

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

SISTEMA DE RECOMENDACIÓN TURISTICA

SISTEMA DE RECOMENDACIÓN PERSONALIZADA DE
RESTAURANTES

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SOFTWARE

BRYAN STEVEN MANOBANDA CHABLA

bryan.manobanda@epn.edu.ec

DIRECTOR: REGINA MARITZOL TENEMAZA VERA PhD.

maritzol.tenemaza@epn.edu.ec

DMQ, Julio 2024

CERTIFICACIONES

Yo, BRYAN STEVEN MANOBANDA CHABLA declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



BRYAN STEVEN MANOBANDA CHABLA

Certifico que el presente trabajo de integración curricular fue desarrollado por BRYAN STEVEN MANOBANDA CHABLA, bajo mi supervisión.

Regina Maritzol Tenemaza Vera PhD.

DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

BRYAN STEVEN MANOBANDA CHABLA

Regina Maritzol Tenemaza Vera PhD.

DEDICATORIA

Dedico este trabajo a mi familia, por su apoyo incondicional durante este trayecto.

A mis profesores y mentores, por su sabiduría y guía invaluable.

Y a mis compañeros, por compartir este viaje lleno de aprendizaje y colaboración.

Bryan Manobanda

AGRADECIMIENTO

Expreso mis más sinceros agradecimientos a la Doctora Maritzol Tenemaza por su invaluable orientación y apoyo a lo largo de este proyecto. Su profundo conocimiento, experiencia y observaciones fueron fundamentales para el éxito de este proyecto.

Agradezco profundamente a mi familia por su constante apoyo incondicional a lo largo de todas las etapas de mi vida. Han sido mi guía constante y mi mayor fuente de fortaleza durante este camino.

También quiero extender mi gratitud a todos los profesores que han cultivado mi curiosidad y mi deseo de aprendizaje, contribuyendo significativamente a mi desarrollo académico y personal.

A mis amigos y a las personas que he conocido durante este trayecto, les agradezco sinceramente por su apoyo constante y por los valiosos consejos que han enriquecido esta experiencia.

Cada una de estas personas ha sido fundamental para mi crecimiento tanto profesional como personal a lo largo de estos años.

Bryan Manobanda

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	V
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS	IX
RESUMEN.....	X
ABSTRACT	XI
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general	2
1.2 Objetivos específicos.....	2
1.3 Alcance.....	3
1.4 Marco teórico.....	4
Sistema de recomendación	4
Sistema de recomendación turística	5
Sistema de recomendación de restaurantes	6
Metodología aplicada.....	8
Herramientas, frameworks y librerías de desarrollo	10
2 METODOLOGÍA.....	16
2.1 Sprint 0.....	16
Lógica del proyecto.....	17
Arquitectura del proyecto	18
Definición de roles	19
Historias épicas	19
Product Backlog.....	20
Planificación del Release.....	21
2.2 Sprint 1.....	22
Objetivo del Sprint	22
Planeación.....	22
Implementación	24
Revisión.....	28
Retrospectiva.....	30
2.3 Sprint 2.....	31
Objetivo del Sprint	31

Planeación.....	31
Implementación	33
Revisión.....	37
Retrospectiva.....	39
2.4 Sprint 3.....	40
Objetivo del Sprint	40
Planeación.....	40
Implementación	42
Revisión.....	44
Retrospectiva.....	46
2.5 Sprint 4.....	47
Objetivo del Sprint	47
Planeación.....	47
Implementación	48
Revisión.....	50
Retrospectiva.....	52
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	53
3.1 Resultados	53
Producto Final	53
Encuesta de usabilidad.....	59
3.2 Conclusiones.....	61
3.3 Recomendaciones.....	62
4 REFERENCIAS BIBLIOGRÁFICAS.....	63
5 ANEXOS.....	67
ANEXO I. Historias Épicas.....	67
ANEXO II. Historias de usuario del Sprint 1	68
ANEXO III. Historias de usuario del Sprint 2	71
ANEXO IV. Historias de usuario del Sprint 3.....	73
ANEXO V. Historias de usuario del Sprint 4.....	75
ANEXO VI. Mockup de las interfaces de usuario	76
ANEXO VII. Protocolo de evaluación	78
ANEXO VIII. Guía de prueba	82
ANEXO IX. Encuesta de usabilidad	90
ANEXO X. Respuestas encuesta de usabilidad	93

ÍNDICE DE FIGURAS

Figura 1. Marco de trabajo Scrum [22].....	8
Figura 2: Diagrama de flujo del algoritmo híbrido	17
Figura 3. Arquitectura MVC aplicada al proyecto	18
Figura 4. Primera versión de la interfaz de usuario	24
Figura 5. Código para obtener la ubicación del turista	24
Figura 6. Configuración de CORS	25
Figura 7. Respuesta del servidor	25
Figura 8. Habilitar acceso con correo en el servicio de autenticación de Firebase	26
Figura 9. Prueba unitaria desarrollada con Karma y Jasmine	26
Figura 10. Lógica para almacenar datos de un turista nuevo	27
Figura 11. Almacenamiento de preferencias de un turista	27
Figura 12. Gráfico Burndown del Sprint 1.....	30
Figura 13. Cuerpo de solicitud de restaurantes.....	33
Figura 14. Respuesta de los restaurantes dentro de un área	35
Figura 15. Respuesta de los detalles de un restaurante.....	35
Figura 16. Escenario escrito en Gherkin para obtener un restaurante	36
Figura 17. Implementación de escenario para obtener un restaurante	37
Figura 18. Gráfico Burndown del Sprint 2.....	39
Figura 19. Cuerpo de petición para obtener una ruta.....	43
Figura 20. Interfaz gráfica de ruta desde la ubicación hasta el restaurante	43
Figura 21. Proveedores para inicio de sesión.....	44
Figura 22. Gráfico Burndown en el Sprint 3	46
Figura 23. Análisis de sentimiento para un comentario positivo (Solicitud en Postman)..	48
Figura 24. Análisis de sentimiento para un comentario negativo (Solicitud en Postman)	48
Figura 25. Código para obtener los comentarios de los restaurantes	49
Figura 26. Parámetros para realizar análisis de sentimiento	49
Figura 27. Código de lista ordenada de restaurantes en base al análisis de sentimiento	49
Figura 28. Lista ordenada de restaurantes según el puntaje de análisis de sentimiento .	50
Figura 29. Gráfico Burndown del Sprint 4.....	52
Figura 30. Pantalla de registro e inicio de sesión.....	53
Figura 31. Pantalla de recomendaciones	54
Figura 32. Pantalla de recomendaciones “Cerca de Ti”	55
Figura 33. Pantalla “Para ti” cuando un turista tiene preferencias.....	55
Figura 34. Pantallas de información de un restaurante.....	56
Figura 35. Ruta desde la posición del turista a un restaurante seleccionado.....	56
Figura 36. Diversas pantallas de filtros actualizados según el área de búsqueda	57
Figura 37. Resultados de restaurantes al filtrar opciones.....	57
Figura 38. Pantalla para modificar el radio de búsqueda de restaurantes.....	58
Figura 39. Actualización de recomendaciones al cambiar el radio de búsqueda	58
Figura 40: Representación de resultados SUS (tomada de [50])	60
Figura 41. Diseño de la página de inicio (Realizado en Figma).....	76
Figura 42. Diseño de la interfaz de preferencias (Realizado en Figma).....	76
Figura 43. Diseño de la página de recomendaciones (Realizado en Figma)	77
Figura 44. Encuesta datos demográficos	90

Figura 45. Preguntas de 1 al 5 encuesta usabilidad.....	91
Figura 46. Preguntas 6 al 10 encuesta usabilidad	92
Figura 47. Respuestas del nombre de los participantes	93
Figura 48. Respuestas de la edad de los participantes.....	93
Figura 49. Respuesta al sexo de los participantes.....	94
Figura 50. Respuesta al sector donde viven los participantes	94
Figura 51. Respuesta de la pregunta 1 para usabilidad.....	94
Figura 52. Respuesta de la pregunta 2 para usabilidad.....	95
Figura 53. Respuesta de la pregunta 3 para usabilidad.....	95
Figura 54. Respuesta de la pregunta 4 para usabilidad.....	95
Figura 55. Respuesta de la pregunta 5 para usabilidad.....	96
Figura 56. Respuesta de la pregunta 6 para usabilidad.....	96
Figura 57. Respuesta de la pregunta 7 para usabilidad.....	96
Figura 58. Respuesta de la pregunta 8 para usabilidad.....	97
Figura 59. Respuesta de la pregunta 9 para usabilidad.....	97
Figura 60. Respuesta de la pregunta 10 para usabilidad	97

