacional de Tránsito (ANT)	Para entrenar el modelo de detección de señaléticas de tránsito
2	HU.2	Como usuario	Como usuario Deseo emplear la cámara de mi dispositivo móvil para la detección de señales de tráfico mediante una experiencia de realidad aumentada.	Para facilitar la adecuada interpretación de las señales de tráfico.
3	HU.3	Como usuario	Como usuario quiero visualizar de manera detallada la información de las señales de tránsito vigentes en el manual de tránsito.	Para reforzar los conocimientos sobre señalización vial.
4	HU.4	Como usuario	Como usuario deseo descargar el manual de señalización vial	Para tener conocimiento respecto de las normativas vigentes.

ID	Nombre	Rol	Funcionalidad	Resultado
5	HU.5	Como administrador	Como administrador quiero ingresar, editar y eliminar la información de las señaléticas de tránsito	Para administrar la información que se muestra al usuario.

Nota. Cada historia de usuario, está enfocada en especificar lo que el usuario espera y quiere que contenga la aplicación móvil.

Producto backlog del proyecto

Una vez que hemos definido y priorizado las historias de usuario que conformarán nuestro proyecto, el siguiente paso es la planificación del Product Backlog, El cual nos va ayudar a gestionar todas las historias de usuario, asignándoles plazos específicos que incluyen fechas de inicio y finalización. Esta planificación detallada nos permite ejecutar un flujo de trabajo ordenado y controlado a lo largo de los diferentes Sprint que tendrá nuestro proyecto.

Cada sprint, representa un periodo de tiempo específico durante el cual se desarrollarán y completarán un conjunto específico de tareas que se adaptan a cada historia de usuario y en función de las necesidades del proyecto y de los objetivos establecidos, en nuestro caso, hemos planificado tres sprints que se alinearán con los requisitos y las expectativas del proyecto. Estos sprints se detallan en la Tabla 4 , lo que nos proporciona una visión clara y estructurada del proceso de desarrollo y nos ayuda a mantenernos enfocados en el cumplimiento de nuestros objetivos a lo largo del tiempo.

Tabla 4

Product Backlog 1

Historia de usuario	Nombre	Estimación (días)	Fecha inicio	Fecha fin	N° de Sprint
1	HU.1	20	06/11/2023	01/12/2023	1

Historia de usuario	Nombre	Estimación (días)	Fecha inicio	Fecha fin	N° de Sprint
2	HU.2	20	04/12/2023	02/01/2024	1
3	HU.3	20	03/01/2024	31/01/2024	2
4	HU.4	10	01/02/2024	15/02/2024	2
5	HU.5	5	16/02/2024	22/02/2024	3

Nota. Asignación de fechas para cada historia de usuario

Para la implementación de la aplicación móvil, las pruebas y la recopilación de datos, se elige estratégicamente cualquier ubicación en la ciudad de Latacunga, donde se pueda acceder a señalizaciones de tránsito. El objetivo principal es dirigirse a cualquiera de estas señales ubicadas tanto en las calles, parques, sitios turísticos, etc. Y utilizar la cámara del teléfono móvil para obtener información sobre la señal específica que se esté enfocando, además para tener un buen funcionamiento, en este lugar se coloca un servidor que se encarga de procesar las imágenes enviadas desde la aplicación móvil, es importante contar con una red wifi para permitir la comunicación entre la aplicación móvil y el servidor ya que este proceso solo se puede realizar con el uso de internet, este entorno es el más apropiado para realizar pruebas y nos asegurará que la aplicación móvil funcione de manera óptima.

Selección de la plataforma de desarrollo

Luego de detallar la planificación del desarrollo de la aplica, en esta sección vamos a detallar las herramientas que se utilizaron para el desarrollo de este proyecto. A continuación, en la tabla 5 se proporciona una descripción más detallada de las herramientas elegidas.

Tabla 5*Herramientas y plataformas de desarrollo*

Herramienta	Descripción
Entorno de desarrollo integrado (IDE)	Para la generación de código se utiliza Visual Studio 2022, Xamarin como framework de desarrollo móvil multiplataforma, c# como lenguaje de programación y xaml como lenguaje de marcado para las interfaces.
Procesamiento de imágenes	Para el procesamiento de imágenes se utiliza OpenCv
Base de datos en la Nube	Para la base de datos en la nube se utiliza Firebase
Control de versiones	Para gestionar el código de manera efectiva, hacemos uso de herramientas de control de versiones en este caso Git, junto con plataforma de repositorios "GitHub".

Nota. Descripción detallada de cada tecnología elegida

Con el uso de estas herramientas se espera alcanzar una base sólida para el desarrollo y la implementación exitosa de nuestra aplicación móvil, permitiéndonos maximizar la productividad, la calidad y la colaboración de equipos durante todo el proceso.

Programación de la funcionalidad de reconocimiento de señales de tránsito

En esta sección examinaremos el método Scrum como guía básica para la gestión ágil de proyectos. Una vez creada y priorizada la cartera de pedidos, el siguiente paso es planificar los Sprint individuales. Hay que tener en cuenta varios factores, como la complejidad de las tareas, los recursos disponibles y las necesidades del cliente, para asignar correctamente las tareas en cada sprint.

Según la metodología Scrum, es importante mantener una comunicación constante entre todos los equipos de un proyecto a través de reuniones presenciales o reuniones

virtuales. Para facilitar la comunicación, puedes utilizar diversas herramientas tecnológicas como Google Meet o Zoom, que permiten organizar reuniones sin necesidad de encontrarse cara a cara, agilizando así el proceso de comunicación. Sin embargo, si es necesario, puede utilizar las instalaciones de la Universidad de las Fuerzas Armadas Sede Latacunga para organizar reuniones especiales de retroalimentación y discusión entre los miembros del equipo, en las que todos los miembros del equipo tienen la oportunidad de participar en persona.

Sprint 01: Generación de conjunto de datos (DataSet) de las señaléticas de tránsito

Basándonos en la planificación previamente establecida, se han tenido en cuenta para el sprint 01 las historias de usuario 01 y la historia de usuario 02, detalladas en la Tabla 3. El propósito principal en este primer sprint, es la captura de fotografías de las señales de tránsito y su clasificación, las cuales sirven para generar un DataSet o conjunto de datos que a su vez sirven como datos de entrenamiento para el modelo de inteligencia artificial. Este proceso es indispensable para garantizar la eficacia y precisión del modelo de predicción y alcanzar los resultados esperados por el usuario.

H.U.01: Generación del DataSet de las señales de tránsito

En la historia de usuario H.U.01 se detalla la necesidad de crear un conjunto de datos que abarque imágenes de las señales de tránsito conforme al manual establecido, por lo que es importante tratar de cubrir la mayor cantidad de señales posibles. En la tabla 5 se presenta de forma más detallada la historia de usuario en cuestión, donde se establecen los requisitos específicos que deben cumplirse para garantizar la correcta implementación del conjunto de datos y la aceptación del cliente. Esta planificación facilita la asignación de tareas y recursos necesarios, asegurando así que se cumplan los objetivos del proyecto de manera efectiva y eficiente.

Tabla 6*Historia de usuario 01 detallada*

Historia de Usuario	
Numero: H.U.01	Usuario: Usuario
Nombre de la historia de usuario: Generación del DataSet de las señales de tránsito	Numero de Sprint: 01
Prioridad: Alta	Riesgo de desarrollo: Baja
Días estimados: 20	Interacción asignada: 1
Desarrolladores responsables: Kevin Gualpa, Angie Chuquitarco, Anthony Unda	
Descripción: Como usuario necesito un conjunto de datos (Dataset) que contenga imágenes de señalización de tráfico conforme al manual de la Agencia Nacional de Tránsito (ANT)	
Validación: Realizar la captura de imágenes de las señales de tránsito presentes tanto en las vías, parques, zonas turísticas, etc. Fotografiar cada señal de tránsito en varios ángulos así también en distintas distancias. Clasificar cada grupo de señales de tránsito en distintas carpetas con su respectivo nombre de ser el caso.	

Nota. Detalles de historia de usuario 1 referente a generación de DataSet

Todas las tareas del Sprint 01, nombrado como Sprint backlog 01, recopila todas las actividades que se van a ejecutar durante este período, incluyendo detalles como las personas responsables, las fechas programadas y la estimación de horas dedicadas a cada tarea. Es importante destacar que, en el momento actual, todas las actividades del sprint han sido completadas.

Tabla 7*Sprint Backlog 1*

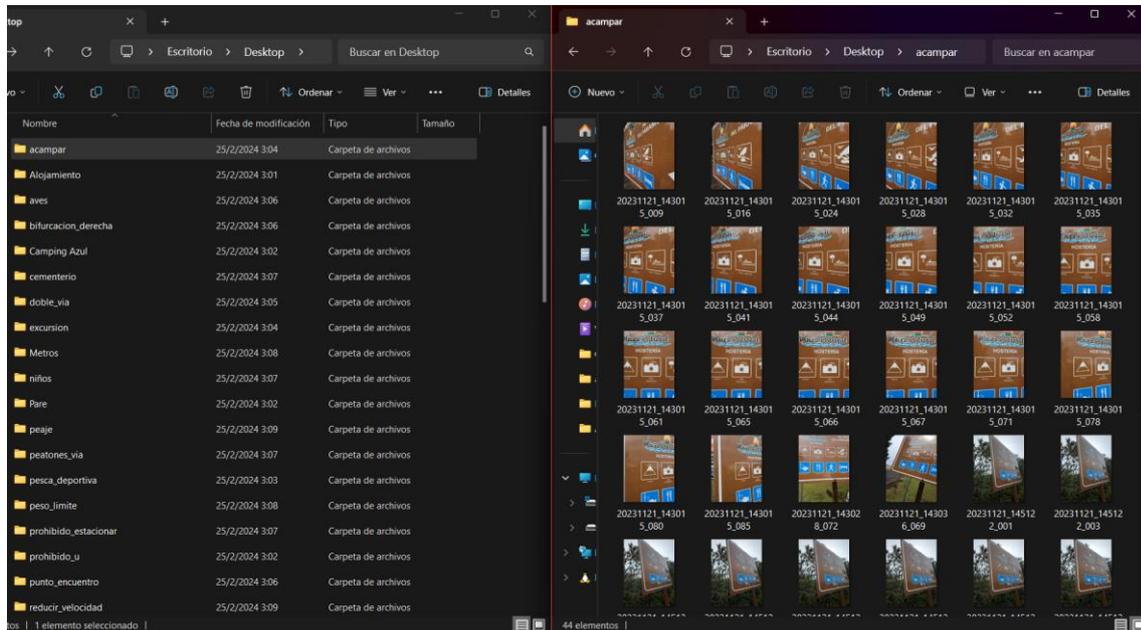
Sprint 1						
HU ID	Tareas	Horas	Inicio	Fin	Responsable	Estado
HU01	Capturar varias fotografías en varios ángulos y distancias de las señaléticas de tránsito, así como en varias locaciones.	24	06/11/2023	12/11/2023	Kevin Gualpa Anthony Unda	Finalizado
	Realizar la clasificación de las imágenes por su categoría de acuerdo al manual establecido	36	13/11/2023	01/12/2023	Klever Moreno Angie Chuquitarco	Finalizado

