

UCUENCA

Universidad de Cuenca

Facultad de Ingeniería

Carrera de Ingeniería de Sistemas

Prototipo de una aplicación móvil con información centralizada de los recursos turísticos: Caso de uso de las parroquias A y B.

Trabajo de titulación previo a la obtención del título de Ingeniero de Sistemas


Autores:

Angel Ricardo Jarro Pineda

Juan Carlos Lazo Lozado

Director:

Víctor Hugo Saquicela Galarza

ORCID:  0000-0002-2438-9220

Cuenca, Ecuador

2023-09-11

Resumen

El turismo desempeña un papel crucial en el crecimiento económico y el desarrollo de las comunidades locales. A medida que el mundo se vuelve más conectado y la tecnología avanza, las aplicaciones móviles han emergido como una herramienta indispensable en la industria del turismo. Estas aplicaciones proporcionan a los viajeros información instantánea y acceso a una amplia gama de servicios y actividades turísticas. En el Ecuador estas aplicaciones se enfocan en lugares específicos, como ciudades y cantones, lo que obliga a los turistas a instalar múltiples aplicaciones que tienen el mismo propósito. Esta fragmentación dificulta la experiencia del usuario y dificulta la centralización de la información turística. Además, la falta de una aplicación móvil integral obliga a los turistas a buscar información en los medios oficiales de cada lugar, un proceso tedioso y poco eficiente. Como resultado, la descentralización de la información y la escasez de promoción en las zonas rurales dificultan el potencial del turismo en estas áreas. Bajo este contexto, se ha desarrollado un prototipo de aplicación móvil y una aplicación web complementaria con el objetivo de promover y gestionar de manera centralizada los recursos turísticos.

Palabras clave: software, aplicación móvil, m-tourism, ontología



El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Cuenca ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por la propiedad intelectual y los derechos de autor.

Institutional Repository: <https://dspace.ucuenca.edu.ec/>

Abstract

Tourism plays a crucial role in economic growth and the development of local communities. As the world becomes more connected and technology advances, mobile applications have emerged as an indispensable tool in the tourism industry. These applications provide travelers with instant information and access to a wide range of services and tourist activities. In Ecuador, these applications focus on specific locations such as cities and towns, which forces tourists to install multiple applications that serve the same purpose: promoting tourist resources. This fragmentation hinders the user experience and complicates the centralization of tourist information. Additionally, the lack of a comprehensive mobile application compels tourists to search for information through official channels of each location, a tedious and inefficient process. Consequently, the decentralization of information and the limited promotion in rural areas pose challenges to the potential of tourism in these regions. Under this context, a prototype mobile application and a complementary web application have been developed with the objective of promoting and managing tourist resources in a centralized manner.

Keywords: software, mobile application, m-tourism, ontology



The content of this work corresponds to the right of expression of the authors and does not compromise the institutional thinking of the University of Cuenca, nor does it release its responsibility before third parties. The authors assume responsibility for the intellectual property and copyrights.

Institutional Repository: <https://dspace.ucuenca.edu.ec/>

Índice de contenido

1	Introducción	11
2	Antecedentes y trabajos relacionados	13
2.1	Turismo	13
2.1.1	Turismo en Ecuador	14
2.2	Aplicaciones móviles	14
2.2.1	Frameworks	15
2.3	Web Semantica y Ontologías	17
2.3.1	SPARQL	19
2.3.2	Ontologías en el Turismo	20
2.4	Arquitectura orientada a microservicios	20
2.4.1	Spring Boot	22
2.5	OAuth 2	22
2.6	Scrum	22
2.7	Pruebas de usabilidad	24
2.8	Trabajos relacionados	24
3	Desarrollo del Prototipo	28
3.1	Metodología	28
3.2	Análisis de Requisitos	34
3.3	Desarrollo de la ontología	38
3.4	Diseño de mockups	46
3.4.1	Prototipo de interfaces de aplicación móvil	46
3.4.2	Prototipo de interfaces de aplicación web	51
3.5	Arquitectura de Software	52
3.5.1	Frontend	52
3.5.2	Backend	64
4	Evaluación del prototipo	78
4.1	Ejecución de la evaluación	79

UCUENCA

	5
4.2 Obtención y Análisis de resultados	80
4.2.1 Cuestionario SUS	80
4.2.2 Cuestionario UEQ	82
5 Conclusiones y trabajos futuros	87
5.1 Conclusiones	87
5.2 Trabajos futuros	88
Anexo A Backlog	97
Anexo B Instancias de clases de la ontología	102
Anexo C Consulta principal para obtención de recursos turísticos	104
Anexo D Documento de análisis de requerimientos	106
Anexo E Manual de usuario de la aplicación Turis Up	121
Anexo F Manual de administrador de la aplicación Turis Up	146

Índice de figuras

2.1 Consulta Sparql (DuCharme, 2013)	19
3.1 Diagrama de caso de uso para registro de usuario	36
3.2 Especificación de caso de uso para registro de usuario	37
3.3 Diagrama de secuencia para registro de usuario	37
3.4 Diagrama de clases	38
3.5 Ontología de Tourpedia (Gazzè et al., 2015)	43
3.6 Clases y subclases de la ontología OnTour (Prantner et al., 2007)	44
3.7 Ontología resultante representada a través de un grafo	46
3.8 Página de login y splash screen	48
3.9 Página explorar	48
3.10 Ventana de información del recurso y comentarios	49
3.11 Ventana del mapa de recursos	50
3.12 Ventana de las rutas creadas por el administrador	50
3.13 Ventanas del perfil de usuario	51
3.14 Arquitectura del sistema	53
3.15 Diagrama de capas de la aplicación móvil	55
3.16 Crear cliente OAuth en Google Cloud	56
3.17 Permisos necesarios para la autenticación	56
3.18 Pantalla Explorar	57
3.19 Pantalla de registro de nuevos recursos turísticos	58
3.20 Pantalla del mapa de la aplicación	60
3.21 Pantallas de rutas	61
3.22 Pantallas de perfil	63
3.23 Pantalla de configuración de la aplicación	64
3.24 Pantallas de aplicación web	65
3.25 Flujo de acceso desde la aplicación web	67
3.26 Flujo de acceso desde la aplicación móvil	68
3.27 Modelo usuarios	69

3.28 Información de la aplicación	70
3.29 Imágenes de recursos subidos a Facebook	70
3.30 Atributos necesarios para RestFB	71
3.31 Métodos para subir imágenes y vídeos a Facebook con RestFB	71
3.32 Creación de un recurso con Apache Jena	72
3.33 Obtención de una ruta mediante RDF API	73
3.34 Consulta Sparql para la obtención de recursos	74
3.35 Consulta Sparql para la obtención de recursos con la distancia a ellos	75
3.36 Consulta Sparql para la obtención de recursos ordenados segun las valoraciones de los usuarios	76
3.37 Resultado de la consulta general para la obtención de recursos	76
4.1 Datos individuales del cuestionario SUS	80
4.2 Resultados individuales calculados en base a la formula del cuestionario SUS	81
4.3 Escala de resultados del cuestionario SUS	81
4.4 Análisis de los datos obtenidos en la encuesta UEQ: Resultados numéricos	82
4.5 Análisis de los datos obtenidos en la encuesta UEQ: Resultados gráficos	83
4.6 Análisis de las dimensiones de la encuesta UEQ: Resultados numéricos	84
4.7 Análisis de las dimensiones de la encuesta UEQ: Resultados gráficos	84
4.8 Análisis de las categorías de la encuesta UEQ: Resultados gráficos	86
4.9 Análisis de las categorías de la encuesta UEQ: Resultados gráficos	86
B.1 Instancia de una organización	102
B.2 Instancia de una región	102
B.3 Instancia de un recurso	103

Índice de tablas

3.1 Roles Scrum	29
3.2 Épicas	30
3.3 Valoración de prioridad	30
3.4 Herramientas y lenguajes de programación utilizados	34
3.5 Aplicaciones de turismo con más descargas desarrolladas en Ecuador	34
3.6 Ontologías utilizadas	44
3.7 Versiones de herramientas comunes de los microservicios	65
3.8 Endpoints del backend	77
A.2 Backlog	100

Agradecimientos

A mis queridos padres, quienes han sido mi mayor apoyo y fuente de inspiración, les agradezco de todo corazón. Gracias por su incondicional amor, por creer en mí y por alentarme a seguir mis sueños. Su constante aliento, sacrificio y dedicación han sido pilares fundamentales en mi formación académica y personal.

No puedo dejar de mencionar y agradecer de manera especial a nuestro director de tesis, Víctor Saquicela. Su experiencia, conocimientos y orientación han sido invaluable para el éxito de este proyecto. Agradezco su dedicación, paciencia y sabios consejos que han guiado en el desarrollo de este trabajo.

- *Ricardo*

Quiero aprovechar este espacio para expresar mi más sincero y profundo agradecimiento a mi amada madre. Su amor incondicional, su apoyo constante y sus palabras de aliento han sido un faro de luz en mi camino. Gracias por ser mi fuente inagotable de inspiración y por estar siempre ahí, brindándome tu constante sacrificio y dedicación. Tu ejemplo ha sido un pilar fundamental en mi formación personal y académica, demostrándome que todo es alcanzable con perseverancia. Tu amor y guía han sido fundamentales en todos mis logros. También quiero agradecer a mis hermanas por su apoyo y confianza, su compañía ha hecho de este viaje un camino aún más significativo. Juntos, hemos superado obstáculos y celebrado éxitos. Estoy profundamente agradecido por tener una familia tan maravillosa a mi lado.

Además, quiero expresar mi más sincero agradecimiento a nuestro director de tesis, Víctor Hugo Saquicela. Su apoyo incondicional, enseñanzas y dedicación han sido fundamentales para el desarrollo y éxito de este proyecto. Agradezco profundamente sus valiosos conocimientos, experiencia y guía, que nos han impulsado a alcanzar resultados sobresalientes. Sus ideas, sugerencias, paciencia y dirección han sido indispensables en cada etapa de este proyecto. Estoy verdaderamente agradecido por su invaluable contribución y por haber sido un mentor excepcional en esta etapa de mi formación.

- *Juan Carlos*

1. Introducción

En la era digital, el turismo ha experimentado una notable evolución. La integración de aplicaciones móviles en el sector turístico, conocidas como m-Tourism apps, ha revolucionado la forma en que los turistas acceden a la información y disfrutan de sus experiencias de viaje (Saura et al., 2017). Estas aplicaciones móviles no solo transforman la manera en que los turistas se conectan con los destinos, sino que también desempeñan un papel fundamental en el desarrollo local al promover la participación de las comunidades y generar beneficios económicos en las regiones visitadas (Inversini et al., 2020). Sin embargo, es importante reconocer que no siempre se logra esta integración efectiva entre el turismo y el desarrollo local. A menudo, existen desafíos en la implementación de estas aplicaciones y en la participación activa de las comunidades, lo que puede limitar su impacto positivo.

En Ecuador, aplicaciones móviles de turismo como Airbnb, TripAdvisor y Foursquare son muy populares (Encalada, 2017), pero se enfocan principalmente en brindar servicios relacionados con precios de vuelos, hoteles, hospedaje y restaurantes, sin prestar atención específica a los recursos turísticos en sí mismos. Según la definición de recurso turístico, son aquellos bienes y servicios que atraen visitantes y hacen posible la actividad turística (Sánchez, 2010), estas aplicaciones no se centran exclusivamente en ellos. Además, en el contexto de las aplicaciones turísticas nacionales en Ecuador, se han desarrollado y registrado alrededor de 150 aplicaciones (Alejandro et al., 2022), pero la mayoría de ellas se limitan en áreas específicas y reducidas como ciudades y cantones, lo que obliga a los turistas a instalar varias aplicaciones con el mismo propósito: promocionar recursos turísticos. Esto resulta en un proceso tedioso para los turistas que deben buscar información en diferentes fuentes y una descentralización de la información de los recursos turísticos. Además, no existe una aplicación de referencia que abarque toda la información de los recursos turísticos, lo cual sería beneficioso para impulsar el turismo, especialmente en las zonas rurales que no reciben tanta promoción.

Basado en el contexto expuesto en este trabajo se plantea el desarrollo de un prototipo de una aplicación móvil y una aplicación web complementaria con el objetivo de promover y gestionar de manera centralizada los recursos turísticos. El prototipo de la aplicación móvil permite a los usuarios explorar y descubrir sitios turísticos relevantes, basándose en una

ontología que centraliza la información relacionada. Por otro lado, la aplicación web proporciona herramientas y funcionalidades para la gestión eficiente de estos recursos. Ambas aplicaciones se complementan entre sí, ofreciendo una experiencia integral para los usuarios y contribuyendo a potenciar el turismo en las áreas rurales de manera más efectiva y accesible.

Objetivo General

Desarrollar un prototipo de una aplicación móvil, con información centralizada de los recursos turísticos de las parroquias A y B, con el fin de mejorar la experiencia de los turistas. Para este proyecto se tomarán como referencia las parroquias Sayausí y Baños del cantón Cuenca.

Objetivos específicos

- Analizar, revisar y comprender la relación existente entre las aplicaciones móviles y el turismo a través de artículos científicos y documentación.
- Determinar requerimientos del prototipo mediante una recolección de funcionalidades esperadas del sistema y analizando aplicaciones de turismo existentes en el mercado.
- Desarrollar un aplicativo móvil con un frontend basado en un framework y un backend usando microservicios, siguiendo la metodología ágil Scrum.
- Realizar la evaluación del prototipo generado mediante pruebas de usabilidad y experiencia de usuario.

La estructura del documento es la siguiente: en primer lugar, en la sección 2 se presentan los conceptos teóricos fundamentales que son necesarios para comprender el proyecto, así como también se revisan trabajos previos relacionados con el turismo móvil. Esto permite establecer una base sólida de conocimiento en el área. En la sección 3 se detalla la metodología utilizada y el desarrollo en si del proyecto, tanto en el aspecto técnico del backend como en el diseño del frontend. Se describen los pasos y las herramientas utilizadas para implementar la propuesta. La sección 4 se dedica a describir la evaluación realizada, junto con el análisis de los resultados obtenidos. Se examinan los aspectos clave del proyecto y se evalúa su usabilidad y la experiencia de usuario. Finalmente, en la sección 5 se presentan las conclusiones derivadas del proyecto, resumiendo los hallazgos y resultados obtenidos. Además, se plantean posibles trabajos futuros que podrían realizarse para ampliar o mejorar aún más el tema abordado en el proyecto.

Este enfoque estructurado y detallado permite una comprensión clara del proceso seguido en el desarrollo del proyecto, desde los conceptos teóricos iniciales hasta las conclusiones finales y las posibilidades de investigación futura.

2. Antecedentes y trabajos relacionados

En esta sección, se exponen los conceptos teóricos fundamentales y se analiza el estado del arte relacionado con el trabajo de tesis, con el objetivo de brindar un pleno entendimiento de la solución propuesta.

2.1. Turismo

El turismo es una actividad global que permite a las personas viajar y visitar diversos destinos con diferentes propósitos, proporcionando la oportunidad de descubrir nuevos lugares, sumergirse en tradiciones diversas y fomentar la interacción entre personas de diferentes culturas. Según la World Travel & Tourism Council (WTTC), el turismo es una actividad de gran importancia debido a su impacto social, económico y cultural. Representó aproximadamente el 10% del Producto Interno Bruto (PIB) a nivel global en el año 2019, además de generar millones de empleos en todo el mundo (Nexotur, 2019). También desempeña un papel crucial en el desarrollo socioeconómico de muchas regiones, especialmente en aquellos lugares con recursos naturales, culturales o históricos destacados (WTTC, 2020).

El turismo no solo impulsa la economía local a través del gasto directo de los turistas en alojamiento, sino que también tiene efectos positivos en otros sectores relacionados. Por ejemplo, el turismo estimula la demanda de servicios de construcción para la mejora de infraestructuras turísticas, como hoteles, aeropuertos y centros de convenciones (WTTC, 2020). De igual manera, impulsa el comercio minorista al aumentar la demanda de productos y servicios locales, como artesanías, alimentos típicos y recuerdos. Además, el sector turístico requiere una amplia gama de servicios, como transporte, guías turísticos, agencias de viajes y servicios de entretenimiento, generando oportunidades de empleo tanto directas como indirectas (Page, 2014).

Por otro lado, el turismo no solo implica el desplazamiento de los turistas hacia nuevos destinos, sino que también permite un enriquecedor intercambio cultural. Al visitar otros destinos, los

turistas tienen la oportunidad de sumergirse en diferentes culturas, tradiciones y formas de vida, lo que contribuye a una mayor apreciación y respeto por la diversidad cultural (Page, 2014).

2.1.1. Turismo en Ecuador

En Ecuador, se reconoce formalmente la importancia del turismo como motor de desarrollo nacional, ya que se destaca por ser la tercera fuente de ingresos no petroleros, después del banano y el camarón, lo que evidencia la relevancia de este sector para la economía Ecuatoriana. (Ministerio de Turismo del Ecuador, 2020), mostrándose como un sector con potencial para diversificar la estructura económica en comparación con las actividades tradicionales. El turismo desempeña un papel fundamental en la generación de empleo, la obtención de divisas y la actividad económica en general (Medina, 2022).

De acuerdo con los registros estadísticos de abril de 2022 sobre entradas y salidas internacionales, se observa que ingresaron al país 590.006 extranjeros, de los cuales el 46.7% lo hicieron con motivos turísticos (Lucía et al., 2020). Estos datos respaldan la consideración del turismo como una estrategia de desarrollo local, especialmente en territorios en fases iniciales de desarrollo. En estas comunidades, el turismo trae consigo mejoras en infraestructuras, instalaciones públicas, inversiones y empleo, lo que incentiva a la población local a emprender nuevos negocios (Page, 2014).

2.2. Aplicaciones móviles

Las aplicaciones móviles, son un tipo de software de aplicación diseñado para ejecutarse en un dispositivo móvil como un smartphone o tablet. Las aplicaciones móviles, con frecuencia sirven para proporcionar a los usuarios servicios similares a los que accede en las PC (Personal Computer), aunque generalmente suelen ser unidades de software más pequeñas y limitadas (Rouse, 2020). Estas aplicaciones se desarrollan para diferentes sistemas operativos, como iOS (utilizado en dispositivos Apple) y Android (utilizado en la mayoría de los dispositivos móviles actualmente), pueden ser descargadas e instaladas desde tiendas de aplicaciones, como la App Store¹ de Apple o Google Play Store².

Las aplicaciones móviles ofrecen una amplia variedad de funcionalidades y se utilizan en diversas actividades, como la comunicación en redes sociales, jugar videojuegos, las compras en línea, el acceso a servicios bancarios, el monitoreo de la actividad física, la realización de reservas de viajes, entre otras opciones disponibles. Estas aplicaciones hacen uso de las

¹<https://www.apple.com/la/app-store/>

²<https://play.google.com/>

capacidades y características de los dispositivos móviles, como la cámara, el GPS, los sensores de movimiento y la conexión a Internet, para brindar a los usuarios una experiencia interactiva y personalizada.

Las aplicaciones móviles, con el pasar de los años se han convertido en una parte integral de nuestro día a día, ya que han transformado la forma en la que las personas se comunican, entretienen, y han ayudado a acceder a múltiples servicios de comercio y de la banca. Su conveniencia, versatilidad y capacidad para adaptarse a las necesidades individuales las han posicionado como una herramienta indispensable en la actualidad. A medida que la tecnología móvil continúa avanzando y las aplicaciones se vuelven más sofisticadas, es muy probable que sigan desempeñando un papel fundamental en nuestra sociedad, mejorando la productividad, entretenimiento y conectividad en todos los aspectos de la sociedad.

2.2.1. Frameworks

Dentro del desarrollo de aplicaciones móviles, existen numerosos frameworks que facilitan y agilizan el proceso de creación de aplicaciones. Estos frameworks son conjuntos de herramientas, librerías y recursos que permiten a los desarrolladores construir aplicaciones de manera más eficiente y efectiva. Al proporcionar una base sólida y ya establecida, estos frameworks simplifican tareas comunes, como el diseño de interfaces de usuario, la gestión de datos y la integración de funcionalidades específicas del dispositivo. A continuación, se explorará algunos de los frameworks más populares en el desarrollo de aplicaciones móviles y se analiza sus características y beneficios.

Flutter

Flutter es un framework de código abierto desarrollado por Google que permite crear aplicaciones móviles nativas para iOS y Android utilizando un solo código base. Una de las principales ventajas de Flutter es su capacidad para construir interfaces de usuario atractivas y fluidas, gracias a su enfoque en la creación de interfaces mediante widgets personalizables y flexibles. Además, Flutter ofrece un rendimiento de alta velocidad al aprovechar la compilación en código nativo, lo que resulta en aplicaciones rápidas y receptivas (IONOS, 2020).

Otra ventaja de Flutter es su hot reload, una función que permite a los desarrolladores ver los cambios realizados en el código en tiempo real, lo que agiliza significativamente el proceso de desarrollo y facilita la depuración. Asimismo, Flutter cuenta con una amplia gama de paquetes y bibliotecas que ofrecen una variedad de funcionalidades y características listas para usar (F. Dev, 2018).

Sin embargo dado que es un framework relativamente nuevo, la curva de aprendizaje inicial es alta para los desarrolladores. Finalmente, aunque Flutter permite crear aplicaciones nativas, en algunos casos puede haber limitaciones en el acceso a ciertas características y APIs (Application Programming Interface) específicas de cada plataforma (IONOS, 2020).

Xamarin

Xamarin es un framework de desarrollo de aplicaciones móviles creado por Microsoft, que permite crear aplicaciones nativas para múltiples plataformas utilizando un lenguaje de programación compartido. Xamarin ofrece ventajas significativas en el proceso de desarrollo y en la creación de aplicaciones multiplataforma (X. Dev, 2016).

Al igual que Flutter, Xamarin puede compartir código entre diferentes plataformas. Al utilizar C# como lenguaje de programación compartido, los desarrolladores pueden escribir una vez el código y compartirlo entre iOS, Android y otras plataformas compatibles, ahorrando tiempo y esfuerzo en el desarrollo, ya que se evita tener que escribir y mantener código separado para cada plataforma. Xamarin cuenta con una comunidad activa de desarrolladores y una amplia gama de herramientas y recursos de soporte. Además, Microsoft ofrece una integración sólida con su suite de desarrollo, lo que facilita el desarrollo y la depuración de aplicaciones Xamarin. Sin embargo, una posible desventaja de Xamarin es que requiere conocimientos de C# y del ecosistema de desarrollo de Microsoft, esto puede ser una barrera para los desarrolladores que no están familiarizados con estas tecnologías (James, 2022).

React Native

React Native es un framework de desarrollo de aplicaciones móviles que permite crear aplicaciones nativas para iOS y Android utilizando JavaScript y React. Desarrollado por Facebook, React Native ofrece varias ventajas para el desarrollo de aplicaciones multiplataforma (R. Dev, 2020). Al igual que Flutter y Xamarin, una de las principales ventajas de React Native es su capacidad para compartir código entre plataformas. Utiliza JavaScript como lenguaje de programación, los desarrolladores pueden escribir una vez el código y reutilizarlo en Android e iOS, lo que agiliza el proceso de desarrollo y reduce los costos. Además, React Native permite el acceso directo a las API nativas de cada plataforma, lo que proporciona un alto rendimiento y una apariencia y experiencia de usuario nativa.

Otra ventaja de React Native es su enfoque en la interfaz de usuario declarativa. Utilizando componentes reutilizables, React Native facilita la creación de interfaces de usuario interactivas y modernas. Además, cuenta con una amplia comunidad de desarrolladores y una amplia

gama de bibliotecas y componentes disponibles, lo que agiliza el desarrollo y permite agregar rápidamente funcionalidades a las aplicaciones.

Sin embargo, una posible desventaja de React Native es que algunas funcionalidades avanzadas o específicas de una plataforma pueden requerir el desarrollo de módulos nativos personalizados, esto implica tener que escribir código nativo para cada plataforma, lo que puede requerir conocimientos adicionales y tiempo adicional de desarrollo (AltexSoft, 2018).

Para el desarrollo de la aplicación en el marco de este trabajo, se ha decidido utilizar Flutter debido a las numerosas ventajas que ofrece. Flutter, con su enfoque en el desarrollo rápido y eficiente de aplicaciones multiplataforma, permite crear interfaces de usuario atractivas y fluidas, aprovechando la potencia del lenguaje de programación Dart. Además, el hecho que los autores cuentan con experiencia previa en el uso de Flutter agilizará el proceso de desarrollo y permitirá aprovechar al máximo las capacidades y características de este framework. Su naturaleza de código abierto, su amplia comunidad de desarrolladores y el respaldo continuo por parte de Google también son aspectos que brindan confianza en la elección de Flutter como la herramienta principal para el desarrollo de esta aplicación.

2.3. Web Semantica y Ontologias

En la actualidad, la cantidad de información disponible en la web es enorme y crece a un ritmo exponencial, ya que es un espacio diseñado para el intercambio de información. No existe un formato común para mostrar la información, por lo cual, los desarrolladores crean sus páginas web basados en sus potenciales clientes (A. Lozano, 2020). Por lo que, encontrar y acceder a esta información de manera eficiente y efectiva sigue siendo un desafío importante, haciendo que se gaste mucho tiempo en seleccionar la información que puede ser útil. Bajo este contexto, la web semántica surge como una solución para facilitar la organización y búsqueda de información en la web, ya que permite a los usuarios y máquinas procesar, interpretar y comprender la información alojada en la web de manera más efectiva.

Según (A. Lozano, 2020) para que lo mencionado anteriormente pueda llevarse a cabo, se necesita que el conocimiento de la web sea representado de forma que sea legible para un ordenador, esté consensuado y sea reutilizable, esta vía para representar conocimiento lo proporciona las ontologías.

El término ontología proviene de la filosofía, pero en IA tiene diferentes connotaciones, la definición más consolidada menciona que “una especificación explícita y formal sobre una conceptualización compartida” (Pérez, 2002), dicho en otras palabras las ontologías definen

conceptos y relaciones de algún dominio de forma compartida y consensuada, permitiendo a los usuarios compartir y disentir sobre el conocimiento representado, lo que facilita la interoperabilidad y la reutilización de la información en diferentes contextos. Las ontologías tienen los siguientes componentes para representar el conocimiento de algún dominio:

Conceptos: Son las entidades o términos que representan objetos, eventos, procesos o cualquier cosa que se pueda modelar del mundo real.

Relaciones: Representan la interacción y enlace entre los conceptos del dominio, se establece como su relación entre sí en el mundo real.

Funciones: Son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos en la ontología.

Instancias: Se utilizan para definir objetos de un concepto.

Axiomas: Son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Permiten junto con la herencia de conceptos, inferir conocimiento que no esté indicado explícitamente en la taxonomía de conceptos.

Además de las ontologías, para poder explorar la web semántica, se necesitan lenguajes de mercado apropiados que representan el conocimiento de las ontologías como:

RDF: Resource Description Framework (RDF) es una base para procesar datos, proporciona interoperabilidad entre aplicaciones que intercambian información legible por máquina en la web, permitiendo representar propiedades y valores a propiedades, permitiendo crear una jerarquía de conceptos (W3C, 2004).

RDFS : RDF Schema (RDFS) Es un lenguaje de esquema para RDF que permite a los usuarios definir clases, propiedades y restricciones para los recursos. Permite a los usuarios crear un vocabulario común que se puede utilizar para describir los recursos y sus relaciones, proporcionando una forma de estructurar los recursos y sus propiedades, lo que ayuda a mejorar la interoperabilidad entre las aplicaciones y el sistema (W3C, 2023).