ÍNDICE DE TABLAS

Tabla 1. Roles del equipo Scrum	19
Tabla 2. Historias épicas.....	19
Tabla 3. Product Backlog generado.....	20
Tabla 4. Planificación de los Sprints	21
Tabla 5. Historias de usuario del Sprint 1	22
Tabla 6. Sprint Backlog del Sprint 1.....	23
Tabla 7. Pruebas de aceptación del Sprint 1.....	28
Tabla 8. Product Backlog Adaptado en el Sprint 1	29
Tabla 9. Historias de Usuario del Sprint 2.....	31
Tabla 10. Sprint Backlog del Sprint 2.....	32
Tabla 11. Pruebas de aceptación del Sprint 2	37
Tabla 12. Product Backlog actualizado en el Sprint 2.....	38
Tabla 13. Historias de usuario para el Sprint 3	40
Tabla 14. Sprint Backlog para el Sprint 3.....	41
Tabla 15. Matriz de pesos para la especialidad culinaria	42
Tabla 16. Parámetros para obtener una ruta	43
Tabla 17. Pruebas de aceptación para el Sprint 3.....	44
Tabla 18. Product Backlog actualizado en el Sprint 3.....	45
Tabla 19. Historias de usuario usadas en el Sprint 4.....	47
Tabla 20. Sprint Backlog del Sprint 4.....	47
Tabla 21. Pruebas de aceptación del Sprint 4	50
Tabla 22. Product Backlog actualizado en el Sprint 4.....	51
Tabla 23: Puntaje SUS individual de cada participante.....	60
Tabla 24. Historia Épica CAP01	67
Tabla 25. Historia Épica CAP02	67
Tabla 26. Historia Épica CAP03	67
Tabla 27. Historia de Usuario CAP01-01	68
Tabla 28. Historia de Usuario CAP01-02	69
Tabla 29. Historia de Usuario CAP01-03	70
Tabla 30. Historia de Usuario CAP02-01	71
Tabla 31. Historia de Usuario CAP02-02	71
Tabla 32. Historia de Usuario CAP02-03	72
Tabla 33. Historia de Usuario CAP02-02 actualizada para el Sprint 3.....	73
Tabla 34. Historia de Usuario CAP03-01	73
Tabla 35. Historia de Usuario CAP01-04	74
Tabla 36. Historia de Usuario CAP02-04	75

RESUMEN

Este documento presenta un sistema de recomendación de restaurantes, diseñado para facilitar la toma de decisiones de los turistas mediante la integración de sus preferencias personales y su ubicación geográfica. El sistema emplea un enfoque híbrido que combina técnicas de filtrado colaborativo y basado en contenido para proporcionar las mejores sugerencias de restaurantes.

Las recomendaciones se basan en los establecimientos mejor evaluados por Google, que luego se clasifican mediante un análisis de sentimiento aplicado a los comentarios de los visitantes. Este proceso garantiza que los restaurantes sugeridos sean relevantes para los turistas. La información de los restaurantes se obtiene de la API de Google, asegurando datos actualizadas y disponibilidad de los establecimientos.

El sistema proporciona funcionalidades adicionales, como la posibilidad de ajustar el radio de búsqueda basado en la ubicación del usuario, refinar las opciones según la especialidad culinaria, el rango de precios o las puntuaciones de Google, y acceder al sistema mediante redes sociales. Además, permite obtener información detallada de los restaurantes y acceder a rutas a pie o en vehículo, mejorando significativamente la experiencia del turista.

El desarrollo del sistema se llevó a cabo bajo el marco de trabajo Scrum, permitiendo al equipo adaptarse eficazmente a los cambios y requisitos emergentes. Desde una perspectiva tecnológica, se utilizó Angular y Express para la estructura del sistema. Además, se integró la API de IBM Watson para el análisis de sentimientos, mejorando la funcionalidad y precisión del sistema de recomendación.

PALABRAS CLAVE: recomendación de restaurantes, algoritmo híbrido, filtrado colaborativo, filtrado basado en contenido, análisis de sentimiento, preferencias del turista, API de Google, IBM Watson, Scrum.

ABSTRACT

This document introduces a restaurant recommendation system designed to enhance tourists' decision-making by integrating their personal preferences and geographical location. The system employs a hybrid approach that combines collaborative filtering and content-based techniques to provide the most suitable restaurant suggestions.

The recommendations are based on establishments highly rated on Google, which are then classified through sentiment analysis of user reviews. This process ensures that the suggested restaurants are relevant to tourists. Restaurant information is retrieved from the Google API, ensuring up-to-date data and the availability of the establishments.

The system offers additional functionalities, such as adjusting the search radius based on the user's location, refining options according to culinary specialty, price range, or Google ratings, and accessing the system via social media. Furthermore, it enables users to obtain detailed information about the restaurants and access walking or driving directions, significantly enhancing the tourist experience.

The system development followed the Scrum framework, enabling the team to effectively adapt to changes and emerging requirements. Technologically, Angular and Express were used for system architecture. Additionally, the IBM Watson API was integrated for sentiment analysis, enhancing the functionality and precision of the recommendation system.

KEYWORDS: restaurant recommendation, hybrid algorithm, collaborative filtering, content-based filtering, sentiment analysis, tourist preferences, Google API, IBM Watson, Scrum.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Los sistemas de recomendación son herramientas que analizan grandes volúmenes de datos para identificar patrones y ofrecer recomendaciones personalizadas y relevantes a los usuarios [1], [2]. Entre los tipos de sistemas de recomendación, están los comerciales como Amazon, los de streaming como Netflix, los sociales como Facebook, y los turísticos como Tripadvisor o Airbnb [1], [2], [3].

Los sistemas de recomendación turística tienen aplicaciones diversas, permitiendo la personalización de itinerarios y la sugerencia de destinos exclusivos, teniendo en cuenta factores como el clima y otros aspectos relevantes [4], [5], [6]. Un sistema de recomendación turística, por ejemplo, puede sugerir actividades, lugares de interés y eventos según las preferencias individuales y el comportamiento pasado del usuario [5], [6]. Este tipo de sistemas ayuda a los viajeros a descubrir experiencias únicas y optimizar su tiempo, proporcionando información útil y adaptada a sus gustos y necesidades [4].

Dentro del ámbito turístico, un sistema de recomendación de restaurantes es una aplicación especializada que ofrece una selección personalizada de establecimientos gastronómicos en una zona determinada [7], [8]. Estos programas se basan en criterios de interés definidos previamente por el turista, proporcionando opciones que se alinean mejor con sus preferencias [8], [9], [10]. Utilizan datos como las calificaciones de otros usuarios, las especialidades culinarias, los rangos de precios y la ubicación para ofrecer recomendaciones precisas [11], [12].

Cuando un turista visita un lugar nuevo, suele utilizar aplicaciones para encontrar restaurantes en la zona [12]. Sin embargo, el problema es que estas aplicaciones no personalizan la experiencia del usuario porque desconocen sus intereses y carecen de técnicas para analizar la información proporcionada [6], [9], [13]; esto resulta en sugerencias irrelevantes, obligando a los turistas a analizar múltiples opciones sin acceso a datos esenciales como: horarios de funcionamiento, especialidades culinarias, precios o ubicación [12], [14]. Frecuentemente, los turistas se ven obligados a emprender largas caminatas con la esperanza de encontrar un lugar que cumpla con sus necesidades gastronómicas, haciendo que la búsqueda sea un proceso desafiante y poco eficaz [14].

El componente propuesto presenta un sistema de recomendación de restaurantes a turistas, basado en su ubicación y preferencias individuales, con el fin de facilitar el escoger restaurantes a su gusto. Adicionalmente se ofrece algunas ventajas como: el turista podrá establecer el radio de búsqueda de restaurantes a partir de su ubicación, será posible, refinar las opciones según la especialidad culinaria, el rango de precios o

las puntuaciones otorgadas por Google. Los restaurantes recomendados son aquellos mejor evaluados por Google y luego clasificados mediante un análisis de sentimiento a los comentarios de los visitantes, asegurando que los restaurantes presentados sean relevantes para el turista.

La aplicación está diseñada para ser utilizada en tiempo real, garantizando que los restaurantes sugeridos estén abiertos y que la información proporcionada esté actualizada. Los turistas tienen la libertad de explorar recomendaciones basadas en sus intereses personales o descubrir nuevas opciones en la zona donde se encuentran, resolviendo así el problema de la sobrecarga de información y facilitando una experiencia gastronómica satisfactoria.

1.1 Objetivo general

Diseñar y construir un sistema de recomendación de restaurantes considerando preferencias del turista y su posición geográfica.

1.2 Objetivos específicos

1. Diseñar una interfaz que permita a los turistas especificar sus intereses y requerimientos, mismos que deberán ser almacenados para identificar intereses del mismo turista en futuros accesos a la aplicación.
2. Implementar algoritmos que presenten restaurantes considerando la evaluación realizada por Google y clasificados según el resultado del análisis de sentimiento aplicado a los comentarios de los visitantes sobre cada restaurante a ser considerado. Los restaurantes a proponerse deben considerar las preferencias almacenadas del turista, estar abiertos y encontrarse en un radio específico.
3. Señalar la ruta para que el turista pueda ir desde su ubicación actual hasta el restaurante escogido, incluyendo el tiempo estimado para llegar, sea a pie o en automóvil.
4. Permitir a los turistas ajustar el radio de búsqueda, ampliando o reduciendo el área para encontrar restaurantes que se adapten mejor a sus preferencias y ubicación actual.

1.3 Alcance

El alcance de este componente es el desarrollo de un sistema de recomendación de restaurantes que ofrezca una experiencia gastronómica adaptada a los intereses y preferencias del turista. Además, el sistema proporcionara una interfaz intuitiva que permita al turista indicar sus preferencias, acceder a información detallada sobre los restaurantes, explorar rutas y recibir recomendaciones personalizadas.

Al finalizar su desarrollo, el sistema de recomendación de restaurantes contará con las siguientes características:

- **Interfaz de Navegación:** Permitir al turista navegar a través de las funcionalidades e indicar sus preferencias como: la evaluación del restaurante, el rango de precios y las especialidades culinarias. Las preferencias del turista serán almacenadas para futuras consultas.
- **Recomendaciones Personalizadas:** Presentación de los mejores restaurantes evaluados por Google y clasificados mediante un análisis de sentimiento basado en los comentarios de los visitantes. Las recomendaciones se ajustarán a las preferencias almacenadas del turista, asegurando que los restaurantes sugeridos estén abiertos y ubicados dentro de un radio adecuado según la ubicación actual del turista.
- **Información Detallada:** Acceso a información relevante sobre cada restaurante, incluyendo: horarios de funcionamiento, nombre, número de contacto, página web, especialidades culinarias, precios, servicios y ubicación.
- **Registro e Inicio de Sesión:** Posibilidad de registrarse e iniciar sesión usando métodos de redes sociales como Facebook o Gmail.
- **Búsqueda variable:** Opción de cambiar el radio de búsqueda, permitiendo ampliarlo o disminuirlo según la posición actual del turista.
- **Rutas de Navegación:** Visualización de las rutas a pie o en vehículo desde la posición del turista hasta el restaurante elegido.

