

**APLICACIÓN MÓVIL PARA LA LOCALIZACIÓN DE ACTIVIDADES DE TURISMO NÁUTICO
EN JUANCHACO, LADRILLEROS Y LA BARRA CON SISTEMA DE RECOMENDACIÓN
UTILIZANDO TÉCNICAS DE MACHINE LEARNING**



NICOLÁS ROJAS BARRIOS
1526680
rojas.nicolas@correounivalle.edu.co

SHARON ANDREA ROJAS
1532273
Sharon.rojas@correounivalle.edu.co

Universidad del Valle Sede Cali
Facultad de Ingeniería
Escuela de Ingeniería Civil y Geomática
Programa Académico de Ingeniería Topográfica
Cali, febrero 26 de 2020

APLICACIÓN MÓVIL PARA LA LOCALIZACIÓN DE ACTIVIDADES DE TURISMO NÁUTICO EN
JUANCHACO, LADRILLEROS Y LA BARRA CON SISTEMA DE RECOMENDACIÓN UTILIZANDO
TÉCNICAS DE MACHINE LEARNING



NICOLÁS ROJAS BARRIOS
1526680
rojas.nicolas@correounivalle.edu.co

SHARON ANDREA ROJAS
1532273
Sharon.rojas@correounivalle.edu.co

Director Fabio Andrés Herrera Roza MSc.
Fabio.herrera@correounivalle.edu.co

Codirector Paola J. Rodríguez C. PhD
paola.rodriguez@correounivalle.edu.co

Trabajo de grado para optar el título de profesional en Ingeniería
Topográfica

Universidad del Valle Sede Cali
Facultad de Ingeniería
Escuela de Ingeniería Civil y Geomática
Programa Académico de Ingeniería Topográfica
Cali, febrero 26 de 2020

Nota de aceptación

Presidente jurado

Presidente jurado

Presidente jurado

Santiago de Cali, 26 de febrero del 2021

Agradecimientos

A nuestra directora Paola Johanna Rodríguez, que con su conocimiento y dedicación nos apoyó en cada fase de este proyecto, siempre brindándonos los mejores consejos y estando siempre atenta a todas nuestras dudas en las que nos orientó con paciencia y empeño, ayudándonos a solucionar cada problema que se podía presentar, haciendo posible lograr los resultados que esperábamos.

A nuestro director Fabio Herrera, que nos aconsejó en momentos críticos, permitiendo lograr de forma mucho más rápida los objetivos planteados; además sus consejos nos ayudaron a mejorar en diversos aspectos al momento de realizar mejorar a la aplicación.

Al profesor Robin Alexis Olaya, que, a pesar de no estar en todas las fases del proyecto, nos guió ampliamente en la conceptualización y el planteamiento de objetivos y actividades, que fue fundamental para el progreso de este trabajo de grado.

Al profesor Wilfredo Alfonso Morales, que nos guió de manera positiva al momento de realizar el sistema de recomendaciones, sus conocimientos y consejos fueron de gran ayuda para el desarrollo del proyecto.

A la Universidad del Valle, por ser una institución sin igual, por darnos todo el conocimiento con el que vamos a emprender una vida laboral y estar a disposición de nosotros siempre. Apoyando a jóvenes como nosotros en el crecimiento profesional.

Dedicatorias

Dedicatorias Nicolás Rojas Barrios

A mi compañera Sharon Andrea Rojas, por su apoyo y compromiso, porque sin sus consejos que siempre me ayudaron a mejorar no habría sido posible lograr terminar este proyecto. Tus ideas de mejora siempre fueron completamente acertadas y ayudaron a que la aplicación desarrollada sea lo que es ahora.

A mi hermana Karen Melissa Rojas y mi primo Sergio Restrepo, que fueron, son y serán mi inspiración y modelos para seguir por las etapas posteriores de mi vida y mi inspiración para mejorar cada día.

En general, a mi familia, con quienes siempre puedo contar y recibir su apoyo, consejos, e inspiración y debido a ello gracias a ellos siempre tendré confianza en que, sin importar los días malos, tendré personas en quien puedo confiar en todo momento para seguir adelante.

A mis compañeros, con los que logré conformar un grupo extraordinario, con quienes puedo contar siempre, en cualquier aspecto. En definitiva son las mejores personas que he conocido y gracias a ellos he vivido momentos increíbles, que solo han sido posibles gracias a ellos.

Dedicatorias Sharon Andrea Rojas Carvajal

A mi compañero Nicolas Rojas Barrios, por su dedicación y apoyo incondicional, a pesar de querer realizar este trabajo de grado en el menor tiempo posible, supimos juntos tener paciencia para realizar un buen proyecto y dar nuestro mayor esfuerzo.

A mi familia, mi madre Adriana Carvajal que hizo hasta lo imposible para que yo saliera adelante y tener un estudio universitario. Dándome las herramientas y las fuerzas para nunca desistir de mis metas. A mi hermano Miguel Rojas que siempre con mucha paciencia estuvo para apoyarme en momentos difíciles.

A mi abuela, Aleyda Torres, que a pesar de no estar aquí para ver un triunfo por el que tanto se esforzó, se que estas a mi lado siempre dándome muchas fuerzas y brindándome todo su amor, porque fue por ti que me proyecte a crecer, a tener metas y poder cumplirlas. Gracias Mamita.

A mis compañeros de estudio que, si no fuera por las largas horas de estudio, por el apoyo que nos dimos unos a otros para no dejarnos caer durante toda la carrera no hubiera podido llegar hasta esta etapa. Fueron los mejores compañeros que el mundo me pudo dar. Gracias.

Al ingeniero Jhon Orozco, que me aconsejó mucho durante gran etapa del trabajo de grado, que me enseñó mucho sobre cosas de las que me sentía muy perdida y me angustiaba, pero estuvo ahí, con paciencia y sabiduría guiándome para poder llegar hasta aquí. Gracias.

Tabla de Contenido

Agradecimientos.....	4
Dedicatorias	5
Resumen	12
CAPITULO 1. DESCRIPCIÓN DEL TRABAJO DE GRADO	13
1.1. Introducción	13
1.2. Planteamiento y Formulación del Problema	13
1.3. Justificación del Problema.....	15
1.4. Objetivos	15
1.4.1. Objetivo General	16
1.4.2. Objetivos Específicos.....	16
1.5. Resultados del Trabajo de Grado.....	16
1.6. Alcances de la Propuesta.....	17
CAPITULO 2. MARCO REFERENCIAL.....	17
2.1 Marco Teórico.....	17
Turismo.....	17
Sistemas de Información Geográfica	18
Bases de Datos.....	18
Bases de Datos en Tiempo Real (Firebase).....	19
Método de ruteo.....	19
Aplicaciones Móviles.....	20
Inteligencia Artificial	23
Sistemas de Recomendaciones.....	23
Scikit-learn.....	24
Algoritmo A-star.....	24
Método Djikstra	24
Red Neuronal.....	25
Arboles de Decisión.....	25

Métricas de rendimiento.....	26
2.2 Antecedentes o Estado del Arte	27
CAPITULO 3. DISEÑAR UNA PRUEBA PILOTO DE UN SISTEMA DE INFORMACIÓN GEOGRÁFICA (SIG) EN LAS ZONAS DONDE SE REALIZAN LAS DIFERENTES ACTIVIDADES NÁUTICAS, TENIENDO EN CUENTA SUS CARACTERÍSTICAS Y LA CERCANÍA A OTROS SITIOS DE INTERÉS.....	31
3.1 Definición de la zona de Estudio:.....	31
3.2 Levantamiento de información de turistas.	32
3.3 Análisis de requerimientos.....	34
3.4 Análisis de almacenamiento.....	35
3.5 Requerimientos Funcionales	37
3.6 Diagramas de casos de uso.....	39
3.7 Desarrollo del sistema de información geográfica	41
3.8 Adquisición de Información:.....	43
3.9 Levantamiento de datos de campo	44
3.10 Implementación de la base de datos	44
3.11 Implementación de la base de datos	47
CAPITULO 4. IMPLEMENTAR EL SIG EN UN APLICATIVO MÓVIL EN ANDROID, CON LA INFORMACIÓN ESPACIAL DEL TURISMO	48
4.1 Metodología de desarrollo de software Scrum	48
4.2 Módulo de gestión de usuarios	53
4.2.1 Análisis.....	53
4.2.2 Implementación.....	55
4.3 Módulo de gestión espacial.....	58
4.3.1 Análisis.....	58
4.3.2 Implementación.....	59
CAPITULO 5. DESIGNAR UN MÓDULO EN QUE EL USUARIO COMO ADMINISTRADOR PUEDA ACTUALIZAR LA INFORMACIÓN EN TIEMPO REAL Y REALIZAR RESPECTIVOS MANTENIMIENTOS.....	63
5.1 Módulo de Administrador	63
5.1.1 Análisis.....	64
5.1.2 Implementación.....	65
CAPITULO 6. GENERAR UN SISTEMA DE RECOMENDACIONES A PARTIR DE LAS PREFERENCIAS DEL USUARIO, EMPLEANDO TÉCNICAS DE MACHINE LEARNING.....	72
6.1 Módulo del Sistema de Recomendaciones.	73

6.1.1	Análisis.....	73
6.1.2	Implementación.....	76
	PRUEBAS DE USABILIDAD.....	79
	DISCUSIÓN DE RESULTADOS	85
	CONCLUSIONES Y TRABAJO FUTURO.....	86
	Conclusiones.....	86
	Trabajo Futuro	87
	REFERENCIAS.....	89
	ANEXOS	93

Tabla de Figuras

Figura 1. Ubicación de las playas de Juanchaco, Ladrilleros y La Barra.....	32
Figura 2. Conocimiento de las actividades que ofrece la playa.....	33
Figura 3. Herramienta que más utiliza para conocer la ruta a su destino deseado.....	33
Figura 4. Considera que el conocimiento de las actividades que ofrecen las playas le sería más fácil viajar y conocer.....	34
Figura 5. Diagrama de caso de uso para el inicio en el sistema.	39
Figura 6. Diagrama de caso de uso para las funciones que tiene el actor Administrador en el sistema.	40
Figura 7. Diagrama de caso de uso para las funciones del actor Usuario en el sistema.....	41
Figura 8. Modelo Entidad-Relación para el desarrollo del SIG.	42
Figura 9. Diagrama Relacional para el desarrollo del SIG.....	43
Figura 10. Cartografía de las ubicaciones de los diferentes tipos de actividades turísticas que ofrece las playas de Juanchaco, Ladrilleros y La Barra.	45
Figura 11. Cartografía de las ubicaciones de restaurantes y hoteles que se encuentran en las playas de Juanchaco, Ladrilleros y La Barra.	46
Figura 12. Base de datos, tabla de las actividades que ofrecen las playas, con sus atributos.....	47
Figura 13. Creación de la base de datos.....	48
Figura 14. Metodología de desarrollo de software Scrum.	49
Figura 15. Boceto creado para la interfaz del inicio de sesión.	52
Figura 16. Flujograma del inicio de sesión para el ingreso al sistema.....	54
Figura 17. Interfaz de registro de usuario. Código de validación.....	55
Figura 18. Interfaz del inicio de sesión y del mapa principal, con sitios de interés.....	56
Figura 19. Interfaz de la configuración de datos del usuario, menú de la aplicación.	58
Figura 20. Archivo en formato PHP para la conversión de los datos a Geojson.....	60
Figura 21. Visualización de las burbujas para las actividades, restaurantes y hoteles.....	61
Figura 22. Ejemplo de la ruta que es generada por el sistema.....	63
Figura 23. Diagrama de caso de uso de las funciones que puede realizar cada tipo de usuario.	65
Figura 24. Interfaz de comprobación de usuario, interfaz para agregar o modificar datos del mapa.	66
Figura 25. Interfaz del tipo de cambio, interfaz para la elección de tipo de datos para agregar.....	67
Figura 26. Interfaz de la ventana para agregar una nueva actividad turística. Interfaz del mapa con el marcador nuevo.....	68
Figura 27. Interfaz de la ventana para actualizar una actividad turística. Interfaz para la visualización del dato nuevo.....	69
Figura 28. Interfaz de la ventana para eliminar una actividad turística. Interfaz del mapa demostrando que se eliminó una actividad.....	71
Figura 29. Interfaz de la ventana de información. Ventana de Gmail para enviar correo solicitando información.	72
Figura 30. Código para determinar la precisión del modelo.....	76
Figura 31. Resultados de las métricas en los tres modelos evaluados.....	77
Figura 32. API donde se ejecuta el sistema de recomendación.	78

Figura 33. Interfaz del formulario para la recomendación. Interfaz de las recomendaciones que dio el sistema.	
Interfaz del mapa donde se visualizan las actividades de una de las recomendaciones seleccionada.	79
Figura 34. Edad de los usuarios encuestados de la prueba de usabilidad.....	80
Figura 35. Veces en el año que el usuario realiza viajes de turismo.	81
Figura 36. Pregunta sobre la utilización de un dispositivo móvil para orientarse en el transcurso del viaje.	81
Figura 37. Pregunta de si el usuario pudo realizar las consultas que ofrece la aplicación de los diferentes lugares.	82
Figura 38. Pregunta de si el usuario considera la información suministrada, a través de la aplicación, completa.	82
Figura 39. Pregunta de si al usuario le pareció fácil la navegación (desplazamientos entre pantallas) a través de la aplicación.	83
Figura 40. Pregunta de si al usuario pudo obtener buenas recomendaciones de las actividades turísticas.	83
Figura 41. Pregunta de si el usuario considera que el diseño de la aplicación es consistente.	84
Figura 42. Pregunta de si la aplicación fue de su agrado y satisfacción.	84
Figura 43. Código de registro de usuario en la base de datos Firebase.....	95
Figura 44. Código de inicio de sesión.....	96
Figura 45. Código de actualización de datos de usuario.	97
Figura 46. Código para la visualización de la información de los marcadores al ser clicados.....	98
Figura 47. Código para generar la ruta.	99
Figura 48. Código de acceso para la modificación de datos de la base de datos.....	100
Figura 49. Código para el envío de datos a la API.	101
Figura 50. Código de la API que procesa y envía datos a la base de datos.....	102

Tabla de Tablas

Tabla 1. Resultados del Trabajo de Grado.	16
Tabla 2. Análisis de los objetivos planteados en este proyecto.	34
Tabla 3. Requerimientos de Información almacenada.	35
Tabla 4. Requerimientos Funcionales.....	37
Tabla 5. Los actores que podrán interactuar en el sistema.	38
Tabla 6. Product Backlog.....	49
Tabla 7. Release Plan.....	50
Tabla 8. Historia de Usuario de las actividades náuticas.	51
Tabla 9. Historia de usuario del registro de usuarios en el sistema.	53
Tabla 10. Historia de usuario del visualizador de actividades náuticas.	59
Tabla 11. Historia de usuario de agregar nueva ubicación de actividad, hotel o restaurante.	64
Tabla 12. Historia de usuario para el sistema de recomendaciones.....	75

Tabla de Anexos

Anexo A. Estructura de la encuesta realizada a los turistas en el municipio de Buenaventura.	93
Anexo B. Código del registro a la base de datos.	94
Anexo C. Código del inicio de sesión.	95
Anexo D. Código de la configuración de datos del usuario.	96
Anexo E. Código para la visualización de la información de los marcadores.	97
Anexo F. Código para la generación de la ruta.	98
Anexo G. Código de acceso para la modificación de datos de la base de datos.	99
Anexo H. Código para el envío de los datos a la API.	100
Anexo I. Código para el procesamiento y envío de los datos.	101
Anexo J. Estructura de la encuesta realizada a personas que realizan visitas usualmente visitas a sitios turísticos.	102

Resumen

En el marco del proyecto Aplicación móvil para la Localización de Actividades de Turismo Náutico en Juanchaco, Ladrilleros y La Barra con Sistema de Recomendación utilizando Técnicas de Machine Learning se presenta el desarrollo de la aplicación Navi_App, con el fin de apoyar a los turistas para tener su ubicación en tiempo real y la posibilidad de guiarse en su viaje con mayor seguridad y eficacia, con información de actividades náuticas que se pueden realizar en Juanchaco, Ladrilleros y La barra. Se realizó debido al crecimiento en el uso de aplicaciones móviles en el turismo y el desconocimiento sobre las atracciones turísticas en el municipio de Buenaventura. Esta aplicación cuenta con herramientas de visualización y gestión, para actividades, hoteles y restaurantes de la zona, además, de una herramienta de recomendación de actividades náuticas.

Inicialmente, se hizo una recopilación de información, con la que se generó la base de datos, por medio del software PostgreSQL permitiendo el almacenamiento, procesamiento, análisis y modelación de información. A partir de esta base de datos, se desarrolló la aplicación mediante la IDE Android Studio®, y con herramientas como Mapbox, Firebase, Heroku, entre otros, integrando varias API's que permitieron la gestión de sitios turísticos y las recomendaciones de actividades. De esta forma, se realizó la aplicación en 4 módulos: gestión de usuarios, gestión de información espacial, administrador y recomendaciones, haciendo posible la interacción con una herramienta dinámica que permite al turista estar informado sobre las actividades que ofrecen las playas de Juanchaco, Ladrilleros y La barra. Finalmente, se realizó una prueba de usabilidad a un total de 9 usuarios, logrando identificar las mejoras que se deben realizar a futuro en aspectos estéticos, funcionales y de información, haciendo de la aplicación lo más completa e intuitiva posible.

Palabras clave – Turismo, Buenaventura, Sistemas de Información Geográfica, Aplicaciones móviles, Android, Sistemas de recomendaciones, Machine Learning.

CAPITULO 1. DESCRIPCIÓN DEL TRABAJO DE GRADO

1.1. Introducción

Hoy en día, el turismo es una de las actividades que generan más ingresos debido a su continuo crecimiento y la alta diversificación de viajeros. Se puede decir entonces que esta actividad es muy importante para el país dado que podría igualar o hasta superar a la industria del petróleo. Adicionalmente, incentiva diferentes actividades como la agricultura, la construcción, entre otros que funcionan con relación a esta (World Tourism Organization, n.d.).

En la actualidad, el uso de aplicaciones web y móviles ayuda a los turistas en diferentes aspectos como la ubicación, formas de llegar a un destino, entre otros aspectos que permiten conocer con mayor profundidad el turismo de una región. Estas tecnologías se han visto muy implementadas con el uso de sistemas de información geográfica (SIG); en el mercado se encuentran diferentes programas y herramientas como ArcGIS, QuantumGIS, PostGIS, etc., que son útiles para la generación de aplicaciones web que promocionan el turismo. El uso de estas aplicaciones se observa comúnmente en las ciudades grandes, sin embargo, en ciudades pequeñas no es muy marcado dicho uso para ayudar a los turistas a conocer los lugares que hay en el municipio (Nieto Acevedo. 2014).

Así mismo, la ingeniería topográfica con su capacidad de obtener, organizar, analizar y representar datos puede crear cartografía digital de rápido y fácil acceso, en la que los usuarios puedan encontrar actividades en su destino turístico. En esta ingeniería, se tiene como uno de los principales campos los sistemas de información geográfica, que sirven para conocer la información espacial de las diferentes actividades que brinda el municipio, teniendo como principal enfoque la eficacia al conseguir información segura y precisa de los lugares de su interés, además, el SIG se puede vincular a una aplicación web o móvil, que es la herramienta más utilizada en los últimos años.

El objetivo de este proyecto fue crear una herramienta que pueda permitir impulsar el turismo en las zonas de Buenaventura, concretamente en las playas de Juanchaco, Ladrilleros y La Barra, tal herramienta se realizó a partir de la implementación de un aplicativo móvil, utilizando técnicas de inteligencia artificial. Todo esto, debido a que en Buenaventura hay muy poca difusión de los lugares turísticos que se pueden visitar, por el mal manejo de los medios de comunicación y la desinformación por parte de los turistas y los mismos habitantes, problemas de infraestructura, etc. (Torre. 2017).

1.2. Planteamiento y Formulación del Problema

El turismo es un sistema complejo conformado por procesos ecológicos, políticos, económicos, sociales y culturales de las comunidades y regiones turísticas (Salinas. 2006). Este tiene diferentes características en las que se pueden interesar las personas, dependiendo del tipo de actividad. Por ejemplo, el turismo náutico, que es una de las modalidades que ha generado mayor interés puesto que posee actividades que tienen alto contacto con el agua y un encuentro con la naturaleza, asimismo, pueden llegar a ser deportivas o lúdicas (Macaronesia. 2015).

En los últimos años, Colombia ha tenido grandes avances en el sector turístico, de tal manera que se ha posicionado como el tercer generador de divisas en el país, después del petróleo y el carbón, generando US\$ 2600 de los 4600 que llegan en el sector de servicios (Montoya. 2016). De acuerdo con (Díaz Granados. 2013), el 70\% de las actividades del turismo náutico pertenecientes a Colombia son realizadas por los habitantes locales, destacándose zonas como: La Guajira, el Caribe Metropolitano, El Archipiélago de San Andrés y Providencia, el Litoral Pacífico, al igual que otras regiones del país. Por esta razón, en Colombia se han buscado propuestas para impulsar este turismo, como proyectos de infraestructura hotelera, la construcción de la guía náutica turística de Colombia, entre otras, sin embargo, estas propuestas no han alcanzado el propósito que se tiene para los próximos años.

Adicionalmente, los entes administrativos del Valle del Cauca están buscando impulsar la llegada de turistas, con la finalidad de triplicar la cantidad registrada en el 2016, un propósito con vista hacia el 2020. Entre las estrategias implementadas para fortalecer este sector, se encuentran la creación de una oficina encargada de promocionar al Valle del Cauca y la articulación de los diferentes actores de la cadena productiva del turismo (Franco. 2016).

En el municipio de Buenaventura existe poca difusión de los diferentes atractivos turísticos a nivel nacional e internacional, la información suministrada a través de los medios de comunicación utilizados hasta ahora es incompleta y negativa en cuanto a la seguridad y precios del lugar, etc. (Mayki. 2014). De acuerdo con (Alzate. 2014) en una publicación del resumen ejecutivo de Buenaventura, para las empresas que ofrecen actividades turísticas en Buenaventura es fundamental el uso del internet para promocionar sus ventas, y solo el 14,3\% de los empresarios encuestados y dedicados al turismo tienen página web. Según datos presentados por (Valencia. 2016), en Buenaventura se realizó una recopilación estadística de las actividades practicadas por las personas que visitan el municipio, con una duración aproximada de 3 meses, donde se encontró que más del 90\% de turistas no tienen conocimiento de las actividades culturales y sitios emblemáticos de la ciudad.

Entre los principales destinos que se encuentran en Buenaventura están Juanchaco, Ladrilleros y La Barra, estos son exóticos lugares con impresionantes atractivos como lo son la llegada de las ballenas jorobadas entre los meses de julio y octubre, las viviendas artesanales de las tribus indígenas pertenecientes a esta región, playas de marea baja, ideales para turistas y amantes del mar, entre otros (SoyValle. S.f.). Sin embargo, el 41\% de los turistas extranjeros desconoce la oferta de alojamiento, restaurantes y actividades que ofrecen estos lugares por falta de promoción, según el plan de desarrollo turístico del Valle de Cauca, (Huitonava. 2012). Por consiguiente, como afirma (Torre. 2017), estos destinos son poco conocidos por los turistas extranjeros y nacionales, debido al poco interés por parte de los empresarios en el desarrollo turístico de la zona, y el desconocimiento de extensas actividades turísticas representativas, haciendo evidente el abandono en esta región, que se manifiesta en tasas de desempleo y pobreza del 60\% y 80\% respectivamente, falta de acueducto y alcantarillado, etc.

La mayoría de las agencias de viaje no tiene un conocimiento amplio de las nuevas tecnologías de geolocalización por lo que no pueden ofrecer efectivamente las diferentes actividades turísticas que brinda el municipio, esto conlleva a que los visitantes no tengan un punto de referencia dentro de la geografía conocida por ellos, desaprovechando las riquezas visuales, culturales y sociales de la localidad. Muchos de los sitios donde se realizan este tipo de actividades no están identificados o no se pueden encontrar en los mapas que se utilizan actualmente, lo que ocasiona confusión, desconocimiento y hace que las personas no pueden acceder a estos servicios (Torre. 2017).

En conclusión, se observa que las formas de promocionar el turismo en Juanchaco, Ladrilleros y La Barra se basan principalmente en la creación de encuestas, generación de nuevas infraestructuras, oficinas de promoción turística y demás. No obstante, no hay evidencias de la existencia de una aplicación basada en la localización, que permita acceder a la información turística de forma permanente y al alcance de la mano que dé impulso del turismo en la región.

De acuerdo con lo mencionado anteriormente, nace la siguiente pregunta: ¿cómo impulsar el turismo náutico en los corregimientos de Juanchaco, Ladrilleros y La Barra, pertenecientes a Buenaventura?

1.3. Justificación del Problema

Buenaventura es una de las ciudades de Colombia con menos popularidad turística debido al poco interés de los empresarios en el desarrollo del puerto y el bajo fortalecimiento de su turismo. Según (Torre. 2017), esta ciudad a pesar de tener una gran diversidad de atractivos propios, entre los que se destacan las playas de Juanchaco, Ladrilleros y La Barra, hay un claro desconocimiento por parte de los visitantes con respecto a las actividades turísticas; por lo que el impulso del turismo es de gran importancia para el desarrollo del municipio, (Montoya. 2016) afirman que el crecimiento de este sector es fundamental para la generación de divisas y empleos para la comunidad.

Debido a lo anterior, es necesario recurrir a técnicas que permitan impulsar el turismo de forma eficaz, y que incentiven a los turistas locales e internacionales a visitar las diferentes playas y destinos que ofrece el municipio, y de esta forma disminuir la mala imagen mostrada. Entre algunas estrategias utilizadas para la promoción turística pueden encontrarse la creación de campañas publicitarias a partir de las experiencias positivas de los visitantes (Revista P&M. 2016), inversiones en infraestructura como muelles y embarcaderos para el arribo de regatas y rallies internacionales (Reportur. 2016), entre otras.

Con los avances en la tecnología, los SIG cobran gran importancia para la promoción del turismo, ya que optimizan la información suministrada y permiten planificar actividades en lugares en donde no se están obteniendo beneficios debido al desconocimiento. Además de esto, permiten representar la información de forma dinámica, con la capacidad de elaborar mapas, videos, páginas web y aplicaciones, dando aportes significativos en la planificación y toma de decisiones en un lugar determinado, además, facilita la visión integrada del territorio, aspecto que cobra relevancia para la planificación, análisis y gestión de la actividad turísticas (Nino. 2016). La creación de la aplicación móvil atiende la necesidad de buscar un método de difusión turística de un territorio que acerque a los visitantes, mediante las técnicas apropiadas, se puede georreferenciar todas las actividades turísticas de la región (Torres. 2016).

El proyecto utiliza información de campo o capas pertenecientes a entidades gubernamentales, obteniendo información más precisa, esta aplicación ofrecerá no solo información detallada sobre las actividades náuticas de las playas, sino que también, mostrará rutas desde la ubicación del usuario hasta la actividad seleccionada, con esto se podrá ubicar en el mapa y no tendrá riesgos de perderse en una ciudad desconocida, además, el usuario podrá conocer cuánto tiempo se puede demorar en cada visita turística y así poder planificar su estadía.

Adicionalmente, en dicho trabajo se desarrolló un sistema de recomendaciones con la aplicación de técnicas de inteligencia artificial. Todo esto con la finalidad de apoyar a las empresas y fundaciones que buscan incentivar la economía del municipio, brindándoles una publicidad constante de su negocio, teniendo estas un rol importante, haciendo una alianza estratégica para la transformación turística a largo plazo fortaleciendo el sector de manera directa.

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar una herramienta para el acceso y uso a la información geoespacial de las actividades náuticas que se ofrecen en los corregimientos de Juanchaco, Ladrilleros y La Barra, pertenecientes al municipio de Buenaventura.

1.4.2. Objetivos Específicos

- Diseñar una prueba piloto de un sistema de información Geográfica (SIG) que muestre los lugares donde se realizan las diferentes actividades náuticas, teniendo en cuenta su información alfanumérica y otros sitios de interés.
- Implementar el SIG en un aplicativo móvil en Android, con la información espacial del turismo.
- Designar un módulo en que el usuario como administrador pueda actualizar la información en tiempo real y realizar respectivos mantenimientos.
- Generar un sistema de recomendaciones a partir de las preferencias del usuario, empleando técnicas de machine learning.

1.5. Resultados del Trabajo de Grado

Tabla 1. Resultados del Trabajo de Grado.

Objetivos Específicos	Producto(s) Esperados	Documentado en:
Diseñar una prueba piloto de un sistema de información Geográfica (SIG) que muestre los lugares donde se realizan las diferentes actividades náuticas, teniendo en cuenta su información alfanumérica y otros sitios de interés.	Base de datos con las actividades náuticas que ofrece el turismo en las playas de Juanchaco, Ladrilleros y La Barra. Cartografías con la información obtenida en campo, almacenada en la base de datos.	En el capítulo 3 se puede encontrar la información sobre las fuentes de información, la creación de la base de datos y los parámetros que se tuvieron para el sistema de información geográfico.
Implementar el SIG en un aplicativo móvil en Android, con la información espacial del turismo.	Prototipo funcional de la aplicación móvil y diseños de interfaz de la misma.	En el capítulo 4 se encuentra la información sobre el desarrollo y el diseño del aplicativo móvil.
Diseñar un módulo en que el usuario como administrador pueda actualizar	Un módulo en el que las empresas puedan realizar operaciones de actualización de la base de datos	En el capítulo 5 se muestra el proceso que se llevó a cabo para el módulo que tiene el administrador

la información en tiempo real y realizar respectivos mantenimientos.	de actividades, restaurantes y hoteles.	para actualizar la información de la aplicación.
Generar un sistema de recomendaciones a partir de las preferencias del usuario, empleando técnicas de machine learning.	Documentación con selección y justificación de la técnica de recomendación usada. Módulo funcional del sistema de recomendación.	En el capítulo 6 se muestra el desarrollo del sistema de recomendaciones, con su respectiva teoría y explicación de las elecciones de las técnicas de machine learning que se utilizaron.

Fuente. Elaboración propia.

1.6. Alcances de la Propuesta

Alcances:

1. El presente informe explora las diferentes actividades turísticas que ofrece algunas playas más reconocidas del municipio de Buenaventura, con el fin de impulsar el turismo.
2. La propuesta de aplicación abarca los sitios de interés, restaurantes y hospedajes de las playas Juanchaco, Ladrilleros y La Barra.
3. La aplicación permite actualizar los datos turísticos en cualquier momento, por medio del módulo de administrador.

Limitaciones:

1. Debido a que es una prueba piloto, se recogieron una cantidad de datos en campo limitada, debido a la temporada en que se realizó la muestra.
2. El informe solo contiene información de las playas de Juanchaco, Ladrilleros y La Barra.
3. Las actualizaciones solo se pueden realizar manualmente por el administrador, por lo que podría representar una demora.

CAPITULO 2. MARCO REFERENCIAL

2.1 Marco Teórico

Turismo

El turismo corresponde a las actividades que se pueden realizar en un determinado lugar, siendo estas del tipo recreativo o de ocio con el fin de que el consumidor pueda disfrutar su estadía y conocer más a fondo las costumbres y culturas que se desarrollan en una determinada región (OMT, 2019). Entre los tipos de turismo que existen, se encuentra el turismo náutico que hace referencia a las actividades realizadas en el agua como el surfing, kayak, esquí náutico, entre otras (Plataforma Tecnológica de la Macaronesia, 2015).

Sistemas de Información Geográfica

En temas turísticos cobran gran importancia los Sistemas de Información Geográfica. Estos corresponden a un conjunto de datos que contienen una ubicación geográfica definida y unas características en las que se pueden hacer diferentes análisis y a partir de los resultados de un procesamiento, tomar decisiones que permitan resolver una problemática que se esté desarrollando en una zona determinada (Geoinnova, 2018).

Entre los principales elementos que conforman un SIG se encuentran los datos, siendo estos la información utilizada para realizar los análisis que contienen su correspondiente etiqueta espacial para ser asociada a una ubicación geográfica, también, el hardware que son las máquinas empleadas para la captura o procesamiento de los datos, como lo son los equipos GPS o computadores, y el software, que hace referencia a los programas usados para procesar y desplegar los datos recolectados (FAO, 2006).

Bases de Datos

Para el manejo de información relacionada con Sistemas de Información Geográfica, es de gran importancia el uso de bases de datos, estas corresponden a un banco de almacenamiento de información que permite mantener los datos de forma organizada y realizar consultas a partir de diferentes criterios que defina el usuario, estos generalmente se visualizan en gestores de bases de datos que funcionan como interfaz para las consultas realizadas (EKCIT, 2019). A diferencia de las bases de datos estándar, se conocen las bases de datos espaciales que además de presentar las diferentes características de los datos, contienen una nueva columna que representa su ubicación espacial, su sistema de referencia correspondiente y su tipo de geometría (Valdés, 2007).

Un aspecto importante para considerar con el manejo de información son los modelos de bases de datos, ya que estos permiten representar de forma gráfica y comprensible la información que se va a mostrar y que sea funcional, eficiente y, además, facilite la representación de las diferentes características correspondientes a la realidad para hacer más comprensible la visualización de una problemática en concreto (Proteccion, 2019).