OWL: Es una extensión del lenguaje de descripción de recursos (RDF) y del esquema de RDF (RDFS), el cual proporciona una semántica más rica y expresiva en la modelación de ontologías. Permite la definición de ontologías con una mayor complejidad que RDF y RDFS, al permitir la creación de restricciones y reglas de inferencia más complejas (Bechhofer, 2009).

El modelo de ontología se utiliza para representar información relacionada entre varios dominios, con el objetivo de facilitar la comunicación y el intercambio de información entre diferentes

sistemas y entidades. Se han realizado varios estudios e investigaciones que han generado la creación de varias ontologías relacionadas con el dominio del turismo. A partir de la aplicación de las ontologías al dominio del turismo, se ha impulsado el modelado y la estandarización de datos en el turismo, dando lugar al desarrollo de diversas ontologías turísticas públicas y privadas, presentándose algunas ontologías como base para el desarrollo de ontologías más específicas, y otras como solución final a la representación de problemáticas más concretas, gracias a su estado de madurez avanzado (Losada, 2010).

2.3.1. SPARQL

SPARQL (SPARQL Protocol And RDF Query Language) es un lenguaje de consulta para RDF, que permite a los usuarios expresar consultas sobre diversas fuentes de datos, si los datos se almacenan de forma nativa como RDF o son mapeados a RDF a través de algún sistema middleware. SPARQL contiene las capacidades para la consulta de los patrones obligatorios y opcionales de grafo, junto con sus conjunciones y disyunciones. Además, SPARQL permite ampliar o restringir el alcance de las consultas especificando los grafos en los que se deben operar. Los resultados de las consultas SPARQL pueden ser conjuntos de resultados o grafos RDF. SPARQL es el protocolo estándar para recuperar Linked Data de la web o de las tiendas triples RDF (W3C, 2009).

Una consulta SPARQL puede adoptar una de las cuatro formas existentes: SELECT, DESCRIBE, CONSTRUCT o ASK. Estas consultas utilizan un conjunto de patrones triples compuestos por un sujeto, un predicado y un objeto. Cualquier variable en estos patrones puede ser reemplazada por un comodín para obtener un resultado específico (Hernández, 2018).

En la Figura 2.1, se muestra un ejemplo donde se busca el nombre y apellido de la persona asociada al número de teléfono "(229) 276-5135".

```
PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?first ?last
WHERE
{
  ?person ab:homeTel "(229) 276-5135" .
  ?person ab:firstName ?first .
  ?person ab:lastName ?last .
}
```

Fig 2.1: Consulta Sparql (DuCharme, 2013)

2.3.2. Ontologías en el Turismo

El turismo, una industria que se caracteriza por su constante necesidad de información, ha experimentado un cambio radical en la forma en que se recopila y se comparte dicha información desde la llegada de la Web. Este cambio ha sido un fenómeno generalizado en diversos ámbitos, incluido el turismo. A medida que la sociedad ha adoptado cada vez más las tecnologías digitales, se ha generado una gran cantidad de datos relacionados con los destinos turísticos, servicios, opiniones de usuarios y más (Prantner et al., 2007).

Con el fin de gestionar eficientemente esta vasta cantidad de información y promover la interoperabilidad entre diferentes sistemas y aplicaciones, se introdujo la web semántica y las ontologías en el campo del turismo. Estas ontologías proporcionan un marco común de entendimiento y un lenguaje compartido para describir y representar los elementos clave del turismo, al permitir una mejor gestión y estructuración de la información, las ontologías facilitan la integración de datos de diferentes fuentes y promueven la colaboración entre actores del sector turístico. Al utilizar ontologías en el turismo, se logra una mayor precisión y coherencia en la representación de la información, facilitando la búsqueda, el análisis y la recuperación de datos relacionados con el sector. Además, las ontologías permiten la integración de diferentes fuentes de datos y la generación de conocimiento a partir de la información turística disponible (Sir et al., 2015).

Según (Lee et al., 2017) entre las aplicaciones prácticas de las ontologías en el turismo se encuentran: la planificación de viajes, la recomendación de destinos turísticos, la personalización de experiencias, la gestión de contenidos turísticos y la mejora de los sistemas de información turística.

2.4. Arquitectura orientada a microservicios

La arquitectura basada en microservicios es un enfoque para diseñar aplicaciones de software como un conjunto de servicios pequeños e independientes, cada uno con su propia lógica de negocio y comunicación a través de interfaces. Cada uno de estos servicios es independiente y debe implementar una funcionalidad de negocio individual dentro de un contexto delimitado. Esta arquitectura promueve la escalabilidad, flexibilidad y eficiencia en el desarrollo de software, al permitir que los servicios se desarrollen, desplieguen y actualicen de forma independiente. Por ejemplo, un microservicio puede representar el inventario, otro la gestión de pedidos y otro el envío, pero juntos pueden constituir un sistema de comercio electrónico completo. Los microservicios son una opción de arquitectura que se enfoca en brindarle muchas opciones

para resolver los problemas que se pueden enfrentar (Newman, 2015).

Desde el exterior, un solo microservicio se trata como una caja negra. Aloja la funcionalidad comercial en uno o más puntos finales de la red, por ejemplo en un API REST (Representational State Transfer). Los consumidores, ya sean otros microservicios u otros tipos de programas, acceden a esta funcionalidad a través de estos puntos finales en red. Los detalles de implementación internos como la tecnología en la que está escrito el servicio o la forma en que se almacenan los datos están completamente ocultos al mundo exterior. Al utilizar protocolos ligeros como REST o RPC (Remote Procedure Call) se permite una comunicación ágil y eficiente entre los servicios, además la utilización de diferentes tecnologías y lenguajes de programación para cada microservicio puede aumentar la eficiencia y la calidad de cada servicio individual (Newman, 2015).

Otra gran ventaja de los microservicios es el escalado, es decir la capacidad de ajustar el tamaño y la cantidad de instancias de los microservicios para manejar la carga y la demanda. El escalado puede ser vertical, aumentando los recursos de hardware en un servidor; también puede ser horizontal, agregando más instancias distribuidas en múltiples servidores (Newman, 2015). El escalado horizontal es más flexible y escalable, y se puede lograr mediante el uso de contenedores y herramientas de orquestación. Herramientas como kubernetes³ y Docker⁴ son ampliamente utilizadas para la orquestación y gestión de contenedores en entornos de producción. Estas herramientas permiten adaptar el sistema a medida que crece y optimizar el rendimiento según las necesidades, brindando una mayor eficiencia y capacidad de respuesta (Cargo, 2023).

Actualmente la flexibilidad de implementación y el aumento de las opciones de implementación nativas de la nube sin servidor y su función como servicio (AWS Lambda, Microsoft Azure Cloud Functions, IBM Cloud Functions, Google Cloud Functions, etc.), crearon el entorno perfecto para que los microservicios prosperen en el panorama actual de las tecnologías de la información. Estas plataformas permiten que los microservicios y las funciones escalen de la inactividad a un gran volumen y viceversa, mientras que los clientes pagan sólo por lo que utilizan.

³Kubernetes es una plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios.

⁴Docker es una plataforma que permite empaquetar y ejecutar aplicaciones de manera aislada, proporcionando portabilidad y escalabilidad a través de contenedores.

2.4.1. Spring Boot

Para el desarrollo de los microservicios de este proyecto, se ha optado por utilizar Spring Boot (Spring boot Dev, 2023) debido a sus numerosas ventajas y características. Spring Boot es un framework de desarrollo de aplicaciones basado en Java que facilita la creación rápida y eficiente de microservicios escalables y robustos. Una de las principales ventajas de Spring Boot es su enfoque en la configuración automática, lo que permite ahorrar tiempo y esfuerzo (Tanzu, 2023). Además, Spring Boot ofrece una amplia gama de funcionalidades integradas, como gestión de dependencias, seguridad, control de transacciones y conectividad con bases de datos, lo que simplifica el desarrollo y mejora la productividad. Gracias a su arquitectura basada en módulos, Spring Boot facilita la modularización y el mantenimiento de los microservicios. (Tanzu, 2023).

2.5. OAuth 2

OAuth2 es un protocolo de autenticación y autorización que permite a los usuarios compartir su información en una aplicación sin compartir sus credenciales de inicio de sesión (Hardt, 2012), permitiendo a los usuarios iniciar sesión en una aplicación con sus credenciales de Google. El marco teórico de OAuth2 se basa en cuatro roles principales: el propietario del recurso, el servidor de recursos, el servidor de autorización y el cliente. El propietario del recurso es el usuario que posee los datos protegidos, el servidor de recursos es el servidor que aloja los recursos protegidos, el servidor de autorización es el servidor que emite tokens de acceso y el cliente es la aplicación que solicita acceso a los recursos protegidos. OAuth2 utiliza tokens de acceso en lugar de credenciales de inicio de sesión para proteger la información del usuario. El protocolo también utiliza flujos de autorización para permitir que los clientes obtengan tokens de acceso en nombre del usuario (Moreno Reyes, 2015).

2.6. Scrum

Scrum es una metodología de desarrollo ágil de gestión de proyectos y se define como la gestión de procesos y equipos para abordar proyectos complejos que requieren entornos dinámicos y cambiantes, por lo que exigen rapidez de resultados y flexibilidad, de manera iterativa e incremental. Cada iteración consta de sprints que duran entre 2 a 4 semanas, donde el objetivo es construir primero las características más importantes y generar un producto potencialmente entregable. Se integran más funcionalidades del producto en los

siguientes sprints y estas funcionalidades se ajustan en función de los comentarios de las partes interesadas y los clientes entre sprints. Según información obtenida de (S. Lozano and Anaya, 2014) dentro de la metodología Scrum, se pueden encontrar los siguientes componentes:

Roles.

- **Product Owner:** es la persona responsable de representar los intereses y definir las características y prioridades dentro del proyecto, definiendo los objetivos y garantizando su correcta implementación.
- **Scrum Master:** es la persona encargada de resolver los problemas que puedan surgir dentro del equipo, eliminando obstáculos para que el trabajo sea productivo. Ayuda al equipo y aconseja al Product Owner para maximizar el retorno de la inversión (ROI) en el proyecto.
- **Scrum Team:** Está conformado por el equipo que desarrolla y entrega los proyectos al cliente, los cuales trabajan juntos de manera colaborativa para desarrollar y entregar el producto.

Artefactos.

- **Backlog del Producto:** Lista priorizada de requisitos y características del producto, mantenida por el Product Owner.
- **Backlog del Sprint:** Conjunto de elementos del backlog del producto seleccionados para el sprint actual.
- **Incremento:** Producto funcional y potencialmente entregable al final de cada sprint.

Eventos

- **Sprint Planning:** Reunión para definir el objetivo del sprint y seleccionar elementos del backlog del producto para el backlog del sprint.
- **Daily Scrum:** Breve reunión diaria en la que el equipo revisa el progreso, actualiza el plan y discute los obstáculos. Revisión del Sprint: Demostración del incremento completado al Product Owner y otras partes interesadas.
- **Retrospectiva del Sprint:** Reflexión del equipo sobre el sprint pasado para mejorar su rendimiento en el futuro.

2.7. Pruebas de usabilidad

Las pruebas de usabilidad de software son evaluaciones de la experiencia del usuario que miden la facilidad de uso, eficiencia y satisfacción al interactuar con un producto, las cuales detectan y corrigen problemas en la interfaz y usabilidad antes del lanzamiento. Estas pruebas pueden ser remotas o presenciales, en distintas etapas del desarrollo. Involucran observación del usuario, recopilación de comentarios mediante cuestionarios y entrevistas y ciertas veces medición de tiempos de respuesta y tasa de error. Son esenciales para asegurar que los sistemas de software sean accesibles, eficientes y satisfactorios para los usuarios (Enriquez and Casas, 2013).

Algunas pruebas comunes incluyen:

- **Pruebas de usabilidad basadas en tareas:** consisten en asignar tareas específicas a los usuarios y observar cómo interactúan con el sistema para completarlas.
- **Pruebas de usabilidad basadas en la retroalimentación del usuario:** se centran en la recopilación de comentarios y opiniones de los usuarios finales acerca del sistema. Esta prueba mide la satisfacción del usuario y proporciona información valiosa para mejorar la experiencia de usuario.
- **Pruebas de usabilidad basadas en la observación:** se llevan a cabo mediante la observación directa de los usuarios mientras interactúan con el sistema. Esta prueba permite evaluar cómo los usuarios utilizan el sistema y detectar posibles problemas de usabilidad.

2.8. Trabajos relacionados

Los viajes y el turismo han sufrido un crecimiento exponencial en la última década, este incremento se debe a varios factores, siendo la digitalización el más importante. La digitalización ha llevado la experiencia de viajar a un nuevo nivel, no solo para planificar vacaciones, sino también para los negocios turísticos. Este avance tecnológico ha sido facilitado gracias a diversas aplicaciones móviles de turismo que permiten al usuario preparar todo su viaje antes de llegar a su destino (Estudio Alfa, 2019). Además, el aumento de la movilidad de los turistas, quienes utilizan dispositivos móviles, ha generado la necesidad de adaptar la distribución de productos y servicios turísticos

reorientándolos hacia el entorno móvil. (Filofteia, 2016). En muchos países alrededor del mundo, se han desarrollado y lanzado varias aplicaciones móviles de turismo, con el objetivo de ofrecer información a los viajeros, estas aplicaciones van desde guías turísticas, mapas interactivos, recomendaciones de lugares, reservas de alojamiento y actividades, restaurantes, visualización de rutas, entre otros. Por lo tanto, los teléfonos móviles se convierten en instrumentos muy utilizados para organizar viajes e integrarse como un canal más de marketing dentro de la estrategia de comunicación de los destinos turísticos.

En Al Falah University - Dubai se realizó un estudio con el objetivo de analizar las reseñas de viajeros en aplicaciones móviles de toma de decisiones en un viaje a Dubai, buscando examinar las relaciones entre las reseñas de los viajeros, las intenciones de visita, la imagen del destino y la elección del destino. De los datos recogidos de 892 encuestados, indican que las reseñas de los viajeros en las aplicaciones móviles influyen positivamente en la elección de Dubai como destino turístico. Además se encontró en el estudio que estas reseñas tienen un impacto significativo en la imagen del destino y las intenciones de la visita en Dubai. Terminando con la sugerencia al gobierno de Dubai y el Consejo Nacional de la Industria Turística en desempeñar un papel activo en la creación de aplicaciones móviles turísticas, que contengan un espacio para la integración y el intercambio de información entre los propios turistas (Jahmani et al., 2020).

En Rumania, se realizó un estudio indicando el perfil del turista-usuario de dispositivos móviles, captando tres niveles, a saber: un perfil sociodemográfico, un perfil de movilidad y un perfil de usuario de aplicación móviles relacionadas con el turismo, además obteniendo que de 905 sujetos existe un 36.97% de usuarios que no utilizan aplicaciones móviles de turismo, los cuales se apoyan sobre la falta de preparación de las organizaciones turísticas de Rumanía para adaptar su oferta al entorno móvil. Con el pasar del tiempo, los proveedores se han enfocado exclusivamente en las ofertas tradicionales de sitios web, sin siquiera actualizar las páginas a versiones responsive y mucho menos adaptarlas a dispositivos móviles o en su defecto, proporcionar aplicaciones móviles dedicadas a los clientes potenciales o existentes (Filofteia, 2016).

Los destinos turísticos están adaptando cada vez más el uso de aplicaciones móviles como parte fundamental de su producto de destino y experiencia global para los

turistas. En la actualidad, se consideran a las aplicaciones móviles turísticas como parte indispensable de la oferta turística global. Las aplicaciones móviles de turismo ofrecen beneficios significativos a los turistas, como la capacidad de planificar itinerarios, compartir impresiones y acceder a contenidos multimedia relacionados con el destino. Sin embargo, hay algunas desventajas, como el limitado número de idiomas disponibles de la aplicación, la falta de control sobre el contenido multimedia y la necesidad constante de actualizaciones para evitar la obsolescencia de la aplicación. Belgrado, como destino turístico, ha reconocido tempranamente la importancia de desarrollar aplicaciones turísticas para gestionar el destino y mejorar la experiencia general de los turistas durante su visita a este lugar (Štetić et al., 2021).

Siguiendo el ejemplo de naciones desarrolladas y aprovechando el progreso tecnológico, en Ecuador se han creado aplicaciones con enfoque en el sector turístico. Go UIO es una de las más relevantes en el país, esta aplicación buscaba aumentar las visitas de turistas en la ciudad de Quito integrando restaurantes, hoteles, lugares y eventos que se pueden realizar en la Carita de Dios (Encalada, 2018). En el año 2018, la aplicación Go UIO fue nominada como una de las mejores aplicaciones turísticas del mundo (Soto, 2019). Sin embargo, en la actualidad, dicha aplicación ya no está disponible en las tiendas de aplicaciones de Google Play y App Store. Además, la plataforma web que compartía la misma información con la aplicación móvil ya no está disponible para acceder.

Para la ciudad de Guayaquil, se han desarrollado varias aplicaciones; entre ellas, se tiene a “Guayaquil es mi destino”, “Gye Minube”, “Conoce Guayaquil” y “GYA guía turística”, sin embargo, poseen las mismas falencias que en muchas aplicaciones de este tipo, carecen de contenido informativo y poseen errores en diversos aspectos de sus funcionalidades (Jiménez and Ripoll, 2017).

En el caso de la ciudad de Cuenca, la aplicación más relevante desarrollada para el sector turístico ha sido “Visit Cuenca”, esta aplicación con el apoyo del Municipio de Cuenca, prometía ser una oportunidad para que todo el sector turístico cuencano pueda mostrar la diversidad turística y sus diferentes espacios de la ciudad con todos sus atributos de manera fácil y gratuita, además de contar con información detallada de hoteles, restaurantes, museos, una agenda de eventos, iglesias, rutas turísticas y guías en 3D con realidad aumentada y edificios inteligentes (Cuenca, 2019). La aplicación está disponible en Google Play, sin embargo, a la fecha en la que se escribe este documento la aplicación no funciona, no muestra ningún tipo de recursos, su última actualización fue

en 2019 y la gran mayoría de calificaciones son negativas.

En total, en Ecuador se han registrado 150 apps de acuerdo con la plataforma Google Play, siendo Pichincha (Quito) una de las provincias con mayor desarrollo de Apps. Estas aplicaciones son ampliamente desconocidas por los turistas, así como por los prestadores de servicios. Por lo que, se puede evidenciar una falta de cultura digital por parte de los prestadores de servicios y de las autoridades locales, quienes aún no son conscientes de los beneficios que conlleva la adopción de nuevas tecnologías para el sector turístico (Alejandro et al., 2022).

Como puede observarse, actualmente no existe una aplicación móvil de referencia que brinde información sobre los sitios turísticos de manera completa y actualizada. Esta falta de una solución digital integral dificulta a los turistas y viajeros obtener información precisa sobre los destinos y recursos turísticos disponibles en el país.

En este proyecto, se busca solucionar esta problemática mediante el desarrollo de una aplicación móvil. Esta aplicación registrará y mostrará recursos turísticos georreferenciados, centrándose en dos parroquias de la ciudad de Cuenca: Baños y Sayausi. Para lograrlo se utilizará una ontología como almacén de información que permitirá organizar y estructurar los datos relacionados con los recursos turísticos de estas zonas georreferenciadas. Los usuarios podrán acceder a esta información detallada y navegar fácilmente a través de un mapa interactivo. Además, se implementarán características como rutas personalizadas, búsquedas geolocalizadas y navegación offline, con el objetivo de facilitar la planificación de los viajes y exploraciones. La navegación offline resulta especialmente útil, ya que las parroquias rurales de Baños y Sayas pueden presentar áreas con falta de cobertura y/o conexión a internet estable.

Sin embargo, es fundamental abordar los desafíos asociados con la precisión y actualización de la información en las aplicaciones móviles de turismo. Considerando estos aspectos, la aplicación móvil propuesta a la cual se le dio el nombre de Turis Up permite a los usuarios subir su propio contenido, controlando el problema de contenidos no deseados, ya que contará con un aplicativo web destinado a la gestión de los recursos turísticos. Dicho aplicativo web permitirá a los administradores ingresar, modificar y eliminar información sobre los recursos subidos por los administradores o los usuarios, lo cual permite tener control sobre el contenido y la información que ofrece la aplicación, asegurando así la actualización y calidad de los datos presentados en la aplicación móvil.

3. Desarrollo del Prototipo

En esta sección, se presentará de forma detallada la metodología y el desarrollo de la aplicación móvil y web, incluyendo tanto el frontend como el backend, con el objetivo principal de proporcionar un enfoque sistemático que permita comprender los pasos y procesos involucrados en la creación de estas aplicaciones. Se comenzará describiendo la planificación inicial, que incluye la definición de los objetivos y requerimientos del proyecto. A continuación, se abordará el diseño de la interfaz de usuario, donde se explicará el enfoque adoptado, las decisiones de diseño y las herramientas utilizadas para intentar crear una experiencia de usuario intuitiva y atractiva. En cuanto a la selección de tecnologías, se detallarán las diferentes opciones consideradas y las razones por las cuales se optó por las tecnologías utilizadas en el desarrollo de la aplicación. Posteriormente, se profundizará en la arquitectura de la aplicación, destacando la estructura general, los patrones de diseño empleados y la integración de los componentes clave. También se describirán las funcionalidades específicas implementadas, brindando una visión detallada de cómo se desarrollaron y se integran en la aplicación. Además, se explicará en detalle el proceso de desarrollo de la ontología, que juega un papel fundamental en la organización y representación del conocimiento dentro de la aplicación. Finalmente, se abordará la implementación de los microservicios en el backend, explicando cómo se diseñaron, desarrollaron y se integran en la arquitectura global de la aplicación. Esta sección será una guía completa que permitirá comprender de manera exhaustiva el proceso de desarrollo de la aplicación móvil desde una perspectiva técnica.

3.1. Metodología

La metodología utilizada para el desarrollo del proyecto fue Scrum, una metodología ágil de gestión de proyectos que se centra en la colaboración, la entrega iterativa e incremental, y la adaptación a los cambios en el proceso de desarrollo. A continuación, se describen los pasos que se siguieron durante la ejecución del proyecto.

Formación del equipo Scrum

Se conformó un equipo de desarrollo compuesto por el Product Owner, el Scrum Master y el Development Team, tal como se detalla en la Tabla 3.1.

Roles Scrum	
Scrum Master	Victor Saquicela
Product Owner	Victor Saquicela
Development Team	Ricardo Jarro, Juan Lazo

Table 3.1: Roles Scrum

Generación del Product Backlog

El Product Backlog consiste en una lista de funcionalidades y requisitos necesarios para el desarrollo de la aplicación móvil de turismo, con el objetivo de cumplir con las necesidades del cliente y del mercado. Para este apartado, se llevaron a cabo reuniones entre el Development Team y el Scrum Master, con el fin de definir la creación de una aplicación móvil para mostrar y registrar recursos turísticos, una aplicación web para gestionar dichos recursos, y un sistema que permita la gestión integrada de ambas aplicaciones mediante el almacenamiento de información en una ontología. Para cumplir con estas necesidades, se definieron las épicas, las cuales se detallan en la Tabla 3.2.

Después de definir las épicas, se procedió a desglosarlas en tareas más pequeñas y específicas. Estas tareas fueron registradas como elementos en el Backlog, representando las unidades de trabajo que el equipo de desarrollo abordaría durante los sprints. Todas estas tareas están disponibles en la sección de Anexos. Cabe destacar que cada tarea cuenta con su respectivo número de prioridad, que se puede visualizar en la Tabla 3.3. El Backlog fue una herramienta visual que facilitó el seguimiento y la gestión de las actividades del proyecto.

Al descomponer las épicas en tareas manejables y tenerlas registradas en la tabla, se logró una mayor transparencia y coordinación en el equipo de desarrollo. Esto a su vez facilitó la planificación y el progreso constante hacia los objetivos del proyecto, ya que se pudo tener una clara visión de las tareas que se debían abordar en cada sprint.

Código	Nombre	Descripción
T01	Análisis de Requisitos	Como desarrollador, este recopilará información, definirá funcionalidades y atributos, priorizará los requisitos y documentará de manera clara, para tener una guía clara durante el desarrollo del sistema y asegurar la correcta implementación de las funcionalidades requeridas.
T02	Desarrollo de la ontología	Como desarrollador este revisará e implementará de la metodología NEON
T03	Desarrollo de microservicios	Como desarrollador, este tendrá que configurar e implementar los microservicios cada uno con diferentes funcionalidades
T04	Aplicación móvil	Como usuario, este necesitará poder registrar, ver y comentar los recursos turísticos.
T05	Aplicación web	Como administrador, existirá la necesidad de revisar los recursos turísticos, modificarlos de ser necesario, o eliminarlos definitivamente

Table 3.2: Épicas

Prioridad	Valoración
Baja	1
Media	2
Alta	3
Muy alta	4

Table 3.3: Valoración de prioridad

Planificación

Durante el proyecto, se adoptó una planificación de sprint que se llevó a cabo cada dos semanas. En total, se realizaron 18 sprints a lo largo del proceso de desarrollo. El proyecto comenzó el 15 de agosto del 2022 y concluyó el 30 de mayo del 2023, abarcando un período de 35 semanas en total. Esta planificación estructurada y regular permitió un ritmo constante de trabajo y aseguró una gestión eficiente del tiempo y los recursos disponibles.

De igual manera, se estableció un calendario de reuniones con el Scrum Master. Estas reuniones se programaron para llevarse a cabo los días martes, cada dos semanas al finalizar los sprints. Durante estas sesiones, se revisaban los avances del sprint, se realizaban demostraciones de los entregables desarrollados y se discutían los próximos pasos a seguir. Estas reuniones proporcionaron un espacio regular para la comunicación y la colaboración, permitiendo una supervisión efectiva del progreso y asegurando la entrega oportuna de los entregables en cada iteración del sprint. Las tareas realizadas en cada sprint están enumeradas en la sección de Anexos. Este anexo proporciona una lista detallada de las actividades abordadas en cada iteración, permitiendo una referencia clara y concisa de las tareas realizadas en el proyecto.

Herramientas y lenguajes de programación utilizados

En la Tabla 3.4, se presenta un desglose exhaustivo de todas las herramientas y lenguajes de programación utilizados en el proyecto, proporcionando una visión general de las tecnologías empleadas, incluyendo los frameworks, bibliotecas, bases de datos y otras herramientas utilizadas para el desarrollo. Además, se detallan los lenguajes de programación específicos utilizados en cada componente del sistema.

Herramienta	Descripción
Figma	Herramienta de diseño de UI, wireframes, prototipos interactivos para aplicaciones web y móviles, permite tener una guía clara y visual de lo que se espera ver en la aplicación y sus funcionalidades en términos de diseño.
Protégé	Software para la creación y modificación de ontologías y la realización de inferencias en lógica descriptiva y lógica de reglas.