El desarrollo se llevará a cabo utilizando el marco de trabajo Scrum, el cual permitirá una gestión eficiente del proyecto y una adaptación continua a las necesidades del turista. Cada sprint, con una duración de dos semanas, presentará una versión del producto basada en una lista de requisitos priorizados y ajustados en conjunto con el turista.

Además, se realizarán pruebas continuas y revisiones de retroalimentación al final de cada sprint para mejorar la calidad y funcionalidad de la aplicación.

Este componente se centrará exclusivamente en la recomendación de restaurantes y no incluirá otros servicios turísticos ni se involucrará en transacciones comerciales o reservas para los turistas. Además, la actualización y disponibilidad de datos se basará en la información proporcionada por la interfaz de programación de aplicaciones (API) de Google.

1.4 Marco teórico

Sistema de recomendación

Los sistemas de recomendación son herramientas diseñadas para personalizar la experiencia del usuario en diversas plataformas web y aplicaciones [1], [3], [13]. Estos sistemas recopilan datos sobre las preferencias de los usuarios mediante valoraciones explícitas y datos implícitos derivados de su comportamiento en línea [1], [2], ofreciendo recomendaciones personalizadas que facilitan la toma de decisiones del usuario [13].

El propósito principal de los sistemas de recomendación es aumentar las ventas y mejorar la satisfacción del usuario mediante recomendaciones personalizadas [2]. Los objetivos operacionales y técnicos incluyen [2]:

- **Relevancia:** Proporcionar ítems que sean de interés para el usuario, aumentando la probabilidad de que los consuma.
- **Novedad:** Recomendar productos que el usuario no haya visto anteriormente para mantener su interés y fomentar la exploración de nuevos productos.
- **Serendipia:** Ofrecer recomendaciones inesperadas que sorprendan positivamente al usuario, revelando nuevos intereses potenciales y aumentando la diversidad de consumo.
- **Incrementar la diversidad de recomendaciones:** Sugerir una variedad de ítems para aumentar las probabilidades de que el usuario encuentre algo de su agrado, evitando la monotonía de recomendaciones demasiado similares.

Los sistemas de recomendación operan identificando patrones y dependencias significativas entre las actividades de los usuarios y los productos disponibles. Estas dependencias se analizan a distintos niveles, ya sea entre categorías de productos o

ítems específicos, y se aprenden mediante algoritmos que procesan grandes volúmenes de datos [1], [2]. Los principales tipos de sistemas de recomendación incluyen:

- **Sistemas que aplican técnicas de Filtrado Colaborativo:** Utilizan las valoraciones y comportamientos de múltiples usuarios para predecir las preferencias de otros usuarios [2], [3]. Estos modelos agrupan a usuarios con intereses similares y basan las recomendaciones en los gustos colectivos del grupo [1], [13].
- **Sistemas Basados en Contenidos:** Se centran en las características de los productos [2], analizando las propiedades de los ítems que el usuario ha valorado o accedido previamente para recomendar productos con atributos similares [1], [3], [13].
- **Sistemas Basados en Conocimiento:** Generan recomendaciones en función de requisitos específicos del usuario [2], [13] y combinan esta información con el conocimiento del dominio para ofrecer recomendaciones precisas [1].
- **Sistemas Basados en Información Demográfica:** Incorporan datos adicionales como información temporal, geográfica, social o de red para mejorar la precisión y relevancia de las recomendaciones [1], [2].
- **Sistemas Híbridos:** Combinan varias técnicas de recomendación para aprovechar las ventajas de cada una y mitigar sus desventajas [1], [3], [13]. En [1], [2] ejemplifica que los métodos de filtrado colaborativo pueden sufrir problemas con ítems nuevos, ya que no pueden recomendar artículos sin valoraciones previas. Esto no limita a los enfoques basados en contenido, ya que la predicción para nuevos ítems se basa en sus características descriptivas que generalmente están disponibles. Al combinar dos o más técnicas básicas de recomendación, se pueden crear sistemas híbridos que superen estas limitaciones [1], [2].

Los sistemas de recomendación se utilizan en distintos campos, desde el comercio electrónico hasta los medios de comunicación [3]. Una de las áreas donde estos sistemas han demostrado ser particularmente útiles es en el turismo [3], [12].

Sistema de recomendación turística

Un sistema de recomendación turística es una aplicación especializada que proporciona sugerencias personalizadas para destinos turísticos, actividades y experiencias de viaje a los usuarios [3], [4], [15]. Estos sistemas recopilan y analizan una variedad de datos

sobre los usuarios, como sus preferencias de viaje, historial de visitas, valoraciones de destinos previos y datos contextuales (por ejemplo, la temporada del año o la duración del viaje) [5], [6]. Con esta información, el sistema puede ofrecer recomendaciones adaptadas que optimicen la experiencia del viajero, sugiriendo lugares de interés, actividades adecuadas a sus gustos y optimizando rutas de viaje [5], [15].

Dentro de las recomendaciones generadas, suelen incluirse lugares gastronómicos [12]. Los turistas son atraídos por las experiencias culinarias del lugar que están visitando, y la gastronomía a menudo se considera una parte clave de la experiencia de viaje [12], [14]. Por esta razón, también existen sistemas especializados en la recomendación de restaurantes, que ayudan a los turistas a descubrir y disfrutar de la oferta culinaria local [14], [16].

Sistema de recomendación de restaurantes

Un sistema de recomendación de restaurantes es una aplicación diseñada para sugerir opciones de restaurantes a los usuarios basándose en sus preferencias culinarias, historial de visitas, valoraciones y otros datos relevantes [7], [12], [17]. Estos sistemas ayudan a los usuarios a descubrir nuevos restaurantes que se ajusten a sus gustos y necesidades, facilitando la toma de decisiones en un contexto con una gran variedad de opciones disponibles [10], [17].

Para ofrecer recomendaciones precisas y personalizadas, los sistemas de recomendación de restaurantes emplean diversas técnicas [9], [10]. Una de las más comunes es el filtrado colaborativo, que sugiere restaurantes basándose en las preferencias de usuarios con gustos similares [9]. En el sistema de recomendación de restaurantes desarrollado por Mahadi, et al. [18], utiliza un algoritmo de filtrado colaborativo con el método K-means para predecir el grupo al que pertenece el usuario. Similarmente, el sistema de recomendación de restaurantes [9] adopta un enfoque de filtrado colaborativo basado en usuarios, generando recomendaciones a partir de las calificaciones de usuarios con preferencias similares.

La aplicación web Eat-Smart utiliza el modelo ALS (Alternating Least Squares) de Spark MLlib, basado en filtrado colaborativo, debido a su buen equilibrio entre precisión y rendimiento [19]. Otro ejemplo es el estudio presentado por Lavanya, et al. [20], que considera diferentes metodologías de filtrado colaborativo, como Slope One, K-Means y clasificación multiclase SVM.

Además de las técnicas de filtrado colaborativo, la factorización de matrices es otra técnica clave en estos sistemas, aplicada para descomponer grandes matrices de calificaciones en componentes más manejables. En el sistema de recomendación de restaurantes de [7], se utilizan tres técnicas diferentes de factorización de matrices: la factorización de matrices no negativas (NMF), la descomposición de valores singulares (SVD) y la descomposición de valores singulares optimizada (SVD++). Además, este sistema incluye un modelo de filtrado basado en contenido que mide la similitud entre diferentes características de los restaurantes, permitiendo recomendaciones personalizadas basadas en las preferencias del usuario. El estudio [16] emplea métodos de lógica difusa, distancia de similitud coseno, selección y optimización para mejorar las recomendaciones para usuarios individuales o grupos.

Un enfoque interesante en los sistemas de recomendación de restaurantes es el análisis de sentimiento, que se utiliza para extraer las preferencias alimenticias de los comentarios de los usuarios. Por ejemplo, en [10] sugiere restaurantes en función de estas preferencias, capturando aspectos cualitativos de la experiencia del usuario que no se reflejan únicamente en las calificaciones numéricas.

El uso de datos es crucial para estos sistemas. La mayoría de los trabajos aprovechan valoraciones de usuarios y reseñas recogidas de varias bases de datos como Zomato [9], [16], Yelp [19], [20], Foursquare [7] y en [8] sitios web como Foodmandu y Yellowpages. Sin embargo, los datos pueden llegar a desactualizarse y ofrecer recomendaciones no adecuadas.

El componente propone un sistema de recomendación de restaurantes que utiliza un enfoque híbrido que combina filtrado colaborativo y filtrado basado en contenido para ofrecer sugerencias precisas y personalizadas. Además, incorpora análisis de sentimiento realizado a los comentarios de los visitantes para identificar los mejores restaurantes junto con obtener los mejores restaurantes evaluados por Google. Las recomendaciones se ajustan a las preferencias del turista, asegurando que los restaurantes sugeridos estén abiertos y dentro de un radio adecuado según la ubicación actual. Se va a utilizar la API Places de Google para obtener información detallada sobre los restaurantes, incluidas rutas, ubicación y comentarios. Las puntuaciones de los restaurantes se basan en las calificaciones de Google, actualizadas regularmente, garantizando información precisa y actualizada.

Metodología aplicada

Scrum

Scrum es un marco de trabajo ágil diseñado para gestionar y completar proyectos complejos [21]. Promueve la colaboración, la transparencia y la adaptabilidad mediante iteraciones cortas y bien definidas llamadas *Sprints* [22]. Aunque es muy popular en el desarrollo de software, sus principios son aplicables a diversos sectores.