Entre los modelos de bases de datos más importantes se encuentran: el modelo entidad-relación, que consiste en un diagrama que, a partir de las entidades que se van a modelar y sus relaciones con las demás entidades, genera un esquema que facilita la representación de la base de datos que se va a diseñar; el modelo relacional, que a partir de tablas, genera un esquema más elaborado del diseño de la base de datos con sus entidades, tipos de datos correspondientes y sus relaciones entre los datos; y finalmente, el modelo físico, que consiste en la generación de un esquema de tablas con mayor detalle, teniendo en cuenta sus claves de restricción para la creación final de la base de datos (Lucid Software Inc, n.d.).

Trigger

Un Trigger o disparador, es un script que se almacena y ejecuta en una base de datos; para activarse, se debe realizar una acción que la modifique, esta puede ser una inserción, actualización o eliminación, permitiendo realizar operaciones que dependen de esos cambios realizados, o guardar información con anterioridad a la modificación, evitando que se pierdan datos al realizar una operación de forma incorrecta. Para realizar un trigger utilizando postgres, se debe realizar una función que consiste en las actividades que va a realizar, definir las tablas que van a ser afectadas y el momento en que se ejecuta (antes o después de la modificación) (PostgreSQL., s.f.).

Bases de Datos en Tiempo Real (Firebase)

Firebase es una plataforma de Google, que tiene una gran variedad de herramientas para mejorar las funcionalidades de las aplicaciones móviles. Entre estas herramientas que ofrece firebase, se encuentra Realtime Database ®, que cuenta con una base de datos no relacional en la que se pueden guardar, visualizar, actualizar y eliminar datos, por medio del formato JSON. Entre los productos más destacados está la autenticación, en donde se pueden usar diferentes métodos como Autenticación por Google, Facebook, Twitter, Número telefónico, correo electrónico, entre otros. Adicionalmente, cuenta con productos como Almacenamiento en la nube, hosting de aplicaciones, funciones de machine learning, analytics, etc. Permitiendo tener un mayor dinamismo y funcionalidad en las aplicaciones. (Google Developers, 2020).

Método de ruteo

Mapbox, desde el año 2018 usa como método de ruteo el motor de enrutamiento de código abierto Valhalla. Este a su vez utiliza información proporcionada por Open Street Maps, para de esta forma realizar diferentes procesos en el cálculo de rutas, matriz de tiempo y distancia, isócronas, muestreo de elevación, entre otros. Para realizar el enrutamiento, utiliza una API basada en el lenguaje C++ y un algoritmo de enrutamiento A*, con el que obtiene el camino más corto, teniendo en cuenta diversos factores como el viajar en vehículo, bicicleta o a pie, haciendo que las rutas proporcionadas sean más personalizadas (Eric Gundersen, 2018); (Valhalla, 2018).

Mapbox

Es un proveedor de mapas para aplicaciones web y móviles fundada en el 2010. Esta empresa provee una gran cantidad de servicios y API's de mapas, suministrando diversas funcionalidades en aplicaciones como rutas, direcciones, visualización y uso de datos geojson, búsqueda y geocodificación de ubicaciones, realidad aumentada, mapeo en tiempo real, entre otros. La SDK de Mapbox para Android permite realizar todas estas funcionalidades con facilidad, permitiendo usarlas de forma nativa para que haya un consumo de recursos menor respecto a

realizarlo con herramientas web, haciendo que sea mucho más rápido y eficiente su uso en dispositivos móviles (Climent, 2012).

Aplicaciones Móviles

Las aplicaciones móviles son un software con una gran cantidad de herramientas, que permiten la interacción con el usuario y facilita la visualización en tiempo real de la información de manera dinámica. Existen diversos tipos de aplicaciones de acuerdo con su forma de desarrollo, una es de tipo nativo que hace referencia a las aplicaciones realizadas en un lenguaje de programación específico y las de tipo híbrido, que se desarrollan para ser utilizadas en varios sistemas operativos.

Para el funcionamiento de estas aplicaciones es necesario utilizar diversos lenguajes de programación que además de permitir el correcto diseño de las aplicaciones, hacen que sean prácticas y eficientes. Los lenguajes de desarrollo de aplicaciones más conocidos son Android y iOS. El primero, consiste en un lenguaje basado en la plataforma Linux y utiliza una sintaxis de programación similar al lenguaje Java; el segundo, utiliza para su desarrollo el lenguaje web como CSS para sus estilos y el lenguaje Swift para sus funcionalidades (Softcorp, 2019).

Diagrama de actividades

Consiste en un diagrama que describe el comportamiento que debe tener sistema que se está modelando. Es una serie de pasos que debe realizar el usuario para moverse entre aplicaciones o software, describiendo las actividades que debe realizar, y que sucede al ejecutar una u otra, demostrando de forma gráfica la lógica que se encuentra detrás de la aplicación, facilitando al usuario comprender su funcionamiento (Torres, 2020).

Actividad

Una actividad hace referencia a la pantalla en que se desarrollan todos los eventos que ocurren en la aplicación; esta es un complemento entre la interfaz gráfica que consiste en un archivo xml donde se diseñan los fondos, botones, cajas de texto, etc y un archivo de lógica que se enlaza al archivo de diseño, donde se programan las funcionalidades que va a tener (o no) cada elemento añadido a la interfaz gráfica, permitiendo que sea dividido el diseño y la lógica de la aplicación (Android Developers, 2019).

Fragmento

Es una fracción de interfaz que está contenida en la actividad y con un funcionamiento muy similar, permitiendo realizar las mismas funciones, eventos y diseño de interfaz, realizando cambios muy leves en el llamado al fragmento, lo que hace que la aplicación sea eficiente y de menor peso, ya que puede ser reutilizable y todas las utilidades se están realizando sobre la misma pantalla (Developers, 2020).

HashMap

Un Hashmap es una estructura de datos de tipo matricial en el que se almacena información, donde cada elemento tiene una clave que lo identifica, permitiendo encontrar con facilidad el valor correspondiente. Esto hace posible la realización de búsquedas con gran facilidad y eficacia; además, al permitir valores nulos tiene una tendencia menor a fallos, permitiendo tener una mayor seguridad, evitando fallos, ocasionados por este tipo de valores. (Geeksforgeeks, 2020).

GPS

Las aplicaciones móviles enmarcadas en la creación de sistemas de información geográfica permiten la obtención de una ubicación espacial en tiempo real. Esto es posible gracias a la incorporación de los Sistemas de Posicionamiento Global (más conocidos por sus siglas en inglés como GPS). Estos dispositivos funcionan a partir de la trilateración de señales entre los satélites que se encuentran fuera de la órbita terrestre y el receptor que envía la señal, de esta forma, se obtiene la ubicación en la que se encuentra el usuario de forma actualizada e inmediata (Gabri, 2018).

CRUD

Hace referencia a las 4 principales operaciones básicas que se pueden realizar sobre bases de datos, que consisten en crear, leer, actualizar y eliminar información, permitiendo que pueda ser visualizada por el usuario; también debe ser posible actualizar y eliminar la información para que, al realizar cambios, estos puedan ser reflejados sobre las aplicaciones y que los archivos no sean obsoletos, y en caso de que hayan datos que dejen de existir, se puedan realizar estos cambios, manteniendo actualizada la base de datos y la aplicación (Codecademy, 2021).

HTTP

Un protocolo de transferencia de hipertexto, (más conocido por sus siglas en inglés HTTP) es un método de comunicación utilizado en aplicaciones web, para el envío de archivos html, scripts, JSON, información procedente de una base de datos, entre otros. Para la transferencia de información es necesario la existencia de un cliente y un servidor; el primero, realiza una petición por medio de un método HTTP, y el segundo procesa los datos y devuelve una respuesta que es visualizada por el cliente.

Entre los métodos más utilizados para peticiones de datos están GET y POST. El primero consiste en la asignación de valores sobre la url por medio de llaves, con el fin de obtener un resultado a partir de los parámetros ingresados; El segundo hace un envío de datos similar al método GET, con la diferencia que los datos ingresados no son visibles al usuario, añadiendo un elemento de seguridad a la petición. Estos métodos son utilizados ampliamente para la creación de API's, permitiendo el envío y procesamiento de datos para obtener un resultado, que pueda ser interpretado por el usuario (Mozilla, 2019).

Heroku

Es un servidor en el que se pueden alojar diferentes tipos de aplicaciones microservicios y API's, para lenguajes como Php, Ruby, Ruby on Rails, Python, Go, NodeJS, Java, Scala y Clojure. En esta plataforma es posible subir estas apps sin necesidad de realizar configuraciones muy complejas, siendo muy útil en su versión gratuita para pruebas y despliegue de aplicaciones sencillas que no requieran mucho procesamiento o espacio. Permite también el manejo de bases de datos relacionales como postgresql, mysql, ms sql server, entre otras y bases de datos no relacionales como mongo DB; para ser desplegadas y consumir datos de ellas con facilidad.

Para hacer uso de este heroku y desplegar aplicaciones sobre él, es necesario usar el software Git, ya que este permite desplegar archivos sobre el servidor y tener un control sobre los cambios o versiones de los elementos subidos, llevando así un correcto control de ellos y conocer las alteraciones al código y quien realizó tales modificaciones para que, al momento de presentar errores, estos puedan ser corregidos de forma inmediata (Heroku, 2021).

API

Consiste en una interfaz desde la que se pueden comunicar varias aplicaciones independientemente del lenguaje en el que fueron creadas, pudiendo conectarse a una base de datos y realizar procesos internos para devolver información, que pueda ser consumida y analizada por otra aplicación diferente, que puede o no estar realizada en un lenguaje de programación distinto. Para emplear los datos enviados por medio de estas API's es comúnmente usado el formato JSON, que permite enviar información desde aplicaciones web, con una codificación que puede ser leída por cualquier lenguaje, facilitando la comunicación entre programas (Altexsoft, 2019).

Flask

Es un micro framework con el que se pueden realizar aplicaciones web, microservicios, API's, entre otros que sean implementados en python. Con este es posible realizar solicitudes REST y obtener una respuesta de un servidor y visualizar los datos obtenidos en diferentes formatos como JSON, String, entre otros; con esto es posible subir aplicaciones que tengan características en su desarrollo que sean solo compatibles con python y que sus API's puedan ser consumidas fácilmente por aplicaciones escritas en otros lenguajes. Adicionalmente, presenta

una documentación muy extensa y una comunidad bastante grande, facilitando la corrección de errores que se puedan presentar en el desarrollo (Hunt-walker, 2018).

Inteligencia Artificial

La inteligencia artificial es un conjunto de algoritmos con la capacidad de simular las actividades realizadas por el cerebro humano. Estos funcionan a partir de la incorporación de una gran cantidad de datos que permiten entrenarlos y de esta forma, tomar decisiones en la temática para la que fueron creada. Entre los diferentes procesos que pueden ser realizados por la inteligencia artificial están el reconocimiento de patrones, los métodos de predicción, etc., que son de gran utilidad para diferentes ramas como la economía, la medicina y la agricultura (Mira, 1922).

En el turismo, la inteligencia artificial es utilizada para realizar sistemas de recomendaciones a partir de datos de las redes sociales y la ubicación del usuario, el análisis estadístico, los algoritmos matriciales, entre otros, con esto se obtiene información que permite realizar sugerencias a las personas acerca de las actividades y los destinos a visitar (Navarro Benites, 2013).

Regresión

Consiste en una serie de métodos que permiten determinar el comportamiento de una variable respecto a otras, para de esta manera predecir los valores que tomará al utilizar información nueva de la que se desconoce su resultado. Existen diversos métodos de regresión utilizados en inteligencia artificial como lo son la regresión lineal, regresión logística y random forest. Estos modelos son usados en gran medida, debido a su fácil uso y eficiencia, pudiendo ser encontrados la mayoría de los lenguajes de programación ejemplos de estos algoritmos (Dave, 2018).

Sistemas de Recomendaciones

Los sistemas de recomendación son algoritmos que permiten realizar la sugerencia óptima de acuerdo a un artículo en particular, a partir de las diferentes calificaciones que proporciona un usuario, que pueden ser implícitos como calificaciones textuales e implícitas como likes y dislikes. En los últimos años se han implementado diferentes algoritmos de inteligencia artificial, como lo son las redes neuronales artificiales, los árboles de decisión, entre otros; realizando recomendaciones a usuarios de forma automática y con una precisión bastante alta.

El filtrado colaborativo es un proceso en el que se tienen las diferentes características de una calificación o fuentes de datos, a partir de estos, se obtienen criterios que permiten realizar sugerencias a usuarios sobre un asunto en específico. Este método realiza un proceso automático según la relación entre las preferencias de muchos usuarios o colaboradores y la similitud entre las opiniones que realiza cada uno (Bagnato, 2019).

Entre los métodos más conocidos para generar sistemas de recomendación están los árboles de decisión y las redes neuronales artificiales. Los árboles de decisión son algoritmos de aprendizaje automático basados en ejemplos, que funcionan a partir de nodos vacíos. Estos se van alimentando con la información que se les proporciona y según las reglas establecidas toman una decisión entre varias opciones, formando ramificaciones. De igual forma es de gran uso las redes neuronales artificiales, estas funcionan a través de capas, donde se encuentra la capa de entrada, en la que se ingresan los datos de entrenamiento; la capa oculta donde se procesan los datos, y se entrena la red y la capa de salida, que arroja el resultado final del entrenamiento (Noren, 2019); (Rocha Muñoz, 2017).

Scikit-learn

scikit-learn es un conjunto de librerías de código abierto hecho para el lenguaje Python, que permite desarrollar, entrenar, ejecutar y guardar modelos de redes neuronales, clasificación, regresión y cluster; Está escrito en programas como C++, Python, C y Cython, permitiendo operar con diferentes librerías de Python como Numpy, SciPy, entre otras. Entre los clasificadores que soporta están el modelo lineal, logístico y random forest, con la capacidad de realizar varias salidas con la función multioutput, permitiendo obtener por cada una un valor de peso y conocer las mejores predicciones entre los resultados obtenidos (Brownlee, 2014).

Algoritmo A-star

Es un algoritmo de búsqueda que permite hallar la ruta más corta, entre un nodo de inicio y un final, utilizando métodos heurísticos (Por ejemplo, distancia euclidiana), con el fin de obtener el camino que presenta un menor peso, y evitando obstáculos sin alejarse demasiado de ellos, garantizando la distancia más óptima entre 2 puntos (Liu et al., 2019). El modelo que determina este algoritmo es el que se muestra a continuación:

$$f(N[i]) = g(N[i]) + h(N[i]) \quad (1)$$

Donde:

$g(N[i])$ es el costo de la ruta entre $N[inicio]$ y $N[i]$

$h(N[i])$ es una función heurística que estima el menor costo desde $N[i]$ hasta $N[final]$

Método Dijkstra

Este método permite hallar la ruta más corta entre dos puntos por medio de una malla vial definida. Para obtener esta ruta, el algoritmo determina los segmentos que conectan la red y el nodo de inicio y final; a partir de esta conexión y el costo que se obtiene de las restricciones respecto a las condiciones de la vía, hallando la ruta más óptima (Zhang & He, 2012).

Red Neuronal

Este modelo funciona a partir del aprendizaje utilizando datos de entrada ingresados por el usuario, que son enviados como señales sobre capas ocultas de la red y con ello se obtienen valores de salida (Moses & Dhinesh Babu, 2017). En el caso del sistema de recomendaciones, un usuario ingresa calificaciones de los sitios que son de su interés y con esta información el sistema obtiene una predicción que se actualiza con cada nueva calificación ingresada. De acuerdo con los autores mencionados anteriormente, el modelo que representa el funcionamiento de la red neuronal es:

$$S = \sum_{i=0}^n U_{Y0}(I_x)C_{Y0} \quad (2)$$

$$F(S) = f \sum_{i=0}^n U_{Y0}(I_x)C_{Y0} \quad (3)$$

Donde:

U_{Y0} = Usuarios

I_x = Calificación realizada por cada usuario

C_{Y0} = Correlación entre usuarios y la calificación

S = Valor de predicción para cada calificación

$F(S)$ = Predicción definitiva a partir de todos los elementos calificados y el valor de S más alto

Arboles de Decisión

Los árboles de decisión son algoritmos que toman decisiones a partir de ejemplos que lo alimentan. Esta información se almacena en nodos que se ramifican en cada decisión que toma el sistema. Esto permite realizar diferentes análisis y a partir de esto realizar predicciones en una temática determinada. Para alimentar el sistema de recomendación, el usuario ingresa calificaciones que son evaluadas en los nodos del árbol y se obtiene la predicción a partir del modelo planteado por (Karimi et al., 2015), como se muestra a continuación:

$$\hat{r}_{ti} = \frac{\sum_{(u,r) \in D_i^t} r + \lambda_1 \hat{r}_{si}}{|D_i^t| + \lambda_1} \quad (4)$$

Donde:

u = Conjunto de usuarios

i = ítem a calificar

r = calificación realizada
 t = Nodo en el árbol de decisión
 λ_1 = Factor de regularización (evitar ajustes excesivos)
 $\hat{r}_{si}$ = Nodo principal del árbol de decisión
 D_i^t = Perfil del ítem i en el nodo t en el entrenamiento

Este último se determina con la siguiente ecuación:

$$Dt := \{(u, i, r) \in U \times I \times R \mid u \in Ut\} \quad (5)$$

Donde:

u = usuario particular
 i = artículo a calificar
 r = calificación realizada
 U = Conjunto de usuarios
 I = conjunto de artículos
 R = total de calificaciones

Métricas de rendimiento

Entre las métricas utilizadas para evaluar modelos de aprendizaje automático se encuentran el Error Medio Absoluto, Error Medio Cuadrático y Error Medio Cuadrático Logarítmico; estas evalúan el desempeño de los modelos de inteligencia artificial a partir de la diferencia entre el resultado real y el valor predicho. El primero calcula la diferencia entre el promedio y el punto evaluado con el fin de conocer que tan alejado se encuentra el valor a la media de los datos. Por otro lado, el RMSE calcula la diferencia entre el punto evaluado y el real para conocer el error cuadrado de estos; el RMSLE tiene un procedimiento similar al RMSE, con la diferencia que se comporta mejor en cuanto a los valores atípicos (Barua Gono Bishwabidyalay et al., 2019), (Zeroual et al., 2020). La ecuación de cada métrica se presenta a continuación:

$$MAE = \sum_{t=1}^N |y_t - \hat{y}_t| \quad (6)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (7)$$

$$RMSLE = \sqrt{\frac{1}{n} \sum_{t=1}^n (\log(y_t) - \log(\hat{y}_t))^2} \quad (8)$$

Donde:

y_t son los valores reales

$\hat{y}_t$ valores predichos

2.2 Antecedentes o Estado del Arte

El proyecto pretendió buscar una metodología que permita crear un espacio donde el usuario pueda encontrar información relevante sobre el turismo náutico de la ciudad de buenaventura, por lo que se debe comenzar por identificar las variables necesarias, como lo expresan Altimira Vega & Muñoz Vivas (2007) en un estudio sobre el turismo como motor del aumento de ingresos en determinado lugar. Consideran que la forma o los parámetros a tener en cuenta para crear una herramienta que se base en el turismo o sirva para promoverlo, es conocer la demanda que hay en la zona y las necesidades que puede suplir la región con las actividades turísticas para saber el comportamiento del turismo. Además, es indispensable conocer las tendencias mundiales, el nivel de preferencias por parte de los visitantes en cuanto a las diferentes actividades para que la herramienta o método sirva adecuadamente. Por lo tanto, proponen buscar estrategias como la competitividad entre empresas para aumentar la oferta. Con lo anterior, se espera que las actuaciones de estas sean conscientes con el desarrollo sustentable e incentiven a los visitantes y habitantes a apropiarse y preservar de la cultura de esta región.

Posteriormente, el proyecto desarrolló un sistema de información geográfica para el manejo y procesamiento de los datos, tanto para variables espaciales, como variables no espaciales. Tomando referencia de Torres, (2020), que realizó una investigación sobre la aplicación de un sistema de información geográfica para la explotación de recursos turísticos, por medio de softwares libres y datos turísticos no reales para demostrar su utilidad como herramienta turística. Para el desarrollo del SIG, ella estableció Travis (Tejas, EEUU) como su zona de estudio, organizando los datos de los recursos según el turismo. Luego, diseñó un modelo relacional para establecer las relaciones entre los elementos turísticos denominados “entidades”. A cada entidad se le definió atributos que son los datos de los recursos turísticos como lo son nombre, horarios, precios, etc., con esto se pueden establecer las tablas organizadas para la base de datos. En la creación de esta y de la cartografía, finalmente, utilizó el software libre KOSMO, añadiendo los datos espaciales y aplicándoles una simbología adecuada para que los usuarios puedan interpretar la información más fácil.

Con el SIG creado, se llevó a cabo la posterior implementación para una plataforma móvil, de tal forma que se pueda realizar los respectivos estudios, por lo que se analizó la metodología empleada por Cruz et al., (2015), quienes realizaron un SIG como base de la modelización y análisis territorial para las principales áreas turísticas del Litoral de Canarias. Propusieron 5 fases para la metodología. Para la primera fase, obtuvieron el conjunto de variables temáticas que permitan realizar una medición. Para la fase 2, agruparon las anteriores variables en dos bloques en función de su escala. La fase 3 consistió en el diseño e implementación del SIG, por lo que los autores construyeron las variables a partir de varias fuentes disponibles. Una vez implementado el SIG, se procedió a la aplicación de los procesos previstos para la obtención, mediante mediciones sistemáticas. Para esto, utilizaron herramientas de geoprocésamiento y análisis espacial a las variables seleccionadas de forma independiente. Este modelo aplicado de los SIG, no solo ha enriquecido la base teórica del turismo, además, su diseño e implementación han aportado información valiosa a las bases prácticas para este estudio.

Para la realización del SIG se necesitó un software en el que se pueda almacenar, procesar y analizar los datos, por lo tanto, los autores Leonelli & Fernández, (2007) realizaron el diseño e implementación de un aplicativo que permita la localización y visualización de mapas, teniendo como plataforma los teléfonos móviles. Para esto, los autores utilizaron la plataforma Java 2 Micro Edition (J2ME) y los sistemas de posicionamiento. Crearon la base de datos utilizando el software libre PostgreSQL 8.0 y con su extensión de Postgis. Una vez creada la base de datos, utilizaron un servidor web Apache 2.0.54 con PHP 4.3.1, para el almacenamiento de datos y que el aplicativo no necesite ocupar cachés innecesarios en el dispositivo móvil. Además, para el acceso a la base de datos y las consultas espaciales utilizaron el software org2org. Por último, para la descarga de datos se empleó un servicio de bluetooth.

Para la creación del módulo o la interfaz del aplicativo, utilizaron MIDlet. Mediante este módulo los usuarios pueden descargar información, obtener su ubicación o la de un sitio de su interés, etc. Para implementar MIDlet se empleó la Suite J2ME Wireless Toolkit v2.2 ya que cuenta con más características que hacen más fácil el desarrollo de las aplicaciones. Además, los autores pusieron un módulo donde el usuario pudiera encontrar la ruta óptima desde su posición actual hasta su lugar de destino. Para esto, emplearon el algoritmo Dijkstra. Por último, realizaron pruebas del sistema y como resultado obtuvieron grandes avances y mejoraron la facilidad de ofrecer información espacial.

Se hace necesario que los datos que tuvieron su debido proceso se subieran a un servidor, con el fin de que se puedan utilizar y visualizar en diferentes plataformas que no sean locales, por lo que los autores Amores Bassante & Suárez Loaiza, (2016) diseñaron un aplicativo web que le muestra a los usuarios el grado de inseguridad en el distrito de Quito, por medio de un mapa de calor. Para lo anterior, los autores se basan en las publicaciones que los usuarios realizan en la red social Twitter. Estos datos que se recogen, los almacenan en una base de datos llamada "Django" que es eficaz en el momento de hacer consultas. Una vez creada la base de datos, los autores utilizan un servidor web que se llama "Heroku", que permite el uso de varios lenguajes de programación y es una plataforma que ofrece acceso a aplicaciones desplegadas un url y dominios para que se puedan subir ya sea a un dispositivo móvil o a un sitio web para que los datos estén en la nube, lo que permite tener la información en cualquier momento y realizar cambios a estos. El aplicativo web obtuvo buenos resultados siguiendo una metodología Scrum, teniendo parámetros para su precisión en el escrito correcto de las direcciones y la funcionalidad de este aplicativo depende directamente de que los usuarios en los twittees cumplan con formatos específicos.

Una vez creado el SIG, se procedió a realizar la implementación del mismo a la plataforma en la que se mostraran e interactuarán con los datos, para esto, los autores Caicedo et al., (2016) crearon un aplicativo móvil como una estrategia de marketing para impulsar la productividad del Cantón Jipijapa (Ecuador), por medio de una plataforma APP Inventor 2 y Android Studio, siendo este último utilizado para optimizar el lenguaje de programación y corregir los errores de la adaptación a los dispositivos y por último, para el análisis estadístico utilizaron el paquete de Microsoft Office, lo que ayudó a estudiar los resultados y demostrar la efectividad de la aplicación, por lo tanto, muestra la necesidad de implementar este tipo de aplicaciones en la ciudad para impulsar el turismo.

La creación del aplicativo móvil se basa en la visualización de los datos espaciales que se almacenaron y procesaron en la base de datos, por lo que se buscó utilizar un mapa base, tal y como utilizaron Camacho & Cornavaca, (2017) que crearon un aplicativo móvil que consistía en un reconocimiento de cada especie que se encuentra en el jardín botánico Ambiental UNAN-León en Nicaragua, con el fin de guiar a los turistas sin la

necesidad de un guía. Además, el aplicativo les brinda un mapeo de la zona, utilizando sistemas de información geográfica en la web, lo que les permite ingresar cartografía a la nube, en el que incluyeron datos gratuitos que ofrece OpenStreetMap. Para el mapa base utilizaron la librería de mapbox, que escogieron por ser de código abierto, soporta todas las plataformas que utilizan ahora los smartphones y se puede configurar con lenguaje que ya es muy conocido como lo es JavaScript. Obtuvieron un mapa completo con todos los marcadores que se necesitan para que el turista pueda reconocer todo el jardín botánico.

Para la creación de estos sistemas inteligentes se utilizan diferentes lenguajes de programación, como por ejemplo el proyecto realizado por Hurtado Jerves & Zabala Bustamante, (2018) que desarrollaron un sistema inteligente web para la recomendación de destinos turísticos en el Cantón Morona. Utilizando la metodología Microsoft Solution Framework. Esta metodología incluía la utilización del lenguaje de programación Python, conocido por su velocidad en ejecución y facilidad en la creación de software, siendo un lenguaje multiplataforma. Además, los autores emplearon el paradigma orientado a objetos y la división en módulos de la aplicación desarrollada. Por último, realizaron pruebas del sistema, obteniendo una influencia positiva en la exactitud del sistema de recomendación de los destinos turísticos en un 63\%.

En el caso de las aplicaciones enfocadas al turismo, se observa en gran medida la creación de sistemas inteligentes con el fin de darles un plus y atraer con mayor facilidad a los usuarios. Uno de estos sistemas es la realidad aumentada implementada a los sistemas de recomendación; los autores Luis Leiva Olivencia et al. (2015) buscaron mejorar las aplicaciones que están orientadas a servicios turísticos, adicionando un módulo de recomendaciones. Los autores crearon el sistema RAMCAT (Realidad Aumentada Móvil Contextual Aplicada al Turismo) que mejora la forma en que las aplicaciones fomentan el turismo, por medio de varios módulos: el filtrado colaborativo, contextual, recomendación por contenido, etc., basándose en una arquitectura modular compuesto por: web RAMCAT, el sistema de almacenamiento y un gestor de contenidos de RA (Realidad Aumentada), el sistema de procesamiento avanzado, un estor de perfil de usuarios, un sistema de trazabilidad, un sistema de recomendaciones, un gestor de rutas, entre otros.

Una vez creada la aplicación con sus diferentes módulos y sus diseños, se procedió a hacer una herramienta adicional que pretenda darle recomendaciones al usuario para que le quede mucho más fácil encontrar una actividad que se adecue a sus preferencias, por lo que se investigaron diferentes autores que realizaron sistemas recomendadores aplicados a diferentes temáticas. Unos de ellos fueron Gao et al. (2018) que proponen un nuevo sistema de recomendaciones de POI llamado “GeoEISo”. Este modelo adopta un método de estimación de la densidad de la función del núcleo adaptable en diferentes niveles de banda, para modelar la influencia geográfica mediante diversos anchos de banda de kernel al adaptar algoritmos de estimación de la densidad de kernel. Este modelo lo basaron en el modelo SVD++, que establece una relación entre el usuario y las calificaciones para generar predicciones.

Debido a que el motor de recomendaciones realizado en este trabajo es diferente al evaluado anteriormente se estudió el de Karimi et al. (2015) que realizaron un sistema de recomendaciones que se actualiza con las calificaciones de los nuevos usuarios sin afectar la precisión. Para esto utilizaron una combinación de árboles de decisión de 6 vías con factorización matricial, para registrar calificaciones de 1 a 5 y desconocidas. Este método fue comparado con otros como promedio de 6 vías y de 3 vías, valores aleatorios, más popular local y más popular global, donde descubrieron que a diferencia de los demás, la técnica propuesta mejora la precisión con el paso del tiempo y el aumento de usuarios.

Otro trabajo realizado en el que se implementa un sistema de recomendaciones basado en calificaciones fue el realizado por los autores Wang et al. (2017) que crearon una propuesta para el marco de recomendaciones a partir de calificaciones, ubicaciones y etiquetas, generando a partir de estas recomendaciones personalizadas. Para esto,

utilizaron dos métodos, combinándolos para obtener mejores resultados. Una de estas, fue la asignación de Dirichlet Latente (LDA) que tiene un método de agrupamiento de usuarios, a los que se les recomienda una lista con elementos más visitados por otros usuarios. El otro método es híbrido, puesto que junta la búsqueda global basada en la población y la búsqueda local diseñada para acelerar la búsqueda y encontrar mejores resultados.

Debido a que los sistemas mencionados con anterioridad se basaban en calificaciones de otros usuarios, se investigaron distintos enfoques como el de Betarte et al. (2006) que crearon un sistema de recomendación de música, en el que se basaron en el filtrado colaborativo, donde la recomendación se basa en algoritmos de memoria, realizando un cálculo de similitud entre dos usuarios, por medio del coeficiente de correlación de Pearson, utilizando una regresión lineal con el fin de disminuir el error en la recomendación. Los resultados que obtuvieron los autores fueron satisfactorios, sin embargo, tuvieron limitantes como lo es la cantidad máxima de vecinos que admite el cálculo de predicción y la dificultad que tuvieron en la recolección de datos, pues no pudieron realizar las pruebas de los puntajes a partir de información implícita, que afecta directamente en la calidad de las recomendaciones.

Por último; los autores Moses & Dhinesh Babu (2017) realizaron un sistema de recomendación inteligente utilizando redes neuronales de dos tipos: red neuronal feed forward con un algoritmo de propagación hacia atrás, y red neuronal recurrente. La primera permite al usuario corregir las calificaciones realizadas, y a la red aprender de estos errores para mejorar la calificación; la segunda utiliza capas ocultas que funcionan internamente de forma cíclica, para procesar los datos ingresados y obtener un resultado final en la capa de salida. Estos métodos fueron realizados con el fin de reducir el gasto de memoria que tienen los sistemas de recomendaciones tradicionales, debido a la gran cantidad de información que se obtiene en internet y los servicios web.

Entre los ejemplos estudiados de aplicaciones de turismo creadas en Colombia, se encuentran las siguientes:

Cali Guía turística con mapa CO

Cali Guía turística con mapa CO, es una aplicación móvil creada por los desarrolladores minube. Le brinda al usuario una guía turística a partir de las recomendaciones y fotos que suben a la plataforma los viajeros de minube, además, permite guardar listas de sitios de tal forma que estando sin internet permite visualizarla. A partir de esta aplicación se puede obtener una idea de la acogida que puede llegar a tener esta aplicación móvil, puesto que se observa que ahora muchas personas sea que hayan tenido experiencias como las que son su primera vez visitando estos lugares, les parece muy bueno y sencillo consignarlo en una plataforma como esta y les da seguridad en su viaje. (Minube, 2020)

Oís- Secretos turísticos

Esta es una aplicación móvil, que permite el descubrimiento de sitios turísticos en las zonas urbanas y rurales, creada por los desarrolladores Gfourmis. Esta aplicación se enfocó en la ciudad de Santiago de Cali, lo que muestra que la aplicación que se logró obtener en este trabajo de grado cubre una zona que no tiene este tipo de aplicaciones, donde muestre esas actividades turísticas y así impulse el turismo en el municipio de Buenaventura. (GFourmis, 2020)

ViajaxColombia

Es una aplicación creada por Socolombia.com, que permite a los usuarios planear los viajes que piensen realizar en sus vacaciones, mostrando todo tipo de actividad turística. Además, ofrece información sobre sitios vacacionales, parques, hoteles, entre otros, mostrando también el valor de los peajes de toda Colombia. La aplicación muestra fechas sobre ferias y fiestas y demás. A pesar de ser una aplicación tan completa y a nivel

nacional, Navi_App tiene un factor a favor y es el sistema de recomendaciones que le recomienda actividades según las preferencias del usuario. (Socolombia, 2019)

CAPITULO 3. DISEÑAR UNA PRUEBA PILOTO DE UN SISTEMA DE INFORMACIÓN GEOGRÁFICA (SIG) EN LAS ZONAS DONDE SE REALIZAN LAS DIFERENTES ACTIVIDADES NÁUTICAS, TENIENDO EN CUENTA SUS CARACTERÍSTICAS Y LA CERCANÍA A OTROS SITIOS DE INTERÉS.