Visual Studio Code	Editor de código fuente gratuito y de código abierto, usado para el desarrollo de aplicaciones en la mayoría de lenguajes conocidos.
Android Studio	Entorno de desarrollo integrado (IDE), utilizado para el desarrollo de aplicaciones móviles.
Flutter	Framework de código abierto desarrollado por Google, utilizado para el desarrollo de aplicaciones móviles multi-plataforma.
React	Librería de JavaScript utilizada para construir interfaces de usuario interactivas y reactivas. Permite crear componentes reutilizables y manejar eficientemente los cambios en los datos, facilitando el desarrollo de aplicaciones web rápidas y eficientes.
Springboot	Framework de Java que simplifica el desarrollo rápido de aplicaciones web y microservicios, con configuración automática y fácil integración con otros proyectos de Spring.
PostgreSQL	Base de datos relacional de código abierto, escalable y confiable, con características avanzadas y amplia adopción en aplicaciones empresariales.
Fuseki	Servidor SPARQL de código abierto que permite almacenar y consultar datos RDF de manera eficiente.
Docker	Docker es una plataforma de contenedores de código abierto que permite empaquetar, distribuir y ejecutar aplicaciones de manera eficiente y reproducible. Proporciona un entorno aislado y ligero para ejecutar aplicaciones y sus dependencias en cualquier entorno, lo que facilita la implementación y escalabilidad de aplicaciones

GIT	Sistema de control de versiones distribuido, diseñado para gestionar eficientemente el desarrollo de proyectos de software. Permite realizar un seguimiento de los cambios en los archivos a lo largo del tiempo
Github	Plataforma de desarrollo colaborativo basada en la nube que utiliza el sistema de control de versiones Git. Permite alojar repositorios de código, facilitando la colaboración y el trabajo en equipo en proyectos de software.
Dart	Lenguaje de programación de código abierto desarrollado por Google. Es un lenguaje orientado a objetos con características modernas, como inferencia de tipos, asincronía y programación reactiva, fácil de aprender y adecuado para la creación de aplicaciones en diferentes plataformas.
Java	Lenguaje de programación de propósito general y orientado a objetos ampliamente utilizado en el desarrollo de aplicaciones empresariales, móviles y web.
JavaScript	Lenguaje de programación interpretado y orientado a objetos, ampliamente utilizado para crear aplicaciones web interactivas y dinámicas. Es compatible con la mayoría de los navegadores web y permite agregar interactividad a páginas HTML, manipular el contenido de una página y responder a eventos del usuario.
Maven	Herramienta de gestión de proyectos para Java que facilita la construcción, el empaquetado y la gestión de dependencias de software. Proporciona una estructura y configuración estándar para organizar el código fuente y los recursos del proyecto.

Node Package Manager (NPM)	Gestor de paquetes por defecto para el entorno de ejecución de JavaScript. NPM permite la instalación y gestión sencilla de paquetes de código reutilizables, que incluyen bibliotecas, frameworks y herramientas.
Postman	Herramienta popular y poderosa utilizada en el desarrollo y pruebas de APIs
SoapUI	Herramienta para realizar pruebas y consumos de servicios RESTful y SOAP, además de pruebas de estrés.

Table 3.4: Herramientas y lenguajes de programación utilizados

3.2. Análisis de Requisitos

El análisis de requisitos se llevó a cabo mediante la recolección de funcionalidades de aplicaciones de viajes y turismo que han obtenido cierto grado de éxito en el país. Para este fin, se realizó una investigación basada en un estudio previo realizado por (Lucía et al., 2020), donde se identificaron 150 aplicaciones relacionadas con viajes y turismo disponibles en Google Play. De estas aplicaciones, se seleccionaron las 10 más descargadas para su análisis, como se muestra en la Tabla 3.5. Lamentablemente, se encontró que algunas de estas aplicaciones ya no estaban disponibles en la plataforma o presentaban problemas de funcionamiento.

Nombre de la aplicación	Región	Año	Descargas
Mapas de Ecuador Gratis – Navegación sin Internet	Nacional	2018	+10.000
Ecuador Travel Guide	Nacional	2016	+10.000
Cuenca Cultura	Azuay	2017	+5.000
Guayaquil en tu Mano	Guayas	2016	+10.000
Guayaquil es mi Destino	Guayas	2017	+5.000
Mapa de Guayaquil	Guayas	2018	+5.000
Quito App	Pichincha	2018	+10.000
MovilizateUIO	Pichincha	2018	+10.000
Quito en tu Mano	Pichincha	2016	+5.000

Table 3.5: Aplicaciones de turismo con más descargas desarrolladas en Ecuador

Una vez analizadas las aplicaciones que seguían en funcionamiento, se procedió a examinar también aplicaciones turísticas extranjeras como: Civitatis, Minube, España Smart Guide,

España Top Travel, Airbnb, CiclaMadrid Conecta, Visit London Official City Guide, entre otras. Gracias a este análisis, se pudieron identificar de manera clara y precisa algunas funcionalidades esenciales e interesantes que deben estar presentes en la aplicación.

Por otro lado para la aplicación web que tiene como función administrar los recursos turísticos registrados, se optó por requisitos simples y fundamentales para un administrador. De esta manera se tuvieron los siguientes requisitos tanto funcionales como no funcionales.

Requisitos funcionales

1. El usuario deberá iniciar sesión para usar la aplicación móvil o aplicación web.
2. La autenticación y registro para la aplicación móvil será mediante Google y para la aplicación web email y contraseña.
3. Los usuarios normales y administradores podrán crear nuevos recursos geolocalizados con la información necesaria.
4. La aplicación móvil será capaz de mostrar un camino entre dos o más recursos.
5. El usuario podrá crear, eliminar y modificar un conjunto de rutas.
6. La aplicación móvil mostrará un conjunto de recursos disponibles en forma lista o a través de un mapa mostrando su ubicación.
7. El usuario podrá acceder a cualquier recursos mostrado para observar las características e información del mismo.
8. El usuario podrá comentar y puntuar cualquier recurso disponible.
9. El usuario podrá buscar y visualizar la existencia de un camino entre dos o más recursos.
10. El usuario podrá buscar y visualizar desde su posición actual hasta un recurso en específico.
11. El usuario podrá filtrar recursos en base a su categoría.
12. El usuario podrá realizar una búsqueda de recursos.
13. La aplicación móvil proveerá de la funcionalidad para agregar y eliminar recursos a una lista de favoritos.
14. La aplicación móvil podrá mostrar recursos previamente descargados sin la conexión a internet.
15. El usuario podrá crear nuevos recursos sin la conexión a internet.
16. El usuario podrá comentar y puntuar recursos sin la conexión a internet

Requisitos no funcionales

1. La aplicación móvil y web tendrán una interfaz sencilla e intuitiva
2. Se garantizará el acceso a la información de acuerdo al rol del usuario.
3. La aplicación móvil y web deberán contar con manuales de usuario estructurados adecuadamente.

Diagramas

Posteriormente, con el fin de comprender mejor la interacción del usuario con la aplicación móvil y web, se crearon algunos casos de uso que abordaban escenarios particulares. Un ejemplo de esto es el caso de uso del registro de un nuevo usuario en el sistema, que se puede visualizar en las Figuras 3.1 y 3.2. Estos casos de uso proporcionaron una representación clara y detallada de las acciones que un usuario puede realizar en la aplicación.

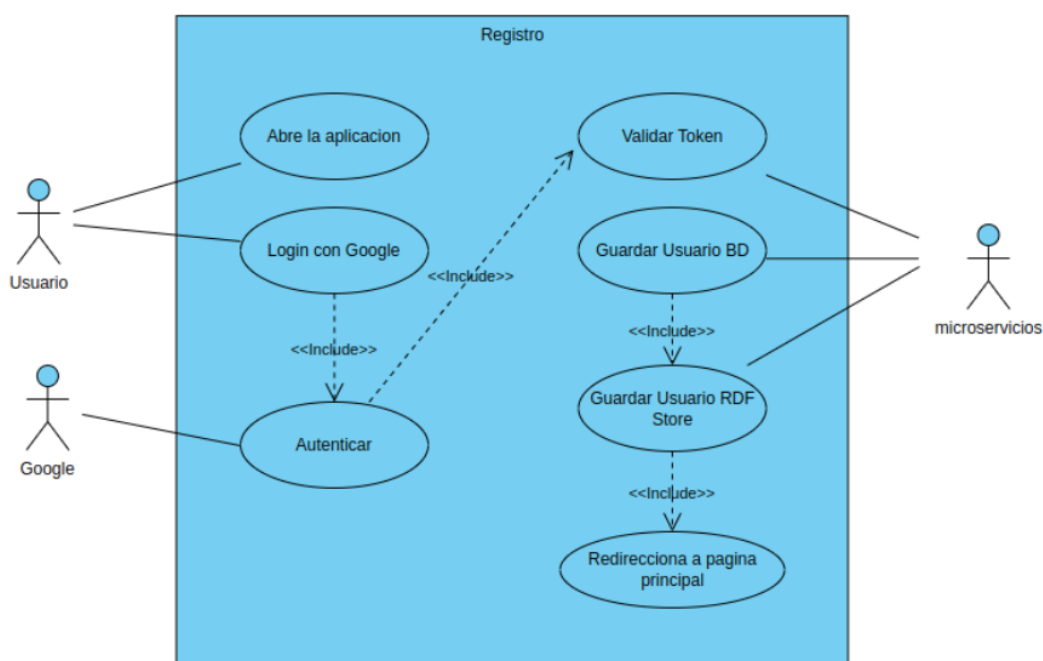


Fig 3.1: Diagrama de caso de uso para registro de usuario

Así mismo se desarrollaron diagramas de secuencia para las actividades más importantes que realizan los usuarios dentro de la aplicación móvil, en la Figura 3.3 se puede observar como es la secuencia a seguir cuando un nuevo usuario es registrado.

Con el propósito de evitar una sobrecarga de información visual en esta sección, se presentarán únicamente los diagramas de caso de uso y de secuencia correspondientes al proceso de registro de usuarios. En caso de requerir una revisión más detallada de otros diagramas

Caso de Uso	Registro	CU1
Actor	Usuarios, Google, microservicios	
Referencia	RF1, RF2, RNF1	
Precondición	ninguna	
Postcondición	El usuario es redirigido a la pantalla principal	
Flujo Normal	<ol style="list-style-type: none"> 1. Usuario abre la aplicación 2. Se ingresa a la pantalla de login de la aplicación 3. Presionar el botón de "login con Google" 4. Autenticación con Google <ol style="list-style-type: none"> 4.1. Valida token otorgado por Google 4.2. Selecciona la cuenta con la que se quiere registrar 4.3. Se crea un nuevo usuario en la BD 4.4. Se crea un nuevo usuario en la RDF store 4.5. Se redirige a la pantalla principal del la aplicación 	
Excepciones	<ol style="list-style-type: none"> 4.1 Excepción guardando el usuario en BD <ol style="list-style-type: none"> 4.1.1 Muestra error, no redirecciona 4.2 Excepción guardando usuario en RDF Store <ol style="list-style-type: none"> 4.2.1 Muestra error y no hace commit en la BD 	

Propósito
Registro de un usuario en el sistema

Resumen
Caso de uso para el registro de un nuevo usuario con una cuenta de Google mediante la aplicación Móvil

Fig 3.2: Especificación de caso de uso para registro de usuario

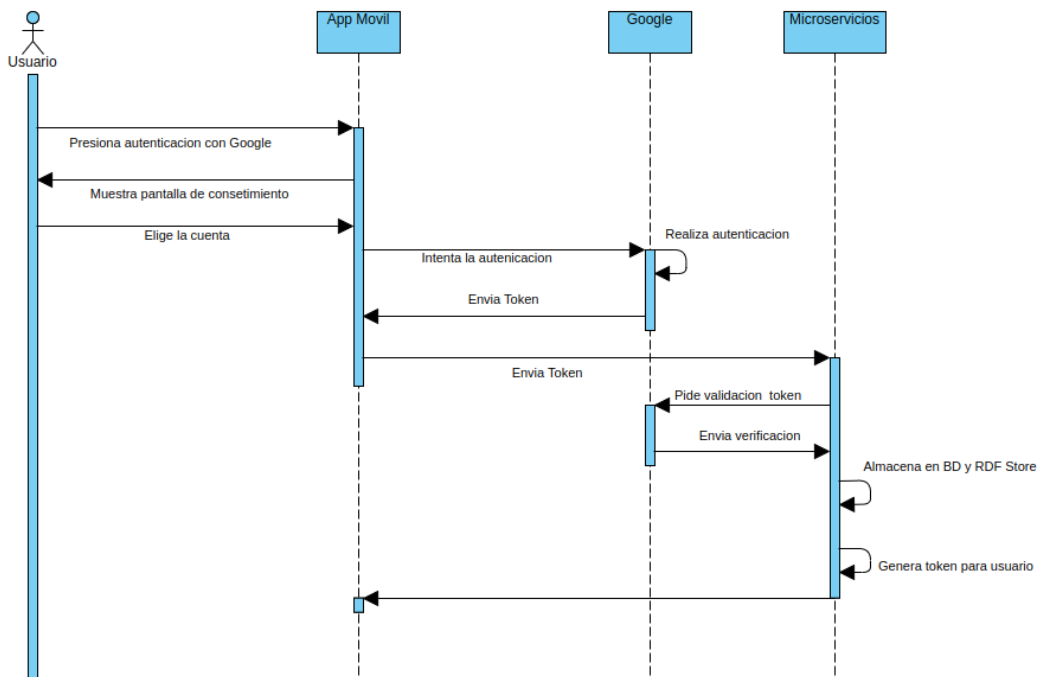


Fig 3.3: Diagrama de secuencia para registro de usuario

relevantes, se sugiere consultar el apartado Anexos adjunto al final del documento, donde se encuentran disponibles todos los diagramas adicionales correspondientes al desarrollo de requisitos.

Finalmente, con una visión mas amplia de las necesidades se pudo desarrollar un diagrama de clases (Figura 3.4), el cual permitió visualizar de manera clara y detallada las clases, atributos, métodos y relaciones además de los componentes del sistema. Con esta representación gráfica, se pudo definir la arquitectura del sistema.

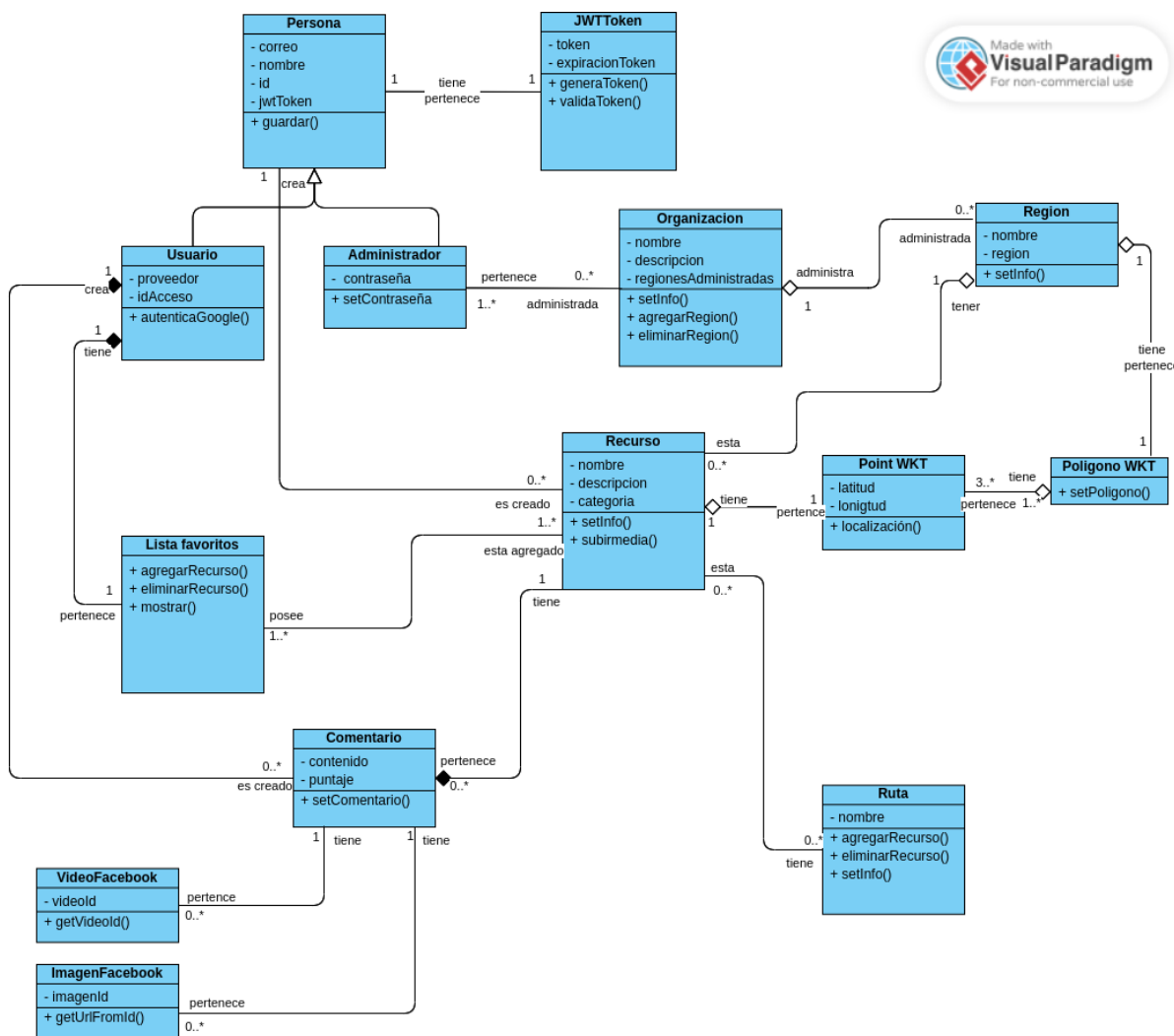


Fig 3.4: Diagrama de clases

3.3. Desarrollo de la ontología

Para abordar el problema de manera formal y estructurada se optó por la utilización de una ontología como herramienta de representación del conocimiento, ya que dispone de múltiples ventajas que menciona (Ontotext, n.d.; Sir et al., 2015) y que se detallan a continuación:

- A diferencia de una base de datos, la ontología no necesariamente se crea desde cero. De hecho, una de las principales ventajas de las ontologías es que se pueden reutilizar ontologías ya existentes para crear una ontología nueva. Esto significa que en lugar de tener que crear todas las clases, propiedades y relaciones desde cero, se pueden tomar como punto de partida ontologías existentes y adaptarlas a las necesidades específicas de un nuevo proyecto o dominio.
- Son útiles para la integración de datos semánticos y la comunicación entre sistemas heterogéneos, así como para compartir una comprensión común de la estructura de la información entre personas y software, lo que resulta especialmente útil en entornos en los que se maneja una gran cantidad de información.
- Al tener las relaciones esenciales entre los conceptos incorporadas, permiten el razonamiento automatizado sobre los datos, funcionan como un "cerebro". Ellos "trabajan y razonan" con conceptos y relaciones de maneras que están cerca de la forma en que los humanos perciben los conceptos interrelacionados.
- Son flexibles y adaptables, ya que su estructura permite agregar fácilmente nuevas relaciones y conceptos a las ontologías existentes. Como resultado, el modelo ontológico puede evolucionar y crecer junto con los datos, sin afectar los procesos y sistemas que dependen de él, incluso en caso de que se necesiten cambios o correcciones.

Para el desarrollo de la ontología, se adoptó la metodología NEON. Esta metodología, desarrollada por (Suárez, 2012), propone un enfoque basado en escenarios para abordar todos los aspectos clave de la ontología. A diferencia de un flujo de trabajo rígido, NEON sugiere rutas y actividades para 9 escenarios diferentes.

Según esta metodología los puntos claves para el desarrollo de la ontología es tener en claro los objetivos del porque se construye la ontología, quienes son los usuarios finales y cuales son los requisitos debe cumplir la ontología. A continuación se detallan estos puntos.

- **Objetivos:** El objetivo principal de la ontología fue crear un modelo de conocimiento en el dominio del turismo, que pudiera ser utilizado por entidades públicas y privadas. Su propósito era proporcionar información georreferenciada sobre recursos turísticos en una determinada zona, definiendo variables y relaciones entre dichos recursos.
- **Usuarios:** En relación a los usuarios de la ontología, se identificaron cuatro grupos principales:
 - * Turistas: Estos usuarios finales buscan información sobre destinos turísticos,

actividades, alojamiento, restaurantes y otros servicios relacionados con el turismo. Necesitan acceder a información precisa y actualizada para planificar sus viajes.

- * Profesionales del turismo: Incluye agentes de viajes, guías turísticos, hoteleros y otros actores que trabajan en el sector turístico. Estos usuarios requieren acceso a información detallada y actualizada sobre destinos turísticos y servicios, para ofrecer a sus clientes opciones y recomendaciones relevantes.
- * Entidades gubernamentales: Como los GADs (Gobiernos Autónomos Descentralizados), municipios y parques nacionales, estas entidades pueden utilizar la ontología para gestionar y promocionar los destinos turísticos que se encuentran dentro de su jurisdicción. Les permite tener un mejor control y visión sobre los recursos turísticos disponibles.
- * Investigadores: Este grupo está compuesto por académicos, sociólogos, geógrafos y otros profesionales dedicados al estudio del turismo. Requieren acceso a datos y análisis detallados para llevar a cabo investigaciones en el campo del turismo.

Usos previstos: Una vez definidos los usuarios que harán uso de la ontología, el siguiente paso fue determinar las funcionalidades y aplicaciones para las cuales esta herramienta sería utilizada. En este caso, la ontología fue creada principalmente para ser empleada en una aplicación móvil que permita a los usuarios visualizar los recursos turísticos disponibles en su entorno cercano. Por lo tanto, se identificaron los siguientes usos previstos:

- * Definir recursos turísticos geolocalizados.
- * Definir rutas.
- * Definir regiones mediante polígonos.
- * Consultas de recursos geoespaciales.
- * Definir usuarios, regiones y organizaciones.
- * Manejar comentarios y puntajes de los recursos.

La metodología NEON, como se mencionó anteriormente, se basa en escenarios que abarcan desde la creación de una ontología desde cero hasta la adaptación de una ontología a diferentes idiomas y comunidades culturales (Suárez, 2012). En el caso de este proyecto, al utilizar ontologías existentes, fue necesario abordar el escenario 3 o 4 de la metodología. Finalmente, se optó por el escenario 4 debido a que permite realizar una reingeniería de los recursos

ontológicos, lo cual resultó más conveniente.

La elección del escenario 4 se debe a que, a diferencia del escenario 3, permite realizar modificaciones en las propiedades ontológicas existentes. Esto resultó beneficioso, ya que las posibilidades de encontrar una ontología que se ajustara completamente a los requisitos del proyecto eran relativamente bajas. Mediante el escenario 4, se pudo agregar nuevas propiedades y realizar ajustes necesarios para satisfacer todos los requisitos específicos del proyecto. Esta flexibilidad en la adaptación de la ontología permitió asegurar su pertinencia y adecuación a las necesidades del dominio del turismo.

Escenario 4: Reutilización y reingeniería de recursos ontológicos. Los desarrolladores de ontologías reutilizan los recursos ontológicos existentes y los rediseñan antes de integrarlos en la red ontológica. (Suárez, 2012)

A continuación se detallan paso a paso todos los procesos que se realizaron para obtener la red de ontologías resultante

Actividad 1 Búsqueda de ontologías

Durante esta actividad, se llevó a cabo una búsqueda de recursos ontológicos que pudieran satisfacer los requisitos específicos de la ontología en desarrollo. En el proceso de búsqueda, se encontró que algunas ontologías son ampliamente utilizadas como estándares en la web semántica. Un ejemplo de ello es la ontología Friend Of A Friend (FOAF¹), que se utiliza para describir usuarios. También se encontró la ontología Organization², la cual se utiliza para la descripción de organizaciones, y la ontología Geospatial and Metadata (GeoM³), que se encarga de la representación y gestión de datos geoespaciales.

En cuanto a las regiones, se encontraron ontologías como la de DBpedia⁴, que maneja múltiples clases para diferentes zonas geoespaciales. Sin embargo, en el caso de los comentarios, no se encontró ninguna ontología existente que satisficiera las necesidades y objetivos específicos del proyecto. Como resultado, se tomó la decisión de diseñar recursos ontológicos propios para abordar la cuestión de los comentarios de manera específica y adaptada a las necesidades de la aplicación.

Esta búsqueda de recursos ontológicos existentes, combinada con el diseño de recursos propios cuando fue necesario, permitió construir una ontología que se ajusta de manera precisa a los requerimientos y objetivos del proyecto.

¹<http://xmlns.com/foaf/0.1/>

²<https://www.w3.org/TR/vocab-org/>

³<http://www.opengis.net/ont/>

⁴<https://www.dbpedia.org/resources/ontology/>

Para la representación de recursos turísticos se realizó una búsqueda más profunda puesto que, iba a ser la parte principal de la red de ontologías. (Suárez, 2012) recomendaba buscar en repositorios y registros como Knowledge Zone, ONTHOLOGY.org, Oyster, Swoogle o WATSON, por desgracia al momento de acceder a los repositorios mencionados estos ya no se encontraban disponibles. Por lo que se buscaron en otros repositorios como DataHub.io y artículos publicados. Esta búsqueda arrojó 3 ontologías: TourPedia Ontology, Harmonise Ontology, OnTour Ontology.

La ontología TourPedia es creada bajo una iniciativa abierta que contiene un conjunto de datos enlazados de lugares turísticos, alojamientos, atracciones, puntos de interés (POI) y restaurantes (Gazzè et al., 2015). La ontología Harmonise por otro lado fue parte de un proyecto europeo que pretendía crear una infraestructura tecnológica basada en una ontología compartida para mejorar la cooperación de las pequeñas y medianas empresas europeas del sector turístico (Missikoff et al., 2003). Finalmente, la ontología OnTour es una ontología creada especialmente para el ámbito del turismo y desarrollada por DERI (Digital Enterprise Research Institute), además de los conceptos turísticos habituales (ubicación, alojamiento...), también incluye conceptos que describen actividades de ocio y datos geográficos (Prantner et al., 2007).

Actividad 2 Evaluación ontológica

Esta actividad consistió en inspeccionar el contenido y granularidad de los recursos ontológicos obtenidos en la actividad 1. Esta actividad al igual que la actividad 3 se realizó únicamente con las ontologías encontradas para la representación de los recursos turísticos.

Por un lado estaba Tourpedia que era representada por una ontología única (Figura 3.5), que contenía información derivada de clases y propiedades extraída de cada red social. El prefijo marl: se refiere a la ontología MARL ⁵, mientras que el prefijo acco: se refiere a la ontología ⁶ Acco y h: a la ontología Hontology (Chaves et al., 2012), ambos relacionados con el dominio del turismo.

Por otro lado de la ontología del proyecto Hamornise lamentablemente no se encontraron artículos o alguna clase de documentación que explique a detalle como esta construida la ontología, tampoco pudo hallarse su archivo owl, rdf o cualquier otro formato.

Finalmente, la ontología OnTour. Esta ontología se compone de 10 clases principales y 7 subclases, entre las cuales se encuentran las clases para alojamiento, actividades, ubicación, habitación, evento, entre otros. Cada una de estas clases define conceptos específicos y proporciona una estructura lógica para clasificar los datos relacionados. Además, la ontología

⁵<http://www.gsi.dit.upm.es/ontologies/marl>

⁶<http://ontologies.sti-innsbruck.at/acco/ns.html>

tp:Place
+ vcard:fn
+ dbpedia-owl:address
+ vcard:hasTelephone
+ vcard:hasPhoto
+ wgs84_pos:lat
+ wgs84_pos:long
+ dbpedia-owl:location
+ dbpedia-owl:wikiPageExternalLink
+ marl:hasOpinion
tp:Accommodation <i>extends</i> tp:Place
+ acco:feature
+ h:InternalFeature
+ skos:related
tp:Restaurant <i>extends</i> tp:Place
tp:POI <i>extends</i> tp:Place
tp:Attraction <i>extends</i> tp:Place
+ owl:sameAs
marl:AggregatedOpinion
+ marl:polarityValue

Fig 3.5: Ontología de Tourpedia (Gazzè et al., 2015)

OnTour posee un total de 30 propiedades que permiten establecer relaciones y atributos entre las clases. Algunas de estas propiedades incluyen "hasRoom" (posee habitación), "hasParking" (posee estacionamiento), "hasElevator" (posee ascensor) y "hasGPSCoordinates" (posee coordenadas GPS), entre otras (Figura 3.6).

