

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**INTERFAZ DE REALIDAD AUMENTADA PARA RECOMENDADOR DE
SITIOS TURÍSTICOS EN QUITO**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ESPECIALISTA EN SISTEMAS INFORMATICOS Y DE
COMPUTACIÓN**

RAFAEL EDUARDO VINUEZA GARCIA

rafael.vinueza@epn.edu.ec

DIRECTOR: MSc. REGINA MARITZOL TENEMAZA VERA

maritzol.tenemaza@epn.edu.ec

Quito, febrero 2022

APROBACIÓN DEL DIRECTOR

Como directora del trabajo de titulación “Interfaz de Realidad Aumentada para recomendador de sitios turísticos en Quito” desarrollado por Rafael Eduardo Vinuesa García, estudiante de la carrera de Ingeniería en Sistemas Informáticos y de Computación, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.

**REGINA
MARITZOL
TENEMAZA VERA**

Firmado digitalmente por REGINA
MARITZOL TENEMAZA VERA
Nombre de reconocimiento (DN):
cn=REGINA MARITZOL TENEMAZA VERA,
serialNumber=201221114438,
ou=ENTIDAD DE CERTIFICACION DE
INFORMACION, o=SECURITY DATA S.A. 2,
c=EC
Fecha: 2022.02.20 22:38:39 -05'00'

MSc. Maritzol Tenemaza

DIRECTORA DE PROYECTO

DECLARACIÓN DE AUTORÍA

Yo, Rafael Eduardo Vinueza García, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Rafael Eduardo Vinueza García

DEDICATORIA

A mis padres, que me han educado como un hombre de bien.

A mi perrito Chester, por haberme acompañado durante toda mi vida académica.

A mis amigos, por todas las experiencias compartidas en el transcurso de esta etapa universitaria.

Y a mis profesores, que debido a sus conocimientos impartidos me han convertido en un excelente profesional.

Rafael Vinueza

AGRADECIMIENTO

Le agradezco cordialmente a mi directora de tesis MSc. Maritzol Tenemaza por su guía durante todo el desarrollo del proyecto; a MSc. Tania Calle por sus conocimientos brindados en la materia de Sistemas de Información Geográfica que fueron de gran ayuda para la realización de este trabajo; a mi compañero Renato Burbano le brindo un gran agradecimiento, esperando que siga creciendo y haciendo las cosas lo mejor posible; y a mi madre por su apoyo incondicional durante el transcurso de mi etapa universitaria.

Rafael Vinuesa

ÍNDICE DE CONTENIDO

1	<u>INTRODUCCIÓN</u>	18
1.1	PLANTEAMIENTO DEL PROBLEMA	18
1.2	OBJETIVOS.....	19
1.2.1	OBJETIVO GENERAL	19
1.2.2	OBJETIVOS ESPECÍFICOS.....	19
1.3	MARCO TEÓRICO.....	19
1.3.1	REALIDAD AUMENTADA.....	19
1.3.2	GEOLOCALIZACIÓN	21
1.3.3	SISTEMA DE COORDENADAS CARTESIANAS	22
1.3.4	FÓRMULA DE HAVERSINE.....	24
1.3.5	ORIENTACIÓN	24
1.3.6	COMPONENTES NECESARIOS DE RA.....	25
1.4	HERRAMIENTAS, FRAMEWORKS Y LIBRERÍAS DE DESARROLLO	28
1.5	SCRUM.....	30
1.5.1	EQUIPO SCRUM (<i>SCRUM TEAM</i>).....	31
1.5.2	ARTEFACTOS DE SCRUM.....	32
1.5.3	EVENTOS DE SCRUM	32
1.5.4	SPRINT 0	33
2	<u>METODOLOGÍA</u>	34
2.1	SPRINT 0.....	34

2.1.1	ARQUITECTURA	34
2.1.2	DEFINICIÓN DE ROLES.....	36
2.1.3	HISTORIAS ÉPICAS	37
2.1.4	PRODUCT BACKLOG	37
2.1.5	DEFINICIÓN DE <i>SPRINTS</i>	38
2.2	SPRINT 1	39
2.2.1	SPRINT PLANNING	39
2.2.2	EJECUCIÓN DEL SPRINT.....	40
2.2.3	SPRINT REVIEW.....	47
2.2.4	RETROSPECTIVA Y VELOCIDAD DEL EQUIPO	51
2.3	SPRINT 2	53
2.3.1	SPRINT PLANNING	53
2.3.2	EJECUCIÓN DEL SPRINT.....	54
2.3.3	SPRINT REVIEW.....	57
2.3.4	RETROSPECTIVA Y VELOCIDAD DEL EQUIPO	60
2.4	SPRINT 3	61
2.4.1	SPRINT PLANNING	61
2.4.2	EJECUCIÓN DEL SPRINT.....	63
2.4.3	SPRINT REVIEW.....	68
2.4.4	RETROSPECTIVA Y VELOCIDAD DEL EQUIPO	71
2.5	SPRINT 4	73

2.5.1	SPRINT PLANNING	73
2.5.2	EJECUCIÓN DEL SPRINT.....	74
2.5.3	SPRINT REVIEW.....	75
2.5.4	RETROSPECTIVA Y VELOCIDAD DEL EQUIPO	78
3	<u>RESULTADOS Y DISCUSIÓN</u>	79
3.1	PRODUCTO FINAL.....	79
3.1.1	TOMA DE DATOS.....	80
3.1.2	VISTA DE REALIDAD AUMENTADA.....	82
3.1.3	VISTA DE MAPA 2D.....	83
3.2	PRUEBAS CON USUARIOS FINALES	84
3.2.1	FORMULARIO DE RA	85
3.2.2	UTILIDAD PERCIBIDA	86
3.2.3	FACILIDAD DE USO PERCIBIDA	87
4	<u>CONCLUSIONES Y RECOMENDACIONES</u>	89
4.1	CONCLUSIONES.....	89
4.2	RECOMENDACIONES.....	90
5	<u>REFERENCIAS BIBLIOGRÁFICAS</u>	91
6	<u>ANEXOS</u>	94
6.1	ANEXO I: HISTORIAS ÉPICAS	94
6.2	ANEXO II: HISTORIAS DE USUARIO DEL SPRINT 1	95

6.3	ANEXO III: HISTORIAS DE USUARIO DEL SPRINT 2	96
6.4	ANEXO IV: HISTORIAS DE USUARIO DEL SPRINT 3.....	97
6.5	ANEXO V: HISTORIAS DE USUARIO DEL SPRINT 4.....	100
6.6	ANEXO VI: DISEÑO DE LAS INTERFACES DE USUARIO FINAL.....	102
6.7	ANEXO VII: ENCUESTAS REALIZADAS	104

LISTA DE FIGURAS

Figura 1. Primer de dispositivo de RA.	20
Figura 2. Orden de los conceptos de realidad desde la Realidad hasta la Virtualidad.	21
Figura 3. Sistema de coordenadas cartesiano ortogonal (perpendicular entre ejes).	22
Figura 4. Representación de una sección esférica del planeta Tierra como un plano.	23
Figura 5. Los puntos cardinales en una brújula con sus grados de rotación.	25
Figura 6. Aplicación para guiar al conductor utilizando la funcionalidad de GPS.	26
Figura 7. Aplicación Pokémon GO utilizando la brújula del dispositivo móvil.	27
Figura 8. El marco de trabajo Scrum.	30
Figura 9. Diagrama de flujo para la vista de RA.	35
Figura 10. Arquitectura de capas del proyecto.	36
Figura 11. Cámara de RA proporcionada por Vuforia y su llave de licencia	41
Figura 12. Creación de una nueva llave de licencia de Vuforia en la página de desarrolladores	42
Figura 13. Solicitud de permisos para el uso de GPS del dispositivo móvil	43
Figura 14. Adquisición de los últimos valores de latitud y longitud del dispositivo móvil del usuario	43
Figura 15. Obtención de datos cardinales de la brújula del dispositivo móvil.	44
Figura 16. Adquisición de valores del acelerador en la clase GyroManager.	45
Figura 17. Asignación de valores del giroscopio para rotación de la cámara	45
Figura 18. Representación de la fórmula de Haversine en código C#.	46
Figura 19. Función EncontrarCuadrante para colocar elementos en el escenario.	46

Figura 20. Valores de geolocalización obtenidos de la aplicación.....	48
Figura 21. Valor de orientación obtenido de la aplicación.....	49
Figura 22. Primera aproximación para la ubicación de un objeto en 3D en escena	50
Figura 23. Gráfico de Burndown del Sprint 1	52
Figura 24. Modelo para el POI de tipo parque creado en Blender.....	55
Figura 25. Uso de las librerías para detectar el toque en el dispositivo móvil.	56
Figura 26. Función para calcular la escala de los modelos 3D.....	56
Figura 27. Comparación de tamaño entre los iconos a distintas distancias	58
Figura 28. Despliegue del panel de información con datos grabados.	58
Figura 29. Gráfico de Burndown del Sprint 2.....	60
Figura 30. Formulario móvil para la recolección de datos.....	64
Figura 31. Recursos disponibles para el desarrollo de mapas con Mapbox.	65
Figura 32. Uso del token de acceso de Mapbox en Unity	65
Figura 33. Objeto Map con su script AbstractMap.....	66
Figura 34. Función para añadir capa de POIs al mapa 2D.	67
Figura 35. Ejecución del comando npm start para levantar el servicio.....	67
Figura 36. Función para enviar solicitud POST al servicio.....	68
Figura 37. Elemento Player dentro del elemento Map de Mapbox	70
Figura 38. Gráfico <i>burndown</i> del <i>Sprint 3</i>	72
Figura 39. Función createDirections para la creación de rutas.....	74
Figura 40. Instancia del módulo EC2 en AWS para el almacenamiento del servicio	75

Figura 41. Ruta generada entre el elemento Player y un POI	76
Figura 42. Gráfico <i>burndown</i> del <i>Sprint</i> 4	78
Figura 43. Formulario final para la recolección de datos.....	79
Figura 44. Capturas de pantalla de la aplicación móvil final.	80
Figura 45. Campos para los días de viaje y fecha de inicio.	81
Figura 46. Campos para los horarios del tour y de almuerzo.	81
Figura 47. Lugares de preferencia disponibles.....	82
Figura 48. Ejemplo de icono representativo para iglesias.....	82
Figura 49. Selección del día deseado para el despliegue de POIs.	83
Figura 50. Distancia a la que se encuentra un POI del usuario.	83
Figura 51. Vista de Mapa en 2D con sus POIs.....	84
Figura 52. Generación de ruta desde el usuario al POI seleccionado.....	84
Figura 53. Resultados promedio obtenidos en las preguntas del formulario de RA.....	86
Figura 54. Resultados promedio obtenidos en las preguntas del formulario de utilidad percibida.	87
Figura 55. Resultados promedio obtenidos en las preguntas del formulario de facilidad de uso.	88
Figura 56. Diseño del formulario para la toma de datos (Realizado en MockFlow)	102
Figura 57. Diseño de la pantalla de RA (Realizado en MockFlow)	103
Figura 58. Diseño de la pantalla de Mapa en 2D (Realizado en MockFlow)	103
Figura 59. Formulario para medir aspectos de RA.....	104
Figura 60. Formulario para medir la utilidad percibida.	105

Figura 61. Formulario para medir la facilidad de uso.	106
Figura 62. Respuestas para la pregunta 1 de RA.	107
Figura 63. Respuestas para la pregunta 2 de RA.	107
Figura 64. Respuestas para la pregunta 3 de RA.	107
Figura 65. Respuestas para la pregunta 1 de utilidad percibida.	108
Figura 66. Respuestas para la pregunta 2 de utilidad percibida.	108
Figura 67. Respuestas para la pregunta 3 de utilidad percibida.	108
Figura 68. Respuestas para la pregunta 1 de facilidad de uso.	109
Figura 69. Respuestas para la pregunta 2 de facilidad de uso.	109
Figura 70. Respuestas para la pregunta 3 de facilidad de uso.	109

LISTA DE TABLAS

Tabla 1. Roles del equipo Scrum en el proyecto	36
Tabla 2. Historias épicas	37
Tabla 3. <i>Product Backlog</i> inicial	37
Tabla 4. Historias de usuario escogidas para el Sprint 1	39
Tabla 5. Sprint backlog del Sprint 1	40
Tabla 6. Pruebas de aceptación del Sprint 1	47
Tabla 7. Product Backlog adaptado tras finalizar el Sprint 1	50
Tabla 8. Historias de usuario escogidas para el Sprint 2 (Sus detalles se encuentran en el Anexo III).....	53
Tabla 9. Sprint Backlog del Sprint 2.....	53
Tabla 10. Pruebas de aceptación del Sprint 2.....	57
Tabla 11. Product Backlog adaptado tras finalizar el Sprint 2.....	59
Tabla 12. Historias de usuario escogidas para el Sprint 3 (Sus detalles completos se encuentran en el Anexo IV).....	61
Tabla 13. Sprint backlog del Sprint 3	62
Tabla 14. Pruebas de aceptación del Sprint 3.....	68
Tabla 15. Product backlog adaptado tras la ejecución del Sprint 3	70
Tabla 16. Historias de usuario escogidas para el Sprint 4 (Sus detalles completos se encuentran en el Anexo V).....	73
Tabla 17. Sprint backlog del Sprint 3	73
Tabla 18. Product backlog tras la finalización del proyecto.	77

Tabla 19. Historia Épica AH1	94
Tabla 20. Historia Épica AH2	94
Tabla 21. Historia Épica AH3	94
Tabla 22. Historia de Usuario AH1-01	95
Tabla 23. Historia de Usuario AH1-02	95
Tabla 24. Historia de Usuario AH1-03.	96
Tabla 25. Historia de Usuario AH1-04.	96
Tabla 26. Historia de Usuario AH1-05.	97
Tabla 27. Historia de Usuario AH2-01.	98
Tabla 28. Historia de Usuario AH2-02.	98
Tabla 29. Historia de Usuario AH3-02.	99
Tabla 30. Historia de Usuario AH2-03.	100
Tabla 31. Historia de Usuario AH3-01.	101

LISTA DE ANEXOS

6.1	Anexo I: Historias Épicas.....	94
6.2	Anexo II: Historias de Usuario del Sprint 1	95
6.3	Anexo III: Historias de Usuario del Sprint 2	96
6.4	Anexo IV: Historias de Usuario del Sprint 3.....	97
6.5	Anexo V: Historias de Usuario del Sprint 4.....	100
6.6	Anexo VI: Diseño de las interfaces de usuario final	102
6.7	Anexo VII: Encuestas Realizadas	104

RESUMEN

El presente proyecto integrador propone el desarrollo de una aplicación de realidad aumentada (RA) para enriquecer la presentación de información en sistemas de recomendación turística para el centro histórico de Quito-Ecuador, para lo cual se propone una interfaz de RA creada en Unity y Vuforia que permita al usuario interactuar con los elementos virtuales que representan a los puntos de interés (POIs) reales y su información. Paralelamente los datos y las coordenadas geográficas de los POIs se obtienen a partir de una solicitud HTTP a un Rest Api, este servicio web fue desarrollado previamente en un trabajo de titulación y alojado temporalmente en un servidor de la plataforma Amazon Web Services. Seguidamente para ubicar íconos 3D y actualizar su posición con respecto a la del usuario se utilizó la fórmula de Haversine. Adicionalmente, se desarrolló una vista para la representación de un mapa 2D el cual puede ser utilizado por el usuario como referencia para su desplazamiento mediante la herramienta Mapbox. Finalmente, como producto final se obtiene un recomendador turístico con RA visualmente atractivo, que despliega información útil en un dispositivo móvil durante una visita turística.

Palabras clave: Interfaz, Realidad Aumentada, recomendador turístico, puntos de interés, geolocalización, Haversine.

ABSTRACT

This integrative project proposes the development of an augmented reality (AR) application to enrich the presentation of information in tourist recommendation systems for the historic center of Quito-Ecuador, for which an AR interface created in Unity and Vuforia is proposed to allow the user to interact with the virtual elements that represent the real points of interest (POIs) and their information. In parallel, the data and geographic coordinates of the POIs are obtained from an HTTP request to a Rest Api, this web service was previously developed in a degree work and temporarily hosted on a server of the Amazon Web Services platform. Next, the Haversine formula was used to locate 3D icons and update their position with respect to the user's position. Additionally, a view was developed for the representation of a 2D map which can be used by the user as a reference for its displacement through the Mapbox tool. Finally, the final product is a visually attractive AR tourist recommender, which displays useful information on a mobile device during a tourist visit.

Key words: Interface, Augmented Reality, tourism recommender, points of interest, geolocation, Haversine.

1 INTRODUCCIÓN

1.1 Planteamiento del problema

La Realidad Aumentada (RA) es una tecnología que enriquece la percepción de la realidad, la cual añade información virtual a la percepción del mundo real y se genera a partir de un sistema informático [1], es decir la RA potencia nuestros sentidos (vista, oído, tacto y gusto) haciendo que la información del mundo real se complemente con la del mundo digital [2], por ejemplo, hoy en día los teléfonos inteligentes sirven como lente para enlazar el mundo real con el digital; por tanto el objetivo de la RA es simplificar la vida del usuario brindando información virtual (texto, ilustraciones e indicaciones) a través de dispositivos móviles que permitan una interpretación más amigable del entorno cercano y una mayor comprensión del mundo real [1].

El Centro Histórico de Quito es el lugar predilecto para los turistas que visitan la ciudad durante la temporada de vacaciones, alcanzando un 59.69% de total de turistas que visitan esta ciudad [3], además, otros sitios turísticos con gran afluencia son la mitad del Mundo, Miradores y la Mariscal Sucre [3]. Por este motivo, se considera necesaria la creación de un sistema de recomendación turística que ayude a la planificación y experiencia de un viaje o visita turística, de esta manera facilitando la información adecuada sobre rutas y datos turísticos.

Por tal razón y con el fin de fomentar el turismo y ofrecer una alternativa para mejorar la experiencia del turista en la ciudad Quito, se desarrollará una aplicación móvil que despliegue la información sobre puntos de interés (POIs) a través de una interfaz de RA, la cual permitirá que la aplicación disponga de una nueva alternativa en la presentación de la información sobre una realidad atractiva existente.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar una interfaz de RA enfocada a una aplicación en dispositivos móviles para un sistema de recomendación turístico en la ciudad de Quito.

1.2.2 Objetivos Específicos

- Identificar las APIs y herramientas de software necesarias para aplicar Realidad Aumentada.
- Implementar los módulos para presentar los POIs a visitar, los cuales serán extraídos mediante un servicio provisto desde un proyecto de titulación previamente desarrollado con ese fin.
- Implementar la interfaz de Realidad Aumentada enfocada a dispositivos móviles.
- Probar la funcionalidad de la interfaz de Realidad Aumentada en la ciudad de Quito, comprobando que cumple los parámetros establecidos. Se probará con al menos 3 usuarios.
- Evaluar la usabilidad de la aplicación mediante encuestas.

1.3 Marco Teórico

1.3.1 Realidad Aumentada

Realidad aumentada es la forma de ver el mundo y como este se ve transformado al combinar la realidad virtual con la real. Para comprender el concepto de RA se inicia con los sentidos humanos, los cuales son receptores de la realidad y brindan una interpretación del mundo que nos rodea, si se añade esta realidad una capa virtual tendremos como resultado la RA [4]. En la realidad virtual cuando el usuario se sumerge en un ambiente artificial, imposibilita el contacto con el mundo real [4], en cambio, la RA le ofrece la capacidad al usuario de visualizar el mundo real añadiéndole objetos virtuales sobrepuestos al mismo [4].

1.3.1.1 Historia

La historia de la RA data de los años 50's, donde el director cinematográfico Morton Heilig propuso la idea de combinar la experiencia del observador con la actividad que se desarrolla en pantalla, haciendo uso de todos los sentidos de manera efectiva. En años posteriores esta idea se convirtió en un primer prototipo llamado Sensorama [5]. Para el

año de 1968, Iván Sutherland fue el primero en crear un sistema de RA usando un dispositivo colocado en la cabeza del usuario con una pantalla transparente como puede observarse en la Figura 1.



Figura 1. Primer de dispositivo de RA [5].

Posteriormente en el año 1975 se crea el primer “Video place” por Myron Krueger, el cual permitía a su clientela interactuar por primera vez con objetos virtuales. A partir de estos acontecimientos, la cantidad de sistemas y aplicaciones enfocadas en RA han aumentado considerablemente, incluso llegando a incursionar en el campo de los dispositivos móviles como son los iPhone de la empresa Apple [5].

1.3.1.2 Realidad Aumentada en dispositivos Móviles (MAR)

La RA ha tenido un gran desarrollo en la última década llamando la atención de entidades académicas, así como empresas interesadas en el desarrollo de aplicaciones para su consumo en el mercado. Esta tecnología que se encuentra en dispositivos móviles o MAR por sus siglas en inglés busca complementar el entorno real de un usuario mediante el despliegue de contenido virtual, generado por su dispositivo móvil [6]. En la Figura 2 podemos apreciar cómo se categorizan las distintas versiones de realidad y virtualidad dependiendo del grado en que una aplicación cambia el entorno mediante la generación de elementos virtuales; la realidad es el entorno que observa el usuario sin el uso de dispositivos que alteren su percepción, mientras que la Realidad Virtual es el extremo opuesto donde la realidad del usuario esta generada por computadora sin interacción alguna con el entorno real [6].

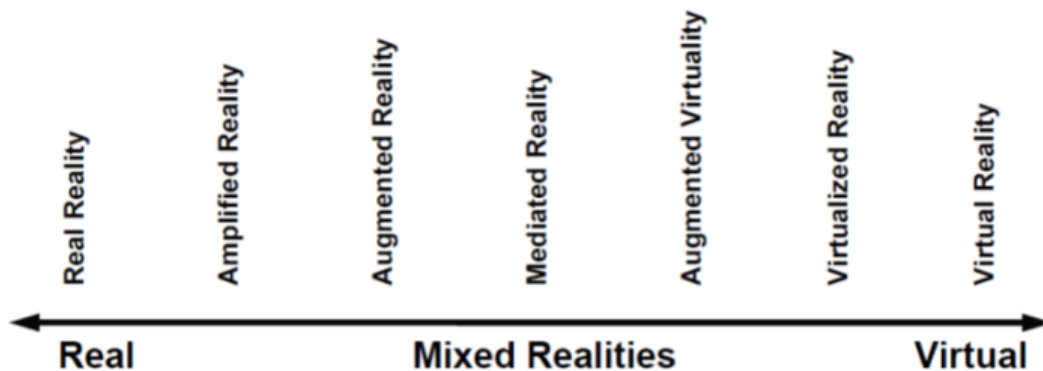


Figura 2. Orden de los conceptos de realidad desde la Realidad hasta la Virtualidad [6].

Debido a su naturaleza, la MAR puede ser diseñada para ejecutarse en una gran variedad de dispositivos móviles como pueden ser teléfonos inteligentes, gafas inteligentes, relojes inteligentes, tabletas y en algunos casos incluso en laptops [6]. Para que una aplicación pueda ser categorizada como MAR debe poseer 3 características fundamentales [6]:

- **Entrada:** Hacer uso de los sensores integrados en el dispositivo tales como la cámara, el GPS, el giroscopio, el acelerómetro, etc.
- **Procesamiento:** Determinar el tipo de información que se va a desplegar en la pantalla del dispositivo móvil.
- **Salida:** Proyectar la salida a la pantalla del dispositivo móvil en conjunto con la vista por defecto del usuario, aumentando de esta manera la realidad que percibe el usuario.

1.3.2 Geolocalización

La geolocalización es un término utilizado para hacer referencia a la situación que ocupa un objeto en el espacio, se lo conoce comúnmente como localización y en el Internet adopta conceptos como geoposicionamiento y georreferenciación [7]. La geolocalización utiliza coordenadas como la latitud (x), longitud (y) y altura (z) y para representar un objeto se describe mediante un punto, vector, área o volumen [8]. La geolocalización al igual que la RA comparten la característica de ser herramientas de comunicación entre el mundo físico y digital, puesto que la llegada del internet y dispositivos móviles ha otorgado una nueva dimensión a la geografía.