El proyecto se desarrolló en tres etapas, la primera consistió en una fase exploratoria, en donde se recogió todo lo necesario en campo, para crear la base de datos con información actualizada, verificada y, además, se tomaron las coordenadas más exactas de estas actividades náuticas, de restaurantes de la zona y del hospedaje, teniendo lo necesario para la siguiente etapa. La segunda etapa, es una fase de desarrollo e implementación del sistema de información geográfica.

3.1 Definición de la zona de Estudio:

Este proyecto se realizó en el municipio de Buenaventura, en las playas de Juanchaco, Ladrilleros y La Barra, ubicados en la zona Noroeste, con coordenadas (3.9186813 N, -77.3684048 O), (3.6500613 N, -76.9420648 O) y (3.9666173 N, -77.3823166 O) respectivamente.

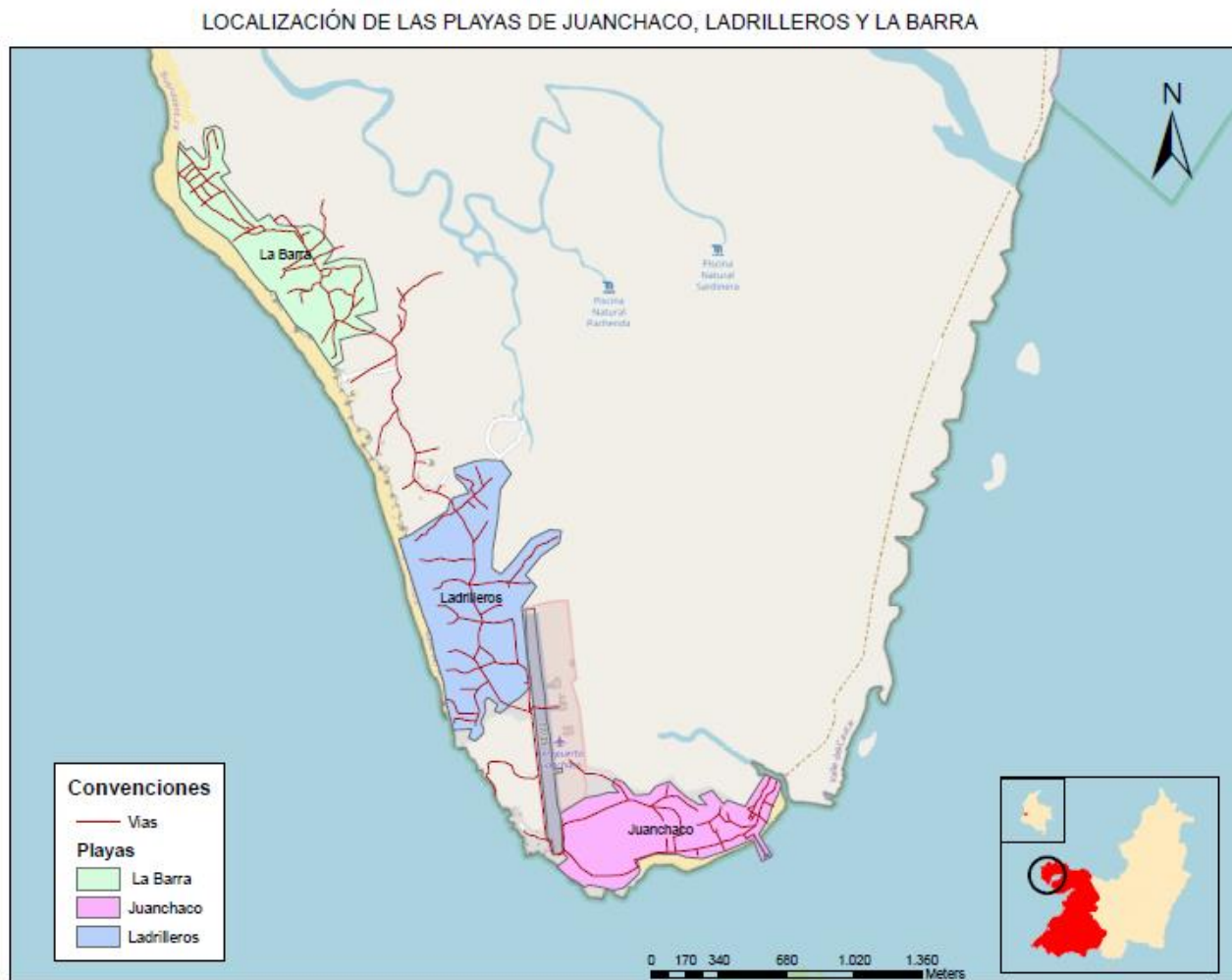


Figura 1. Ubicación de las playas de Juanchaco, Ladrilleros y La Barra.

Fuente. Elaboración propia. (2020)

3.2 Levantamiento de información de turistas.

En la etapa exploratoria, durante la salida de campo se realizó una encuesta que se puede consultar más detallada en la sección de anexos (Anexo A). Esta encuesta se les hizo a algunos visitantes en el puerto de Buenaventura, donde salen las lanchas y barcos hacia las diferentes playas que conforman el municipio, las encuestas fueron sobre su experiencia en cuanto al desplazamiento y ubicaciones de los sitios turísticos que se ofrecen. Esto con el fin de conocer el perfil de un turista que visita esta región, de esta manera se identificó la necesidad de desarrollar la aplicación móvil que los ayude a localizar las actividades con las características que busca cada cliente, facilitando su visita por las playas y ofreciendo más seguridad.

Debido a las condiciones en se encuentra el mundo actualmente solo se alcanzaron a realizar de manera presencial 20 encuestas a las personas que se encontraban en el puerto de Buenaventura y de manera virtual se realizaron 152 encuestas a través de las redes sociales. En la encuesta se encontró que el 52.6% de los turistas no tiene un conocimiento con respecto a las actividades que se ofrecen en las tres playas y buscan otros medios de información poco seguros y eficaces, tal y como se muestra en la figura 2.

Tenia usted conocimiento de las actividades que ofrece esta playa?

152 respuestas

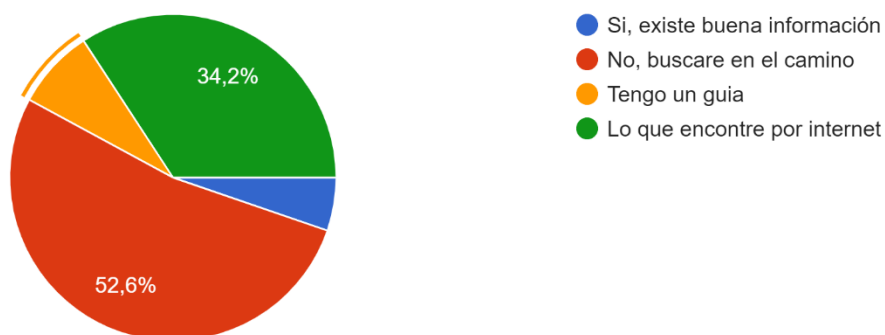


Figura 2. Conocimiento de las actividades que ofrece la playa.

Fuente. Elaboración propia.

Entre los resultados de la encuesta se tiene que el 51% de los turistas utiliza información que obtiene de los habitantes de las playas para poder orientarse y dirigirse al destino que busca. (Figura 3).

Qué herramienta utiliza para conocer la ruta a su destino deseado?

151 respuestas

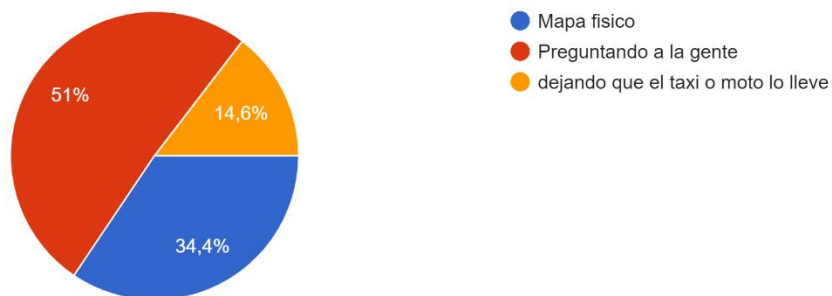


Figura 3. Herramienta que más utiliza para conocer la ruta a su destino deseado.

Fuente. Elaboración propia.

El 94.7% de las personas considera que tener una herramienta que les permita conocer sobre las diferentes actividades que ofrece el sitio turístico le sería más fácil viajar y conocer. (Figura 4).

Si tuviera la oportunidad de conocer sobre las diferentes actividades que ofrece las playas ¿consideraría que sería mucho más fácil viajar y conocer?
151 respuestas

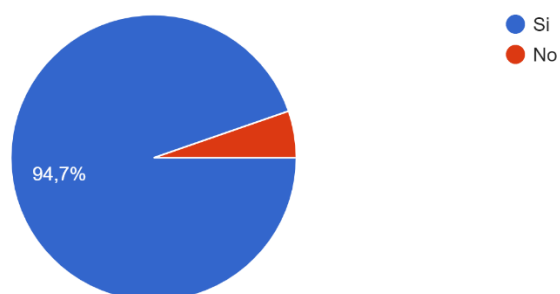


Figura 4. Considera que el conocimiento de las actividades que ofrecen las playas le sería más fácil viajar y conocer.

Fuente. Elaboración propia.

3.3 Análisis de requerimientos.

Con base con la información que se obtuvo de las encuestas, se desarrolló un análisis de requerimientos según las necesidades. Con este análisis se define las entidades y atributos que se necesitan para brindarle una completa información al turista. Por medio de la siguiente tabla se analizan los objetivos que se plantearon al comienzo del proyecto.

Tabla 2. Análisis de los objetivos planteados en este proyecto.

OBJ-01	Diseñar una prueba piloto de un Sistema de Información Geográfica (SIG) en las zonas donde se realizan las diferentes actividades náuticas, teniendo en cuenta sus características y la cercanía a otros sitios de interés.
Descripción	El sistema deberá mostrar todas las actividades náuticas que se ofrecen en las playas de Juanchaco, Ladrilleros y La Barra, además, de sus características y sitios de interés cercanos.
Estabilidad	alta
Comentarios	ninguno
OBJ-02	Implementar el SIG en un aplicativo móvil en Android, con la información espacial del turismo.

Descripción	El sistema deberá mostrar en el aplicativo la información espacial correspondiente a las actividades náuticas como lo es la ubicación y la ruta optima.
Estabilidad	alta
Comentarios	ninguno
OBJ-03	Designar un módulo en que el usuario como administrador pueda actualizar la información en tiempo real y realizar respectivos mantenimientos.
Descripción	El sistema deberá tener un espacio donde el usuario como administrador pueda ingresar nuevas actividades, editar las actuales y eliminar datos.
Estabilidad	alta
Comentarios	ninguno
OBJ-04	Generar un sistema de recomendaciones a partir de las preferencias del usuario, empleando técnicas de machine learning.
Descripción	El sistema deberá ofrecer recomendaciones de actividades turísticas náuticas que le ayuden al usuario a escoger la que mejor se adapte a sus preferencias.
Estabilidad	alta
Comentarios	ninguno

Fuente. Elaboración propia.

3.4 Análisis de almacenamiento.

Con lo anterior se define que tipo de información se necesitará almacenada para cumplir con los objetivos, por lo tanto, se realiza una lista con los requisitos de información detallada. En la tabla 3 se observa los RF, que hace referencia a los requerimientos funcionales que posteriormente se explicarán con más detalle.

Tabla 3. Requerimientos de Información almacenada.

RI-01	Información sobre las actividades náuticas
Objetivos asociados	OBJ-01 Diseñar una prueba piloto de un Sistema de Información Geográfica (SIG) en las zonas donde se realizan las diferentes actividades náuticas, teniendo en cuenta sus características y la cercanía a otros sitios de interés.
Requisitos asociados	RF-01 Información espacial y alfanumérica de la actividad RF-03 Consulta de la ruta óptima RF-05 Actualización de datos de las actividades náuticas RF-08 Recomendación de actividades náuticas
Descripción	El sistema deberá almacenar la información correspondiente a las actividades náuticas. En concreto:
Datos específicos	1. Nombre de la actividad náuticas 2. Horario 3. Precio 4. Reseña 5. Categoría 6. Subcategoría

	7. Geometría 8. Latitud 9. Longitud
Intervalo temporal	pasado y presente
Estabilidad	alta
Comentarios	ninguno
RI-02	Información sobre los usuarios
Objetivos asociados	OBJ-03 Designar un módulo en que el usuario como administrador pueda actualizar la información en tiempo real y realizar respectivos mantenimientos.
Requisitos asociados	RF-02 Identificación de usuario RF-04 Actualización de datos del usuario
Descripción	El sistema deberá almacenar la información correspondiente a los usuarios. En concreto:
Datos específicos	1. Nombre 2. Correo 3. Contraseña 4. Tipo de usuario
Intervalo temporal	sólo presente
Estabilidad	alta
Comentarios	Solo el usuario de tipo administrador podrá realizar este tipo de acciones.
RI-03	Información sobre restaurantes
Objetivos asociados	OBJ-01 Diseñar una prueba piloto de un Sistema de Información Geográfica (SIG) en las zonas donde se realizan las diferentes actividades náuticas, teniendo en cuenta sus características y la cercanía a otros sitios de interés.
Requisitos asociados	RF-06 Información espacial y alfanumérica de los restaurantes ubicados en las playas.
Descripción	El sistema deberá almacenar la información correspondiente a los restaurantes. En concreto:
Datos específicos	1. Nombre 2. Geometría 3. Latitud 4. Longitud
Intervalo temporal	sólo presente
Estabilidad	alta
Comentarios	Ninguno.
RI-04	Información sobre hospedajes
Objetivos asociados	OBJ-01 Diseñar una prueba piloto de un Sistema de Información Geográfica (SIG) en las zonas donde se realizan las diferentes actividades náuticas, teniendo en cuenta sus características y la cercanía a otros sitios de interés.
Requisitos asociados	RF-07 Información espacial y alfanumérica de los hospedajes ubicados en las playas.
Descripción	El sistema deberá almacenar la información correspondiente a los hospedajes. En concreto:
Datos específicos	5. Nombre 6. Horario 7. Precios 8. Reseña 9. Geometría

	10. Latitud 11. Longitud
Intervalo temporal	sólo presente
Estabilidad	alta
Comentarios	Ninguno.

Fuente. Elaboración propia.

3.5 Requerimientos Funcionales

Una vez creada la lista de los requerimientos de almacenamiento de información se procede a determinar los requerimientos del sistema que son funcionales, es decir, que permite que el sistema realice las diferentes funciones para satisfacer las necesidades del usuario. A continuación, se presentan la tabla de caso de uso, que permiten identificar cada requerimiento funcional con respecto a su objetivo.

Tabla 4. Requerimientos Funcionales.

RF- 01	Información espacial y alfanumérica de la actividad
Objetivos asociados	OBJ-01 Diseñar una prueba piloto de un Sistema de Información Geográfica (SIG) en las zonas donde se realizan las diferentes actividades náuticas, teniendo en cuenta sus características y la cercanía a otros sitios de interés.
Requisitos asociados	RI-01 Información sobre las actividades náuticas
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando alguien solicite información de una actividad seleccionada.
RF- 02	Identificación de usuario
Objetivos asociados	OBJ-03 Designar un módulo en que el usuario como administrador pueda actualizar la información en tiempo real y realizar respectivos mantenimientos.
Requisitos asociados	RI-02 Información sobre los usuarios
Descripción	El sistema deberá almacenar toda la información solicitada del usuario y/o administrador.
RF- 03	Consulta de la ruta óptima
Objetivos asociados	OBJ-01 Diseñar una prueba piloto de un Sistema de Información Geográfica (SIG) en las zonas donde se realizan las diferentes actividades náuticas, teniendo en cuenta sus características y la cercanía a otros sitios de interés.
Requisitos asociados	RI-0 Información sobre las actividades náuticas
Descripción	El sistema deberá mostrar la ruta desde la ubicación del usuario hasta la actividad seleccionada.
RF- 04	Actualización de datos del usuario
Objetivos asociados	OBJ-03 Designar un módulo en que el usuario como administrador pueda actualizar la información en tiempo real y realizar respectivos mantenimientos.
Requisitos asociados	RI-02 Información sobre los usuarios
Descripción	El sistema tendrá una opción donde el administrador podrá actualizar los datos registrados de los usuarios.

RF- 05	Actualización de datos de las actividades náuticas
Objetivos asociados	OBJ-03 Designar un módulo en que el usuario como administrador pueda actualizar la información en tiempo real y realizar respectivos mantenimientos.
Requisitos asociados	RI-01 Información sobre las actividades náuticas.
Descripción	El administrador podrá crear, editar o eliminar datos de las actividades náuticas para el mantenimiento constante de la aplicación.
RF- 06	Información espacial y alfanumérica de los restaurantes ubicados en las playas.
Objetivos asociados	OBJ-01 Diseñar una prueba piloto de un Sistema de Información Geográfica (SIG) en las zonas donde se realizan las diferentes actividades náuticas, teniendo en cuenta sus características y la cercanía a otros sitios de interés.
Requisitos asociados	RI-03 Información sobre los restaurantes
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando alguien solicite información de un restaurante seleccionada.
RF- 08	Información espacial y alfanumérica de los hospedajes ubicados en las playas.
Objetivos asociados	OBJ-01 Diseñar una prueba piloto de un Sistema de Información Geográfica (SIG) en las zonas donde se realizan las diferentes actividades náuticas, teniendo en cuenta sus características y la cercanía a otros sitios de interés.
Requisitos asociados	RI-04 Información sobre hospedajes
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando alguien solicite información de un restaurante seleccionada.

Fuente. Elaboración propia.

Según el anterior análisis de requerimientos, el sistema tiene dos tipos de usuarios, que son el usuario turista, es decir, el usuario que tiene la disponibilidad de visualización de las ubicaciones de las actividades náuticas, los hospedajes y restaurantes que ofrecen las playas de Juanchaco, Ladrilleros y La Barra, además, de obtener información más detallada, también, recomendaciones de actividades a través de las preferencias del usuario. El otro usuario es el tipo administrador, que puede realizar todas las funciones que tiene el usuario turista, agregando funciones para agregar, actualizar y eliminar datos de la base de datos. En la siguiente tabla muestra los dos actores del sistema.

Tabla 5. Los actores que podrán interactuar en el sistema.

ACT-01	Usuario
Descripción	Este actor representa al usuario de la aplicación.
Comentarios	ninguno
ACT-02	Administrador
Descripción	Este actor representa al administrador, realizará las modificaciones al SIG sobre las actividades y usuarios.
Comentarios	Este actor debe generar toda la información sobre las actividades náuticas.

Fuente. Elaboración propia.

3.6 Diagramas de casos de uso

Posteriormente, se realizaron los diagramas de casos de uso que muestra más detalladamente las funciones de cada actor del sistema.

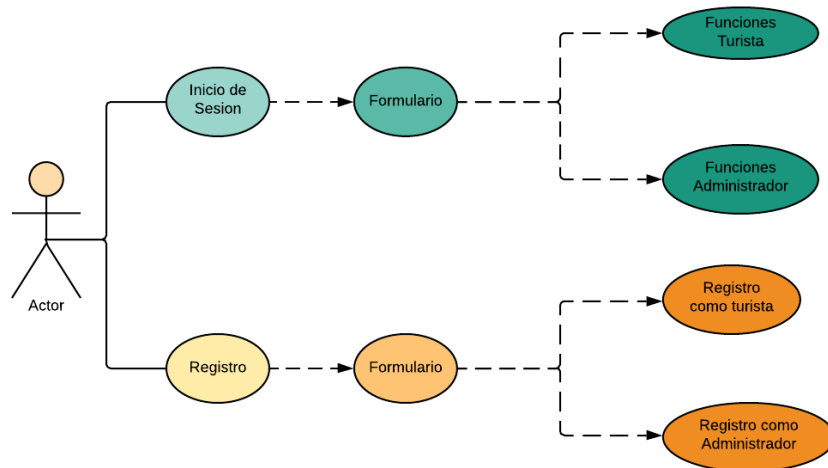


Figura 5. Diagrama de caso de uso para el inicio en el sistema.

Fuente. Elaboración propia.

En la figura 5, se ve reflejado las primeras funciones que tiene cualquier actor, cuando inicia la aplicación. Tiene la opción de iniciar sesión si ya está registrado o registrarse como usuario nuevo por medio de un formulario y seleccionando la opción de tener cuenta como turista o como administrador (Empresa).

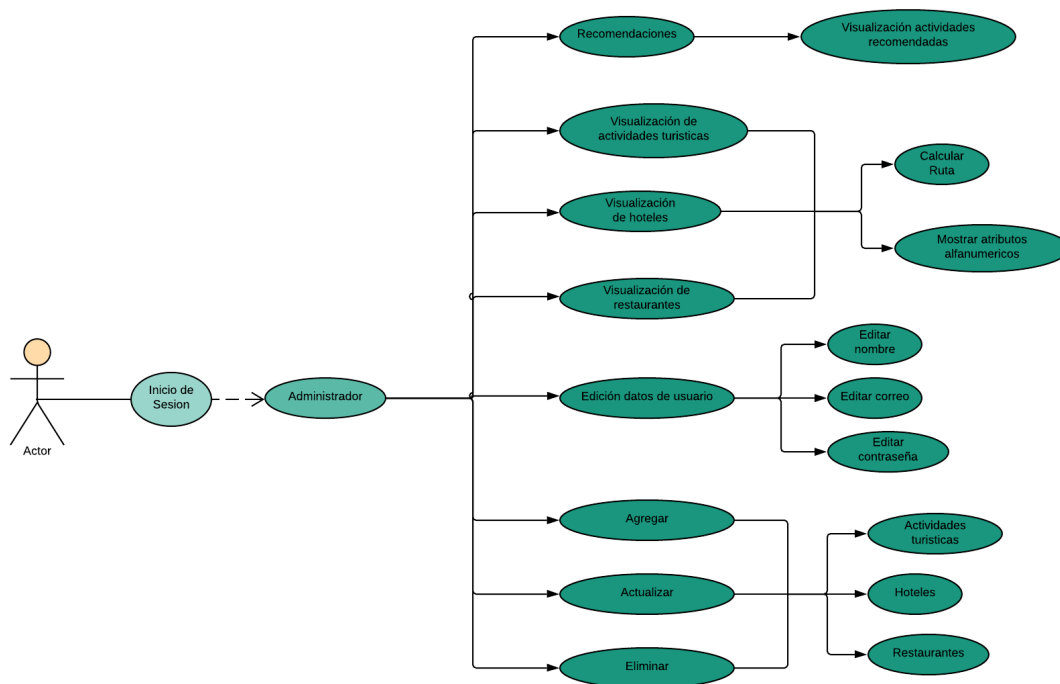


Figura 6. Diagrama de caso de uso para las funciones que tiene el actor Administrador en el sistema.

Fuente. Elaboración propia.

En la figura 6, se muestra las diferentes acciones que tiene el actor administrador (Empresa), este podrá acceder a la visualización de las actividades, los restaurantes y hoteles, con su respectiva información, y generar una ruta que proporciona el sistema para que el usuario pueda llegar a su destino. El usuario puede obtener recomendaciones de actividades que más se adecuen a sus preferencias, visualizando estas recomendaciones en el mapa. Además, solo el usuario administrador tiene acceso a un módulo en el que puede agregar, actualizar y eliminar datos de las actividades, hoteles y restaurantes, con el fin de mantener actualizado la base de datos. También, puede actualizar los datos de usuario.

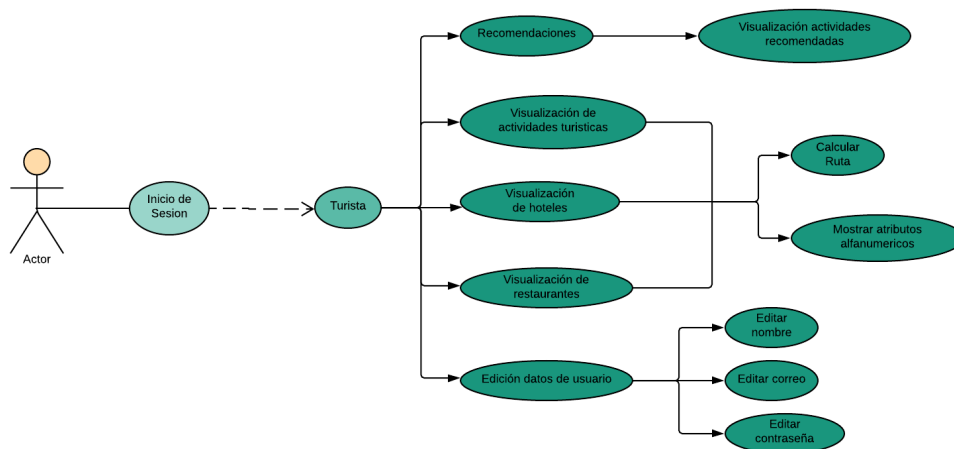


Figura 7. Diagrama de caso de uso para las funciones del actor Usuario en el sistema.

Fuente. Elaboración propia.

En la figura 7, se observa las funciones que tiene acceso el actor turista, que son visualización de las actividades, hoteles y restaurantes con su respectiva información, generar una ruta que proporciona el sistema para que el usuario pueda llegar a su destino elegido. Además, podrá obtener recomendaciones de las actividades que más se adecuen a sus preferencias, visualizando estas recomendaciones en el mapa. También, puede actualizar los datos de usuario.

A partir del análisis de requerimientos, se procedió al desarrollo del sistema de información geográfica, que se estructura en base a los requerimientos, puesto que, se establecieron los datos requeridos para el funcionamiento de la aplicación.

3.7 Desarrollo del sistema de información geográfica

En el desarrollo del sistema de información geográfica (SIG) se establece el diseño general del proyecto, obteniendo los requerimientos y alcances necesarios para la creación de la aplicación, con esto se garantiza el cumplimiento de los objetivos planteados. El SIG consiste en una base de datos espaciales creada a partir de modelos de Entidad-Relación, relacional y lógico. En la figura 8 se visualiza el modelo entidad-relación, que establece los enlaces que existen entre las entidades, que son datos principales en el diseño de la aplicación, como lo son las actividades turísticas, hoteles, restaurantes, entre otros. Además, se observan los atributos que contiene cada entidad.

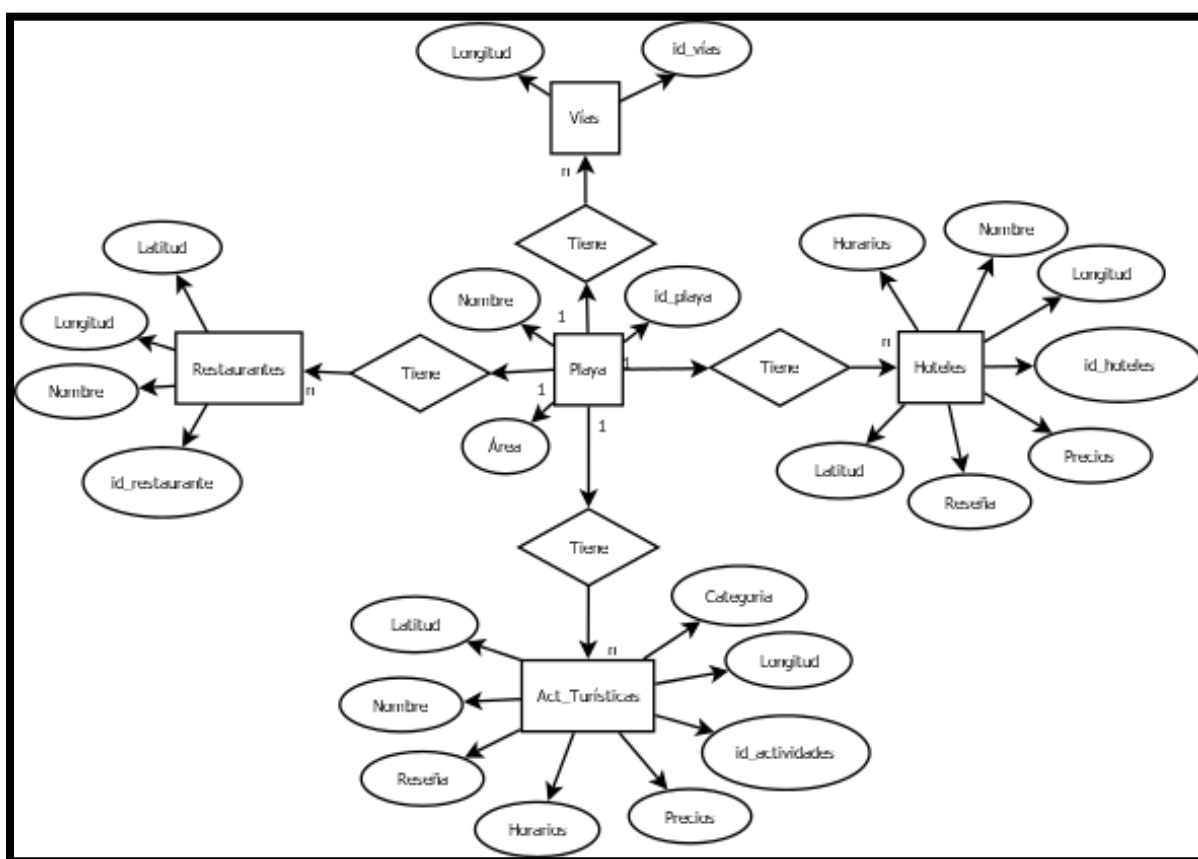


Figura 8. Modelo Entidad-Relación para el desarrollo del SIG.

Fuente. Elaboración propia.

Con el modelo anterior, se genera el modelo relacional, que ayuda a identificar las llaves foráneas, es decir, el atributo con el que la entidad o tabla se distingue, por lo que se pueden establecer el tipo de relación que existen entre ellas. En la siguiente figura XX se observa las llaves foráneas y como se relacionan con estas las entidades, este diagrama se realizó por medio del software DBDesigner, que permite crear este tipo de figuras y posteriormente genera un código que sirve para construir la base de datos.

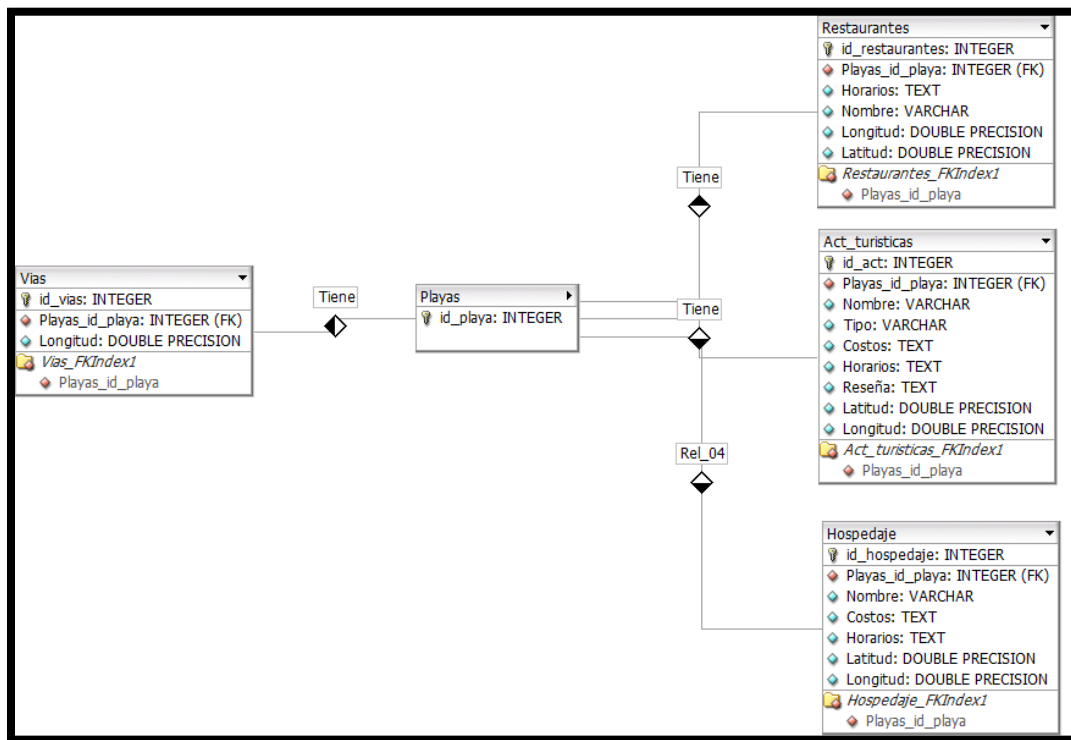


Figura 9. Diagrama Relacional para el desarrollo del SIG.