Si bien la ontología OnTour ofrece una estructura detallada, es esencial tener en cuenta que su inclusión podría aumentar la complejidad de la ontología principal, lo cual no es deseable en el contexto de este trabajo. El enfoque principal de este proyecto se centra en el desarrollo de un aplicativo móvil, no en la elaboración de una ontología compleja. Por lo tanto, se prioriza la creación de una ontología más sencilla y de fácil mantenimiento que cumpla con los requisitos específicos del aplicativo.

Sin embargo, es importante destacar que la ontología OnTour puede ser considerada como una base sólida para futuros trabajos y extensiones. La inclusión de las clases "Activity" (actividad) y "Event" (evento) puede brindar un aporte significativo en la representación y organización de información relacionada.

Actividad 3 Comparación ontológica

En este paso se compararon los recursos ontológicos evaluados en la actividad 2, teniendo en cuenta un conjunto de criterios identificados por los desarrolladores (por ejemplo, documentación de ontologías existentes).

Debido a que la ontología Harmonise no fue encontrada, no fue considerada para el resto de pasos, resultando solo la comparación entre la ontología de TourPedia y OnTour. Por un lado, TourPedia es una ontología bastante simple que representa diferentes tipos de recursos mediante los atributos netamente necesarios, en cambio OnTour si bien tiene más clases y

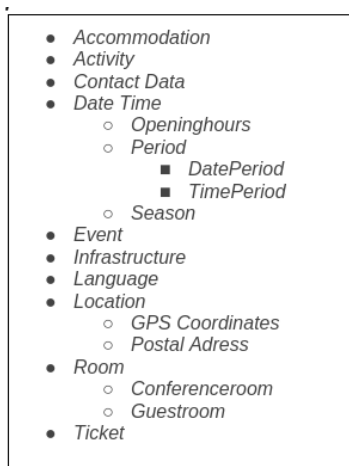


Fig 3.6: Clases y subclases de la ontología OnTour (Prantner et al., 2007)

aunque interesantes no son necesarias para este caso simplemente aumenta su complejidad, aunque también agrega clases interesantes como eventos y actividades, también es válido mencionar que la ontología OnTour fue desarrollada en 2004 hace poco menos de 20 años y la información que se ha encontrado de esta ontología también es bastante limitada, en cambio TourPedia fue creada en el 2010 y ha sido actualizada un par de veces siendo su última vez en 2015 y según (Gazzè et al., 2015) se va a seguir trabajando en ella. Además ofrece atributos definidos para integrar a recursos de DBPedia que sin duda suma un plus muy importante.

Actividad 4 Selección de ontologías

Clase	Ontologia
Usuarios	FOAF
Organizaciones	Organization
Regiones	DBPedia
Geolocalizaciones	GEOM
Recursos turisticos	TourPedia
Rutas y comentarios	a desarrollar

Table 3.6: Ontologías utilizadas

Como se mencionó en la actividad 1, se seleccionaron ontologías específicas para su utilización en el proyecto, teniendo en cuenta su popularidad y amplia adopción en el campo. Además de las ontologías previamente mencionadas, se agregó la ontología de TourPedia a la lista de ontologías seleccionadas para este trabajo. En la Tabla 3.6 se presenta un resumen de las ontologías utilizadas en el proyecto.

Actividad 5 Fusión e integración de ontologías

En este apartado, se fusionaron los recursos utilizados de cada una de las ontologías previamente seleccionadas para crear la nueva ontología. Para la representación de usuarios, se reutilizó la clase "Person" y algunas propiedades relacionadas tanto a esta clase como a su clase padre "Agent", como "hasName" y "hasNick", entre otras. De la ontología "Organization", se reutilizó la clase "Organización" y la propiedad "memberOf" para representar las organizaciones y los usuarios que forman parte de ellas. Para la representación de las regiones, se reutilizó la clase "Region" de DBpedia, que se utiliza para hacer referencia a una "porción de territorio". En cuanto a la representación de los recursos turísticos, se utilizó la ontología "TourPedia" con ciertas modificaciones. En primer lugar, se reemplazaron las propiedades de localización, optando por utilizar el estándar OGC GeoSPARQL en lugar de WGS84. Esto se debió a que en pruebas previas se encontraron resultados incorrectos en consultas geoespaciales con la palabra clave "within" utilizando WGS84, posiblemente debido a incompatibilidades con la triplestore (en este caso, Fuseki). Además, se agregaron propiedades como "hasTitle" y "hasDescription" provenientes de la ontología Dublin Core (dc) para proporcionar más detalles sobre el recurso. También fue necesario almacenar información relacionada con imágenes y vídeos de cada recurso. La multimedia se sube a Facebook y se obtiene un ID para cada imagen y vídeo. En el caso de las imágenes, se puede obtener una URL temporal a través del ID, lo cual ayudaría a reducir el tiempo necesario para obtener las imágenes. Para abordar esta cuestión, se creó una clase denominada "FacebookResource" con una propiedad de tipo "long" para almacenar el ID. Además, se agregó una subclase llamada "FacebookImage" que incluye la propiedad "hasUrl" para almacenar el enlace a la ruta temporal de la imagen. La clase "FacebookResource" se vincula al recurso mediante la propiedad "hasFacebookResource". Para los comentarios, también se creó una clase con propiedades como "hasContent" (que incluye el contenido del comentario), "hasRate" (que indica la puntuación del recurso comentado) y "commentBy" (que relaciona al usuario que realiza el comentario). Al igual que con las imágenes y los vídeos, se utilizó la propiedad "hasFacebookResource" para gestionarlos. Finalmente, para las rutas, se creó una clase llamada "Ruta" con propiedades como "hasTitle"

y "hasDescription" de la ontología Dublin Core, además de una propiedad propia denominada "hasPlaces" que se vincula a una colección de recursos. De manera similar, se abordó el tema de la lista de favoritos utilizando las propiedades correspondientes.

Después de realizar la integración de las distintas ontologías y recursos seleccionados, se obtuvo una ontología más completa y coherente que cumplen con las funcionalidades planteadas. En la Figura 3.7 se puede observar, a través de un grafo, la estructura final de la ontología.

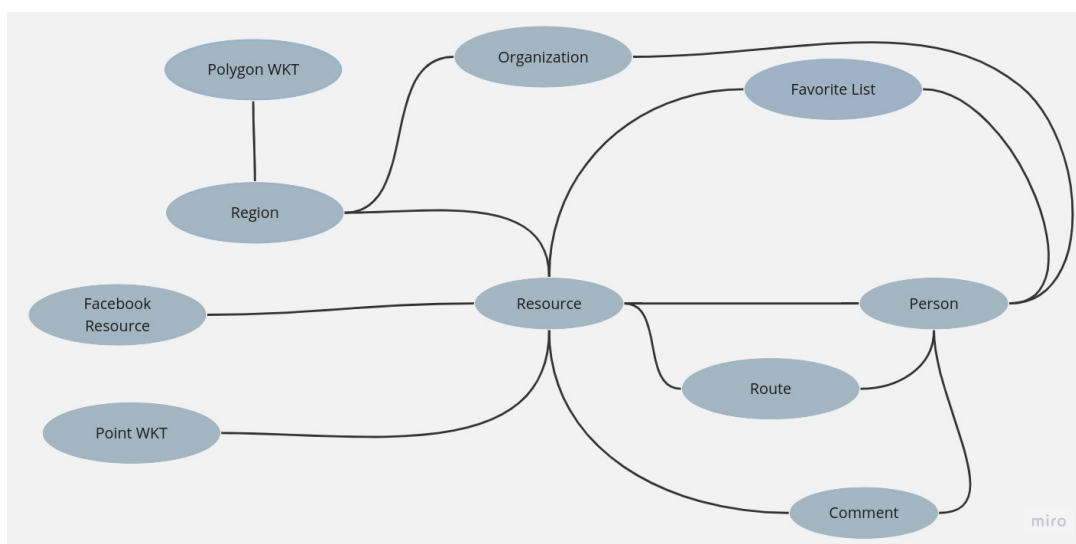


Fig 3.7: Ontología resultante representada a través de un grafo

En la sección de anexos, se encuentran disponibles instancias de recursos, organizaciones y regiones. Además en el repositorio de Git ⁷ se encuentra la ontología completa, ahí puede revisarse de manera más clara y concreta la estructura y las relaciones entre las clases.

3.4. Diseño de mockups

Los mockups son una parte fundamental en la etapa de diseño, ya que permiten visualizar y validar la apariencia y funcionalidad de la aplicación antes de su implementación. Por lo tanto, en esta sección se aborda la creación de los mockups o prototipos de interfaces para la aplicación. Se explica cada una de las pantallas y las herramientas utilizadas.

3.4.1. Prototipo de interfaces de aplicación móvil

Previo a la creación de los mockups de la aplicación móvil de turismo, se llevó a cabo un análisis de varias aplicaciones móviles extranjeras similares que se encuentra en el mercado,

⁷<https://github.com/Argos98p/Tesis>

tales como: Civitatis, Minube, España Smart Guide, España Top Travel, Airbnb, CiclaMadrid Conecta, Visit London Official City Guide, entre otras. Este análisis permitió identificar las fortalezas y debilidades de estas aplicaciones. Con esta información como guía, se diseñaron las interfaces para la aplicación.

Los mockups fueron diseñados en la herramienta Figma y pueden encontrarse en el repositorio de GitHub del proyecto ⁸. A continuación se detalla cada pantalla y su funcionamiento.

Se crearon las páginas para:

- Login y splash screen
- Explorar
- Recurso detallado
- Mapa
- Rutas
- Perfil
- Configuración

Pantalla de login y splash screen

La ventana de Login y splash screen son elementos esenciales en una aplicación móvil. La ventana de Login permite a los usuarios ingresar sus credenciales para acceder a la aplicación, mientras que la pantalla de carga de inicio muestra una animación o imagen que indica que la aplicación está cargando y preparándose para su uso. Es importante que la ventana de Login sea fácil de usar y accesible para todos los usuarios, además la splash screen puede tener una imagen o animación que representa el uso de la aplicación, aquí es importante que la pantalla de carga no sea demasiado larga para que los usuarios no se desesperen y abandonen la aplicación antes de que se cargue por completo. En la Figura 3.8 puede observarse la interfaz de estas pantallas.

Pantalla Explorar

Esta pantalla presenta una lista de recursos turísticos disponibles junto con información adicional importante. Entre los detalles mostrados se encuentra la distancia desde la ubicación actual del usuario hasta cada recurso, así como una puntuación que refleja el promedio de calificaciones y comentarios recibidos. También se incluye un botón que permite agregar el recurso a la lista de favoritos del usuario. Esta pantalla es una herramienta útil para explorar y evaluar los recursos

⁸<https://github.com/Argos98p/Tesis>

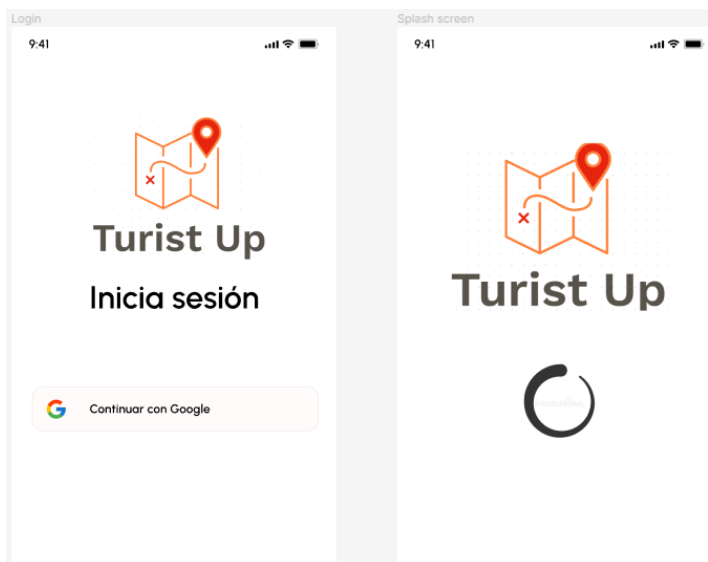


Fig 3.8: Página de login y splash screen

turísticos, brindando datos relevantes para tomar decisiones informadas durante la experiencia de viaje. En la Figura 3.9 se muestra la interfaz de la pantalla explorar.

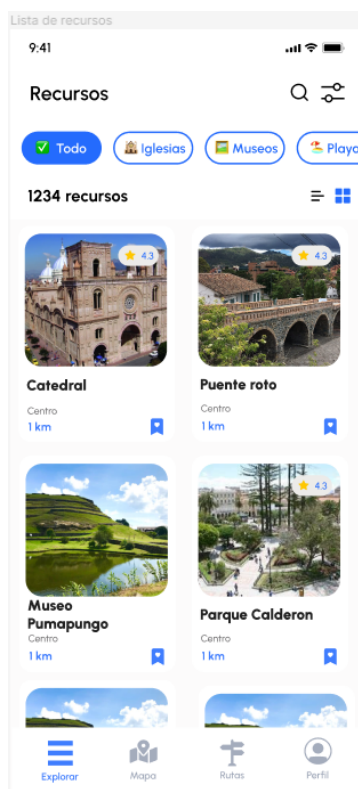


Fig 3.9: Página explorar

Pantalla de información del recurso

Esta pantalla muestra la información detallada sobre un recurso turístico disponible en la aplicación, en donde se incluye fotos, nombre, categoría, calificación, número de opiniones, nombre de la organización a la que pertenece, descripción, fotos de los comentarios, localización en un mapa, opiniones de los usuarios y un sistema de comentarios y valoraciones. En la Figura 3.10 se muestra el mockup de la pantalla de información del recurso.

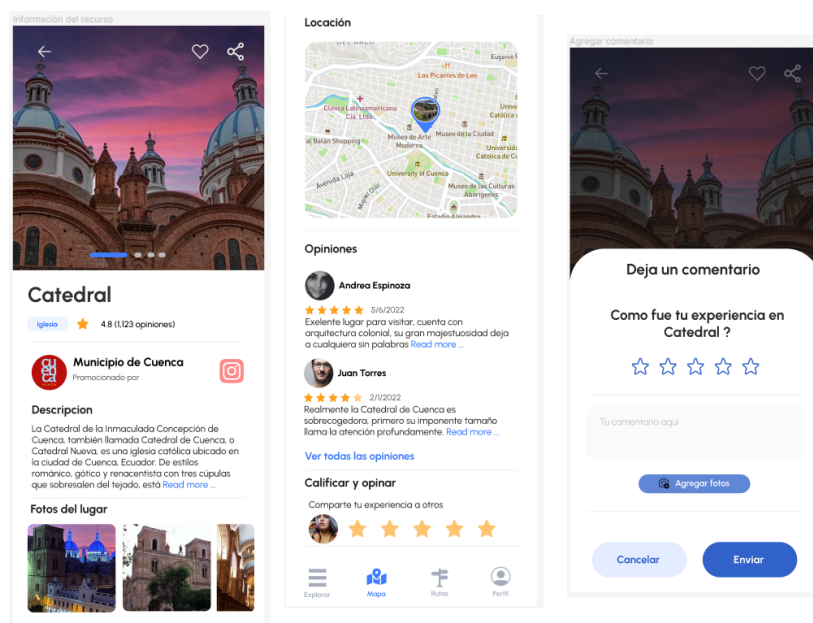


Fig 3.10: Ventana de información del recurso y comentarios

Pantalla Mapa

Esta ventana es un mapa interactivo que muestra las ubicaciones de los recursos turísticos y permite al usuario realizar un zoom y explorar diferentes áreas. Aquí se agregaron marcadores personalizados con una imagen del recurso, los cuales indican la ubicación del recurso sobre el mapa, dentro del mapa también se incluye un filtro de categoría, que permite al usuario ver los recursos que le interesan, mejorando la experiencia de usuario al proporcionar una forma fácil de encontrar lo que está buscando dentro del mapa. En la Figura 3.11, se muestra el mockup de la pantalla mapa.

Pantalla Rutas

En esta ventana, el usuario puede explorar las rutas turísticas que han sido previamente cargadas por los administradores de la aplicación. Los administradores crean rutas personalizadas que son populares o recomendadas por las personas del lugar. Los usuarios pueden ver el

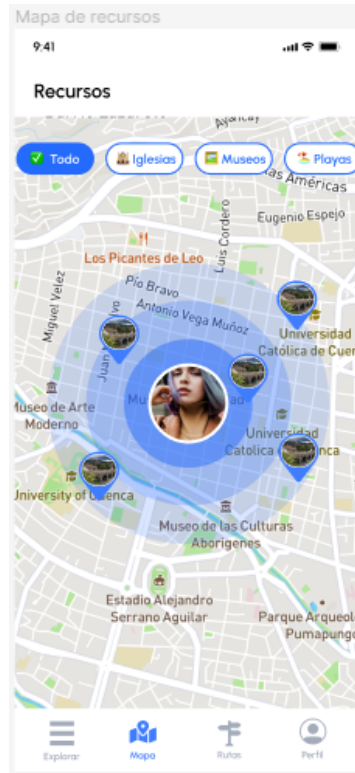


Fig 3.11: Ventana del mapa de recursos

nombre de la ruta y su localización, y al acceder podrán ver en el mapa el camino a seguir de dicha ruta. En la Figura 3.12 se muestra el mockup de la ventana ruta.

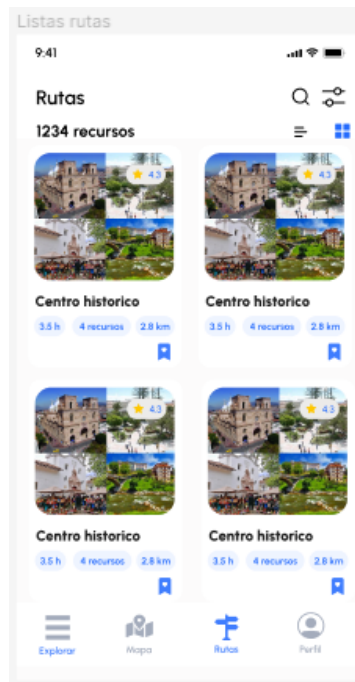


Fig 3.12: Ventana de las rutas creadas por el administrador

Pantalla Perfil

En esta pantalla se puede encontrar las subsecciones de información del usuario, favoritos y mi lista. En la subsección de información se incluye los datos básicos del usuario, como su nombre, email, foto de perfil, etc. En la subsección de favoritos el usuario puede encontrar una lista de los recursos turísticos que ha seleccionado previamente como sus favoritos dentro de la aplicación. Esto permite un acceso rápido y fácil a los recursos turísticos que el usuario ha marcado como preferidos y puede ser útil para planificar futuras actividades turísticas. Finalmente, en la subsección de mi lista, el usuario puede encontrar las rutas que ha creado personalmente para planificar sus visitas según sus necesidades y preferencias de viajes. En la Figura 3.13 se muestra el mockup de la ventana perfil.

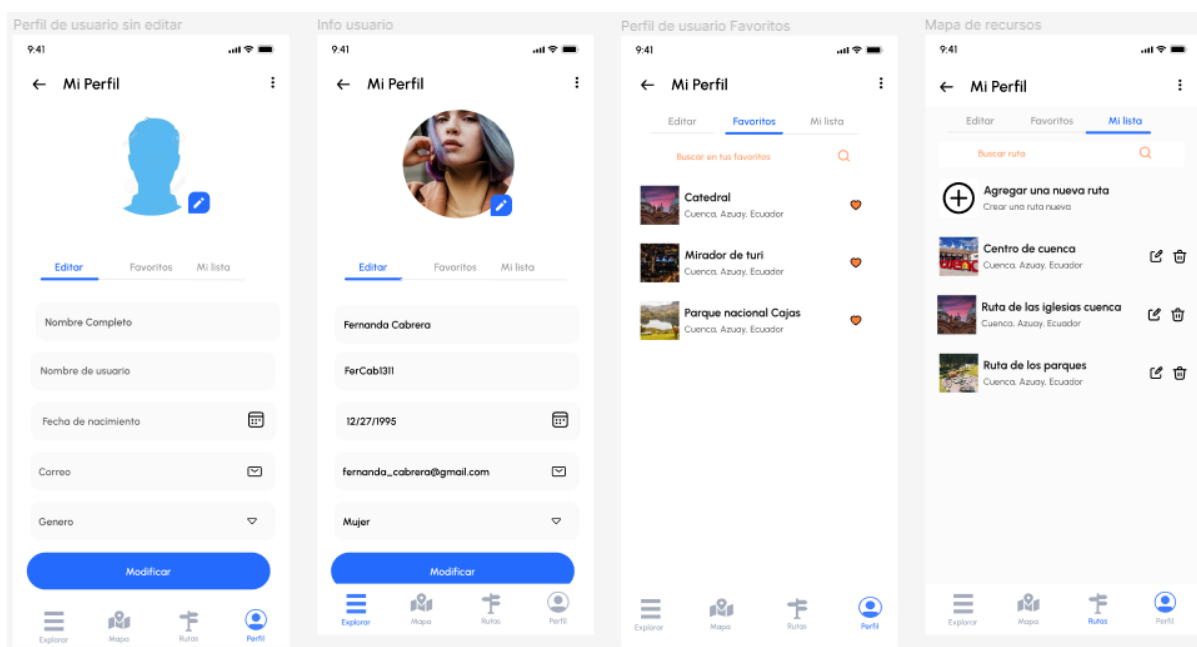


Fig 3.13: Ventanas del perfil de usuario

3.4.2. Prototipo de interfaces de aplicación web

En el desarrollo del proyecto, se decidió enfocarse exclusivamente en la creación de interfaces para la aplicación móvil, sin desarrollar interfaces específicas para la aplicación web. Esta decisión se basó en el análisis de los requisitos y las necesidades de los usuarios, así como en consideraciones de tiempo y recursos disponibles. Al concentrar los esfuerzos en la aplicación móvil, se buscó proporcionar una experiencia de usuario óptima y adaptada a los dispositivos móviles, aprovechando las funcionalidades y características propias de estos dispositivos. Además para la aplicación web se utilizaron templates preexistentes, realizando modificaciones

y adaptaciones necesarias para cumplir con los requisitos específicos del sistema. Estos templates ofrecían una base sólida y funcional para la interfaz de usuario, lo que permitió ahorrar tiempo en el diseño y desarrollo de la aplicación web. A través de estas modificaciones, se aseguró que la interfaz de la aplicación web fuera intuitiva y amigable para los usuarios, brindando una experiencia coherente con la propuesta general del proyecto. Se priorizó la usabilidad y la accesibilidad, manteniendo la coherencia visual con la aplicación móvil y garantizando una experiencia fluida para los usuarios en ambos entornos.

3.5. Arquitectura de Software

La arquitectura de software propuesta, basada en los requisitos del proyecto y la ontología definida, se ilustra en la Figura 3.14. Esta arquitectura ha sido cuidadosamente diseñada para cumplir con los objetivos del sistema y garantizar un rendimiento eficiente y escalable. En la Figura 3.14 se representan los diferentes componentes y su interacción, proporcionando una visión clara de cómo se estructura el sistema y cómo se comunican entre sí. La arquitectura propuesta se considera óptima para satisfacer las necesidades del proyecto, aprovechando al máximo las tecnologías y herramientas seleccionadas.

La arquitectura del sistema está dividida en dos partes esenciales: el Backend y el Frontend. El Backend fue realizado bajo un enfoque de microservicios con Java, Spring Boot y desplegados en contenedores Docker en un servidor de Google Cloud. El Frontend está constituido por la aplicación móvil desarrollada en Flutter y la aplicación web realizada en React. Como repositorios de información se tiene a Apache TDB, Fuseki, Postgress y Facebook.

3.5.1. Frontend

También conocido como la capa de presentación, es la parte visible y accesible para los usuarios finales de una aplicación. Está compuesto por una variedad de elementos, como botones, menús, textos e imágenes, que se combinan estratégicamente para crear una experiencia de usuario intuitiva y atractiva. El frontend desempeña un papel fundamental en el desarrollo de aplicaciones, ya que una interfaz poco amigable, poco intuitiva o poco atractiva puede desmotivar a los usuarios y llevarlos a no utilizar la aplicación, incluso en el futuro.

El frontend del sistema está constituido por dos componentes principales: la aplicación móvil y la aplicación web. Estos componentes brindan interfaces de usuario intuitivas y amigables que permiten a los usuarios interactuar con el sistema de manera eficiente. La aplicación móvil se desarrollo siguiendo los mockups descritos en la sección 3.4 de este documento y esta

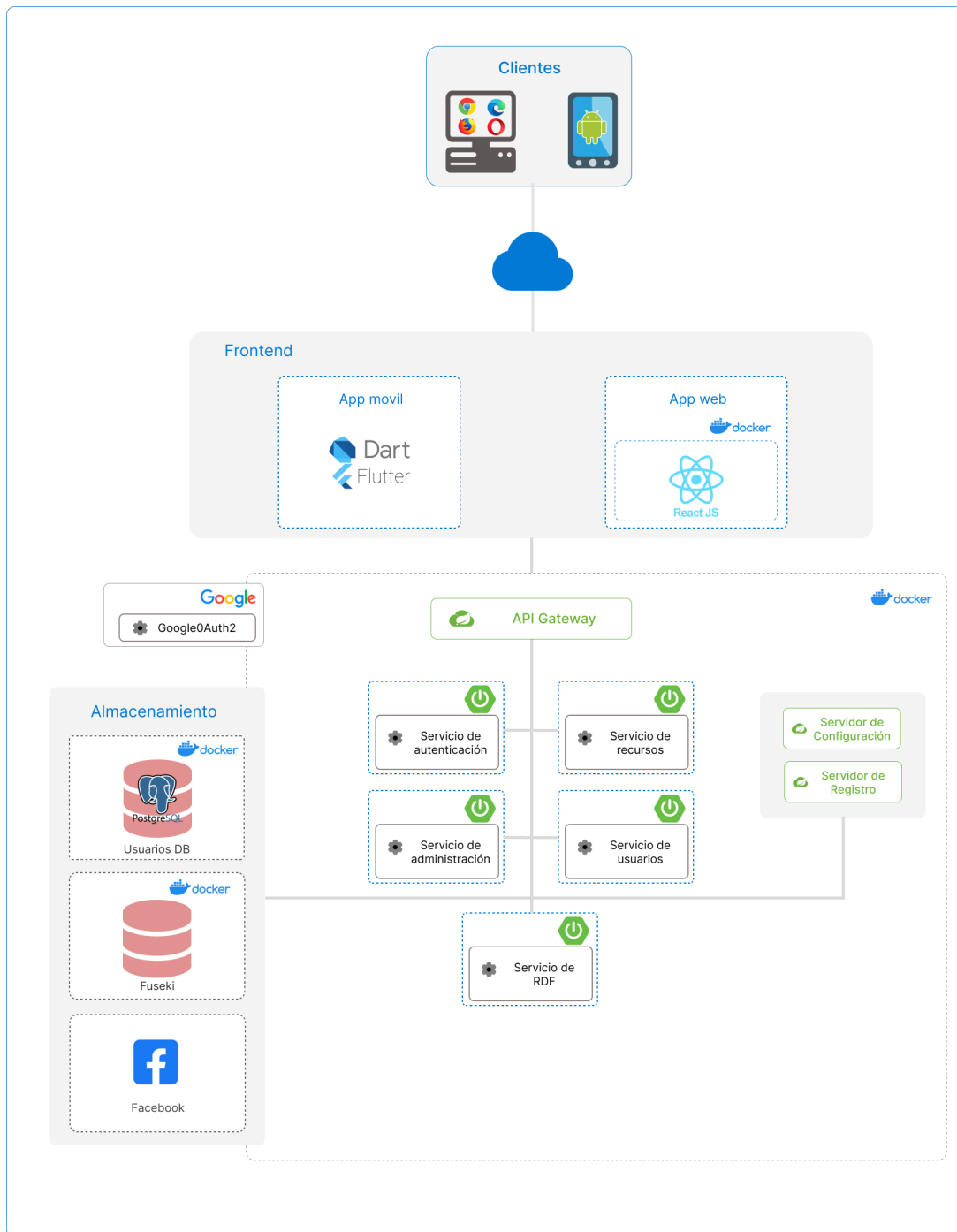


Fig 3.14: Arquitectura del sistema

diseñada específicamente para dispositivos móviles. Ofrece una experiencia optimizada para su uso en smartphones, aprovechando las capacidades y características propias de estos dispositivos. Por otro lado, la aplicación web proporciona una interfaz accesible a través de navegadores de escritorio y dispositivos móviles, brindando flexibilidad en el acceso al sistema. Ambos componentes del frontend se complementan entre sí, la aplicación móvil se encarga de registrar los recursos turísticos mientras que la aplicación web gestiona estos recursos

turísticos.