El marco de trabajo Scrum, presente en la Figura 1, se compone de tres elementos clave: el equipo Scrum, los eventos y los artefactos [21], [22]. Estos elementos se explicarán en detalle en los siguientes apartados.

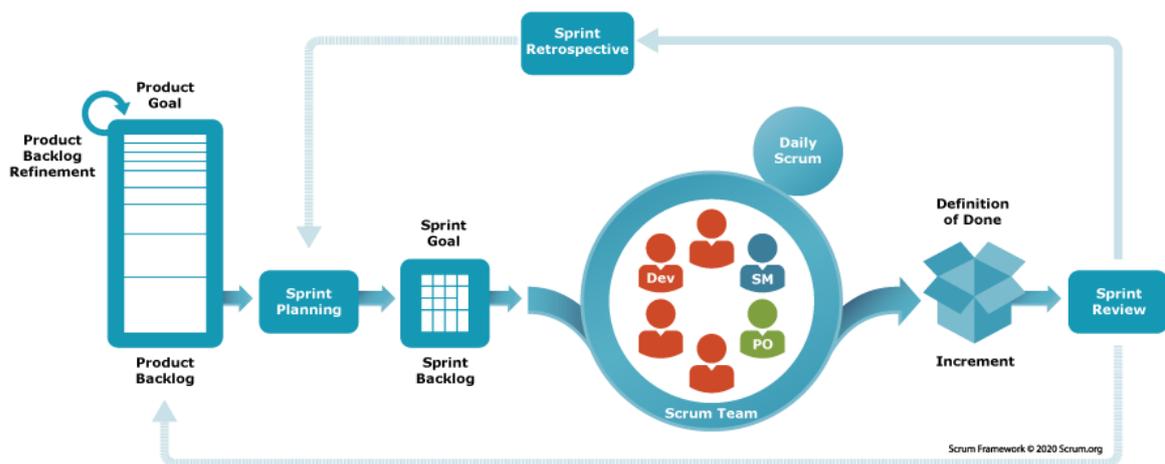


Figura 1. Marco de trabajo Scrum [22]

Equipo Scrum

Scrum se basa en un pequeño grupo de personas con las habilidades necesarias para generar valor en cada *sprint*. Los participantes tienen roles claramente definidos, que se describen a continuación [21]:

- *Scrum Master*: Es el responsable de ayudar al equipo a comprender y adoptar los valores, principios y prácticas de Scrum. Además, se asegura de que no existan impedimentos durante el desarrollo del producto.
- *Product Owner*: Es responsable de decidir las características y funcionalidades a desarrollar, enfocándose en maximizar el valor del producto. Además, mantiene una comunicación clara con todos los miembros del equipo sobre el progreso y las prioridades del desarrollo.

- *Equipo de desarrollo*: Son los responsables de autogestionar el desarrollo del producto, poseyendo las habilidades necesarias para diseñar, desarrollar y evaluar un producto valioso para el negocio.

Artefactos de Scrum

Scrum utiliza varios artefactos que ayudan a visualizar el trabajo y el progreso del equipo [22]:

1. *Product Backlog*: Es una lista priorizada de todo lo que se necesita en el producto. Mantiene todos los requisitos, características y correcciones necesarias, y es gestionada por el *Product Owner*.
2. *Sprint Backlog*: Contiene los elementos seleccionados del *Product Backlog* para el *sprint* actual, así como un plan detallado de cómo se realizará el trabajo. Este artefacto es gestionado por el equipo de desarrollo.
3. *Increment*: Es la suma de todos los elementos del *Product Backlog* completados durante el *Sprint*, junto con los de todos los *Sprints* anteriores. Forma una versión utilizable del producto que cumple con la definición de "hecho" establecida por el equipo.

Eventos de Scrum

Los eventos de Scrum son oportunidades para inspeccionar y adaptar los artefactos de Scrum. Los cinco eventos de Scrum son [22]:

1. *Sprint*: Es el ciclo de trabajo en Scrum, que puede durar entre una y cuatro semanas. Durante cada *sprint*, el equipo desarrolla un incremento del producto que sea potencialmente entregable y utilizable.
2. *Sprint Planning*: Esta reunión se lleva a cabo al inicio de cada *sprint*. En ella, el equipo decide qué elementos del *Product Backlog* se trabajarán durante el *sprint* y cómo se realizará el trabajo para cumplir con los objetivos del *sprint*.
3. *Daily Scrum*: Es una reunión diaria de 15 minutos donde los miembros del equipo sincronizan sus actividades y ajustan su plan de trabajo para las próximas 24 horas, identificando obstáculos y oportunidades de mejora.
4. *Sprint Review*: Al final del *sprint*, el equipo se reúne para presentar y revisar el trabajo completado. Los *stakeholders* y miembros del equipo discuten el progreso y ajustan el *Product Backlog* según sea necesario.

5. *Sprint Retrospective*: Después del *Sprint Review*, el equipo reflexiona sobre el *sprint* completado, identificando lo que funcionó bien y qué áreas necesitan mejorar. Esto permite la adaptación continua del proceso de trabajo.

La elección de Scrum para gestionar el desarrollo del producto se debe a su estructura de eventos y artefactos, que facilita la entrega de productos de alta calidad mediante una colaboración efectiva, transparencia y mejora continua. Esto permite al equipo responder ágilmente a los cambios y desafíos del proyecto.

Herramientas, frameworks y librerías de desarrollo

El sistema de recomendación de restaurantes a construir emplea diversas herramientas organizadas en las siguientes categorías:

Frontend

1. TypeScript

TypeScript es un superset de JavaScript que añade tipado estático y otras características avanzadas de programación a JavaScript. Desarrollado por Microsoft, TypeScript mejora la productividad de los desarrolladores al detectar errores de tipo en tiempo de compilación y permitir un desarrollo más estructurado y mantenible [23]. En el componente, TypeScript se usará tanto en el frontend como en el backend para asegurar la consistencia y robustez del código.

2. Angular

Angular es un framework de desarrollo de aplicaciones web de código abierto mantenido por Google. Facilita el desarrollo y las pruebas de aplicaciones de una sola página mediante una arquitectura basada en componentes y el uso de TypeScript. Proporciona herramientas para la vinculación de datos bidireccional y la inyección de dependencias [24]. En el componente, Angular se utilizará para desarrollar interfaces gráficas de usuario, asegurando una experiencia interactiva y dinámica.

3. Angular Material

Angular Material es una colección de componentes de interfaz de usuario para Angular. Provee una variedad de componentes preconstruidos basados en Material Design de Google [25]. Angular Material se empleará para crear componentes de interfaz de usuario con diseño moderno y coherente.

Backend

1. Node.js

Node.js es un entorno de ejecución de JavaScript que permite a los desarrolladores utilizar JavaScript del lado del servidor. Es usado para construir aplicaciones de red rápidas y escalables [24]. En el componente, Node.js se utilizará para manejar el servidor y la lógica del negocio.

2. Express

Express es un marco minimalista y flexible para Node.js que proporciona un conjunto robusto de características para el desarrollo de aplicaciones web y móviles [26]. Es utilizado para crear APIs RESTful debido a su simplicidad, eficiencia y extensibilidad. Express permite la creación de aplicaciones en una estructura modular, facilitando el mantenimiento y la escalabilidad [24]. En el componente, Express se usará para desarrollar la API que permitirá la comunicación entre el frontend y el backend.

3. Postman

Postman es una herramienta de colaboración y prueba de APIs. Permite diseñar, probar, depurar y documentar APIs de manera sencilla. Ofrece una interfaz gráfica intuitiva para enviar solicitudes HTTP y analizar respuestas, así como para automatizar pruebas y monitorizar el rendimiento de las APIs [27]. Postman se utilizará para probar la API desarrollada, asegurando su correcto funcionamiento y rendimiento.

Servicios Externos

1. Firebase

Firebase es una plataforma integral que facilita el desarrollo, compilación y publicación de aplicaciones modernas [28]. Ofrece diversos servicios, entre los que destacan:

- *Cloud Firestore*: Una base de datos NoSQL que simplifica el almacenamiento y la sincronización de datos para aplicaciones web y móviles [29].
- *Firebase Authentication*: Un servicio de autenticación segura diseñado para mejorar la experiencia de integración y acceso de los usuarios finales, permitiendo la autenticación a través de correo electrónico o redes sociales como Facebook y Gmail [30].

El componente empleara servicios de Firebase que facilitara el desarrollo del proyecto. Cloud Firestore se usará para almacenar datos del turista como sus preferencias, y Firebase Authentication facilitará el registro e inicio de sesión.

2. *Places API (new)*

La Places API, desarrollada por Google, proporciona un conjunto de datos sobre establecimientos basados en ciertos parámetros establecidos. Las solicitudes usadas en el componente son [31]:

- *Nearby Search (new)*: Permite buscar lugares basándose en ciertas variables dentro de un área específica.
- *Place Details (new)*: devuelve información detallada sobre un lugar específico.
- *Place Photos (new)*: devuelve fotos de los lugares que pueden ser usados en la aplicación.

3. *Routes API*

La Routes API facilita encontrar la mejor ruta entre un origen y un destino, proporcionando información detallada sobre rutas, tráfico y medios de transporte. Permite seleccionar el tipo de transporte, y calcular la distancia y el tiempo de viaje [32]. En el componente, Routes API se utilizará para calcular la ruta hacia un restaurante seleccionado, así como el tiempo y la distancia a recorrer.

4. *IBM Watson: Natural Language Understanding*

IBM Watson Natural Language Understanding es un servicio de inteligencia artificial que analiza y extrae información significativa de textos en lenguaje natural. Utiliza técnicas avanzadas de procesamiento del lenguaje natural (NLP) para comprender el contexto, identificar entidades, analizar sentimiento y mucho más [33]. En el componente, se utilizará para realizar un análisis de sentimientos sobre los comentarios de los visitantes a los restaurantes.