Nota. Sprint Backlog con sus tareas detalladas

Resultado del cumplimiento de la historia de usuario 01, se logró crear un dataset que consta de 130 carpetas, cada una conteniendo alrededor de 50 imágenes correspondientes a una señalética específica. Estas imágenes, capturadas desde diversos ángulos y distancias, fueron obtenidas en diferentes escenarios para garantizar una adecuada capacitación del modelo. La figura 12 muestra la clasificación detallada de las carpetas e imágenes generadas en este proceso.

Figura 12

Clasificación de las señaléticas



Nota. En la ilustración se muestra las carpetas de imágenes y su contenido clasificado

H.U.02: Desarrollo del algoritmo de detección de señaléticas de tránsito.

Dentro del contexto de la Historia de Usuario 02, se propone la creación de un algoritmo dedicado a gestionar y procesar las imágenes proporcionadas por el cliente (dispositivo móvil). En este escenario, se requiere obtener un modelo preentrenado con la información sobre las señaléticas de tráfico que contiene nuestro DataSet generado en la historia de usuario 01, para que este sea capaz de reconocer e identificar señales de tránsito.

A continuación, la Tabla 8 presenta de forma detallada la historia de usuario 02, junto con los requisitos específicos que deben cumplirse, enfocado en generar el algoritmo predictor de las señales de tráfico.

Tabla 8*Historia de usuario 2*

Historia de usuario	
Numero: HU02	Usuario: Usuario
Nombre historia: Desarrollo del algoritmo de detección de señaléticas de tránsito.	Numero Sprint: 1
Prioridad: Alta	Riesgo de desarrollo: Baja
Puntos estimados (días): 20 días	Interacción asignada: 1
Desarrolladores responsables: Kevin Gualpa, Angie Chuquitarco, Anthony Unda	
Descripción: Como usuario deseo emplear la cámara de mi dispositivo móvil para la detección de señales de tráfico mediante una experiencia de realidad aumentada.	
Validación:	
Generar un algoritmo de identificación de señales de tránsito robusto y confiable	
Implementar la funcionalidad de reconocimiento de las señaléticas con el framework Xamarin	
<i>Nota.</i> Historia de usuario 2 referente al desarrollo del algoritmo de detección	

El Sprint Backlog 02 proporciona de forma detallada las tareas planificadas para la Historia de Usuario 02 perteneciente al Sprint 01. La Tabla 9 identifica las fechas de inicio y finalización, junto con la duración estimada en horas de cada jornada, adicionalmente se incluye a los desarrolladores responsables de la implementación de este sprint backlog. Además, se detallan las tareas específicas asignadas a cada miembro del equipo para su ejecución.

Tabla 9*Sprint Backlog 2*

Sprint 1						
HU ID	Tareas	Horas	Inicio	Fin	Responsable	Estado
HU02	Desarrollo, entrenamiento y ajuste del algoritmo encargado del reconocimiento de señales de tránsito	24	04/12/2023	15/12/2023	Kevin Gualpa Angie Chuquitarco Anthony unda	Finalizado
	Construcción de la interfaz de usuario que implementa el algoritmo de reconocimiento	20	18/12/2023	02/12/2023	Kevin Gualpa	Finalizado

Nota. Detalle del Sprint Backlog con sus tareas respecto del desarrollo del algoritmo de reconocimiento

Resultado de cumplimiento de la historia de usuario 02. Se obtuvo el desarrollo del algoritmo para detectar señalética. Después de generar y clasificar las imágenes, se utilizaron para entrenar el modelo, luego el siguiente paso es llevar a cabo el desarrollo del algoritmo de detección. Este algoritmo junto con un servidor creado en Flask se encarga de manejar la recepción de la imagen, procesarla en el modelo preentrenado y devolver en formato JSON la información de predicción. En la Tabla 10 se especifica el flujo que sigue el servidor desde que recibe la imagen hasta que devuelve la respuesta de la predicción.

Tabla 10

pseudocódigo servidor de reconocimiento de señales de tránsito

Algoritmo 1: Procesamiento e identificación de las señales de tránsito
Importar YOLO de ultralytics
Importar request, Flask, jsonify de flask
Importar Image de PIL
Importar Flask de flask
Importar CORS de flask_cors
Inicializar la aplicación Flask
App_ Instanciar Flask
Aplicar CORS a la aplicación con origen necesario "*"
Definir una ruta para el directorio raíz "/"
Abrir el archivo "index.html"
Devolver el contenido del archivo
Definir una ruta para "/detect" que acepte solicitudes POST
Obtener el archivo de imagen desde la solicitud
Detectar objetos en la imagen
Devolver los resultados en formato JSON
Definir una función para detectar objetos en una imagen
Crear un modelo YOLO con el algoritmo preentrenado "algoritmo.pt"
Predecir los resultados utilizando la imagen proporcionada
Obtener el primer resultado
Inicializar una lista vacía para almacenar la salida
Para cada caja en las cajas de resultado
Obtener las coordenadas de la caja redondeadas
Obtener el identificador de clase y la probabilidad
Agregar las coordenadas, la clase y la probabilidad a la salida

Algoritmo 1: Procesamiento e identificación de las señales de tránsito

<p>Devolver la salida</p> <p>Colocar el puerto de ejecución</p> <p>Ejecutar la aplicación Flask en modo debug en el host y el puerto que mejor considere</p>
--

Nota. Pseudocódigo de construcción y funcionamiento del servidor Flask

Una vez que contemos con nuestro algoritmo de detección de señales de tránsito, se crea un servidor en Flask que va actuar como BackEnd, por ello necesitamos que exista una comunicación con el FronEnd, por lo que a continuación se va a presentar la arquitectura general del sistema que debe seguir para realizar el envío de imágenes hacia nuestro servidor.

La Figura 13 representa el conjunto de pasos que la aplicación móvil seguirá para implementar el envío de imágenes. Este proceso implica el uso del algoritmo descrito en la Tabla 10 para administrar el envío de imágenes y garantizar una devolución de información precisa en formato JSON. Adicional, se lleva a cabo el proceso de implementación de la modelización de imágenes en tercera dimensión, con el propósito de visualizarlas en pantalla después de la detección.

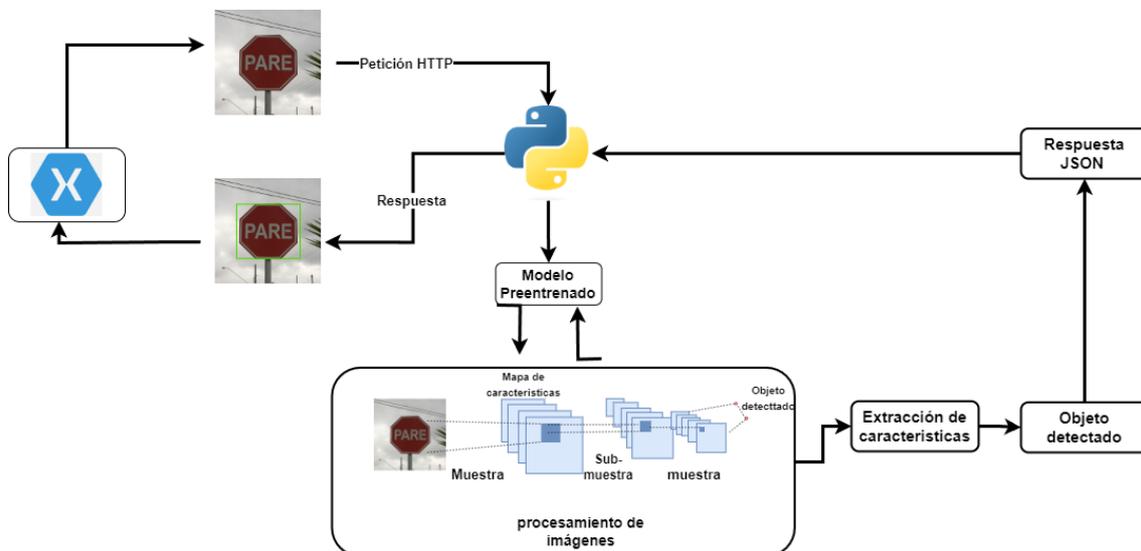
La arquitectura de la aplicación implica el envío de la imagen de una señalética haciendo uso de la cámara del dispositivo, la cual será recibida por el servidor en su ruta "/detect". Es importante aclarar que las imágenes deben ser convertidas a base64 para evitar pérdidas durante la comunicación entre el cliente y el servidor. El servidor una vez que recibe una solicitud POST, se encarga de abrir el modelo de detección de señaléticas, que procesa las imágenes utilizando un modelo de detección de objetos YOLO cargado con el archivo preentrenado para procesar las imágenes. Si las características de la imagen coinciden con un conjunto predefinido de características, se realiza una inferencia que devuelve las coordenadas de la región de interés y la clase correspondiente de la imagen.