Fuente. Elaboración propia.

3.8 Adquisición de Información:

Una vez que se establecieron los datos requeridos para el sistema y los modelos anteriores, se realizó la adquisición de información. Con base en lo establecido por (Molina Torres & Ortiz Urbano, 2016), previo a la obtención de los datos, es necesario identificar las variables relevantes que serán utilizadas en el proyecto. Una vez identificadas, se utilizaron diferentes fuentes de información para obtenerlas. Las fuentes de información son clasificadas en primer, segundo y tercer orden, según la forma en que se capturan.

- **Fuente de información de primer orden:** Variables tomadas en campo, es decir, se visitó la zona de estudio, tomando las características de cada establecimiento, actividad, entre otros, que se tuvieron en cuenta para el estudio.
- **Fuente de información de segundo orden:** Datos de tipo espacial, que permitieron la creación del SIG, como lo son los barrios, vías, perímetros urbanos, etc.
- **Fuente de información de tercer orden:** Base datos de las actividades turísticas de la región, que se pretende obtener de forma conjunta al momento de tomar la información de primer orden.

3.9 Levantamiento de datos de campo

En la etapa exploratoria se realizó una salida de campo, donde se visitó la zona de estudio y se tomó información detallada de las actividades náuticas como los son: ubicación en coordenadas, precios, horarios, tipo de actividad que ofrecen y una breve descripción sobre el sitio. También, se obtuvo información sobre los restaurantes y hospedajes que hay en las playas de Juanchaco, Ladrilleros y La Barra, de tal forma que los usuarios puedan planificar su estadía, teniendo información de horarios, ubicación y para el hospedaje un precio del valor de una habitación por noche.

Para recoger la información se desplazó desde la ciudad de Buenaventura, en lancha hasta el puerto de Juanchaco. Esta información de las playas se recogió en formato análogo, sin embargo, para la ubicación de las actividades, los restaurantes y los hospedajes, se utilizó una aplicación llamada GPS Essentials para dispositivos móviles con sistema operativo ANDROID, la cual utiliza el GPS del celular para ubicar el lugar en el mapa, dándole coordenadas de "Latitud y Longitud" con precisión más o menos 2 m, lo que nos ofrece seguridad en cuanto a la ubicación.

Finalmente, a partir de la información obtenida, se realiza un análisis y procesamiento, para tener unos datos bien estructurados y normalizados que se puedan agregar a la base de datos con mayor facilidad.

3.10 Implementación de la base de datos

Para el manejo de la persistencia (almacenamiento de datos), se usó la base de datos PostgreSQL debido a su fácil manejo en consultas tanto alfanuméricas como espaciales, ya que con su complemento Postgis® es posible manipular información con facilidad; adicionalmente, PostgreSQL permite el manejo de datos en formato JSON, facilitando el envío de información entre API's y aplicaciones. Para la implementación de la base de datos se tiene en cuenta dos tipos de variables: las variables espaciales y las variables no espaciales, las variables espaciales, como su nombre lo indica, son datos que requieren de información geoespacial para poder ser visualizadas, esta información se conoce como capas, que tienen atributos alfanuméricos que le dan peculiaridad en la base de datos. Estos atributos son las variables no espaciales, Las variables que son de tipo alfanuméricas tienen características que son complementarias o tienen una relación con los datos de tipo espacial, y son manejadas por medio de SQL con el fin de procesarlas y almacenarlas en la base de datos.

❖ Variables espaciales:

- Vías
- Sitios de interés
- Actividades náuticas
- Restaurantes
- Hoteles
- Base área y Puerto
- Playas

En las siguientes imágenes muestran los mapas cartográficos que se crearon a partir de las variables espaciales, se observan las 3 categorías en el que se dividieron las actividades, que son Aventura, Deporte (caminata) y Recreación acuática, Además, las ubicaciones de los hoteles y los restaurantes. Esta cartografía se realizó con el software ArcGIS 10.5.

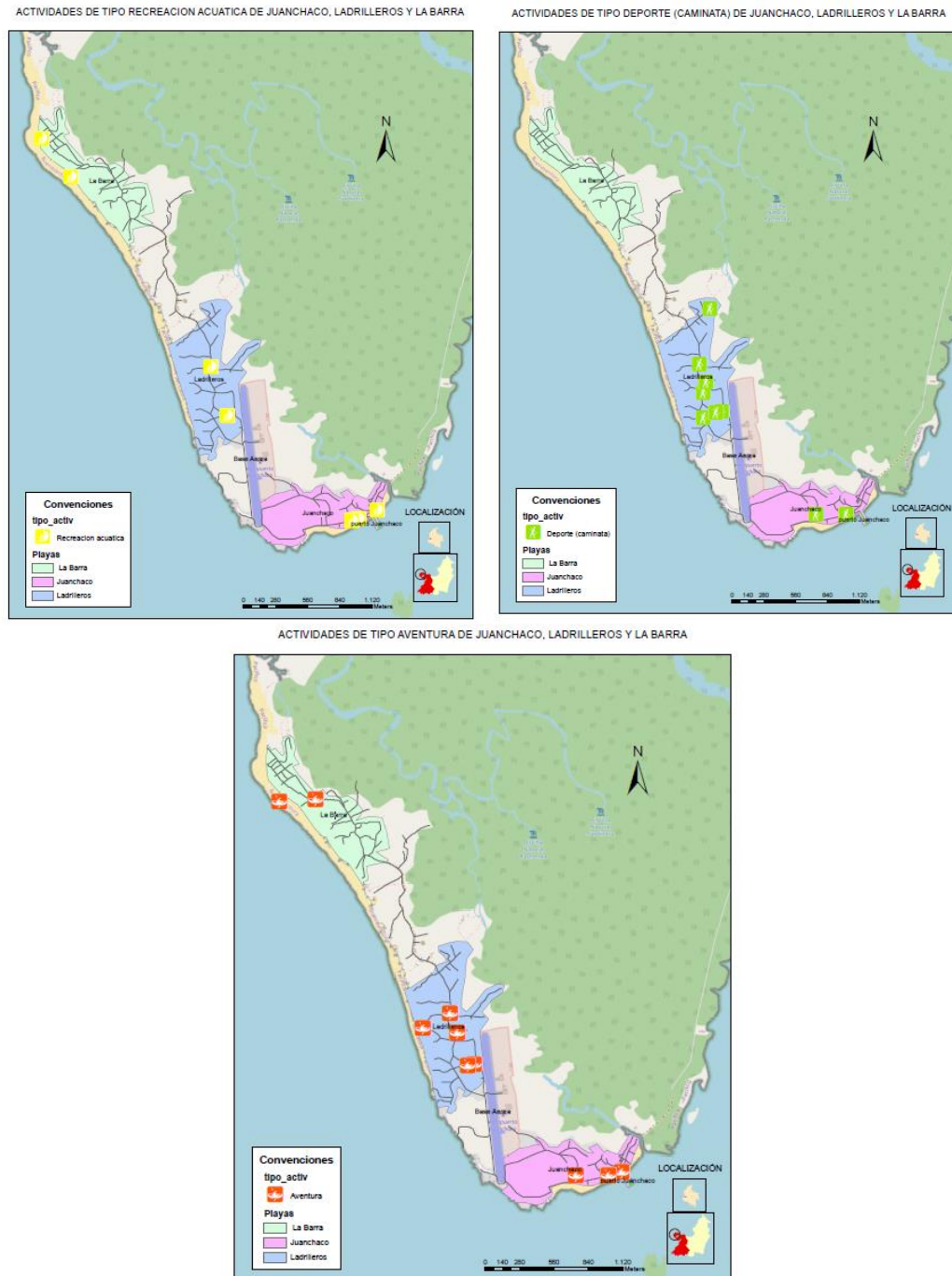


Figura 10. Cartografía de las ubicaciones de los diferentes tipos de actividades turísticas que ofrece las playas de Juancho, Ladrilleros y La Barra.

Fuente. Elaboración propia.

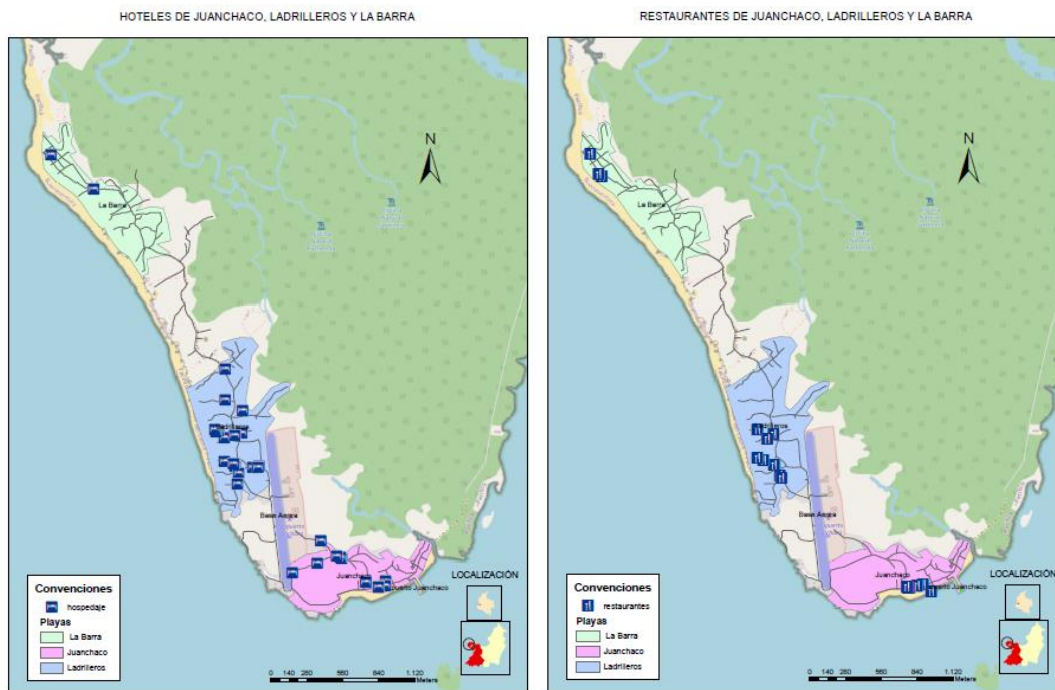


Figura 11. Cartografía de las ubicaciones de restaurantes y hoteles que se encuentran en las playas de Juanchaco, Ladrilleros y La Barra.

Fuente. Elaboración propia.

❖ Variables no espaciales:

- Nombres
- Precios
- Horarios
- Descripción
- Categorías
- Latitud
- Longitud

En la siguiente imagen (figura 12) se observa parte de los atributos que están contenidos en la entidad de actividades turísticas, estos atributos son alfanuméricos y les proporcionan información adicional a las capas espaciales. Entre los atributos se encuentran el nombre de la actividad, el horario y el precio, con el fin de que el usuario tenga información útil para que pueda planificar su visita a las playas.

Data Output

	gid [PK] entero	nombre carácter variable (254)	horario texto	precio texto	descripcion texto	latitud Precisión doble	longitud Precisión doble	geom geometría	categoria carácter variable
1	1	Arenas Del Pacifico Juegos A...	7:00 am - 1...	\$ 10.000 - ...	Se ofrecen activi...	3.929183500000022	-77.35345499999994	0101000020E6100...	Recreacion acuatica
2	2	Arenas del Pacifico Lanchas	7:00 am - 1...	\$ 10.000 - ...	Se ofrece paseo...	3.929183500000022	-77.35345499999994	0101000020E6100...	Recreacion acuatica
3	3	Arenas del Pacifico Buceo	7:00 am - 1...	\$ 10.000 - ...	Se ofrece activid...	3.929183500000022	-77.35345499999994	0101000020E6100...	Recreacion acuatica
4	4	Paseos Playa La Barra Asturias	7:00 am a ...	\$ 30.000 - ...	Se ofrece paseo...	3.9293122000000267	-77.35291999999998	0101000020E6100...	Aventura
5	5	Paseos Pesca deportiva Astur...	7:00 am a ...	\$ 300.00 - ...	Se ofrece paseo...	3.9293122000000267	-77.35291999999998	0101000020E6100...	Recreacion acuatica
6	6	Paseos 3 Marias Asturias	7:00 am a ...	\$ 60.000 - ...	Se ofrece paseo...	3.9293122000000267	-77.35291999999998	0101000020E6100...	Aventura
7	7	Paseos Cascadas Asturias	7:00 am a ...	\$ 75.000	Se ofrece paseo...	3.9293122000000267	-77.35291999999998	0101000020E6100...	Deporte (caminata)
8	8	Ferry Asturias	7:00 am a ...	\$ 80.000 - ...	Se ofrece paseo...	3.9293122000000267	-77.35291999999998	0101000020E6100...	Aventura
9	9	Paseos Manglares Asturias	7:00 am a ...	\$ 30.000 - ...	Se ofrece paseo...	3.9293122000000267	-77.35291999999998	0101000020E6100...	Deporte (caminata)
10	10	Pesca Deportiva Juanchaco	7:00 am a ...	\$ 100.000 - ...	Pesca Deportiva...	3.9292520000000764	-77.35339999999997	0101000020E6100...	Recreacion acuatica
11	11	Paseos Ecológicos Con Juan...	7:00 am a ...	\$ 20.000 - ...	Paseos ecologic...	3.938967000000048	-77.36419999999998	0101000020E6100...	Deporte (caminata)
12	12	Plaza Pública	6:00 am - 9...	Gratis	Entrada pública ...	3.940066600000023	-77.36651599999993	0101000020E6100...	Aventura

Figura 12. Base de datos, tabla de las actividades que ofrecen las playas, con sus atributos.**Fuente.** Elaboración propia.

3.11 Implementación de la base de datos

Inicialmente, se creó la base de datos llamada Navi_App, en la que se realizaron los primeros procesos. Primero, se ingresaron las capas que se tenían. con formato .shp, que eran la capa de vías de Buenaventura, las cuales se crearon a partir de vectorización, debido a que no se encontraron en ninguna plataforma que ofrece el estado, también, se añadió la capa del área que pertenece a cada playa, que al igual que las vías se crearon con vectorización. Las capas de actividades, hoteles, restaurantes y sitios de interés se crearon a partir de los datos tomados en campo.

La base de datos creada en el software PostgreSQL, contiene como tablas; actividades, hospedaje, restaurantes, área de las playas, sitios de interés, vías, base aérea de Juanchaco y el puerto. Además, se crearon disparadores (Triggers), que son consultas que se ejecutan de manera automática una vez haya un cambio determinado en la base de datos. (ver figura 13)

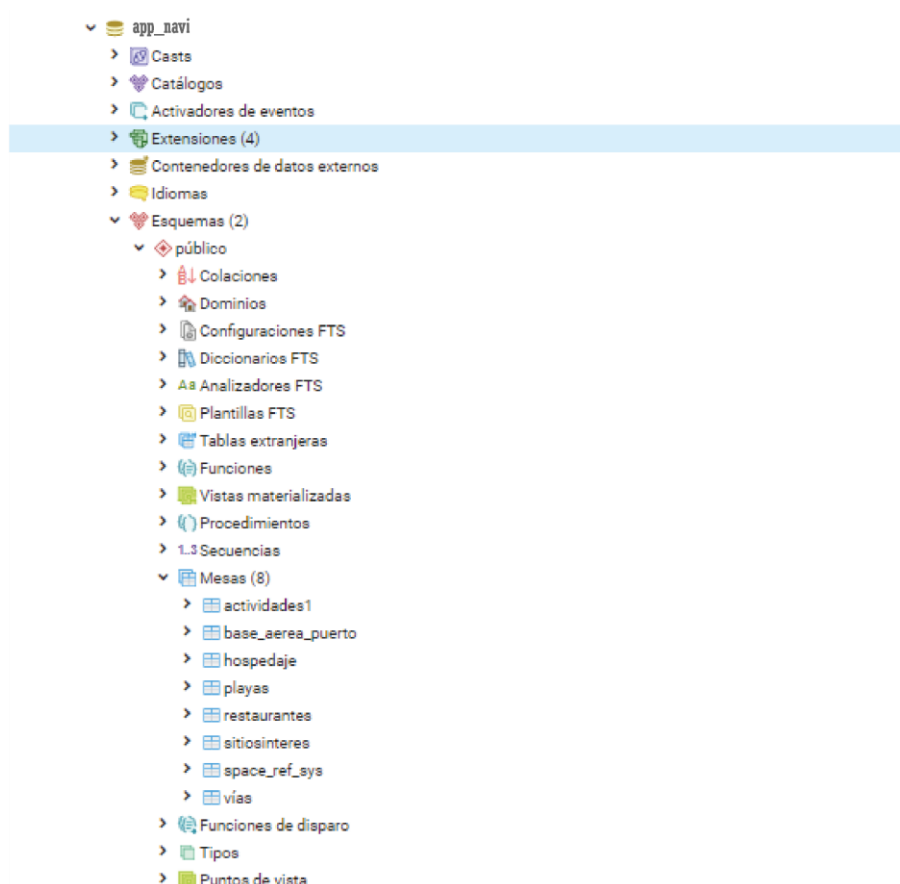


Figura 13. Creación de la base de datos.

Fuente. Elaboración propia.

CAPITULO 4. IMPLEMENTAR EL SIG EN UN APLICATIVO MÓVIL EN ANDROID, CON LA INFORMACIÓN ESPACIAL DEL TURISMO

4.1 Metodología de desarrollo de software Scrum

La etapa del desarrollo del aplicativo móvil se realizó mediante la metodología de desarrollo de software Scrum, que consiste en realizar entregas por partes del producto en periodos de tiempo corto, teniendo 3 pilares: la transparencia, inspección y adaptación. En la figura 14, se puede observar la forma en que se adaptará cada sprint para esta etapa.



Figura 14. Metodología de desarrollo de software Scrum.

Fuente. Elaboración propia.

Esta metodología define como primera instancia las diferentes actividades que se requieren para la ejecución del proyecto, las cuales se condensan en el documento llamado Product Backlog (ver tabla 6). Se muestra los módulos que conforman la aplicación, cada módulo tiene historias de usuario con las que se describe el proceso que realizará el sistema para responder a cada módulo. Para cada historia de usuario se le da una prioridad y se lleva el proceso.

Tabla 6. Product Backlog.

Product Backlog				
Módulo	Id	Nombre de Historia	Prioridad	Progreso
Gestión Usuarios	GU01	Registro Usuarios	Urgente	
	GU02	Login	Urgente	
Gestión información espacial	GIE01	Mostrar Actividades Náuticas	Alta	
	GIE02	Mostrar Restaurantes	Media	
	GIE03	Mostrar Hoteles	Media	
	GIE04	Calcular ruta	Media	
Admin	A01	Editar Información de Usuario	Media	
	A02	Agregar Ubicación	Alta	
	A03	Editar Ubicación	Alta	

Convenciones del progreso	
No empezado	
En progreso	
Terminado	

Recomendaciones	R01	Mostrar Actividades Náuticas Recomendadas	Media	
------------------------	---------------------	---	-------	--

Fuente. Elaboración propia.

Esta metodología Scrum, como se mostró en la figura 14, se realiza mediante Sprint, las cuales se establecen en un periodo de 2 a 3 semanas por actividad específica, lo que nos genera 14 sprints, estos se les relaciona un tiempo determinado para entrega como se puede observar en la tabla 7, el release plan, especifica cada actividad con sus tiempos de inicio y entrega. Para cada Sprints se hizo una reunión con la codirectora, antes y después de la actividad con el fin de determinar los requisitos a implementar y el revisar el resultado del sprint. Cada sprint contempla la aplicación de pruebas de aceptación y usabilidad. Esta etapa se realizó en 11 meses y la documentación fue consignada al final de cada proceso en el documento final del trabajo de grado, el cual se fue construyendo durante el proyecto.

Cabe destacar que, al momento de realizar la definición del proyecto, se especificó una versión de sistema operativo mínimo Jelly Bean 4.2 (Api 17), sin embargo, para aprovechar al máximo las características de la aplicación es recomendado Android 10 (API 29). También es necesario un tamaño de pantalla de mínimo 5.5 pulgadas (13.9 cm). De esta forma se pueden visualizar e interactuar con todos los elementos de la aplicación de forma óptima. Se especificó de esta manera para un mejor funcionamiento y compatibilidad de las librerías y API's implementadas en la aplicación.

Tabla 7. Release Plan.

Release Plan				
Iteración	Fecha de Inicio	Fecha Final	Descripción de la tarea	Progreso
Spring 1	01 de febrero	16 de febrero	Establecer requerimientos del producto	100%
Spring 2	17 de febrero	5 de marzo	Definir y organizar historias de usuario	100%
Spring 3	6 de marzo	21 de marzo	Definir arquitectura del proyecto	100%
Spring 4	22 de marzo	6 de abril	Registro de Usuarios	100%
Spring 5	7 de abril	22 de abril	Login	100%
Spring 6	23 de abril	13 de mayo	Calcular ruta	100%
Spring 7	14 de mayo	4 de junio	Mostrar Actividades Náuticas	100%
Spring 8	5 de junio	12 de junio	Mostrar Restaurantes	100%
Spring 9	12 de Julio	19 de Julio	Mostrar Hoteles	100%
Spring 10	20 de Julio	10 de agosto	Editar Información de Usuario	100%
Spring 11	11 de agosto	1 de septiembre	Agregar Ubicación	100%
Spring 12	2 de septiembre	23 de octubre	Actualizar Ubicación	100%

Spring 13	24 de octubre	14 de noviembre	Eliminar Ubicación	100%
Spring 14	15 de noviembre	5 de enero	Mostrar Actividades Náuticas Recomendadas	100%

Fuente. Elaboración propia.

A partir del Product Backlog se realizaron las historias de usuario, que son documentos cortos que especifican los requerimientos funcionales, es decir, lo que se espera que la aplicación pueda realizar. A manera de ejemplo, la tabla 8 muestra la historia de usuario [GIE01](#). Para ver la totalidad de historias de usuario planteadas para el proyecto ingresar al siguiente enlace <https://docs.google.com/spreadsheets/d/1XoJy1fDU5-8pN2ikKWnK3TAakcYW0Z4HaX5qlwNO0h0/edit#gid=553425353>.

Tabla 8. Historia de Usuario de las actividades náuticas.

Historia de Usuario			
Autor	Nicolás Rojas Sharon Rojas	Módulo	Información espacial
Nombre HU	Mostrar actividades náuticas	Identificador	GIE01
Dependencia		Prioridad	
Actor(es)	Usuario/empresa	Iteración	
Puntos asignados			
Descripción			
Como usuario o empresa deseo poder visualizar las actividades náuticas que se pueden realizar en la zona y al presionar sobre cada una, ver diversa información como el nombre, precios, horarios y una respectiva descripción que me permita decidir cual/cuales visitar de acuerdo con mis preferencias.			
Observaciones			
1. Para visualizar las actividades náuticas que se presentan en la zona, el usuario debe presionar sobre el menú desplegable de la izquierda, y una vez desplegado presionar sobre la opción de actividades náuticas			
2. Al presionar sobre actividades náuticas, se mostrarán todas las ubicaciones relacionadas con este tipo de actividades, que se mostrarán como marcadores			
3. Al presionar sobre un marcador se mostrará toda la información respecto a las actividades como el nombre de la actividad, los diferentes horarios que manejan, los precios por realizar estas actividades y una descripción de cada una.			
4. De igual forma, se puede ocultar dicho menú, presionando sobre el mismo marcador.			
Bocetos	Mapa		
Creada	26/11/2018	Modificada	3/02/2021

Fuente. Elaboración propia.

Una vez se crean las historias de usuario, se procede a crear bocetos en los que se diseña más o menos como va a crear la interfaz de cada historia de usuario. En la figura XX se muestra un ejemplo, que es el boceto de la historia de usuario del inicio de sesión. Los demás bocetos se pueden visualizar en el siguiente enlace: https://drive.google.com/drive/folders/1z1Vde_wfDaDSVdrIRf73ClySPs1FA.

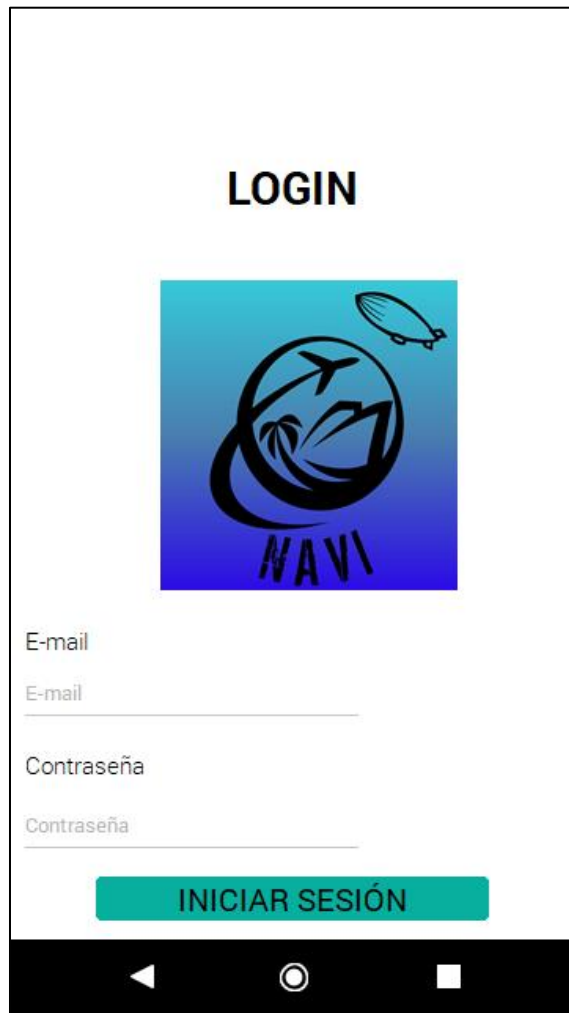


Figura 15. Boceto creado para la interfaz del inicio de sesión.

Fuente. Elaboración propia.

Parte importante para el desarrollo de la aplicación, es la selección del lenguaje de programación y de la librería para visualizar el mapa y sus interacciones. Para el desarrollo se optó por el lenguaje Java y el IDE Android Studio ya que son lenguajes y entornos de desarrollo oficial para aplicaciones nativas Android. Para el manejo de mapas se seleccionó la SDK de Mapbox®, debido a que permite un uso gratuito mensual bastante alto (25,000 usuarios activos, 50,000 visualizaciones de mapas, 100,000 consultas de ruta, entre otros), además de esto, su cartografía está basada en Open Street Maps®, y debido a esto contiene vías de la zona de estudio que no están contenidas en Google Maps®, presentando una diferencia significativa a la hora de hacer uso del cálculo de rutas. También, se escogió esta librería debido que por medio del software JOSM se puede realizar actualizaciones de las capas base, como vías, edificios, entre otros.

A continuación, se muestran evidencias de la funcionalidad implementada para los módulos que conforman el proyecto: módulo de gestión de usuarios, módulo de gestión de información espacial, módulo de administrador y módulo de recomendaciones; los 2 primeros serán abordados en este capítulo y los 2 últimos en los capítulos siguientes.

4.2 Módulo de gestión de usuarios

4.2.1 Análisis

Inicialmente, se realizaron 2 historias de usuario y se establecieron los pasos para el registro de usuarios e inicio de sesión; en el registro se determinaron los datos que debe ingresar el usuario en el formulario, estos son el nombre, correo, contraseña y tipo de usuario, donde se definieron el usuario normal y la empresa. A forma de ejemplo se puede ver la tabla 9 que es la historia de usuario del registro de usuario, [GU01](#). Las 2 historias se pueden ver en el siguiente enlace:

<https://docs.google.com/spreadsheets/d/1XoJy1fDU5-8pN2ikKWnK3TAakcYW0Z4HaX5qlwNO0h0/edit#gid=0&range=A1:K13>.

Tabla 9. Historia de usuario del registro de usuarios en el sistema.

Historia de Usuario			
Autor	Nicolás Rojas Sharon Rojas	Módulo	Usuarios
Nombre HU	Registro de usuarios	Identificador	GU01
Actor(es)	Cliente	Sprint	5
Descripción			
Como usuario deseo poder registrarme para acceder al sistema, pudiendo decidir el tipo de rol con el que voy a acceder (Usuario regular o Empresa)			
Observaciones			
1. Se presentarán los campos que deben ser llenados por el usuario que se va a registrar en la aplicación (Nombre, Email y Contraseña) además de un desplegable donde el usuario decida el tipo de rol que va a manejar en la aplicación (Usuario regular o Empresa) 2. Cuando el cliente presione sobre el botón de registro, la información ingresada será cargada a una base de datos 3. Si los datos fueron ingresados correctamente, la aplicación enviará un mensaje diciendo que los datos se cargaron correctamente y enviará al usuario registrado a una pantalla donde podrá realizar el inicio de sesión respectivo; en caso contrario, enviará un mensaje diciendo que no se ingresaron correctamente los datos para el registro			

Bocetos	MainActivity	Registro	
Creada	29/11/2018	Modificada	03/02/2021

Fuente. Elaboración propia.

En el momento en que el usuario entra en la aplicación podía ver 2 botones con funciones diferentes; el inicio de sesión si tenía una cuenta asociada o el registro en caso de ser necesario crear una; En caso de tener una cuenta debía llenar los datos del correo y la contraseña definida en el registro, en caso de no contar con una, llenar un formulario con la información del nombre, correo, contraseña y tipo de usuario y volver al inicio de sesión y llenar los datos para ingresar. Esto se muestra con mayor claridad en el siguiente diagrama de actividades:

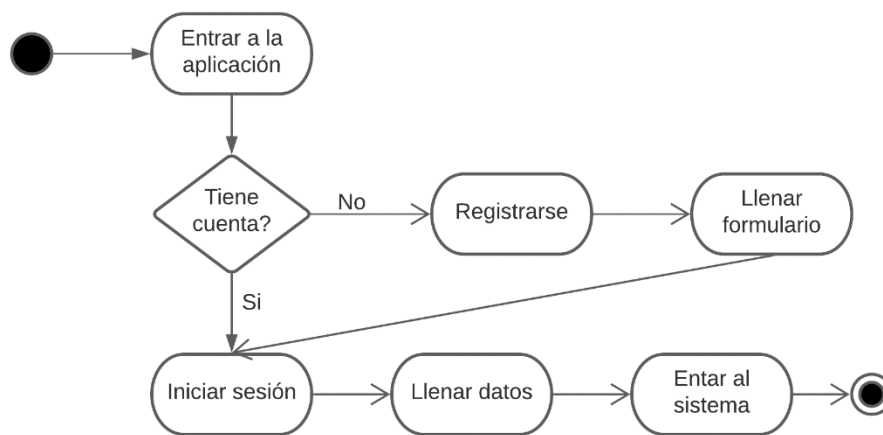


Figura 16. Flujograma del inicio de sesión para el ingreso al sistema.

Fuente. Elaboración propia.

Con el fin de realizar este módulo, se realizaron 2 actividades; una de registro y una de login, y para cada una se utilizó Firebase, que contiene una base de datos en tiempo real, en la que se puede almacenar la información suministrada por los usuarios en los formularios, y acceder a ella con facilidad, ya que cuenta con funciones ya diseñadas, que permiten manipular la base de datos de forma sencilla. Para estas actividades se utilizó la autenticación por correo y contraseña, que permite añadir información adicional como el tipo de usuario, de gran importancia para la división de roles.

4.2.2 Implementación

Registro

En esta pantalla, se muestra un formulario que debe ser llenado con la información requerida, para de esta forma, agregarlos a la base de datos y generar un usuario nuevo. Para esto, se utilizó el método de autenticación mediante correo y contraseña brindado por Firebase. Para el diseño se crearon 3 cajas de texto y un spinner. En las cajas de texto se llenó la información de nombre, correo y contraseña y en el spinner se mostraban las opciones del tipo de usuario. Con esta información se creó la lógica para el almacenamiento en la base de datos realizando inicialmente una función de verificación que determina si todos los campos fueron llenados, y de esta manera subir esta información a la base de datos Firebase, como se muestra en el anexo B.