Entorno de pruebas

1. *Cucumber*

Cucumber es una herramienta de pruebas de comportamiento (BDD) que utiliza Gherkin para definir casos de prueba. Permite la creación de pruebas automatizadas que verifican el comportamiento de la aplicación según las especificaciones del cliente, mejorando la

colaboración y la claridad en el desarrollo de software [34]. En el componente, Cucumber se usará para definir y automatizar las pruebas de comportamiento del backend.

2. *Gherkin*

Gherkin es un lenguaje de dominio específico utilizado para escribir pruebas de comportamiento en formato legible para humanos [35]. En el componente, Gherkin se usará para escribir casos de prueba en lenguaje natural, facilitando la comunicación entre desarrolladores y partes interesadas no técnicas.

3. *Jest*

Jest es un framework de pruebas en JavaScript desarrollado por Facebook. Está diseñado para trabajar con aplicaciones basadas Node.js, proporcionando un entorno de pruebas rápido y seguro. Jest incluye características como pruebas paralelas, creación de snapshots y cobertura de código, lo que facilita la creación y mantenimiento de pruebas [36]. En el componente, Jest se usará para implementar y ejecutar pruebas unitarias, asegurando la calidad del código.

4. *Jest-Cucumber*

Jest-Cucumber es una biblioteca que integra Jest y Cucumber, permitiendo escribir escenarios en Gherkin y ejecutarlas con el framework de pruebas Jest. Esta integración facilita la adopción de BDD en proyectos que ya utilizan Jest para las pruebas unitarias, proporcionando una manera estructurada de definir y ejecutar pruebas de comportamiento [37]. En el componente, Jest-Cucumber se usará para integrar los escenarios junto con las pruebas realizadas en Jest.

5. *Karma*

Karma es una herramienta de prueba para JavaScript que permite ejecutar tests en múltiples navegadores de manera simultánea [38]. En el componente, Karma se usará para ejecutar pruebas automatizadas en el frontend, asegurando la compatibilidad del código en diferentes entornos.

6. *Jasmine*

Jasmine es un framework de pruebas de comportamiento para JavaScript que se centra en la simplicidad y la facilidad de uso. Proporciona una sintaxis limpia y legible para definir pruebas y expectativas, siendo una herramienta popular para realizar pruebas unitarias y de integración en aplicaciones JavaScript [39]. En el componente, Jasmine se usará para escribir pruebas unitarias, verificando la funcionalidad del código.

7. Karma-Jasmine

Karma- Jasmine es una combinación de los frameworks Jasmine y Karma, utilizada para ejecutar pruebas unitarias en el front-end [40]. En el componente, Karma-Jasmine se usará para obtener retroalimentación inmediata sobre la calidad del código y detectar errores de compatibilidad en los navegadores.

Despliegue

1. Netlify

Netlify es una plataforma de hospedaje y automatización para aplicaciones web modernas, que facilita el despliegue continuo y la administración de sitios web estáticos y aplicaciones [41]. Netlify se utilizará para desplegar y hospedar el frontend de la aplicación, asegurando una entrega rápida y eficiente.

2. Railway

Railway es una plataforma de infraestructura como servicio (IaaS) que simplifica el despliegue y administración de aplicaciones. Proporciona un entorno de desarrollo y producción sin complicaciones, con características como despliegue automático, escalabilidad y monitoreo integrado [42]. Railway se utilizará para gestionar y desplegar la infraestructura backend del componente, asegurando que sea escalable y fácil de mantener.

Gestión de proyectos

1. Webstorm

WebStorm es un entorno de desarrollo integrado (IDE) avanzado para JavaScript y tecnologías relacionadas, desarrollado por JetBrains. Ofrece características robustas como autocompletado inteligente, navegación de código, depuración y pruebas integradas, lo que facilita el desarrollo eficiente y efectivo de aplicaciones web modernas [43]. Webstorm será usado para facilitar el desarrollo tanto del backend como del frontend, manteniendo el código coherente y eficiente.

2. Notion

Notion es una herramienta de productividad y colaboración que combina notas, tareas, bases de datos y calendarios en una sola plataforma. Es utilizada para la gestión de proyectos y la documentación, permitiendo organizar y compartir información de manera

eficiente y estructurada [44]. Notion se usará para organizar y compartir la documentación necesaria de manera estructurada

3. GitHub

GitHub es una plataforma de hospedaje de código que utiliza Git para el control de versiones. Permite a los desarrolladores colaborar en proyectos de software, realizar seguimiento de cambios en el código, y gestionar proyectos a través de repositorios [45]. GitHub se utilizará como repositorio de versionamiento para gestionar el desarrollo de la aplicación.

4. Figma

Figma es una herramienta de diseño de interfaces y prototipos basada en la web. Facilita la colaboración en tiempo real entre diseñadores y desarrolladores, permitiendo la creación, revisión y prueba de interfaces de usuario de manera conjunta [46]. En el componente, Figma se usará para crear y revisar los mockups de la aplicación, asegurando un diseño de interfaz de usuario coherente.

2 METODOLOGÍA

Esta sección describe el desarrollo del componente a través de cuatro Sprints utilizando el marco de trabajo Scrum. A excepción del Sprint 0, los cuatro Sprints se estructuran en:

1. Objetivo del Sprint: Detalla el objetivo que tendrá el sprint
2. Planeación: Indica el conjunto de historias de usuario seleccionadas del Product Backlog y se crea el Sprint Backlog.
3. Implementación: Describe el desarrollo de las funcionalidades acordadas.
4. Revisión: Verificación del trabajo realizado con ayuda del gráfico Burndown y verificación de criterios de aceptación.
5. Retrospectiva: Se identifican áreas de mejora para futuros Sprints

2.1 Sprint 0

El Sprint 0 fue crucial para la fase inicial del proyecto. Durante este periodo de dos semanas, se llevó a cabo la selección de herramientas y la subsiguiente estructuración tanto del frontend como del backend. Además de estas tareas esenciales, se realizaron diversas actividades fundamentales que serán detalladas en las siguientes subsecciones.

1. Lógica del Proyecto: Se define el algoritmo que será el encargado de realizar la recomendación de restaurantes.
2. Arquitectura del Proyecto: Se diseñó la arquitectura del proyecto, asegurando que todos los componentes y su interacción estuvieran claramente definidos.
3. Definición de Roles de Scrum: Se asignaron los roles dentro del equipo Scrum, asegurando que cada miembro comprendiera sus responsabilidades y funciones específicas.
4. Creación de Historias Épicas y Product Backlog: Se desarrollaron las historias épicas y se creó el Product Backlog, recopilando todas las funcionalidades y características esenciales que se requerirán a lo largo del desarrollo del proyecto.
5. Definición del Release Planning: Se determinó el número de Sprints necesarios y su duración, estableciendo una estructura temporal para la ejecución del proyecto.

Lógica del proyecto

El algoritmo híbrido comienza obteniendo la ubicación y el radio de búsqueda para encontrar los restaurantes abiertos mejor evaluados por Google. Luego, ordena la lista de restaurantes según la calificación más alta, basada en el promedio del análisis de sentimiento realizado a los comentarios. Si el turista tiene preferencias, los restaurantes se filtran de acuerdo con estas, priorizando las tres más importantes, y se devuelve la lista de restaurantes recomendados. Si el turista no tiene preferencias, se devuelve la lista de restaurantes tal como está. El diagrama de flujo se puede observar en la Figura 2.