Aplicación Móvil

El desarrollo del frontend de la aplicación móvil se realizó utilizando Flutter, debido a que este framework ofrece una amplia variedad de widgets personalizables, lo que permite crear diseños atractivos y funcionales de manera más rápida y eficiente. Otra ventaja de Flutter es su capacidad para compilar código nativo para iOS y Android, lo que garantiza un mejor rendimiento y una experiencia de usuario más fluida en comparación con otros frameworks de desarrollo híbrido. Flutter se considera la mejor opción para el desarrollo móvil debido a su alto rendimiento y facilidad de uso. En este proyecto se optó por utilizar el entorno de desarrollo Visual Studio Code, una herramienta de desarrollo de software que brinda una amplia gama de extensiones y complementos específicos para crear aplicaciones en Flutter.

La arquitectura de Flutter se basa en capas, las cuales pueden ser implementadas utilizando diferentes enfoques según las necesidades del proyecto. En sí, no existe una arquitectura específica predeterminada en Flutter, pero existen patrones de diseño comunes que se pueden seguir para obtener una buena estructura en capas. Para el desarrollo del frontend, se utilizó el patrón MVC (Modelo-Vista-Controlador). En este modelo, se maneja la capa de datos y la lógica de la aplicación, la capa de vista encargada de la representación visual y la interacción con el usuario y la capa de controlador que actúa como intermediario entre la capa de vista y la capa de modelo. En este enfoque, el controlador se encarga de la comunicación entre la vista y el modelo, asegurando que los datos y la lógica se mantengan separados de la interfaz de usuario, permitiendo una mayor reutilización de código, prueba y mantenimiento del proyecto. En base a la información de los mockups y las necesidades del proyecto, se creó un diagrama que detalla todas las pantallas básicas de la aplicación. Este diagrama se encuentra representado en la Figura 3.15, el cual ofrece una visualización completa de las diferentes pantallas y su estructura en la aplicación, proporcionando una referencia visual clara para comprender la navegación y la disposición de las pantallas en el flujo de la aplicación.

A continuación, se detallará cada una de las pantallas observadas en la Figura 3.15.

Pantalla de Inicio de Sesión (Login)

En esta pantalla, cuando el usuario accede por primera vez a la aplicación e intenta iniciar sesión, se establece una comunicación con la API de Google mediante OAuth2 utilizando la librería para flutter 'google sign in'. Una vez establecida la comunicación y recibidos los datos de inicio de sesión, se procede a almacenarlos en el almacenamiento persistente del dispositivo.

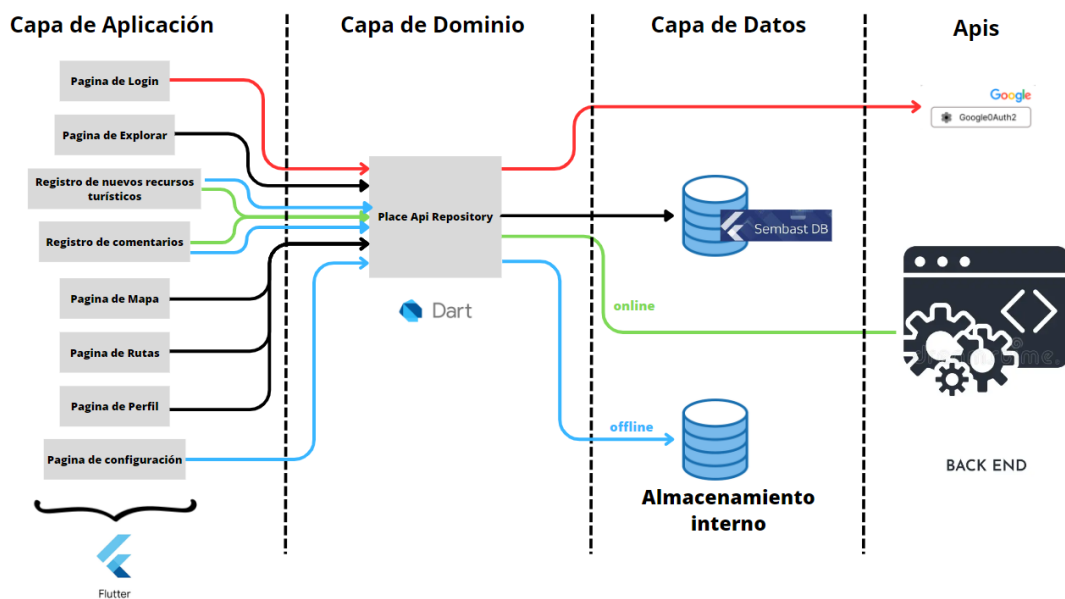


Fig 3.15: Diagrama de capas de la aplicación móvil

Estos datos incluyen el Id, el token, el correo electrónico, el nombre y la imagen del usuario los cuales resultan útiles para consultas posteriores en la base de datos y la ontología.

La integración de la autenticación de Google con la aplicación móvil fue realizada con Google Cloud Platform, para ello fue necesario crear un nuevo proyecto y cliente OAuth en la página de Google Cloud rellenando la información requerida Figura 3.16. El nombre del paquete se encuentra en el archivo "androidManifest.xml" del proyecto de la aplicación móvil y la huella digital requerida por OAuth se la obtuvo mediante el comando "keytool -keystore path-to-debug-or-production-keystore -list -v".

Además del cliente Oauth, fue imprescindible crear la pantalla de consentimiento, esta pantalla es la que se muestra cuando el usuario intenta realizar la autenticación con Google, en la configuración de esta pantalla es necesario solicitar los permisos necesarios para la autenticación Figura 3.17 .

Pantalla Explorar

En esta pantalla, se utiliza el ID del usuario almacenado en la pantalla anterior para realizar consultas y buscar datos relacionados con ese usuario. Al hacer la consulta en la base de datos, se obtienen todos los recursos turísticos disponibles, incluyendo aquellos que el usuario haya marcado como favoritos previamente, si los hubiera definido.

Además, en esta pantalla se encuentra la opción de filtrar los recursos turísticos por categoría. Con el fin de mejorar el rendimiento de la aplicación, este proceso de filtrado se realiza sobre

← ID de cliente para Android BORRAR

Nombre *

 El nombre de tu cliente de OAuth 2.0. Este nombre solo se usa para identificar al cliente en la consola y no se mostrará a los usuarios finales.

Nombre del paquete *

 En tu archivo AndroidManifest.xml

Huella digital del certificado SHA-1 *

 El certificado de firma SHA-1 con huella digital restringe el uso de tus aplicaciones para Android. [Más información](#)

Use this command to get the fingerprint.

```
$ keytool -keystore path-to-debug-or-production-keystore -list
```

Nota: La configuración puede tardar entre 5 minutos y algunas horas en aplicarse

GUARDAR **CANCELAR**

Fig 3.16: Crear cliente OAuth en Google Cloud

Tus permisos no sensibles

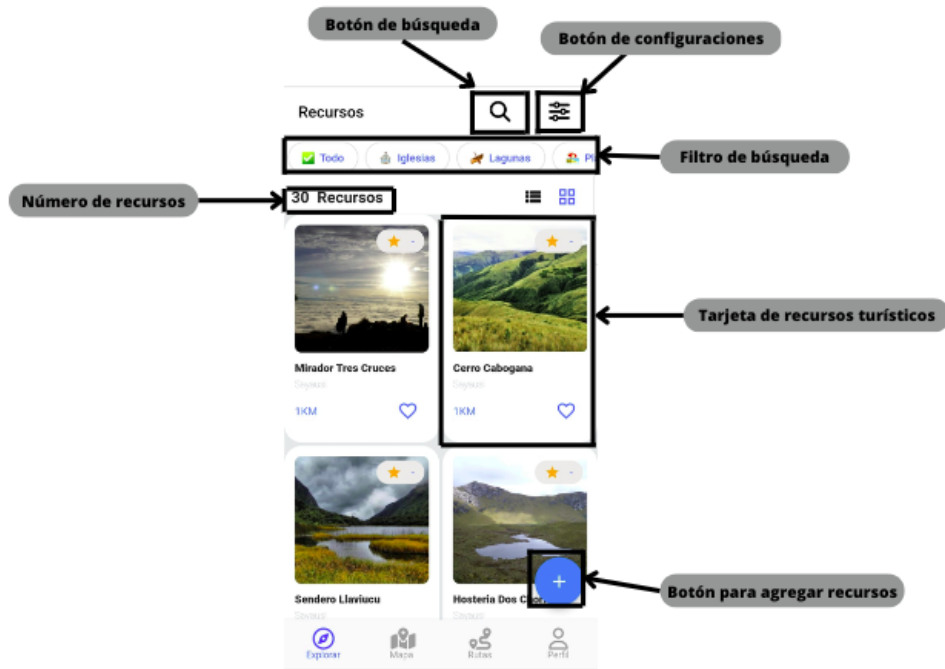
API ↑	Alcance	Descripción para el usuario	
	.. ./auth/userinfo .email	See your primary Google Account email address	🗑️
	.. ./auth/userinfo .profile	See your personal info, including any personal info you've made publicly available	🗑️

Fig 3.17: Permisos necesarios para la autenticación

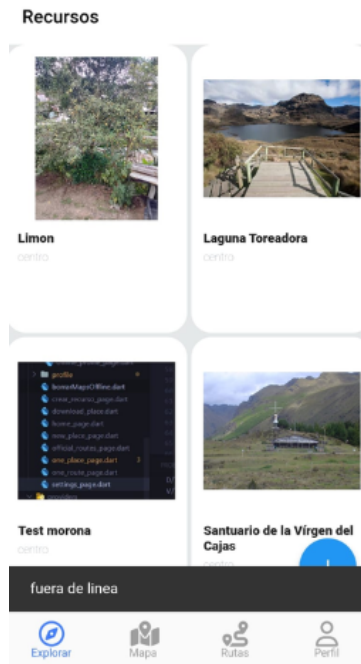
los datos obtenidos anteriormente al consultar los recursos turísticos, en lugar de hacer una consulta adicional a la base de datos, evitando posibles retrasos causados por la capacidad actual del servidor en el que se encuentra el sistema.

En cuanto a la funcionalidad offline, esta pantalla también cuenta con la capacidad de visualizar todos los recursos disponibles, siempre y cuando estos hayan sido descargados previamente en la sección Offline de la página de perfil. Esta funcionalidad permite acceder a la información de un recurso turístico incluso en ausencia de conexión a internet. Sin embargo, es importante tener en cuenta que en el modo offline, la visualización de la información puede ser menos

detallada en comparación con el modo online. Estas pantallas se puede observar en las Figuras 3.18a y 3.18b



(a) Pantalla Explorar Online



(b) Pantalla Explorar Offline

Fig 3.18: Pantalla Explorar

Registro de nuevos recursos turísticos

En esta pantalla, los usuarios tienen la opción de registrar nuevos recursos turísticos. Si el dispositivo tiene conexión a Internet, los recursos se envían directamente a los microservicios del backend para posteriormente guardarlos en la base de datos TDB que maneja Fuseki, donde se encuentran los datos de la ontología. Sin embargo, si el dispositivo se encuentra sin conexión, se detecta automáticamente y los datos se almacenan en la base de datos Sembast DB, que es una base de datos NoSQL embebida en el teléfono. En Sembast DB, se almacenan los siguientes tres datos: type (formato de los datos, en este caso, una cadena de texto), status (pendiente, con error o realizado) y data (los datos del recurso en sí).

Una vez que el dispositivo recupera la conexión a Internet, se verifica este estado y se realiza una búsqueda en la base de datos Sembast, para identificar los datos que tienen un estado pendiente. Luego, se procede a enviar esos datos a los microservicios del backend para ser almacenados en la ontología y se actualiza el status a realizado, finalizando así el proceso de publicación offline. Esta pantalla se puede observar en la Figura 3.19.



Fig 3.19: Pantalla de registro de nuevos recursos turísticos

Registro de comentarios

En esta pantalla, los usuarios tienen la posibilidad de registrar comentarios en cualquier recurso. La función de publicación de comentarios es compatible tanto en modo offline como en modo online y opera de forma offline siguiendo el mismo principio utilizado en la publicación de nuevos recursos offline. La única diferencia radica en el almacenamiento de datos, ya que

en este caso se almacenan los datos del comentario en lugar de los datos del recurso.

Si el dispositivo está conectado a Internet, los comentarios se envía directamente a los microservicios del backend para almacenarse en la ontología. Sin embargo, si el dispositivo está sin conexión, los datos del comentario se almacenan localmente en la base de datos (Sembast DB), junto con los demás comentarios pendientes. Cuando el dispositivo recupera la conexión a Internet, se realiza una verificación del estado y se envían los comentarios pendientes almacenados en la base de datos Sembast a los microservicios del backend, lo cual asegura que los comentarios se publiquen correctamente tanto en modo offline como en modo online.

Página de Mapa

En esta pantalla, los usuarios pueden visualizar los recursos turísticos en un mapa. Para ello, se utiliza la librería 'Location', que permite obtener la ubicación actual del usuario en términos de latitud y longitud, permitiendo mostrar su posición en el mapa en tiempo real. Es importante activar el GPS para utilizar esta funcionalidad de la aplicación. Además, se utiliza la librería 'Flutter Map', la cual ofrece un mapa de forma gratuita. Esto evita la necesidad de utilizar el widget proporcionado por Google Maps, el cual requiere un pago de servicios adicionales por su uso.

Para trazar rutas desde la ubicación del usuario hacia un recurso, se utilizó la librería 'flutter polyline points'. Esta biblioteca permite generar los puntos necesarios en el mapa y trazar la ruta correspondiente. Es importante destacar que la ruta se generará siempre y cuando exista un camino trazado en el mapa. Si no hay un camino disponible, la ruta se dibujará hasta donde sea posible seguir un camino definido.

Además de la visualización del mapa y las rutas, en esta pantalla también se muestra un listado de recursos turísticos, estos recursos se ordenan de acuerdo a su puntuación, desde el más alto hasta el más bajo, de esta manera, los usuarios pueden identificar los destinos turísticos más valorados en el área de búsqueda.

Esta combinación de funcionalidades en la Página de Mapa intenta proporcionar a los usuarios una experiencia completa al explorar y navegar por los recursos turísticos disponibles dentro de la aplicación, esta pantalla se puede observar en la Figura 3.20.



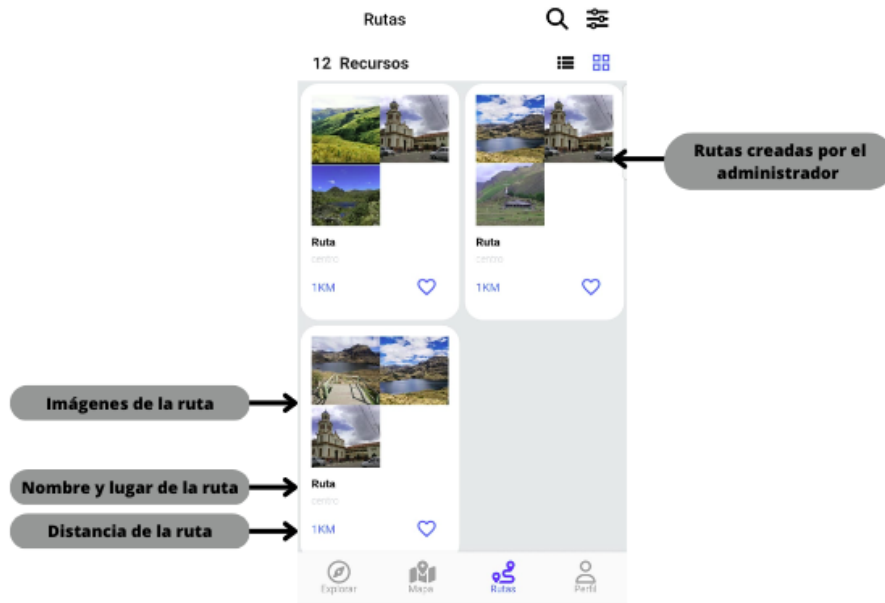
Fig 3.20: Pantalla del mapa de la aplicación

Página de Rutas

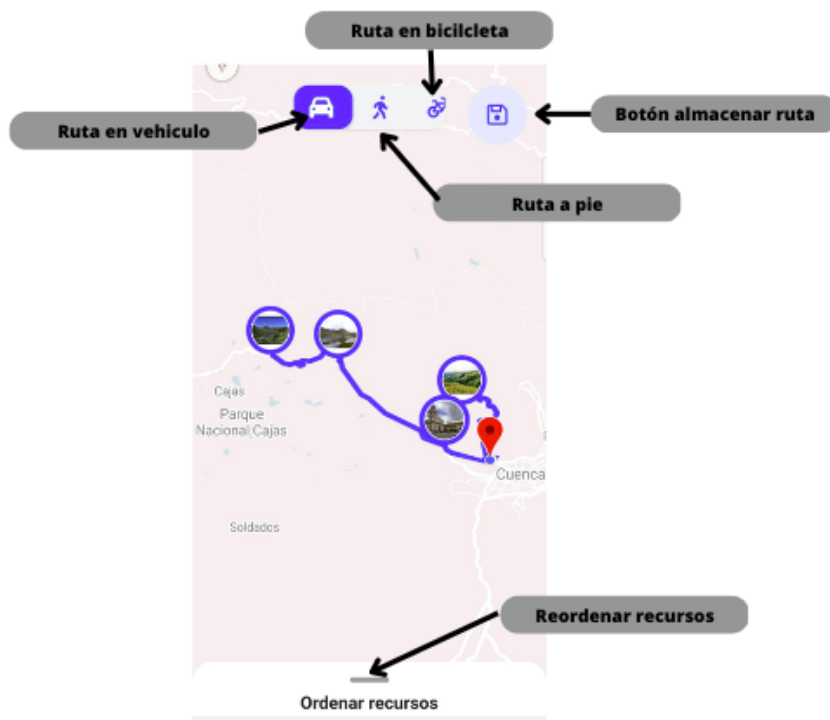
En esta pantalla, los usuarios pueden visualizar las rutas prediseñadas por los administradores de la aplicación, estas rutas son oficialmente reconocidas por los GADs o por los propios administradores del sector, las cuales proporcionan un gran valor para los turistas, ya que permite explorar las rutas más populares y destacadas del lugar en el que se encuentren.

En la Página de Rutas también se utiliza la librería 'flutter polyline points', la cual ayuda a trazar las rutas desde la ubicación actual del usuario hacia todos los recursos turísticos incluidos en cada ruta. Esta funcionalidad permite a los usuarios seguir las rutas de manera guiada y descubrir los recursos turísticos relevantes a lo largo del camino. Al mostrar las rutas prediseñadas en esta página, los usuarios pueden obtener una visión general de las opciones de turismo disponibles en el área, lo cual facilita la planificación de sus actividades y brinda una experiencia turística enriquecedora.

La integración de la librería "flutter polyline points" asegura que las rutas se muestren de manera precisa en el mapa, lo que mejora la usabilidad del usuario al seguir las rutas recomendadas. En la Figura 3.21a se puede observar la pantalla de rutas prediseñadas, al igual que en la Figura 3.21b se puede apreciar una de las rutas dibujadas a partir de las rutas prediseñadas.



(a) Pantalla Rutas



(b) Ruta prediseñada dibujada en el mapa

Fig 3.21: Pantallas de rutas

Pantalla de Perfil

En esta pantalla, los usuarios pueden visualizar y gestionar su información personal, como su nombre, correo electrónico y foto de perfil esta pantalla se puede visualizar en la Figura 3.22a. Además, tienen acceso a una lista que muestra los recursos turísticos que se han marcado

como favoritos, lo que permite tener un registro de los lugares que desean visitar mostrado en la Figura 3.22b.

Una funcionalidad destacada en esta pantalla es la posibilidad de crear rutas personalizadas, es decir, los usuarios pueden diseñar rutas a su gusto, seleccionando los recursos turísticos que desean visitar en un viaje específico. Esto brinda la flexibilidad de planificar su itinerario turístico de acuerdo a sus preferencias y necesidades, lo cual se puede observar en la Figura 3.22c.

Además, se incluye una pestaña llamada "Offline" mostrada en la Figura 3.22, que permite a los usuarios descargar los datos de los recursos turísticos disponibles en la ontología, permitiendo utilizar la aplicación en modo offline, visualizando todos los recursos dentro del mapa sin necesidad de tener conectividad a internet. Esta característica es especialmente útil cuando los usuarios se encuentran en áreas con una señal de internet débil o inexistente. La Pantalla de Perfil ofrece a los usuarios un espacio personal donde pueden visualizar su información, acceder a sus recursos turísticos favoritos y crear rutas personalizadas.

Pantalla de Configuraciones

En esta pantalla, los usuarios tienen la opción de ajustar el rango de búsqueda de recursos turísticos. Pueden elegir entre diferentes opciones de rango, que incluyen 0.5 km, 1 km, 5 km, 10 km, 20 km, 30 km, 50 km y +50 km. Al configurar este rango, la aplicación mostrará los recursos turísticos que se encuentren dentro de esa distancia desde la ubicación actual del usuario. Además, en esta ventana de configuraciones, los usuarios pueden acceder al historial de sus comentarios. También tienen la posibilidad de establecer un rango de fechas para visualizar los comentarios realizados en ese período específico, esto permite revisar y recordar los comentarios que han dejado en recursos turísticos anteriores, facilitando la gestión y referencia de sus experiencias pasadas. La pantalla de configuraciones puede visualizarse en la Figura 3.23

En caso de tener dudas sobre este apartado, se recomienda consultar el Manual de Usuario, el cual proporciona información detallada y está ilustrado con ejemplos y explicaciones adicionales, es una guía completa que abarca todos los aspectos mencionados anteriormente y brinda una visión más amplia de la funcionalidad y características de la aplicación.

Aplicación Web

La aplicación de administración fue hecha con la librería React y se convierte en una herramienta esencial para aquellos usuarios con roles de administrador en la aplicación. Desde esta página,

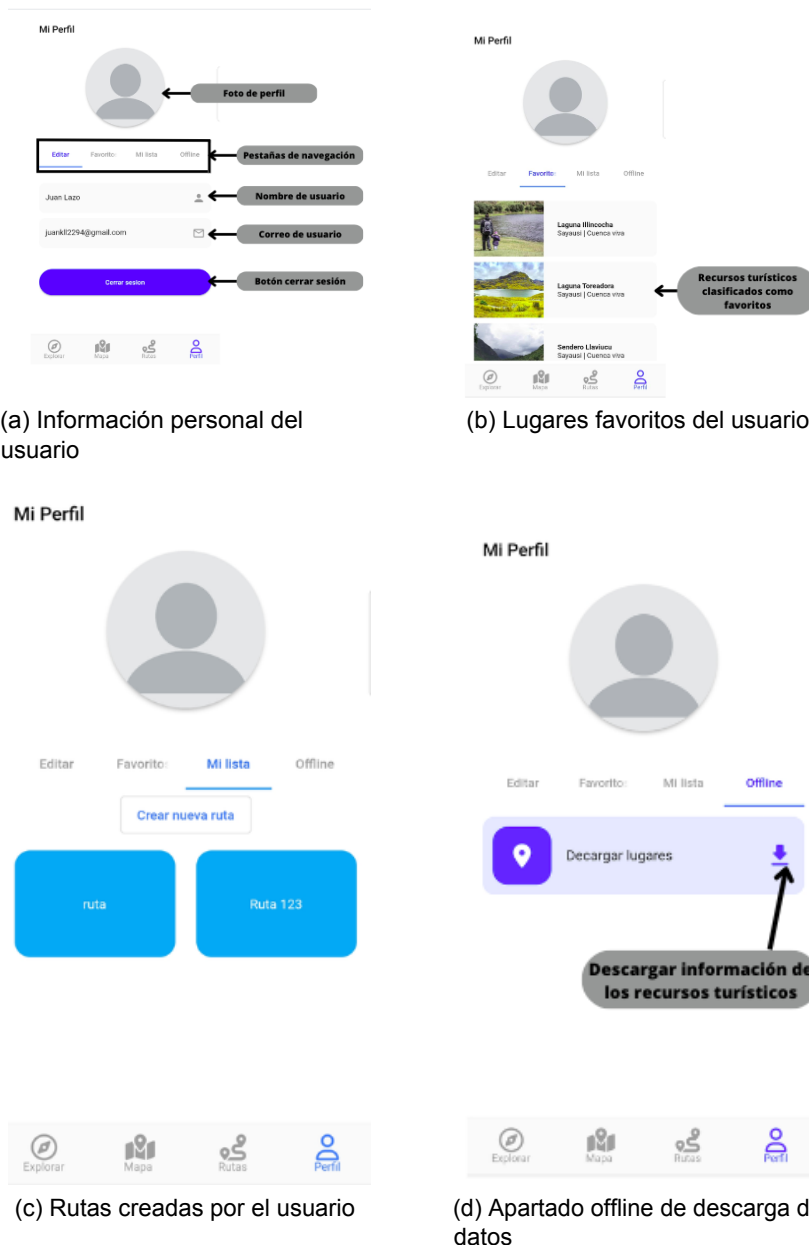


Fig 3.22: Pantallas de perfil

los administradores tienen la capacidad de gestionar y controlar aspectos clave de los recursos turísticos de cada organización.

La aplicación web, tal como se representa en la Arquitectura, sigue un enfoque cliente-servidor. En este modelo, el cliente es el navegador web utilizado por el usuario, mientras que el servidor es la parte que se encarga de procesar las solicitudes del cliente y enviar las respuestas correspondientes. El cliente realiza peticiones al servidor, como cargar una página o enviar datos, y el servidor responde proporcionando los recursos solicitados o ejecutando las acciones requeridas.

La aplicación web cuenta con varias pantallas que ofrecen diferentes funcionalidades y permiten



Fig 3.23: Pantalla de configuración de la aplicación

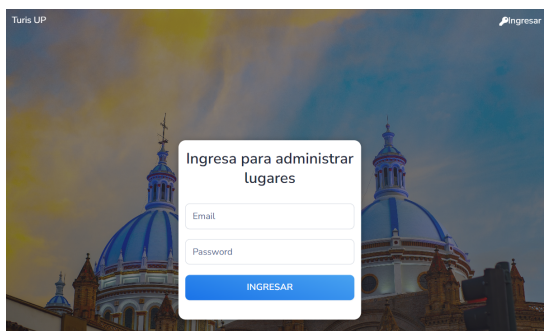
a los usuarios interactuar con el sistema de una manera sencilla. Entre las pantallas principales se encuentran:

1. Pantalla de inicio de sesión (Login): Permite a los administradores autenticarse en el sistema, proporcionando las credenciales correspondientes, Figura 3.24a.
2. Pantalla de organizaciones: Permite a los administradores visualizar las organización que puede administrar, Figura 3.24b.
3. Pantalla de regiones: Proporciona una vista general de las diferentes regiones que son administradas por la organización a la que pertenece el usuario, Figura 3.24b.
4. Pantalla de recursos turísticos: Permite a los administradores gestionar una lista de recursos turísticos disponibles sobre las distintas organizaciones a las que pertenece. El administrador puede aceptar, rechazar, archivar o modificar la información de cada recursos turístico, Figura 3.24d.