Una vez que el sistema valida que los campos no se encontraran vacíos, utilizó la función “CreateUserWithEmailAndPassword” con la que agregó un usuario a Firebase con los datos del correo y la contraseña. Adicional a eso, utilizando un hashmap, ingresó toda la información que va a ser subida a la base de datos y creó un nuevo nodo en Users que contiene la información del nuevo usuario. Si la operación se realizaba correctamente, el usuario era enviado al login, donde podía utilizar esos datos para ingresar al sistema. Esto dio como resultado la siguiente ventana:

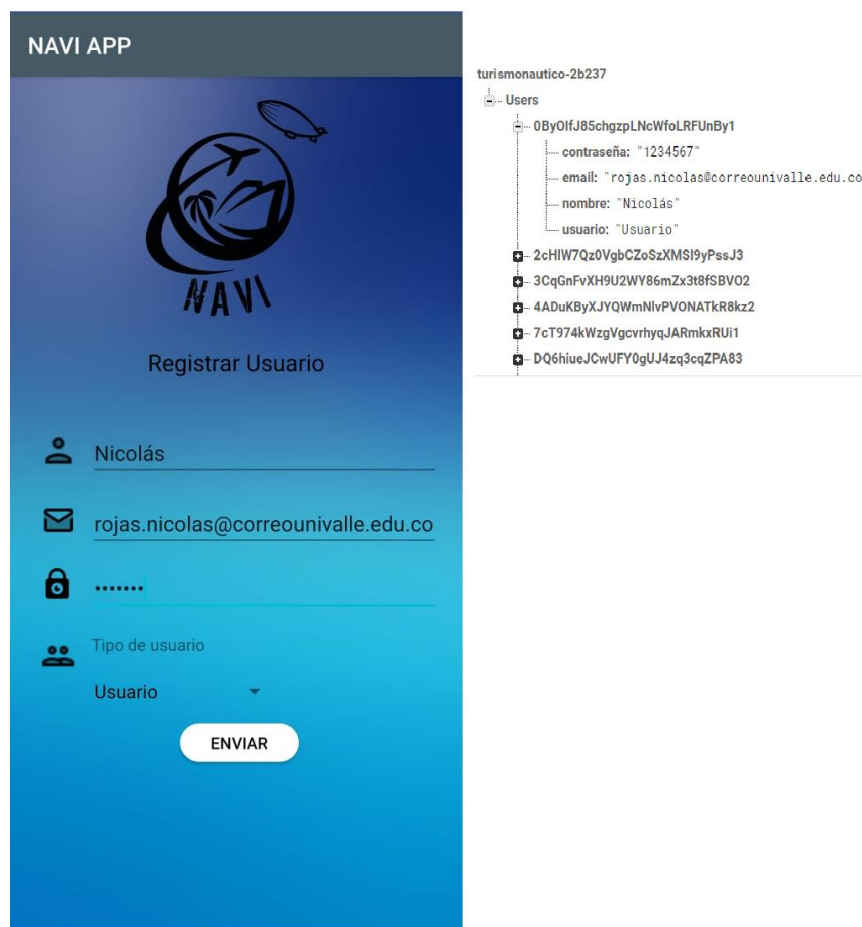


Figura 17. Interfaz de registro de usuario. Código de validación.

Fuente. Elaboración propia.

Login

De forma similar al registro, al entrar en esta pantalla, se muestra un formulario en el que era necesario llenar los datos de correo y contraseña, esto permitiría entrar al módulo de gestión espacial. Para que esto funcione correctamente, el usuario debe haber creado una cuenta anteriormente y llenar los datos mencionados. La aplicación valida esta información y en caso de ser llenada incorrectamente o que el usuario no existiera, muestra un mensaje diciendo que los datos eran incorrectos y no le permitirá entrar al sistema. En caso de ser ingresados correctamente, envía al usuario al siguiente módulo y le mostrará un mensaje de bienvenida.

Para la realización de esta actividad se utilizó la función de autenticación de Firebase que permite ingresar al sistema a partir del correo y contraseña, con el código que se muestra en el Anexo C, con el evento `LoginUsuarios` se estableció si los datos fueron llenados, para posteriormente, con la función de autenticación `signInWithEmailAndPassword` determinar si el correo y contraseña fueron ingresados correctamente; si la operación fue realizada correctamente, envía al usuario a `MapActivity`; en caso contrario le muestra un mensaje diciendo que no se pudo iniciar sesión. En la figura 18 se observa las dos interfaces; cuando se están ingresando los datos para el inicio de sesión y la ventana una vez iniciada la sesión, que contiene el mapa principal, donde se muestran sitios de interés como el centro médico.

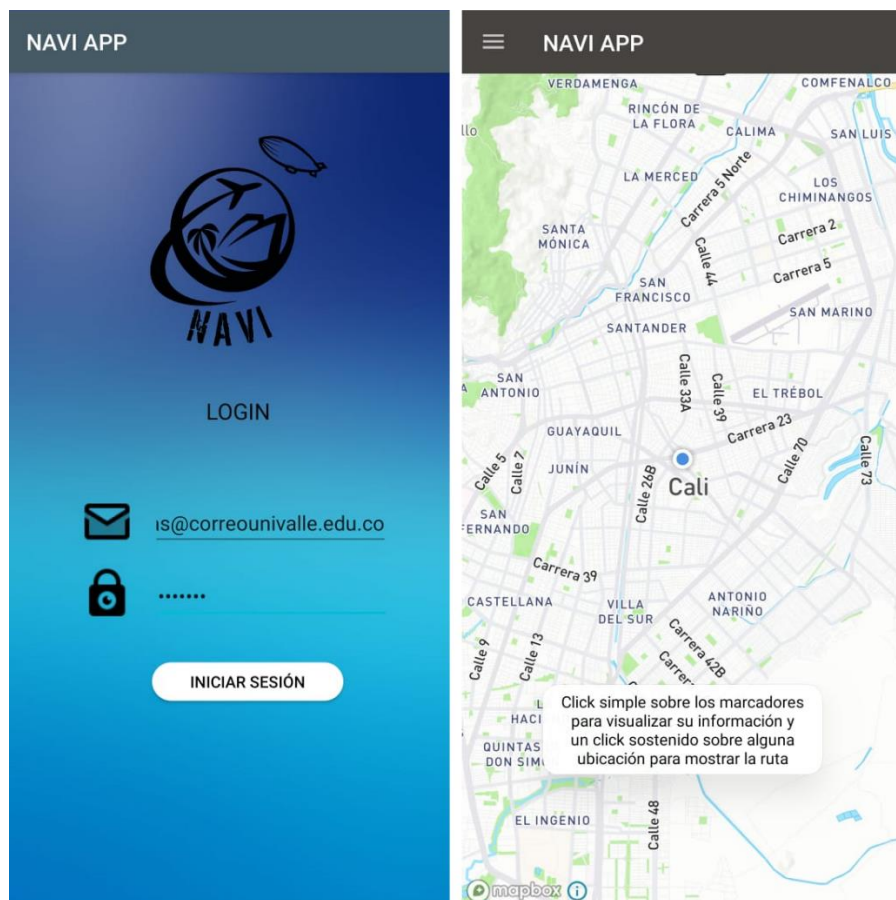


Figura 18. Interfaz del inicio de sesión y del mapa principal, con sitios de interés.

Fuente. Elaboración propia.

Una vez dentro del sistema, y al presionar sobre la barra de menú, se desplegarán un conjunto de opciones para navegar sobre la aplicación, entre ellas se encuentra la de configuración, que permite ver los datos del usuario y realizar cambios en el nombre, correo y contraseña, y al presionar el botón de enviar, volverá al mapa, y al volver a la ventana de configuración podrá ver de nuevo la información del usuario con los cambios realizados. Cabe destacar que la contraseña se encuentra oculta por razones de seguridad.

Este fragmento se realizó utilizando nuevamente las funciones de autenticación proporcionadas por Firebase, en este caso se realizó un evento en el que, al dar clic al botón, actualice la información correspondiente al nodo en que se encuentra la información del usuario que tiene una sesión activa, permitiendo cambiar el nombre, correo y contraseña de forma sencilla (Ver anexo D).

Inicialmente, se recopilaron los datos que se encuentran en las cajas de texto, estas previamente fueron llenadas con la información actual que iba a ser editada, trayéndola desde la base de datos. Al realizar cambios y presionar el botón, el evento `OnClick` determina los valores que se encuentran en las cajas de texto, convirtiéndolas en `String` y fueron añadidas a un `hashmap`; posteriormente, se determinó el `id` del nodo en el que se encuentra el usuario y con la función `“updateChildren”` realizó cambios a la base de datos, pasando como parámetro el `hashmap`. Si hubo algún error al actualizar le muestra un mensaje al usuario, en caso contrario lo envía a la pantalla del mapa, donde puede visualizar los cambios realizados al entrar nuevamente al fragmento de configuración; el resultado es el siguiente:

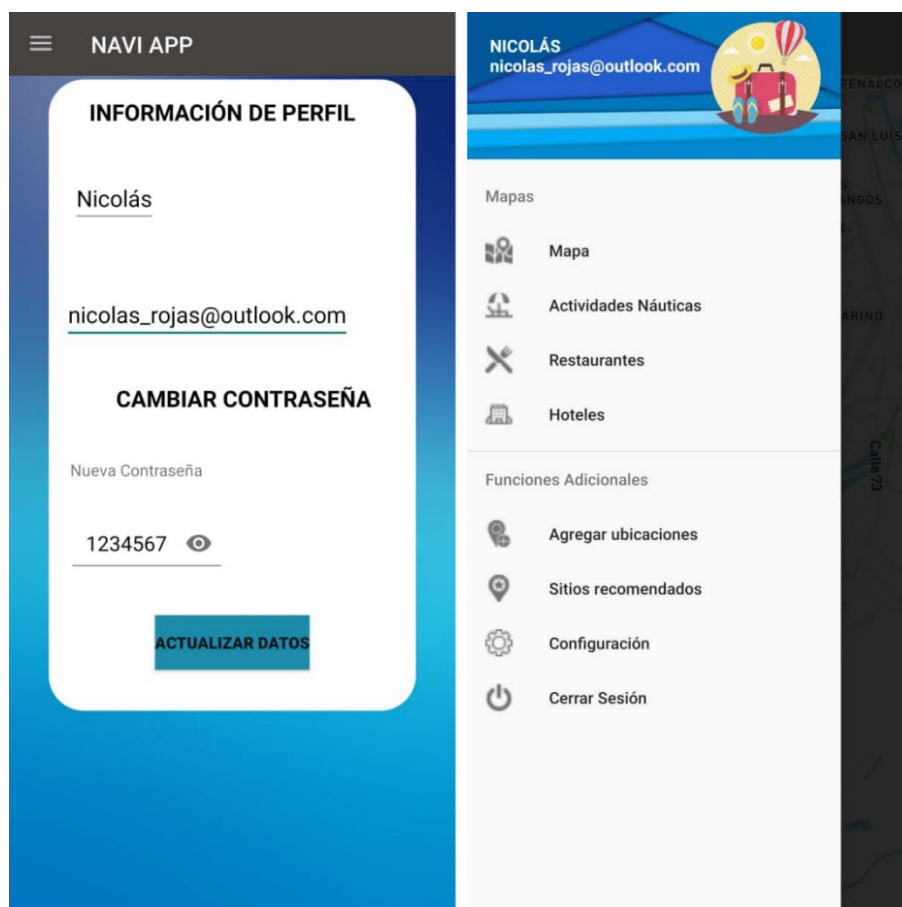


Figura 19. Interfaz de la configuración de datos del usuario, menú de la aplicación.

Fuente. Elaboración propia.

4.3 Módulo de gestión espacial

4.3.1 Análisis

Se realizaron 4 historias de usuario que compilan los requerimientos para este módulo, que consisten en la visualización de información de las actividades, hoteles y restaurantes sobre el mapa y la generación de la ruta más cercana desde la ubicación del usuario a un punto en particular; además de las condiciones que debe realizar para poder ejecutar estas funcionalidades. A modo de ejemplo se mostrará la historia de usuario de la visualización de actividades náuticas; las demás se pueden encontrar en el siguiente vínculo:

<https://docs.google.com/spreadsheets/d/1XoJy1fDU5-8pN2ikKWnK3TAakcYW0Z4HaX5qlwNO0h0/edit#gid=553425353&range=A1:K28>.

Tabla 10. Historia de usuario del visualizador de actividades náuticas.

Historia de Usuario			
Autor	Nicolás Rojas Sharon Rojas	Módulo	Información espacial
Nombre HU	Mostrar actividades náuticas	Identificador	GIE01
Dependencia		Prioridad	
Actor(es)	Usuario/empresa	Iteración	
Puntos asignados			
Descripción			
Como usuario o empresa deseo poder visualizar las actividades náuticas que se pueden realizar en la zona y al presionar sobre cada una, ver diversa información como el nombre, precios, horarios y una respectiva descripción que me permita decidir cual/cuales visitar de acuerdo con mis preferencias.			
Observaciones			
1. Para visualizar las actividades náuticas que se presentan en la zona, el usuario debe presionar sobre el menú desplegable de la izquierda, y una vez desplegado presionar sobre la opción de actividades náuticas 2. Al presionar sobre actividades náuticas, se mostrarán todas las ubicaciones relacionadas con este tipo de actividades, que se mostrarán como marcadores 3. Al presionar sobre un marcador se mostrará toda la información respecto a las actividades como el nombre de la actividad, los diferentes horarios que manejan, los precios por realizar estas actividades y una descripción de cada una. 4. De igual forma, se puede ocultar dicho menú, presionando sobre el mismo marcador.			
Bocetos	Mapa	Hoteles	
Creada	26/11/2018	Modificada	3/02/2021

Fuente. Elaboración propia.

4.3.2 Implementación

En el desarrollo de este módulo, se utilizó la base de datos que se encuentra en PostgreSQL, que contiene la información acerca de las actividades náuticas, restaurantes y hoteles de las playas de Juanchaco, Ladrilleros y La Barra, además, de su ubicación. A partir de esto, se realizó una API que convierte la información en la base de datos en un objeto GeoJSON, que es uno de los tipos de dato que recibe Mapbox para ser añadidos al mapa, y a partir de esto, se desarrolló un evento en el que, al dar clic sobre algún marcador, muestre la información no espacial, que es traída desde la base de datos. De forma paralela, se realizó un evento en el que, si un usuario daba un clic sostenido sobre alguna parte del mapa, mostrara la ruta desde su ubicación actual hasta el punto señalado.

Inicialmente para el desarrollo de la API, se creó la base de datos, utilizando el servidor de aplicaciones heroku, que permite añadir bases de datos relacionales y no relacionales. Estas se alojan en el servidor, haciendo posible su uso de forma remota. Con la base de datos creada y los datos de actividades, hoteles y restaurantes añadidos, se crearon triggers para cada una, permitiendo generar la geometría de la ubicación de forma automática al crear un nuevo punto, ingresando únicamente las coordenadas.

A partir de la información en la base de datos y los triggers, se realizó una API por cada tipo de ubicación, utilizando un archivo php, con el que se realizó la consulta y se generó el objeto GeoJSON. De esta forma, se tiene una url en la que se muestra el objeto GeoJSON resultante de la consulta a la base de datos, sin embargo, este enlace se encontraba local, y por lo tanto se realizó el despliegue de cada API sobre el servidor Heroku, y de esta forma acceder a la información desde cualquier dispositivo, accediendo a la url brindada por el servidor. La API realizada se muestra a continuación:

```
<?php
include('configuracion.php');

$sql="SELECT row_to_json(fc)
FROM ( SELECT 'FeatureCollection' As type, array_to_json(array_agg(f)) As features
FROM (SELECT 'Feature' As type
, ST_AsGeoJSON(lg.geom)::json As geometry
, row_to_json((SELECT l FROM (SELECT name, horario, precio, descripcion, latitud, longitud) As l
)) As properties
FROM actividades1 As lg ) As f ) As fc";

$query = pg_query($conexion,$sql);
$row = pg_fetch_row($query);
echo $row[0];
?>
```

Figura 20. Archivo en formato PHP para la conversión de los datos a Geojson.

Fuente. Elaboración propia.

Inicialmente, se incluyó un archivo de configuración que trae el acceso a la base de datos, en el que se encuentran todos los datos como el servidor, el usuario, contraseña, puerto y base de datos, para conectarse a ella. Con la conexión realizada, se hizo la consulta que convierte la geometría y los datos de horario, precio, descripción, latitud y longitud en un objeto Geojson y finalmente, mostrarlo en la url en que está el archivo php.

En la aplicación, se creó un evento que al ingresar a la url del objeto Geojson, lo lea, para posteriormente ser mostrado sobre el mapa como un marcador que, al dar clic sobre él, abrirá una burbuja en la que se muestran los datos desde la base de datos. Esto se realizó en varios eventos separados dependientes, de los que se va a explicar visualmente el evento del clic sobre el marcador.

El evento que permite leer al archivo Geojson, consistió en una conexión HTTP a la url en que se encuentra el objeto y por medio de la clase buffer Reader, traía el Geojson y lo convertía en String, para posteriormente, ser leído por el evento que transforma el JSON en Feature Collection. Con este evento se obtuvieron los datos del evento anterior y se transformaron utilizando el método “FeatureCollection.fromJson”.

Luego, se creó el evento que configura la imagen que se iba a mostrar como marcador y la ventana que se mostraría al darle clic. El primero traía la imagen que debía estar en la carpeta de recursos gráficos; y el segundo configuraba

la ventana determinando el título, el ancho de los marcadores, intervalo de distancia, etc., además, de los datos que debían ser mostrados sobre la burbuja de texto.

Finalmente, con las configuraciones llevadas a cabo, se realizó el evento que permite mostrar la información determinada sobre la burbuja de texto al dar clic sobre cualquiera de los marcadores mostrados en el mapa. De esta forma, al dar clic sobre algún marcador mostraría la información de esa ubicación desde la base de datos como Geojson. El evento que permitió mostrar la información al presionar sobre algún marcador se puede ver en el anexo E.

Se obtuvo una lista donde se encuentran los datos de los marcadores y la ubicación, se determinó si un marcador fue clicado, y en ese caso ubicar al que fue presionado. Se obtuvieron los datos del nombre y las otras propiedades como horario, precio y descripción de acuerdo con el estado, que cambia dependiendo el marcador seleccionado. Esto permitió desplegar la burbuja de texto al presionar sobre alguno de los iconos y mostrar la información correspondiente que viene desde la base de datos. Esto da como resultado lo siguiente:

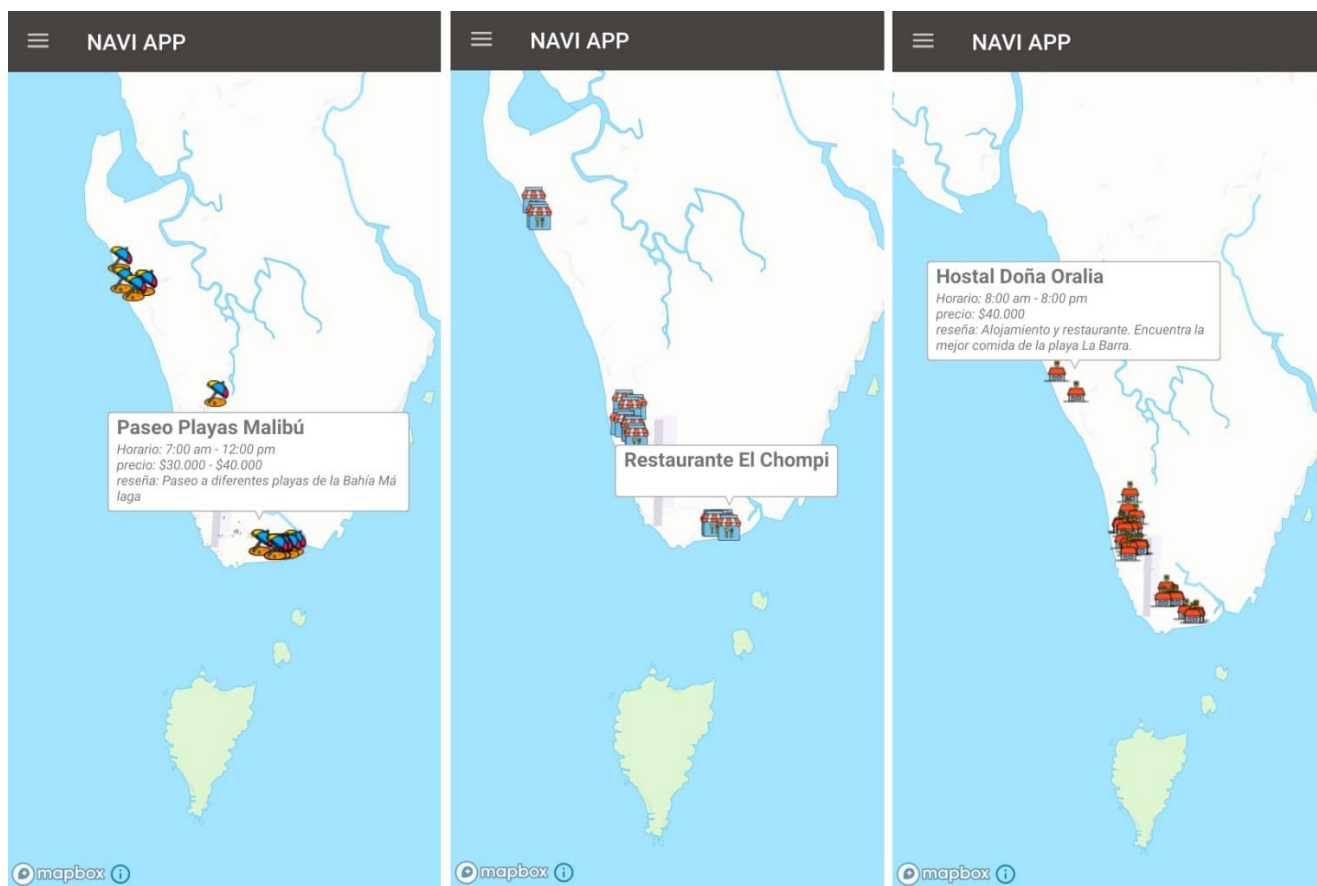


Figura 21. Visualización de las burbujas para las actividades, restaurantes y hoteles.

Fuente. Elaboración propia.

Paralelo a la información sobre las ubicaciones, se creó una función para obtener la ruta desde la ubicación actual al lugar seleccionado. Para ello, se determinó la ubicación actual proporcionada por el GPS, y con ello se generó un evento utilizando las librerías de Mapbox en el que, al dar clic sostenido, se ubica un marcador y dibuja la ruta desde la posición del usuario hasta el punto señalado, permitiendo al usuario saber cómo desplazarse de un lugar a otro, utilizando las vías del lugar.

Para determinar la localización, se pidieron permisos al usuario para utilizar el GPS del celular, y de esta manera, poder capturar las coordenadas en las que se encontraba el usuario, visualizando un punto en el mapa, que es donde se ubica el usuario. Para obtener las coordenadas del dispositivo se utilizó la clase “locationComponent” que utiliza la cámara como medio de rastreo de las coordenadas.

Al tener la ubicación del usuario, se realizó el evento que genera la ruta a partir del clic sostenido. Para esto se creó un objeto Geojson, que fue convertido a Feature, y de esta forma, ubicaba un marcador al presionar sobre el mapa y determinaba esa ubicación como punto de destino para la ruta. Para la creación de esta se utilizó el método “DirectionsResponse”, que obtiene las indicaciones y con esto dibuja la ruta a partir de esas indicaciones. El código utilizado para la generación de la ruta se puede ver en el anexo F.

Se obtuvo el token generado por Mapbox que permite el uso de la API de direcciones y que sea único para la aplicación. Posteriormente, se construyeron las indicaciones de ruta, utilizando el punto de inicio, que es proporcionado por las coordenadas del GPS, y el punto de destino que se genera al dar clic sostenido sobre el mapa, si al recibir respuesta de la API, se obtuvieron las indicaciones para llegar al punto indicado, la ruta fue dibujada sobre el mapa. En caso de ubicar un nuevo punto, la ruta dibujada era removida, y realizaba nuevamente el proceso anterior para crear la ruta al nuevo punto. El resultado se muestra en la siguiente pantalla:

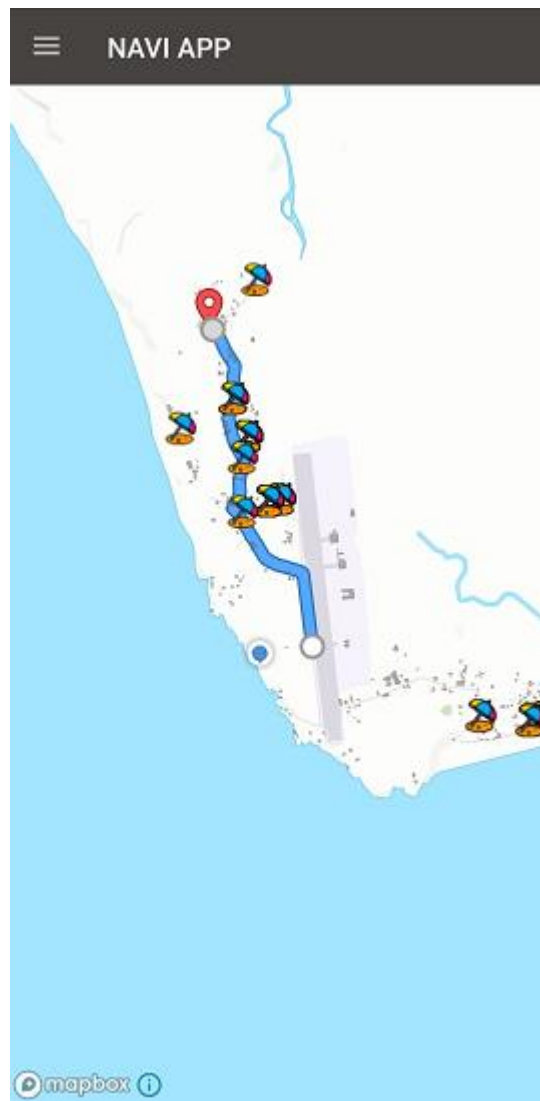


Figura 22. Ejemplo de la ruta que es generada por el sistema.

Fuente. Elaboración propia.

CAPITULO 5. DESIGNAR UN MÓDULO EN QUE EL USUARIO COMO ADMINISTRADOR PUEDA ACTUALIZAR LA INFORMACIÓN EN TIEMPO REAL Y REALIZAR RESPECTIVOS MANTENIMIENTOS.

5.1 Módulo de Administrador

5.1.1 Análisis

Para la realización de este módulo se crearon 3 historias de usuario en las que se definieron las actividades que puede realizar el administrador, que consisten en agregar, editar y eliminar ubicaciones (actividad náutica, hotel o restaurante), además, se determinaron los pasos que debe realizar el administrador para poder realizar estas actividades. Como demostración se puede ver la historia de usuario realizada para la inserción; las otras historias de usuario se pueden ver en el siguiente vínculo: <https://docs.google.com/spreadsheets/d/1XoJy1fDU5-8pN2ikKWnK3TAakcYW0Z4HaX5qlwNO0h0/edit#gid=367695470&range=A1:K27>.

Tabla 11. Historia de usuario de agregar nueva ubicación de actividad, hotel o restaurante.

Historia de Usuario			
Autor	Nicolás Rojas Sharon Rojas	Módulo	Admin
Nombre HU	Agregar ubicación	Identificador	A02
Dependencia		Prioridad	
Actor(es)	Empresa	Iteración	
Puntos asignados			
Descripción			
Como empresa deseo poder agregar nuevas ubicaciones a la aplicación, decidiendo el tipo (actividad náutica, hotel o restaurante) agregando diferentes campos como nombre, los precios, horarios y una descripción			
Observaciones			
1. el tipo de usuario ingresado en la aplicación debe ser administrador para poder acceder al formulario de adición de ubicaciones 2. El usuario debe ingresar en la lista de opciones y seleccionar la opción de agregar actividad, hotel o restaurante 3. Al seleccionar esta opción, se le pedirá al administrador que seleccione en el geovisor, la ubicación de la actividad nueva que va a ingresar, después, se desplegará un formulario en el que se pedirán toda la información necesaria para agregar la nueva ubicación a la base de datos. Los datos por ingresar para cualquier opción (excepto restaurantes) son: Tipo, nombre, precios (pesos colombianos), horarios (horario militar de 24 horas) y reseña. en caso de ser un restaurante solo se pedirá el nombre 4. Si los datos fueron suministrados correctamente, al presionar el botón de agregar, la información será enviada a la base de datos y la aplicación enviará un mensaje de éxito y enviará al usuario a la pantalla del mapa, donde podrá visualizar si el punto fue agregado de forma correcta. 5. En caso contrario, la aplicación enviará un mensaje de error, pidiendo que se digiten correctamente los datos o se llenen todos los campos.			
Bocetos	AgregarUbicacion	Mapa	
Creada	26/11/2018	Modificada	3/02/2021

Fuente. Elaboración propia.

En este módulo, se desarrollaron 2 tipos de usuario, el primero usuario es el cliente o turista común que usa la aplicación para visualizar los datos e interactuar con las diferentes características. El otro es la empresa, que además de interactuar con la aplicación, puede realizar modificaciones a la base de datos, con la posibilidad de hacer operaciones de CRUD sobre las actividades náuticas, hoteles y restaurantes. Para ello, al realizar el registro de usuario se le pidió definir el tipo, con el fin de que al entrar a este módulo se hiciera una comprobación y permitiera entrar a realizar procesos en la base de datos únicamente a las empresas. Esto se ve condensado en el diagrama de casos de uso que se muestra a continuación:

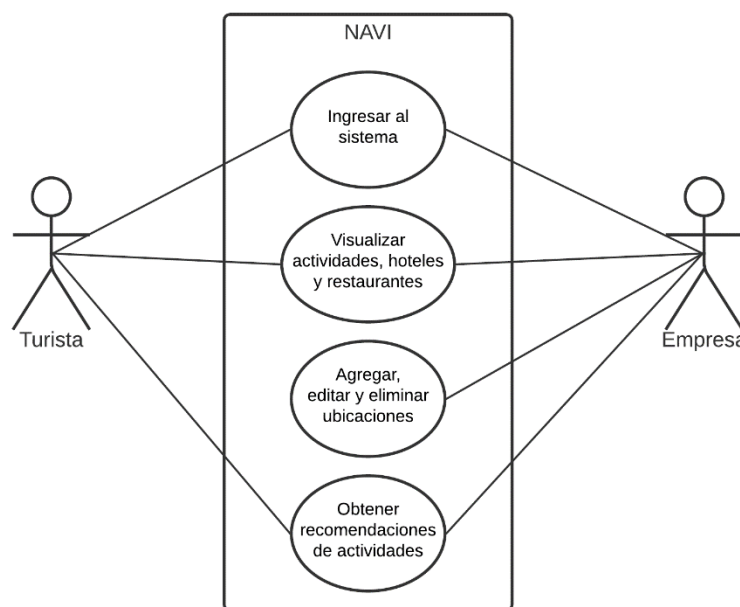


Figura 23. Diagrama de caso de uso de las funciones que puede realizar cada tipo de usuario.

Fuente. Elaboración propia.

5.1.2 Implementación

Al entrar en el módulo de administrador se comprobó el tipo de usuario, en caso de ser un usuario regular, la aplicación le mostraba un mensaje diciendo que el usuario debería tener permisos de administrador y no le permitía seguir, en caso de ser un usuario empresa, le dejaba ingresar de forma correcta a la pantalla donde podrá realizar todo el CRUD en la base de datos. Para comprobar el tipo de usuario, se utilizó Firebase® con su servicio de autenticación, en el que se pueden usar métodos de autenticación mediante contraseñas, números de teléfono, Google, Facebook, Twitter, entre otros.

Entre los métodos mencionados anteriormente, se utilizó la autenticación por contraseña, que provee una función que permite determinar los datos del usuario activo, como su correo y contraseña, además, información adicional que se haya determinado en el registro, con esto se realizó una condición en la que solo redirigía a la siguiente pantalla al usuario empresa (ver anexo G).

Inicialmente, se obtuvo la instancia de autenticación, la base de datos de Firebase y se estableció el Id del usuario activo, para de esta forma poder obtener los datos asociados a ese identificador. De los datos disponibles, se obtuvo el tipo de usuario y se realizó una condición, si el usuario es de tipo empresa, era redirigido a un nuevo fragmento llamado “FormularioFragment”, donde se encontraba un mapa sobre el que se podía clicar para obtener la posición y posteriormente, realizar las operaciones de gestión de la base de datos. En caso de ser usuario normal, se mostraba el mensaje “Debe tener permiso de administrador” y no redirigía al usuario. El resultado es el siguiente:

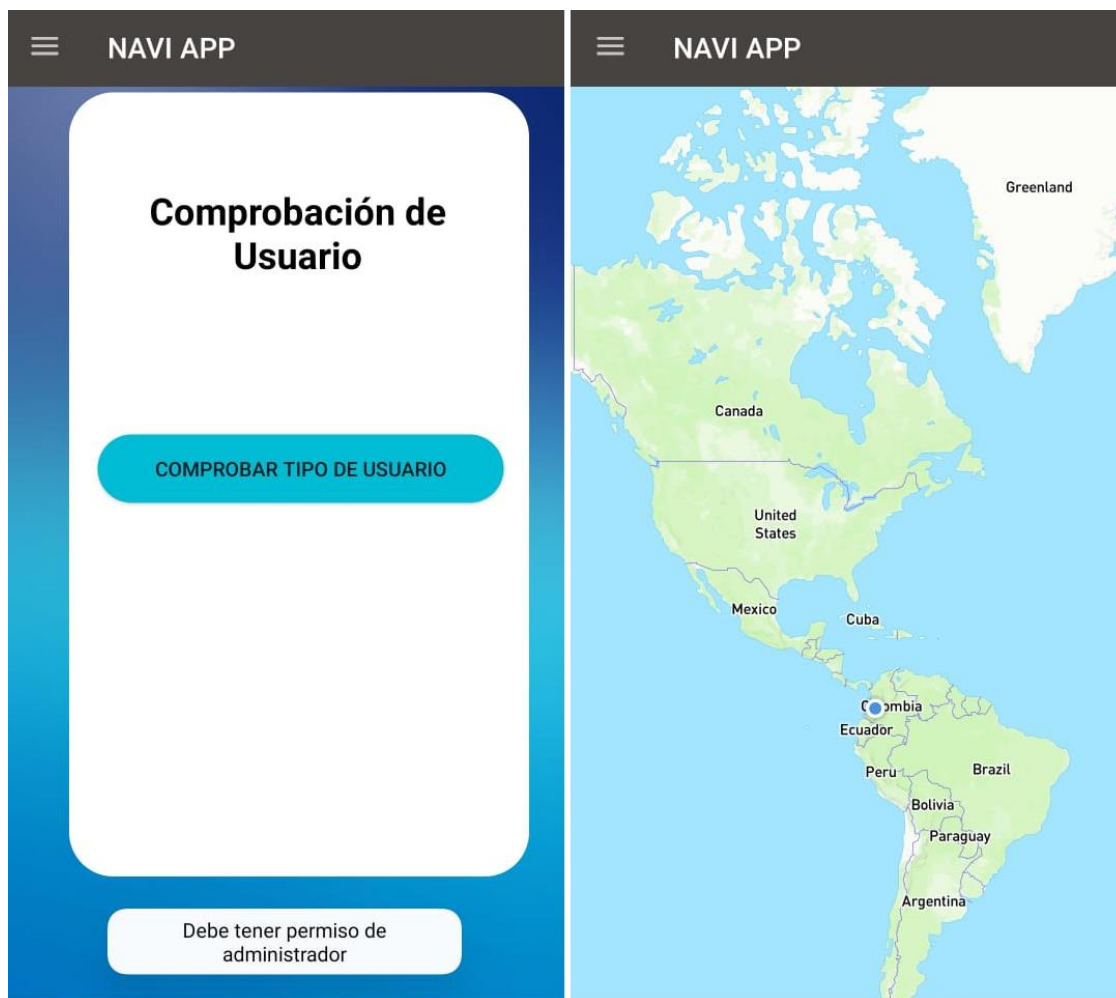


Figura 24. Interfaz de comprobación de usuario, interfaz para agregar o modificar datos del mapa.