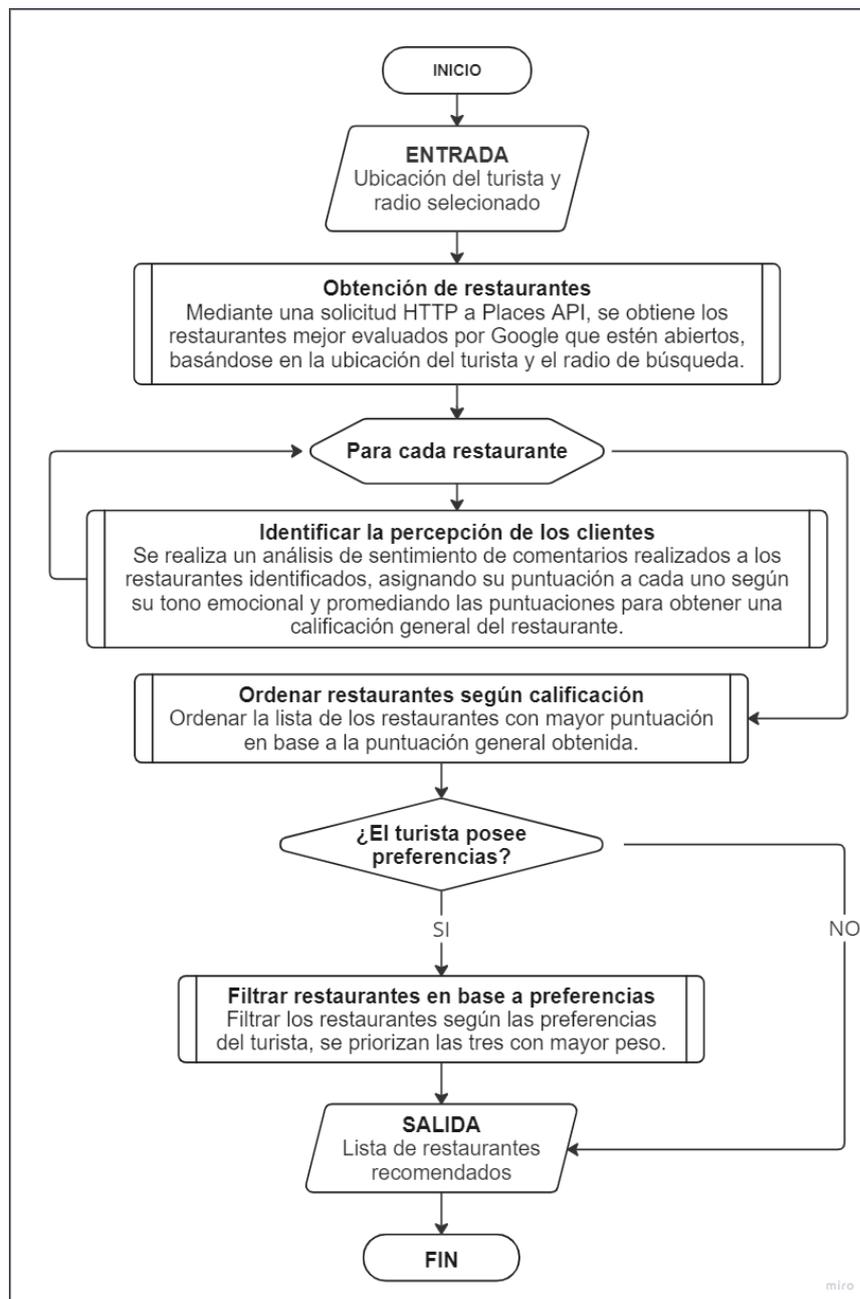


Figura 2: Diagrama de flujo del algoritmo híbrido

Arquitectura del proyecto

Para el proyecto, se ha seleccionado la arquitectura Modelo Vista Controlador (MVC). Esta arquitectura ofrece una clara división de responsabilidades, reutilización del código, mejor organización y compatibilidad con pruebas. A continuación, se detallan las funciones de cada capa en la arquitectura MVC:

1. Modelo: Gestiona los datos mediante Firestore, almacena y recupera información sobre las preferencias de los turistas, y obtiene información adicional de los restaurantes utilizando la API de Google. Además, implementa la lógica de recomendación de restaurantes.
2. Vista: Presenta una interfaz de usuario interactiva y dinámica que permite acceder a todas las funcionalidades del sistema.
3. Controlador: Actúa como intermediario entre el modelo y la vista. Contiene la aplicación frontend y la API REST alojadas en el servidor, facilitando la comunicación entre ambas y asegurando una interacción fluida y eficiente.

La Figura 3, muestra la interacción entre las distintas capas de la arquitectura MVC, ilustrando cómo las diversas tecnologías empleadas se comunican y colaboran entre sí.

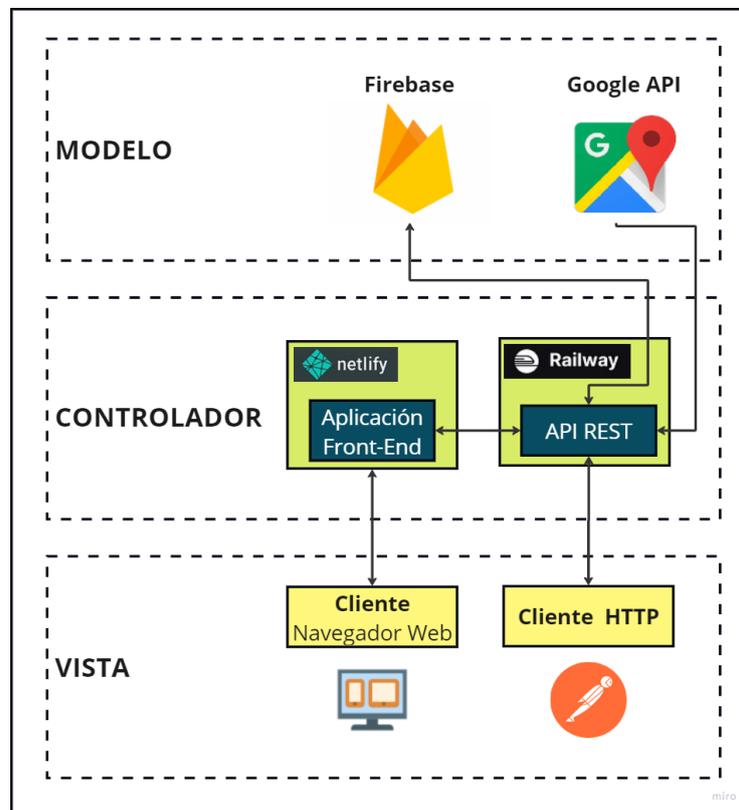


Figura 3. Arquitectura MVC aplicada al proyecto

Definición de roles

Para desarrollar el componente, se cuenta con dos integrantes encargados de cumplir los objetivos dentro del plazo especificado. La Tabla 1, detalla los integrantes y roles, destacando que uno de ellos asumirá dos funciones.

Tabla 1. Roles del equipo Scrum

Equipo Scrum	
Rol	Responsable
Product Owner	Maritzol Tenemaza
Scrum Master	Maritzol Tenemaza
Development team	Bryan Manobanda

Historias épicas

Durante el Sprint 0, se desarrollaron las historias épicas que se muestran en la Tabla 2. En el Anexo 1, se desglosa cada una de estas historias épicas detalladamente.

Tabla 2. Historias épicas

Código	Título	Prioridad
CAP01	Diseño de la interfaz y registro de preferencias por parte del turista	Alta
CAP02	Desarrollo de algoritmo para filtrar los restaurantes	Alta
CAP03	Implementar rutas para llegar al restaurante	Alta
CAP= COME AL PASO		

Product Backlog

La Tabla 3, presenta el *Product Backlog* estimado y priorizado, elaborado durante el Sprint 0. Las historias de usuario fueron estimadas utilizando puntos de historia basados en la escala de Fibonacci, lo que permitió asignarles un valor numérico y evaluar su complejidad. La priorización se llevó a cabo considerando la generación de valor para el producto y las necesidades del *Product Owner*.

Tabla 3. *Product Backlog* generado

Product Backlog				
Historia épica	Historia de usuario			
	ID	Título	Estimación (Puntos de historia)	Prioridad
CAP01	CAP01-01	Interfaz de preferencias del turista	13	Alta
	CAP01-02	Registro de información del turista en la aplicación	8	Alta
	CAP01-03	Almacenar preferencias del turista	6	Alta
CAP02	CAP02-01	Obtener restaurantes cerca de la ubicación geográfica del turista	8	Alta
	CAP02-02	Filtrar restaurantes en base a las preferencias	13	Alta
	CAP02-03	Obtener detalles de los restaurantes	13	Alta
CAP03	CAP03-01	Implementación de ruta hacia el restaurante	13	Alta
CAP01	CAP01-04	Iniciar sesión en la aplicación	8	Media
CAP02	CAP02-04	Analizar sentimiento en comentarios	8	Media
CAP= COME AL PASO				

Planificación del Release

La planificación del proyecto consta de cuatro sprints, cada uno con una duración de dos semanas. Por lo tanto, se estima que el desarrollo del sistema de recomendación tomará dos meses. La Tabla 4, detalla la planificación de cada sprint, incluyendo las historias de usuario con sus respectivas estimaciones y prioridades.

Tabla 4. Planificación de los Sprints

Release Planning				
Sprint / Objetivo	Historia de usuario			
	ID	Título	Estimación (Puntos de historia)	Prioridad
Sprint 1. Desarrollo de interfaz de usuario	CAP01-01	Interfaz de preferencias del turista	13	Alta
	CAP01-02	Registro de información del turista en la aplicación	8	Alta
	CAP01-03	Almacenar preferencias del turista	6	Alta
Sprint 2. Implementación de algoritmo híbrido	CAP02-01	Obtener restaurantes cerca de la ubicación geográfica del turista	8	Alta
	CAP02-02	Filtrar restaurantes en base a las preferencias	13	Alta
	CAP02-03	Obtener detalles de los restaurantes	13	Alta
Sprint 3. Conocer rutas al restaurante	CAP03-01	Implementación de ruta hacia el restaurante	13	Alta
	CAP01-04	Iniciar sesión en la aplicación	8	Media
Sprint 4: Análisis de sentimiento a comentarios	CAP02-04	Analizar sentimiento en comentarios	8	Media
CAP= COME AL PASO				

2.2 Sprint 1

Objetivo del Sprint

Analizar y desarrollar la interfaz de recepción de intereses; almacenar en una base de datos los intereses descritos por el turista y permitir al turista que se registre en la aplicación.

Planeación

Historias de usuario

Las historias de usuario para este Sprint están estimadas y listadas en la Tabla 5. Detalles sobre las historias de usuario del Sprint 1 se encuentran en el Anexo II.

Tabla 5. Historias de usuario del Sprint 1

ID	Título	Estimación (Puntos de historia)	Estado
CAP01-01	Interfaz de preferencias del turista	13	Por implementar
CAP01-02	Registro de información del turista en la aplicación	8	Por implementar
CAP01-03	Almacenar preferencias del turista	6	Por implementar
CAP= COME AL PASO			

Sprint Backlog

La Tabla 6, muestra el Sprint Backlog con sus tareas y estimación por historia de usuario.