Las técnicas de geolocalización son aquellas que nos permiten determinar la ubicación de una persona u objeto dentro de este planeta. Las aplicaciones que implementan técnicas de geolocalización abren las puertas a varios posibles escenarios como la obtención de datos de tráfico e indicaciones detalladas para llegar a una ubicación específica, por ejemplo, la policía y equipos de rescate serán capaces de identificar a civiles perdidos o con heridas que requieran atención médica, las empresas podrán ubicar de sus vehículos u otras pertenencias[9].

1.3.3 Sistema de Coordenadas Cartesianas

La necesidad de ubicar personas, objetos en su posición ha llevado a crear sistemas que se asemejen mucho a la realidad, tal es el caso del sistema de coordenadas cartesianas creado por René Descartes, quien las utilizó de manera formal por primera vez en cartografía, cuya utilidad se mantiene hasta la actualidad, aunque presentan cierto nivel de imprecisión, más aún considerando que nuestro planeta no es una esfera perfecta [10].

Las coordenadas cartesianas son expresadas en pares, cada elemento interpreta un plano y la distancia que existe desde el punto de origen. La distancia entre dos puntos si es paralela al eje x, significa que es una coordenada x, y si es paralela al eje y, se llama coordenada y [10]. En la Figura 3 se puede observar a cada eje perpendicular al otro como regla importante para tener mejor representación del espacio.

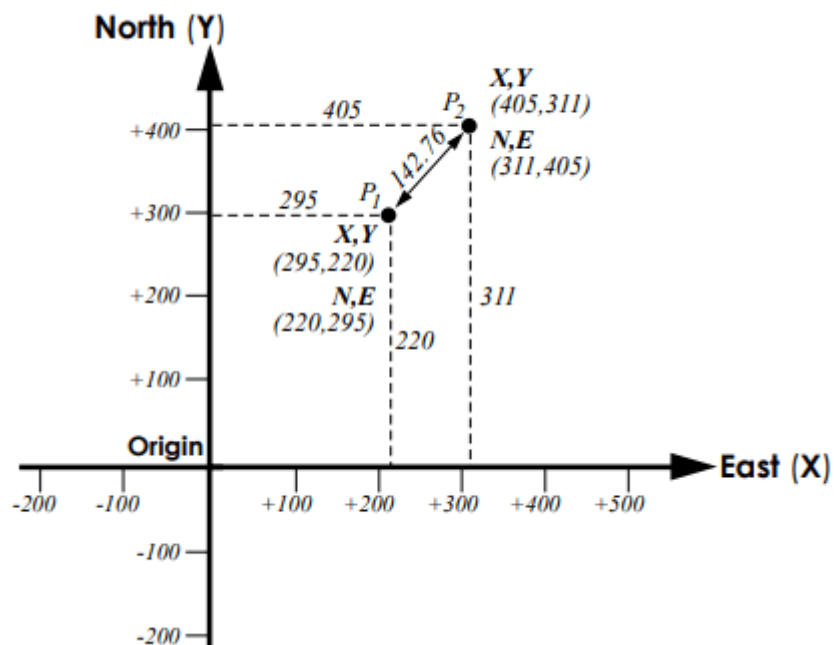


Figura 3. Sistema de coordenadas cartesianas ortogonales (perpendicular entre ejes) [10].

El sistema cartesiano contiene 4 cuadrantes y se distingue por tener en el eje y positivos hacia el norte y negativos hacia el sur, mientras que x maneja valores positivos al este y negativos al oeste. Con estos principios se mantienen las reglas de la geometría euclidiana, en el diseño asistido por computadora (CAD) y el software del Sistema de Información Geográfica (GIS) [10].

1.3.3.1 Coordenadas Cartesianas y la realidad

El uso del sistema cartesiano tiene un fallo con respecto a traer un plano a la realidad, esto se debe a la forma de nuestro planeta por cuanto no es un plano 2D sino un esferoide (esfera achatada por los polos). Entre los sistemas de coordenadas, la representación más conocida es el Universal Transverse Mercator (UTM), diseñado para representar nuestro planeta de manera redonda en el plano cartesiano, como se puede observar en la Figura 4. Para escenarios pequeños la distorsión no se resalta, pero en áreas demasiado grandes esta es notoria [10], finalmente las coordenadas geográficas son útiles en la trigonometría convencional.

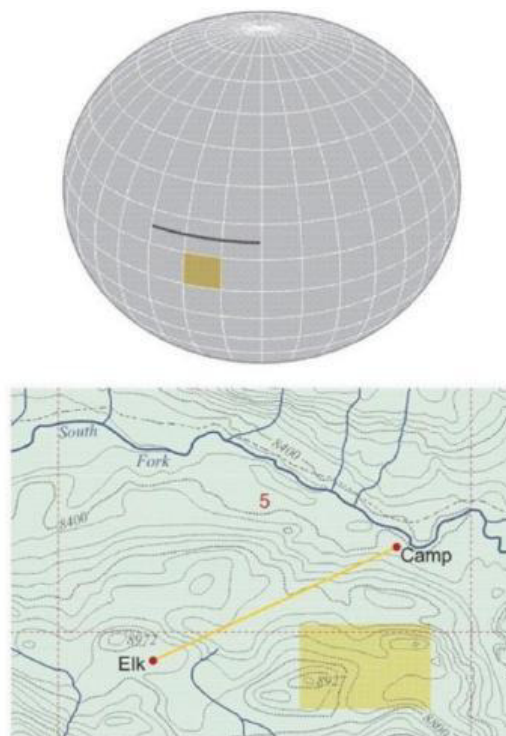


Figura 4. Representación de una sección esférica del planeta Tierra como un plano [10].

1.3.4 Fórmula de Haversine

Uno de los problemas más relevantes de la navegación es que las coordenadas de latitud y longitud son insuficientes para determinar con precisión la posición de un objeto, de tal manera que la fórmula denominada “cosine-haversine” ofrece una solución a este problema, determinando la altitud de un cuerpo utilizando las expresiones “haversine” de la siguiente manera [11]:

$$\text{hav}\left(\frac{d}{r}\right) = \text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1)$$

Donde:

- d es la distancia entre los dos puntos
- r es el radio de la tierra
- φ_2 y φ_1 son la latitud del punto 2 y del punto 1 respectivamente
- λ_2 y λ_1 son la longitud del punto 2 y del punto 1 respectivamente

Es posible desarrollar ampliamente esta fórmula utilizando relaciones fundamentales de trigonometría plana, con el fin de despejar la variable d la cual representa la distancia entre dos puntos. La fórmula de Haversine para calcular la distancia entre dos puntos basándose en la latitud y longitud es la siguiente [12]:

$$d = 2r \sin^{-1} \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

1.3.5 Orientación

El concepto de orientación hace referencia a determinar la ubicación de un objeto haciendo referencia a los puntos cardinales, la manera en que un objeto en el espacio se mueve, en qué dirección se desplazan y el lugar en donde debería estar ubicado físicamente [13]. Una de estas referencias son los puntos cardinales, en su forma más básica podemos identificarlos como 4: Norte, Sur, Este y Oeste, aunque en un sentido más amplio pueden ser ramificados en puntos más específicos con la unión de los 4 nombres básicos como por ejemplo Noreste o Suroeste [13]. En base a los puntos cardinales se puede determinar el ángulo al que una persona u objeto apunta dependiendo de la rotación que posea respecto del Norte Magnético del planeta, el cual representa los 0° [13]. En la Figura 5

podemos apreciar la representación de los puntos cardinales en una brújula, acompañado con los grados a los que se encuentra apuntando el objeto o persona que la sostiene.

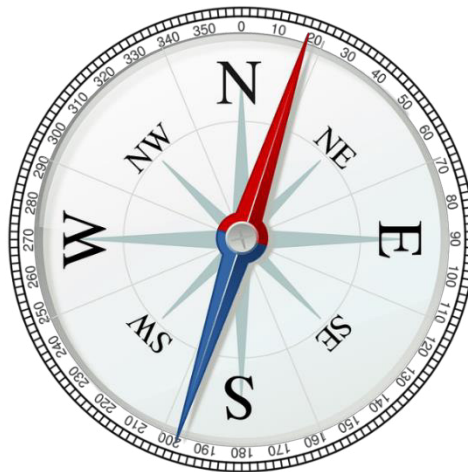


Figura 5. Los puntos cardinales en una brújula con sus grados de rotación [14].

1.3.6 Componentes necesarios de RA

1.3.6.1 GPS

Las siglas GPS hacen referencia a “Global Positioning System”, esta es una tecnología que permite determinar la posición geográfica de un elemento en cualquier sitio del planeta basándose en dos coordenadas de referencia denominadas latitud y longitud [15]. La mayoría de los dispositivos móviles actuales con sistema operativo Android poseen la funcionalidad de GPS integrada por defecto, para acceder a ella es necesario activarla en las configuraciones del dispositivo móvil y para el desarrollo de aplicaciones móviles provee la funcionalidad de utilizar las coordenadas de latitud y longitud en las que se encuentra el usuario al utilizar la aplicación móvil, siempre y cuando cuente con los permisos necesarios [15].

Esta tecnología depende en gran medida de la ubicación del usuario por cuanto requiere conectarse a los satélites que proveen el servicio de GPS y en caso de no estar ubicado en una posición óptima como dentro de túneles o en el subsuelo de un edificio, pueden surgir interferencias e incluso sería imposible acceder a las coordenadas geográficas [15], un ejemplo de aplicaciones que requieren el acceso al GPS del dispositivo móvil son aquellas que utilizan los conductores, identificando la ruta más corta a una ubicación, tal y como podemos observar en la Figura 6.



Figura 6. Aplicación para guiar al conductor utilizando la funcionalidad de GPS [14].

1.3.6.2 Brújula

La brújula es un elemento importante para las aplicaciones móviles, al igual que el GPS viene por defecto integrada en la gran mayoría de dispositivos móviles de la actualidad y su funcionamiento consiste en determinar los 3 ejes cardinales conocidos comúnmente como X, Y y Z; este instrumento en los dispositivos móviles es capaz de detectar el campo magnético del planeta para determinar la dirección donde se encuentra el norte magnético y presentar la dirección en que se desplaza el usuario, además de cuantos grados de rotación se encuentra el observador [16]. Un detalle importante de las brújulas de los Smartphones es que generalmente suelen presentar fallos en sus mediciones, lo cual dificulta la utilización de datos precisos y fiables en las aplicaciones [16]. En la Figura 7 se observa una de las aplicaciones móviles más famosas en el ámbito de los videojuegos, como Pokémon GO, que hace uso de la brújula del dispositivo móvil para determinar la dirección en la que el usuario se desplaza.

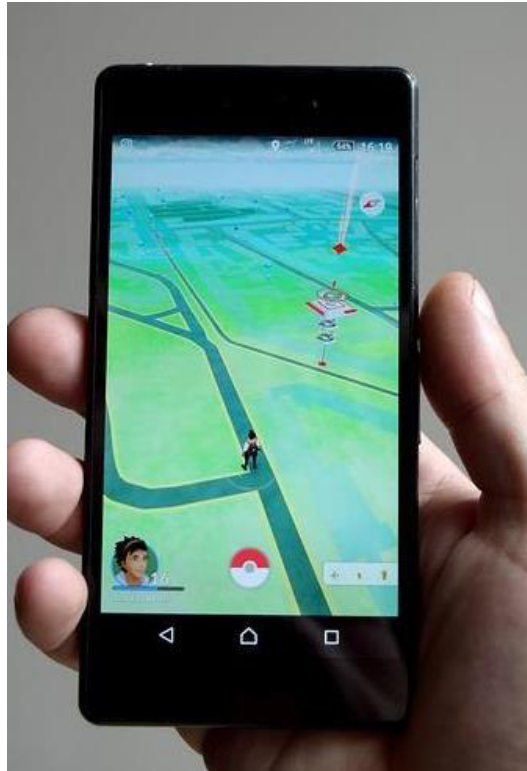


Figura 7. Aplicación Pokémon GO utilizando la brújula del dispositivo móvil [14].

1.3.6.3 Acelerómetro

Esta herramienta tecnológica integrada en la mayoría de los dispositivos móviles es de gran utilidad para el desarrollo de aplicaciones de RA, por cuanto es capaz de medir la aceleración a la que se desplaza el usuario con su dispositivo móvil y las fuerzas que se generan en el mismo por el efecto de la gravedad [17]. Los datos medidos por el acelerómetro del dispositivo móvil son de vital importancia si se desea detectar el movimiento que realiza el usuario y ofrecen la posibilidad de actualizar la posición de los elementos 3D que se generen en una aplicación de RA. Los valores más importantes que podemos obtener del acelerómetro del dispositivo móvil es la aceleración en los ejes x, y, z la cual nos permite desplazar los objetos alrededor del usuario conforme a su movimiento y de esta manera ofrecer la sensación de que los iconos se encuentran fijos en cada POI.

1.4 Herramientas, frameworks y librerías de desarrollo

Unity

Unity es una plataforma para el desarrollo de aplicaciones tanto en 2D como en 3D, creada por la compañía Unity Technologies, esta plataforma y sus herramientas principales están enfocadas en el desarrollo de videojuegos, incluyendo experiencias de realidad virtual y RA [18]. Unity ofrece una amplia gama de librerías y opciones que facilitan el desarrollo de aplicaciones móviles en el sistema operativo Android y en un ambiente 3D, ofrece una buena integración con herramientas de RA como lo son: Vuforia y AR Foundation de la empresa Google [18], por estos motivos se ha considerado que la plataforma sirva para el desarrollo del recomendador turístico.

Vuforia

Esta herramienta está basada en el reconocimiento de imágenes para desplegar los elementos 3D [19], es un kit de desarrollo de software (SDK), como por ejemplo aplicaciones basadas en RA. Vuforia posee una buena integración con la plataforma Unity, por cuanto permite desplegar la cámara de RA reconociendo los elementos 3D de Unity que se despliegan en escena, permitiendo mostrar a través de la cámara del dispositivo móvil del usuario estos elementos y generando la experiencia de RA que se busca en la aplicación.

Mapbox

Es la plataforma de mapas de código abierto con mayor reconocimiento en internet, posee su propia base de datos geográficos, gran cantidad de APIs y servicios, permitiendo integrar esta herramienta de manera sencilla con cualquier otra plataforma de desarrollo [20]. El SDK de Mapbox se incorpora en la plataforma Unity, el cual facilita recrear una vista en 2D del mapa sobre la ubicación del usuario, acompañado de los POIS en los alrededores.

Visual Studio

Es un entorno de desarrollo integrado (IDE), utilizado para la generación de códigos y aplicaciones en sistemas operativos Windows y MacOS, soporta varios lenguajes de programación entre los cuales se destacan C++, Visual Basic y C#. Este entorno es el IDE por defecto para la creación de scripts de la plataforma Unity y permite la integración de un

enorme catálogo de librerías que facilitan el desarrollo tanto de aplicaciones de escritorio, aplicaciones móviles y aplicaciones web [21].

Blender

Es un software informático que está enfocado en el diseño, modelado y animación de elementos 2D y 3D, posee una amplia gama de herramientas que favorecen el modelado, animación, iluminación, renderizado y creación de gráficos 3D, llegando a ser altamente utilizado por los desarrolladores de videojuegos para la creación de assets, gracias a su gran compatibilidad con Unity [22]. Se ha escogido esta herramienta para la creación de iconos desplegados que se usan en la aplicación y representan a los POIs.

Github

Es una plataforma y servicio incorporado en la nube cuya utilidad es facilitar a los desarrolladores almacenar y mantener el código de sus proyectos de software utilizando git, además, es un sistema de control de versiones de código abierto creado por Linus Torvalds en 2005 [23]. Github se utiliza para versionar y almacenar el código a lo largo del proceso del proyecto.

Postman

Es un cliente HTTP disponible para cualquier sistema operativo y sirve para analizar cualquier API RESTful de una manera cómoda y sencilla [24]. Esta aplicación permite hacer pruebas a servicios web, por ejemplo, se aplicó en el servicio web generado en un proyecto de investigación antecesor, el cual devuelve los POIs cercanos dependiendo de los parámetros enviados por el usuario.

Amazon Web Services (AWS)

Es la plataforma de servicios en la nube más populares y robustas en Internet, entre sus principales servicios se destacan el cloud computing, máquinas virtuales en la nube con características de gama alta, buckets para el almacenamiento de archivos y páginas web estáticas con una amplia capacidad de almacenamiento, entre muchos otros [25].

EC2

El servicio EC2 de AWS consiste en la creación de máquinas virtuales en la nube de Amazon con una amplia gama de características, las cuales pueden ser configuradas para adaptarse de mejor manera a las necesidades y presupuesto de los usuarios. Al utilizar este servicio es posible crear servidores en la nube en un corto lapso y evitar el costo de mantenimiento en el caso de optar por servidores locales [26].

1.5 Scrum

Esta metodología es la más adecuada para el cumplimiento de los objetivos planteados en el proyecto de titulación, por cuanto permite el desarrollo ágil de entregables de un proyecto, facilita su acoplo con cronogramas previamente establecidos y sirve para equipos de trabajo pequeños.

Scrum es un marco de trabajo diseñado con el fin de desarrollar, entregar y mantener productos y servicios de manera innovadora. Está basado en el empirismo, por cuanto el conocimiento se genera a partir de las experiencias y en el proceso de toma de decisiones, las cuales están fundamentadas en el conocimiento previo y en el respeto a los tres pilares fundamentales del empirismo, como son: transparencia, inspección y adaptación [27]. Una abstracción del marco de trabajo de Scrum se puede observar en la Figura 8.

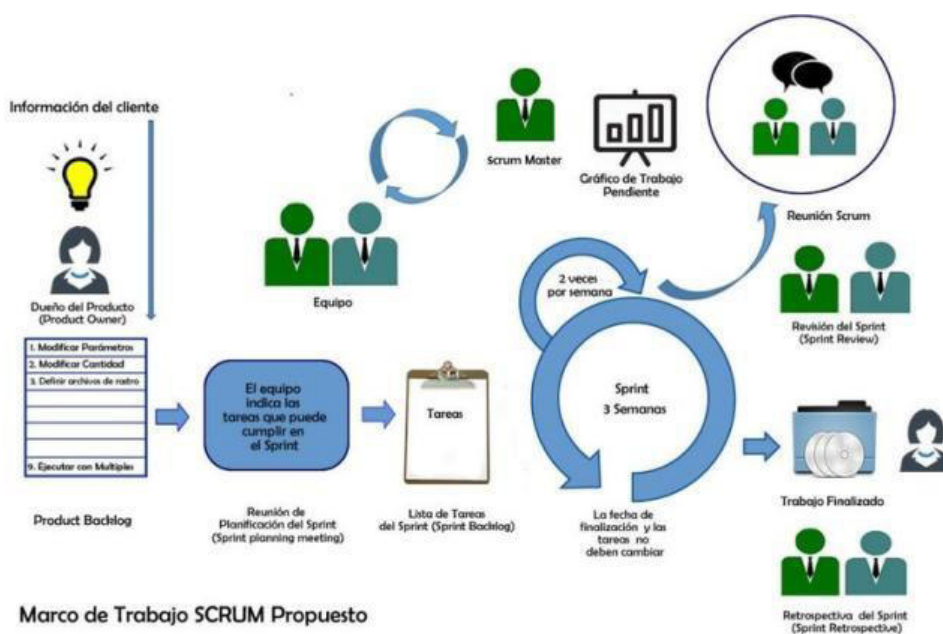


Figura 8. El marco de trabajo Scrum [28].

Origen

Ken Schwaber y Jeff Sutherland fueron los primeros en publicar un trabajo de investigación acerca de Scrum en la OOPSLA de 1995, pero las primeras referencias a esta metodología ágil se pueden encontrar en el artículo “*The New Product Development Game*” publicado por Takeuchi y Nonaka en el año 1986 [29]. A partir del año 1995 Ken y Jeff empezaron a producir varias producciones de Scrum, las cuales sirven como instructivo para el desarrollo de productos y servicios, entre las publicaciones sobre esta metodología ágil podemos encontrar: *Agile Software Development with Scrum*, *Agile Project Management with Scrum* y *The Scrum Guide* [29].

1.5.1 Equipo Scrum (**Scrum Team**)

El equipo Scrum estará conformado por varias personas, entre las cuales estarán asignados tres roles existentes: el Dueño del Producto (Product Owner), el Equipo de Desarrollo (Development Team) y el Scrum Master, además cada miembro deben cumplir con 2 características fundamentales: ser auto organizado, es decir ser capaz de realizar su trabajo sin necesidad de una guía externa y multifuncional al utilizar las capacidades de las personas para crear un incremento en el producto [27].

- Dueño de Producto (Product Owner): Es aquella persona encargada de tomar las decisiones y conoce la visión del producto, así como el negocio del cliente, de tal forma que pueda traducir las ideas del cliente en tareas con distintos niveles de prioridad, la mismas que posteriormente serán colocadas en el producto backlog [30].
- Scrum Master: Es la persona encargada de alentar y ofrecer apoyo al equipo de desarrollo, entre sus funciones esta facilitar la comprensión de las reglas y buenas prácticas recomendadas por Scrum, comprobar que el modelo y metodología funcionan y ayuda al equipo a resolver cualquier inconveniente que surja durante el desarrollo [30].
- Equipo de Desarrollo (Development Team): Son los profesionales encargados de realizar las tareas necesarias para obtener un incremento del producto funcional, este debe trabajar en conjunto sin distinciones de rangos o sube equipos, teniendo en cuenta que la responsabilidad de cada incremento recae sobre cada uno de los miembros [30].

1.5.2 Artefactos de Scrum

Son instrumentos que proporcionan los medios necesarios para maximizar la transparencia y oportunidades en cuanto a inspección y adaptación. Scrum utiliza los siguientes artefactos [27]:

- Lista de Producto (Product Backlog): Se enlistan todas las tareas y funcionalidades basadas en los requerimientos del cliente, organizada por prioridad y con capacidad de adaptación a nuevos requerimientos que surjan durante el desarrollo.
- Lista de Pendientes del Sprint (Sprint Backlog): Corresponde al conjunto de elementos seleccionados de la lista de productos, los cuales se resolverán en el sprint asignado, esta lista es una predicción con respecto a que funcionalidad será parte del siguiente incremento.
- Incremento: Es el resultado alcanzado al final de cada sprint, cada incremento se suma al valor de los incrementos de todos los sprints anteriores; para que un incremento se defina como “terminado” es necesario que el entregable cumpla con los requisitos de utilidad.

1.5.3 Eventos de Scrum

Los eventos de Scrum se pueden considerar bloques de tiempo con actividades predefinidas, los cuales fueron diseñados con el fin de evitar el surgimiento de reuniones innecesarias y fuera de horarios planificados pudiendo retrasar el avance del proyecto.

- El Sprint: Este evento es considerado como el “corazón” de Scrum ya que durante este periodo de tiempo de un mes o de una duración definida por el equipo, se realizan los incrementos necesarios para una versión utilizable y potencialmente desplegable que sea capaz de otorgarle valor al cliente [30].
- Planificación del Sprint (*Sprint Planning*): Consiste en una reunión con una duración aproximada de ocho horas, con el fin de establecer el objetivo principal y planificar las tareas que se deben llevar a cabo durante la ejecución del sprint, acciones que deberá cumplirse hasta finalizar el sprint [30]. La persona encargada de llevar a cabo esta reunión es el Scrum Master, el cual tiene la misión de capacitar al resto del equipo Scrum para que sean capaces de mantenerse dentro de los horarios establecidos [27].
- Scrum Diario (*Daily Scrum*): Esta reunión se realiza diariamente entre los miembros del equipo de desarrollo, se estima que dure un máximo de 15 minutos y en ella se

debe llegar a un acuerdo acerca del trabajo que se realizará durante todo el día. Este evento se realiza con el fin de favorecer la cooperación y el trabajo en equipo organizado, revisando cuales fueron los avances alcanzados hasta la fecha y planificando el avance diario para el incremento que se debe presentar al final del sprint [27].