Fuente. Elaboración propia.

En la ventana del mapa, al presionar sobre alguna zona se captura las coordenadas de esa ubicación, para posteriormente, ser utilizadas sobre los formularios de inserción de datos. Las coordenadas fueron obtenidas utilizando el método OnMapClick, que consiste en un evento perteneciente al SDK de Mapbox®, este permite realizar operaciones al momento de dar clic sobre el mapa. Sobre este evento hay una gran cantidad de funciones que permiten la interactividad en la aplicación, entre las que se encuentra Point, que captura la latitud y longitud de la ubicación clicada, de esta forma podían ser enviadas a una nueva pantalla en la que se van a realizar

operaciones de inserción, utilizando esos datos de coordenadas. Estas coordenadas fueron enviadas a la nueva pantalla por medio de una clave para cada una, permitiendo ser llamados con esta clave en la actividad. Al redirigir al usuario inicialmente le pedía que tipo de operación desea realizar (agregar, actualizar o eliminar) y dependiendo de ello, lo enviaba a una ventana u otra.

Al estar en la ventana escogida, puede realizar el cambio en la base de datos del tipo de elemento seleccionado (actividad, hotel o restaurante), generando un formulario para cada uno, en el que al llenar los datos y volver al mapa, podrá visualizar los cambios realizados. La interacción del usuario con el módulo se muestra en la siguiente imagen:

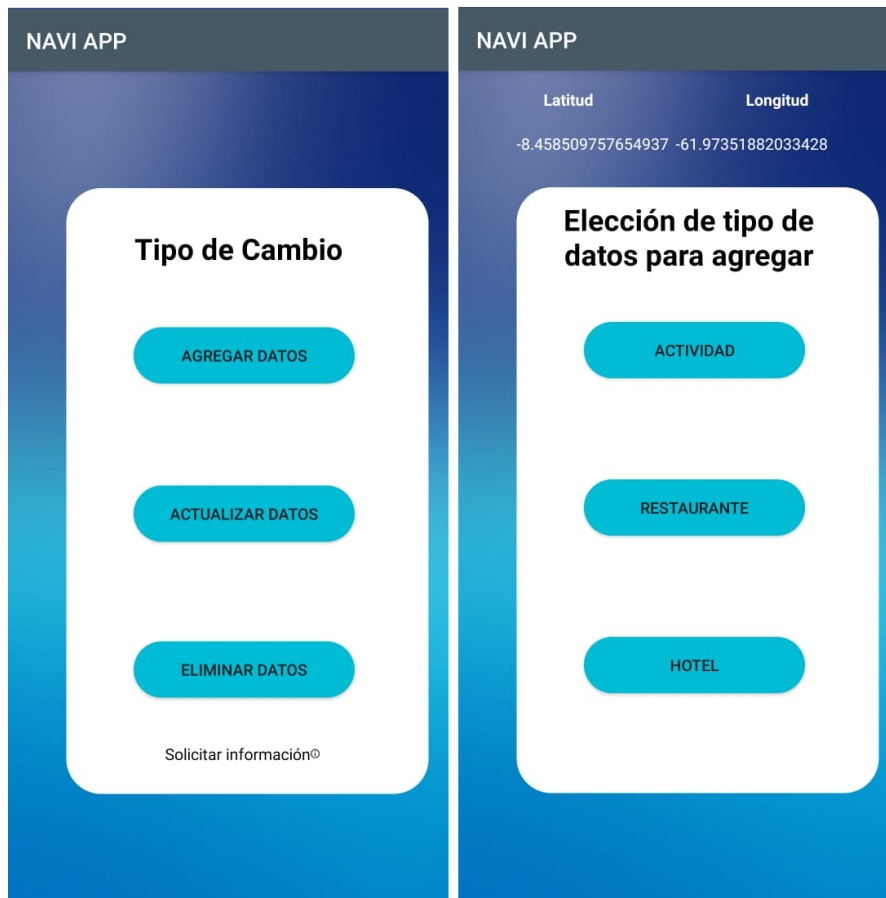


Figura 25. Interfaz del tipo de cambio, interfaz para la elección de tipo de datos para agregar.

Fuente. Elaboración propia.

Al estar sobre algún botón de inserción se pidieron los datos de la ubicación que se iba a adicionar y se incluyeron la latitud y longitud capturadas anteriormente. Los datos solicitados en caso de ser una actividad o un hotel eran el nombre, horario, precio y descripción. En caso de ser un restaurante, solo era necesario suministrar el nombre de este. Con esta información, se generó un hashmap con los valores obtenidos y la clave con la que iba a ser reconocido al enviarlo a la API y se convirtió en un objeto JSON, que es redirigido a una url donde se procesaron estos datos para ser enviados a la base de datos y ser insertados.

Esta información fue enviada a la API por medio de una petición POST con el objeto JSON, que se encargó de insertarla a la base de datos y posteriormente, el cambio se viera reflejado en el mapa. Se utilizó una función que

determinó si los campos fueron llenados, en caso contrario, envía un mensaje diciendo que se debían llenar todos los campos. Al ser llenados, la aplicación los convierte en String y eran ingresados a un Hashmap con una clave para cada dato. Esta clave debe coincidir con la que se encuentra en la API. El código utilizado para enviar los datos a la API se muestra en el anexo H. El Hashmap fue convertido a formato JSON y enviado a la API, que procesó esos datos y los envió a la base de datos (ver anexo I).

Inicialmente, se llamó al archivo SQLGlobal2, que contiene el acceso a la base de datos y un conjunto de plantillas para la generación de consultas. A partir de esto, se determinó el método para el envío a la base de datos. Una vez determinado el método, se obtuvieron los datos provenientes del objeto JSON y se creó la consulta de inserción con esta información. Estos fueron agregados a un array y a partir de este fueron insertados a la base de datos. En caso de que fueran agregados correctamente, la aplicación muestra un mensaje diciendo que “la operación se realizó de forma satisfactoria” y enviaba al usuario a la ventana del mapa. En caso contrario, muestra un mensaje diciendo que “hubo un error en la inserción”. Si los datos fueron ingresados correctamente, al entrar al mapa, en el fragmento del tipo de ubicación agregada, se mostrará un marcador en las coordenadas seleccionadas y con los datos ingresados. El resultado al hacer lo anterior es el siguiente:

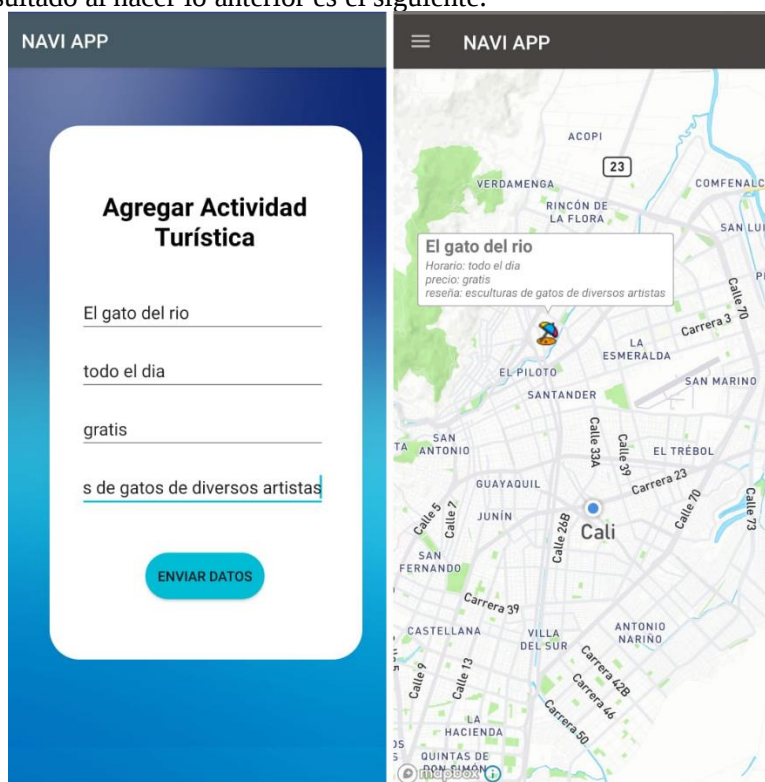


Figura 26. Interfaz de la ventana para agregar una nueva actividad turística. Interfaz del mapa con el marcador nuevo.

Fuente. Elaboración propia.

Al volver a la pantalla en que se muestran los botones de inserción, actualización o eliminación, y seleccionando la ventana de actualización se mostraba un formulario de acuerdo con el tipo de ubicación seleccionada para actualizar. En caso de ser una actividad o un hotel era necesario conocer el nombre anterior y el identificador único que se encuentra en la base de datos, permitiendo tener un mayor control sobre la seguridad de los datos al momento de actualizar o eliminar, evitando que cualquier persona pueda realizar cambios sobre la base de datos. Adicionalmente, en el formulario fue necesario suministrar la información nueva que se iba a actualizar, como lo son el nuevo nombre, horario, precio y descripción. En caso de encontrarse en el formulario de restaurante, solo era necesario suministrar el nombre antiguo, identificador y nombre nuevo.

De forma similar al método de inserción, que se muestra en el anexo H, se realizó la captura de los datos suministrados en el formulario, con los que se generó un hashmap, que fue convertido a un objeto JSON, y por medio de la URL suministrada, se enviaron esos datos a una API por el método POST, en la que se procesaron y analizaron los datos, verificando si se encontraba una ubicación del tipo seleccionado y habían coincidencias con el nombre e identificador proporcionados en el formulario. En tal caso, se realizaba la actualización con los nuevos datos suministrados y se enviaba un mensaje al usuario diciendo que los datos fueron actualizados correctamente y se redirigía al usuario a la pantalla del mapa, donde era posible observar los cambios realizados sobre la actividad, hotel o restaurante.

En la API, similarmente a la imagen que se encuentra en el anexo I, Se hizo un llamado al archivo SQLGlobal2 que contiene el acceso a la base de datos y las diferentes funciones de consulta, en los que están la inserción actualización y eliminación. A continuación, se determinó el tipo de método de envío de datos. En este caso se utilizó POST y se obtuvo los datos a partir de sus llaves. Con ello, se realizó la operación de actualización, se guardó en un array y se envió una respuesta de acuerdo con el resultado que arrojará el servidor.

Finalmente, al actualizar la información, el usuario fue redirigido a la pantalla del mapa donde podía ver los cambios sobre el marcador de la actividad, hotel o restaurante, como se muestra a continuación:

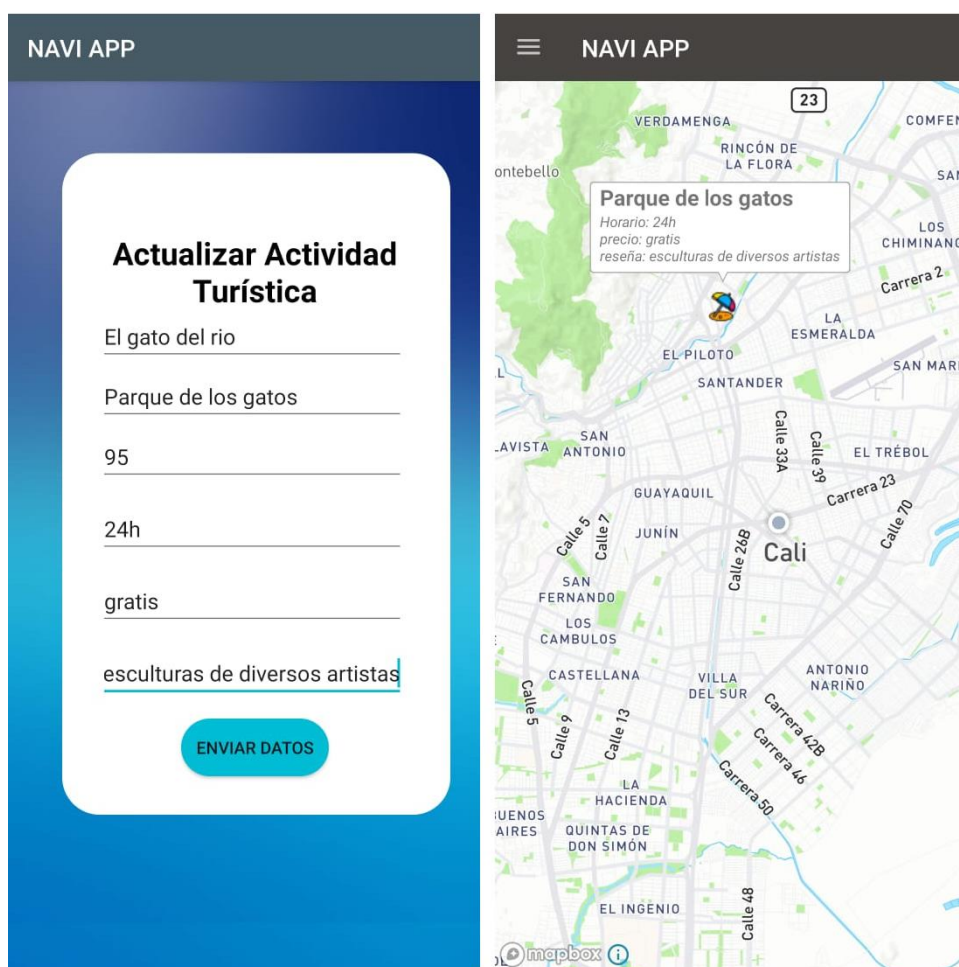


Figura 27. Interfaz de la ventana para actualizar una actividad turística. Interfaz para la visualización del dato nuevo.

Fuente. Elaboración propia.

Si el usuario se devuelve a la pantalla en que se muestran los botones de inserción, actualización y eliminación, selecciona la opción de eliminar en alguna de las opciones, entre actividades, hoteles y restaurantes, podía visualizar nuevamente un formulario que debería ser completado con la información de la ubicación que será eliminada. Para ello, se le pidió al usuario llenar los datos del nombre e identificador. Esta información sobre el formulario es la misma para todos los tipos de ubicación.

Para la realización de los métodos de eliminación se llevó a cabo un proceso similar al de la imagen del anexo H, realizando la captura de los datos del nombre e identificador suministrados en el formulario, con estos datos fue creado un hashmap, que contiene los valores del formulario y sus respectivas claves. Este hashmap fue convertido a un objeto JSON y enviado a una URL por el método POST, donde fue procesado por una API, en la que se realizó la consulta de eliminación a la base de datos, una vez que el sistema verificara que hay una coincidencia entre los datos suministrados y los contenidos en la BD.

El funcionamiento de la API era similar al de la imagen que se encuentra en el anexo I, que llamó a SQLGlobal2 para acceder a la base de datos y por el método POST fue recibido el objeto JSON, por medio de sus claves se determinaron el nombre y el gid, a partir de esto, se creó la consulta de eliminación, que fue guardada en un array. este array fue enviado al servidor y de acuerdo con la respuesta mostrada por este, se mostraba un mensaje al usuario en la aplicación, y en caso de ser realizado correctamente, el usuario era redirigido al mapa, donde podía visualizar los cambios, de acuerdo con el tipo de ubicación que hubiera eliminado, como se muestra a continuación:

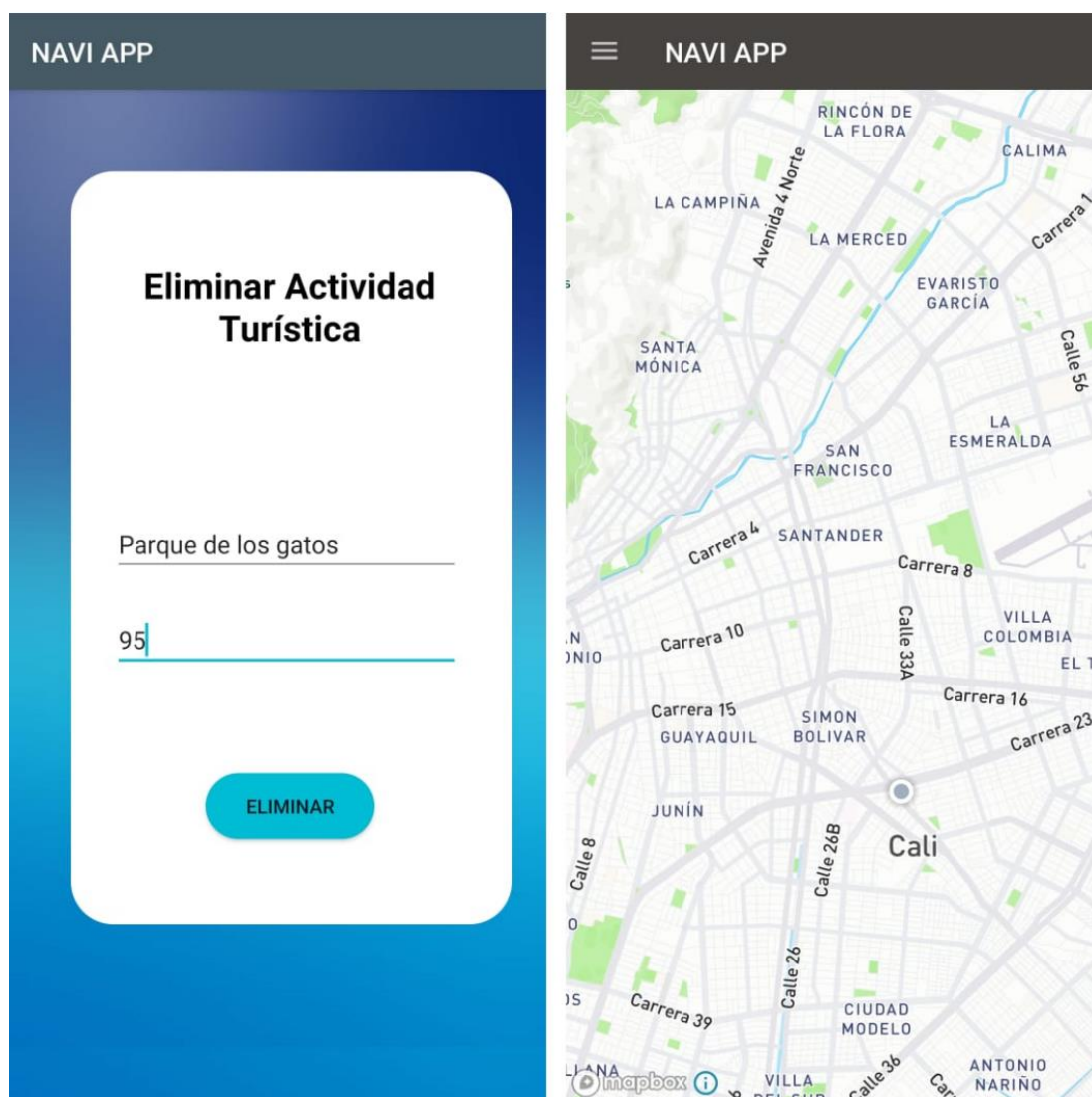


Figura 28. Interfaz de la ventana para eliminar una actividad turística. Interfaz del mapa demostrando que se eliminó una actividad.

Fuente. Elaboración propia.

Adicionalmente, si el usuario no tiene toda la información para llenar el formulario, al estar en la pantalla de selección del tipo de cambio, podrá presionar sobre el texto de solicitar información. En esta ventana puede visualizar un texto en el que se explica la información que se debe llenar para solicitar datos y los correos a los que puede solicitar esta información.

Al dar clic sobre alguno de estos correos, el usuario será redirigido hacia el correo de Gmail, donde se encuentra ya llenado el correo del destinatario y el asunto con el que iba a ser identificado el correo, de esta forma la empresa podrá solicitar la información necesaria, y el super administrador comprobará la información recibida con el fin de decidir si enviar los datos solicitados o no. El resultado se muestra en la siguiente imagen:

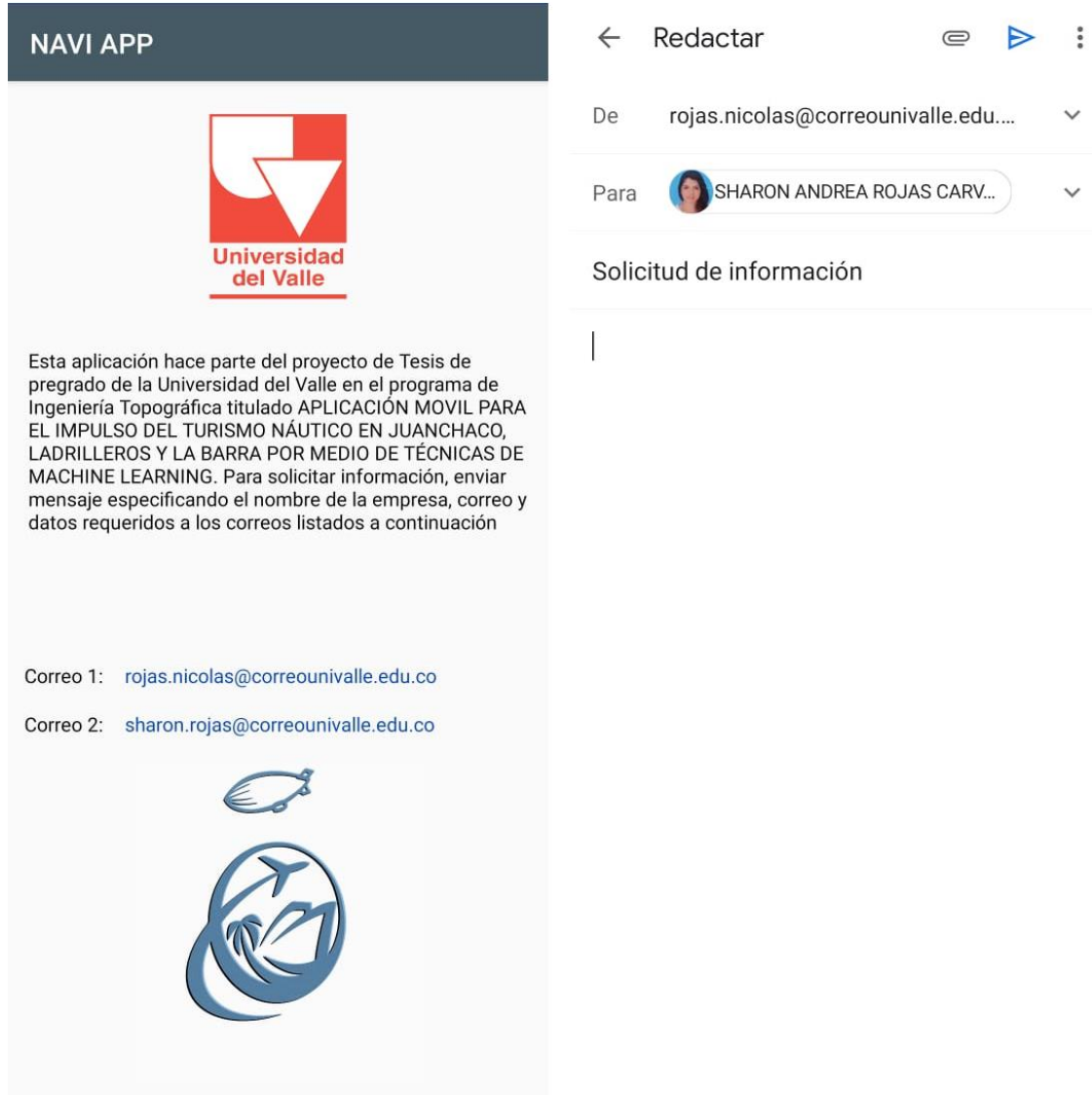


Figura 29. Interfaz de la ventana de información. Ventana de Gmail para enviar correo solicitando información.

Fuente. Elaboración propia.

CAPITULO 6. GENERAR UN SISTEMA DE RECOMENDACIONES A PARTIR DE LAS PREFERENCIAS DEL USUARIO, EMPLEANDO TÉCNICAS DE MACHINE LEARNING

6.1 Módulo del Sistema de Recomendaciones.

6.1.1 Análisis

Los sistemas de recomendación tienen como objetivo ayudar a un usuario a filtrar información sobre algo, ya sea música, películas, etc., con el fin de que al usuario le sea mucho más fácil encontrar información relevante y adecuada a sus gustos. Debido a lo anterior el sistema de recomendaciones intenta identificar coincidencias entre las características de los usuarios y los ítems de información. Este proceso lo realiza mediante un filtrado de contenido el cual tiene diferentes formas de hacerlo, entre los sistemas de recomendación que existen están los siguientes:

Filtrado basado en Demografía: Este sistema de recomendación que se basa principalmente en la demografía realiza una identificación del tipo de usuarios a los que les gusta un objeto y crea estereotipos. Este tipo de sistema permite que se hagan recomendación aun sin que el usuario haya realizado algún tipo de consulta o haya escogido un ítem específico. Sin embargo, entre las desventajas que tiene es que las recomendaciones no son muy personalizadas, debido a que no todas las mujeres de cierta edad les gusta un mismo género de películas, por lo que el sistema generaliza sin tener en cuenta que hay diferentes opiniones sobre las cosas.

Filtrado basado en Contenido: Este sistema de recomendación se basa en el contenido, es decir, crea el supuesto que todos los ítems que tienen características similares deben ser valorados de manera similar, por lo que depende de las descripciones que tiene cada ítem. Una ventaja de este sistema es que sin importar cual es el ítem, realiza una recomendación comparándolo con otros ítem. Una desventaja es que, si no tiene características similares con ningún otro ítem, jamás podrá ser recomendado, por lo que si cada ítem tiene características diferentes no funcionará adecuadamente, puesto que no dará ninguna recomendación.

Filtrado Colaborativo: Este sistema es uno de los más usado por las aplicaciones que tienen este tipo de filtrado. Este sistema se basa en ítems que a otros usuarios le ha gustado en el pasado y con esto el sistema recomienda eso al usuario. Para realizar esto, el sistema utiliza el historial de calificaciones de que han hecho otros usuarios en el pasado. Entre las ventajas que tiene este sistema es que al igual que el de enfoque demográfico no necesita ningún conocimiento previo acerca de los productos. Una de las desventajas es el tamaño de los datos que influye en la calidad de las recomendaciones. Además, que si hay algún ítem sin calificación no es posible hacerle la recomendación.

Basado en conocimiento: Este sistema recomienda ítems que responden a la necesidad y preferencia del usuario, y en última instancia a la forma útil que es el ítem para el usuario. Este sistema recomienda los ítems basados en el conocimiento de dominio específico, acerca de cómo los ítems corresponden a las necesidades de los usuarios. Una ventaja de este método es no se basa en datos históricos por lo que puede generar recomendación solo indicando las preferencias del usuario, otra ventaja es que el sistema no tiene que almacenar cualquier información sobre el usuario en un largo plazo, o que hace que la aplicación sea más ligera. Entre las desventajas que tiene es que no aprende de sus preferencias, entonces si un usuario decide cambiar alguna preferencia le indicará probablemente otras recomendaciones.

Para crear el sistema de recomendaciones, que brinda unas opciones rápidas al usuario para escoger una de las actividades que se muestran en la aplicación de las playas de Juanchaco, Ladrilleros y La Barra, se utilizó el sistema basado en conocimiento, que utiliza datos del usuario real para entrenarse y poder modelar el sistema de recomendaciones. Para esto, se hace uso de variables significativas de los usuarios, para utilizarlas en algunos algoritmos de aprendizaje. Estas variables son centradas a un usuario turista definiendo las tareas que realiza el sistema de recomendaciones, están relacionadas con la fecha en que se realiza el viaje, los gastos que haría por actividad, el tipo de actividades que le gustan, si viaja en grupo, con la familia o solo, etc., logrando con estos datos entrenar una red neuronal, que posteriormente a un usuario del sistema le pueda recomendar el sitio que más se adecue a sus necesidades.

A continuación, se detalla cada una de las variables que se utilizaron para el desarrollo del sistema de recomendaciones:

- ❖ Género y rango de edad: Son datos personales del usuario para generar diversos conjuntos de datos, así el sistema podía crear varios escenarios. El género es femenino o masculino y para los rangos de edad estaban definidos como: Menor a 18, de 18 a 30, de 30 a 50 y Mayor de 50.
- ❖ Tipo de actividad: Son los tipos en que se caracterizaron las diferentes actividades que se mostrarían en la aplicación, es importante tener un conocimiento del tipo de actividades que son más atractivas para los usuarios y poder tener una posterior predicción para los usuarios. Los tres tipos de actividad que se escogieron para identificar las actividades que se tienen son: aventura, recreación y Deporte.
- ❖ Tipología: Se refiere a la característica del usuario para viajar, si viaja usualmente en grupo, con la familia o individual, definiendo conjuntos de datos para el entrenamiento de la red neuronal.
- ❖ Posible gasto: Es el precio por el cual el turista está dispuesto a pagar por cada actividad que quiere realizar, debido a que cada una tiene un rango de precios, se debe tener en cuenta si el usuario está dispuesto a pagarlos, por lo tanto, es necesario esta variable para darle la mejor recomendación al usuario.
- ❖ Mes: Hace referencia a la época del año en el que el turista se encontrará en alguna de estas playas, es importante debido a que algunas de las actividades que se ofrecen tienen su disposición en determinados meses del año, como lo es el avistamiento de ballenas que solo es posible en julio y en noviembre. Por lo tanto, es necesario que el usuario especifique este ítem.
- ❖ Actividades de preferencia: Este ítem se refiere a las diferentes actividades que les gusta realizar al usuario, por lo que puede entre determinadas alternativas, escoger la de su preferencia, las cuales pueden ser: Juegos acuáticos, Paseo en lancha, Buceo, surf, entre otros. Con esto la red neuronal pudo entrenarse para que según los datos anteriores del usuario ofreciera recomendaciones de estas actividades.

Una vez que se estableció las variables necesarias para que el sistema realice una recomendación aceptable que se ajuste a los usuarios, se procedió a la obtención de gran cantidad de estos datos para poder entrenar la red neuronal. Esta toma de información se realizó mediante un formulario creado en Google Forms, el formulario tiene respuestas de selección múltiple (Ver Anexo J), con el fin de obtener resultados similares para luego realizar el procesamiento de datos y el entrenamiento. Debido al estado actual de la pandemia, el formulario solo se pudo realizar por medio de redes sociales y correos, compartiendo el enlace del formulario. Se obtuvieron 202 resultados, no fueron la cantidad esperada, pero se continuó con el entrenamiento.

Para la realización de este módulo se generó 1 historia de usuario en la que se definieron los pasos que debía realizar el usuario para obtener las recomendaciones, estos son: haber iniciado sesión y llenar el formulario correspondiente. A partir de esto la aplicación mostraría un par de recomendaciones en las que, al dar clic sobre alguna, redirige al usuario a un mapa donde podrá ver las actividades de acuerdo con el tipo recomendado. La historia de usuario se muestra a continuación:

Tabla 12. Historia de usuario para el sistema de recomendaciones.