Tabla 6. *Sprint Backlog del Sprint 1*

Sprint Backlog		
Historia de usuario	Tareas	Estimación
CAP01-01	Diseñar un mockup general del sistema de recomendación	2
	Implementar la interfaz de filtros de la aplicación	2
	Implementar las pruebas de la interfaz de filtros	2
	Implementar el diseño de la interfaz de restaurantes con el mapa de Google	1
	Implementar la lógica para obtener las preferencias de la interfaz y mandar al backend	3
	Restringir el movimiento del mapa a un perímetro de 5km y mostrar la ubicación del turista	1
	Investigar y configurar las pruebas en Karma y Jasmine, incluyendo el manejo de espías	2
CAP01-02	Implementar y probar la interfaz para registrar al turista con los campos de correo y nombre validados	2
	Configurar los CORS para permitir la comunicación entre el frontend y backend	1
	Asignar rutas y controladores que reciban los datos de nombre y correo	2
	Implementar la lógica para registrar al turista y guardar en la base de datos	2
	Configurar las respuestas obtenidas en el backend y mostrar en el frontend.	1
CAP01-03	Asignar rutas y controladores para enviar datos de preferencias desde el frontend	1
	Escribir la prueba de almacenamiento de preferencias	3
	Implementar la lógica para guardar los datos de preferencias en Firestore.	2
CAP= COME AL PASO		

Implementación

El Sprint 1 comenzó con el diseño general de la interfaz, esta interfaz fue desarrollada en Figma y se puede apreciar en el Anexo VI. Una vez finalizado el mockup del sistema, durante el desarrollo se destacan los siguientes aspectos:

Implementación de interfaz gráfica

La implementación de la interfaz se llevó a cabo siguiendo el mockup y dividiéndola en componentes para facilitar su desarrollo. Los principales componentes incluyen el mapa, la ventana de preferencias y la sección de restaurantes. Para cada uno de estos componentes, se utilizaron componentes preconstruidos de Angular Material, que fueron modificados para ajustarse al diseño del mockup y facilitar la implementación de las interfaces gráficas. La Figura 4, muestra la implementación general de una interfaz de usuario, donde se observa la integración del mapa y la posición actual del turista. Durante el sprint, la interfaz fue modificándose.

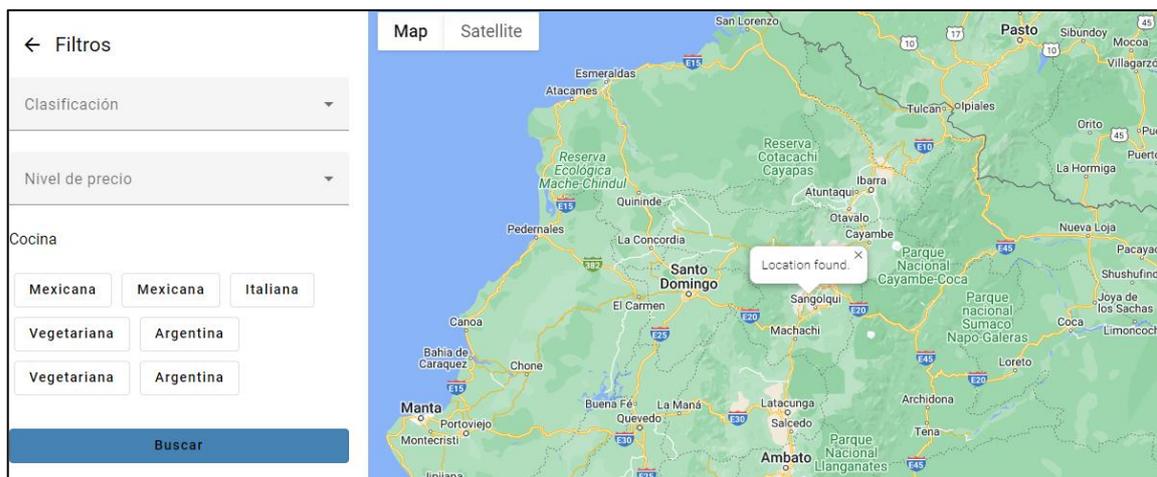


Figura 4. Primera versión de la interfaz de usuario

Determinar la posición del turista es esencial para definir el radio de búsqueda de restaurantes y simplificar el acceso al sistema. Además, permite trazar la ruta hacia un restaurante sin necesidad de especificar un punto de partida. La Figura 5, presenta el código para obtener las coordenadas geográficas de la posición del turista.

```
navigator.geolocation.getCurrentPosition(  
  successCallback: (position: GeolocationPosition) : void => {  
    this.pos = {  
      lat: position.coords.latitude,  
      lng: position.coords.longitude,  
    }  
  }  
);
```

Figura 5. Código para obtener la ubicación del turista

Comunicación entre el frontend y backend

Tras finalizar la implementación de las interfaces gráficas y la configuración inicial del servidor, se procedió a crear los endpoints necesarios para el sistema de recomendación. Sin embargo, la falta de configuración de CORS afectó la comunicación entre el frontend y el backend. CORS es un mecanismo que permite la integración de aplicaciones de diferentes dominios, requiere una lista de URLs autorizadas para las solicitudes. Para resolver este problema, se configuró dicha lista, como se detalla en la Figura 6.

```
const whitelist = ['http://localhost:4200']

const corsOptions = {
  origin: function (origin: string | undefined, callback: (error: Error
    if (whitelist.includes(origin as string) || (origin == null)) {
      callback( error: null, allow: true)
    } else {
      callback(new Error('Not allowed by CORS'), allow: false)
    }
  }
}

this.app.use(express.json())
this.app.use(cors(corsOptions))
```

Figura 6. Configuración de CORS

Una vez configurado CORS, se realizaron pruebas de conexión que arrojaron resultados positivos, como se muestra en la Figura 7, confirmando que el backend estaba recibiendo respuestas de manera adecuada.

```
SERVER ON PORT 3000
OPTIONS /api/places 204 2.126 ms - 0
POST /api/places 200 76.196 ms - 34
Respuesta recibida {rating: 'two', price_level: 'two', cuisines:
[ 'Mexicana' ] }
```

Figura 7. Respuesta del servidor

Registro de turistas

Para evitar interrumpir al turista con formularios extensos, se decidió permitir el registro mediante correo electrónico. El registro de turistas se llevó a cabo a través del servicio de autenticación de Firebase, el cual requería habilitar la autenticación por correo electrónico y activar la opción de vínculo de correo electrónico, como se muestra en la Figura 8. Esto simplificó el acceso del turista al eliminar la necesidad de establecer una contraseña.

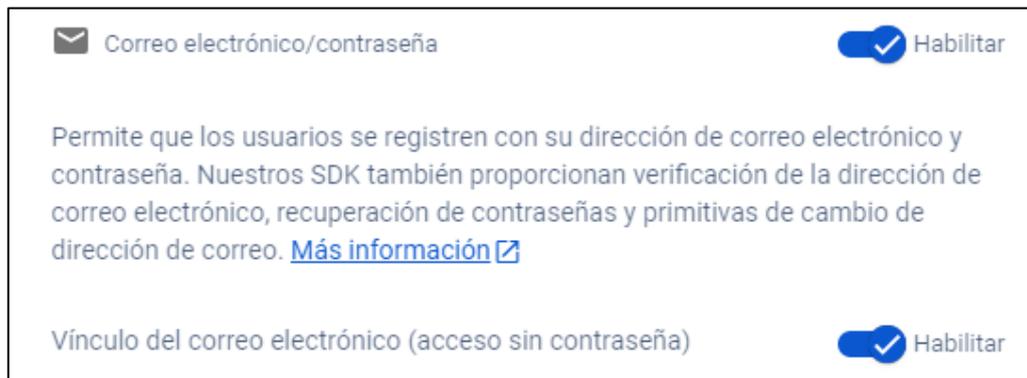


Figura 8. Habilitar acceso con correo en el servicio de autenticación de Firebase

Se desarrolló una interfaz gráfica para capturar el nombre y el correo del turista, con sus respectivas validaciones. Posteriormente, se realizó una prueba utilizando Karma y Jasmine para asegurar el correcto funcionamiento del componente, como se muestra en la Figura 9.

```
it( expectation: 'debería registrarse exitosamente y mostrar mensaje de confirmación', assertion: () : void
const turista: Turista = {
  uid: '123',
  nombre: 'Alexander Martinez',
  correo: 'alexander1324@hotmail.com',
}
const response : {status: number, turista: {uid... = {
  status: 201,
  turista: {uid: '123', nombre: turista.nombre, correo: turista.correo},
  message: '¡Registro exitoso!'
};

page.updateValue(page.correoInput, turista.correo);
page.updateValue(page.nombreInput, turista.nombre);

const serviceSpy : jasmine.Spy<jasmine.Func> = spyOn(securityService, method: "registrarUsuario").and.

page.submitButton.click();
fixture.detectChanges();
expect(registroComponent.mensajeError).toEqual( expected: '');
expect(registroComponent.mensajeConfirmacion).toEqual(response.message);
expect(registroComponent.turista.correo).toEqual(turista.correo);
expect(registroComponent.turista.nombre).toEqual(turista.nombre);
expect(serviceSpy).toHaveBeenCalled( params: {nombre: turista.nombre, correo: turista.correo});
```

Figura 9. Prueba unitaria desarrollada con Karma y Jasmine

Además, se implementó la lógica para almacenar la información del turista en la base de datos. La Figura 10, muestra el proceso de obtención de los datos del turista y la verificación de su registro. Si el turista ya está registrado, accede a la pantalla principal de recomendaciones; de lo contrario, sus datos se almacenan en Firestore.

```
export const postTurista = async (req: Request, res: Response) : Promise<void> => {
  try {
    const nuevoTurista: Turista = req.body as Turista;

    let existingUser;
    try {
      existingUser = await auth.getUserByEmail(nuevoTurista.correo);
    } catch (getUserError) {
      console.error('Error al obtener el usuario:', getUserError);
    }

    if (!existingUser) {

      const userRecord : UserRecord = await auth.createUser( properties: {
        email: nuevoTurista.correo
      });

      nuevoTurista.uid = userRecord.uid

      await firestore.collection( collectionPath: 'turist').doc(userRecord.uid).set(nuevoTurista);

      res.status( code: 201).json( body: {message: ';Registro exitoso!', nuevoTurista});
    } else {
      res.status( code: 201).json( body: {message: 'El correo se ha registrado previamente.'});
    }
  } catch (error) {
    res.status( code: 500).json( body: {message: 'Error al registrarse'});
  }
}
```

Figura 10. Lógica para almacenar datos de un turista nuevo

Almacenamiento de preferencias

Las preferencias del turista se almacenan en Firestore, asociadas al perfil de turista previamente creado. Estas preferencias son enviadas al backend, donde se procesa la respuesta del cuerpo y se guardan las preferencias como se muestra en la Figura 11.

```
▶ calidad_servicio: {4: 1}
▶ cocina: {bar: 1, seafood_restauran...}
  correo: "bryanstve2658@gmail.com"
  foto: "https://lh3.googleusercontent.com/a/ACg8ocIALSdq
        2Ek7GSz0LX3QNsUyI93BSi65f4UrJo=s96-c"
▶ nivel_precio: {PRICE_LEVEL_MODERATE: 1}
  nombre: "bryssta"
```

Figura 11. Almacenamiento de preferencias de un turista

Revisión

Pruebas de aceptación

La Tabla 7, presenta las pruebas de aceptación verificadas durante este sprint. En ella se puede verificar que todas las pruebas fueron superadas correctamente.