Posteriormente, la respuesta se envía de vuelta al FrontEnd en formato JSON, permitiendo así la gestión de la visualización en pantalla con la información obtenida de la detección.

Resultados de la historia de usuario 02, el proceso que sigue esta historia de usuario es la implementación de la interfaz gráfica y se trata de implementar la funcionalidad de realidad aumentada conjuntamente con la detección de objetos previamente logrado con el modelo pre-entrenado, por lo que el principal reto de esta sección es combinar las dos tecnologías, procesamiento de imágenes con reconocimiento de señaléticas.

Figura 13

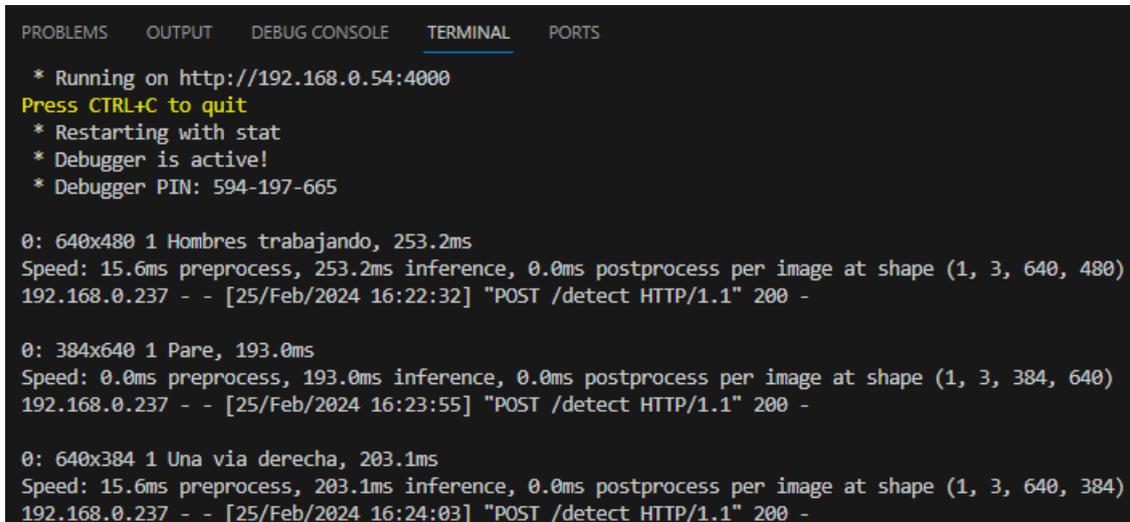
Arquitectura BackEnd



Una vez que logramos combinar las dos tecnologías, buscamos una forma de verificar si el servidor y el cliente se están comunicando correctamente, esto lo logramos cuando el servidor este realizando predicciones en base a imágenes que estamos enviando al servidor, la figura 14 muestra la terminal de nuestro IDE, donde cada vez que el servidor recibe una petición POST de parte del cliente, nos muestra el resultado de su predicción.

Figura 14

Comprobación de funcionamiento de BackEnd



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

* Running on http://192.168.0.54:4000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 594-197-665

0: 640x480 1 Hombres trabajando, 253.2ms
Speed: 15.6ms preprocess, 253.2ms inference, 0.0ms postprocess per image at shape (1, 3, 640, 480)
192.168.0.237 - - [25/Feb/2024 16:22:32] "POST /detect HTTP/1.1" 200 -

0: 384x640 1 Pare, 193.0ms
Speed: 0.0ms preprocess, 193.0ms inference, 0.0ms postprocess per image at shape (1, 3, 384, 640)
192.168.0.237 - - [25/Feb/2024 16:23:55] "POST /detect HTTP/1.1" 200 -

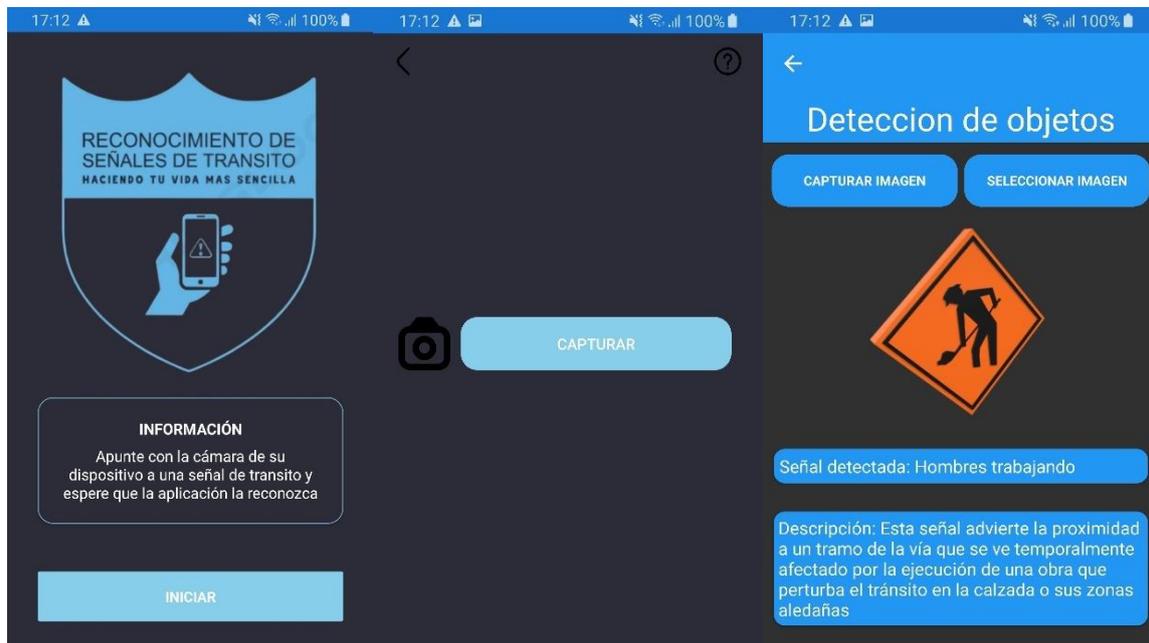
0: 640x384 1 Una via derecha, 203.1ms
Speed: 15.6ms preprocess, 203.1ms inference, 0.0ms postprocess per image at shape (1, 3, 640, 384)
192.168.0.237 - - [25/Feb/2024 16:24:03] "POST /detect HTTP/1.1" 200 -
```

Como se observa en la figura 14, tenemos varias predicciones realizadas por nuestro modelo, de esta forma verificamos que el envío, procesado y reconocimiento de las imágenes se están realizando correctamente.

Luego, debemos realizar la implementación de la interfaz que maneja las respuestas del servidor y muestra la información de cada señal de tránsito con un enfoque de realidad aumentada al usuario. Para ello primero se construye una pantalla inicial, la cual nos dará acceso a varias opciones, en este punto tenemos una primera opción para acceder a un apartado donde la aplicación nos permite acceder a la cámara y tomar una foto o en caso de requerirlo podemos elegir una fotografía de la galería, una vez que capturemos oelijamos una fotografía será enviada al servidor, luego esta interfaz nos mostrará los resultados de la predicción, esta funcionalidad se observa de forma más detallada en la ilustración 15.

Figura 15

Aplicación móvil con implementación de detección de señales



a)

b)

c)

Nota. En a) se muestra la pantalla de inicio de la aplicación, en b) se muestra el menu con la opción captura y finalmente en c) la detección de las señales de tránsito sin la implementación del enfoque de realidad aumentada

El resultado de esta historia de usuario 02 tiene varias observaciones, un punto importante es que no se logro culminar con éxito, debido a varios problemas en la implementación del enfoque de realidad aumentada, ya que las librerías probadas para acceder a la cámara del dispositivo, tienen varias limitaciones al momento de acceder a las APIS de los dispositivo, por lo que tienen acceso solo a funciones básicas de la cámara. Para la implementación de realidad aumentada se necesita enviar varios frames de imágenes al servidor para su detección, sin embargo al tener acceso solo a funcionalidad basica de la cámara realizar este envio de frames no es factible. Otro factor que influye de forma negativa en el desarrollo de este sprint es que, una vez que el servidor retorna la respuesta, debemos sobreponer la información sobre la fotografía que se capturo, sin embargo esta funcionalidad tampoco esta disponible en xamarin por el mismo motivo de

que el framework solo permite el acceso a funcionalidades básicas de cada sistema operativo. Esto dificultó el normal desarrollo del sprint 01. Motivo por el que se optó por implementar la funcionalidad de detección y muestra de información sin el enfoque de realidad aumentada al no encontrar herramientas que faciliten este proceso para el framework xamarin.

Sprint 02. Desarrollo de la funcionalidad de mostrar información de las señaléticas de tránsito vigentes

Para iniciar con la ejecución del Sprint 02, se priorizó la implementación de las historias de usuario 03 y 04, las cuales están enfocadas en mostrar información como nombre, categoría, descripción y una imagen de referencia de las señaléticas, esto se logra con la implementación de otra interfaz de usuario, adicional se va crear una interfaz adicional para poder descargar el manual vigente de la ley de tránsito alojados en los sitios oficiales de la agencia nacional de tránsito.

H.U.03: Desarrollo de la opción de tipos de señales

La Tabla 11 detalla de forma mas clara la historia de usuario 3, junto con los requisitos que debe cumplir para lograr la implementación de la interfaz de información de las señaléticas de tránsito y posterior aceptación del usuario.

Tabla 11

historia de usuario 3

Historia de usuario	
Numero: HU03	Usuario: Usuario
Nombre historia: interfaz de información acerca de las señaléticas de tránsito	Numero Sprint: 2
Prioridad: Media	Riesgo de desarrollo: Baja
Puntos estimados (días): 20 días	Interacción asignada: 1

Historia de usuario

Desarrolladores responsables: Kevin Gualpa

Descripción: Como usuario quiero visualizar de manera detallada la información de las señales de tránsito vigentes en el manual de tránsito.

Validación:

Clasificar las señaléticas en base al manual de señalización de la ANT, se debe organizar por clase a la que pertenece y subir la información a la base de datos.

Obtener y mostrar la información especificada de las señaléticas de tránsito.

Nota. Detalle de la historia de usuario 3, centrada en la funcionalidad de tipos de señales.

A continuación, en la tabla 12 se detalla el sprint backlog 03 que incluye las tareas a ejecutarse, agregamos el número de sprint al que pertenece, la fecha de inicio y finalización. Es importante mencionar que las actividades realizadas en este apartado se encuentran finalizadas.

Tabla 12

Sprint Backlog 3

Sprint 2						
HU ID	Tareas	Horas	Inicio	Fin	Responsable	Estado
HU03	Recopilar datos del manual actual de señalización vial	10	03/01/2024	16/01/2024	Kevin Gualpa Anthony Unda Angie Chuquitarco	Finalizado
	Desarrollar una API para almacenar y exhibir la información relacionada con las	10	17/01/2024	24/01/2024	Kevin Gualpa	Finalizado

Sprint 2

	señales de tráfico.					
	Desarrollar una interfaz de					
HU03	usuario para mostrar la información	20	25/01/2024	31/01/2024	Kevin Gualpa	Finalizado

Nota. Detalle y asignación de tareas pertenecientes al Sprint Backlog 3

El objetivo principal de esta historia de usuario es implementar la interfaz para visualizar las señaléticas de tránsito actuales mediante la implementación de una API, junto con el uso de Firebase que será responsable de administrar los datos.

La programación del algoritmo responsable de gestionar la información (API) se describe en la tabla 13, que especifica el flujo de comunicación entre el backend (API) y la interfaz de usuario. Se busca lograr una baja cohesión y un alto acoplamiento en este proceso. El enfoque incluye la gestión de rutas para acceder a la información de manera adecuada, así como la implementación de operaciones CRUD (Crear, Leer, Actualizar, Eliminar), que son fundamentales en este sistema. }

Tabla 13

algoritmo de gestión de información de señalización vial

Algoritmo 2: API información de las señaléticas