Historia de Usuario			
Autor	Nicolás Rojas Sharon Rojas	Módulo	Admin
Nombre HU	Agregar ubicación	Identificador	A02
Dependencia		Prioridad	
Actor(es)	Empresa	Iteración	
Puntos asignados			
Descripción			
Como usuario o empresa deseo poder visualizar las actividades náuticas más probables a visitar de acuerdo con mi perfil de usuario, obteniendo el nombre del lugar y una imagen asociada a la actividad.			
Observaciones			
1. Para visualizar las actividades recomendadas el usuario debe tener una cuenta asociada y estar logueado. 2. Para que funcione correctamente el sistema de recomendaciones, el usuario debe llenar un formulario donde se le preguntarán las características del viaje que va a realizar 3. Al presionar sobre el botón de sitios recomendados, la aplicación enviará al usuario a un nuevo fragment donde se mostrarán 2 actividades recomendadas con su tipo y una foto y al dar click sobre alguna, enviará al usuario a un mapa donde mostrará a detalle cada actividad de este tipo y mostrará su información como nombre precio, horario y descripción.			
Bocetos	Mapa	Recomendados	
Creada	26/11/2018	Modificada	3/02/2021

Fuente. Elaboración propia.

Debido al tipo de datos, al escoger un clasificador, y al tener en el entrenamiento un vector de respuestas, en el que un usuario puede escoger más de una, el sistema tenía un espacio donde las características eran las mismas y el resultado diferente, por lo tanto, no había forma de separar los datos y clasificarlos como un tipo u otro. Por lo tanto, el utilizar una regresión y al no separar por clases, se obtenía una probabilidad de ocurrencia, y con esto, la posibilidad de que a un usuario le guste un tipo de actividad. De esta forma, se obtuvieron de mejor forma las actividades que le pueden gustar a un usuario, mejorando la precisión y sin necesidad de dividir o disminuir la información.

6.1.2 Implementación

Para el desarrollo de esta herramienta, se utilizó el lenguaje de programación Python, con las librerías de inteligencia artificial proporcionadas por Sci-kit learn. Estas son de gran utilidad, ya que permiten realizar clasificaciones, regresiones, redes neuronales, entre otros, con la característica de poder realizar regresiones multi salida. De esta forma, se pudo obtener las predicciones con mejor probabilidad de ocurrencia y obtener las 2 mejores recomendaciones.

Para el manejo de los datos se usó la librería panda®, que presenta una cantidad de funciones para manipular datos de tablas, archivos JSON, etc. De esta forma se convirtieron los datos provenientes del formulario de Google, que contenía datos nominales en forma de cadenas de texto, a numéricos, facilitando su manejo y manipulación.

Con los datos organizados por clases, se utilizó la función “train_test_split”, que separó los datos en entrenamiento y test, dejando el 70% para entrenamiento, definiendo las características como datos de entrada y las actividades como salidas. Una vez separados los datos, se utilizó la medida de preprocesamiento de estandarización, con el fin, de tener los datos más juntos y de esta forma, tener unos resultados más precisos con el regresor. Esta técnica de estandarización consiste en transformar los datos de tal forma que tengan una media 0 y desviación estándar 1.

Posteriormente, con los datos organizados, se procedió a realizar el entrenamiento del regresor. Para ello, se probaron los 3 modelos más usados: regresión lineal, regresión logística y bosques aleatorios. Todos ellos en sus configuraciones básicas, utilizando los datos de entrenamiento y realizando las predicciones hacia los datos de prueba, para determinar la precisión del modelo. El código utilizado se muestra a continuación:

```
from sklearn.multioutput import MultiOutputRegressor
from sklearn.ensemble import RandomForestRegressor
clf = MultiOutputRegressor(RandomForestRegressor(n_estimators=100, random_state=0)).fit(X_trainscaled, y_train)
arr=clf.predict(X_test)
arr
```

Figura 30. Código para determinar la precisión del modelo.

Fuente. Elaboración propia.

Inicialmente, se llamaron las librerías de regresión multi salida y regresión Random Forest. En caso de ser lineal o logística se utilizaban las librerías correspondientes. Luego, se realizó el regresor ingresando la función “MultiOutputRegressor” y poniendo como argumentos la regresión que se iba a realizar y se ajustaron a los datos de entrenamiento, de esta forma, realizar la predicción sobre los datos de prueba y posteriormente, calcular la precisión del modelo y determinar si se presentaban buenos resultados. Para determinar la precisión se utilizaron 3 métricas proporcionadas por sklearn: Error medio absoluto (MAE), Error Medio Cuadrático (MSE) y Error Medio Cuadrático Logarítmico (MSLE).

La primera métrica, determina el error entre las estimaciones y los resultados reales, calculando la media entre los valores absolutos de los errores. Para que funcione correctamente, los datos debían tener la misma escala. La segunda, calcula la raíz de los errores al cuadrado, haciendo que el error sea siempre positivo, además, al realizar

la raíz de los cuadrados, es una medida ponderada que se centra en los valores atípicos, ya que las diferencias grandes al cuadrado se incrementan mucho más que las pequeñas (menores a 1), estas últimas, se vuelven incluso más pequeñas. Por último, el MSLE, que es una variación del MSE, determina el error entre las predicciones y los resultados reales, con la diferencia que trata los errores grandes y los pequeños con aproximadamente el mismo peso. Al aplicar estas métricas sobre los 3 modelos evaluados se obtuvo los siguientes resultados:

Error: Mean Absolute Error: 0.45402901896262676 Mean Squared Error: 0.326577267766198 Precision: MAE: 0.5459709810373732 MSE: 0.673422732233802	Error: Mean Absolute Error: 0.4568306010928962 Mean Squared Error: 0.4568306010928962 Mean Squared Log Error: 0.21948563914514552 Precision: MAE: 0.5431693989071038 MSE: 0.5431693989071038 MSLE: 0.7805143608548545	Error: Mean Absolute Error: 0.3984664845173042 Mean Squared Error: 0.2673349044019429 Mean Squared Log Error: 0.13004927969674732 Precision: MAE: 0.6015335154826957 MSE: 0.7326650955980571 MSLE: 0.8699507203032527
--	--	--

Figura 31. Resultados de las métricas en los tres modelos evaluados.

Fuente. Elaboración propia.

En comparación con los otros dos modelos, el de RandomForest presentó mejores resultados respecto a las precisiones, y, por lo tanto, fue el que se usó para realizar las recomendaciones de las actividades náuticas. Con el regresor creado, se generaron las predicciones utilizando datos de prueba. Esto dio como resultado un array con 15 datos, un dato por posible recomendación de actividad. Cada uno tenía un valor de probabilidad, donde los valores más altos eran las recomendaciones de actividades para los datos suministrados. Habiendo probado su funcionamiento, se guardó el modelo con extensión joblib.pkl y se guardó en un servidor de Github, para posteriormente acceder a él por medio de una URL.

Para la construcción de la API, se importaron las diferentes librerías necesarias para su desarrollo, estas son Flask, numpy, pandas, joblib y json. Posteriormente, utilizando Flask, se definió la ubicación donde iba a estar corriendo la aplicación, y sobre esa ruta definida se realizó todo el proceso de desarrollo. A partir del archivo joblib.pkl, se llamó al archivo para luego realizar predicciones a partir de él.

Utilizando la librería json, se obtuvo los datos que va a recibir la API, definiendo como método de entrada POST. Con los datos obtenidos, se realizó una conversión del objeto JSON resultante a un numpy array, ya que este es el tipo de datos que recibe el modelo para realizar las predicciones. Este objeto se metió en el modelo, obteniendo 15 probabilidades de ocurrencia, donde las que tuvieran un mayor valor, serían las actividades más recomendables para realizar con las características definidas.

Para que el sistema mostrara las actividades recomendadas y no solo unas probabilidades de ocurrencia que a vista del usuario no representarían nada, se ingresó un índice para cada valor, obteniendo de esta forma la actividad correspondiente a cada probabilidad, y para mostrar simplemente las 2 mejores recomendaciones, se obtuvo el índice de las actividades con mayor probabilidad de ocurrencia, mostrando en el sistema un array con estas 2 actividades.

Con el proceso anterior, se tuvo la API desarrollada, sin embargo, su funcionamiento es solo de forma local, por lo tanto, era necesario subir esta aplicación a un servidor sobre la que pudiera correr y que se pueda acceder desde cualquier dispositivo y en todo momento. Para esto, se creó un entorno virtual, que es sobre el que debía funcionar la aplicación al momento de estar en el servidor. Utilizando GIT, se subió sobre Heroku, obteniendo de esta manera una URL pública que corre la API en todo momento para realizar peticiones sobre ella. El resultado se muestra a continuación:

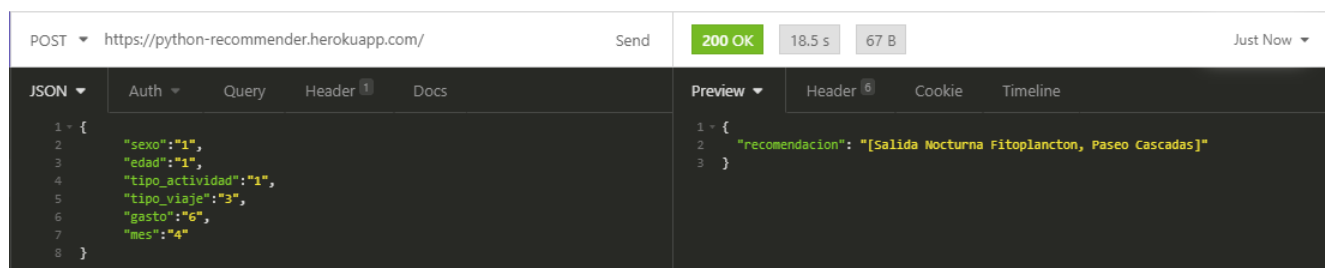


Figura 32. API donde se ejecuta el sistema de recomendación.

Fuente. Elaboración propia.

Al enviar por POST el objeto JSON, la API lo recibe y corre el sistema de recomendaciones, obteniendo un vector con 2 posibles actividades: salida nocturna Fitoplancton y Paseo cascadas; estas son las que contenían los mayores valores de probabilidad con los datos ingresados.

Con la API corriendo de forma correcta, utilizando Android Studio, se creó un formulario, en el que se realizó cada pregunta, utilizando Spinners, para que el usuario escoja entre un valor y otro. Al escoger un valor en cada uno, la aplicación obtenía el identificador de cada uno, pasando estos datos a numéricos, de acuerdo con el valor seleccionado en el spinner. Los datos obtenidos, fueron pasados a un hashmap, donde se definía la clave correspondiente a cada dato.

Posterior a la creación del hashmap, se creó un objeto JSON con los datos y sus llaves correspondientes. Este objeto fue pasado a la API, por medio de la URL donde se encuentra el sistema de recomendaciones, de esta forma, se obtuvo en la aplicación un vector con las actividades recomendadas de acuerdo con los datos que suministró el usuario. Este vector fue separado en 2 variables, que contenía en cada una, la recomendación obtenida, mostrando al usuario por separado cada recomendación en forma de texto.

Para mejorar la experiencia de usuario, se cargaron 15 imágenes a la aplicación, una por cada tipo de actividad, haciendo una condición en la que, al cargar cada texto, muestra una imagen de acuerdo con el valor de la recomendación, de esta forma, en vez de mostrar solo un texto con la recomendación, se pudiera ver una imagen que muestre la actividad recomendada.

Adicionalmente, se realizó un evento en el que, al dar clic sobre alguna de las actividades, la aplicación redirige al usuario a un mapa en el que dependiendo de la actividad clicada, mostrará las que tienen que ver con ese tipo. Para esto se realizó una condición en la que de acuerdo con la actividad clicada fuera a un enlace u otro y consumiera un determinado archivo GeoJSON, mostrando solo los marcadores del tipo de actividad seleccionada. Finalmente, el resultado del módulo se muestra a continuación:

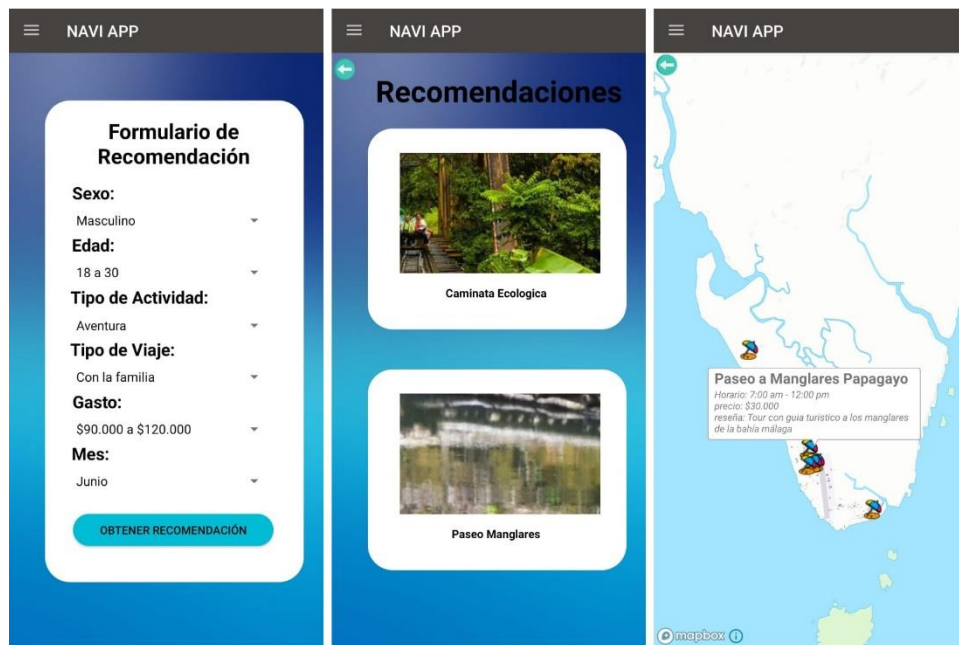


Figura 33. Interfaz del formulario para la recomendación. Interfaz de las recomendaciones que dio el sistema. Interfaz del mapa donde se visualizan las actividades de una de las recomendaciones seleccionada.

Fuente. Elaboración propia.

PRUEBAS DE USABILIDAD

Una vez terminada la aplicación, se realizó una prueba de usabilidad con el fin de determinar cómo es percibida la aplicación por los usuarios, para ello, se realizaron un conjunto de encuestas a 30 usuarios con el fin de cuantificar aspectos como la capacidad de realizar consultas, información suministrada, navegación entre pantallas, recomendaciones y consistencia en el diseño.

Para realizar esta prueba, fue necesario contar con la aplicación instalada y fake GPS, esta última con el fin de simular la ubicación en alguna de estas playas y poder ubicar al usuario como si se encontrara en el lugar, para que se pueda utilizar la generación de la ruta de forma correcta y ver la información de las actividades, restaurantes y hoteles sin necesidad de mover el mapa hasta la zona de estudio. La cantidad de usuarios encuestados y el simulador del GPS en el dispositivo móvil, se debió a la situación actual que presenta el mundo entero por la pandemia.

Una vez en la aplicación los usuarios realizaron pruebas situándose en cada módulo, y a partir de ello, realizaron una encuesta con las que podían realizar una calificación de los aspectos de consultas, navegación entre pantallas, entre otros, obteniendo las estadísticas que se muestran a continuación. Es de gran importancia, como se mencionó en el capítulo 4, que los usuarios que probaran la aplicación tuvieran un sistema operativo 4.2 (Jelly Bean) y un tamaño de pantalla de 5.5 pulgadas. En caso contrario, la aplicación sería incompatible con el dispositivo y en el caso de la pantalla no podrían visualizar correctamente los elementos, encontrando errores en la distribución de botones y textos.

Entre las preguntas que les realizó a los usuarios para las pruebas se encuentra, que el 63.3% de los encuestados tenían menos de 25 años, esta población son los que tienen mayor manejo en la interacción de aplicaciones, por lo que tienen mayor criterio para definir la calidad de un de estas. (Ver figura 34)

Edad

30 respuestas

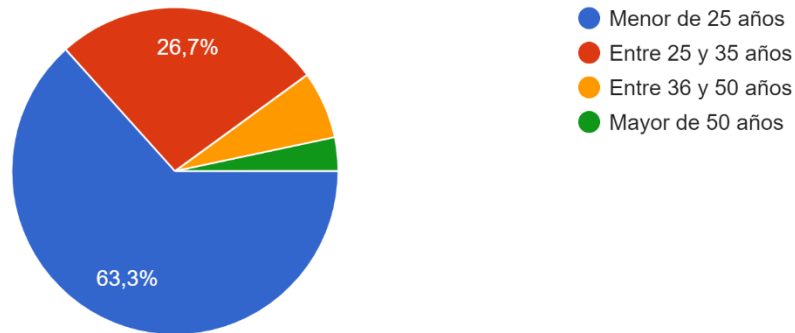


Figura 34. Edad de los usuarios encuestados de la prueba de usabilidad.

Fuente. Elaboración propia.

Se evaluó con qué frecuencia los encuestados realizan viajes de turismo, esto con el fin de saber que tanta experiencia han tenido con este tema y que puedan apreciar de mejor forma la aplicación y ser más consientes para la evaluación. (Ver figura 35)

¿Cuántas veces al año realiza viajes de turismo?

30 respuestas

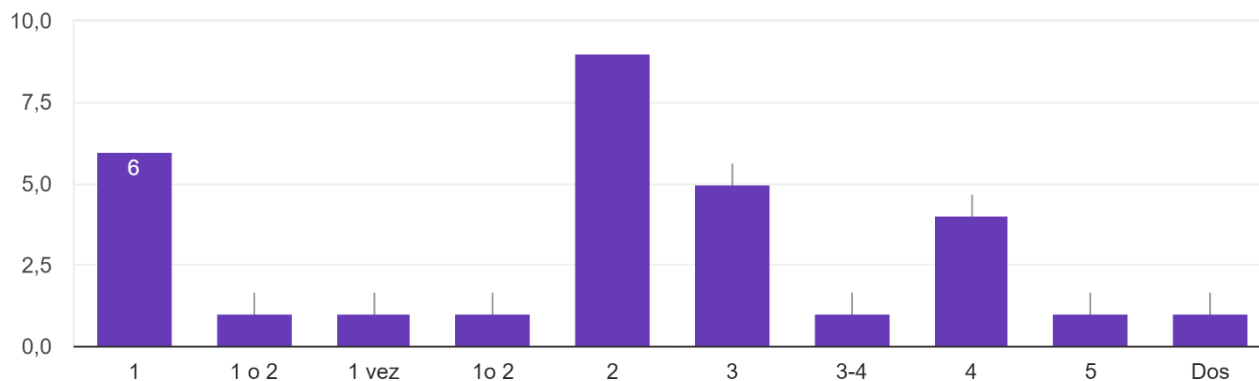


Figura 35. Veces en el año que el usuario realiza viajes de turismo.

Fuente. Elaboración propia.

Más del 90% de los encuestados usan el celular para orientarse durante un viaje de turismo como lo muestra la figura 36

¿Ha utilizado tecnología móvil para orientarse en el transcurso de su viaje de turismo?

30 respuestas

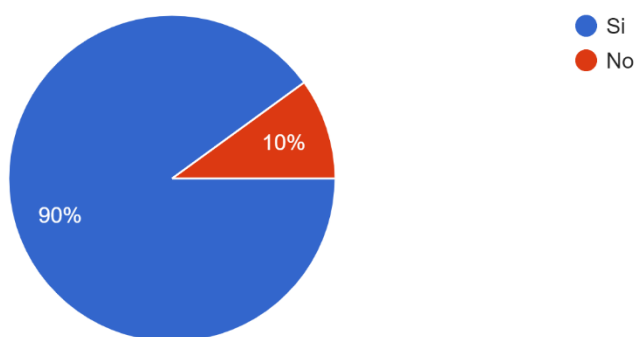


Figura 36. Pregunta sobre la utilización de un dispositivo móvil para orientarse en el transcurso del viaje.

Fuente. Elaboración propia.

El 56.7% de los encuestados pudo realizar fácilmente las diferentes consultas que ofrece la aplicación y un 16.7% muy fácilmente. (figura 37)

¿Pudo realizar las consultas que ofrece la aplicación de los diferentes lugares?

30 respuestas

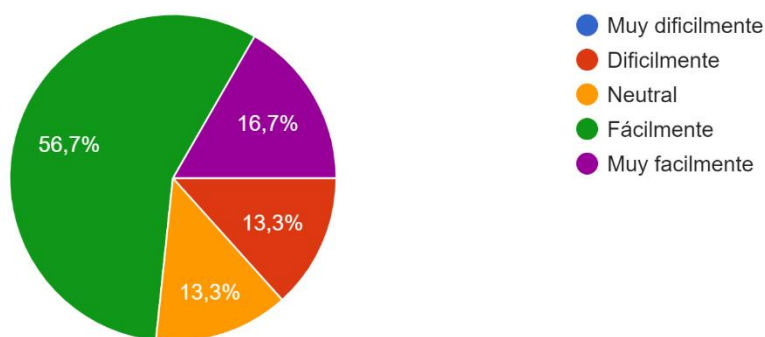


Figura 37. Pregunta de si el usuario pudo realizar las consultas que ofrece la aplicación de los diferentes lugares.

Fuente. Elaboración propia.

El 53.3% está de acuerdo de que la información suministrada es completa, el 10% está completamente de acuerdo y el 30% esta neutral con respecto a la pregunta cómo se puede visualizar en la figura 38.

¿Considera la información suministrada, a través de la aplicación, completa?

30 respuestas

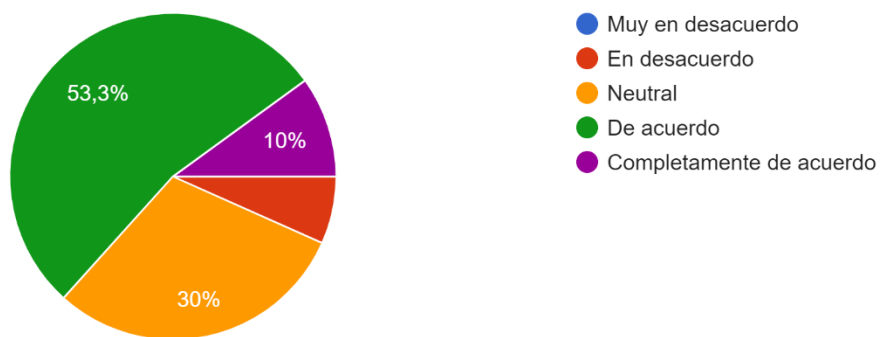


Figura 38. Pregunta de si el usuario considera la información suministrada, a través de la aplicación, completa.

Fuente. Elaboración propia.

Con respecto a la experiencia que tuvieron durante la prueba entre las pantallas, a el 50% de los usuarios les pareció fácil la navegación a través de la aplicación y un 16.7% muy fácil. (Ver figura 39)

¿Le pareció fácil la navegación (desplazamientos entre pantallas) a través de la aplicación?

30 respuestas

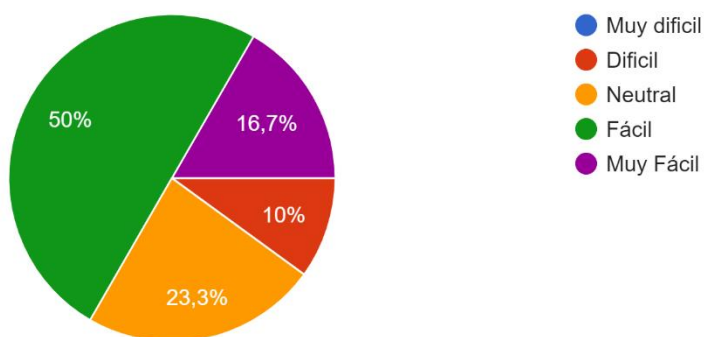


Figura 39. Pregunta de si al usuario le pareció fácil la navegación (desplazamientos entre pantallas) a través de la aplicación.

Fuente. Elaboración propia.

Con respecto a las recomendaciones que se ofrecen en la aplicación por medio de las preferencias del usuario, el 36.7% de los encuestados se encuentra de acuerdo con las recomendaciones dadas por el sistema y el 26.7% completamente de acuerdo. (Ver figura 40)

¿Pudo obtener buenas recomendaciones de actividades turísticas?

30 respuestas

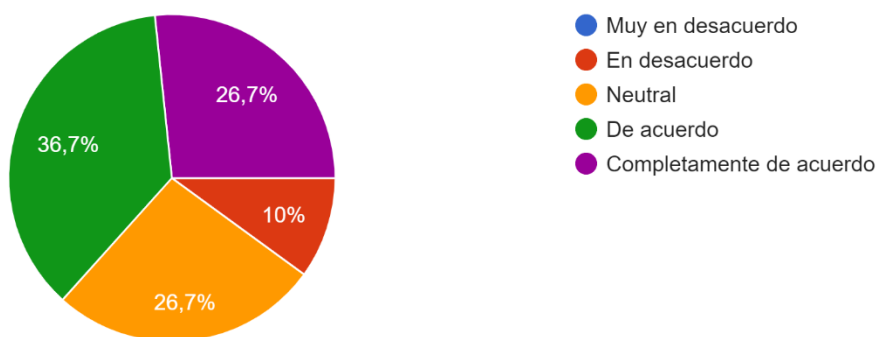


Figura 40. Pregunta de si al usuario pudo obtener buenas recomendaciones de las actividades turísticas.

Fuente. Elaboración propia.

Al 46.7% de las personas encuestadas están de acuerdo con el diseño de la aplicación y un 20% completamente de acuerdo. (figura 41)

¿Considera que el diseño de la aplicación es consistente?

30 respuestas

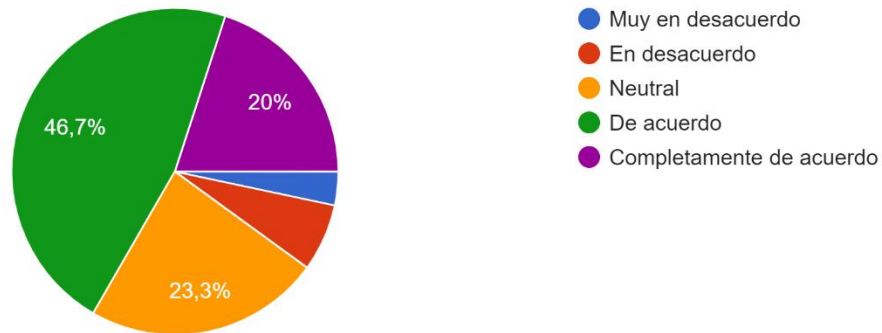


Figura 41. Pregunta de si el usuario considera que el diseño de la aplicación es consistente.

Fuente. Elaboración propia.

Al 53.3% les pareció satisfactorio la usabilidad de la aplicación, les pareció fácil la interacción con el visor y muchas de las funciones que tiene la aplicación eran muy intuitivas. (Ver figura 42)

¿Cuál es su grado e satisfacción en el uso de la aplicación?

30 respuestas

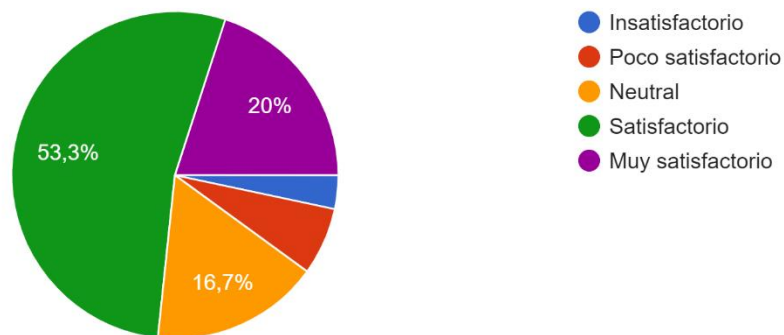


Figura 42. Pregunta de si la aplicación fue de su agrado y satisfacción.

Fuente. Elaboración propia.

Entre los comentarios que se destacaron positivamente, están los siguientes:

- Buena interfaz gráfica
- Muy útil para esos lugares de Colombia que son poco conocidos
- Brinda información útil

Entre los comentarios para mejora se encuentran:

- Información de ruta desde diferentes partes del país
- Mejor interacción con las burbujas de información
- Búsqueda de actividades por nombre o tipo

A partir de algunas sugerencias por los usuarios con respecto a la usabilidad de la aplicación, se realizó las respectivas revisiones y se arreglaron muchos de los errores que presentaba la aplicación, como lo es el botón hacia atrás, el cual fallaba y sacaba al usuario de la aplicación, que se corrigió y se implementó un botón en la mayoría de las ventanas para devolverse a la ventana anterior. Sin embargo, algunas recomendaciones fueron compiladas e incluidas en la sección de trabajos futuros, y de esta forma determinar las mejoras que se deben realizar a la aplicación y mantenerla en constante desarrollo.

DISCUSIÓN DE RESULTADOS

Uno de los principales resultados es la visualización que se obtuvo de las diferentes actividades que ofrecen estas playas, lo que permite que el turista tenga un conocimiento más detallado y confiable de las actividades que puede realizar en el momento de su visita al municipio. Contribuye al aumento del turismo en las playas, ya que es uno de los municipios que más ofrece cultura y recreación para este sector a nivel nacional y en la actualidad no tiene ese gran alcance que debería, por lo que esta aplicación, ofrece ese apoyo a las empresas y personas nativas para que el resto de las personas tanto nacional como internacionalmente conozca sus atributos y pueda disfrutar de forma eficaz y segura.

La tecnología ha avanzado en gran medida y en muchas de las ciudades actualmente contamos con aplicaciones que nos ofrecen mucha información sobre turismo, sin embargo, Navi_App se enfocó en una zona que no tiene hasta el momento ningún registro de alguna aplicación que cuente con la suficiente información para atraer a las personas a visitar este maravilloso lugar. Además, la información se consiguió a partir de trabajo de campo, puesto que las entidades del gobierno no brindaban una información así de detallada y no tienen mucho conocimiento de que establecimientos ofrecen actividades con sus precios y horarios.

A pesar de no encontrar una aplicación enfocada en el turismo que ofrecen las playas de Buenaventura, realizamos una comparación con un conjunto de aplicaciones enfocadas en otras zonas de estudio, en las que se relacionaron las funcionalidades, información mostrada, entre otros aspectos relevantes.

Inicialmente, se comparó con Cali Guía turística con mapa co ®, desarrollada por minube. Esta aplicación tiene una gran cantidad de información sobre sitios turísticos, pudiendo encontrar actividades, hoteles y restaurantes que se pueden visualizar sobre un mapa, con la ventaja de contar con imágenes que facilitan la visualización al usuario y la capacidad de realizar filtros de búsqueda; sin embargo, debido a que muestra información para muchos municipios, en muchos de estos es incompleta. Adicionalmente, Navi_App se diferencia de esta aplicación por la capacidad mostrar la ruta para llegar a un determinado lugar y el manejo de roles que permite a empresas realizar modificaciones a las actividades, hoteles y restaurantes pautados.

Por otro lado, se puede encontrar la aplicación Oís-secretos turísticos ®. Esta aplicación se centra en Cali, mostrando los diversos sitios turísticos de la ciudad, sobre los que se puede encontrar una reseña completa del lugar, su ubicación y mito urbano, permitiendo al turista conocer a fondo los atractivos que se encuentran en el municipio. Sin embargo, Navi_App se destaca respecto a ella por la capacidad de realizar recomendaciones personalizadas con información proporcionada por los mismos usuarios, realizar un CRUD completo de los sitios turísticos, manejo de sesiones de usuario y el poder obtener la ruta más corta hacia un determinado lugar sobre la misma aplicación.

Finalmente está ViajaxColombia ®, desarrollada por socolombia.com, que se enfoca en el turismo de todo el País. Tiene una gran cantidad de características como la capacidad de calcular los peajes de un municipio a otro, mostrar actividades turísticas, hoteles, entre otros que se encuentran en cada departamento y ver fotos y videos de ellos; además de poder ver noticias y fiestas proximas que se puedan encontrar en Colombia. A pesar de todo ello, la experiencia de usuario que brinda no es satisfactoria ya que para cada ubicación hay que ir por muchos botones y devolverse para volver a buscar una nueva ubicación; además, no cuenta con roles de usuario, ni la capacidad de agregar o editar ubicaciones, generar rutas en la misma aplicación, o mostrar recomendaciones personalizadas para los usuarios. Tampoco muestra el total de lugares sobre el mapa, haciendo que sea incomodo buscar cada sitio turístico potencial a visitar.

Para finalizar, se da respuesta a la pregunta que orientó este proyecto: ***¿cómo impulsar el turismo náutico en los corregimientos de Juanchaco, Ladrilleros y La Barra, pertenecientes a Buenaventura?*** Para esto se realizó el cumplimiento de pequeños objetivos que consistieron en la obtención de información y la realización de cada módulo de la aplicación, obteniendo así una herramienta centrada en actividades náuticas que permite mostrar el turismo de las playas de Juanchaco, Ladrilleros y La barra con herramientas que facilitan el dinamismo dentro de la aplicación, lo que permite cumplir el objetivo planteado.

CONCLUSIONES Y TRABAJO FUTURO

Conclusiones

Debido a la falta de información sobre los atractivos turísticos en las playas de Juanchaco, Ladrilleros y La barra, se diseñó e implementó una aplicación con el fin de promover el turismo en esta zona, haciendo énfasis en las actividades náuticas que se pueden realizar; además, permitiendo a las empresas dar a conocer los servicios que prestan, y que el usuario pueda conocer toda esta información al momento de visitar estas playas.

Esta aplicación se diseñó e implementó utilizando la metodología SCRUM que permite un desarrollo rápido, realizando entregables con fechas por módulo a partir de historias de usuario en las que se compilan los requisitos de la aplicación.