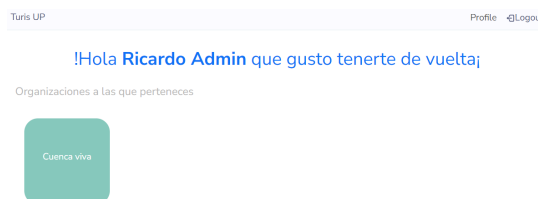
Si se desea obtener una visión más detallada de cada una de las pantallas mencionadas anteriormente, así como de todas las funcionalidades y características de la aplicación web, se puede consultar el Manual de Usuario en la sección de anexos. El manual proporciona una guía completa y detallada sobre cómo utilizar la aplicación web, explicando paso a paso cada funcionalidad.

3.5.2. Backend

El backend es la parte en donde se detalla la lógica de negocio, es decir todas las funcionalidades que requiere el cliente además de interactuar con bases de datos y/o repositorios de información, verificar maniobras de sesiones de usuarios y proveer una API para la comunicación con el cliente.



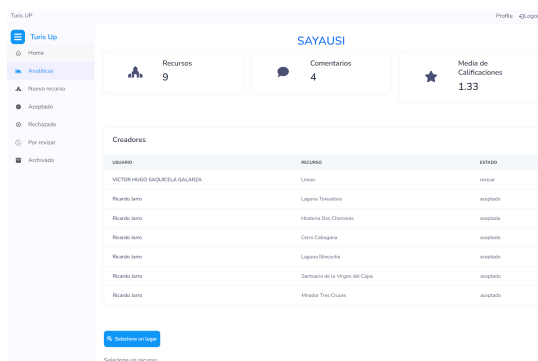
(a) Pantalla login



(b) Pantalla Organizaciones



(c) Pantalla Regiones



(d) Pantalla Recursos Turísticos

Fig 3.24: Pantallas de aplicación web

La sección del backend de la aplicación se construyó utilizando una arquitectura basada en microservicios tal como se muestra en la Figura 3.14, está compuesta por 7 microservicios, cada uno con una funcionalidad específica. Esta elección se basó en las múltiples ventajas que ofrece una arquitectura de microservicios, como la escalabilidad, la flexibilidad, la integración y la facilidad de mantenimiento. Los microservicios se desarrollaron utilizando el lenguaje de programación Java, junto con el framework Spring Boot. Las versiones de las herramientas usadas en común en todos los microservicios se detallan en la Tabla 3.7.

Herramienta	Versión
Java	11.0.7
Spring boot	2.4.5
Maven	3.9.1
IntelliJ IDEA EDU	22.2.2

Table 3.7: Versiones de herramientas comunes de los microservicios

A continuación, se destacarán los aspectos más relevantes en el desarrollo de los microservicios en este proyecto. Algunos microservicios de mayor relevancia serán abordados con mayor

detalle debido a su importancia estratégica. Sin embargo, es importante tener en cuenta que todos los microservicios, incluyendo su código fuente y configuraciones, están almacenados en un repositorio de Git⁹.

Microservicio Configuración

También conocido como Config Server este microservicio es responsable de gestionar y proporcionar la configuración necesaria para los demás microservicios. Actúa como un servidor centralizado de configuración, permitiendo que los microservicios obtengan la configuración necesaria de manera dinámica y actualizada. La configuración reside en los archivos Bootstrap YAML, almacenan las propiedades y valores de configuración, contienen información como la configuración de bases de datos, las claves de API, las URL de servicios externos, entre otros. Este microservicio utiliza la configuración almacenada en GitHub para proporcionar la configuración necesaria que permite un almacenamiento centralizado y controlado de la configuración.

Microservicio de Registro

Microservicio de registro o Eureka Server actúa como un repositorio centralizado donde los demás microservicios pueden registrarse y descubrirse entre sí de manera dinámica, facilitando la comunicación y la colaboración entre los diferentes microservicios, lo que permite tener un entorno ágil y escalable.

Microservicio de Gateway

El Gateway actúa como punto de entrada único para todas las solicitudes provenientes tanto de la aplicación móvil como web hacia los microservicios subyacentes. La configuración del Gateway se basa en patrones de enrutamiento y filtrado a través de configuraciones de archivos YAML.

Microservicio Autenticación

Este microservicio es responsable de autenticar a los usuarios que intentan acceder a la aplicación. En el sistema se tiene dos posibilidades de autenticación dependiendo desde donde se acceda: si se accede desde la aplicación web que está diseñada para los administradores, se debe proporcionar un correo electrónico y una contraseña; si se accede desde la aplicación web la autenticación se realiza mediante Google.

⁹<https://github.com/Argos98p/microservicios>

El flujo de acceso mediante la aplicación web se visualiza en la Figura 3.25. como se puede observar los parámetros de correo y contraseña en la solicitud post son obligatorios, estas credenciales son buscados en la base de datos y si son encontrados se creará un token JWT (Json Web Token) para el usuario y sera devuelto en la petición, caso contrario se retornará un mensaje de error.

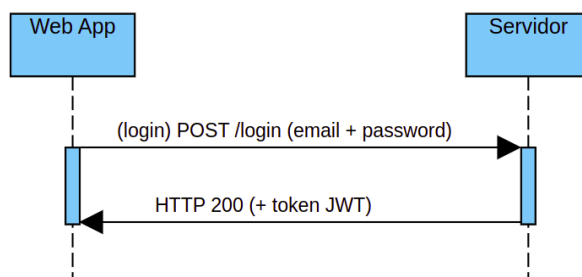


Fig 3.25: Flujo de acceso desde la aplicación web

El acceso desde la aplicación móvil difiere debido a que se lo realiza mediante Google que utiliza el protocolo OAuth2. Este proceso es más seguro que simplemente compartir su nombre de usuario y contraseña de Google, ya que la aplicación no tiene acceso a sus credenciales de inicio de sesión.

El proceso de autenticación de Google OAuth utiliza tokens de acceso que se generan después de que el usuario otorga permiso a la aplicación para acceder a sus datos. Estos tokens de acceso son temporales y solo se pueden usar para acceder a la información específica autorizada por el usuario, por ende cada vez que se obtiene el token por parte de Google se envía este token para ser validado en el servidor y obtener el token propio de la aplicación, Figura 3.26.

Además, el servicio de autenticación de usuarios también tiene otras funcionalidades importantes. Por ejemplo, es capaz de validar los tokens de acceso que recibe de los usuarios y verifica si el usuario tiene los permisos necesarios para realizar la operación solicitada. También puede generar nuevos tokens de acceso cuando sea necesario, siempre y cuando la autenticación sea exitosa.

El microservicio de autenticación de usuarios puede proporcionar una interfaz RESTful para recibir solicitudes de autenticación de los usuarios y devolver respuestas JSON que contengan el token de acceso y otros detalles necesarios, permitiendo que los usuarios puedan interactuar con el servicio de manera sencilla y eficiente.

Es importante destacar que el microservicio de autenticación de usuarios puede integrarse con otros microservicios de la aplicación para proporcionar seguridad y autorización en toda la

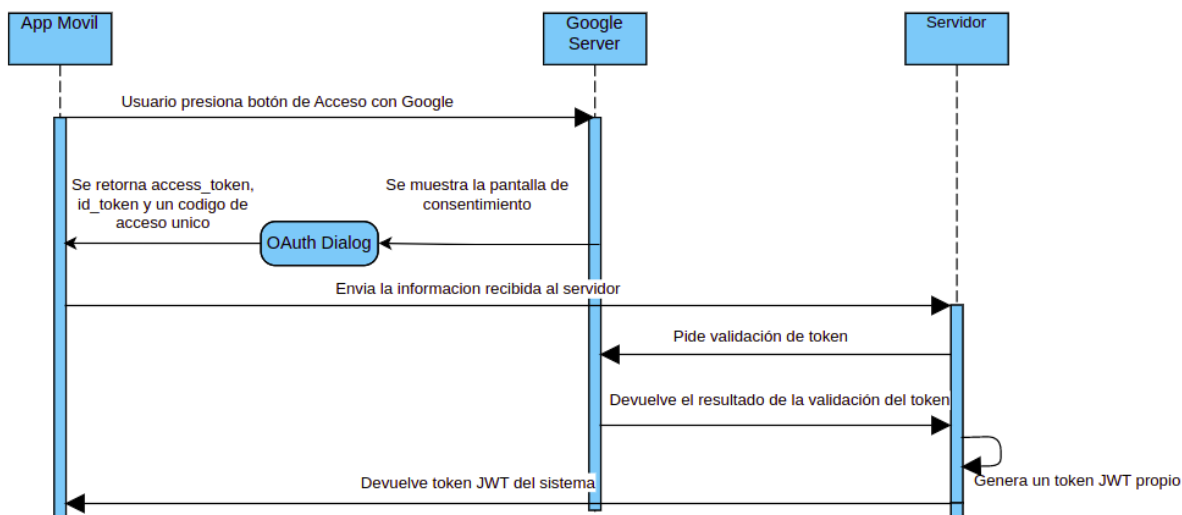


Fig 3.26: Flujo de acceso desde la aplicación móvil

aplicación, asegurando que los usuarios sólo puedan acceder a los recursos y servicios para los que tienen permisos adecuados, protegiendo así la integridad y privacidad de la información en la aplicación.

Este microservicio, mostrado también en la Figura 3.14 del diagrama de la arquitectura, está conectado a una base de datos PostgreSQL. Esta base de datos utiliza un modelo entidad-relación para almacenar la información y credenciales de los usuarios, cuenta con una tabla con 9 atributos, Figura 3.27. Los atributos, `imageUrl`, `authProvider`, `providerId` son atributos puntuales para el login mediante OAuth ya que son provistos por la API de Google por otro lado, el campo `password` es para los usuarios administradores únicamente.

Como se mencionó, este servicio usa una base de datos relacional debido a la naturaleza delicada de la información que maneja, como las contraseñas de los usuarios. Además, se utiliza Spring Security, que está diseñado para trabajar con bases de datos, lo que permite garantizar las autorizaciones adecuadas.

Por otro lado, en el desarrollo de la ontología se añadió una clase específica para representar la información no sensible relacionada con los usuarios de la base de datos. Esta clase permite mapear y transferir dicha información a la ontología, separándola de los datos sensibles almacenados en la base de datos relacional. De esta manera, se dispone de la información del usuario en la ontología para establecer relaciones con otros datos.

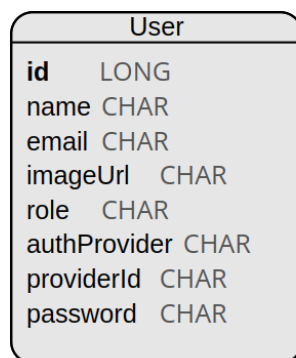


Fig 3.27: Modelo usuarios

Microservicio de Media

El microservicio de media se encarga de subir imágenes y vídeos a Facebook. Si bien es cierto Facebook no está diseñado para ser un repositorio de imágenes puede usarse con ese fin aunque ciertamente con limitaciones. Además usar Facebook como repositorio de imágenes puede traer ventajas como :

- Costo: A diferencia de bases de datos, utilizar Facebook como un repositorio de imágenes puede ser gratuito o muy económico en comparación con el costo de implementar y mantener una base de datos orientada a imágenes.
- Accesibilidad: Como Facebook es una plataforma muy popular y utilizada por muchas personas, las imágenes alojadas en su plataforma pueden ser fácilmente accesibles y compartidas a través de diferentes dispositivos y plataformas.
- Fácil integración: Facebook proporciona una API que permite integrar fácilmente imágenes en el sistema. Con la API, se puede recuperar imágenes alojadas en Facebook y mostrarlas en cualquier sistema.
- Ahorro de almacenamiento: Al utilizar Facebook como repositorio de imágenes, se ahorra espacio de almacenamiento en el sistema.

La comunicación con Facebook se realiza mediante el API Graph de Facebook. La API Graph es la forma principal de ingresar y extraer datos en la plataforma de Facebook, es una API basada en HTTP, que permite a las aplicaciones consultar datos, publicar nuevas historias, administrar anuncios, subir fotos y realizar una amplia variedad de tareas de manera programática (Meta, n.d.).

Para que la API Graph funcione, es necesario contar con una cuenta de desarrollador y registrar una aplicación en Facebook. La cuenta de desarrollador puede ser solicitada en Meta for

developers ¹⁰. Una vez que la cuenta esté activa, se debe registrar la aplicación en Facebook. En la Figura 3.28 se muestra la información relevante de la aplicación registrada, incluyendo un identificador que será utilizado en el servidor en conjunto con la API Graph.

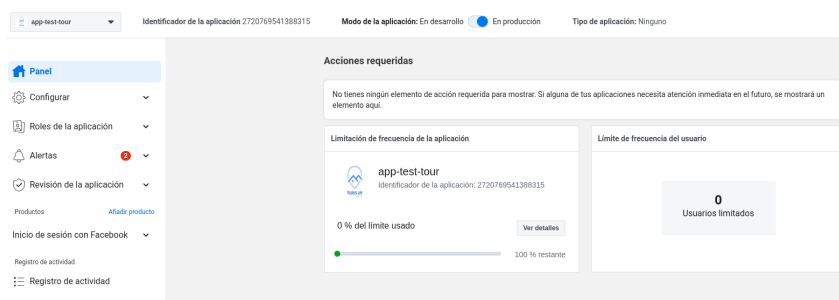


Fig 3.28: Información de la aplicación

El token de acceso es obligatorio para poder publicar en la página de Facebook creada para este proyecto, la página creada se la puede encontrar bajo el nombre de Turis Cuenca ¹¹ Figura 3.29. El token de acceso se obtiene con la herramienta web API Graph Explorer añadiendo los permisos para publicar. 3.30.

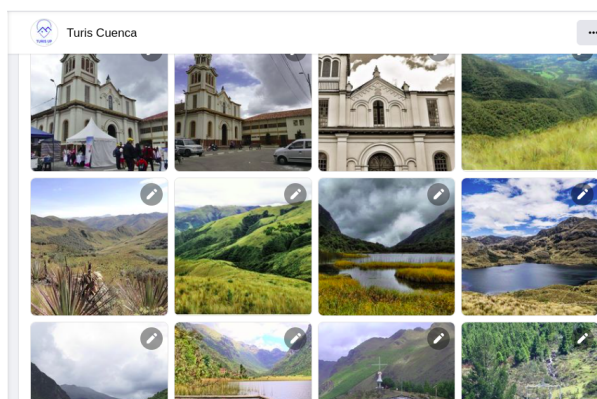


Fig 3.29: Imágenes de recursos subidos a Facebook

Para vincular el microservicio con API Graph es posible realizarlo únicamente con peticiones HTTP como menciona la documentación, sin embargo por facilidad y simpleza se utilizo RestFB, una biblioteca de Java que permite interactuar con la API de Facebook. Esta biblioteca proporciona una forma sencilla y elegante de acceder y manipular datos de Facebook, como perfiles de usuario, páginas, publicaciones, comentarios, etc. En este caso solo se utilizaron los métodos para subir imágenes y vídeos a la página Turis Cuenca Figura 3.31. Para el funcionamiento del cliente de API Graph se debe proporcionar el access token, page id (id de

¹⁰<https://developers.facebook.com/async/registration>

¹¹<https://www.facebook.com/turisUp>

la página) y el application id los cuales fueron añadidos en el fichero "application.properties" del microservicio Figura 3.30.

```
fb.token=EAAmqhoYHuBsBAEpkZAF9zRXDUirL6X0ypyg2Va0CQofVAdpNz85vCUwz8s1wZC0LHZA7QD2wf2WLzs1Pudpeh...
fb.appId=a368875014f7654717a2e1f24c07b98c
fb.pageId=165980483492633
```

Fig 3.30: Atributos necesarios para RestFB

```
public static String UploadVideo(String filePath, String name) throws IOException {
    InputStream in2 = new FileInputStream(filePath);
    GraphResponse response = facebookClient.publish(connection: PAGEID + "/videos", GraphResponse.class,
        BinaryAttachment.with(name, IOUtils.toByteArray(in2)),
        Parameter.with(name: "description", value: ""));
    return response.getId();
}

3 usages
public static String UploadImage(String filePath) throws FileNotFoundException {
    InputStream is = new FileInputStream(filePath);
    FacebookType publishVideoResponse = facebookClient.publish(connection: "me/photos", FacebookType.class,
        BinaryAttachment.with(filePath, is),
        Parameter.with(name: "description", value: ""));
    return publishVideoResponse.getId();
}
```

Fig 3.31: Métodos para subir imágenes y vídeos a Facebook con RestFB

Microservicio de RDF

Este microservicio ha sido desarrollado con el propósito de simplificar las interacciones con una RDF Store, en particular TBD que utiliza Apache Fuseki. Su objetivo es permitir la lectura y escritura de los recursos necesarios para la aplicación móvil, como usuarios, regiones y organizaciones, de forma centralizada. Gracias a este microservicio, se logra establecer una comunicación eficiente con el triplestore, lo que facilita la gestión de la información y su posterior utilización en la aplicación.

Inicialmente, el desarrollo de este microservicio se basaba exclusivamente en el uso de peticiones POST HTTP, siguiendo el estándar del Protocolo SPARQL (W3C, 2013). En un principio, la creación o actualización de recursos, usuarios y otros elementos no presentaba grandes inconvenientes: se construían las consultas INSERT o UPDATE como cadenas de texto y se enviaban a Fuseki en el cuerpo de las peticiones HTTP. Sin embargo, a medida que aumentaba el número de consultas, el control se volvía cada vez más complejo de gestionar. La construcción manual de las consultas requería asegurarse de que la sintaxis fuera correcta

y también manejar y formatear las respuestas recibidas. Esto aumentaba la posibilidad de cometer errores en la construcción de las consultas.

Además, cualquier modificación en las consultas, como cambios en una propiedad, implicaba tener que realizar modificaciones en todas las consultas SPARQL en las que aparecía dicha propiedad. Esto generaba una dificultad adicional y aumentaba la posibilidad de errores en el proceso de mantenimiento y actualización del microservicio. Para abordar este inconveniente se optó por usar Apache Jena concretamente el subsistema RDF API. RDF API es una biblioteca para trabajar con datos RDF que permite la creación, manipulación y consulta de datos RDF utilizando una interfaz de programación de aplicaciones (API) en Java. Gracias al uso de RDF API de Apache Jena, el sistema pudo eliminar una gran carga de trabajo en cuanto a la creación y manejo de consultas SPARQL. Con la utilización de esta librería, la creación de consultas se simplificó además de aumentar flexibilidad y escalabilidad en el manejo de los datos, permitiendo que la adición de nuevas propiedades o clases sea más eficiente. En la Figura 3.32 puede apreciarse como fue la creación de un recurso haciendo uso de esta librería.

```
try (Connection myConnection = DBConnection.createConnection()){
    Model myModel = SDJenaFactory.createModel(myConnection);
    myModel.begin();

    Resource placeModel = myModel.createResource( uri: "http://turis-ucuenca#" + newPlace.getId());
    placeModel.addProperty(RDF.type, myModel.getProperty(tp, localName: "Place"));
    placeModel.addProperty(RDF.type, OWL2.NamedIndividual);
    placeModel.addProperty(DC.title, newPlace.getNombre());
    placeModel.addProperty(VCARD4.fn, newPlace.getNombre());
    placeModel.addProperty(DC.description, newPlace.getDescripcion());
    placeModel.addProperty(DC.date, myModel.createTypedLiteral(ZonedDateTime.now( ZoneOffset.UTC ).
        format( DateTimeFormatter.ISO_INSTANT ), typeURI: "http://www.w3.org/2001/XMLSchema#dateTime"));
    placeModel.addProperty(myModel.getProperty(BASE, localName: "#hasStatus"), o: "revisan");
    placeModel.addProperty(myModel.getProperty(BASE, localName: "#hasCategory"), newPlace.getCategoria());
    placeModel.addProperty(DC.creator, myModel.getResource( uri: BASE + "/user/" + newPlace.getUsuarioId()));

    for (String imageId : imageIds) {
        Resource FacebookImageResource = myModel.createResource();
        FacebookImageResource.addProperty(myModel.getProperty(BASE, localName: "#hasFacebookId"), imageId);
        FacebookImageResource.addProperty(myModel.getProperty(BASE, localName: "#hasFacebookImageUrl"), imageId);
        placeModel.addProperty(myModel.getProperty(BASE, localName: "#hasFacebookResource"), FacebookImageResource);
    }

    for (String videoId : videoIds) {
        Resource FacebookResource = myModel.createResource();
        FacebookResource.addProperty(myModel.getProperty(BASE, localName: "#hasFacebookId"), videoId);
        placeModel.addProperty(myModel.getProperty(BASE, localName: "#hasFacebookResource"), FacebookResource);
    }

    placeModel.addProperty(Geo.HAS_GEOMETRY_PROP, myModel.createResource()
        .addProperty(RDF.type, myModel.createResource( uri: geo + "Geometry"))
        .addProperty(Geo.AS_WKT_PROP, WKTLiteralFactory
            .createPoint(newPlace.getCoordenadas().getLongitud(), newPlace.getCoordenadas().getLatitud()));
    myModel.commit();
}
```

Fig 3.32: Creación de un recurso con Apache Jena

Este microservicio no solo facilita la creación y actualización de recursos, sino que también


```
try (Connection myConnection = DBConnection.createConnection()){

    Model myModel = SDJenaFactory.createModel(myConnection);

    Resource oneRoute = myModel.getResource( uri: "http://turis-ucuenca#" + rutaId);
    ruta.put("id", rutaId);
    ruta.put("nombre", oneRoute.getProperty(DC.title).getObject().toString());
    ruta.put("descripcion", oneRoute.getProperty(DC.description).getObject().toString());
    ruta.put("creador", oneRoute.getProperty(DC.creator).getObject()
        .toString().replace( target: "http://turis-ucuenca#", replacement: ""));
    Statement st = oneRoute.getProperty(myModel.createProperty( namespace: "http://turis-ucuenca", localName: "/hasPlaces"));

    Bag placesInRoute = myModel.getBag(st.getObject().asResource());
    NodeIterator iter = placesInRoute.iterator();
    while (iter.hasNext()) {
        RDFNode placeItem = iter.nextNode();
        String placeId = placeItem.toString().replace( target: "http://turis-ucuenca#", replacement: "");
        System.out.println(placeId);
        QueryOptions queryOptions = new QueryOptions();
        queryOptions.setUserId(userId);
        queryOptions.setLugarId(placeId);
        List<PlaceResponse> response = placeService.all(queryOptions);
        System.out.println(response);
        if(!response.isEmpty()){
            PlaceResponse onePlace = response.get(0);
            placesInRouteArray.add(onePlace);
        }
    }
    ruta.put("lugares", placesInRouteArray);
}
```

Fig 3.33: Obtención de una ruta mediante RDF API

proporciona todas las operaciones necesarias de lectura para acceder a la información almacenada en la RDF Store. La obtención de usuarios, organizaciones, regiones, listas de favoritos se realizó de igual forma usando RDF API de Apache Jena de manera directa, ya que son clases sin muchos atributos y resulta más sencillo acceder a esa información invocando directamente al recurso ontológico, por ejemplo en la Figura 3.33 se muestra como fue el código para recuperar una ruta específica.

La obtención de los recursos turísticos presentó un reto mayor, ya que se necesitaban consultas geoespaciales. La opción más adecuada en este caso fue utilizar consultas GeoSparql a través de la librería RDF API, que también permite obtener registros mediante consultas definidas mediante cadenas de texto. Si bien Apache Jena posee una herramienta para funciones GeoSparql desde Java, la programación para la obtención de recursos turísticos iba a ser bastante extensa, con múltiples validaciones y varias llamadas a Fuseki. En cambio, con el uso de consultas SPARQL, se ahorra todo ese trabajo y se podía obtener una respuesta con registros bastante similares al objeto *RecursoTuristico* utilizado en los microservicios y que se devuelve al usuario final.

Fuseki y TBD soporta múltiples funciones para consultas geoespaciales como: "geof:within", "geof:distance", "geof:relate", "geof:area", "geo:crosses", "geo:overlaps", etc. Para este trabajo fueron utilizadas las 2 primeras funciones.

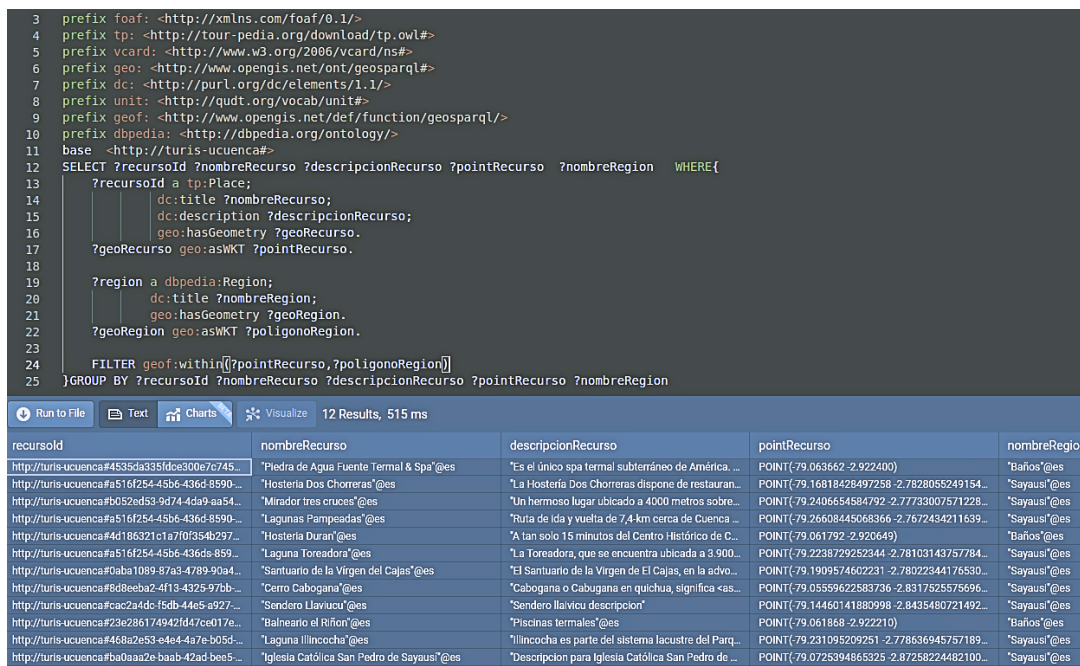


Fig 3.34: Consulta Sparql para la obtención de recursos

La función "within" busca recursos que se encuentran dentro de una región geográfica determinada, es decir, busca los recursos turísticos en las regiones de Sayausi y Baños. Estas regiones, como se mencionó en la sección de creación de la ontología, fueron instanciadas mediante polígonos con formato WKT obtenidos de la página MapIt Global¹². Esta página proporciona el perímetro no solo en formato WKT, sino también en KML y GeoJSON de cualquier tipo de zona administrativa. Para el cantón Cuenca, provee información de todas las parroquias tanto urbanas como rurales¹³. Por lo tanto, si se desea ampliar la aplicación a más zonas, solo se debe instanciar nuevas regiones utilizando la información obtenida de esta página. Una vez definida la región y el recurso turístico, cada uno con su respectiva geometría, se puede utilizar la función "geof:whitin". La Figura 3.34 muestra una consulta en la que se listan los recursos turísticos que están dentro de una región previamente establecida (Baños o Sayausi). Cabe recalcar que aquí los recursos turísticos no tienen ninguna propiedad que los vincule directamente con la región, esta vinculación se realiza a través del filtro con la función "geof:whitin".