- **Revisión del Sprint (*Sprint Review*):** Ocurre tras haber finalizado un sprint, su objetivo es analizar el incremento que se ha desarrollado y en caso de ser necesario, adaptar la lista del producto. Esta reunión es de carácter informal y se estima que tenga una duración máxima de cuatro horas, durante su transcurso es conveniente que el equipo Scrum colabore para determinar las actividades a realizar a continuación y analizar de qué manera se puede optimizar el valor del próximo incremento [27].
- **Retrospectiva de Sprint (*Sprint Retrospective*):** Esta reunión tiene lugar tras haber concluido la revisión del sprint y antes de que se realice la siguiente planificación del sprint, dura un máximo de tres horas, y el propósito es el determinar los eventos que ocurrieron durante el transcurso del sprint, tales como las relaciones entre el personal, los procesos, la creación e implementación de planes de mejora para el siguiente sprint [27].

1.5.4 Sprint 0

El sprint 0 es considerado un evento extra, el cual se realiza al inicio del proyecto cuyo fin es determinar su alcance y definir los roles del equipo scrum, la creación de un Product Backlog inicial y establecer una infraestructura base en caso de ser necesario. Sin embargo, este evento no consta en la documentación oficial de Scrum, su nombre no es el más adecuado por cuanto no cumple con las actividades y requisitos de un sprint común como lo es la generación de un incremento o la realización de reuniones diarias de Scrum [31].

2 METODOLOGÍA

2.1 Sprint 0

El principal objetivo de este sprint fue el definir la arquitectura de la aplicación, los roles que tomara cada integrante del proyecto y establecer las historias épicas e historias de usuario iniciales. El “sprint 0” fue desarrollado en el transcurso de 1 mes y medio, durante este periodo se recopiló toda información pertinente para empezar el proyecto, se realizó la investigación de las herramientas necesarias para la implementación de RA en dispositivos móviles, se planificó la arquitectura global, se asignaron los roles del equipo Scrum, las historias épicas, el Product Backlog y, por último, el número de sprints con su respectiva duración.

2.1.1 Arquitectura

La representación del aplicativo de RA puede ser apreciado en el flujo de la Figura 9. El proceso comienza al obtener los POIs según las preferencias turísticas del usuario y sus días de visita. Los POIs son adquiridos del servicio web desarrollado en el trabajo de titulación previo “Desarrollo de un Algoritmo Híbrido para recomendaciones de Itinerarios Turísticos de acuerdo con las preferencias del usuario” mediante el ingreso de datos en un formulario inicial; luego se realiza el proceso de geolocalización mediante el cual se calcula la ubicación para cada punto de interés (POI) en el plano virtual basándose en su latitud y longitud, este proceso es posible gracias a la aplicación de la fórmula de Haversine, finalmente se le asigna un ícono 3D específico con respecto al tipo de POI y se renderiza en el escenario.

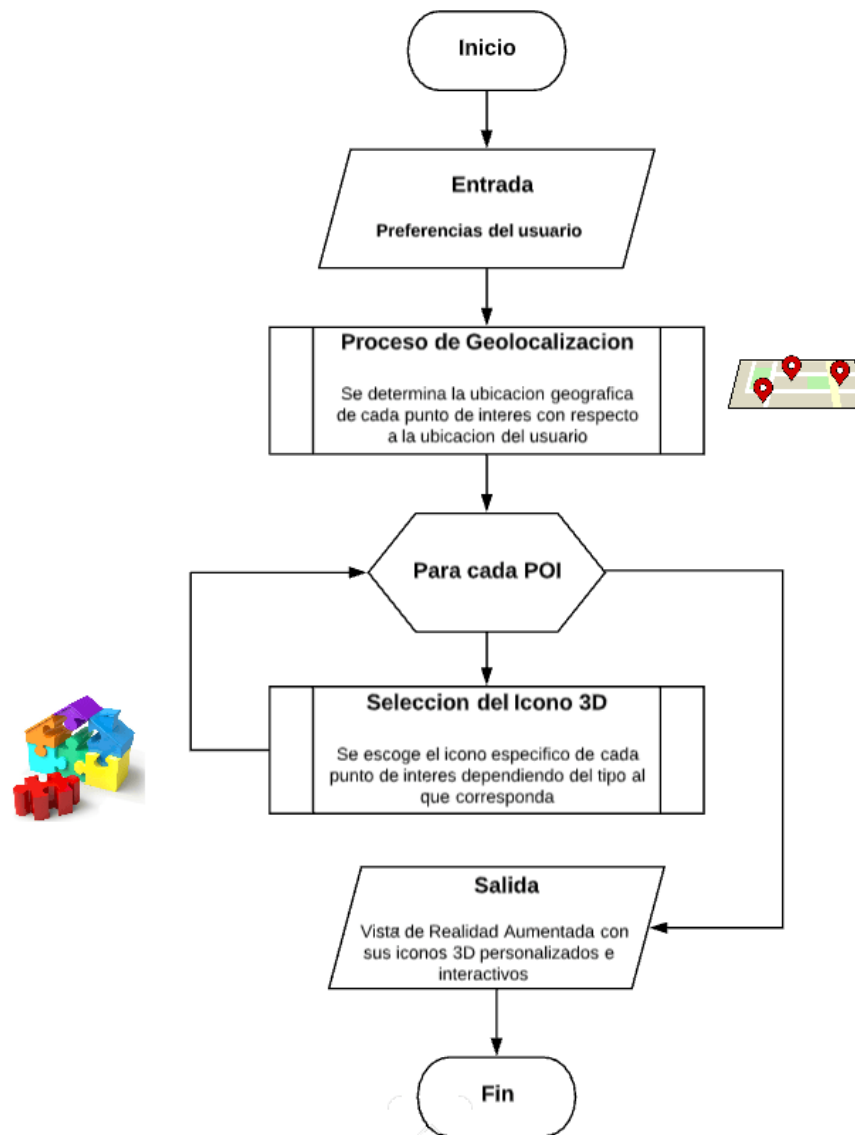


Figura 9. Diagrama de flujo para la vista de RA.

Para la infraestructura global del proyecto se ha Implementado la Arquitectura de Capas como se muestra en la Figura 10. Dentro de la primera capa se ha colocado el servicio AWS, del cual se obtiene la información del itinerario de POIs más cercanos al usuario; en la segunda capa se define la vista de RA, la cual accede a la cámara trasera del dispositivo del usuario y muestra los POIs como íconos en 3D sobrepuestos a su imagen real; para la tercera capa se realiza una integración con Mapbox, para una vista aérea del mapa de la zona mostrando los POIs cercanos. Se puede encontrar los modelos iniciales de las vistas de la aplicación o mockups adjuntos en el Anexo VI.

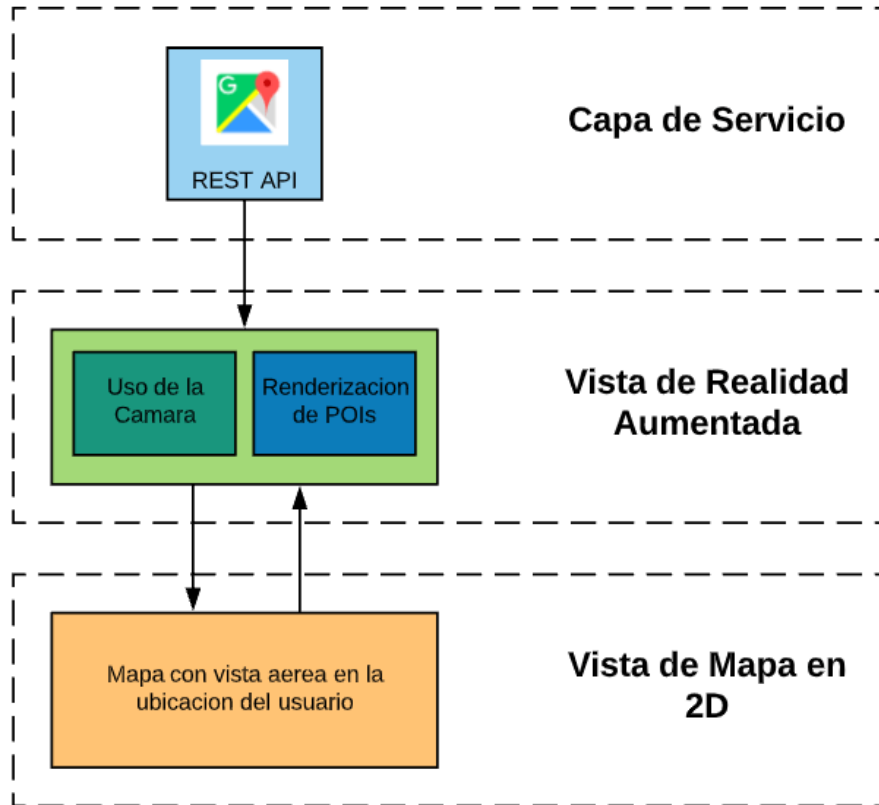


Figura 10. Arquitectura de capas del proyecto.

2.1.2 Definición de roles

El proyecto de titulación está conformado por tres integrantes, cada uno cumplirá un rol específico (ver Tabla 1). El número de integrantes permitió sin mayor inconveniente el cumplimiento de los objetivos y en el tiempo establecido del proyecto.

Tabla 1. Roles del equipo Scrum en el proyecto

Equipo Scrum	
Rol	Responsable
<i>Product Owner</i>	Maritzol Tenemaza
<i>Scrum Master</i>	Maritzol Tenemaza
<i>Development Team</i>	Renato Burbano y Rafael Vinueza

2.1.3 Historias Épicas

Se puede apreciar las historias épicas identificadas para el desarrollo de todo el proyecto como indica en la Tabla 2 y el detalle de cada historia épica se puede visualizar en el Anexo I.

Tabla 2. Historias épicas

CÓDIGO	TÍTULO	PRIORIDAD
AH1	Desarrollo de la vista de Realidad Aumentada.	Alta
AH2	Desarrollo de un mapa con vista aérea.	Media
AH3	Publicación del servicio en una plataforma de servicios en la nube y conexión con el aplicativo.	Baja

2.1.4 Product Backlog

Se puede apreciar el primer *product backlog* creado en el sprint 0 en la Tabla 3.

Tabla 3. *Product Backlog* inicial

PRODUCT BACKLOG				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Estimación (días)	Prioridad
AH1	AH1-01	Uso de una herramienta de software que implemente RA.	5	Alta

	AH1-02	Desarrollo de funciones de geolocalización para los POIs.	10	Alta
AH2	AH2-01	Integración de herramienta de software para la representación de mapas.	5	Media
	AH2-02	Desarrollo de las funcionalidades requeridas en la vista de mapa.	15	Media
AH3	AH3-01	Publicación del proyecto de titulación predecesor en una plataforma de servicios en la nube.	5	Baja
	AH3-02	Desarrollo de las funciones para el consumo del servicio web.	5	Baja

2.1.5 Definición de *sprints*

Una vez definidas un total de 6 historias de usuario con sus respectivas aproximaciones temporales, se planeaba realizar las tareas definidas previamente en un estimado de 2 meses. Y con ello la decisión de desarrollar el proyecto de titulación en un total de 4 sprints, donde cada uno tendrá una duración de 2 semanas.

2.2 Sprint 1

2.2.1 Sprint Planning

Objetivo del Sprint

Desarrollar un primer acercamiento a la vista de RA y posicionar los POIs de forma virtual, transformando coordenadas geográficas a posicionamiento virtual.

Historias de Usuario

Se puede encontrar desglosadas las historias de usuario destinadas al primer sprint en la Tabla 5 al igual que su tiempo estimado. Para mayor información consultar el Anexo II donde se encuentra en detalle de cada historia de usuario.

Tabla 4. Historias de usuario escogidas para el Sprint 1

ID	Título	Estimación (días)	Estado
AH1-01	Uso de una herramienta de software que implemente RA.	5	Por implementar
AH1-02	Desarrollo de funciones de geolocalización para los POIs.	10	Por implementar

Sprint Backlog

Tabla 5. Sprint backlog del Sprint 1

SPRINT BACKLOG	
HISTORIA DE USUARIO	TAREAS
AH1-01	Crear proyecto en Unity.
	Importar librerías de Vuforia a Unity.
	Crear una clase para gestionar los POIs.
	Desplegar un objeto inicial para ejecutar pruebas con Vuforia.
	Crear y publicar de proyecto en un repositorio de Github.
AH1-02	Desarrollar funciones para el uso de geolocalización en conjunto con el dispositivo móvil.
	Ejecutar pruebas para comprobar la obtención y funcionamiento de coordenadas geográficas.
	Desarrollar funciones para el uso de brújula y acelerador en conjunto con el dispositivo móvil.
	Ejecutar pruebas para comprobar la obtención y funcionamiento de la orientación de la brújula y su acelerador.
	Desplegar objetos transformando coordenadas geográficas con la fórmula de Haversine.
	Ejecutar pruebas de campo para comprobar la correcta ubicación de objetos virtuales en el entorno real.

2.2.2 Ejecución del Sprint

A lo largo de la ejecución del Sprint han surgido detalles importantes que se los describe a continuación:

Librería Vuforia

Para el desarrollo del proyecto se usó la versión de Unity 2019.1.0f2, para el uso de la librería es necesario desplegar la cámara de RA 'ARCamera', al igual que la Figura 11, la cual permite integrar elementos de Unity en el escenario real, además, como se puede ver

en la Figura 12 para el uso de la librería se requiere una llave de licencia de desarrollo otorgada por Vuforia en su página de desarrolladores [32].

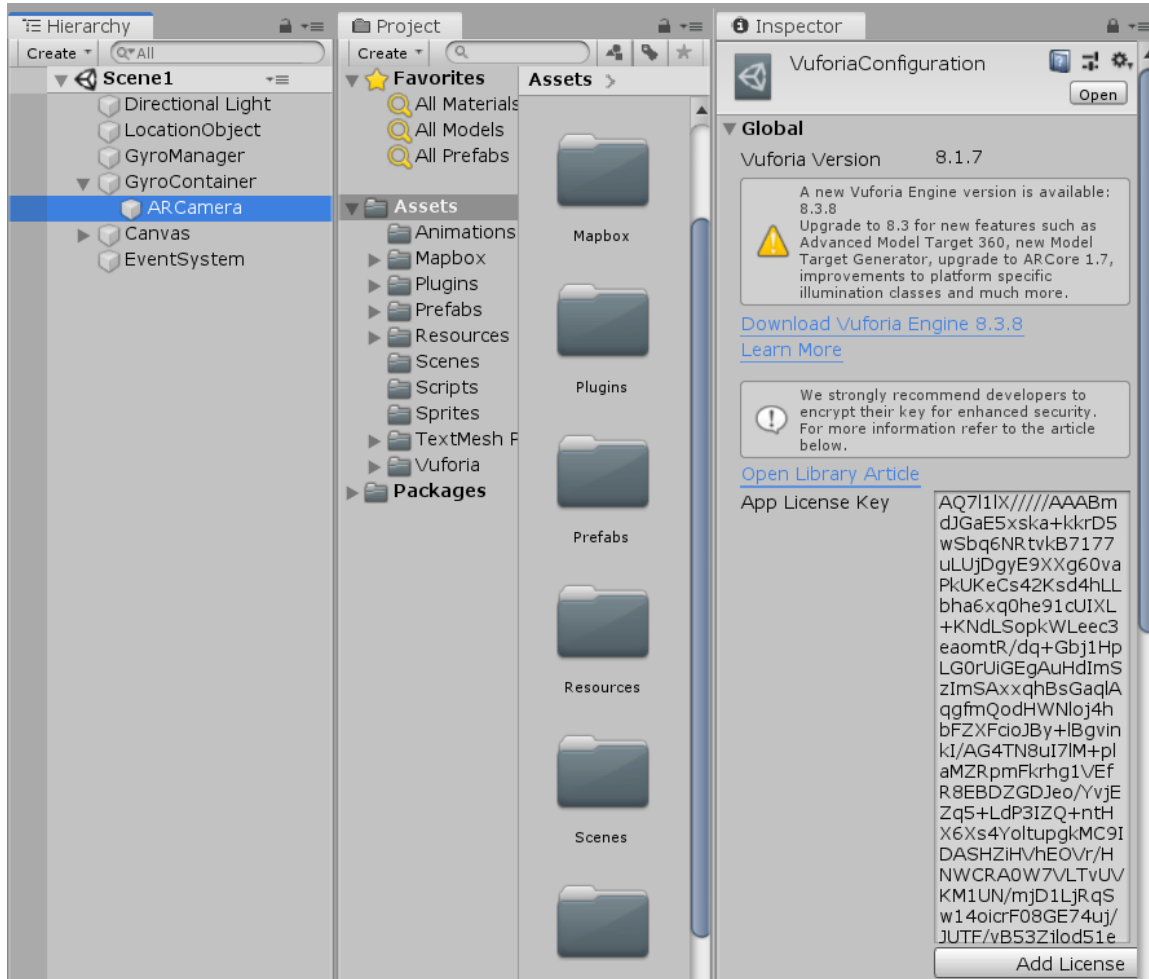


Figura 11. Cámara de RA proporcionada por Vuforia y su llave de licencia

[Back To License Manager](#)

Add a free Development License Key

License Name *

You can change this later

License Key

Develop

Price: No Charge

Reco Usage: 1,000 per month

Cloud Targets: 1,000

VuMark Templates: 1 Active

VuMarks: 100

By checking this box, I acknowledge that this license key is subject to the terms and condition

Cancel

Confirm

Figura 12. Creación de una nueva llave de licencia de Vuforia en la página de desarrolladores

Precisión de datos de GPS, Brújula y acelerador

Para el uso del GPS, brújula y el acelerador del dispositivo móvil del usuario es necesario obtener los debidos permisos de autorización de uso en la función Awake de la clase GPS, como se muestra en la Figura 13. Unity integra la clase Input.location que con los respectivos permisos es capaz de extraer los datos de geolocalización tales como la latitud y longitud del usuario, esto se representa en la Figura 14.

```

private void Awake()
{
    if (!Permission.HasUserAuthorizedPermission(Permission.FineLocation))
    {
        Permission.RequestUserPermission(Permission.FineLocation);
    }

    Instance = this;
    DontDestroyOnLoad(gameObject);
    StartCoroutine(GPSLocation());
}

```

Figura 13. Solicitud de permisos para el uso de GPS del dispositivo móvil

```

private void UpdateGPSData()
{
    if (Input.location.status == LocationServiceStatus.Running)
    {
        latitud = Input.location.lastData.latitude;
        longitud = Input.location.lastData.longitude;
    }
}

```

Figura 14. Adquisición de los últimos valores de latitud y longitud del dispositivo móvil del usuario

Respecto a la brújula se ha hecho uso de la librería `Input.compass` integrada por Unity al igual que el GPS, se han utilizado las funcionalidades de esta librería para obtener la dirección en grados a la que apunta el usuario y de esta manera colocar correctamente los POIs respecto a los puntos cardinales. Este valor se recalcula por cada frame en ejecución como se observa en la Figura 15.

```

void Update()
{
    if (startTracking)
    {
        transform.rotation = Quaternion.Euler(0, Input.compass.trueHeading, 0);
        grados.text = ((int)Input.compass.trueHeading).ToString()
            + "° " + DegreesToCardinalDetailed(Input.compass.trueHeading);
    }
}

IEnumerator InitializeCompass()
{
    yield return new WaitForSeconds(1f);
    startTracking |= Input.compass.enabled;
}

private static string DegreesToCardinalDetailed(double degrees)
{
    string[] caridnals = { "N", "NNE", "NE", "ENE", "E", "ESE",
        "SE", "SSE", "S", "SSW", "SW", "WSW", "W", "WNW", "NW", "NNW", "N" };
    return caridnals[(int)Math.Round(((double)degrees * 10 % 360) / 225)];
}

```

Figura 15. Obtención de datos cardinales de la brújula del dispositivo móvil

Adicionalmente, se obtuvieron los valores de aceleración proporcionados por el giroscopio del dispositivo móvil, los cuales se utilizaron para permitir que la cámara de RA pueda rotar en conjunto con el movimiento del celular del usuario. Se desarrolló la clase GyroManager para obtener dichos valores como se observa en la Figura 16. Para su complemento se creó la clase FollowGyro en la cual se asigna el valor de desplazamiento rotacional que deberá tener la cámara dependiendo de los valores de aceleración obtenidos en la clase GyroManager esto se puede apreciar en la Figura 17.

```

private void Update()
{
    if (gyroActivate)
    {
        rotation = gyro.attitude;
    }
}

public Quaternion getGyroRotation()
{
    return rotation;
}

public void enableGyro()
{
    if (gyroActivate)
        return;

    if (SystemInfo.supportsGyroscope)
    {
        gyro = Input.gyro;
        gyro.enabled = true;
        gyroActivate = gyro.enabled;
    }
}

```

Figura 16. Adquisición de valores del acelerador en la clase GyroManager

```

private Quaternion baseRotation = new Quaternion(0, 0, 1, 0);

void Start()
{
    GyroManager.Instance.enableGyro();
}

void Update()
{
    transform.localRotation = GyroManager.Instance.getGyroRotation() * baseRotation;
}

```

Figura 17. Asignación de valores del giroscopio para rotación de la cámara

Fórmula de Haversine

Para ubicar los POIs desde el escenario virtual en unidades de Unity a medidas reales en metros se implementó la fórmula de Haversine en código C#. Esta función requiere la latitud

y longitud de dos puntos, para retornar el valor de distancia en metros entre ellos, como se puede ver en la Figura 18.

```
public float FormulaHaversine(float lat1, float long1, float lat2, float long2)
{
    float earthRad = 6371000;
    float lRad1 = lat1 * Mathf.Deg2Rad;
    float lRad2 = lat2 * Mathf.Deg2Rad;
    float dLat = (lat2 - lat1) * Mathf.Deg2Rad;
    float dLong = (long2 - long1) * Mathf.Deg2Rad;
    float a = Mathf.Sin(dLat / 2.0f) * Mathf.Sin(dLat / 2.0f) +
        Mathf.Cos(lRad1) * Mathf.Cos(lRad2) *
        Mathf.Sin(dLong / 2.0f) * Mathf.Sin(dLong / 2.0f);
    float c = 2 * Mathf.Atan2(Mathf.Sqrt(a), Mathf.Sqrt(1 - a));
    return earthRad * c;
}
```

Figura 18. Representación de la fórmula de Haversine en código C#

Ubicación de elementos 3D en escena

Para colocar los objetos que en este caso son cubos de Unity, se desarrolló una función para determinar la posición virtual de cada elemento dependiendo de la latitud y longitud de un POI seleccionado. Esta transformación se determina al establecer equivalencia de la latitud a la coordenada z, mientras que longitud corresponde a la coordenada x. Siguiendo este concepto, la función EncontrarCuadrante permite triangular la posición del POI con respecto a la posición del usuario esto mediante el método triangular. Este puede ser representado en código C# en la Figura 19.

```
public double[,] EncontrarCuadrante(float lat1, float long1, List<PointOfInterest> places)
{
    float lat3 = lat1;
    double[,] ubicacionesVirtuales = new double[places.Count, 2];

    for (int i = 0; i < places.Count; i++)
    {
        if (lat1 <= float.Parse(places[i].location.lat, CultureInfo.InvariantCulture.NumberFormat) &&
            long1 <= float.Parse(places[i].location.lng, CultureInfo.InvariantCulture.NumberFormat))
        {
            ubicacionesVirtuales[i, 0] = TransformarMetrosUnidades(FormulaHaversine((float)miLat,
                (float)miLong, lat3, float.Parse(places[i].location.lng, CultureInfo.InvariantCulture.NumberFormat)));
            ubicacionesVirtuales[i, 1] = TransformarMetrosUnidades(FormulaHaversine(float.Parse(places[i].location.lat,
                CultureInfo.InvariantCulture.NumberFormat), float.Parse(places[i].location.lng,
                CultureInfo.InvariantCulture.NumberFormat), lat3, float.Parse(places[i].location.lng,
                CultureInfo.InvariantCulture.NumberFormat)));
        }
        else if (lat1 > float.Parse(places[i].location.lat, CultureInfo.InvariantCulture.NumberFormat) &&
            long1 <= float.Parse(places[i].location.lng, CultureInfo.InvariantCulture.NumberFormat))
        {
            ubicacionesVirtuales[i, 0] = TransformarMetrosUnidades(FormulaHaversine((float)miLat,
                (float)miLong, lat3, float.Parse(places[i].location.lng, CultureInfo.InvariantCulture.NumberFormat)));
            ubicacionesVirtuales[i, 1] = TransformarMetrosUnidades(FormulaHaversine(float.Parse(places[i].location.lat,
                CultureInfo.InvariantCulture.NumberFormat), float.Parse(places[i].location.lng,
                CultureInfo.InvariantCulture.NumberFormat), lat3, float.Parse(places[i].location.lng,
                CultureInfo.InvariantCulture.NumberFormat))) * -1;
        }
    }
}
```

Figura 19. Función EncontrarCuadrante para colocar elementos en el escenario

2.2.3 Sprint Review

El objetivo del primer sprint se cumplió en su totalidad, sin embargo, se presentó cierta dificultad en la comprensión de los términos de GPS, brújula y acelerador del dispositivo móvil y su implementación con la herramienta Unity.