Para el levantamiento de información se realizó una visita en campo, donde se tomaron un conjunto de características para los tipos de datos determinados y estableciendo su ubicación utilizando el GPS proporcionado por el celular, obteniendo de esta forma un conjunto de datos que fue compilado y almacenado en una base de datos relacional PostgreSQL®, sobre la que se realizaron un conjunto de API's, facilitando la interacción de los usuarios con la aplicación.

La aplicación se desarrolló en 4 módulos que incluyen la gestión de usuarios, información espacial, administración de ubicaciones y un sistema de recomendaciones en los que se utilizó un conjunto de librerías y API's como Mapbox, Firebase, entre otras que facilitaron el desarrollo e implementación de los diferentes eventos y funciones de cada módulo.

Se implementó un sistema de recomendación que permite a los usuarios obtener recomendaciones de actividades náuticas de acuerdo con un conjunto de características proporcionadas por el mismo, y visualizar las empresas que ofrecen este tipo de actividades para que el usuario tenga un conjunto de opciones de acuerdo con la selección realizada.

Esta aplicación es de gran utilidad tanto para turistas como para empresas, ya que a los primeros les permite guiarse en las playas de Juanchaco, Ladrilleros y La barra, permitiendo a usuarios que visitan estas playas por primera vez, consultar información y recibir recomendaciones de actividades a realizar; y a los segundos les facilita la difusión de actividades, restaurantes y hoteles, para que tantos turistas nuevos como antiguos puedan conocer los servicios que ofrecen estas empresas.

Finalmente, se realizó un conjunto de pruebas en las que se evaluó la realización de consultas, información, recomendaciones y consistencia en el diseño que permitieron conocer, de acuerdo con los comentarios realizados por los usuarios, las mejoras que se deben realizar a futuro, con miras a fortalecer el turismo en las playas de Juanchaco, Ladrilleros y La barra.

Trabajo Futuro

Añadir más información a las actividades, hoteles y restaurantes, permitiendo ver datos como el teléfono, página web, fotografía del lugar, etc., permitiendo al usuario ver una información mucho más completa y detallada del turismo que ofrecen las playas.

Expandir la zona de estudio, para ofrecer información no solo de las playas de Juanchaco, Ladrilleros y La Barra, sino del resto de las playas que componen el municipio de Buenaventura. También, se podría expandir hacia otros departamentos como el Chocó. Todo esto con el fin de impulsar el turismo de la región pacífico, ya que ofrece mucha cultura y recreación que muy poca gente conoce.

Implementar un visor alfanumérico, donde se pueda ver la información de cada actividad, restaurante y hotel sin la necesidad de ubicarlos sobre un mapa; así como un filtro tanto alfanumérico como espacial, que facilite al usuario la visualización de sitios turísticos que se adapten a las preferencias del usuario.

Aumentar la cantidad de datos y características para alimentar el modelo de inteligencia artificial, teniendo así mejores recomendaciones y un modelo más preciso. Así como evaluar alternativas para la implementación del sistema de recomendaciones, como predictions de Firebase, que utiliza las estadísticas de las actividades realizadas por los usuarios, generando predicciones mucho más personalizadas, que se basan en el comportamiento del usuario.

Mejorar las estrategias de seguridad en el manejo del usuario administrador al momento de realizar cambios en la base de datos, impidiendo que se realicen modificaciones que no sean verificadas por un super administrador.

Aumentar la cantidad de encuestas tanto en la apreciación turística, como en el uso de la aplicación, con el fin de tener un análisis mas preciso de la población a la que se dirige, y de esta forma determinar las necesidades de los usuarios tanto en aspectos turísticos como de la aplicación móvil y hacer mejoras a partir de las observaciones realizadas por los turistas.

Es necesario hacer seguimiento a la percepción de los usuarios respecto al turismo en la región, debido a esto recomendable realizar encuestas a los usuarios de la aplicación en las que se detalle cómo se percibe el turismo al utilizar esta y así poder compararlas con las encuestas de apreciación turísticas realizadas con anterioridad, con el fin de determinar si hay mejoras al utilizar esta aplicación en la percepción de los turistas en las playas de Juanchaco, Ladrilleros y La barra.

REFERENCIAS

- Altexsoft. (2019). *What is API: Definition, Specifications, Types, Documentation* | AltexSoft. Altexsoft.
<https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/>
- Altimira Vega, R., & Muñoz Vivas, X. (2007). El turismo como motor de crecimiento económico. *Anuario Jurídico y Económico Escurialense*, 40, 677–710.
- Amores Bassante, A. A., & Suárez Loaiza, C. A. (2016). *Diseño y desarrollo de un prototipo de aplicación web y móvil nativa android para el monitoreo de seguridad en el Distrito Metropolitano de Quito a través de la red social twitter*. <http://repositorioslatinoamericanos.uchile.cl/handle/2250/2790513>
- Android Developers. (2019). *Introducción a las actividades* | *Desarrolladores de Android*. Developers.
<https://developer.android.com/guide/components/activities/intro-activities?hl=es>
- Bagnato, J. I. (2019). *Sistemas de Recomendación* | *Aprende Machine Learning*. Aprende Machine Learning.
<https://www.aprendemachinelearning.com/sistemas-de-recomendacion/>
- Barua Gono Bishwabidyalay, L., Akter Gono Bishwabidyalay, T., Barua, L., Bishwabidyalay Savar, G., Md Sharif Ahamed, D., & Tania Akter, D. (2019). Analyzing Cervical Cancer by using an Ensemble Learning Approach based on Meta Classifier Smart Autism View project Predicting Accidental Locations of Dhaka-Aricha Highway in Bangladesh using Different Data Mining Techniques View project Analyzing Cervical Cancer by using an Ensemble Learning Approach based on Meta Classifier. *Article in International Journal of Computer Applications*, 182(46), 975–8887. <https://doi.org/10.5120/ijca2019918619>
- Betarte, L., Machado, R., & Molina, V. (2006). *PGMÚSICA SISTEMA DE RECOMENDACIÓN DE MÚSICA*.
- Brownlee, J. (2014). *Una suave introducción a Scikit-Learn*. Machine Learning Mastery.
<https://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/>
- Caicedo, C., Acuña, J., Rodríguez, A. del C., & Wellington, R. (2016). Aplicativo móvil como estrategia de marketing para el impulso de la matriz productiva en el área de Turismo Mobile. *3Ciencias*, 5(1), 41–53.
<https://doi.org/http://dx.doi.org/10.17993/3ctecno.2016.v5n1e17.41-53>
- Camacho, M. J. J., & Cornavaca, A. R. F. (2017). *Aplicación móvil bajo plataforma Android para la interacción y obtención de información de las áreas y especies del Jardí Botánico Ambiental UNAN-León a través de la utilización de Códigos QR y Mapeo Web*.
- Climent, P. V. (2012). *¿Qué productos y servicios ofrece Mapbox? - MappingGIS*. MappingGIS SLU.
<https://mappinggis.com/2019/11/que-productos-y-servicios-ofrece-mapbox/>
- Codecademy. (2021). *¿Qué es CRUD?* | Codecademy. Codecademy.
<https://www.codecademy.com/articles/what-is-crud>
- Cruz, G., Israel, J., Análisis, E. L., Impacto, D. E. L., Del, T., Boom, T., En, T., & Sig, I. G. (2015). *EL ANÁLISIS DEL IMPACTO TERRITORIAL DEL TERCER BOOM TURÍSTICO EN CANARIAS (ESPAÑA)*.
- Dave, A. (2018). *Regression in Machine Learning. Regression models are used to predict a...* | by Apoorva Dave | Data Driven Investor | Medium. Data Driven Investor.

<https://medium.com/datadriveninvestor/regression-in-machine-learning-296caae933ec>

Developers, A. (2020). *Fragmentos | Desarrolladores de Android | Android Developers*. Developers. <https://developer.android.com/guide/fragments?hl=es>

EKCIT. (2019). Base de datos : ¿qué tipos hay y cómo funciona conectada a un software? In *TIC Portal*. <https://www.ticportal.es/glosario-tic/base-datos-database>

Eric Gundersen. (2018). *Mapbox ❤️ Valhalla. Por Eric Gundersen | de Mapbox | mapas para desarrolladores*. Mapbox. <https://blog.mapbox.com/mapbox--valhalla-1fdda95ac4f>

FAO. (2006). *Sistemas de Información Geográficos (SIG) en Salud Animal*. http://www.fao.org/tempref/GI/Reserved/FTP_FaoRlc/old/prior/segalim/animal/sig/intro/compo.htm

Gabri. (2018). *¿Cómo funcionan los dispositivos GPS? Trilateración vs Triangulación | El blog de franz*. ArcGeek. <https://acolita.com/como-funcionan-los-dispositivos-gps-trilateracion-vs-triangulacion/>

Gao, R., Li, J., Li, X., Song, C., & Zhou, Y. (2018). A personalized point-of-interest recommendation model via fusion of geo-social information. *Neurocomputing*, 273, 159–170. <https://doi.org/10.1016/j.neucom.2017.08.020>

Geeksforgeeks. (2020). *LinkedHashSet en Java con ejemplos - GeeksforGeeks*. GeeksforGeeks. <https://www.geeksforgeeks.org/java-util-hashmap-in-java-with-examples/>

Geoinnova. (2018). *¿Qué son los Sistemas de Información Geográfica (SIG)? - Cursos Geoinnova.org*. 1. <https://geoinnova.org/cursos/que-son-los-sistemas-de-informacion-geografica-sig/>

GFourmis. (2020). *Oís - Secretos Turísticos (0.1.13)*. https://play.google.com/store/apps/details?id=com.gfourmis.ois&hl=es_419&gl=US

Google Developers. (2020). *Firebase Realtime Database | Firebase Realtime Database*. 2020. <https://firebase.google.com/docs/database>

Heroku. (2021). *Aplicaciones personales | Heroku*. Cloud Application Platform. <https://dashboard.heroku.com/apps>

Hunt-walker, N. (2018). *Una introducción al marco de la aplicación web Flask Python | Opensource.com*. Opensource.Com. <https://opensource.com/article/18/4/flask>

Hurtado Jerves, J. C., & Zabala Bustamante, B. A. (2018). *DESARROLLO DE UN SISTEMA INTELIGENTE WEB ADAPTATIVO PARA LA RECOMENDACIÓN DE DESTINOS TURÍSTICOS EN EL CANTÓN MORONA BASADO EN REGLAS Y HECHOS*.

Karimi, R., Nanopoulos, A., & Schmidt-Thieme, L. (2015). A supervised active learning framework for recommender systems based on decision trees. *User Modeling and User-Adapted Interaction*, 25(1), 39–64. <https://doi.org/10.1007/s11257-014-9153-z>

Leonelli, R. C. B., & Fernández, J. P. S. (2007). *Diseño e implementación de un servicio de localización y visualización de mapas utilizando J2ME para dispositivos móviles y herramientas de libre distribución*. 2, 39–57.

Liu, C., Mao, Q., Chu, X., & Xie, S. (2019). An Improved A-Star Algorithm Considering Water Current, Traffic

- Separation and Berthing for Vessel Path Planning. *Applied Sciences*, 9(6), 1057.
<https://doi.org/10.3390/app9061057>
- Lucid Software Inc. (n.d.). *¿Qué es un modelo de base de datos?* | Lucidchart. Retrieved February 19, 2021, from <https://www.lucidchart.com/pages/es/que-es-un-modelo-de-base-de-datos>
- Luis Leiva Olivencia, J., Guevara, A., & Rossi, C. (2015). *RAMCAT: Modelo para generar recomendaciones en un sistema de realidad aumentada contextual basándose en las preferencias del turista*. 13, 649–668.
<https://doi.org/10.25145/j.pasos.2015.13.045>
- Minube. (2020). *Cali Guía Turística con Mapa CO*.
https://play.google.com/store/apps/details?id=com.minube.guides.costarica&hl=es_419&gl=US
- Mira, E. (1922). ¿Qué es la inteligencia? *Revista de Pedagogía*, I(2), 56–62.
<https://www.ionos.es/digitalguide/online-marketing/vender-en-internet/que-es-la-inteligencia-artificial/>
- Molina Torres, M. del P., & Ortiz Urbano, R. (2016). DISEÑO DE UNA APLICACIÓN SIG PARA LA EXPLOTACIÓN DE RECURSOS TURÍSTICOS DE UN TERRITORIO. *Aracne REVISTA ELECTRONICA DE RECURSOS EN INTERNET SOBRE GEOGRAFÍA Y CIENCIAS SOCIALES*, 1–18.
- Moses, J. S., & Dhinesh Babu, L. D. (2017). Deep learning model schemes to address the scrutability and in-memory purchase issues in recommender system. *Communications in Computer and Information Science*, 750, 326–335. https://doi.org/10.1007/978-981-10-6544-6_30
- Mozilla. (2019). *An overview of HTTP - HTTP* | MDN. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>
- Navarro Benites, K. A. (2013). *Recomendación social con aplicaciones a turismo. Memoria de Proyecto Fin De Máster en Sistemas*.
- Noren, A. (2019). *Árbol de decisión en Machine Learning (Parte 1)* - sitiobigdata.com. Sitio Big Data.
<https://sitiobigdata.com/2019/12/14/arbol-de-decision-en-machine-learning-parte-1/>
- OMT. (2019). *Glosario de términos de turismo* | OMT. Glosario de Términos Turísticos.
<https://www.unwto.org/es/glosario-terminos-turisticos>
- Plataforma Tecnológica de la Macaronesia. (2015). *Qué es el turismo náutico y qué beneficios tiene*.
<https://ptmac.webs.ull.es/blog/2015/07/29/el-turismo-nautico-en-la-macaronesia/>
- Proteccion, A. ley. (2019). *El Modelo base de datos: Definición y tipos*.
<https://ayudaleyprotecciondatos.es/bases-de-datos/modelos/>
- Rocha Muñoz, T. (2017). *Sistema de recomendación basado en redes neuronales competitivas*.
<http://hdl.handle.net/10630/13461>
- Socolombia. (2019). *ViajaxColombia (1.0.5)*.
https://play.google.com/store/apps/details?id=com.socolombia.viajaxcolombia&hl=es_419&gl=US
- Softcorp. (2019). *Definición y cómo funcionan las aplicaciones móviles*. Servisoftcorp.Com.
<https://servisoftcorp.com/definicion-y-como-funcionan-las-aplicaciones-moviles/>
- Torres, I. (2020). *Diagrama de Flujo - Una Herramienta para Visualizar tus Procesos*. IVE Consultores.
<https://iveconsultores.com/diagrama-de-flujo/>

- Valdés, D. (2007). ¿Qué son las bases de datos? 26/10/2007, 1. <https://imasgal.com/que-son-bases-de-datos-espaciales/>
- Valhalla. (2018). *GitHub - valhalla / valhalla-docs: DEPRECATED: La documentación actualizada para valhalla se puede encontrar en el repositorio principal*. Github. <https://github.com/valhalla/valhalla-docs>
- Wang, S., Gong, M., Li, H., Yang, J., & Wu, Y. (2017). Knowledge-Based Systems Memetic algorithm based location and topic aware recommender system. *Knowledge-Based Systems*, 131, 125–134. <https://doi.org/10.1016/j.knosys.2017.05.030>
- World Tourism Organization. (n.d.). *¿Por qué el Turismo? | OMT*. UNWTO. Retrieved March 25, 2021, from <https://www.unwto.org/es/turismo>
- Zeroual, A., Harrou, F., Dairi, A., & Sun, Y. (2020). Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. *Chaos, Solitons and Fractals*, 140. <https://doi.org/10.1016/j.chaos.2020.110121>
- Zhang, L., & He, X. (2012). Route search base on pgRouting. *Advances in Intelligent and Soft Computing*, 115 AISC(VOL. 2), 1003–1007. https://doi.org/10.1007/978-3-642-25349-2_133

ANEXOS

Anexo A. Estructura de la encuesta realizada a los turistas en el municipio de Buenaventura.

La encuesta que se encuentra a continuación fue aplicada a 20 turistas que se encontraban de visita en el muelle del municipio de Buenaventura, realizada en el mes de febrero del 2020.

Preguntas de la encuesta sobre el turismo

- ¿Qué playa tiene pensado visitar?
 - Juanchaco
 - Ladrilleros
 - La Barra
- ¿Tenía usted conocimiento de las actividades que ofrece esta playa?
 - Si, existe buena información
 - No, buscaré en el camino
 - Tengo una guía
 - Lo que encontré por internet
- ¿Considera la información de internet segura y eficaz?
 - Si
 - No
- ¿Se puede ubicar fácilmente con un mapa físico?
 - Si
 - No
 - A veces
- ¿Qué herramienta utiliza para conocer la ruta a su destino deseado?

- Mapa físico
- Preguntando a la gente
- Dejando que el taxi o moto lo lleve
- ¿Cuál es el dispositivo tecnológico que más utiliza diariamente?
 - Celular
 - Computador
 - Tablet
 - Otro
- ¿Qué sistema operativo utiliza en su dispositivo móvil?
 - Android
 - iOS
 - Blackberry OS
 - Windows Phone
- Si tuviera la oportunidad de conocer sobre las diferentes actividades que ofrece las playas ¿Consideraría que sería mucho más fácil viajar y conocer?
 - Si
 - No
- Durante su viaje, ¿Piensa usted que tener un aplicativo que le muestre las rutas a su destino, le resultaría más seguro y sencillo?
 - Si
 - No

Anexo B. Código del registro a la base de datos.

La siguiente imagen muestra la secuencia lógica que se escribió en un lenguaje JavaScript, para el almacenamiento en la base de datos de la información que ingresa el usuario, una vez el usuario ingresa esos datos y le da en el botón enviar, el sistema verifica que los campos estén correctos y luego los sube a la base de datos de Firebase.

```

if(TextUtils.isEmpty(nombre)){
    Toast.makeText( context: this, text: "Debe ingresar un nombre",Toast.LENGTH_LONG).show();
    return;
}
if(TextUtils.isEmpty(email)){
    Toast.makeText( context: this, text: "Debe ingresar un email",Toast.LENGTH_LONG).show();
    return;
}
if(TextUtils.isEmpty(contraseña)){
    Toast.makeText( context: this, text: "Debe ingresar una contraseña",Toast.LENGTH_LONG).show();
    return;
}
firebaseAuth.createUserWithEmailAndPassword(email, contraseña)
    .addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if(task.isSuccessful()){
                Map<String, Object> map= new HashMap<>();
                map.put("nombre", nombre);
                map.put("email", email);
                map.put("contraseña", contraseña);
                map.put("usuario", usuario);

                String id = firebaseAuth.getCurrentUser().getUid();
                firebaseDatabase.child("Users").child(id).setValue(map);
                Intent intent = new Intent( packageContext: Main3Activity.this,Main2Activity.class);
                Main3Activity.this.startActivity(intent);
            }
            else{
                if (task.getException() instanceof FirebaseAuthUserCollisionException){
                    Toast.makeText( context: Main3Activity.this, text: "Ese usuario ya existe",Toast.LENGTH_LONG).show();
                }
                else {
                    Toast.makeText( context: Main3Activity.this, text: "No se pudo registrar el usuario",Toast.LENGTH_LONG).show();
                }
            }
        }
    });
}

```

Figura 43. Código de registro de usuario en la base de datos Firebase.

Fuente. Elaboración propia.

Anexo C. Código del inicio de sesión.

En la siguiente imagen se muestra el código que hace posible el inicio de sesión. Para esto, el usuario debe haber creado una cuenta anteriormente y llenar los datos mencionados. La aplicación validará esta información y en caso de ser llenada incorrectamente o que el usuario no exista mostrará un mensaje diciendo que los datos son incorrectos y no permitirá entrar al sistema. Una vez se ingresen los datos el sistema con el evento LoginUsuario, establece si los datos fueron llenados, para posteriormente, con la función de autenticación “signInWithEmailAndPassword” determinar si el correo y contraseña fueron ingresados correctamente; si la operación fue realizada correctamente, enviará al usuario a MapActivity.

```

public void onClick2(View view) {
    LoginUsuarios();
}

public void LoginUsuarios() {
    final String email = LoginEmail.getText().toString().trim();
    final String contraseña = LoginContraseña.getText().toString().trim();

    if (TextUtils.isEmpty(email)) {
        Toast.makeText( context this, text: "Debe ingresar un email", Toast.LENGTH_LONG).show();
        return;
    }
    if (TextUtils.isEmpty(contraseña)) {
        Toast.makeText( context this, text: "Debe ingresar una contraseña", Toast.LENGTH_LONG).show();
        return;
    }

    firebaseAuth.signInWithEmailAndPassword(email, contraseña)
        .addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    Toast.makeText( context: Main2Activity.this, text: "Bienvenido: "+ email, Toast.LENGTH_LONG).show();
                    Intent intent = new Intent( packageContext: Main2Activity.this, MapActivity.class);
                    Main2Activity.this.startActivity(intent);
                    finish();
                }
                else {
                    Toast.makeText( context: Main2Activity.this, text: "No se pudo iniciar sesión, comprueba los datos", Toast.LENGTH_LONG).show();
                }
            }
        });
}
}

```

Figura 44. Código de inicio de sesión.

Fuente. Elaboración propia.

Anexo D. Código de la configuración de datos del usuario.

En la siguiente imagen, se muestra la función con la que se pueden actualizar la información de usuario de la sesión activa, permitiendo cambiar nombre, correo y contraseña de manera sencilla. Esta opción se puede encontrar en el menú de la aplicación.

```

boton_actualizar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String edtNombre2=edtNombre.getText().toString();
        String edtEmail2=edtEmail.getText().toString();
        String edtContra2=edtContra.getText().toString();
        Map<String, Object> datos= new HashMap<>();
        datos.put("nombre",edtNombre2);
        datos.put("email",edtEmail2);
        datos.put("contraseña",edtContra2);

        String id = firebaseAuth.getCurrentUser().getUid();

        firebaseDatabase.child("Users").child(id).updateChildren(datos).addOnSuccessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Toast.makeText(getActivity(), text: "Datos Actualizados correctamente",Toast.LENGTH_SHORT).show();
                MapFragment fragmentMap = new MapFragment();
                FragmentManager fm = getFragmentManager();
                FragmentTransaction ft = fm.beginTransaction();
                ft.replace(R.id.container,fragmentMap);
                ft.commit();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(getContext(), text: "Hubo un error al actualizar",Toast.LENGTH_SHORT).show();
            }
        });
    }
});

```

Figura 45. Código de actualización de datos de usuario.

Fuente. Elaboración propia.

Anexo E. Código para la visualización de la información de los marcadores.

En la siguiente imagen se muestra el código con el que se crea el evento que permite que se muestre la información de cada marcador al darles clic, por medio del evento “handleClickIcon”.

```

private boolean handleClickIcon(PointF screenPoint) {
    List<Feature> features = mapboxMap.queryRenderedFeatures(screenPoint, MARKER_LAYER_ID);
    if (!features.isEmpty()) {
        String name = features.get(0).getStringProperty(PROPERTY_NAME);
        List<Feature> featureList = featureCollection.features();
        if (featureList != null) {
            for (int i = 0; i < featureList.size(); i++) {
                if (featureList.get(i).getStringProperty(PROPERTY_NAME).equals(name)) {
                    if (featureSelectStatus(i)) {
                        setFeatureSelectState(featureList.get(i), selectedState: false);
                    } else {
                        setSelected(i);
                    }
                }
            }
        }
        return true;
    } else {
        return false;
    }
}

private void setSelected(int index) {
    if (featureCollection.features() != null) {
        Feature feature = featureCollection.features().get(index);
        setFeatureSelectState(feature, selectedState: true);
        refreshSource();
    }
}

```

Figura 46. Código para la visualización de la información de los marcadores al ser clicados.

Fuente. Elaboración propia.

Anexo F. Código para la generación de la ruta.

El código que se muestra en la siguiente imagen, se realizó para generar la ruta desde la ubicación del usuario, que el sistema lo realiza por medio de la clase “locationcomponent” que coge las coordenadas del GPS del celular. A partir de esto, se utiliza el método “DirectionResponse” para que el sistema genere la ruta.

```

@Override
public boolean onMapLongClick(@NonNull LatLng point) {
    Point destinationPoint = Point.fromLngLat(point.getLongitude(), point.getLatitude());
    Point originPoint = Point.fromLngLat(locationComponent.getLastKnownLocation().getLongitude(),
        locationComponent.getLastKnownLocation().getLatitude());
    GeoJsonSource source = mapboxMap.getStyle().getSourceAs("destination-source-id");
    if (source != null) {
        source.setGeoJson(Feature.fromGeometry(destinationPoint));
    }
    getRoute(originPoint, destinationPoint);
    return true;
}

private void getRoute(Point origin, Point destination) {
    assert Mapbox.getAccessToken() != null;
    NavigationRoute.builder(getActivity())
        .accessToken(Mapbox.getAccessToken())
        .origin(origin)
        .destination(destination)
        .build()
        .getRoute(new Callback<DirectionsResponse>() {
            @Override
            public void onResponse(@NonNull Call<DirectionsResponse> call, @NonNull Response<DirectionsResponse> response) {
                Timber.d("Response code: %s", response.code());
                if (response.body() == null) {
                    Timber.e("No routes found, make sure you set the right user and access token.");
                    return;
                } else if (response.body().routes().size() < 1) {
                    Timber.e("No routes found");
                    return;
                }
                currentRoute = response.body().routes().get(0);
                if (navigationMapRoute != null) {
                    navigationMapRoute.removeRoute();
                } else {
                    navigationMapRoute = new NavigationMapRoute(navigation, null, mapView, mapboxMap, R.style.NavigationMapRoute);
                }
                navigationMapRoute.addRoute(currentRoute);
            }
            @Override
            public void onFailure(@NonNull Call<DirectionsResponse> call, @NonNull Throwable throwable) {
                Timber.e("Error: %s", throwable.getMessage());
            }
        });
}

```

Figura 47. Código para generar la ruta.

Fuente. Elaboración propia.

Anexo G. Código de acceso para la modificación de datos de la base de datos.

En el siguiente código se muestra como hace el sistema para autorizar que solo el usuario empresa pueda acceder a la posibilidad de modificar o agregar datos nuevos a la base de datos. Restringiendo que los usuarios turistas ingresen a este campo.

```

private void getUserInfo(){
    firebaseAuth= FirebaseAuth.getInstance();
    firebasedatabase= FirebaseDatabase.getInstance().getReference();
    String id = firebaseAuth.getCurrentUser().getUid();
    firebasedatabase.child("Users").child(id).child("usuario").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if (dataSnapshot.exists()){
                String usuario= (String) dataSnapshot.getValue().toString();

                if (usuario.equals("Empresa")){

                    FragmentManager fragmentManager;
                    fragmentManager=getActivity().getSupportFragmentManager();
                    fragmentTransaction=fragmentManager.beginTransaction();
                    fragmentTransaction.replace(R.id.container, new FormularioFragment());
                    fragmentTransaction.commit();
                }
                else{
                    Toast.makeText(getActivity(), text: "Debe tener permiso de administrador", Toast.LENGTH_SHORT).show();
                }
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
}

```

Figura 48. Código de acceso para la modificación de datos de la base de datos.

Fuente. Elaboración propia.

Anexo H. Código para el envío de los datos a la API.

En la siguiente imagen se muestra el código con el que el sistema envía los datos a la API, una clave que debe ser verificada para ingresar los datos nuevos o modificados a la base de datos. Si la clave no coincide el sistema lanzará un mensaje de error.

```

private void guardarActividad(){
    if(isValidarCampos()){
        String nombre = nombre2.getText().toString();
        String horario = horario2.getText().toString();
        String precio = precio2.getText().toString();
        String descripcion = descripcion2.getText().toString();
        String lat = latitud.toString();
        String lon = longitud.toString();
        Map<String,String> datos = new HashMap<>();
        datos.put("name",nombre);
        datos.put("horario",horario);
        datos.put("precio",precio);
        datos.put("descripcion",descripcion);
        datos.put("latitud",lat);
        datos.put("longitud",lon);
        JSONObject jsonData = new JSONObject(datos);
        AndroidNetworking.post( url: "https://geojson-url.herokuapp.com/insertar_actividades.php")
            .addJSONObjectBody(jsonData)
            .setPriority(Priority.MEDIUM)
            .build()
            .getAsJSONObject(new JSONObjectRequestListener() {
                @Override
                public void onResponse(JSONObject response) {
                    try {
                        String message = response.getString( name: "message");
                        Toast.makeText(getActivity(),message,Toast.LENGTH_SHORT).show();
                        Intent intent = new Intent(getActivity(), MapActivity.class);
                        startActivity(intent);
                    } catch (JSONException e) {
                        Toast.makeText(getActivity(), text: "Error: "+e.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                }
                @Override
                public void onError(ANError anError) {
                    Toast.makeText(getActivity(), text: "Error: "+anError.getErrorDetail(), Toast.LENGTH_SHORT).show(); });
            }
        }else{
            Toast.makeText(getActivity(), text: "No se puede insertar una actividad si existen campos vacios", Toast.LENGTH_SHORT).show();
        }
    }
}

```

Figura 49. Código para el envío de datos a la API.

Fuente. Elaboración propia.

Anexo I. Código para el procesamiento y envío de los datos.

El siguiente código muestra la API que procesa los datos, convirtiéndolos a formato JSON y enviándolos a la base de datos.

```

<?php
require 'SQLGlobal2.php';

if($_SERVER['REQUEST_METHOD']=='POST'){
    try{
        $datos = json_decode(file_get_contents("php://input"),true);

        $name = $datos["name"]; // obtener parametros POST
        $horario=$datos["horario"];
        $precio = $datos["precio"];
        $descripcion = $datos["descripcion"];
        $latitud=$datos["latitud"];
        $longitud = $datos["longitud"];

        $respuesta = SQLGlobal::cudFiltro(
            "INSERT INTO actividades1 (name, horario, precio, descripcion, latitud, longitud) Values(?,?,?,?,?,?)",
            array($name,$horario,$precio,$descripcion,$latitud,$longitud)
        );//con filtro ("El tamaño del array debe ser igual a la cantidad de los '?'")
        if($respuesta>0){
            echo json_encode(array(
                'respuesta'=>'200',
                'message' => 'Se inserto correctamente la actividad',
                'data'=>'El numero de registros afectados es: '.$respuesta,
                'error'=>'')
            ));
        }else{
            echo json_encode(array(
                'respuesta'=>'100',
                'message' => 'No se inserto correctamente la actividad.',
                'data'=>'El numero de registros afectados es: '.$respuesta,
                'error'=>'')
            ));
        }
    }catch(PDOException $e){
        echo json_encode(
            array(
                'respuesta'=>'-1',
                'message' => 'Ocurrio un error, intentelo mas tarde',
                'data'=>'',
                'error'=>$e->getMessage())
        );
    }
}
?>

```

Figura 50. Código de la API que procesa y envía datos a la base de datos.

Fuente. Elaboración propia.

Anexo J. Estructura de la encuesta realizada a personas que realizan visitas usualmente visitas a sitios turísticos.

La encuesta que se encuentra a continuación fue aplicada a 202 personas que aceptaron colaborar a obtener esta información para el entrenamiento de la red neuronal que posteriormente, permita realizar recomendaciones a los usuarios de Navi_App. Debido a la condición actual que se presentó a nivel mundial por la pandemia, se compartió esta encuesta, por medio de las redes sociales, como Facebook, Instagram y se pidió colaboración con el plan de ingeniería topográfica, para que a través de los correos se pudiera compartir. Por lo anterior, se alcanzaron a recibir 202 respuestas.

Preguntas de la encuesta de información del turista.

- Sexo
 - Femenino
 - Masculino
- Edad
 - Menor de 18 años
 - 18 a 30 años
 - 31 a 50 años
 - Mayor de 50 años
- ¿Cuál es el tipo de actividad que prefieren realizar en su instancia en el sitio turístico?
 - Aventura
 - Recreación
 - Deporte (Caminata)
- ¿Cuándo realiza un viaje de turismo, normalmente viaja?
 - Individual
 - Con la familia
 - En grupo
- ¿Cuánto es el posible gasto que tendría para realizar una actividad turística?
 - \$10.000 a \$15.000

- \$20.000 a \$35.000
 - \$40.000 a \$65.000
 - \$70.000 a \$80.000
 - \$90.000 a \$120.000
 - Más de \$120.000
- Suponga que usted realizará un viaje a una de las siguientes playas, ¿Cuál sería su elección entre ellas?
 - Juanchaco
 - Ladrilleros
 - La Barra
- Suponiendo que esta planeando el viaje a las playas anteriores, ¿Qué mes del año escogería para viajar?
 - Enero
 - Febrero
 - Marzo
 - Abril
 - Mayo
 - Junio
 - Julio
 - Agosto
 - Septiembre
 - Octubre
 - Noviembre
 - Diciembre
- Suponiendo que está de visita en las playas anteriormente mencionadas, ¿Cuáles serían sus 3 actividades que escogería para realizar durante su estadía?
 - Juegos acuáticos
 - Lanchas

- Buceo
- Paseo Playas
- Pesca Deportiva
- Paseo Cascadas
- Ferry
- Paseo Manglares
- Caminata Ecológica
- Canoa
- Avistamiento de Ballenas
- Surf
- Paseo Piscinas Naturales
- Recorrido en Kayak
- Salida Nocturna Fitoplancton