```
# Importa el módulo Router de Express
Importar Módulo Router de Express
# Crea una instancia del enrutador
Instanciar Router como router
# Importa el módulo 'firebase-admin' y obtiene una instancia de Firestore
Importar 'firebase-admin'
Obtener Instancia de Firestore como db
# Exporta el enrutador para su uso en otros archivos
Exportar router como Módulo
```

Algoritmo 2: API información de las señaléticas

Maneja las solicitudes POST para agregar señales

Función ManejarSolicitudAgregarSenial(req, res):

Intentar:

Define un objeto 'newSenial' con los datos de la solicitud

Si no hay req.body.id:

 Crear newSenial con los datos de req.body

Sino:

 Crear newSenial con los datos de req.body incluyendo el id

Agrega la nueva señal a la colección 'seniales' en Firestore

docRef = Agregar newSenial a la colección 'seniales' en Firestore

Devuelve una respuesta exitosa con el ID de la nueva señal

Devolver Respuesta 201 con mensaje "Señal agregada correctamente" e ID de docRef

Capturar Error:

Maneja errores y devuelve una respuesta de error

Imprimir "Error al agregar la señal:" y el error

Devolver Respuesta 500 con error "Error interno del servidor"

Maneja las solicitudes GET para obtener todas las señales

Función ManejarSolicitudObtenerSeniales(req, res):

Intentar:

Obtiene todas las señales de la colección 'seniales' en Firestore

senialesSnapshot = Obtener todas las señales de la colección 'seniales' en Firestore

seniales = []

Itera sobre las señales y las agrega al array

Para cada señal en senialesSnapshot:

 Agregar señal al array seniales con el ID y los datos

Devuelve una respuesta exitosa con todas las señales

Devolver Respuesta 200 con seniales en formato JSON

Capturar Error:

Maneja errores y devuelve una respuesta de error

Imprimir "Error al obtener las señales:" y el error

Devolver Respuesta 500 con error "Error interno del servidor"

Maneja las solicitudes DELETE para eliminar una señal por su ID

Función ManejarSolicitudEliminarSenial(req, res):

Intentar:

Algoritmo 2: API información de las señaléticas

```

# Obtiene el ID de la señal a eliminar de los parámetros de la solicitud
Obtener ID de la señal desde los parámetros de la solicitud
# Elimina la señal de la colección 'seniales' en Firestore
Eliminar señal de la colección 'seniales' en Firestore usando el ID
# Devuelve una respuesta exitosa
Devolver Respuesta 200 con mensaje "Señal eliminada correctamente"
Capturar Error:
# Maneja errores y devuelve una respuesta de error
Imprimir "Error al eliminar la señal:" y el error
Devolver Respuesta 500 con error "Error interno del servidor"

# Maneja las solicitudes GET para obtener una señal por su ID
Función ManejarSolicitudObtenerSenialPorID(req, res):
  Intentar:
    # Obtiene el ID de la señal de los parámetros de la solicitud
    Obtener ID de la señal desde los parámetros de la solicitud
    # Obtiene la señal correspondiente de la colección 'seniales' en Firestore
    docSnapshot = Obtener señal de la colección 'seniales' en Firestore usando el ID
    # Verifica si la señal existe y devuelve la respuesta correspondiente
    Si docSnapshot existe:
      Obtener datos de la señal desde docSnapshot
      Devolver Respuesta 200 con datos de la señal en formato JSON
    Sino:
      Devolver Respuesta 404 con mensaje "Señal no encontrada"
  Capturar Error:
    # Maneja errores y devuelve una respuesta de error
    Imprimir "Error al obtener señal por id:" y el error
    Devolver Respuesta 500 con error "Error interno del servidor"

# Configura las rutas para las diferentes solicitudes
router.post('/api/seniales', ManejarSolicitudAgregarSenial)
router.get('/api/seniales', ManejarSolicitudObtenerSeniales)
router.delete('/api/seniales/id/:id', ManejarSolicitudEliminarSenial)
router.get('/api/seniales/id/:id', ManejarSolicitudObtenerSenialPorID)