Pruebas de aceptación

Tras comprobar el trabajo efectuado en el primer sprint como indica la Tabla 6, se puede visualizar que se cumplieron todos los criterios de aceptación de las historias de usuario.

Tabla 6. Pruebas de aceptación del Sprint 1

Historia de Usuario	Criterio de aceptación	Cumplido
AH1-01	Cuando el usuario abra la vista de RA con la cámara de su dispositivo móvil, entonces se mostrará los POIs correspondientes a su itinerario.	Si
	Cuando la cámara del teléfono este apuntando a la ubicación del POI, entonces el icono 3D será visible en pantalla.	Si
AH1-02	Cuando el usuario entre al aplicativo, entonces la aplicación obtendrá los datos de ubicación y orientación del dispositivo móvil.	Si
	Cuando el usuario se encuentre en la vista de RA apuntando a un POI, entonces la aplicación deberá ser capaz de presentar un icono en 3D a una distancia real en metros.	Si
	Cuando el usuario se encuentre en el rango de aparición de un POI (70 metros), entonces el icono en 3D se ubicará lo más cercano posible a su posición real.	Si

Incremento obtenido

Obtención de valores de ubicación y orientación

Se visualiza el resultado al desplegar los valores de geolocalización obtenidos, además se observa cómo se obtienen estos valores directamente del dispositivo móvil del usuario, siempre y cuando este haya activado la opción de ubicación, ver Figura 20.

GPS Location	
GPS Status:	Running
Latitud:	0.7977225
Longitud:	-77.73058
Altitud	2972.9
Horizontal Acc:	14.274
TimeStamp:	1634057641.21 9

Figura 20. Valores de geolocalización obtenidos de la aplicación

Se observan el valor obtenido de la brújula del dispositivo móvil del usuario, del mismo modo obtenidos tras haber ejecutado la aplicación en un dispositivo móvil con la opción de ubicación activada, ver Figura 21.



Figura 21. Valor de orientación obtenido de la aplicación

Ubicación de elementos de RA en el escenario

Se muestra como se ha realizado la primera aproximación para ubicar elementos de RA en escenarios reales, como se ve en la Figura 22, cabe indicar que los valores obtenidos del GPS, brújula tienen cierto desfase, lo cual hace que la colocación de objetos no sea un 100% precisa, esto ocurre debido a que los dispositivos móviles actuales no son capaces de medir con exactitud estos valores.



Figura 22. Primera aproximación para la ubicación de un objeto en 3D en escena

Adaptación del Product Backlog

Tras ejecutar un análisis del incremento obtenido y evaluar el trabajo que se ha realizado, se llegó a la conclusión de que se han completado las actividades planificadas para el sprint satisfactoriamente, pero se ha detectado que la historia épica AH1 requiere de más historias de usuario en posteriores sprints. Con ello se desarrolló la nueva historia de usuario AH1-04, además de la actualización del Product Backlog con las historias de usuario terminadas y las por implementar, como indica la Tabla 7.

Tabla 7. Product Backlog adaptado tras finalizar el Sprint 1

PRODUCT BACKLOG				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Estimación (días)	Estado
AH1	AH1-01	Uso de una herramienta de software que implemente RA.	5	Terminado

	AH1-02	Desarrollo de funciones de geolocalización para los POIs.	10	Terminado
	AH1-03	Representación de POIs mediante modelos 3D.	3	Por Implementar
	AH1-04	Despliegue de un panel que muestre la información de los POIs.	3	Por Implementar
AH2	AH2-01	Integración de herramienta de software para la representación de mapas.	5	Por Implementar
	AH2-02	Desarrollo de las funcionalidades requeridas en la vista de mapa.	15	Por Implementar
AH3	AH3-01	Publicación del proyecto de titulación predecesor en una plataforma de servicios en la nube.	5	Por Implementar
	AH3-02	Desarrollo de las funciones para el consumo del servicio web.	5	Por Implementar

2.2.4 Retrospectiva y velocidad del equipo

Durante el desarrollo del sprint se puede señalar que en los primeros días existió retraso con respecto a las tareas, por cuanto fue necesario acostumbrarse a las nuevas herramientas, situación que se contrasta con los días finales en los que se desarrollaron el mayor número de tareas, de esta forma se pudo completar con éxito la planificación del

sprint. Se visualiza el gráfico de Burndown donde se presenta el progreso del proyecto durante el primer sprint, lo observamos en la Figura 23.

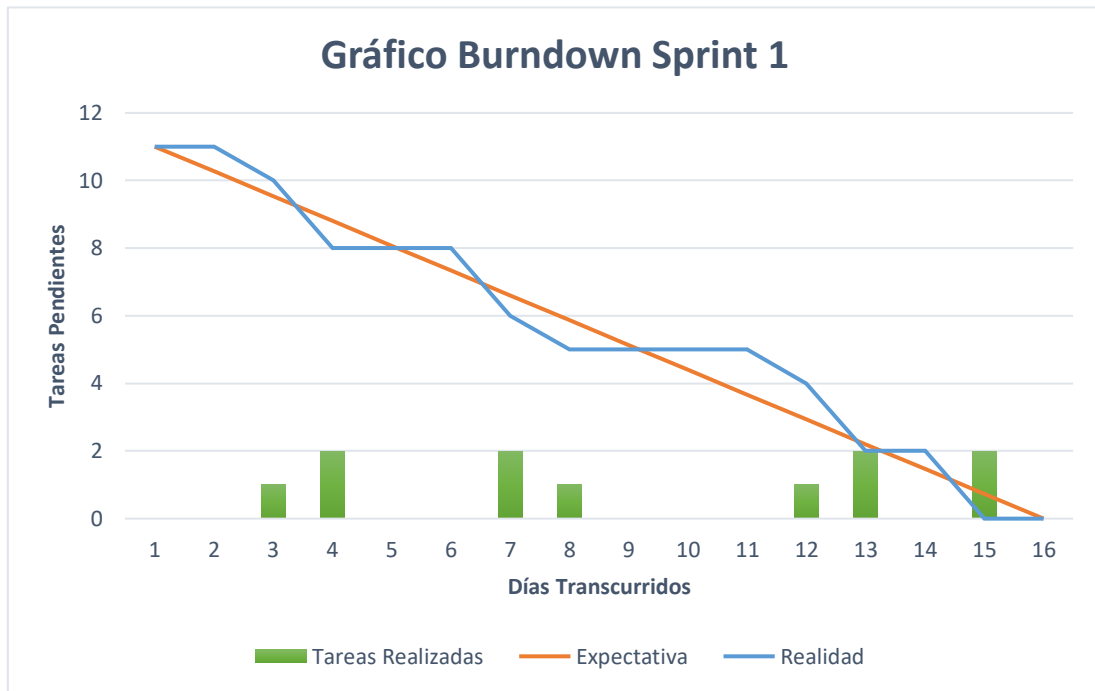


Figura 23. Gráfico de Burndown del Sprint 1

2.3 Sprint 2

2.3.1 Sprint Planning

Objetivo del Sprint

Mejorar el apartado visual para representación de los POIs a través de iconos 3D y despliegue de información relacionada a los mismos.

Historias de Usuario

Tabla 8. Historias de usuario escogidas para el Sprint 2 (Sus detalles se encuentran en el Anexo III)

ID	Título	Estimación (días)	Estado
AH1-03	Representación de POIs mediante modelos 3D.	3	Por implementar
AH1-04	Despliegue de un panel que muestre la información de los POIs.	3	Por implementar

Sprint Backlog

Tabla 9. Sprint Backlog del Sprint 2

SPRINT BACKLOG	
HISTORIA DE USUARIO	TAREAS
AH1-03	Selección de herramienta gratuita para creación de modelos 3D que posea integración con Unity
	Búsqueda de recursos para desarrollo de modelos de POIs dependiendo su tipo.
	Exportación y despliegue de modelos al entorno de Unity.
	Definir las propiedades como la escala y ubicación de cada modelo en el entorno de Unity.
	Pruebas de campo para comprobar la correcta representación de los modelos creados para los POIs.

AH1-04	Desarrollo del panel de información para los POIs utilizando las herramientas de diseño de Unity.
	Desarrollo de funcionalidades para la correcta interacción entre el panel informativo y el usuario.
	Integración del panel informativo en cada modelo 3D.
	Pruebas para comprobar el correcto funcionamiento y despliegue de la información dentro del panel informativo

2.3.2 Ejecución del Sprint

Durante el segundo *sprint* se pueden destacar los siguientes puntos:

Modelado de iconos en Blender

Para la producción de los modelos 3D que representan a cada tipo de POI se ha optado por la herramienta Blender gracias a su buena integración con Unity. Se ha creado el modelo de un tipo de POI mediante la incorporación de distintos recursos obtenidos en Blender y en otras páginas web, como muestra la Figura 24 [33].

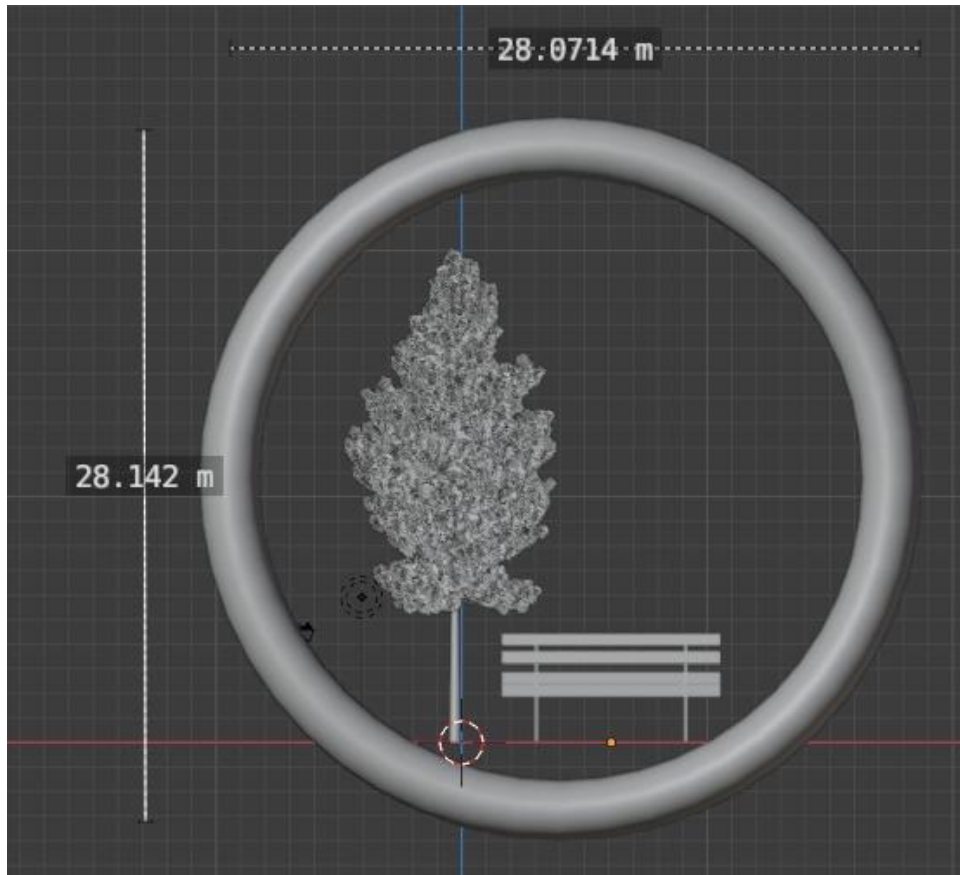


Figura 24. Modelo para el POI de tipo parque creado en Blender.

Clase TouchEvent

En la clase TouchEvent se han implementado varias librerías que permiten al usuario controlar el pulso en la pantalla de su dispositivo móvil; la librería Input que permite detectar si el usuario ha tocado la pantalla una o varias veces; la librería Ray que permite instanciar y lanzar un rayo invisible en la dirección que el usuario ha realizado el toque y la librería RaycastHit que permite detectar el objeto que ha colisionado con el rayo lanzado previamente y desplegar el panel de información. El código necesario para que exista la interacción con el usuario se muestra en la Figura 25.


```

void Update()
{
    if (Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began)
    {
        Ray ray = Camera.main.ScreenPointToRay(Input.touches[0].position);
        RaycastHit hit;

        if (Physics.Raycast(ray, out hit))
        {
            if (hit.collider != null)
            {
                poiData = hit.collider.GetComponent<POIData>();
                if (poiData != null)
                {

```

Figura 25. Uso de las librerías para detectar el toque en el dispositivo móvil.

Ajuste del tamaño de los íconos 3D

Para controlar el tamaño de los iconos que se despliegan en la pantalla, se ha desarrollado la función `CalcularEscala`, la cual permite adaptar la escala de un objeto dependiendo de la distancia a la que se encuentre, de esta manera el usuario tendrá la percepción adecuada, por cuanto al acercarse visualmente al ícono, este permanece a una escala visible como se muestra en la Figura 26.

```

public Vector3 calcularEscala(float distancia)
{
    //float valor = (distancia) / 60;
    float valor = 8;

    if (distancia < 60 && distancia > 30)
        valor = 6;
    else if (distancia <= 30)
        valor = 3;

    return new Vector3(valor, valor, valor);
}

```

Figura 26. Función para calcular la escala de los modelos 3D.

2.3.3 Sprint Review

Pruebas de aceptación

Se enlista los criterios de aceptación cumplidos para el segundo sprint. En este caso, se completó las tareas planteadas y a la vez se identificó que debe existir una mejora con respecto a los iconos y su escalado, como se muestra en la Tabla 10

Tabla 10. Pruebas de aceptación del Sprint 2

Historia de Usuario	Criterio de aceptación	Cumplido
AH1-03	Cuando el usuario se encuentre en el rango de aparición de un POI (70 metros), entonces los iconos 3D serán representados con colores llamativos para su distinción.	SÍ
	Cuando el usuario se encuentre en la vista de RA apuntando a un POI, entonces el icono 3D se desplegará a un tamaño visible para el usuario.	SÍ
	Cuando el usuario se encuentre en el rango de aparición de un POI (70 metros), entonces los iconos 3D se representarán con un símbolo relacionado al tipo de POI al que representan.	SÍ
AH1-04	Cuando el usuario toque el icono 3D de un POI, entonces se desplegarán los datos referentes a este POI en un panel informativo.	NO
	Cuando el usuario toque el icono 3D de un POI mientras su panel informativo está abierto, entonces el panel informativo se ocultará.	SÍ

Incremento obtenido

Representación de iconos en RA

Es el primer acercamiento para la representación de POIs con iconos utilizando RA, se realizó una prueba de campo para comprobar la correcta ubicación de los iconos en un escenario real al igual que su tamaño, como se muestra en la Figura 27.



Figura 27. Comparación de tamaño entre los iconos a distintas distancias

Se visualiza el panel informativo después de haber pulsado el icono del POI (ver Figura 28) y se observan datos grabados con la información más relevante del POI como son:

- Nombre
- Horario de atención
- Tipo de POI



Figura 28. Despliegue del panel de información con datos grabados.

Adaptación del Product Backlog

Una vez terminado el desarrollo de las historias de usuario del sprint y su completitud, se identificó que la historia épica AH1 aún requiere de historias de usuario para completarse, esto debido a que se están utilizando datos grabados en el panel de información, el equipo de trabajo consideró la necesidad de crear una vista para la recolección de información turística del usuario e integrar el servicio desarrollado en el proyecto de titulación antecesor para que retorne los POIs en base a las preferencias del usuario. Por los motivos anteriormente listados, se procedió a adaptar al Product Backlog como se enseña en la Tabla 11.

Tabla 11. Product Backlog adaptado tras finalizar el Sprint 2

PRODUCT BACKLOG				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Estimación (días)	Estado
AH1	AH1-01	Uso de una herramienta de software que implemente RA.	5	Terminado
	AH1-02	Desarrollo de funciones de geolocalización para los POIs.	10	Terminado
	AH1-03	Representación de POIs mediante modelos 3D.	3	Terminado
	AH1-04	Despliegue de un panel que muestre la información de los POIs.	3	Terminado
	AH1-05	Recolección de información de preferencias turísticas mediante un formulario.	2	Por Implementar
AH2	AH2-01	Integración de herramienta de software para la representación de mapas.	5	Por Implementar

	AH2-02	Desarrollo de las funcionalidades requeridas en la vista de mapa.	15	Por Implementar
AH3	AH3-01	Publicación del proyecto de titulación predecesor en una plataforma de servicios en la nube.	5	Por Implementar
	AH3-02	Desarrollo de las funciones para el consumo del servicio web.	5	Por Implementar

2.3.4 Retrospectiva y velocidad del equipo

En el gráfico burndown del segundo sprint se presenta un severo retraso de las tareas realizadas por día, debido a que surgieron complicaciones en el desarrollo de la vista de RA, en la cual se utilizó la fórmula de Haversine para la representación de los iconos en el escenario virtual, de esta forma el desarrollo de las tareas se distanció de lo planificado, postergando así algunas tareas para el siguiente sprint (ver Figura 29).

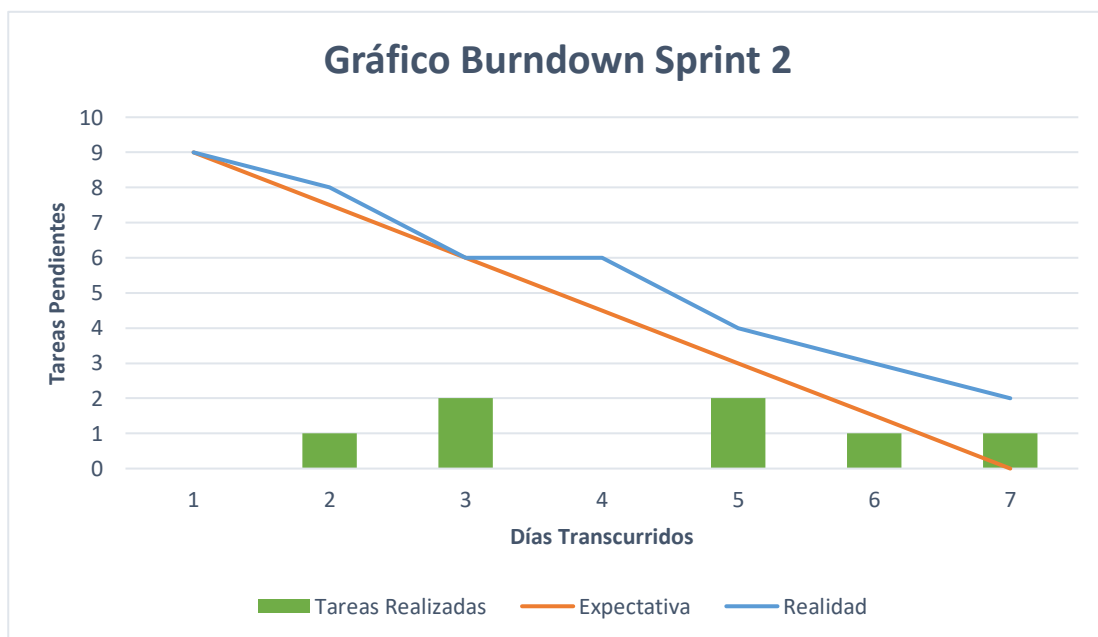


Figura 29. Gráfico de Burndown del Sprint 2.

2.4 Sprint 3

2.4.1 Sprint Planning

Objetivo del Sprint

Finalizar la funcionalidad de la primera capa dentro del aplicativo de RA e integrar los servicios de Mapbox y adquirir los POIs del servicio del proyecto de titulación antecesor.

Historias de Usuario

Las historias de usuario del sprint anterior que no se lograron finalizar se adjuntan con las historias de usuario planificadas para el sprint actual, como se muestra en la Tabla 13.

Tabla 12. Historias de usuario escogidas para el Sprint 3 (Sus detalles completos se encuentran en el Anexo IV)

ID	Título	Estimación (días)	Estado
AH1-05	Recolección de información de preferencias turísticas mediante un formulario.	2	Por implementar
AH2-01	Integración de herramienta de software para la representación de mapas.	5	Por Implementar
AH2-02	Desarrollo de las funcionalidades requeridas en la vista de mapa.	15	Por Implementar
AH3-02	Desarrollo de las funciones para el consumo del servicio web.	5	Por Implementar

Sprint Backlog

Tabla 13. Sprint backlog del Sprint 3

SPRINT BACKLOG	
HISTORIA DE USUARIO	TAREAS
AH1-05	Diseño de formulario adaptado a los requerimientos del servicio.
	Utilizar elementos que permitan el despliegue de formularios en Unity.
	Diseño e implementación de pantalla de carga durante la llamada al servicio.
	Pruebas para comprobar el correcto funcionamiento del formulario y la pantalla de carga.
AH2-01	Investigación de herramientas capaces de integrar mapas 2D con el entorno de Unity.
	Descargar e importar los recursos de Mapbox en la escena de Unity.
	Prueba unitaria para corroborar la integración de las librerías y mapbox dentro de Unity con sus funcionalidades.
AH2-02	Implementación del elemento map de Mapbox para desplegar una ubicación en el mapa dependiendo de la latitud y longitud del usuario.
	Implementación del elemento player de Mapbox para representar la ubicación del usuario con respecto a sus coordenadas geográficas.
	Adición de POIs con latitud y longitud fijas al mapa 2D.
	Creación de elementos de interfaz de usuario para el desplazamiento a través del mapa en 2D.
	Ajuste para obtener los POIs desde la capa de RA y su representación en el mapa 2D.
AH3-02	Descarga e instalación de herramientas requeridas para la ejecución local del servicio desarrollado en el trabajo de titulación antecesor.
	Descarga de archivos del proyecto antecesor desde su repositorio en la nube.
	Instalación de dependencias requeridas para el funcionamiento del proyecto
	Uso de la plataforma Google Console para la generación de llaves de API que permitan el funcionamiento del servicio.

	Ejecución del servicio en el entorno local mediante npm y prueba unitaria para comprobar su funcionamiento.
	Consumo del servicio por parte de la capa de RA para la obtención de POIs.

2.4.2 Ejecución del Sprint

Se describen algunos aspectos que durante el ejercicio del tercer sprint se hicieron evidentes:

Formulario de datos

Para el desarrollo del formulario se ha adaptado el diseño presentado por el proyecto de titulación antecesor con la finalidad de recolectar las preferencias turísticas del usuario. Los datos necesarios para la creación de itinerarios turísticos por parte del servicio son los siguientes:

- Número de días planificados para realizar el tour.
- La fecha en la que se planea iniciar el tour.
- La hora de inicio y finalización del tour por día.
- La hora de inicio y finalización del almuerzo.
- Uno o más sitios de interés que se deseen visitar.
- Las coordenadas geográficas donde se encuentra el usuario.