Por otro lado, se utilizó la función "geof:distance" para obtener la distancia entre los recursos turísticos y la posición del usuario de la aplicación móvil. Para ello, se obtuvo la posición del usuario a través del GPS de su celular y se construyó una consulta con esta información. La respuesta a la búsqueda se estableció para que se muestre en kilómetros. En la Figura 3.35, se muestra la consulta y su respuesta con una ubicación asignada manualmente como ejemplo.

¹²<https://global.mapit.mysociety.org/>

¹³<https://global.mapit.mysociety.org/area/964356/covers.html>

```

8  prefix unit: <http://qudt.org/vocab/unit#>
9  prefix geof: <http://www.opengis.net/def/function/geosparql/>
10 prefix dbpedia: <http://dbpedia.org/ontology/>
11 base <http://turis-ucuenca#>
12 SELECT ?distance ?recursoId ?nombreRecurso ?descripcionRecurso ?nombreRegion ?distance WHERE{
13     ?recursoId a tp:Place;
14     | dc:title ?nombreRecurso;
15     | dc:description ?descripcionRecurso;
16     | geo:hasGeometry ?geoRecurso.
17     ?geoRecurso geo:asWKT ?pointRecurso.
18
19     ?region a dbpedia:Region;
20     | dc:title ?nombreRegion;
21     | geo:hasGeometry ?geoRegion.
22     ?geoRegion geo:asWKT ?poligonoRegion.
23     BIND(geof:distance(?pointRecurso, "POINT(-79.8844 -2.9085)^^geo:wktLiteral, unit:Kilometer) as ?distance )
24     FILTER geof:within(?pointRecurso,?poligonoRegion)
25 }GROUP BY ?recursoId ?nombreRecurso ?descripcionRecurso ?nombreRegion ?distance

```

distance	recursoId	nombreRecurso	descripcionRecurso
6.82322529	http://turis-ucuenca#23e286174942fd47ce017e...	"Balneario el Riñon"@es	"Piscinas termales"@es
28.59144902	http://turis-ucuenca#468a2e53-e4e4-4a7e-b05d...	"Laguna Illincocha"@es	"Illincocha es parte del sistema lacustre del Parq...
16.8087133	http://turis-ucuenca#ca2a4dc-f5db-44e5-a927...	"Sendero Llaviucu"@es	"Sendero llaviucu descripcion"
27.75932816	http://turis-ucuenca#a516f254-45b6-436d-859...	"Laguna Toreadora"@es	"La Toreadora, que se encuentra ubicada a 3.900...
29.59862714	http://turis-ucuenca#b052ed53-9d74-4da9-aa54...	"Mirador tres cruces"@es	"Un hermoso lugar ubicado a 4000 metros sobre..."
32.6218113	http://turis-ucuenca#a516f254-45b6-436d-8590...	"Lagunas Pampeadas"@es	"Ruta de ida y vuelta de 7,4-km cerca de Cuenca ..."
7.01722371	http://turis-ucuenca#4535da335f0c300e7c745...	"Piedra de Agua Fuente Termal & Spa"@es	"Es el único spa termal subterráneo de América. ..."
22.40831698	http://turis-ucuenca#a516f254-45b6-436d-8590...	"Hostería Dos Chorreras"@es	"La Hostería Dos Chorreras dispone de restauran..."
9.52696626	http://turis-ucuenca#8d8eeba2-4f13-4325-97bb...	"Cerro Cabogana"@es	"Cabogana o Cabugana en quichua, significa <cas..."
24.66041696	http://turis-ucuenca#0aba1089-87a3-4789-90a4...	"Santuario de la Virgen del Cajas"@es	"El Santuario de la Virgen de El Cajas, en la advo..."
6.75580458	http://turis-ucuenca#4d186321c1a7f0f354b297...	"Hostería Duran"@es	"A tan solo 15 minutos del Centro Histórico de C..."
8.1791643	http://turis-ucuenca#ba0aa2e-baab-42ad-bee5...	"Iglesia Católica San Pedro de Sayausi"@es	"Descripcion para Iglesia Católica San Pedro de ..."

Fig 3.35: Consulta Sparql para la obtención de recursos con la distancia a ellos

También otra consulta relevante que se realizó fue para la obtención de un promedio de calificaciones para cada uno de los recursos turísticos (Figura 3.36), esto permitía ordenar los recursos turísticos en base a los mejores calificados.

Las consultas que se presentan aquí son ejemplos simples, pero en la implementación real se desarrolló una consulta más compleja que utiliza estos conceptos. Esta consulta más compleja recupera todos los atributos del recurso turístico, así como las clases asociadas, como la región en la que se encuentra, la organización a la que pertenece, el creador del recurso turístico y los identificadores de las imágenes y vídeos. Esta consulta fue diseñada de tal manera que se puede agregar fácilmente un filtro para buscar instancias específicas de región, usuario u organización. Puedes encontrar esta consulta detallada en la sección anexos, y la Figura 3.37 muestra la información que se obtiene al ejecutarla.

Microservicio de recursos

Este microservicio fue desarrollado para funcionar como punto de conexión entre el aplicativo móvil y los demás microservicios de la arquitectura, se integra con los demás microservicios existentes para hacer uso de sus funcionalidades. Para ello, se estableció una comunicación mediante API Restful. Este microservicio expone una API REST que ofrece una variedad de endpoints para acceder a diferentes recursos turísticos, comentarios, regiones, etc. Todos los

```

15 SELECT ?place (AVG(?rate) as ?rating) WHERE
16 ?place a tp:Place.
17 OPTIONAL{
18 ?comment a :Comment.
19 ?comment :place ?place.
20 ?comment :rating ?rate.
21 }
22 } GROUP BY ?place ?date ORDER BY DESC(?rating)
    
```

Run to File | Text | Charts | Visualize | 18 Results, 615 ms

place	rating
http://turis-ucuenca/lugar/bcbe3d08-2ba5-465a-b960-9f3fab58e5ae	5.0
http://turis-ucuenca/lugar/907fd614-39e6-4c91-ae53-9fd8bba7ea8	4.333333333333333
http://turis-ucuenca/lugar/12283c84-d278-4116-a727-e766ff92244d	4.0
http://turis-ucuenca/lugar/3f047e0f-6a5f-4d47-910f-538af4483298	4.0
http://turis-ucuenca/lugar/3115b6f9-ef5a-4d6c-b41d-867b02468de1	3.3333333333333335
http://turis-ucuenca/lugar/521be204-3fd6-487e-9a11-7167fa30cff5	
http://turis-ucuenca/lugar/d50f1983-be1a-4790-bb8a-8210ff6df24e	

Fig 3.36: Consulta Sparql para la obtención de recursos ordenados segun las valoraciones de los usuarios

Run to File | Text | Charts | Visualize | 11 Results, 2990 ms

org	orgName	creator...	region	regionTit...	place	titulo	categoria	distance	favorito	status	date	descripci...	rate	imagenes	creador	nombre	point	fbIDs
http://turis-...	"Quenca wv...	http://turis...	http://turis...	"Sayasi"	http://turis...	"Limon"	"sin catego...	8.1791643	"no"	"revisar"	2023-04-09...	"Prueba de ...	5.0	"https://ec...	http://turis...	"VICTOR H...	POINT(79...	"http://ww...
http://turis-...	"Quenca wv...	https://ima...	http://turis...	"Sayasi"	http://turis...	"Laguna To...	"laguna"	27.759328...	"si"	"aceptado"	2023-03-26...	"La Forest...	4.5	"https://ec...	http://turis...	"Ricardo Ja...	POINT(79...	"http://ww...
http://turis-...	"Moroni"	https://ima...	http://turis...	"Moroni Sa...	http://turis...	"Tiel mora...	"sin catego...	181.656531...	"si"	"revisar"	2023-03-30...	"Tiel mora...	4.0	"https://ec...	http://turis...	"Ricardo Ja...	POINT(77...	"http://ww...
http://turis-...	"Quenca wv...	https://ima...	http://turis...	"Sayasi"	http://turis...	"Hestia D...	"Hestia D...	22.248311...	"no"	"aceptado"	2023-03-26...	"La Forest...	4.0	"https://ec...	http://turis...	"Ricardo Ja...	POINT(79...	"http://ww...
http://turis-...	"Quenca wv...	https://ima...	http://turis...	"Sayasi"	http://turis...	"Iglesia Cal...	"Iglesia"	8.1791643	"no"	"aceptado"	2023-03-26...	"Descripcio...		"https://ec...	http://turis...	"Ricardo Ja...	POINT(79...	"http://ww...
http://turis-...	"Quenca wv...	http://turis...	http://turis...	"Quenca"	http://turis...	"Prueba 1 p...	"laguna"	0.7760068	"no"	"revisar"	2023-04-18...	"Descripcio...		"https://ec...	http://turis...	"Juan Lazo"	POINT(78...	"http://ww...
http://turis-...	"Quenca wv...	https://ima...	http://turis...	"Sayasi"	http://turis...	"Cerro Cab...	"sendero m...	9.52696626...	"no"	"aceptado"	2023-03-26...	"Cabogana...		"https://ec...	http://turis...	"Ricardo Ja...	POINT(79...	"http://ww...
http://turis-...	"Quenca wv...	https://ima...	http://turis...	"Sayasi"	http://turis...	"Laguna Il...	"laguna"	28.331449...	"si"	"aceptado"	2023-03-26...	"Tiscodia...		"https://ec...	http://turis...	"Ricardo Ja...	POINT(79...	"http://ww...
http://turis-...	"Quenca wv...	https://ima...	http://turis...	"Sayasi"	http://turis...	"Mirador T...	"mirador m...	29.3586527...	"no"	"aceptado"	2023-03-26...	"Un homin...		"https://ec...	http://turis...	"Ricardo Ja...	POINT(79...	"http://ww...
http://turis-...	"Quenca wv...	https://ima...	http://turis...	"Sayasi"	http://turis...	"Santuario...	"Iglesia san...	24.660416...	"si"	"aceptado"	2023-03-26...	"El Santuar...		"https://ec...	http://turis...	"Ricardo Ja...	POINT(79...	"http://ww...
http://turis-...	"Quenca wv...	https://ima...	http://turis...	"Sayasi"	http://turis...	"Sendero LL...	"sendero m...	16.8087133	"si"	"aceptado"	2023-03-26...	"Descripcio...		"https://ec...	http://turis...	"Ricardo Ja...	POINT(79...	"http://ww...

Fig 3.37: Resultado de la consulta general para la obtención de recursos

endpoints se visualizan en la tabla 3.8

Tipo	Path
GET	api/recurso/id
POST	api/recurso
POST	api/recurso/actualizar
POST	api/recurso/historial
POST	api/ruta
GET	api/ruta/id
POST	api/ruta/agregarLugares
POST	api/ruta/eliminarLugares

POST	api/ruta/eliminarRuta
POST	api/comentario
GET	api/comentario/id
POST	api/comentario/borrar
GET	api/favorito
POST	api/favorito
POST	api/organización
GET	api/organización/id
POST	api/organización/actualizar
POST	api/organización/anadirUsuario
POST	api/region/nuevo
POST	api/region/actualizar
POST	api/region/matchOrganizacion
GET	api/usuario/id
POST	api/usuario
POST	api/usuarioGoogle/nuevo

Table 3.8: Endpoints del backend

Servidor Fuseki

La ontología desarrollada se carga en un servidor Sparql utilizando Fuseki, que es parte de la biblioteca Apache Jena. Fuseki se puede instalar en el sistema operativo mediante un archivo específico para Linux o Windows, o simplemente puede ejecutarse utilizando los binarios. Otra opción es utilizar contenedores Docker no oficiales creados por la comunidad. En este trabajo, se optó por utilizar esta última opción.

Para habilitar GeoSPARQL en Fuseki, fue necesario realizar ciertas configuraciones. Sin embargo, se utilizó una imagen de Docker preconfigurada que incluía estas configuraciones para facilitar las consultas geoespaciales. En caso de requerir las configuraciones detalladas, se pueden consultar en el apartado de Anexos.

Una vez completada esta configuración, el servidor Fuseki estuvo preparado para procesar consultas y operaciones GeoSPARQL.

4. Evaluación del prototipo

Al momento del lanzamiento en el mercado de las aplicaciones de software, se espera que tenga un cierto grado de aceptación entre los usuarios, el cual va a depender de las características particulares que cada usuario considere importantes, para esto es fundamental realizar una evaluación de los prototipos antes de lanzar la versión final. Esta evaluación tiene como objetivo garantizar la calidad del prototipo, a través de una combinación de atributos deseada en la aplicación, siendo estos la usabilidad, experiencia de usuario y la satisfacción del usuario. Bajo este contexto, se llevó a cabo la evaluación del prototipo de la aplicación Turis Up, utilizando dos metodologías reconocidas en el campo de usabilidad y experiencia de usuario:

- System Usability Scale (SUS)
- User Experience Questionnaire (UEQ)

El Cuestionario SUS se usó para medir la usabilidad general de la aplicación, el mismo que consiste en una serie de 10 preguntas relacionadas con la usabilidad que miden aspectos como la facilidad de uso, la eficiencia del sistema y la satisfacción del usuario (Devin, 2017). En cambio el cuestionario UEQ se empleó para evaluar la experiencia de usuario al momento de utilizar el prototipo de la aplicación, ya que este cuestionario aborda aspectos emocionales y afectivos, como la atracción, la comprensión, la eficiencia y la satisfacción general del usuario permitiendo tener una visión global de la experiencia y satisfacción del usuario al utilizar el prototipo (Andreas Hinderks, Martin Schrepp, Jörg Thomaschewski, 2018).

Combinando estos dos cuestionarios se obtiene una visión completa de la usabilidad y experiencia de usuario de la aplicación Turis Up. Esta evaluación permite identificar las fortalezas y debilidades del prototipo final, con el objetivo de realizar mejoras y optimizaciones que conduzcan a una versión final satisfactoria y cumpla con las expectativas de los usuarios de la aplicación.

4.1. Ejecución de la evaluación

Esta evaluación tuvo como objetivo recopilar información sobre la experiencia de usuario y la usabilidad de la aplicación móvil Turis Up. Se utilizó un prototipo de la aplicación y se aplicaron los cuestionarios SUS y UEQ a un grupo de 14 participantes, los mismos que fueron estudiantes de último año de la Facultad de Hospitalidad de la Universidad de Cuenca. Dada su formación en hospitalidad y turismo, estos estudiantes contaban con una base sólida de conocimientos que los convirtió en excelentes candidatos para realizar la evaluación y proporcionar su criterio sobre la aplicación móvil. A estos estudiantes se les brindó una charla informativa, dónde se explicó el contexto y problemática sobre la industria y las aplicaciones del turismo dentro del ámbito local. Además, se realizó una exposición en donde se explicaron las funcionalidades más relevantes de la aplicación, las cuales están detalladas en este documento y se proporcionó el respectivo manual de usuario de la aplicación, el cual puede encontrarse en el repositorio de GitHub del proyecto ¹.

La evaluación consistió en llevar a cabo actividades que se expusieron y las cuales estaban detalladas en el manual proporcionado. Estas actividades abarcaban diferentes aspectos de la aplicación y son las siguientes:

- Abrir un recurso turístico.
- Comentar un recurso turístico.
- Crear un nuevo recurso turístico.
- Explorar los recursos turísticos en el mapa y pedir su respectiva ruta desde su ubicación actual.
- Visualizar alguna ruta prediseñada.
- Descargar los datos de la aplicación y utilizar la funcionalidad offline.

La evaluación inicialmente estaba planeada para llevarse a cabo conjuntamente con los estudiantes. Sin embargo, al iniciar las actividades mencionadas anteriormente, surgieron problemas con el servidor. Como la evaluación se llevaba a cabo en el auditorio de la Facultad de Hospitalidad, la red de la universidad bloqueó todas las peticiones hacia el servidor debido a la falta de un DNS y la ausencia del protocolo HTTPS. Debido a este problema y tras evaluar diferentes opciones, se tomó la decisión de que los participantes realizaran las actividades en sus hogares con ayuda del manual de usuario. Esta solución permitiría evitar las restricciones de la red

¹<https://github.com/Argos98p/Tesis>

universitaria y asegurar que cada estudiante pudiera acceder a la aplicación Turis Up sin problemas. Una vez que los estudiantes realizaron las actividades, se aplicó una encuesta utilizando la plataforma Google Forms, en la que se mostraban los cuestionarios SUS y UEQ para recopilar información sobre la usabilidad y experiencia de usuario. Los participantes completaron los cuestionarios proporcionando sus respuestas y comentarios sobre la aplicación.

4.2. Obtención y Análisis de resultados

4.2.1. Cuestionario SUS

Se realizó un análisis completo de los datos obtenidos a través del cuestionario SUS para evaluar la usabilidad de la aplicación móvil Turis Up

Puntuaciones individuales

Se separaron los datos del cuestionario SUS proporcionados por los participantes de la evaluación, los valores obtenidos en esta evaluación siguen la escala de Likert, en la cual se tenía una escala del 1 al 5, la cual se muestra a continuación:

1. Totalmente en desacuerdo
2. En desacuerdo
3. Neutro
4. De acuerdo
5. Totalmente de acuerdo

De esta se obtuvo una tabla con los datos individuales, la cual se muestra en la Figura 4.1.

Encuestado	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
Pregunta 1	2	2	4	3	3	3	4	5	3	1	2	5	5	4
Pregunta 2	5	4	3	2	3	3	5	3	1	3	1	1	1	1
Pregunta 3	3	4	5	4	2	4	4	3	5	2	3	5	5	5
Pregunta 4	5	1	5	5	5	5	1	1	1	1	1	1	1	1
Pregunta 5	1	4	5	4	2	2	2	4	4	2	3	5	5	4
Pregunta 6	4	5	2	3	5	3	2	3	3	4	5	1	1	2
Pregunta 7	3	5	5	5	3	3	5	4	5	5	4	4	5	5
Pregunta 8	3	1	1	1	2	3	1	3	1	1	4	1	1	1
Pregunta 9	5	4	4	5	2	2	3	5	4	3	3	5	5	5
Pregunta 10	1	1	1	2	1	4	4	3	1	1	2	1	1	1

Fig 4.1: Datos individuales del cuestionario SUS

Una vez que se extrajeron los datos, se procedió a calcular el resultado del cuestionario SUS con su respectiva fórmula:

1. Sumar los resultados de las respuestas de los enunciados impares y luego restar 5 al resultado.
2. Sumar los resultados de las respuestas de los enunciados pares y luego restar ese total a 25.
3. Sumar los resultados obtenidos en el literal 1 y 2, y multiplicarlos por 2.5

Al realizar estas operaciones se obtuvieron los resultados descritos en la tabla que se muestra en la Figura 4.2

Encuestado	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
Enunciados impares - 5	9	14	18	16	7	9	13	16	16	8	10	19	20	18
25 - enunciados pares	7	13	13	12	9	7	12	12	18	15	12	20	20	19
multiplicado (2,5)	40	67,5	77,5	70	40	40	62,5	70	85	57,5	55	97,5	100	92,5
Promedio	68,21													

Fig 4.2: Resultados individuales calculados en base a la formula del cuestionario SUS

Una vez obtenidos los resultados individuales, se obtuvo el promedio de los resultados, el cual fue de 68.2, que según la escala de resultados se encuentra dentro del rango aceptable, sin embargo esta muy cercano al rango marginal. Comparando la calificación SUS obtenida con el estándar de las industrias, al tener un 68.2 se asemeja a tener una calificación C "lo está haciendo bien, pero podría mejorar" (Thomas, 2015). La escala SUS se muestra en la Figura 4.3.

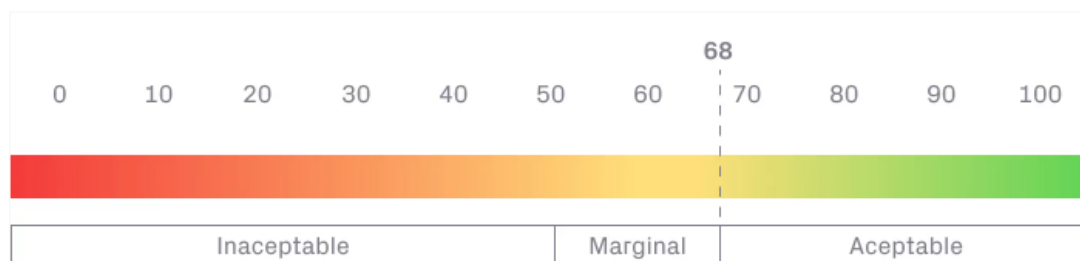


Fig 4.3: Escala de resultados del cuestionario SUS

Con esta calificación la aplicación se considera por encima del promedio y sugiere que los usuarios encuentran la aplicación relativamente fácil de usar y que cumple en gran medida con sus expectativas en términos de utilidad y eficacia. Sin embargo, aún hay margen de mejora para optimizar la usabilidad y brindar una experiencia más fluida y satisfactoria a los usuarios.

4.2.2. Cuestionario UEQ

Se realizó un análisis completo de los datos obtenidos a través del cuestionario UEQ para evaluar la atracción, transparencia, eficiencia, controlabilidad, estimulación y novedad de la aplicación móvil Turis Up. El cuestionario UEQ y la herramienta de análisis de datos se obtuvieron de la página oficial de UEQ ².

Introduciendo los datos en la herramienta de análisis se obtuvieron los siguiente resultados en las 26 preguntas, los cuales se muestran en la Figura 4.4 y 4.5

Item	Mean	Variance	Std. Dev.	No.	Left	Right	Scale
1	↑ 1,1	2,7	1,6	14	desagradable	agradable	Atracción
2	↑ 1,8	1,7	1,3	14	no entendible	entendible	Transparencia
3	↑ 0,9	3,8	2,0	14	creativo	sin imaginacion	Novedad
4	↑ 1,6	2,6	1,6	14	facil de aprender	dificil de aprender	Transparencia
5	↑ 1,4	3,2	1,8	14	valioso	de poco valor	Estimulación
6	↑ 1,2	1,6	1,3	14	aburrido	emocionante	Estimulación
7	↑ 1,4	1,2	1,1	14	no interesante	interesante	Estimulación
8	→ 0,8	2,6	1,6	14	impredecible	predecible	Controlabilidad
9	→ -0,8	6,3	2,5	14	rapido	lento	Eficiencia
10	→ 0,6	4,2	2,1	14	original	convencional	Novedad
11	→ 0,8	3,6	1,9	14	obstructivo	impulsor de apoyo	Controlabilidad
12	→ 0,7	3,3	1,8	14	bueno	malo	Atracción
13	↑ 1,7	2,2	1,5	14	complicado	facil	Transparencia
14	↑ 1,3	1,8	1,3	14	repeler	atraer	Atracción
15	↑ 1,1	2,8	1,7	14	convencional	novedoso	Novedad
16	↑ 1,7	1,9	1,4	14	incomodo	comodo	Atracción
17	↑ 1,7	1,8	1,3	14	seguro	inseguro	Controlabilidad
18	↑ 1,1	2,1	1,5	14	activante	adormecedor	Estimulación
19	↑ 1,3	1,3	1,1	14	cubre expectativas	no cubre expectativas	Controlabilidad
20	→ 0,6	4,3	2,1	14	ineficiente	eficiente	Eficiencia
21	↑ 1,8	1,3	1,1	14	claro	confuso	Transparencia
22	↑ 0,9	2,3	1,5	14	no pragmatico	pragmatico	Eficiencia
23	↑ 1,1	4,0	2,0	14	ordenado	sobrecargado	Eficiencia
24	↑ 2,1	0,7	0,8	14	atractivo	feo	Atracción
25	↑ 1,9	1,4	1,2	14	simpatico	antipatico	Atracción
26	→ 0,4	3,5	1,9	14	conservador	innovador	Novedad

Fig 4.4: Análisis de los datos obtenidos en la encuesta UEQ: Resultados numéricos

El rango de estos valores oscila entre -3 (terriblemente malo) y +3 (extremadamente bueno). Debido al cálculo de medias sobre un rango de diferentes personas con diferentes opiniones y tendencias de respuesta es extremadamente improbable observar valores por encima de +2 o por debajo de -2. Al analizar los resultados, se puede observar que en la Figura 4.6 que la mayoría de las variables evaluadas en el cuestionario tienen puntuaciones superiores a 1, lo cual indica una calificación positiva dentro de la aplicación. Sin embargo, se identificó que la categoría que evalúa la rapidez de la aplicación obtuvo un resultado negativo de -0.8. Esta

²<https://www.ueq-online.org/>



Fig 4.5: Análisis de los datos obtenidos en la encuesta UEQ: Resultados gráficos

evaluación se entiende considerando que los servicios se encuentran en servidores básicos, lo cual puede ocasionar cierta demora al realizar acciones como publicar nuevos recursos turísticos o realizar comentarios, especialmente cuando se incluyen fotos y/o videos.

Por otro lado, es importante destacar que varios aspectos evaluados obtuvieron puntuaciones por encima de 1.5, lo cual es considerado como un rango bastante bueno dentro de la aplicación. Estos resultados indican que existen aspectos destacados en términos de usabilidad y experiencia de usuario que se pueden resaltar como puntos fuertes de la aplicación.

Estos hallazgos brindan información valiosa para mejorar la aplicación móvil Turis Up. Por lo que, recomienda enfocar los esfuerzos en optimizar la rapidez y rendimiento de la aplicación, buscando soluciones para agilizar las acciones que requieren mayor tiempo de respuesta. Además, es importante seguir fortaleciendo los aspectos que obtuvieron puntuaciones positivas, ya que son elementos clave para proporcionar una experiencia de usuario satisfactoria en

el contexto del turismo. En cuanto a las dimensiones medidas por el cuestionario se puede observar en la Figura 4.6 y 4.7 los resultados obtenidos.

UEQ Scales (Mean and Variance)		
Atracción	↑	1,452 0,99
Transparencia	↑	1,732 1,03
Eficiencia	→	0,446 2,07
Controlabilidad	↑	1,143 1,19
Estimulación	↑	1,268 1,09
Novedad	→	0,732 1,91

Fig 4.6: Análisis de las dimensiones de la encuesta UEQ: Resultados numéricos

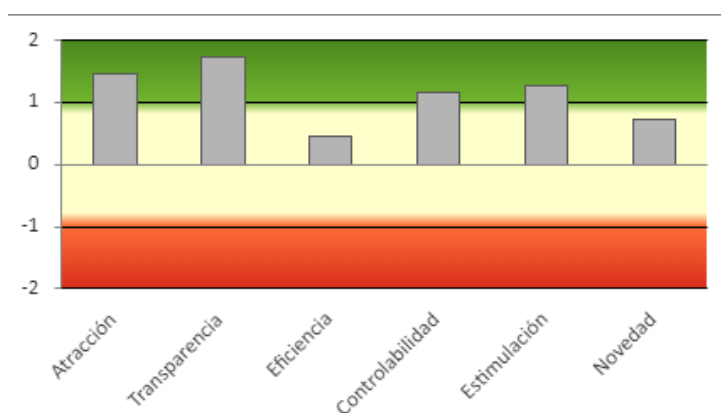


Fig 4.7: Análisis de las dimensiones de la encuesta UEQ: Resultados gráficos

Analizando los resultados, se puede observar que la dimensiones del cuestionario UEQ se clasifican en el siguiente orden de mayor a menor calificación:

1. **Transparencia:** Los participantes encontraron que la aplicación es fácil de familiarizarse y de aprender, lo cual indica que su uso es intuitivo y no requiere un esfuerzo significativo para comprender su funcionamiento.
2. **Atracción:** La impresión general del producto fue altamente positiva, lo cual indica que a los usuarios les gustó la aplicación en términos de diseño y apariencia visual.
3. **Estimulación:** Los usuarios percibieron que la aplicación es estimulante y motivadora, lo cual significa que genera interés y entusiasmo al utilizarla.
4. **Controlabilidad:** Los participantes sintieron que tenían control sobre la interacción con la aplicación, lo cual implica que se les brindó la capacidad de realizar acciones y tomar decisiones dentro de la interfaz de manera efectiva.
5. **Novedad:** La aplicación fue considerada como innovadora y creativa, captando el

interés de los usuarios. Sin embargo, se menciona que se tomaron referencias de otras aplicaciones existentes en el diseño de la interfaz, lo que podría restar originalidad y percepción de novedad.