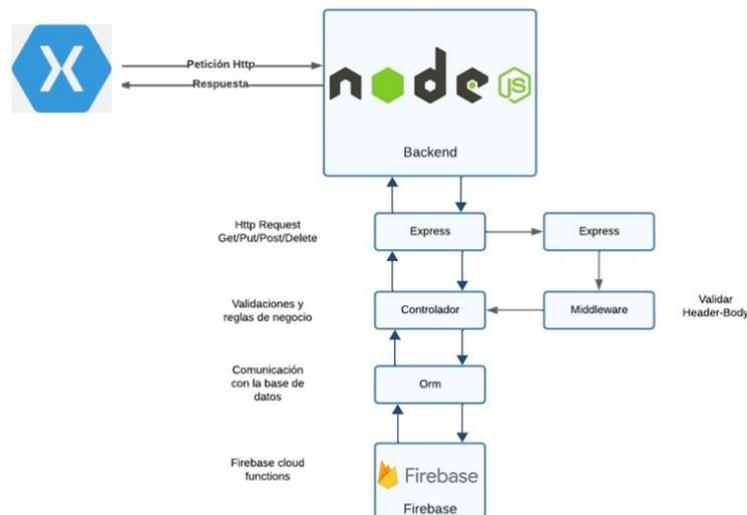
```

Nota. Pseudocódigo de la funcionalidad de la API del sistema web

La Figura 16 muestra la implementación de la arquitectura de la API de información, misma que sigue el flujo detallado en la Tabla 13. En este proceso, Xamarin realiza una solicitud desde el lado del cliente hacia el servidor. Posteriormente, el controlador se encarga de aplicar la lógica de negocio mediante ORM y realiza la interacción con Firebase. Finalmente, se genera un JSON con toda la información cargada. La interfaz de usuario que se va a generar se beneficia de esta estructura para ofrecer una experiencia agradable al usuario con información completa.

Figura 16

Arquitectura de la API de información



Siguiendo la arquitectura mostrada en la Ilustración 16, creamos la interfaz de usuario encargada de presentar la información proveniente del JSON del servidor. El diseño está conformado de la siguiente manera, se exhibe el tipo de señalización en letras grandes, en la parte inferior se acompaña de todas las imágenes asociadas a esa categoría de señal. Luego podemos hacer clic en cualquier señalética, para acceder a información más detallada.

Figura 17

Página de información de señales



a)

b)

c)

Nota. En a) muestra el menú con la nueva opción, en b) muestra toda la información proveniente de la API y en c) se observa de manera detallada la información de cada señalética.

H.U.04: Desarrollo de la opción ley de tránsito de la aplicación

La tabla 14 muestra la historia de usuario 04 de manera detallada, podremos notar que se enfoca en la implementación de la funcionalidad de descargar el manual de la ley de tránsito vigente, además en esta sección se va a especificar los requerimientos necesarios para que la historia de usuario sea implementada de forma correcta.

Tabla 14

Historia de Usuario 4

Historia de Usuario 04	
Numero: H.U.04	Usuario: Usuario

Historia de Usuario 04

Nombre de la historia de usuario: **Numero de Sprint:** 02

Implementar la función descargar PDF del manual de señalización vial vigente

Prioridad: Media

Riesgo de desarrollo: Baja

Días estimados: 15

Interacción asignada: 1

Desarrolladores responsables: Kevin Gualpa

Descripción: Como usuario deseo descargar el manual de señalización vial vigente

Validación:

Acceder al manual de tránsito vigente

Desarrollar una interfaz amigable para descargar el manual

Descargar el manual de tránsito en formato PDF

Nota. Detalle de la historia de usuario 4 enfocada en el desarrollo de la funcionalidad de ley de tránsito.

Sprint Backlog. Habiendo analizado la historia de usuario procedemos a plantear las tareas para el Sprint Backlog, debemos tener en cuenta el número de sprint en el que nos encontramos, fecha de inicio y de finalización, es importante mencionar que estas actividades ya fueron realizadas, por lo que se encuentran finalizadas.

Tabla 15

Sprint Backlog 4

Sprint 2						
HU ID	Tareas	Horas	Inicio	Fin	Responsable	Estado
HU04	Acceder a la normativa de tránsito vigente	10	01/02/2024	02/02/2024	Kevin Gualpa	Finalizado
	Desarrollar la interfaz de usuario para la descarga del manual	15	05/02/2024	08/02/2024	Kevin Gualpa	Finalizado

Sprint 2				
HU04	Desarrollar la	15	09/02/2024	15/02/2024
	lógica para descargar el PDF			
			Kevin Gualpa	Finalizado

Nota. Esta tabla muestra los detalles y las tareas asignadas al equipo, pertenecientes a la funcionalidad de ley de tránsito.

Resultados de la historia de usuario 04, El principal objetivo en esta etapa es la implementación de la funcionalidad que permita al usuario descargar el manual completo de la ley vial vigente, el cual puede ser aprovechado como material de aprendizaje.

Para el cumplimiento de esta historia de usuario se implementó la lógica de descargar el PDF en una interfaz de usuario, en la ilustración 18 se muestran las interfaces involucradas en este proceso.

Figura 18

Diseño móvil de vistas de descarga de manual



a)

b)

c)

Nota. En a) muestra el menú con las opciones actualizadas, en b) muestra las opciones y en c) interfaz de descarga de ley de tránsito.

Observamos que para acceder a la opción para descargar el archivo el usuario deberá ingresar en ley de tránsito, presiona el botón de descarga y así obtendrá el PDF, este material cumple el propósito de servir de apoyo en la educación vial.

Sprint 3. Desarrollo de un sistema web para la administración de la información de las señaléticas de tránsito.

Para la realización del Sprint 03, nos fijamos en la última historia de usuario 05, la cual se enfoca en administrar la información proporcionada por el API en un entorno web. El propósito principal en esta etapa es desarrollar un CRUD que permita manejar la base de datos de las señaléticas de manera completa, por ejemplo, la actualización del manual, la adición de nuevas señaléticas, la edición de las señaléticas existentes y, si es necesario, la eliminación de algunas de ellas.

H.U.05: Desarrollo de sistema web para la gestión de la API

La tabla 16, detalla la historia de usuario 05, esta incluye el nombre a su vez, asignación de prioridad correspondiente en base al riesgo de desarrollo y los requerimientos para la validación.

Tabla 16

Historia de Usuario 5

Historia de Usuario	
Numero: H.U.05	Usuario: Usuario
Nombre de la historia de usuario: Sistema web de administración de la información de la base de datos.	Numero de Sprint: 03
Prioridad: Media	Riesgo de desarrollo: Baja
Días estimados: 15	Interacción asignada: 1

Historia de Usuario

Desarrolladores responsables: Kevin Gualpa

Descripción: Como administrador quiero ingresar, editar y eliminar la información de las señaléticas de tránsito

Validación:

El usuario puede registrarse en el sistema

El usuario ingresa a la página web de gestión

Valida la información y realiza los cambios necesarios

Nota. Se detallan los requerimientos de la historia de usuario 5, enfocada en la realización del sistema web.

Luego generamos el backlog del sprint actual que incluye una lista detallada de las tareas que deben completarse. En la Tabla 17 se especifica el número de sprint correspondiente, así como las fechas de inicio y finalización de cada tarea, junto con el número de la historia de usuario asociada. Es importante destacar que todas las actividades mencionadas en este registro están completadas.

Tabla 17

Sprint Backlog 5

Sprint 3						
HU ID	Tareas	Horas	Inicio	Fin	Responsable	Estado
HU05	Crear y desarrollar el diseño de la interfaz de la página web de gestión	16	16/02/2024	19/02/2024	Kevin Gualpa	Finalizado