Conocidos estos requerimientos se ha diseñado un formulario simple con campos obligatorios y además las coordenadas geográficas se obtienen automáticamente del GPS del dispositivo móvil del usuario (ver Figura 30).

Tourist trip design problem

Total Days:

Start Date:

Desired schedule for tours:

Start:

End:

Desired schedule for lunch:

Start:

End:

Place of preference:

<input checked="" type="checkbox"/> Aquariums	<input type="checkbox"/> Premise	<input type="checkbox"/> Churches
<input type="checkbox"/> Cemeteries	<input type="checkbox"/> Art Galleries	<input type="checkbox"/> Zoo
<input type="checkbox"/> Mosques	<input type="checkbox"/> City Halls	<input type="checkbox"/> Parks
<input type="checkbox"/> Museums	<input type="checkbox"/> Stadiums	
<input type="checkbox"/> Amusement Park	<input type="checkbox"/> Indu Temples	

Figura 30. Formulario móvil para la recolección de datos

SDK de Mapbox

Para implementar el SDK de Mapbox dentro de Unity es necesario descargar el kit de herramientas desde su página oficial, en ella se detallan los pasos para su instalación [34]. En la carpeta Mapbox se encuentran todos los recursos necesarios para la creación y despliegue de un mapa 2D, los cuales se pueden apreciar en la Figura 31.

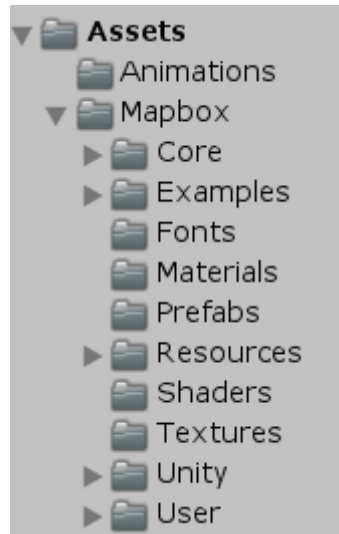


Figura 31. Recursos disponibles para el desarrollo de mapas con Mapbox.

Para el uso del SDK de Mapbox es necesario de un token de acceso que se obtiene en su página oficial [34], el cual es gratuito y debe ser insertado en la ventana emergente que se genera luego de importar los archivos del SDK de Mapbox a Unity, como se observa en la Figura 32.

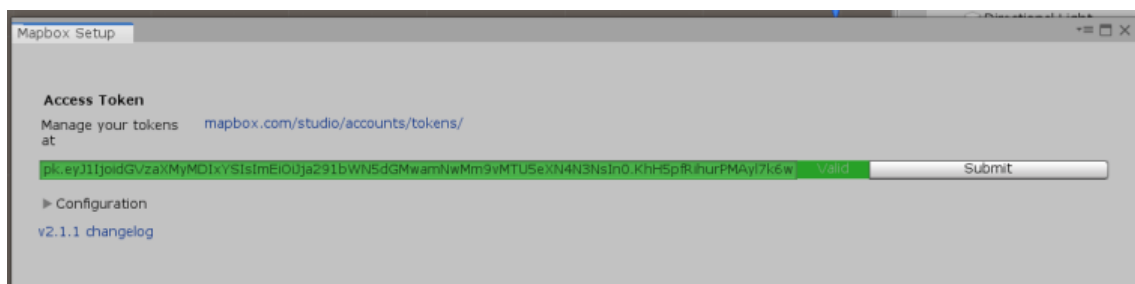


Figura 32. Uso del token de acceso de Mapbox en Unity

Elementos Map y Player de Mapbox

Los elementos principales para el desarrollo de un mapa 2D con Mapbox son el elemento Map y Player, el primero es el elemento más completo y despliega una serie de planos denominados tiles, los cuales representan áreas geográficas del mapa dependiendo de las coordenadas geográficas que se le asignen y su acercamiento (zoom), el segundo elemento permite representar la ubicación del usuario en conjunto con el elemento Map

mediante un icono por defecto. Se muestra el elemento necesario para la creación del mapa 2D con su respectivo Script que controla su funcionamiento en la Figura 33.

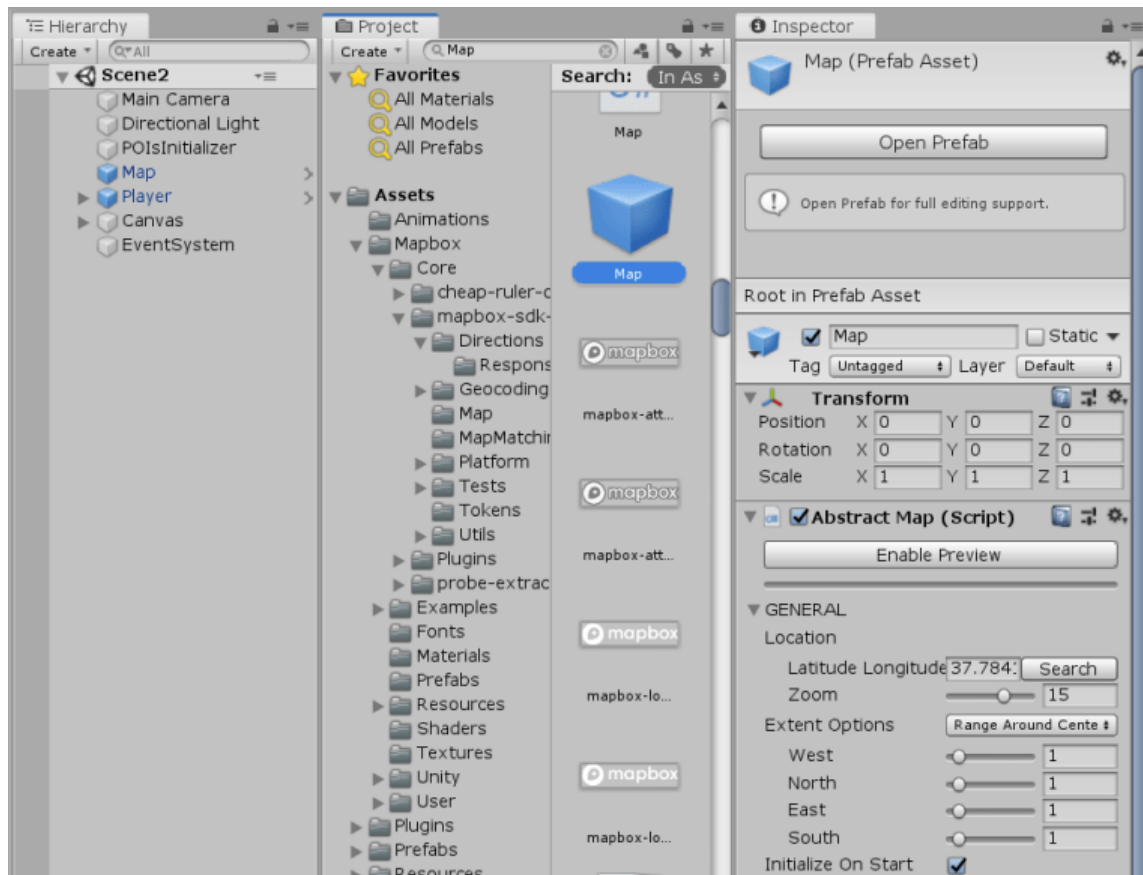


Figura 33. Objeto Map con su script AbstractMap

Script AddPOIs

En el script AddPOIs se obtiene el elemento Map necesario para representar los POIs con iconos predeterminados. Se requiere la creación de una nueva capa en el mapa que almacena las coordenadas de los POIs provenientes de la vista de RA, una vez se tenga esta nueva capa, la función AddPoisOfInterestSublayer(capaDefault) permitirá añadirla al elemento Map como se muestra en la Figura 34.

```

abstractMap = mapa.GetComponent<AbstractMap>();
variables = controller.GetComponent<Variables>();

capaDefault = new PrefabItemOptions();
capaDefault.prefabItemName = "Default";
capaDefault.spawnPrefabOptions = new SpawnPrefabOptions();
capaDefault.spawnPrefabOptions.prefab = prefab;
capaDefault.isActive = true;
capaDefault.snapToTerrain = true;
capaDefault.findByType = LocationPrefabFindBy.AddressOrLatLon;
list_default = new List<string>();
capaDefault.coordinates = new string[1];

for (int i = 0; i < variables.days[variables.daySelected].Count; i++)
{
    list_default.Add(String.Format(CultureInfo.InvariantCulture, "{0},{1}");
}

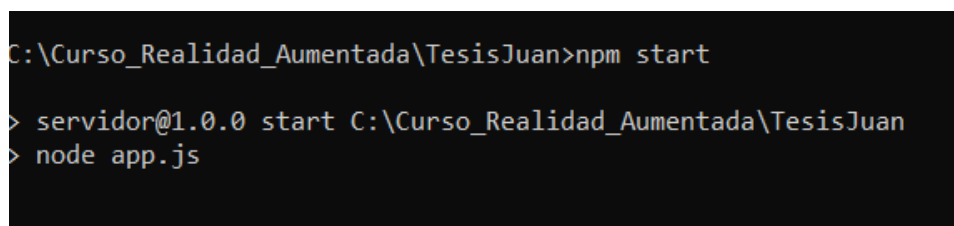
capaDefault.coordinates = list_default.ToArray();
abstractMap.VectorData.AddPointsOfInterestSubLayer(capaDefault);

```

Figura 34. Función para añadir capa de POIs al mapa 2D.

Consumo del servicio local

La ejecución del comando `npm start` permite levantar el servicio de forma local, previamente al realizar su consumo (ver Figura 35).



```

C:\Curso_Realidad_Aumentada\TesisJuan>npm start
> servidor@1.0.0 start C:\Curso_Realidad_Aumentada\TesisJuan
> node app.js

```

Figura 35. Ejecución del comando `npm start` para levantar el servicio.

Para el consumo de servicios Rest, se utiliza la librería `UnityWebRequest`, la cual ofrece un sistema modular para el envío de peticiones HTTP y controlar su respuesta, adicionalmente, se utiliza la estructura de datos `Json`, por cuanto permite encapsular la información como objetos con atributos y en formato sencillo, además de la rapidez del envío y recepción de datos. La función en C# desarrollada para realizar la solicitud puede ser vista en la Figura 36.

```

IEnumerator PostData_Coroutine()
{
    var request = new UnityWebRequest("http://localhost:8080/ttdp", "POST");

    string json = JsonUtility.ToJson(variables.serviceLoadData);

    byte[] bodyRaw = Encoding.UTF8.GetBytes(json);
    request.uploadHandler = (UploadHandler)new UploadHandlerRaw(bodyRaw);
    request.downloadHandler = (DownloadHandler)new DownloadHandlerBuffer();
    request.SetRequestHeader("Content-Type", "application/json");

    yield return request.SendWebRequest();

    if (request.error != null)
    {
        Debug.Log("Error: " + request.error);
        PlayerPrefs.SetString("errorDatosServicio", "Could not connect to the service: " + request.error);
        Screen.orientation = ScreenOrientation.Portrait;
        SceneManager.LoadScene("SceneForm");
    }
}

```

Figura 36. Función para enviar solicitud POST al servicio.

2.4.3 Sprint Review

Pruebas de aceptación

Tabla 14. Pruebas de aceptación del Sprint 3.

Historia de Usuario	Criterio de aceptación	Cumplido
AH1-05	La estructura del formulario deberá adaptarse a los requerimientos de la llamada al servicio web tales como: <ul style="list-style-type: none"> • Días de recorrido • Fecha de inicio de viaje • Horarios deseados para tours y almuerzo • Preferencias turísticas 	SÍ
	Cuando el usuario llené el formulario de recolección de preferencias turísticas, entonces los campos del formulario deberán ser claros y fáciles de entender.	SÍ
	Cuando el usuario se encuentre ingresando los datos del formulario, entonces el formulario deberá ser capaz de captar las preferencias turísticas del usuario.	SÍ

	Cuando el usuario no ingrese todos los datos del formulario, entonces el formulario deberá alertar al usuario sobre los campos obligatorios.	SÍ
	Cuando el usuario envíe la información completa de sus preferencias, entonces deberá aparecer una pantalla de carga.	SÍ
AH2-01	Cuando el usuario ingrese a la vista de mapa, entonces la aplicación deberá ser capaz de presentar un mapa 2D con vista aérea.	SÍ
	Cuando el usuario ingrese a la vista de mapa, entonces se desplegará el mapa 2D centrado en la ubicación del usuario.	SÍ
AH2-02	Cuando el usuario se desplace y sus coordenadas geográficas cambien, entonces el mapa 2D se actualizará a su nueva ubicación.	SÍ
	Cuando se visualice el mapa 2D, entonces los POIs se presentarán como iconos dentro del mapa.	SÍ
	Cuando se visualice el mapa 2D en la ubicación del usuario, entonces el icono del usuario se presentará dentro del mapa.	SÍ
	Cuando el usuario desee controlar el desplazamiento y la vista por el mapa, entonces podrá hacerlo con los elementos de interfaz de usuario.	SÍ
AH3-02	Cuando el dispositivo móvil se encuentre conectado a internet, entonces el usuario podrá usar el aplicativo en su totalidad.	SÍ
	Cuando el aplicativo realice una petición al servicio web, entonces se recibirán los itinerarios turísticos.	SÍ

Incremento Obtenido

Presentación inicial de los elementos de Mapbox: Map y Player

Se muestra el mapa en 2D en ejecución implementando los elementos Map y Player de Mapbox en la Figura 37.

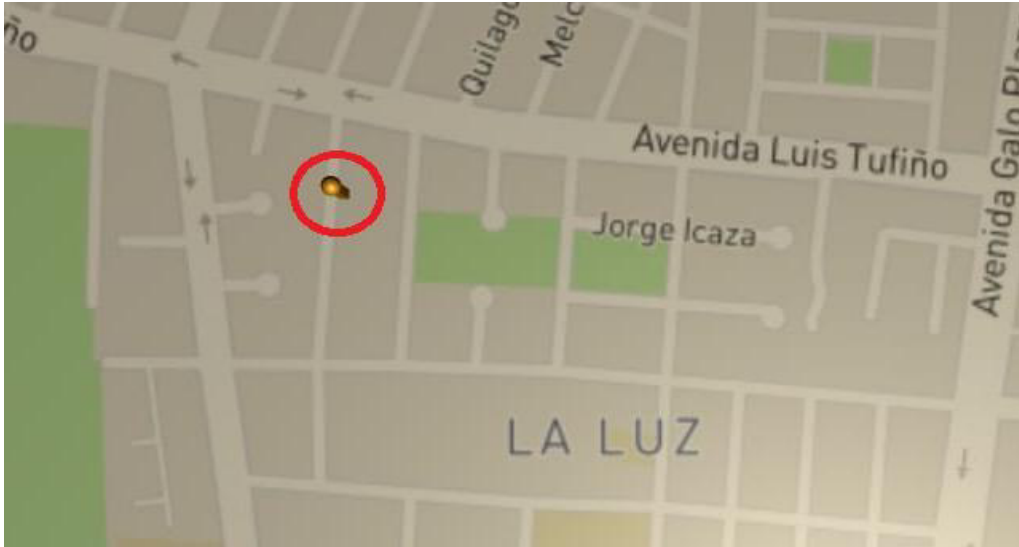


Figura 37. Elemento Player dentro del elemento Map de Mapbox

Adaptación del Product Backlog

Tras la ejecución del sprint y su total cumplimiento se ha identificado la necesario agregar una nueva historia de usuario que complemente a la historia épica AH2, la cual se orienta a la creación de rutas entre el usuario y los POIs dentro del mapa 2D, misma que se puede observar dentro de la actualización del Product Backlog en la Tabla 15.

Tabla 15. Product backlog adaptado tras la ejecución del Sprint 3

PRODUCT BACKLOG				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Estimación (días)	Estado
AH1	AH1-01	Uso de una herramienta de software que implemente RA.	5	Terminado
	AH1-02	Desarrollo de funciones de geolocalización para los POIs.	10	Terminado

	AH1-03	Representación de POIs mediante modelos 3D.	3	Terminado
	AH1-04	Despliegue de un panel que muestre la información de los POIs.	3	Terminado
	AH1-05	Recolección de información de preferencias turísticas mediante un formulario.	2	Terminado
AH2	AH2-01	Integración de herramienta de software para la representación de mapas.	5	Terminado
	AH2-02	Desarrollo de las funcionalidades requeridas en la vista de mapa.	15	Terminado
	AH2-03	Implementación de rutas entre el usuario y los POIs en la vista de mapa.	5	Por Implementar
AH3	AH3-01	Publicación del proyecto de titulación predecesor en una plataforma de servicios en la nube.	5	Por Implementar
	AH3-02	Desarrollo de las funciones para el consumo del servicio web.	5	Terminado

2.4.4 Retrospectiva y velocidad del equipo

En la última reunión de retrospectiva se determinó el cumplimiento de todo el trabajo planificado dentro del tiempo establecido y no se encontró ninguna observación sobre el desarrollo del proyecto, además se puede observar que la distribuyó del tiempo fue de mejor manera para concluir con todas las tareas del proyecto, puede ver el gráfico de burndown de la Figura 38.

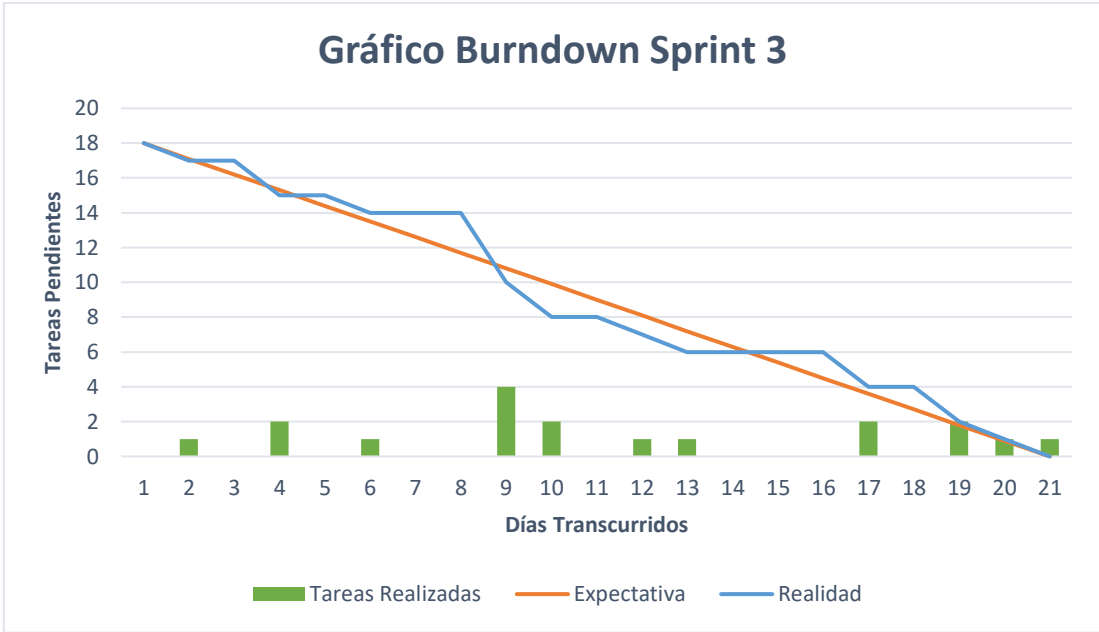


Figura 38. Gráfico burndown del Sprint 3.

2.5 Sprint 4

2.5.1 Sprint Planning

Objetivo del Sprint

Concluir el desarrollo de funcionalidades para la vista de Mapbox y realizar la publicación del servicio web en la nube.

Historias de Usuario

La historia del sprint al igual que el resto de historias restantes experimentarán el proceso de finalización en este sprint, tal como se muestra en la Tabla 17.

Tabla 16. Historias de usuario escogidas para el Sprint 4 (Sus detalles completos se encuentran en el Anexo V).

ID	Título	Estimación (días)	Estado
AH2-03	Implementación de rutas entre el usuario y los POIs en la vista de mapa.	5	Por implementar
AH3-01	Publicación del proyecto de titulación predecesor en una plataforma de servicios en la nube.	5	Por Implementar

Sprint Backlog

Tabla 17. Sprint backlog del Sprint 3

SPRINT BACKLOG	
HISTORIA DE USUARIO	TAREAS
AH2-03	Búsqueda de elementos incluidos en el SDK de Mapbox para generar rutas entre puntos.
	Desarrollo de funciones para generar las rutas necesarias entre el usuario y los POIs del usuario.

	Pruebas para corregir errores en la generación del camino entre el usuario y un POI.
AH3-01	Investigar plataformas que permitan la publicación de un servicio desarrollado en NodeJs a la nube.
	Creación de una máquina virtual en la plataforma AWS para almacenar y ejecutar los archivos del servicio.
	Configuraciones del entorno virtual para el uso posterior del servicio.
	Pruebas para corroborar el correcto funcionamiento del servicio desde cualquier dispositivo con conexión a internet.

2.5.2 Ejecución del Sprint

Los detalles más importantes que han surgido durante la ejecución del sprint se detallan a continuación:

Elemento Directions de Mapbox

Mapbox posee el elemento Directions, este elemento permite gestionar la ruta entre dos elementos del mapa, renderizando el camino más cercano mediante su script DirectionsFactory. Este script es el encargado de recalculando la ruta a medida que el usuario se acerque al POI seleccionado; para el uso es requerido un elemento en la escena, el cual es generado mediante la función createDirections durante la ejecución de la aplicación (ver Figura 39). Al momento en que el usuario seleccione un nuevo POI, la ruta establecida con el anterior POI será destruida y se generará un nuevo elemento Directions.

```
public void createDirections(Transform[] transforms)
{
    GameObject direction = Instantiate(prefabDirection);
    DirectionsFactory directionsFactory = direction.GetComponent<DirectionsFactory>();
    Transform[] waypoints = transforms;
    directionsFactory.addWaypoints(waypoints);
}
```

Figura 39. Función createDirections para la creación de rutas

Módulo EC2 de AWS

Para el uso del módulo EC2 es necesaria la creación de una instancia la cual alberga una máquina virtual con el sistema operativo Linux. Al poseer una máquina virtual recién creada, es necesario realizar la actualización de las librerías de Linux y la descarga e instalación de Nodejs, dentro de esta máquina virtual se almacenan los archivos requeridos para la ejecución del servicio en Nodejs y se realiza la ejecución del mismo con el comando `npm start` [26]. El panel de control con el resumen de la instancia creada en AWS se muestra en la Figura 40.



Figura 40. Instancia del módulo EC2 en AWS para el almacenamiento del servicio

2.5.3 Sprint Review

Pruebas de aceptación

Historia de Usuario	Criterio de aceptación	Cumplido
AH2-03	Cuando el usuario seleccione un POI en la vista de mapa, entonces el mapa 2D presentara la ruta más cercana al POI seleccionado.	SÍ
	Cuando se cambie la selección de un POI en la vista de mapa, entonces se eliminará las rutas anteriores y únicamente se presentará la ruta al POI seleccionado.	SÍ
	Cuando el usuario cambie el valor del zoom superando las 8 unidades, entonces la ruta al POI seleccionado no se mostrará.	SÍ

	Cuando el usuario seleccione la opción de ver todos los POIs, entonces no se presentará ninguna ruta.	Sí
	Cuando que el usuario avance por el mapa, entonces la ruta se actualizará.	Sí
	Cuando el POI salga de la pantalla del usuario en la vista de mapa, entonces la ruta desaparecerá.	Sí
AH3-01	Cuando el servicio este alojado en la nube mediante una plataforma que le asigne una dirección IP publica, entonces el aplicativo podrá funcionar en su totalidad.	Sí
	Cuando un dispositivo móvil tenga el aplicativo instalado y conexión a internet, entonces el aplicativo estará en la capacidad de realizar consultas al servicio web.	Sí

Incremento Obtenido

Ruta generada por el elemento Directions de Mapbox

Un ejemplo de ruta entre el elemento Player y un POI seleccionado gracias al elemento drop down se muestra en la Figura 41.