6. Eficiencia: Aunque se destaca que los usuarios pueden resolver las tareas sin esfuerzos innecesarios, se menciona que la capacidad limitada de los servidores afecta la eficiencia de la aplicación. Esto se evidencia en la demora en la publicación de nuevos recursos turísticos y comentarios, lo cual genera una apariencia de lentitud en la aplicación.

En resumen, los resultados obtenidos indican que la aplicación presenta fortalezas en términos de facilidad de uso, impresión general positiva y capacidad para estimular a los usuarios. Sin embargo, se han identificado áreas de mejora en cuanto a la originalidad y controlabilidad dentro de la aplicación. Asimismo, se ha destacado la necesidad de mejorar la eficiencia en las acciones que dependen de la capacidad de los servidores, especialmente el servidor de Fuseki, y optimizar más las consultas. Estos hallazgos son de gran relevancia para futuras mejoras en la aplicación, ya que permiten enfocar los esfuerzos en los aspectos identificados y ofrecer una experiencia de usuario más satisfactoria. Es importante utilizar esta información para abordar las áreas de mejora y proporcionar novedades en la aplicación, así como optimizar el rendimiento y la capacidad de respuesta de los servidores para garantizar una experiencia fluida y eficiente para los usuarios.

Las 6 dimensiones medida en el cuestionario UEQ se pueden clasificar en dos categorías principales: calidad pragmática y calidad hedónica. La calidad pragmática abarca los aspectos relacionados con la utilidad y eficacia de la aplicación, incluyendo la perspicuidad (claridad de la información), eficacia (capacidad de realizar las tareas de manera efectiva) y fiabilidad (confiabilidad del sistema). Por otro lado, la calidad hedónica se refiere a los aspectos emocionales y subjetivos de la experiencia del usuario, como la estimulación (nivel de interés y emoción generados) y la originalidad (percepción de innovación y creatividad).

En las figuras 4.8 y 4.9 se presentan las medias calculadas para cada uno de estos aspectos de calidad, tanto pragmáticos como hedónicos. Estas medidas representan un promedio de la evaluación realizada por los participantes en relación con estos aspectos.

De los resultados obtenidos mediante el cuestionario UEQ, se puede concluir que el prototipo de la aplicación móvil de turismo ha logrado una aceptación positiva en general por parte de

Pragmatic and Hedonic Quality	
Atracción	1,45
Calidad Pragmática	1,11
Calidad Hedónico	1,00

Fig 4.8: Análisis de las categorías de la encuesta UEQ: Resultados gráficos

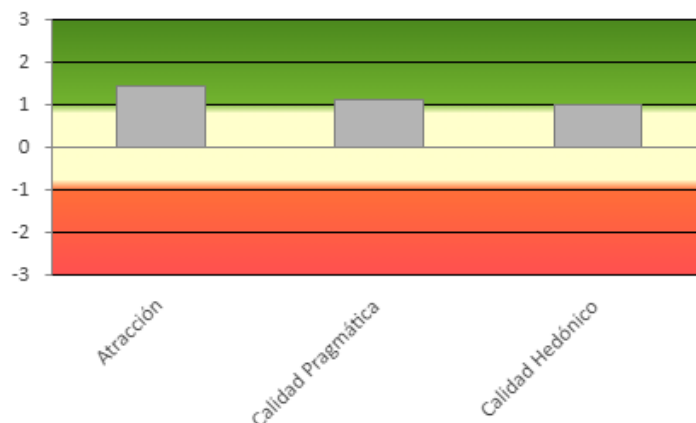


Fig 4.9: Análisis de las categorías de la encuesta UEQ: Resultados gráficos

los usuarios. La aplicación ha sido valorada favorablemente en diversas dimensiones, como la transparencia, atracción, estimulación, controlabilidad y novedad. Estos aspectos indican que la aplicación ha sido percibida como fácil de usar, atractiva, motivadora y capaz de generar interés en los usuarios. Además, se debe destacar que la aplicación ha obtenido una calificación positiva en términos de la calidad pragmática, relacionada con los aspectos funcionales y relacionados con la tarea, ha sido percibida como positiva, lo que indica que la aplicación cumple con los requisitos y expectativas de los usuarios en términos de utilidad y eficacia. Por otro lado, la calidad hedónica relacionada con aspectos emocionales y de satisfacción subjetiva, fue la que obtuvo el menor valor, estando en el mínimo aceptable, lo que confirma el resultado obtenido en el cuestionario SUS, esto sugiere que la aplicación ha logrado generar una experiencia agradable para los usuarios, pero que puede mejorar.

Como puede apreciarse los cuestionarios UEQ y SUS indican pequeñas falencias en el prototipo que pueden ser por los problemas suscitados en la evaluación, ya que los usuarios realizaron las actividades provistas solo desde sus hogares sin tener una supervisión adecuada o alguien que atiende a sus interrogantes. Adicionalmente, los servidores disponibles eran muy básicos, lo que pudo contribuir a la lentitud apreciada por algunos usuarios. Estas complicaciones no afectan al trabajo realizado dentro de este proyecto, pero es importante abordarlas para

optimizar la eficiencia y brindar a los usuarios una experiencia más fluida y sin contratiempos.

5. Conclusiones y trabajos futuros

A continuación se presentan las conclusiones sobre el trabajo con respecto a los objetivos del proyecto para posteriormente analizar trabajos futuros y posibles mejoras.

5.1. Conclusiones

Como resultado final, se ha obtenido un prototipo de la aplicación móvil con información centralizada. Según se observó en la evaluación, el prototipo tiene el potencial de mejorar la experiencia de los turistas en las parroquias de Baños y Sayausí del cantón Cuenca.

El uso de la ontología ha sido sumamente beneficioso y clave en este trabajo para centralizar la información, ya que permite representar de manera estructurada los recursos turísticos, comentarios, usuarios y otros elementos relevantes. Además, la ontología ha facilitado la obtención de recursos turísticos a través de consultas complejas basadas en propiedades y relaciones, utilizando el lenguaje de consulta SPARQL.

Gracias al análisis, revisión y comprensión de la relación entre las aplicaciones móviles y el turismo, se ha proporcionado una visión más clara y detallada de cómo estas dos áreas se entrelazan y se benefician mutuamente. A través de la búsqueda de información para el marco teórico, se pudo identificar el papel crucial que desempeñan las aplicaciones móviles en la industria del turismo, al brindar a los viajeros herramientas y servicios innovadores que mejoran su experiencia de viaje.

Además, la determinación de los requisitos del prototipo a través de la recopilación de funcionalidades esperadas del sistema y el análisis de aplicaciones de turismo existentes en el mercado, sentó las bases para el desarrollo de una solución adaptada a las necesidades y expectativas de los usuarios. Al examinar las características y funcionalidades presentes en las aplicaciones de turismo exitosas, se identificaron elementos clave que se aplicaron en el diseño y desarrollo de la aplicación propuesta, Turis Up. En resumen, la aplicación permite a los usuarios agregar nuevos recursos turísticos geolocalizados, los cuales son gestionados mediante la aplicación web para mantener un control sobre los recursos enviados por los usuarios. Además, los usuarios tienen la posibilidad de comentar y calificar los recursos, y estas funcionalidades

están estrechamente relacionadas con la funcionalidad offline, lo que permite realizar estas acciones en lugares sin conexión a internet y utilizar la aplicación en diferentes ubicaciones, tanto urbanas como rurales. Estas funcionalidades hacen que la aplicación sea una propuesta más competitiva y atractiva en el mercado de las aplicaciones de turismo, al ofrecer una mayor interactividad, personalización y accesibilidad.

Por otro lado, la utilización de microservicios en el backend ha permitido una mayor modularidad y escalabilidad del sistema, facilitando la gestión y el mantenimiento de las funcionalidades. En cuanto al frontend basado en Flutter, ha proporcionado una estructura sólida y flexible para la creación de una interfaz de usuario atractiva y amigable. Todo esto ha sido posible gracias a la metodología ágil Scrum, que ha permitido una gestión dinámica del proyecto, con entregas incrementales y oportunidades constantes de retroalimentación y mejora. En conjunto, este enfoque ha resultado en un prototipo móvil robusto, adaptable y alineado con las necesidades y expectativas de los usuarios.

Finalmente, la evaluación de usabilidad y experiencia de usuario del prototipo ha mostrado una aceptable calidad y satisfacción del producto final. A través de estas pruebas, se ha obtenido información valiosa sobre la interacción de los usuarios con la aplicación móvil y los aspectos a mejorar, siendo la rapidez el punto principal identificado debido a las limitaciones de los recursos y a los problemas suscitados en la evaluación. Estas observaciones serán clave para realizar ajustes y optimizaciones en el prototipo, asegurando una experiencia de usuario más acorde a las expectativas.

5.2. Trabajos futuros

Conforme la aplicación de turismo Turis Up continúe evolucionando y creciendo, hay muchas oportunidades para expandir y mejorar aún más las funcionalidades y la experiencia de usuario de la aplicación, cabe recalcar que la aplicación desarrollada es únicamente un prototipo y existen varias áreas de mejora. Los trabajos futuros podrían concentrarse en varias áreas que enriquezcan la aplicación y brinden a los usuarios una experiencia turística aún más satisfactoria y personalizada. Aquí se nombrarán algunos de los aspectos claves que se podrían mejorar en el futuro para que esta aplicación se convierta en un poderoso aliado para los turistas al momento de viajar a cualquier destino turístico que deseen.

La implementación de un sistema de recomendación es una de las áreas clave que se pueden explorar en trabajos futuros para mejorar la experiencia del usuario en la aplicación. Al utilizar técnicas de análisis de sentimientos en los comentarios realizados por los usuarios sobre cada

recurso turístico, se puede obtener información valiosa sobre sus experiencias y opiniones. Mediante el procesamiento de lenguaje natural y el análisis de sentimientos, es posible identificar patrones y tendencias en los comentarios, lo cual permite comprender mejor las preferencias y gustos individuales de cada usuario. Al recopilar esta información, se puede desarrollar un sistema de recomendación más efectivo y personalizado. Además, se pueden utilizar algoritmos de aprendizaje automático sobre los datos para identificar similitudes entre los comentarios positivos y negativos, y así determinar qué recursos turísticos podrían resultar más atractivos para un usuario en particular. De esta manera, se pueden proporcionar recomendaciones que se ajusten a las preferencias individuales de cada usuario, brindando experiencias turísticas más gratificantes. En este análisis también se puede observar qué emociones generan cada recurso turístico, ya sean positivas o negativas en los usuarios. Esta información es muy valiosa para los administradores de los recursos turísticos, quienes podrán utilizar estos comentarios para mejorar y optimizar sus servicios.

Otro aspecto que se puede implementar en la aplicación es la inclusión de funcionalidades de realidad aumentada, las cuales representan un enfoque prometedor para mejorar la experiencia de los usuarios. Mediante el uso de tecnología de realidad aumentada, se puede proporcionar una capa adicional de información y enriquecimiento visual de los recursos turísticos utilizando la cámara del dispositivo. Esto resulta especialmente útil para los recursos turísticos de carácter histórico, ya que se pueden superponer datos adicionales sobre los recursos en tiempo real. Por ejemplo, cuando los usuarios apunten con la cámara de su dispositivo hacia los recursos históricos, podrían visualizar datos históricos relevantes sobre ese recurso de manera interactiva. Esta funcionalidad mejora la experiencia de usuario y enriquece la exploración de los recursos turísticos, al proporcionar información adicional y contextualizada de manera más inmersiva. Se propone ampliar la base de datos de los recursos turísticos, ya que hasta ahora el enfoque se ha limitado a solo dos parroquias y la aplicación ha sido utilizada por un total de 15 usuarios. Al agregar más datos a la base, se brindará a los usuarios una amplia variedad de opciones para explorar tanto dentro como fuera del cantón. Esto se puede lograr adquiriendo datos de diferentes fuentes y estableciendo colaboraciones con diversos GADs, organizaciones turísticas o instituciones locales. Además, es importante mejorar los servidores para que puedan manejar un mayor número de usuarios, datos y procesamiento de los mismos. Al contar con más usuarios de diferentes regiones, se podrá recopilar más información sobre los recursos turísticos, ya que la aplicación facilita a los usuarios el registro de lugares turísticos. Esto permitirá una rápida expansión de la cantidad y diversidad de recursos turísticos dentro de la aplicación. Asimismo, gracias a la página de administración, se puede garantizar la calidad y

relevancia de la información agregada, lo cual es fundamental para contar con una base de datos de alta calidad.

Otra adición interesante a la aplicación sería la inclusión de eventos locales. Además de mostrar lugares turísticos, la incorporación de información actualizada sobre eventos culturales, festivales, conciertos y otras actividades en cada destino podría brindar a los usuarios una experiencia aún más enriquecedora. Al proporcionar detalles sobre eventos próximos, los viajeros podrán planificar su itinerario de manera más completa, teniendo en cuenta tanto los atractivos turísticos como las oportunidades para participar en emocionantes eventos locales. Esto agregaría un componente adicional de diversión y entretenimiento a la experiencia de viaje, permitiendo a los usuarios sumergirse aún más en la cultura y el ambiente local de cada destino.

En síntesis, los trabajos futuros tienen un gran potencial para mejorar y ampliar aún más la aplicación turística. Existen áreas clave que se pueden abordar, como la mejora de la experiencia del usuario, la adquisición de datos actualizados, la integración de tecnologías emergentes y la expansión de las funcionalidades existentes. Al implementar estos trabajos futuros, la aplicación se volverá más completa, satisfactoria y adaptable a las necesidades cambiantes de los usuarios. Estas mejoras no solo mejorarán la calidad de la experiencia del usuario, sino que también contribuirán al crecimiento y éxito continuo de la aplicación en el ámbito del turismo. Con una planificación adecuada y la ejecución eficiente de los trabajos futuros, se podrá brindar a los usuarios una herramienta aún más valiosa y relevante para explorar el mundo del turismo a nivel nacional.

Bibliografía

- Alejandro, M. A. U., Lastra-Bravo, X. B., & Jaramillo-Moreno, C. (2022). Turismo y aplicaciones móviles. Preferencias de turistas y prestadores de servicios en el cantón Tena, Napo, Ecuador [Number: 1]. *PASOS Revista de Turismo y Patrimonio Cultural*, 20(1). <https://doi.org/10.25145/j.pasos.2022.20.006>
- AltexSoft, E. (2018). Mobile Frameworks Comparison. Retrieved June 12, 2023, from <https://www.altexsoft.com/blog/engineering/xamarin-vs-react-native-vs-ionic-vs-nativescript-cross-platform-mobile-frameworks-comparison/>
- Andreas Hinderks, Martin Schrepp, Jörg Thomaschewski. (2018). User Experience Questionnaire (UEQ). Retrieved June 19, 2023, from <https://www.ueq-online.org/>
- Bechhofer, S. (2009). OWL: Web Ontology Language. In L. LIU & M. T. ÖZSU (Eds.), *Encyclopedia of Database Systems* (pp. 2008–2009). Springer US. https://doi.org/10.1007/978-0-387-39940-9_1073
- Chaves, M., Freitas, L., & Vieira, R. (2012). Hontology: A multilingual ontology for the accommodation sector in the tourism industry.
- Cuenca, M. (2019). Cuenca cuenta con Aplicación Móvil de Turismo. Retrieved April 2, 2023, from <https://www.turismo.gob.ec/cuenca-cuenta-con-aplicacion-movil-de-turismo/>
- Dev, F. (2018). Flutter Documentation. Retrieved June 12, 2023, from [//flutter.dev/](https://flutter.dev/)
- Dev, R. (2020). React Native Documentation. Retrieved June 12, 2023, from <https://reactnative.dev/>
- Dev, X. (2016). Xamarin Documentation. Retrieved June 12, 2023, from <https://dotnet.microsoft.com/es-es/apps/xamarin>
- Devin, F. (2017). Sistema de Escalas de Usabilidad: ¿qué es y para qué sirve? - UXpañol. Retrieved June 19, 2023, from <https://uxpanol.com/teoria/sistema-de-escalas-de-usabilidad-que-es-y-para-que-sirve/>
- DuCharme, B. (2013). Learning SPARQL: Querying and Updating with SPARQL 1.1. Retrieved July 6, 2023, from <https://books.google.com.ec/books?hl=en&lr=&id=j2kXeNeZ00YC&oi=fnd&pg=PR7&dq=Learning+SPARQL&ots=->

I2KfgiiLm&sig=pKlc93WHgSopYLOe_DzQ3VXPI7w&redir_esc=y#v=onepage&q=Learning%20SPARQL&f=false

- Encalada, E. (2017). Las 'apps' facilitan el desarrollo del turismo local. Retrieved January 20, 2022, from <https://www.elcomercio.com/viajar/apps-facilitan-desarrollo-turismo-local.html>
- Encalada, E. (2018). Las 'apps' que ayudan al turista en sus destinos. Retrieved January 20, 2022, from <https://www.elcomercio.com/viajar/apps-ayudan-movilidad-turista-destinos.html>
- Enriquez, J. G., & Casas, S. I. (2013). Usabilidad en aplicaciones móviles [Publisher: Universidad Nacional de la Patagonia Austral Section: Informe Científico Técnico UNPA]. *Informe Científico Técnico UNPA*, 5(2), 25–47. Retrieved June 17, 2023, from <https://dialnet.unirioja.es/servlet/articulo?codigo=5123524>
- Estudio Alfa. (2019). ¿Qué ventajas tienen las apps de turismo para la industria de viajes? | Estudio Alfa. Retrieved June 9, 2023, from <https://estudioalfa.com/que-ventajas-tienen-las-apps-de-turismo-para-la-industria-de-viajes>
- Filofteia, T. M. (2016). MOBILE APPLICATIONS FOR TOURISM. STUDY REGARDING THEIR USE BY ROMANIANS. *Annals - Economy Series*. Retrieved June 18, 2023, from <https://www.semanticscholar.org/paper/MOBILE-APPLICATIONS-FOR-TOURISM.-STUDY-REGARDING-BY-Filofteia/7b6477ca6b0140dae12e37c81c6433ad935ac31c>
- Gazzè, D., Lo Duca, A., Marchetti, A., & Tesconi, M. (2015). An overview of the tourpedia linked dataset with a focus on relations discovery among places. *Proceedings of the 11th International Conference on Semantic Systems*, 157–160. <https://doi.org/10.1145/2814864.2814876>
- Hardt, D. (2012). *The OAuth 2.0 Authorization Framework* (Request for Comments RFC 6749) [Num Pages: 76]. Internet Engineering Task Force. <https://doi.org/10.17487/RFC6749>
- Hernández, M. (2018). Desmitificando las similitudes entre SPARQL y SQL. Retrieved July 6, 2023, from <https://base22.com/es/blog-es/desmitificando-las-similitudes-entre-sparql-y-sql/>
- Inversini, A., Rega, I., & Gan, S. W. (2020). E-Tourism as a Tool for Socio-economic Development. In Z. Xiang, M. Fuchs, U. Gretzel, & W. Höpken (Eds.), *Handbook of e-Tourism* (pp. 1–16). Springer International Publishing. https://doi.org/10.1007/978-3-030-05324-6_108-1

- IONOS. (2020). Flutter: Introducción al framework multiplataforma. Retrieved June 12, 2023, from <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-flutter/>
- Jahmani, A., Abokhoza, R., Zghyer, R. N., & Jawabreh, O. (2020). THE INFLUENCE OF TRAVELER REVIEWS ON MOBILE APPLICATIONS ON TRAVEL DECISION-MAKING TO DUBAI. Retrieved June 18, 2023, from <https://www.semanticscholar.org/paper/THE-INFLUENCE-OF-TRAVELER-REVIEWS-ON-MOBILE-ON-TO-Jahmani-Abokhoza/b92d99b4f2e149b1589925cce7c3fe70add29d12>
- James, E. (2022). Flutter VS Xamarin VS React Native: Which One to Choose in 2022? Retrieved June 12, 2023, from <https://blog.devgenius.io/flutter-vs-xamarin-vs-react-native-which-one-to-choose-in-2022-48368604be8>
- Jiménez, M. Á. S., & Ripoll, R. R. (2017). Análisis de las aplicaciones móviles de destinos turísticos y su accesabilidad. *Teoría y Praxis*, 3–26. <https://doi.org/10.5377/typ.v0i31.6379>
- Lee, C.-I., Hsia, T.-C., Hsu, H.-C., & Lin, J.-Y. (2017). Ontology-based tourism recommendation system. *2017 4th International Conference on Industrial Engineering and Applications (ICIEA)*, 376–379. <https://doi.org/10.1109/IEA.2017.7939242>
- Lozano, A. (2020). Ontologías en la Web semántica [Accepted: 2020-06-09T23:05:33Z]. Retrieved June 5, 2023, from <http://148.202.167.116:8080/xmlui/handle/123456789/3393>
- Lozano, S., & Anaya, R. (2014). Caso de estudio sobre apropiación de SCRUM en empresas que han adoptado CMMI. Retrieved June 19, 2023, from <https://www.semanticscholar.org/paper/Caso-de-estudio-sobre-apropiaci%C3%B3n-de-SCRUM-en-que-Lozano-Anaya/b40775bcde63a3b8c7c93029e06cf5dbe3c8fe7e>
- Lucía, L. B., Zoe, M. V., & Macías Plaza Nila. (2020). Turismo comunitario en Ecuador: Apuntes en tiempos de pandemia. *Revista de Ciencias Sociales (Ve)*.
- Medina, K. R. (2022). Estadísticas de la situación Digital en Ecuador 2021-2022. Retrieved September 16, 2022, from <https://branch.com.co/marketing-digital/estadisticas-de-la-situacion-digital-en-ecuador-2021-2022/>
- Meta. (n.d.). Información general - API Graph. Retrieved April 24, 2023, from <https://developers.facebook.com/docs/graph-api/overview/>
- Ministerio de Turismo del Ecuador. (2020). Plan Estratégico Institucional 2019-2021. <https://www.turismo.gob.ec/wp-content/uploads/2021/05/plan-estrategico-institucional.pdf>

- Missikoff, M., Werthner, H., Höpken, W., erba, M., Fodor, O., Formica, A., & Taglino, F. (2003). Harmonise—Towards Interoperability in the Tourism Domain. https://doi.org/10.1007/978-3-7091-6027-5_7
- Moreno Reyes, F. (2015). *Las redes sociales en internet como herramienta de estimación del voto* (<http://purl.org/dc/dcmitype/Text>). Universidad Complutense de Madrid. Retrieved July 10, 2023, from <https://dialnet.unirioja.es/servlet/tesis?codigo=98662>
- Newman, S. (2015). *Building Microservices* [Google-Books-ID: 1uUDoQEACAAJ]. O'Reilly Media.
- Nexotur. (2019). El Turismo representa el 10,4% del PIB mundial. Retrieved July 4, 2023, from <https://www.nexotur.com/noticia/101605/nexotur/el-turismo-representa-el-104-del-pib-mundial.html>
- Ontotext. (n.d.). What are Ontologies? Retrieved May 1, 2023, from <https://www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies/>
- Page, C. M. H., Stephen J. (2014). *The Geography of Tourism and Recreation: Environment, Place and Space* (4th ed.). Routledge. <https://doi.org/10.4324/9780203796092>
- Pérez, C. (2002). Chantal Pérez, Explotación de los corpórea textuales informatizados para la creación de bases de datos terminológicas basadas en el conocimiento. Retrieved July 4, 2023, from <http://elies.rediris.es/elies18/index.html>
- Prantner, K., Ding, Y., Luger, M., Yan, Z., & Herzog, C. (2007). Tourism ontology and semantic management system: State-of-The-Arts analysis. *IADIS International Conference: IADIS*.
- Rouse, M. (2020). Mobile Application. Retrieved June 12, 2023, from <https://www.techopedia.com/definition/2953/mobile-application-mobile-app>
- Sánchez, R. A. (2010). *Hostelería y Turismo. Material Didáctico: Módulo: Recursos Turísticos. Bloque 1* [Google-Books-ID: MX4mQg6rUmUC]. Liber Factory.
- Saura, J. R., Sánchez, P. R. P., & Menéndez, A. R. (2017). Marketing a través de aplicaciones móviles de turismo (m-tourism): Un estudio exploratorio [Publisher: Facultad de Turismo y Finanzas Section: International journal of world of tourism]. *International journal of world of tourism*, 4(8 (Diciembre)), 45–56. Retrieved September 20, 2022, from <https://dialnet.unirioja.es/servlet/articulo?codigo=6300897>
- Sir, M., Bradac, Z., & Fiedler, P. (2015). Ontology versus Database. *IFAC-PapersOnLine*, 48(4), 220–225. <https://doi.org/10.1016/j.ifacol.2015.07.036>

- Soto, L. (2019). 'GoUIO' nominada entre las mejores aplicaciones turísticas del mundo. Retrieved April 2, 2023, from <http://www.quitoinforma.gob.ec/2019/01/22/gouio-entre-las-mejores-aplicaciones-turisticas-del-mundo/>
- Spring boot Dev. (2023). Spring Boot Documentation. Retrieved July 10, 2023, from <https://spring.io/projects/spring-boot>
- Štetić, S., Dario, Š., Aksentijević, J., & Trišić, I. (2021). Mobile apps as a tool for destination management - case study of Belgrade. <https://doi.org/10.31410/tmt.2020.569>
- Suárez, M. (2012). *NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse* [Google-Books-ID: UfbfkgEACAAJ]. IOS Press.
- Tanzu, V. W. (2023). Spring Boot. Retrieved June 14, 2023, from <https://spring.io/projects/spring-boot>
- Thomas, N. (2015). How To Use The System Usability Scale (SUS) To Evaluate The Usability Of Your Website [Section: Terminology]. Retrieved July 20, 2023, from <https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/>
- W3C. (2004). RDF/XML Syntax Specification (Revised). Retrieved June 19, 2023, from <https://www.w3.org/TR/REC-rdf-syntax/>
- W3C. (2009). SPARQL Lenguaje de consulta para RDF. Retrieved July 6, 2023, from <https://skos.um.es/TR/rdf-sparql-query/>
- W3C. (2013). SPARQL 1.1 Protocol. Retrieved May 6, 2023, from <https://www.w3.org/TR/sparql11-protocol/>
- W3C. (2023). RDF 1.2 Schema. Retrieved July 4, 2023, from <https://www.w3.org/TR/rdf12-schema/>
- WTTC. (2020). *Travel & Tourism - Global Economic Impact & Trends 2020* (tech. rep.). World Travel & Tourism Council. <https://www.wttc.org/Portals/0/Documents/Reports/2020/Global%20Economic%20Impact%20Trends%202020.pdf>3D2021-02-25-183118-360