	Desarrollar la lógica para integrar un	40	20/02/2024	22/02/2024	Kevin Gualpa	Finalizado

Sprint 3

sistema

CRUD.

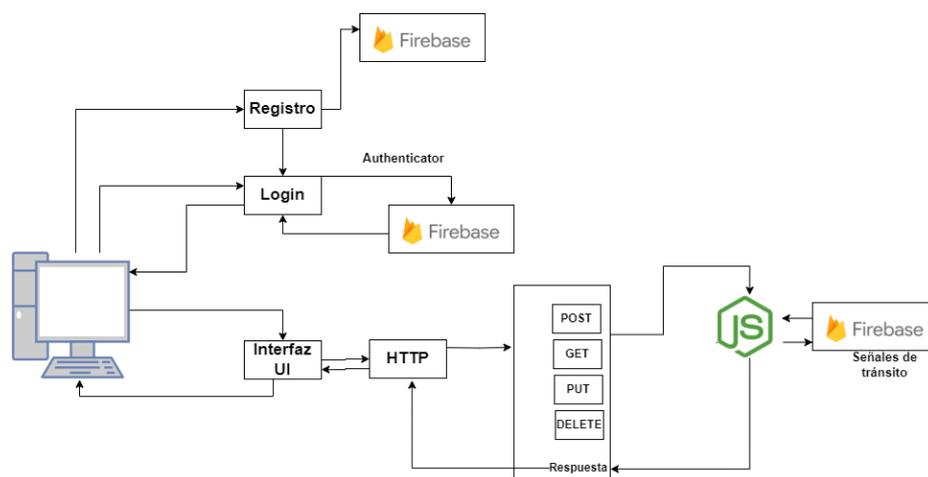
Nota. Se detallan las tareas y responsables de cumplirlas, enfocadas en el desarrollo del sistema web.

Resultados de la historia 5, esta historia de usuario se enfoca en el desarrollo de un sistema web destinada a la administración de la información relacionada con las señalizaciones de tránsito. Además, se busca proporcionar interfaces intuitivas para los usuarios administradores que accederán al panel de gestión.

En este caso, la Figura 19 detalla el flujo del sistema web, que comienza con el inicio de sesión y el registro del usuario. Primero, se verifican las credenciales existentes en la base de datos de Firebase y se accede a la página principal. En cuanto al proceso para cargar una nueva señalética, se realiza una solicitud HTTP al algoritmo mencionado en la Tabla 13, donde se gestiona la información y se guarda en la base de datos. Para cualquier otra solicitud, se envía una petición HTTP para ser procesada por la lógica del BackEnd.

Figura 19

Arquitectura web

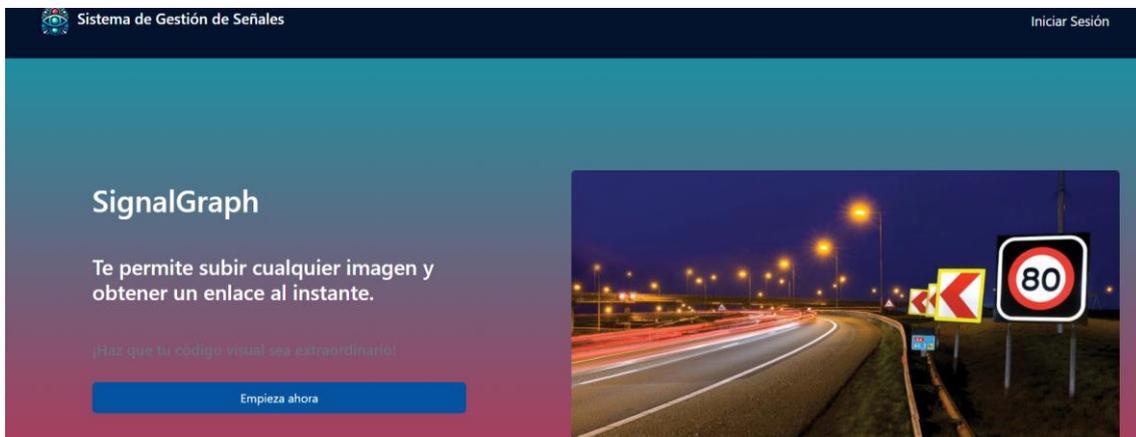


El proceso comienza con la definición de especificaciones del sistema web de gestión. El primer paso consistió en el desarrollo de la página de inicio, cuyo diseño se muestra en la figura 20.

La página principal consta de una breve descripción que informa la función principal de esta página, cuenta con botones de acceso destinado a los usuarios previamente registrados o a los nuevos que deseen registrarse.

Figura 20

Página de inicio de sistema web



Nota. Vista de la página principal

A continuación, figura 21, muestra el funcionamiento de la página de registro, la cual consta de un formulario para que el usuario pueda colocar sus credenciales y crear una cuenta en el sistema, para que así pueda acceder a mayores funcionalidades.

Figura 21

Página de formulario de registro



La figura 22, presenta el formulario de inicio de sesión, mismo que debe completarse con la información previamente registrada, este formulario incluye campo para el correo electrónico y contraseña. Una vez que ingresemos las credenciales correctas accederemos al panel principal.

Figura 22

Página de registro web



Nota: Página de inicio de sesión del sistema web

La figura 23 detalla el diseño del panel de gestión destinada a los usuarios, cuenta con una barra de búsqueda que permite filtrar la señalética según su nombre. Además, presenta una lista de las señalizaciones, información que obtiene de la base de datos Firebase. En esta pantalla poder cerrar sesión y subir nueva información al sistema.

Figura 23

Página principal del sistema web



A continuación, la figura 24 detalla el panel para subir información de una nueva señal. Se dispone de un formulario, con los campos necesarios como el nombre, tipo y clase de la señal, junto con una descripción detallada. También se debe seleccionar la imagen apropiada para la señal. Finalmente, hay un botón de "Guardar" para enviar toda esta información a la base de datos.

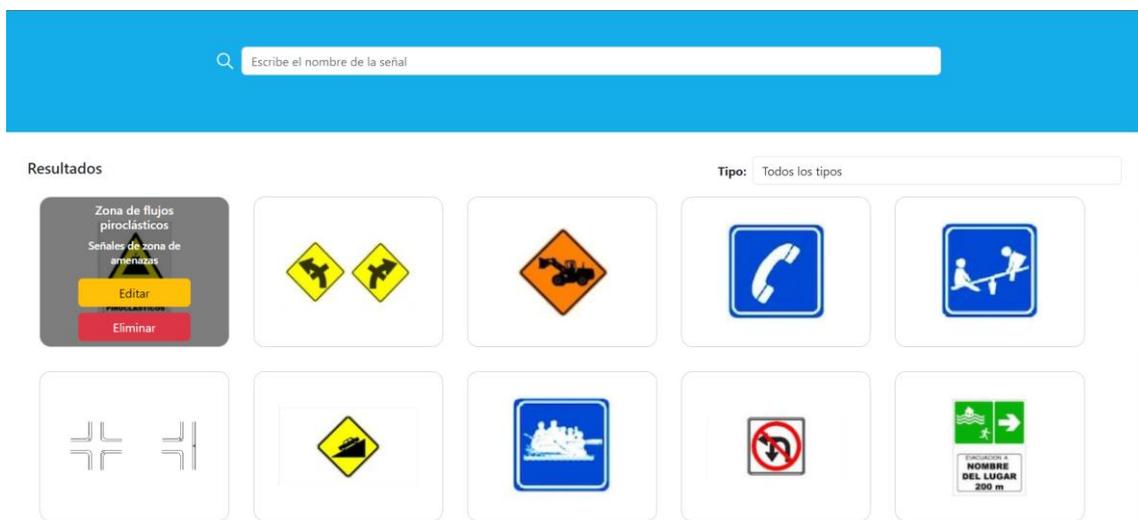
Figura 24

Formulario para subir nueva información

Otra funcionalidad que ofrece el sistema, es que, al pasar el cursor sobre las imágenes de las señaléticas, nos mostrara más opciones como el eliminar y editar en un formato de botones, esto se observa de forma más detallada en la ilustración 25.

Figura 25

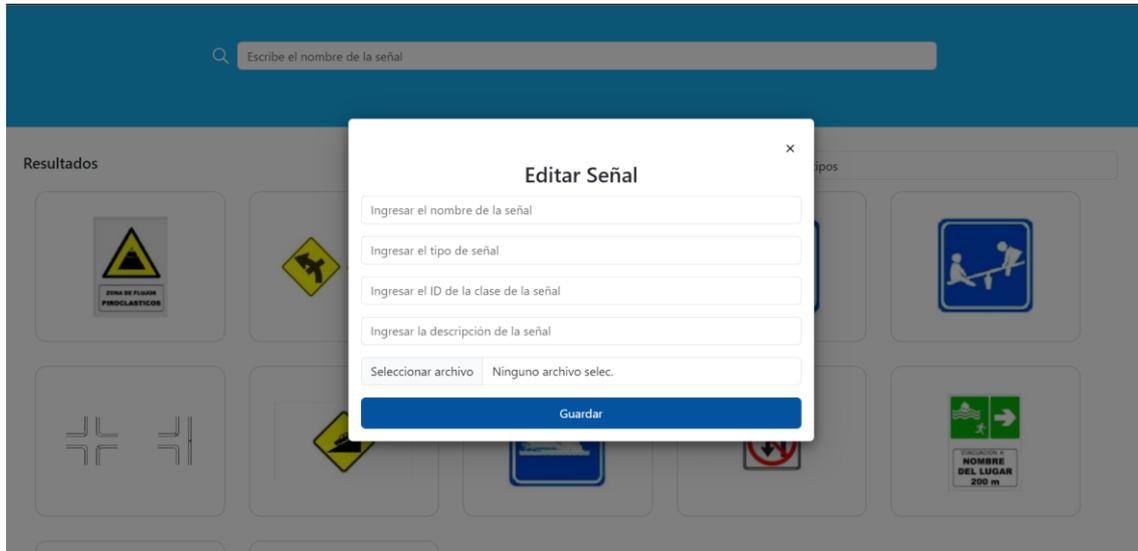
Opciones de editar y eliminar una señalética de tránsito



Si damos clic sobre la opción de editar, se nos desplegara un formulario para realizar la edición eso se muestra de mejor manera en la figura 26.

Figura 26

formulario de editar una señal



La figura 27 muestra la información de contactos de emergencia la cual se divide por secciones presenta el icono de la institución a la que hace referencia, a lado el nombre y el número de teléfono

Figura 27

Página de información de servicios de emergencia



Capítulo VI

Evaluación y Resultados

Validación del sistema

En este capítulo, se valida la eficacia y funcionalidad de nuestra aplicación de detección de señales de tránsito. Para llevar a cabo esta validación, diseñaremos y ejecutaremos una serie de escenarios de prueba que simulen diferentes condiciones. El objetivo principal es presentar los resultados obtenidos, tomando como referencia los diferentes Sprint realizados durante el desarrollo del proyecto.

Cada escenario de prueba se seleccionará para abordar aspectos específicos de la funcionalidad del sistema, como la precisión de la detección y la respuesta en diferentes condiciones ambientales y de iluminación, Se realizará la validación en base a cada sprint.

Marco de Validación

La aplicación móvil diseñada para el reconocimiento de señaléticas, con un enfoque centrado en la realidad aumentada, se ha organizado en dos principales casos distintos para su validación. En el primero de ellos, es identificar las señales de tránsito a través de capturar una fotografía o eligiendo una fotografía de la galería. El segundo caso, se enfoca en presentar y evaluar las diferentes funcionalidades que integra la aplicación. Esto abarca la funcionalidad de tipos de señales de tránsito y la opción ley de tránsito.

En esta sección la única métrica que se va a utilizar es la precisión, Esto se debe a que la funcionalidad principal de la aplicación móvil consiste en identificar señales de tráfico utilizando la cámara frontal del dispositivo. Para llevar a cabo la validación adecuada de todo el sistema, según lo especificado en el proceso de desarrollo se utilizará cada sprint.

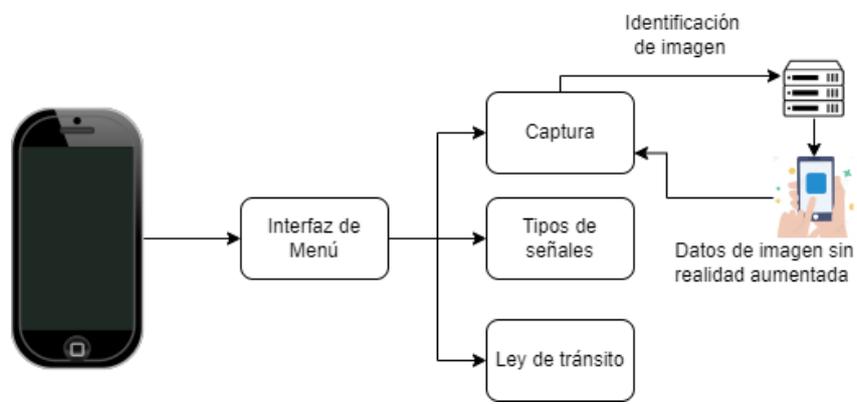
El primer escenario de validación, está conformada por la opción de reconocimiento de nuestra aplicación, por lo que se usaron fotografías de distintas señaléticas ubicadas en

diferentes lugares de la ciudad, estas señaléticas están en buen estado y algunas se encuentran un poco descuidadas.

En la ilustración 28 detalla cómo será el escenario de prueba, donde se realiza la detección de señaléticas de tránsito en base a este flujo podemos validar la funcionalidad de reconocimiento de señales de tránsito.

Figura 28

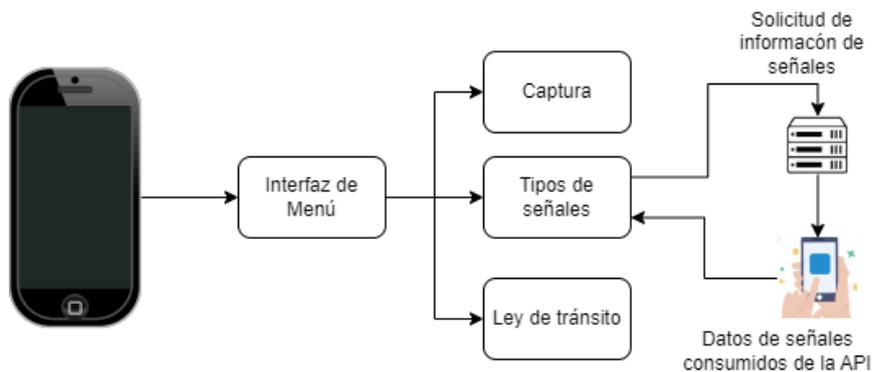
Escenario de validación 1



A continuación, en la ilustración 29 se presenta el esquema del escenario de validación diseñado para la obtención de información de las señaléticas de tránsito que se encuentran en una API. A través de este flujo de validación, podremos confirmar el correcto funcionamiento de este apartado.