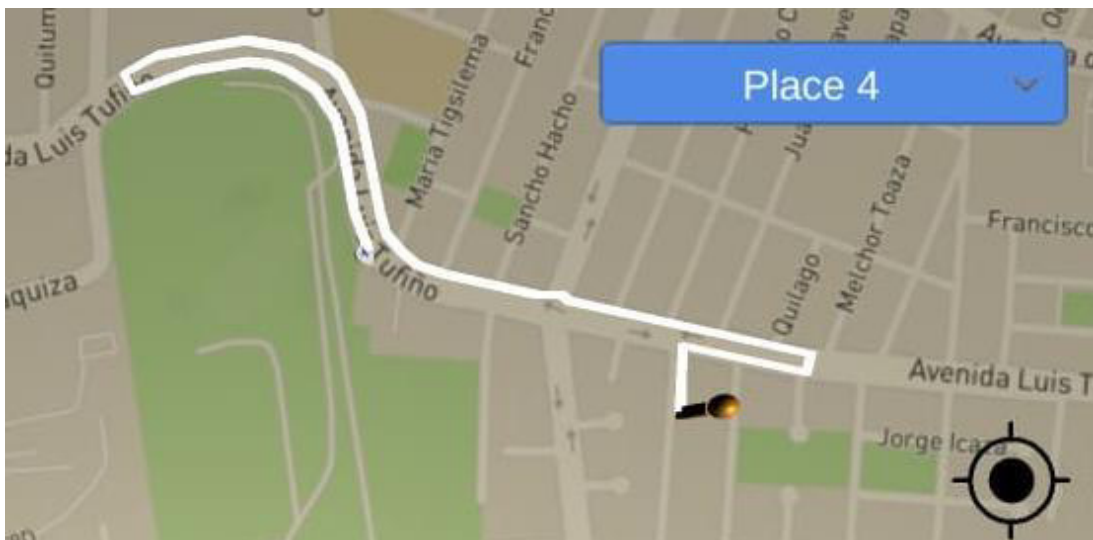


Figura 41. Ruta generada entre el elemento Player y un POI

Adaptación del Product Backlog

Tabla 18. Product backlog tras la finalización del proyecto.

PRODUCT BACKLOG				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Estimación (días)	Estado
AH1	AH1-01	Uso de una herramienta de software que implemente RA.	5	Terminado
	AH1-02	Desarrollo de funciones de geolocalización para los POIs.	10	Terminado
	AH1-03	Representación de POIs mediante modelos 3D.	3	Terminado
	AH1-04	Despliegue de un panel que muestre la información de los POIs.	3	Terminado
	AH1-05	Recolección de información de preferencias turísticas mediante un formulario.	2	Terminado
AH2	AH2-01	Integración de herramienta de software para la representación de mapas.	5	Terminado
	AH2-02	Desarrollo de las funcionalidades requeridas en la vista de mapa.	15	Terminado
	AH2-03	Implementación de rutas entre el usuario y los POIs en la vista de mapa.	5	Terminado
AH3	AH3-01	Publicación del proyecto de titulación predecesor en una plataforma de servicios en la nube.	5	Terminado

	AH3-02	Desarrollo de las funciones para el consumo del servicio web.	5	Terminado
--	--------	---------------------------------------------------------------	---	-----------

2.5.4 Retrospectiva y velocidad del equipo

En el último sprint se presentaron varias dificultades que dilataron la terminación del proyecto, las complicaciones ocurrieron debido a la dificultad de adaptación de la herramienta Mapbox en la realización de funciones requeridas en el trabajo, por cuanto la documentación con respecto al código implementado en sus componentes carece de claridad, sin embargo la mayor cantidad de tareas fueron realizadas con éxito en las últimas etapas de la planificación pudiendo concluir el trabajo sin inconvenientes (ver Figura 42).

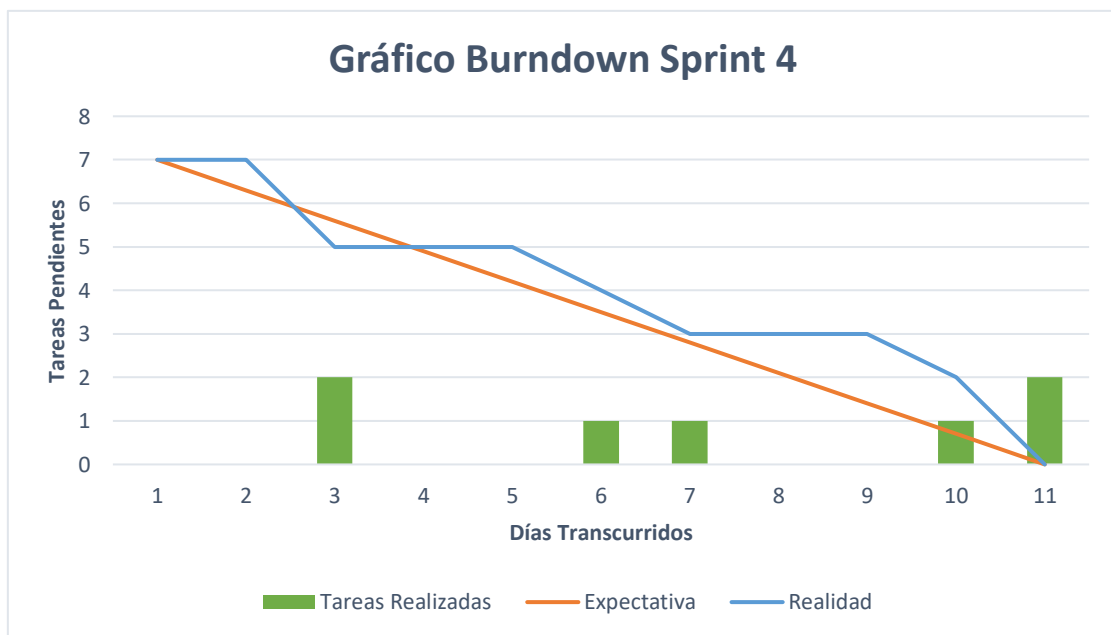


Figura 42. Gráfico *burndown* del *Sprint 4*

3 RESULTADOS Y DISCUSIÓN

3.1 Producto Final

La aplicación final está conformada por 3 pantallas y una extra de transición, la primera pantalla funciona para la toma de información con respecto a las preferencias del usuario, como se muestra en la Figura 43, seguidamente se despliega la pantalla extra mientras sucede la carga los datos, la segunda pantalla muestra los POIs cercanos en RA y la última pantalla presenta los mismos POIs sobre un mapa 2D, como se observa en la Figura 44.

Tourist trip design problem

Total Days:

Start Date:

Desired schedule for tours:

Start:

End:

Desired schedule for lunch:

Start:

End:

Place of preference:

<input type="checkbox"/> Aquariums	<input type="checkbox"/> Premise	<input type="checkbox"/> Churches
<input type="checkbox"/> Cemeteries	<input type="checkbox"/> Art Galleries	<input type="checkbox"/> Zoo
<input type="checkbox"/> Mosques	<input type="checkbox"/> City Halls	<input checked="" type="checkbox"/> Parks
<input type="checkbox"/> Museums	<input type="checkbox"/> Stadiums	
<input type="checkbox"/> Amusement Park	<input type="checkbox"/> Indu Temples	

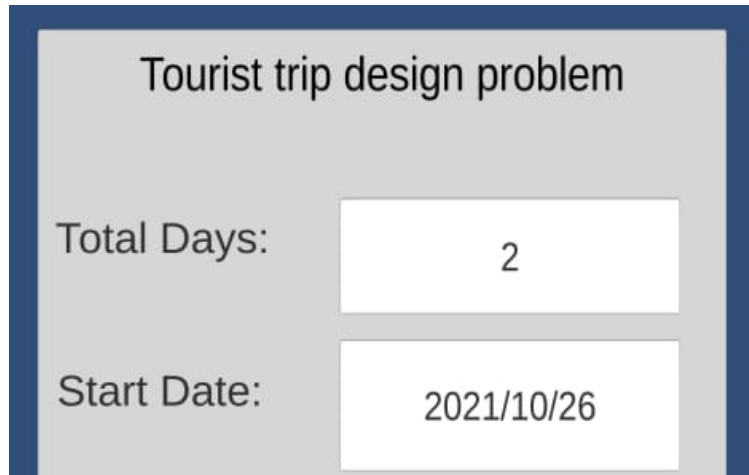
Figura 43. Formulario final para la recolección de datos.



Figura 44. Capturas de pantalla de la aplicación móvil final.

3.1.1 Toma de Datos

Con respecto a las preferencias del turista, se extrae la información necesaria para consultar los itinerarios, donde se muestra el número de días y fecha de inicio de su viaje (ver Figura 45).



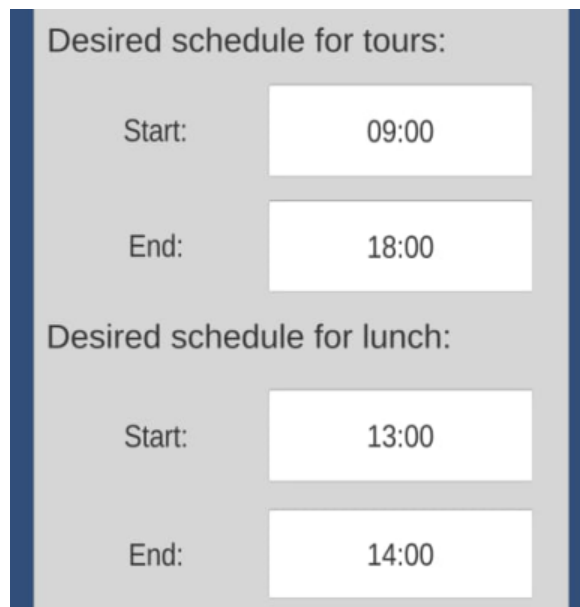
Tourist trip design problem

Total Days:

Start Date:

Figura 45. Campos para los días de viaje y fecha de inicio.

Del mismo modo, se puede asignar el horario de inicio y fin para los tours, de igual manera que el inicio y fin de la hora de almuerzo, esto se puede observar en la Figura 46.



Desired schedule for tours:

Start:

End:

Desired schedule for lunch:

Start:

End:

Figura 46. Campos para los horarios del tour y de almuerzo.

Para que el itinerario sea personalizado, se selecciona los POIs que el turista desea visitar entre la lista disponible, los lugares posibles para visitar se pueden observar en la Figura 47.

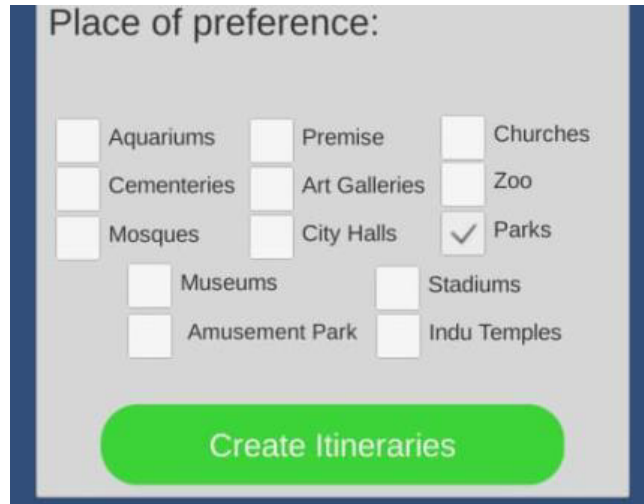


Figura 47. Lugares de preferencia disponibles.

La ubicación de partida se determina automáticamente dependiendo de las coordenadas geográficas del usuario.

3.1.2 Vista de Realidad Aumentada

Realizado él envió de datos se obtiene la respuesta del servicio con los itinerarios para los días especificados, la función de la vista de RA consiste en mostrar estos POIs representados como iconos interactivos, cada POI dependiendo de su tipo se presentará con su icono representativo, esto se puede observar en la Figura 48.



Figura 48. Ejemplo de icono representativo para iglesias.

Los POIs a mostrar en la vista de RA dependen del día seleccionado (ver Figura 49) y de la distancia a la que se encuentre el usuario de ellos, en este caso se ha optado porque sean 70 metros, como se muestra en la Figura 50.



Figura 49. Selección del día deseado para el despliegue de POIs.

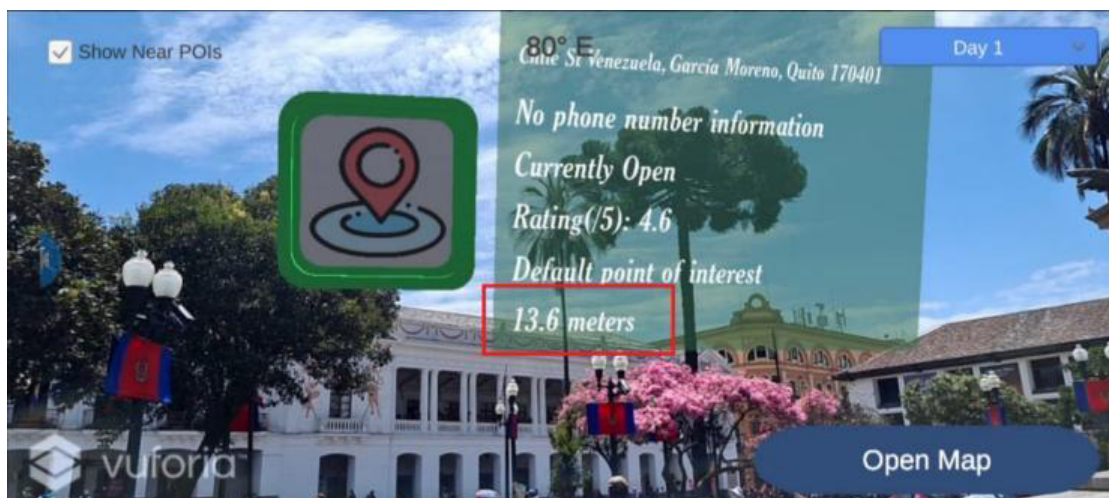


Figura 50. Distancia a la que se encuentra un POI del usuario.

3.1.3 Vista de Mapa 2D

Otra forma de representar los POIs ocurre en la vista de mapa 2D, para acceder a ella se utiliza el botón "open map" de la vista de RA. A la vista de mapa 2D se transfieren los POIs del día seleccionado en la vista anterior, se muestran inicialmente todos los puntos además de la posición actual del usuario como se ve en la Figura 51. El usuario puede seleccionar

el POI al que desee dirigirse y se creará el camino más cercano a dicho punto, esto se puede observar en la Figura 52.

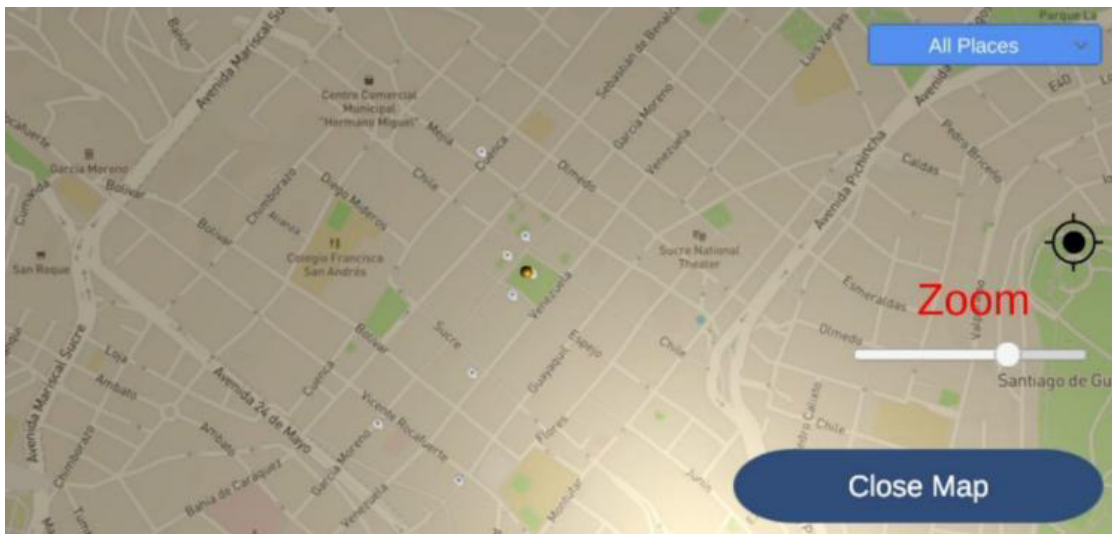


Figura 51. Vista de Mapa en 2D con sus POIs.

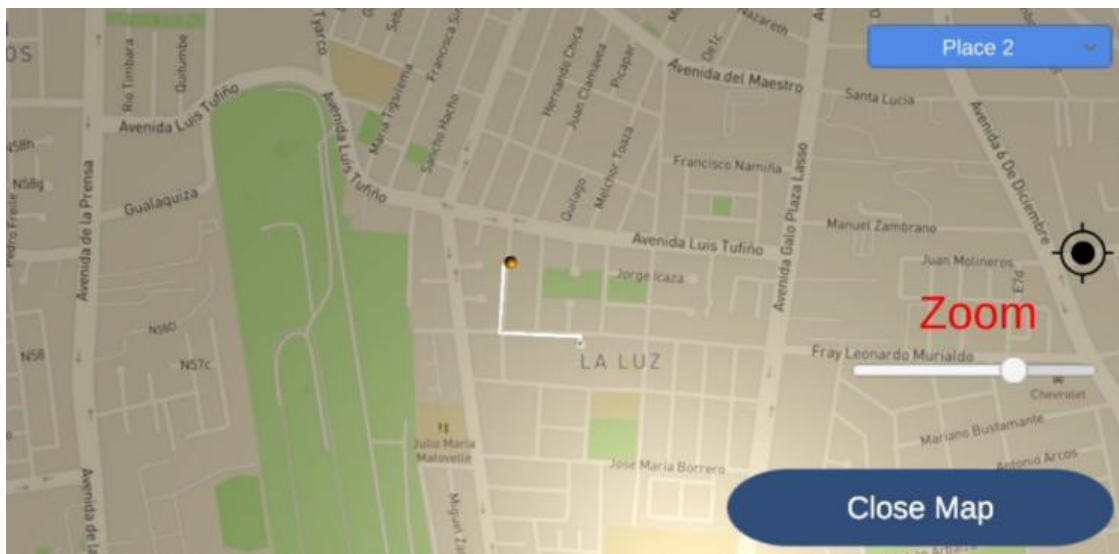


Figura 52. Generación de ruta desde el usuario al POI seleccionado

3.2 Pruebas con usuarios finales

Se ha realizado pruebas de la aplicación móvil con un total de 25 personas, las cuales laboran en la empresa EGM Robotics, se solicitó a los participantes que llenen un formulario diseñado con la intención de medir la utilidad y facilidad de uso percibidas para la representación en RA de los POIs y el uso de la aplicación en general de acuerdo con

el modelo TAM (*Technology Acceptance Model*) [35]. Se realizaron un total de nueve preguntas por cada participante, tres referentes a RA, tres para la utilidad percibida y tres para la facilidad de uso; la valoración de cada pregunta se definió con una escala del 1 al 7, donde 1 es la calificación más baja y 7 es la calificación más alta. Los resultados del formulario se encuentran adjuntos en el Anexo VII.

3.2.1 Formulario de RA

El formulario de RA tiene como objetivo consultar la opinión del usuario con respecto a los elementos de RA implementados en la aplicación, en el cual la primera pregunta es ¿Considera que los elementos de Realidad Aumentada mejoran la experiencia turística?; la cual refiere a la opinión del usuario respecto a los elementos de RA y si estos son capaces de enriquecer una visita turística, cuyo resultado alcanzó el promedio de 6.12 / 7 y demuestra que los usuarios consideran que la RA puede ser un gran complemento para una visita turística (ver Figura 53).

La segunda pregunta realizada es ¿Le parece útil que la interfaz de Realidad Aumentada complemente a la interfaz de mapas 2D?, la cual refiere a la vista de RA dentro de la aplicación y si esta se complementa correctamente con la vista de mapas la cual le permite al usuario guiarse dentro de la ciudad. El resultado que se obtuvo fue un promedio de 6 / 7 lo cual indica que la vista de mapas se complementa adecuadamente con la vista de RA, permitiendo a los usuarios desplazarse correctamente por la ciudad durante su visita turística al usar la aplicación.

La tercera pregunta del formulario de RA es ¿Consideraría utilizar aplicaciones que integren Realidad Aumentada en un futuro cercano?, la cual refiere a la opinión del usuario respecto a las aplicaciones que integran elementos de RA y si consideraría utilizar dichas aplicaciones en un futuro cercano. El resultado alcanzó fue un promedio de 5.44 / 7 por cuanto aproximadamente la mitad de los usuarios no tienen planeado utilizar otras aplicaciones relacionadas con RA, ya sea porque no son de su agrado o porque desconocen la existencia de otras aplicaciones que integren RA.

Formulario de Realidad Aumentada

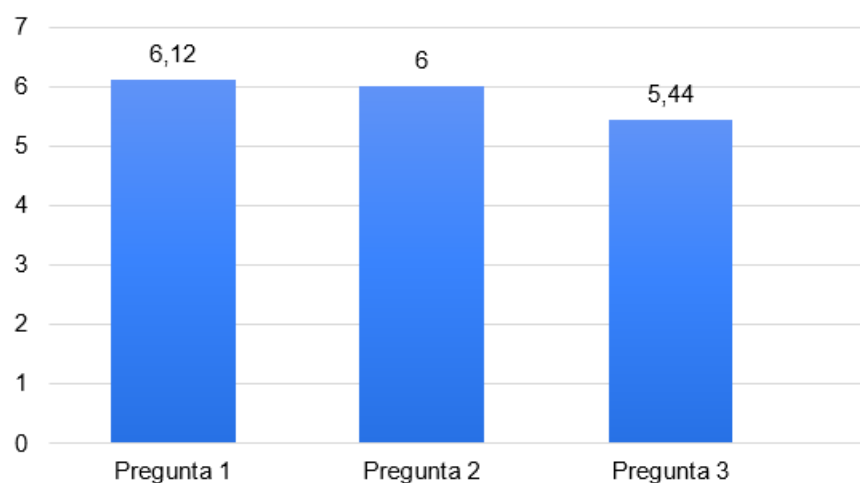


Figura 53. Resultados promedio obtenidos en las preguntas del formulario de RA.

3.2.2 Utilidad percibida

Los promedios obtenidos en las preguntas del formulario de utilidad percibida se pueden observar en la Figura 54. La primera pregunta realizada es ¿Le resultó útil la información que se muestra para cada punto de interés?, esta refiere a el valor de la información desplegada en los iconos 3D de cada POI y si el usuario considera que esta información sea útil. El resultado obtenido fue un promedio de 6.29 / 7 lo cual indica que los usuarios consideran útil esta información y les facilita el desplazarse por los distintos POI de la ciudad.

La segunda pregunta realizada es Tras usar el aplicativo, ¿Mejoró su experiencia de visita en la ciudad?, esta refiere si el usuario considera que la aplicación móvil es capaz de mejorar la experiencia turística. El resultado que se obtuvo fue un promedio de 6.25 / 7 lo cual indica que la aplicación si puede llegar a mejorar la experiencia turística de la mayoría de usuarios.

Formulario de Utilidad Percibida

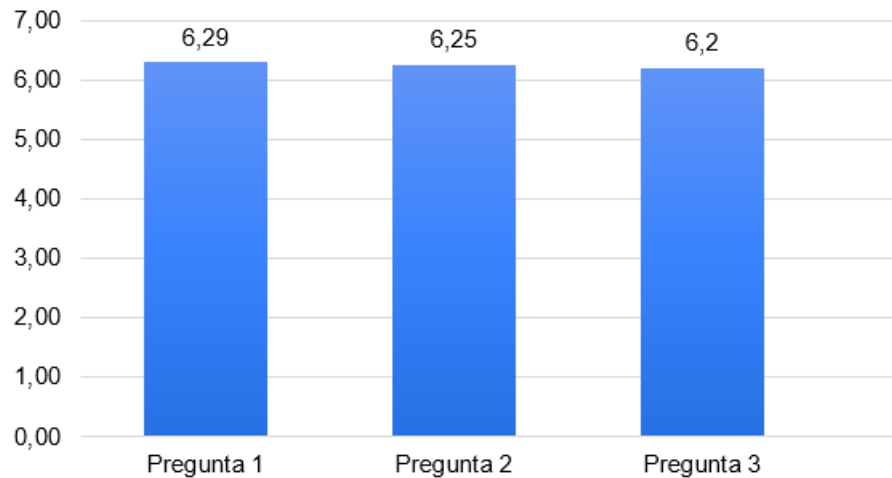


Figura 54. Resultados promedio obtenidos en las preguntas del formulario de utilidad percibida.

La tercera pregunta realizada es ¿Utilizaría o recomendaría el aplicativo a otra persona para mejorar la experiencia durante un viaje turístico?, esta refiere si el usuario recomendaría la aplicación a otras personas con el fin de mejorar su experiencia turística en la ciudad. El resultado obtenido fue un promedio de 6.2 / 7 y refleja que la mayoría de usuarios consideran útil la aplicación y que puede llegar a ser utilizada por otras personas.

3.2.3 Facilidad de uso percibida

Se pueden observar los promedios obtenidos de cada pregunta en la figura 55. La primera pregunta del formulario de facilidad de uso es ¿Los iconos en 3D tiene un tamaño correcto y son entendibles visualmente?, esta se refiere al tamaño y visibilidad de los iconos 3D desarrollados para representar a los POIs. El resultado obtenido fue un promedio de 6.16 / 7 debido a que los usuarios consideran que el tamaño y visibilidad de los iconos son adecuados.

Formulario de Facilidad de Uso

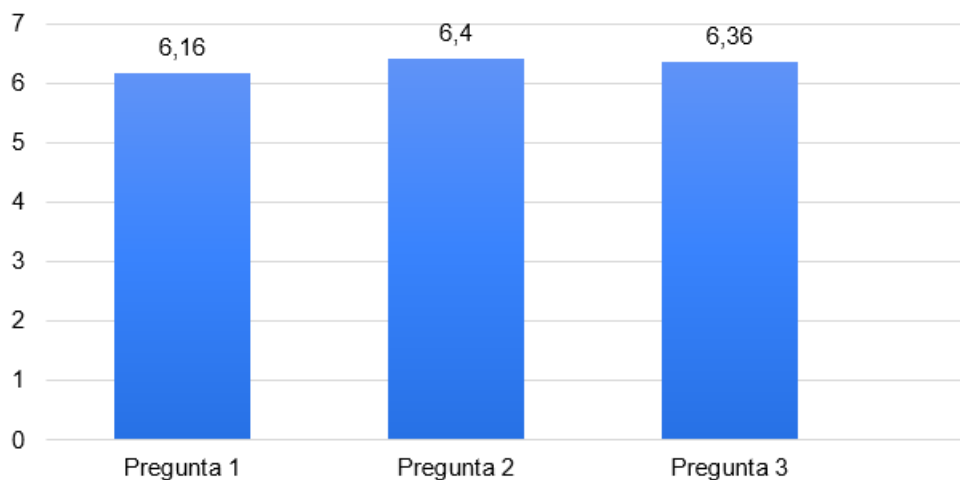


Figura 55. Resultados promedio obtenidos en las preguntas del formulario de facilidad de uso.

La segunda pregunta realizada es ¿Le resulta sencillo desplazarse entre las distintas pantallas de la aplicación?, esta se refiere a la facilidad percibida por el usuario para desplazarse entre las distintas pantallas de la aplicación. El resultado obtenido fue un promedio de 6.4 / 7 lo cual refleja que la mayoría de usuarios no encontraron complicaciones al desplazarse por las distintas pantallas de la aplicación.

La tercera y última pregunta realizada es ¿Le resultó sencillo obtener la información del punto de interés que desea?, esta refiere a la facilidad para obtener la información de un POI utilizando la aplicación. El resultado obtenido fue un promedio de 6.36 / 7 lo cual indica que para los usuarios fue sencillo el obtener información de un POI deseado.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- La aplicación desarrollada ofrece una propuesta que enriquece la presentación de información en sistemas de recomendación turística, representando los POIs de una manera más atractiva para los usuarios mediante el uso de elementos de RA.
- Con el uso de las herramientas Unity y Vuforia fue posible integrar elementos de RA a la aplicación móvil, los cuales están conformados por una imagen representativa del POI y por un panel informativo que despliega los datos más importantes del POI.
- Los itinerarios turísticos que contienen la información de los POIs se generan mediante una consulta inicial al Rest Api desarrollado en el trabajo de titulación anterior, para lo cual se implementó una pantalla inicial de la aplicación como formulario para la recopilación de preferencias turísticas del usuario.
- Se desarrolló la interfaz de RA enfocada únicamente para dispositivos móviles con el sistema operativo Android versión 10 o superior y que posean una brújula interna integrada.
- Se verificó la funcionalidad de la aplicación dentro de la ciudad de Quito, en la zona del centro histórico, demostrando un rendimiento aceptable en cuanto a ubicación y tamaño de los iconos de RA, cumpliendo con la finalidad de enriquecer las visitas turísticas en la ciudad.
- La mayoría de los usuarios encuentran atractiva la aplicación y poseen calificaciones positivas en cuanto a utilidad y facilidad de uso, lo cual demuestra que la aplicación tiene un gran potencial si se continúa con el desarrollo y corrección de errores a futuro.
- La cercanía de los POIs depende en gran medida de la ubicación del usuario, además de sus preferencias turísticas, esto implica que al utilizar la aplicación en zonas rurales lo más probable es que se obtengan escasos POIs mientras que al utilizarla en sitios de alta concentración turística se obtendrá una mayor cantidad.
- El GPS y brújula de los dispositivos móviles actuales no poseen el 100% de exactitud en sus medidas, lo que conlleva a la presencia de desfases al momento de ubicar los iconos de RA en la ubicación geográfica exacta de un POI, esto puede provocar que algunos iconos se muestren ligeramente fuera del rango con respecto a la ubicación real del POI.

- La aplicación móvil depende en gran medida del servicio desarrollado por el trabajo antecesor, el cual ha sido publicado temporalmente en la plataforma AWS con el fin de desarrollar el presente trabajo, es necesario que este servicio permanezca activo, para que el usuario pueda obtener los POIs en la ejecución del aplicativo.
- El uso de Mapbox para mapa 2D soluciona el enfoque pesado que entrega la vista de RA, de esta forma es más accesible encontrar los POIs que se necesita con una vista aérea. El mapa 2D sirve para entregar mejor experiencia y percepción a los nuevos usuarios que utilicen la aplicación.

4.2 Recomendaciones

Para trabajos futuros se recomienda:

- Las investigaciones posteriores deberán ofrecer mayor flexibilidad al momento de mostrar los POIs, de tal forma que el usuario pueda ordenar a que distancia desea empezar a ver los POIs, así como también darle prioridad a resaltar ciertos tipos de POIs con respecto a los demás.
- Es necesario volver a publicar la aplicación web desarrollada en el trabajo de tesis antecesor en una plataforma en la nube como AWS, con el fin de poder realizar las peticiones requeridas para el correcto funcionamiento de la aplicación desarrollada en el presente trabajo.
- La aplicación de RA debe complementarse con una aplicación que obtenga datos climáticos para determinar si es viable el visitar los POIs, debido a que el clima en la ciudad de Quito puede llegar a variar dependiendo de la ubicación donde se encuentre el usuario.
- El mapa 2D se trazan caminos considerando que se transporta en automóvil, de tal manera que se debe ampliar la trayectoria para más tipos de desplazamientos que puede realizar el usuario, ya sea caminando, bicicleta, moto u otros transportes.
- El aplicativo para planificar itinerarios debe utilizarse para un pequeño número de días y de esta manera evitar que el tiempo de espera al obtener los datos sea demasiado largo.

5 Referencias Bibliográficas

- [1] B. Furht, Handbook of Augmented Reality, Florida Atlantic University, 2011.
- [2] Realidad Aumentada: una nueva lente para ver el mundo, Madrid: Ariel S.A, 2011.
- [3] «El Comercio,» 2017. [En línea]. Available: <https://www.elcomercio.com/viajar/centrohistorico-visitas-turistas-quito-vacaciones.html>. [Último acceso: 19 Noviembre 2021].
- [4] R. T. Azuma, «A Survey of Augmented Reality,» *Teleoperators and Virtual Environments*, vol. 6, nº 4, 1997.
- [5] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani y M. Ivkovic, «Augmented reality technologies, systems and applications,» *Multimedia Tools and Applications*, vol. 51, 2010.
- [6] D. Chatzopoulos, C. Bermejo, Z. Huang y P. Hui, «Mobile Augmented Reality Survey: From Where We Are to Where We Go,» *IEEE*, vol. 5, pp. 6917-6950, 2017.
- [7] G. B. López, «La geolocalización social,» *Polígonos, revista de Geografía*, pp. 97-118, 2015.
- [8] J. García, J. Fernandez, P. Gálvez, S. Antonio y M. Grimaldi, «La Geolocalización como Herramienta para Conocer a los Consumidores: El caso de los centros de Fitness,» *PODIUM Sport*, pp. 263-276, 2017.
- [9] R. R. G.M. Djuknic, «Geolocation and assisted GPS,» *IEEE*, vol. 34, nº 2, pp. 123-125, 2001.
- [10] J. V. Sicle, Basic GIS Coordinates, CRC Press, 2017.
- [11] R. C, «The Cosine-Haversine Formula,» *The American Mathematical Monthly*, vol. 64, pp. 38-40, 1957.
- [12] M. Kisanrao Nichat, «Landmark based shortest path detection by using A* Algorithm and Haversine Formula,» *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, 2013.
- [13] T. Ryan y M. Ryan, «Geographical Orientation,» *Journal of Social Sciences*, vol. 53, nº 2, p. 204, 1940.
- [14] «Free images and royalty free stock,» 2021. [En línea]. Available: <https://pixabay.com/>. [Último acceso: 26 de noviembre 2021].

- [15] P. Zandbergen y S. Barbeau, «Positional Accuracy of Assisted GPS Data from High-Sensitivity GPS-enabled Mobile Phones,» *Journal of Navigation*, vol. 64, nº 3, 2011.
- [16] J. Pfaff, «Mobile Phone Geographies,» *Geography Compass*, vol. 4, nº 10, 2010.
- [17] K. Muhammand, P.-V. Nguyen y S. Gan, «VibraTilt: Accelerometer & Gyroscope measurement app,» *Scientific Phone Apps and Mobile Device*, vol. 2, nº 1, 2016.
- [18] «Unity - Manual: Working in Unity,» 2021. [En línea]. Available: <https://docs.unity3d.com/Manual/UnityOverview.html> . [Último acceso: 19 de noviembre 2021].
- [19] «U. Technologies, "Vuforia - Unity Manual",» 2021. [En línea]. Available: <https://docs.unity3d.com/es/2018.4/Manual/vuforia-sdk-overview.html>. [Último acceso: 22 de noviembre 2021].
- [20] «Maps SDK | Unity,» 2021. [En línea]. Available: <https://docs.mapbox.com/unity/maps/guides/>. [Último acceso: 22 de noviembre 2021].
- [21] «Visual Studio: IDE y Editor de código para desarrolladores de software y Teams,» 2021. [En línea]. Available: <https://visualstudio.microsoft.com/es/>. [Último acceso: 22 de noviembre 2021].
- [22] «Introduction — Blender Manual,» 2021. [En línea]. Available: https://docs.blender.org/manual/en/dev/getting_started/about/introduction.html. [Último acceso: 22 de noviembre 2021].
- [23] «Productos de GitHub - GitHub Docs,» 2021. [En línea]. Available: <https://docs.github.com/es/get-started/learning-about-github/githubs-products>. [Último acceso: 22 de noviembre 2021].
- [24] «Postman Docs,» 2021. [En línea]. Available: <https://www.postman.com/>. [Último acceso: 22 de noviembre 2021].
- [25] «¿Qué es AWS?, Amazon Web Services,» 2021. [En línea]. Available: <https://aws.amazon.com/es/what-is-aws/>. [Último acceso: 22 de noviembre 2021].
- [26] «AWS | Elastic compute cloud (EC2) de capacidad modificable en la nube,» 2021. [En línea]. Available: <https://aws.amazon.com/es/ec2>. [Último acceso: 22 de noviembre 2021].
- [27] K. Schwaber y J. Sutherland, «La Guía de Scrum,» 2017. [En línea]. Available: <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf>. [Último acceso: 22 de noviembre 2021].

- [28] A. Armijos, A. Ordonez, M. Villavicencio y C. Abad, «Aplicación de Scrum en la construcción de un simulador de Redis,» 2015.
- [29] «The new new product development game,» *Journal of Product Innovation Management*, vol. 3, nº 3, 1986.
- [30] K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, Addison-Wesley, 2012.
- [31] J. Schiel, *The ScrumMaster Study Guide*, CRC Press, 2016.
- [32] «Vuforia Developer Portal,» 2021. [En línea]. Available: <https://developer.vuforia.com/>. [Último acceso: 24 de noviembre 2021].
- [33] «Home of the Blender project - Free and Open 3D Creation Software,» 2021. [En línea]. Available: <https://www.blender.org/>. [Último acceso: 24 de noviembre 2021].
- [34] «Install the Maps SDK for Unity,» 2021. [En línea]. Available: <https://www.mapbox.com/install/unity/>. [Último acceso: 24 de noviembre 2021].
- [35] F. Davis, *Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology*, *MIS Quarterly*, 1989.

6 ANEXOS

6.1 Anexo I: Historias Épicas

Tabla 19. Historia Épica AH1

HISTORIA DE USUARIO ÉPICA	
ID: AH1	Prioridad: Alta
Título: Desarrollo de la vista de Realidad Aumentada.	
Descripción: Como usuario, necesito visualizar la información de mi visita a través de la cámara de mi celular.	

Tabla 20. Historia Épica AH2

HISTORIA DE USUARIO ÉPICA	
ID: AH2	Prioridad: Media
Título: Desarrollo de un mapa con vista aérea.	
Descripción: Como usuario, necesito ver mi ubicación en mi teléfono en un mapa y también los POIs.	

Tabla 21. Historia Épica AH3

HISTORIA DE USUARIO ÉPICA	
ID: AH3	Prioridad: Baja
Título: Publicación del servicio en una plataforma de servicios en la nube y conexión con el aplicativo.	
Descripción: Como usuario, necesito recibir la planificación de mi itinerario dependiendo de mis preferencias cuando esté conectado a internet.	

6.2 Anexo II: Historias de Usuario del Sprint 1

Tabla 22. Historia de Usuario AH1-01

HISTORIA DE USUARIO		
ID: AH1-01		Días estimados: 5
Título: Uso de una herramienta de software que implemente RA.		
Sprint N°: 1	Prioridad: Alta	Estado: Por implementar
Descripción: Como usuario, necesito visualizar los puntos de mi interés a través de la cámara de mi teléfono junto con la información de los mismos.		
Criterios de aceptación	Cuando el usuario abra la vista de RA con la cámara de su dispositivo móvil, entonces se mostrará los POIs correspondientes a su itinerario.	
	Cuando la cámara del teléfono este apuntando a la ubicación del POI, entonces el icono 3D será visible en pantalla.	

Tabla 23. Historia de Usuario AH1-02

HISTORIA DE USUARIO		
ID: AH1-02		Días estimados: 10
Título: Desarrollo de funciones de geolocalización para los POIs.		
Sprint N°: 1	Prioridad: Alta	Estado: Por implementar
Descripción: Como usuario, necesito que los POIs que observe mediante la cámara de mi teléfono se encuentren colocados de manera precisa con respecto a su ubicación real.		
Criterios de aceptación	Cuando el usuario entre al aplicativo, entonces la aplicación obtendrá los datos de ubicación y orientación del dispositivo móvil.	
	Cuando el usuario se encuentre en la vista de RA apuntando a un POI, entonces la aplicación deberá ser capaz de presentar un icono en 3D a una distancia real en metros.	
	Cuando el usuario se encuentre en el rango de aparición de un POI (70 metros), entonces el icono en 3D se ubicará lo más cercano posible a su posición real.	

6.3 Anexo III: Historias de Usuario del Sprint 2

Tabla 24. Historia de Usuario AH1-03.

HISTORIA DE USUARIO		
ID: AH1-03		Días estimados: 3
Título: Representación de POIs mediante modelos 3D.		
Sprint N°: 2	Prioridad: Alta	Estado: Por implementar
Descripción: Como usuario, necesito ver los POIs representados mediante algún objeto llamativo y que sea fácil de interpretar a través de la cámara del teléfono.		
Criterios de aceptación	Cuando el usuario se encuentre en el rango de aparición de un POI (70 metros), entonces los iconos 3D serán representados con colores llamativos para su distinción.	
	Cuando el usuario se encuentre en la vista de RA apuntando a un POI, entonces el icono 3D se desplegará a un tamaño visible para el usuario.	
	Cuando el usuario se encuentre en el rango de aparición de un POI (70 metros), entonces los iconos 3D se representarán con un símbolo relacionado al tipo de POI al que representan.	

Tabla 25. Historia de Usuario AH1-04.

HISTORIA DE USUARIO		
ID: AH1-04		Días estimados: 3
Título: Despliegue de un panel que muestre la información de los POIs.		
Sprint N°: 2	Prioridad: Alta	Estado: Por implementar
Descripción: Como usuario, necesito visualizar la información correspondiente a un POI en conjunto con su modelo.		
Criterios de aceptación	Cuando el usuario toque el icono 3D de un POI, entonces se desplegarán los datos referentes a este POI en un panel informativo.	
	Cuando el usuario toque el icono 3D de un POI mientras su panel informativo está abierto, entonces el panel informativo se ocultará.	

6.4 Anexo IV: Historias de Usuario del Sprint 3

Tabla 26. Historia de Usuario AH1-05.

HISTORIA DE USUARIO		
ID: AH1-05		Días estimados: 2
Título: Recolección de información de preferencias turísticas mediante un formulario.		
Sprint N°: 3	Prioridad: Alta	Estado: Por implementar
Descripción: Como usuario, necesito enviar la información de mis preferencias a través de un formulario simple y recibir los itinerarios personalizados.		
Criterios de aceptación	La estructura del formulario deberá adaptarse a los requerimientos de la llamada al servicio web tales como: <ul style="list-style-type: none"> • Días de recorrido • Fecha de inicio de viaje • Horarios deseados para tours y almuerzo • Preferencias turísticas 	
	Cuando el usuario llené el formulario de recolección de preferencias turísticas, entonces los campos del formulario deberán ser claros y fáciles de entender.	
	Cuando el usuario se encuentre ingresando los datos del formulario, entonces el formulario deberá ser capaz de captar las preferencias turísticas del usuario.	
	Cuando el usuario no ingrese todos los datos del formulario, entonces el formulario deberá alertar al usuario sobre los campos obligatorios.	
	Cuando el usuario envíe la información completa de sus preferencias, entonces deberá aparecer una pantalla de carga.	

Tabla 27. Historia de Usuario AH2-01.

HISTORIA DE USUARIO		
ID: AH2-01		Días estimados: 5
Título: Integración de herramienta de software para la representación de mapas.		
Sprint N°: 3	Prioridad: Media	Estado: En desarrollo
Descripción: Como usuario, necesito visualizar mi ubicación en tiempo real y la ubicación de los POIs en un mapa.		
Criterios de aceptación	Cuando el usuario ingrese a la vista de mapa, entonces la aplicación deberá ser capaz de presentar un mapa 2D con vista aérea.	
	Cuando el usuario ingrese a la vista de mapa, entonces se desplegará el mapa 2D centrado en la ubicación del usuario.	

Tabla 28. Historia de Usuario AH2-02.

HISTORIA DE USUARIO		
ID: AH2-02		Días estimados: 15
Título: Desarrollo de las funcionalidades requeridas en la vista de mapa.		
Sprint N°: 3	Prioridad: Media	Estado: Por implementar
Descripción: Como usuario, necesito tener un mapa interactivo en el que me pueda desplazar y hacer zoom.		
Criterios de aceptación	Cuando el usuario se desplace y sus coordenadas geográficas cambien, entonces el mapa 2D se actualizará a su nueva ubicación.	
	Cuando se visualice el mapa 2D, entonces los POIs se presentarán como iconos dentro del mapa.	
	Cuando se visualice el mapa 2D en la ubicación del usuario, entonces el icono del usuario se presentará dentro del mapa.	
	Cuando el usuario desee controlar el desplazamiento y la vista por el mapa, entonces podrá hacerlo con los elementos de interfaz de usuario.	

Tabla 29. Historia de Usuario AH3-02.

HISTORIA DE USUARIO		
ID: AH3-02		Días estimados: 5
Título: Desarrollo de las funciones para el consumo del servicio web.		
Sprint N°: 3	Prioridad: Baja	Estado: Por implementar
Descripción: Como usuario, necesito obtener mi itinerario tras haber llenado el formulario de preferencias turísticas estando conectado a internet.		
Criterios de aceptación	Cuando el dispositivo móvil se encuentre conectado a internet, entonces el usuario podrá usar el aplicativo en su totalidad.	
	Cuando el aplicativo realice una petición al servicio web, entonces se recibirán los itinerarios turísticos.	

6.5 Anexo V: Historias de Usuario del Sprint 4

Tabla 30. Historia de Usuario AH2-03.

HISTORIA DE USUARIO		
ID: AH2-03		Días estimados: 5
Título: Implementación de rutas entre el usuario y los POIs en la vista de mapa.		
Sprint N°: 4	Prioridad: Media	Estado: Por implementar
Descripción: Como usuario, necesito visualizar la ruta que existe entre mi ubicación y un POI que haya seleccionado dentro del mapa.		
Criterios de aceptación	Cuando el usuario seleccione un POI en la vista de mapa, entonces el mapa 2D presentara la ruta más cercana al POI seleccionado.	
	Cuando se cambie la selección de un POI en la vista de mapa, entonces se eliminará las rutas anteriores y únicamente se presentará la ruta al POI seleccionado.	
	Cuando el usuario cambie el valor del zoom superando las 8 unidades, entonces la ruta al POI seleccionado no se mostrará.	
	Cuando el usuario seleccione la opción de ver todos los POIs, entonces no se presentará ninguna ruta.	
	Cuando que el usuario avance por el mapa, entonces la ruta se actualizará.	
	Cuando el POI salga de la pantalla del usuario en la vista de mapa, entonces la ruta desaparecerá.	

Tabla 31. Historia de Usuario AH3-01.

HISTORIA DE USUARIO	
ID: AH3-01	Días estimados: 5
Título: Publicación del proyecto de titulación predecesor en una plataforma de servicios en la nube.	
Sprint N°: 4	Prioridad: Baja Estado: Por implementar
Descripción: Como usuario, necesito que mi aplicativo funcione al estar conectado a internet	
Criterios de aceptación	Cuando el servicio este alojado en la nube mediante una plataforma que le asigne una dirección IP publica, entonces el aplicativo podrá funcionar en su totalidad.
	Cuando un dispositivo móvil tenga el aplicativo instalado y conexión a internet, entonces el aplicativo estará en la capacidad de realizar consultas al servicio web.