Figura 29

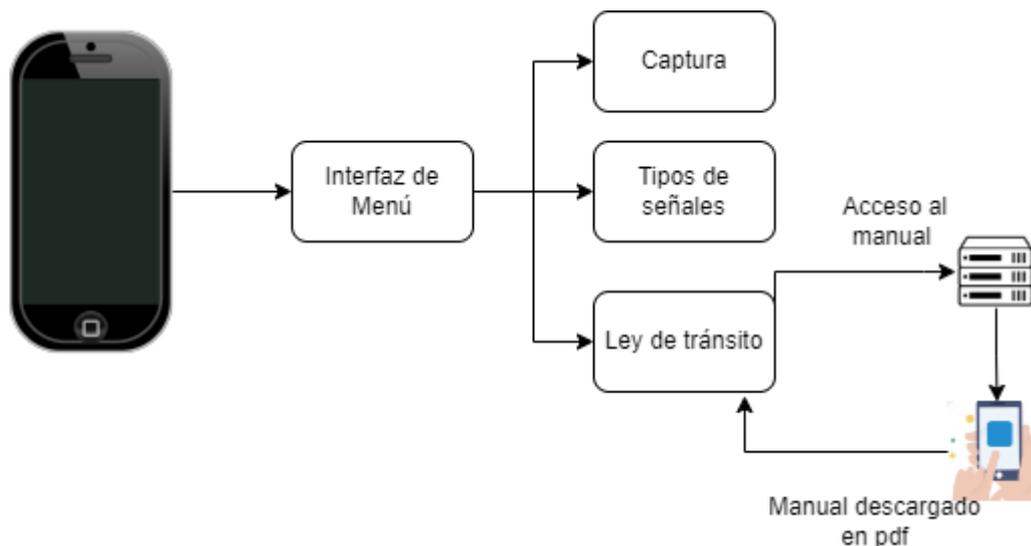
Escenario de validación 2



Finalmente, la figura 30 describe el diseño de un escenario de validación para el caso de ley de tránsito. A través de esta secuencia de pruebas planificada, se podrá verificar el correcto funcionamiento de esta opción de nuestra aplicación.

Figura 30

Escenario de validación 3

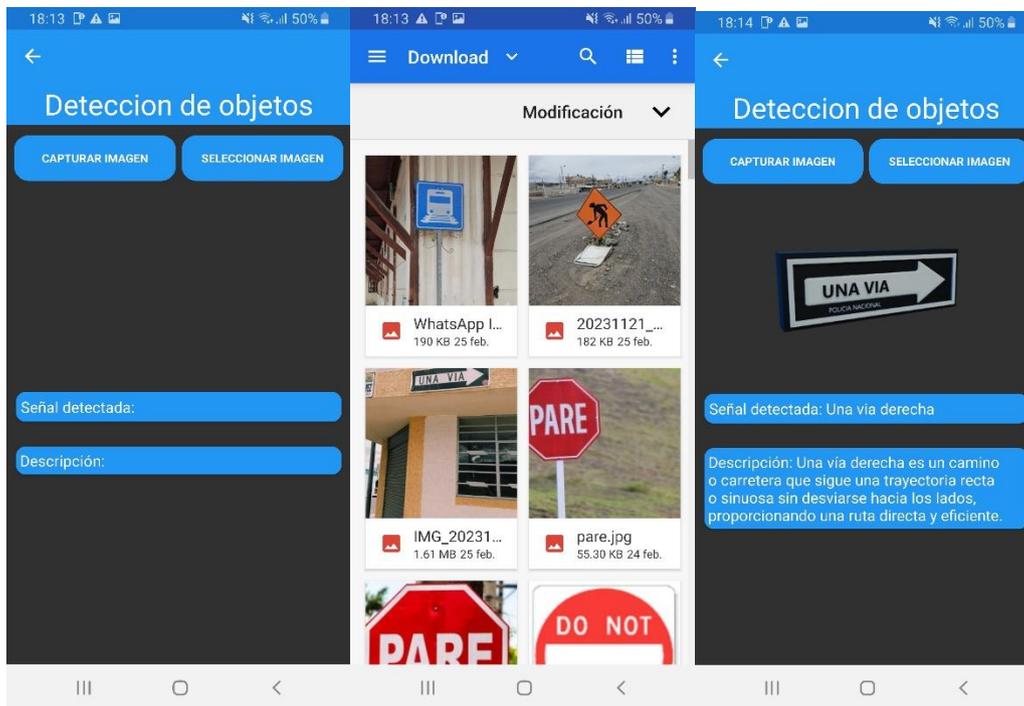


Resultados Obtenidos y Análisis

Fruto de ejecutar el escenario de evaluación 1 tenemos un resultado no muy favorable. La validación se realizó enviando al servidor una fotografía de la señal de tránsito llamada “Una vía”, la fotografía fue tomada en la ciudad de Latacunga en el día, lo que constituye una condición óptima para que el sistema funcione, esto accediendo a la galería del teléfono, Si bien el servidor identifico y retorno de forma correcta la información de la señal de tránsito, no se logró implementar el enfoque de realidad aumentada debido a inconvenientes tecnológicos del framework. Este escenario se muestra en la figura 31.

Figura 31

Resultados escenario 1



a)

b)

c)

Nota. En a) se muestra la interfaz de captura o elección de imagen en b) la galería del dispositivo y en c) se muestra los resultados que proporcione el servidor.

A continuación, en base a los resultados que se obtuvieron de la ejecución del escenario de validación 1, se realizara una lista de verificación con el objetivo de confirmar si las historias de usuario propuestas se cumplen al final del proyecto, podemos observar esta información de forma detallada en la tabla 18.

Tabla 18

Lista de chequeo 1

Lista de chequeo 1		
HU02: Como usuario quiero utilizar mi cámara del dispositivo móvil para el reconocimiento de señaléticas de tránsito con un enfoque de realidad aumentada		
Criterio de aceptación	SI	NO

Lista de chequeo 1

Los usuarios interactúan de forma fluida e intuitiva con la aplicación

X

Los usuarios podrán acceder a la cámara para capturar una fotografía.

X

Los usuarios pueden observar los resultados de la predicción proporcionada por el servidor

X

Los usuarios observan las señaléticas de tránsito con un enfoque de realidad aumentada

X

Nota. En esta tabla se detallan el cumplimiento de las especificaciones requeridas de acuerdo al escenario de Validación 1

Posteriormente, se realizó la ejecución del escenario de validación 2 en este caso el resultado es satisfactorio. La validación se realizó con la ejecución de la opción tipos de leyes, la comunicación con la API, retorno y tratamiento de los datos se realizó con éxito. Este escenario se muestra de forma detallada en la figura 32.

Figura 32*Resultado de escenario 2*

La tabla 19 detalla los resultados que se obtuvieron en la ejecución del escenario de validación 2, a través de una lista de verificación que se utiliza para confirmar si las historias de usuario propuestas se cumplen al final del proyecto, y si se cumplen los criterios de aceptación

Tabla 19*Lista de chequeo 2*

Lista de chequeo 2		
HU02: Como usuario quiero visualizar de manera detallada la información de las señales de tránsito vigentes en el manual de tránsito.		
Criterio de aceptación	SI	NO
Los usuarios pueden acceder a la información de la API	X	
Los usuarios observan las señaléticas de tránsito clasificadas por tipos	X	

Lista de chequeo 2

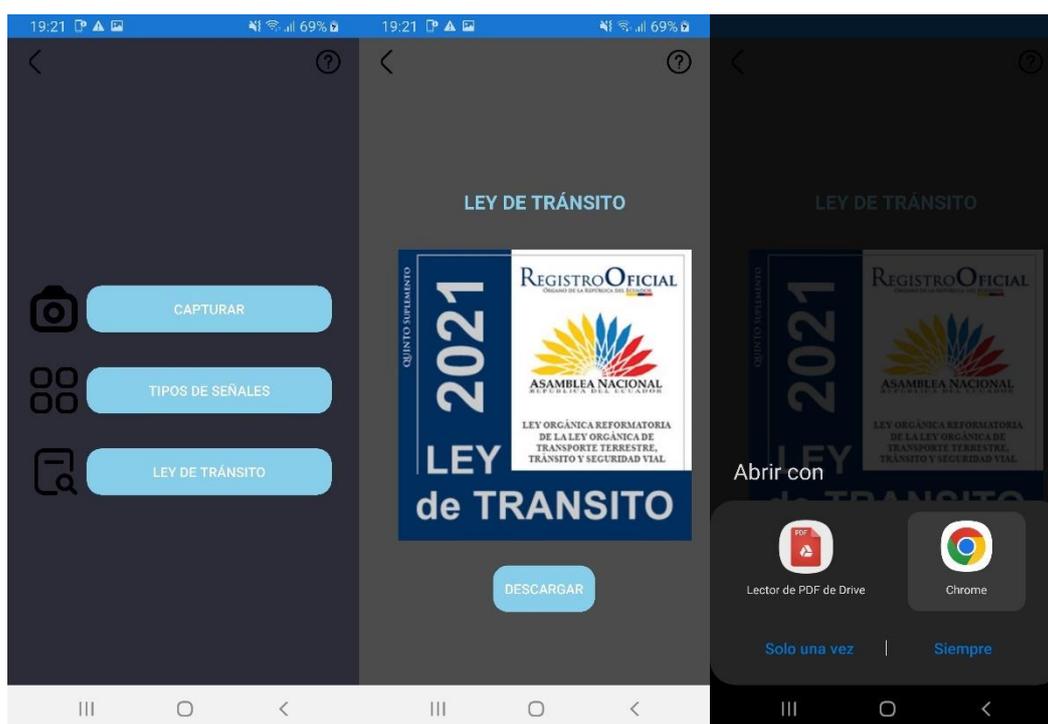
Los usuarios pueden X
seleccionar una imagen para
acceder a mayor información.

Nota. En esta tabla se detallan el cumplimiento de requerimientos de la aplicación en base al escenario de validación 2, enfocada en la opción tipos de señales

Finalmente, se realizó la ejecución de escenario de verificación 3 cuyo resultado fue satisfactorio. La ejecución de la opción ley de tránsito que permite la descarga del manual fue exitosa. Este escenario se muestra de forma detallada en la figura 32.

Figura 33

Resultado de validación 3



La tabla 20 detalla los resultados que se obtuvieron en la ejecución del escenario de validación 3, a través de una lista de verificación que se utiliza para confirmar si las historias de usuario propuestas se cumplen al final del proyecto, y si se cumplen los criterios de aceptación

Lista de chequeo 3

HU02: Como usuario deseo descargar el manual de señalización vial

Criterio de aceptación	SI	NO
-------------------------------	-----------	-----------

Los usuarios pueden acceder a la interfaz correcta	X	
--	---	--

Los usuarios pueden descargar el manual de usuario	X	
--	---	--

Nota. Detalles del cumplimiento de requerimientos de acuerdo a la funcionalidad de ley de tránsito de la aplicación.

Capítulo VII

Conclusiones y Recomendaciones

Conclusiones

- El desarrollo de la aplicación móvil utilizando el Framework Xamarin logró con éxito la implementación del reconocimiento de señales de tránsito, Sin embargo, se encontraron obstáculos para la correcta integración de la realidad aumentada debido a limitaciones técnicas del Framework.
- La exhaustiva investigación de tecnologías disponibles en el campo del reconocimiento de imágenes y realidad aumentada, ha sido de vital importancia para establecer una base sólida de conocimientos que ha guiado el desarrollo de la aplicación.
- El proceso de desarrollo de la aplicación móvil utilizando Xamarin, a pesar de las limitaciones técnicas se ha centrado en la creación de interfaces intuitivas y accesibles para los usuarios. Se han implementado principios de diseño centrado en el usuario para asegurar la usabilidad y la facilidad de navegación, con el objetivo de proporcionar una experiencia fluida y satisfactoria para los usuarios finales.
- La implementación del sistema de reconocimiento de señales de tránsito a través, del servidor realizado con Flask que contiene el modelo Preentrenado se realizó con éxito. Obteniendo como resultado una detección precisa de las señales de tránsito dentro de la aplicación.
- La utilización de una variedad de escenarios de prueba para simular condiciones reales de uso ha sido fundamental para evaluar el rendimiento y la exactitud de la aplicación, permitiendo validar el correcto funcionamiento de la aplicación e identificar áreas de mejoras futuras.

Recomendaciones

- A pesar del éxito en la implementación de la funcionalidad de reconocimiento de señales de tránsito en la aplicación móvil, lamentablemente no se pudo integrar el enfoque de realidad aumentada. Se recomienda explorar otros frameworks que permitan un acceso más amplio a las funcionalidades de los dispositivos, ya que Xamarin ofrece características limitadas.