6.6 Anexo VI: Diseño de las interfaces de usuario final

Formulario

The image shows a mobile app interface for a 'Tourist trip design problem' form. The form is displayed on a smartphone screen and includes the following elements:

- Title:** Tourist trip design problem
- Total Days:** Input field with the value '2'.
- Start Days:** Input field with the value '2021/10/21'.
- Desired schedule for tours:**
 - Start:** Input field with the value '09:00'.
 - End:** Input field with the value '16:00'.
- Desired schedule for lunch:**
 - Start:** Input field with the value '13:00'.
 - End:** Input field with the value '14:00'.
- Place of preference:** A grid of 12 checkboxes for different locations:
 - Aquariums
 - Premise
 - Churches
 - Cementeries
 - Art Galleries
 - Zoo
 - Mosques
 - City Halls
 - Parks
 - Museums
 - Stadiums
 - Amusement Park
 - Indu Temples
- Action:** A button labeled 'Create Itineraries' at the bottom of the form.

Figura 56. Diseño del formulario para la toma de datos (Realizado en MockFlow)

Pantalla de Realidad Aumentada

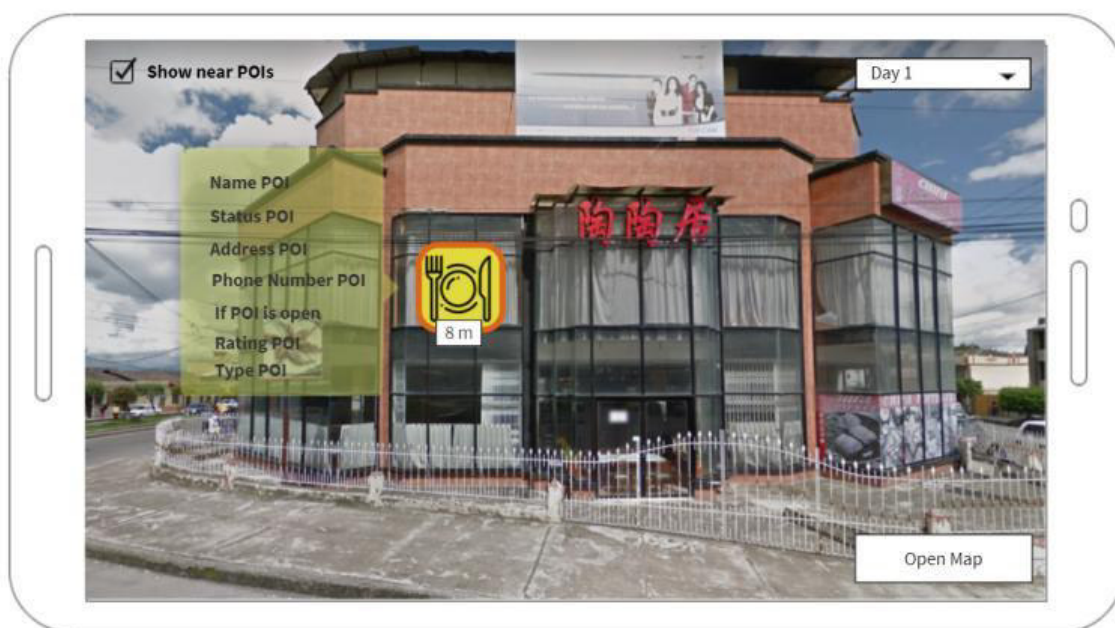


Figura 57. Diseño de la pantalla de RA (Realizado en MockFlow)

Pantalla de Mapa 2D

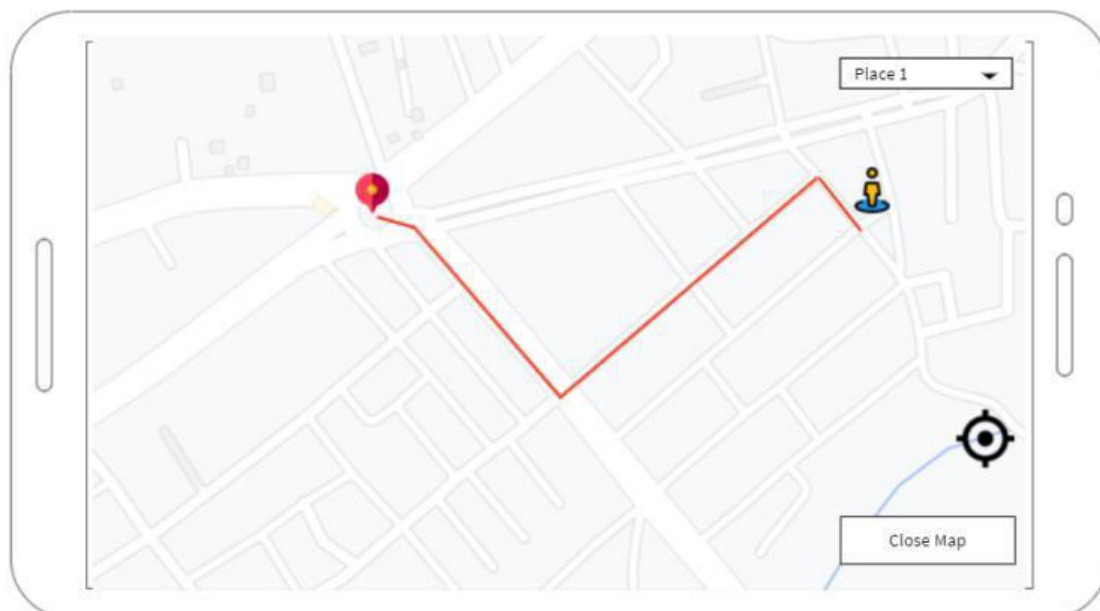


Figura 58. Diseño de la pantalla de Mapa en 2D (Realizado en MockFlow)

6.7 Anexo VII: Encuestas Realizadas

Formulario de RA

Formulario de Realidad Aumentada								
¿Considera que los elementos de Realidad Aumentada mejoran la experiencia turística?								
	1	2	3	4	5	6	7	
Para nada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente
¿Le parece útil que la interfaz de Realidad Aumentada complemente a la interfaz de mapas 2D?								
	1	2	3	4	5	6	7	
Para nada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente
¿Consideraría utilizar aplicaciones que integren Realidad Aumentada en un futuro cercano?								
	1	2	3	4	5	6	7	
Para nada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente

Figura 59. Formulario para medir aspectos de RA

Formulario de utilidad percibida

Formulario de Utilidad Percibida								
¿Le resultó útil la información que se muestra para cada punto de interés?								
	1	2	3	4	5	6	7	
Para nada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente
Tras usar el aplicativo, ¿Mejóro su experiencia de visita en la ciudad?								
	1	2	3	4	5	6	7	
Para nada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente
¿Utilizaría o recomendaría el aplicativo a otra persona para mejorar la experiencia durante un viaje turístico?								
	1	2	3	4	5	6	7	
Para nada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente

Figura 60. Formulario para medir la utilidad percibida.

Formulario de facilidad de uso

Formulario de Facilidad de Uso								
¿Los iconos en 3D tiene un tamaño correcto y son entendibles visualmente?								
	1	2	3	4	5	6	7	
Para nada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente
¿Le resulta sencillo desplazarse entre las distintas pantallas de la aplicación?								
	1	2	3	4	5	6	7	
Para nada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente
¿Le resultó sencillo obtener la información del punto de interés que desea?								
	1	2	3	4	5	6	7	
Para nada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente

Figura 61. Formulario para medir la facilidad de uso.

Resultados de las encuestas

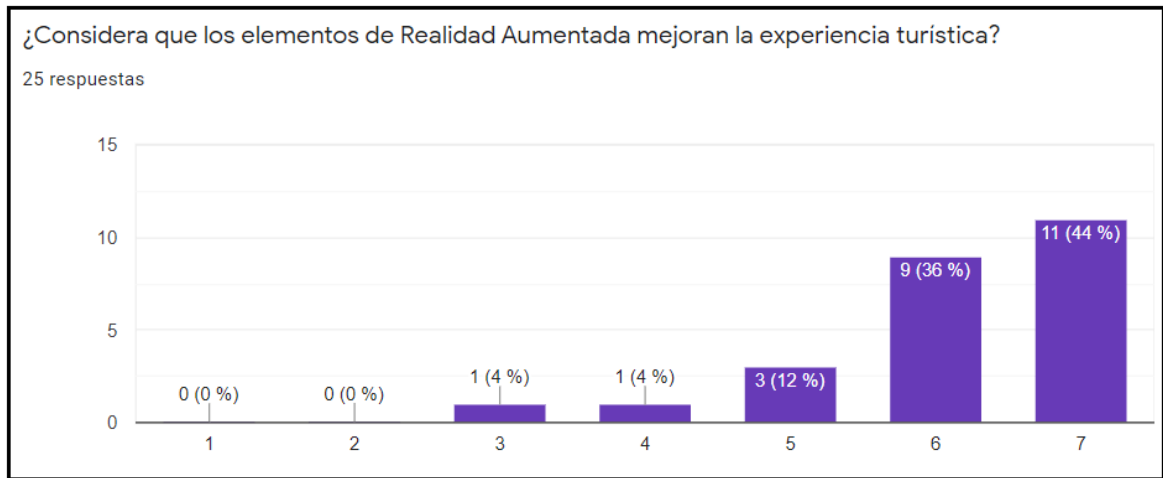


Figura 62. Respuestas para la pregunta 1 de RA.

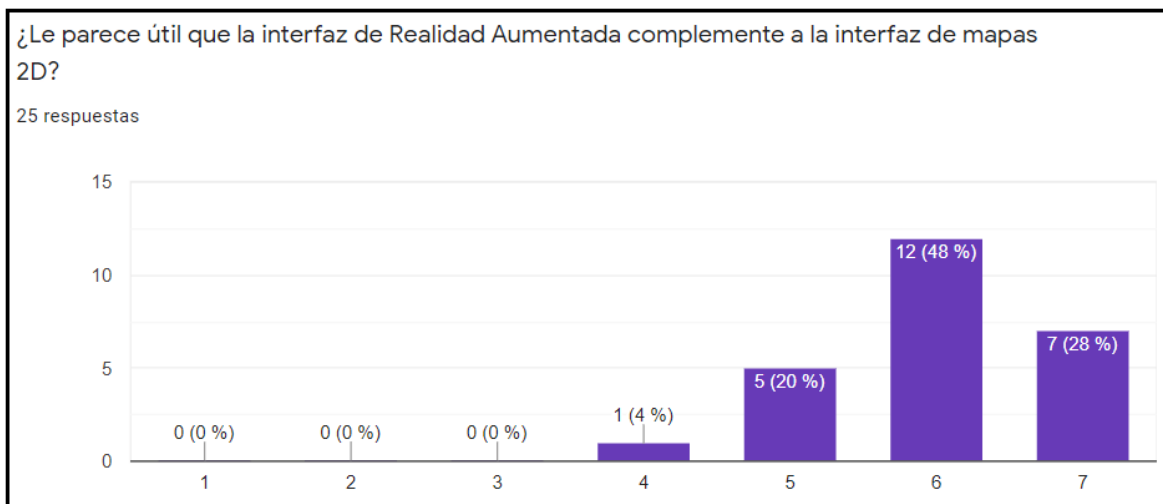


Figura 63. Respuestas para la pregunta 2 de RA.

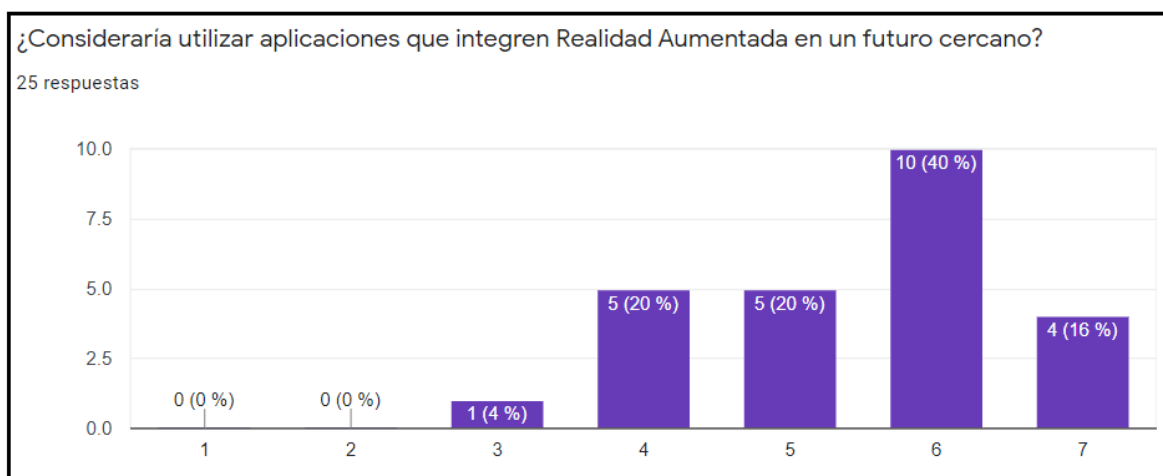


Figura 64. Respuestas para la pregunta 3 de RA.

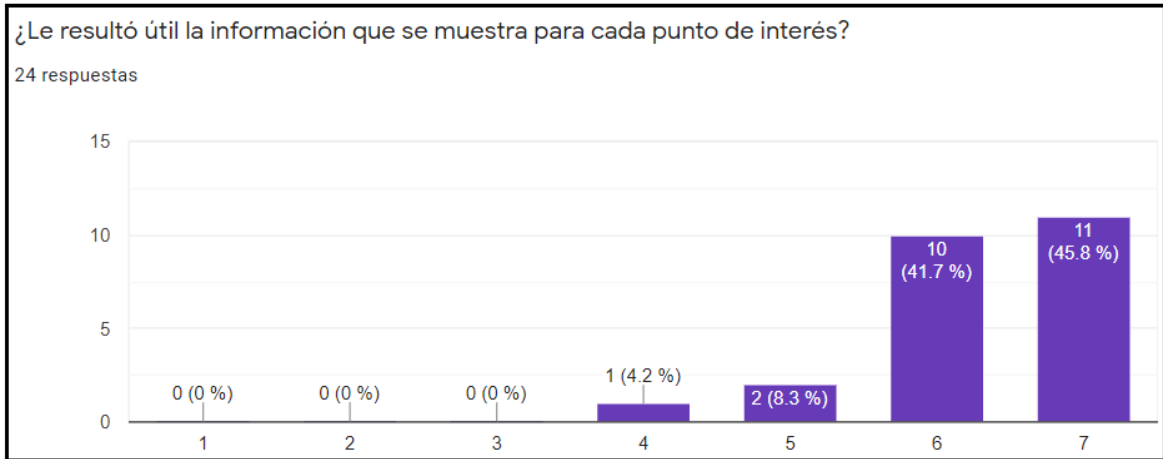


Figura 65. Respuestas para la pregunta 1 de utilidad percibida.

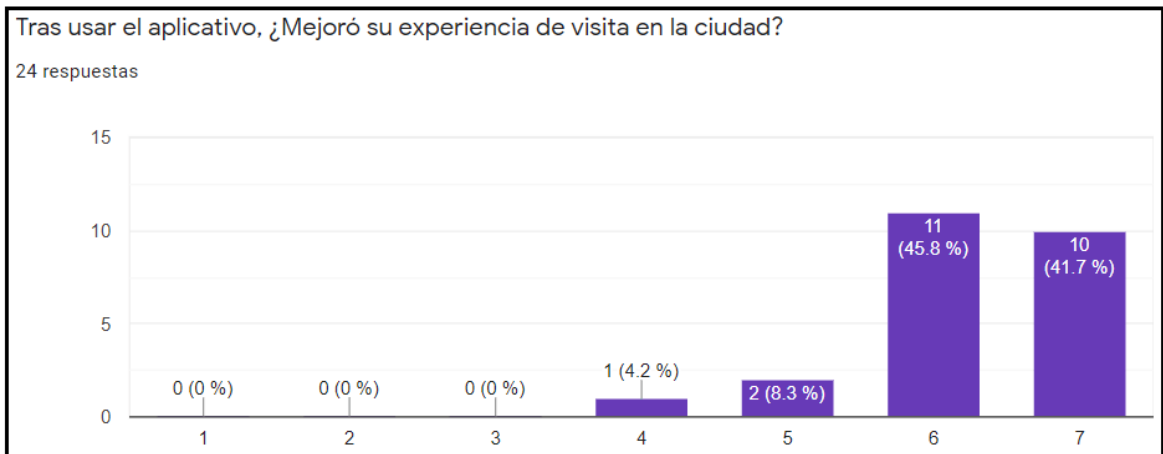


Figura 66. Respuestas para la pregunta 2 de utilidad percibida.

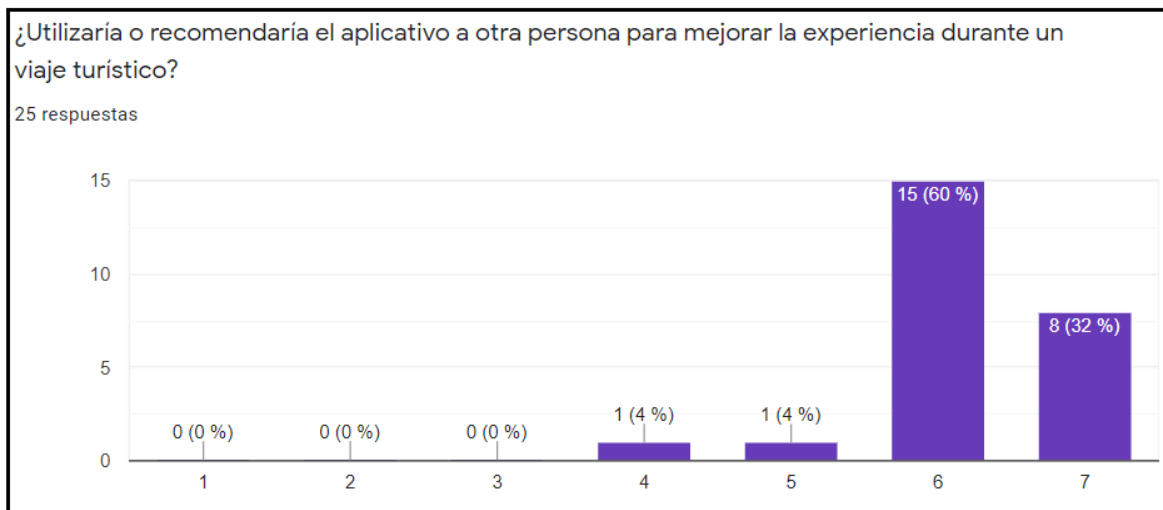


Figura 67. Respuestas para la pregunta 3 de utilidad percibida.

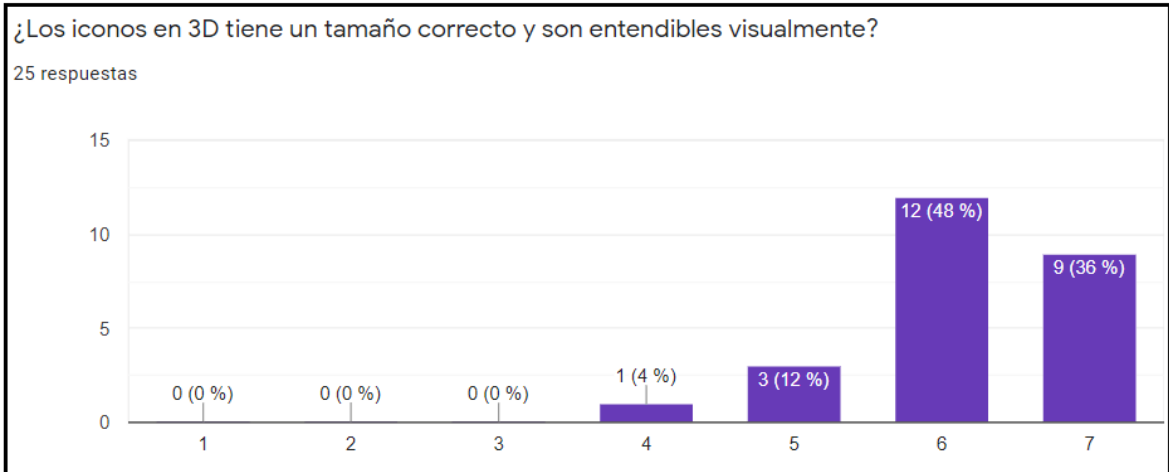


Figura 68. Respuestas para la pregunta 1 de facilidad de uso.

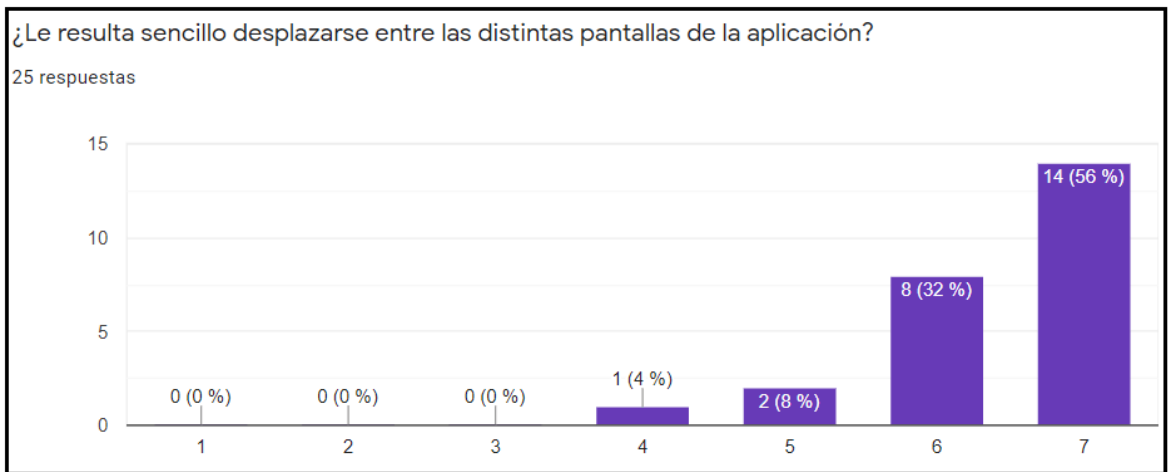


Figura 69. Respuestas para la pregunta 2 de facilidad de uso.

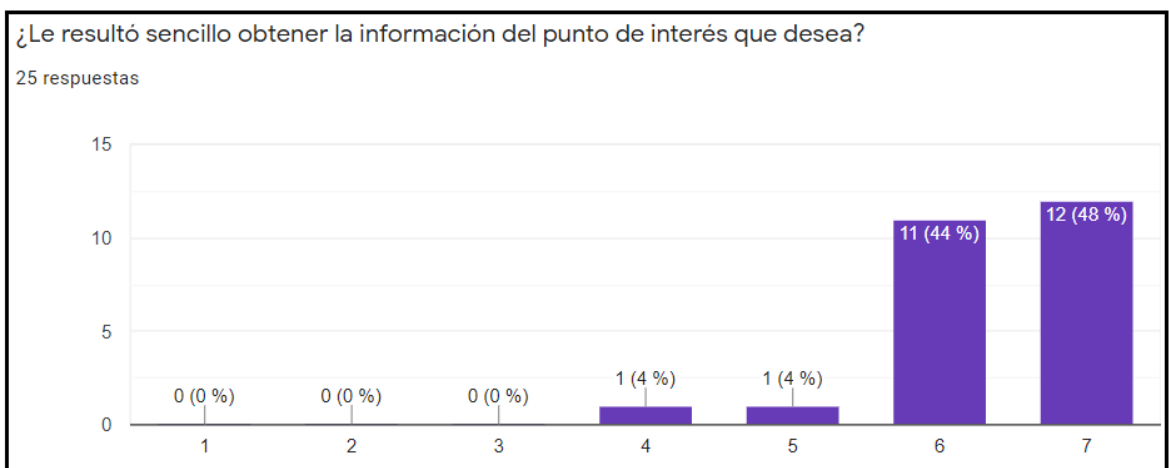


Figura 70. Respuestas para la pregunta 3 de facilidad de uso.