- Durante el desarrollo de la aplicación, Microsoft anunció el retiro de soporte para Xamarin en mayo de 2024, En consecuencia, es crucial considerar este aspecto al seleccionar Xamarin como framework de desarrollo multiplataforma y explorar posibles alternativas disponibles.
- Se recomienda mantener la información de las apis actualizadas ante posibles modificaciones a las normativas actuales.
- Para garantizar que los usuarios finales aprovechen al máximo el sistema, es esencial proporcionar capacitación continua a través de manuales y recursos de formación. Estos materiales deben estar actualizados regularmente para reflejar las nuevas funciones y actualizaciones del sistema.

Bibliografía

- Abdi, L., Meddeb, A., & Abdallah, F. B. (2015). Augmented Reality Based Traffic Sign Recognition for Improved Driving Safety. En M. Kassab, M. Berbineau, A. Vinel, M. Jonsson, F. Garcia, & J. Soler (Eds.), *Communication Technologies for Vehicles* (pp. 94-102). Springer International Publishing. https://doi.org/10.1007/978-3-319-17765-6_9
- Agudelo, R. F., & Borda, C. M. (2012). *SISTEMA DE RECONOCIMIENTO DE SEÑALES DE TRÁNSITO CON REALIDAD AUMENTADA EN DISPOSITIVOS MÓVILES*.
- Alam, A., & Jaffery, Z. A. (2020). Indian Traffic Sign Detection and Recognition. *International Journal of Intelligent Transportation Systems Research*, 18(1), 98-112. <https://doi.org/10.1007/s13177-019-00178-1>
- Ayunda, P. L., & Budiardjo, E. K. (2020). Evaluation of Scrum Practice Maturity in Software Development of Mobile Communication Application. *2020 3rd International Conference on Computer and Informatics Engineering (IC2IE)*, 317-322. <https://doi.org/10.1109/IC2IE50715.2020.9274625>
- Bañon Rubio, C. (2019). *AUGMENTED REALITY FOR COMPANY ADVERTISING* [fi=AMK-opinnäytetyö|sv=YH-examensarbete|en=Bachelor's thesis]. <http://www.theseus.fi/handle/10024/171816>
- Bouaziz, R., Alhejaili, M., Al-Saedi, R., Mihdhar, A., & Alsarrani, J. (2020). Using Marker Based Augmented Reality to teach autistic eating skills. *2020 IEEE International Conference on Artificial*
- Bytyqi, M. (2022). *Creating a graphical user interface for cross-platform devices with a user-centered agile process*. <https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-474082>
- Čarapina, M. (s. f.). *Developing a multiplatform solution for mobile learning*.

- Carmigniani, J., & Furht, B. (2011). Augmented Reality: An Overview. En B. Furht (Ed.), *Handbook of Augmented Reality* (pp. 3-46). Springer. https://doi.org/10.1007/978-1-4614-0064-6_1
- Criteria for the design and evaluation of traffic sign symbols. (1990). *Applied Ergonomics*, 21(1), 79-80. [https://doi.org/10.1016/0003-6870\(90\)90101-3](https://doi.org/10.1016/0003-6870(90)90101-3)
- Cruz Sanchez, V. G. (2020, diciembre 31). *Detección y Clasificación de Señales de Tráfico Mexicanas Mediante Aprendizaje Profundo* [Artículo]. <http://cathi.uacj.mx/handle/20.500.11961/16403>
- de Souza Cardoso, L. F., Mariano, F. C. M. Q., & Zorzal, E. R. (2020). A survey of industrial augmented reality. *Computers & Industrial Engineering*, 139, 106159. <https://doi.org/10.1016/j.cie.2019.106159>
- Englestone, L. (2021). *.NET Developer's Guide to Augmented Reality in iOS: Building Immersive Apps Using Xamarin, ARKit, and C#*. Apress. <https://doi.org/10.1007/978-1-4842-6770-7>
- Garcia, L. A., Oliveira Jr, E., Leal, G. C. L., Morandini, M., & Urbanowski, S. (2020). Adaptations of Scrum roles in software projects: Survey and representation tentative with feature models. *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*, 47-51. <https://doi.org/10.1145/3422392.3422403>
- Gonzales, I. B. C. (s. f.). *Arq. María de los Ángeles Duarte Pesantez*.
- Hema, V., Thota, S., Kumar, S. N., Padmaja, C., Krishna, C. B. R., & Mahender, K. (2020). Scrum: An Effective Software Development Agile Tool. *IOP Conference Series: Materials Science and Engineering*, 981(2), 022060. <https://doi.org/10.1088/1757-899X/981/2/022060>

- Jacome Leal, J. F. (2023). *Análisis comparativo de los lenguajes de Programación Ionic y Xamarin orientados a entornos de desarrollo móvil*. [bachelorThesis, Babahoyo: UTB-FAFI. 2023]. <http://dspace.utb.edu.ec/handle/49000/13976>
- Karunathilaka, A., Jayasundara, M. A. C. P., Rasanjana, D., Senanayake, S., & Vithana, V. (2015). *Road sign identification application using image processing and augmented reality*. <https://www.semanticscholar.org/paper/Road-sign-identification-application-using-image-Karunathilaka-Jayasundara/765dad186dfdf09307ecc12b6e908c6184d8ec51>
- La realidad aumentada: Lo que debemos conocer | Tecnología Investigación y Academia*. (2018). <https://revistas.udistrital.edu.co/index.php/tia/article/view/11278>
- Makhataeva, Z., & Varol, H. A. (2020). Augmented Reality for Robotics: A Review. *Robotics*, 9(2), Article 2. <https://doi.org/10.3390/robotics9020021>
- Mota, J. M., Ruiz-Rube, I., Doderó, J. M., & Arnedillo-Sánchez, I. (2018). Augmented reality mobile app development for all. *Computers & Electrical Engineering*, 65, 250-260. <https://doi.org/10.1016/j.compeleceng.2017.08.025>
- Prajapati, M., Phadake, D., & Poddar, A. (2016). *STUDY ON XAMARIN CROSS-PLATFORM FRAMEWORK*. 4(4).
- profexorgeek. (2023, julio 13). *¿Qué es Xamarin? - Xamarin*. <https://learn.microsoft.com/es-es/xamarin/get-started/what-is-xamarin>
- Rodríguez Ventura, D. (2016, junio 13). *Diseño e implementación de una aplicación dirigida al control del tránsito vehicular que genere realidad aumentada* [Info:eu-repo/semantics/bachelorThesis]. E.T.S.I y Sistemas de Telecomunicación (UPM). <https://oa.upm.es/44912/>
- Santos, A. R., Sales, A., Fernandes, P., & Nichols, M. (2015). Combining Challenge-Based Learning and Scrum Framework for Mobile Application Development. *Proceedings of*

the 2015 ACM Conference on Innovation and Technology in Computer Science Education, 189-194. <https://doi.org/10.1145/2729094.2742602>

Scharff, C., & Verma, R. (2010). Scrum to support mobile application development projects in a just-in-time learning context. *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*, 25-31. <https://doi.org/10.1145/1833310.1833315>

The History of Mobile Apps: A Comprehensive Examination. (s. f.). Recuperado 26 de febrero de 2024, de <https://www.pocketapp.co.uk/blog/2023/09/25/the-history-of-mobile-apps-a-comprehensive-examination/>

Vishal, K., & Kushwaha, A. S. (2018). Mobile Application Development Research Based on Xamarin Platform. *2018 4th International Conference on Computing Sciences (ICCS)*, 115-118. <https://doi.org/10.1109/ICCS.2018.00027>

Anexos