Tabla 7. Pruebas de aceptación del Sprint 1

Historia de usuario	Criterios de aceptación	Cumplido
CAP01-01	Búsqueda de Preferencia Dado que existe un formulario para elegir mis preferencias, Cuando selecciono todos los campos de preferencia y presiono el botón de búsqueda, Entonces se muestra restaurantes que cumplen con las preferencias seleccionadas.	Si
CAP01-02	Registro Exitoso con Nombre y Correo Válidos Dado que existe un formulario de registro, Cuando ingreso mi nombre y correo válidos, Y hago clic en el botón registrarme, Entonces mi información se almacena en la base de datos correctamente, Y se muestra un mensaje de confirmación de registro exitoso.	Si
	Mensaje de Datos Inválidos al Ingresar Información Incorrecta o campos vacíos Dado que existe un formulario de registro, Cuando ingreso un nombre o correo inválido, Y hago clic en el botón registrarme, Entonces aparece un mensaje indicando que los datos ingresados son inválidos.	Si
	Mensaje de Correo Existente al Registrar un Correo ya Utilizado Dado que existe un formulario de registro, Cuando ingreso mi nombre y un correo que ya está registrado en la base de datos, Y hago clic en el botón registrarme, Entonces se muestra un mensaje informando que el correo ingresado ya ha sido registrado previamente.	Si
CAP01-03	Almacenamiento de las Preferencias en la Base de Datos Dado que he seleccionado mis preferencias, Cuando hago clic en buscar, Entonces las preferencias elegidas se guardan correctamente en la base de datos.	Si
CAP= COME AL PASO		

Adaptación del Product Backlog

Una vez finalizado el Sprint 1 y presentados los resultados al Product Owner, se concluyó que el registro de turistas debía ser diferente para facilitar el acceso a la aplicación mediante redes sociales. Debido a este cambio, la historia de usuario CAP01-04 fue modificada para cumplir con este nuevo requisito. Para evitar retrasar otras funcionalidades, este ajuste se desarrollará en el tercer sprint. Como resultado, se actualizó el Product Backlog, como se muestra en la Tabla 8.

Tabla 8. Product Backlog Adaptado en el Sprint 1

Product Backlog				
Historia épica	Historia de usuario			
	ID	Título	Estimación (Puntos de historia)	Estado
CAP01	CAP01-01	Interfaz de preferencias del turista	13	Terminado
	CAP01-02	Registro de información del turista en la aplicación	8	Terminado
	CAP01-03	Almacenar preferencias del turista	6	Terminado
CAP02	CAP02-01	Obtener restaurantes cerca de la ubicación geográfica del turista	8	Por implementar
	CAP02-02	Filtrar restaurantes en base a las preferencias	13	Por implementar
	CAP02-03	Obtener detalles de los restaurantes	13	Por implementar
CAP03	CAP03-01	Implementación de ruta hacia el restaurante	13	Por implementar
CAP01	CAP01-04	Iniciar sesión en la aplicación	8	Por implementar
CAP02	CAP02-04	Analizar sentimiento en comentarios	8	Por implementar
CAP= COME AL PASO				

Gráfico Burndown

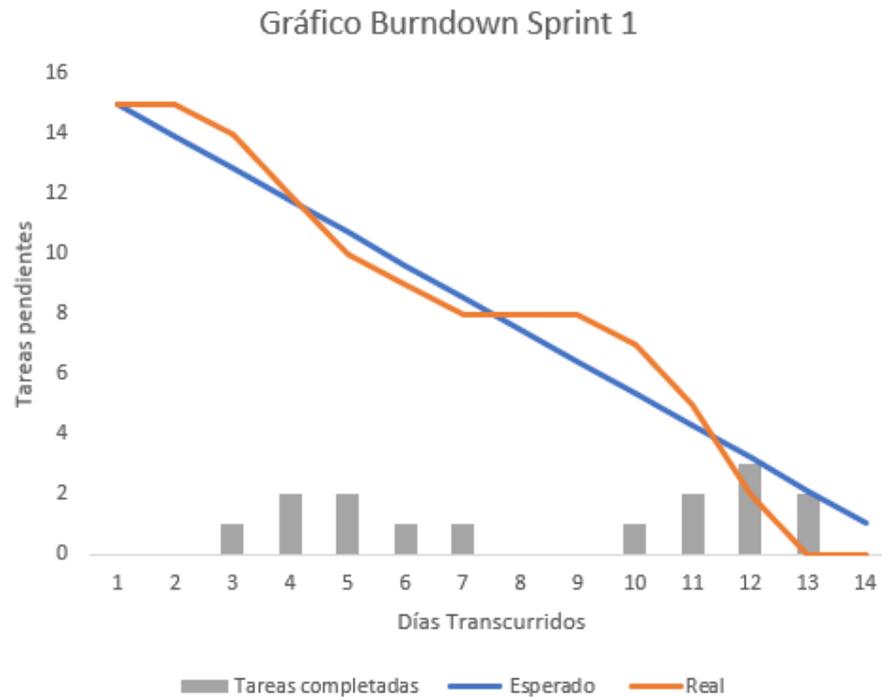


Figura 12. Gráfico Burndown del Sprint 1

La Figura 12, representa el gráfico Burndown del Sprint 1, durante la primera semana, el gráfico muestra un desarrollo normal, con un progreso constante y sin mayores desfases respecto al trabajo esperado. Sin embargo, la segunda semana presenta un gran desfase, indicando la falta de avance en las tareas planificadas debido a un cambio de enfoque y a desafíos técnicos, como: la integración entre el frontend y el backend y la implementación de pruebas. A pesar de este desfase, al final del Sprint 1 se logró completar todo el trabajo planificado, cumpliendo con las entregas previstas. La velocidad del equipo durante este sprint fue de 1.25 tareas por día hábil, lo que sugiere un ritmo de trabajo constante y establece una base de referencia para los siguientes sprints.

Retrospectiva

Tras realizar la retrospectiva, se obtuvieron las siguientes conclusiones que se considerarán para los próximos sprints:

1. Verificar los criterios de aceptación con el Product Owner antes de comenzar el desarrollo, lo que permitirá una mejor comprensión de los requisitos y reducirá el tiempo invertido en ajustes posteriores.
2. Establecer un repositorio de documentación y actualizarlo constantemente según el estado del proyecto.

2.3 Sprint 2

Objetivo del Sprint

Presentar los restaurantes cerca de la posición del turista, filtrar en base a sus preferencias y mostrar el detalle del restaurante seleccionado.

Planeación

Historias de usuario

Las historias de usuario para este Sprint están estimadas y listadas en la Tabla 9. En el Anexo III se encuentran detalles adicionales sobre las historias de usuario del Sprint 2.

Tabla 9. Historias de Usuario del Sprint 2

ID	Título	Estimación (Puntos de historia)	Estado
CAP02-01	Obtener restaurantes cerca de la ubicación geográfica del turista	8	Por implementar
CAP02-02	Filtrar restaurantes en base a las preferencias	13	Por implementar
CAP02-03	Obtener detalles de los restaurantes	13	Por implementar
CAP= COME AL PASO			

Sprint Backlog

La Tabla 10, muestra el Sprint Backlog con sus tareas y estimación por historia de usuario.

Tabla 10. Sprint Backlog del Sprint 2

Sprint Backlog		
Historia de usuario	Tareas	Estimación
CAP02-01	Implementar la prueba para obtener los restaurantes	2
	Crear ruta, controlador y servicios para realizar peticiones de obtener restaurantes	1
	Configurar la consulta a la API, para obtener restaurantes en base a ciertos parámetros	2
	Limpiar y agrupar la información obtenida de la API	1
	Presentar los datos obtenidos de los restaurantes en la interfaz de restaurantes y mostrar la ubicación en el mapa	2
CAP02-02	Implementar la prueba para filtrar los restaurantes	2
	Investigar sobre el algoritmo de recomendación	2
	Crear la ruta y el controlador para generar las peticiones para obtener recomendaciones	1
	Implementar el código para obtener restaurantes en base a recomendaciones	8
CAP02-03	Diseñar vista para mostrar la información	1
	Implementar la prueba para obtener información	2
	Crear ruta, controlador y servicios para realizar la petición para obtener información de un restaurante	1
	Implementar la lógica para obtener información del restaurante seleccionado	3
	Configurar la consulta para obtener información del restaurante desde la API	3
	Presentar la información de los restaurantes en la vista	3
CAP= COME AL PASO		

Implementación

Durante el Sprint 2, se adoptó un nuevo enfoque para la implementación del proyecto, destacándose los siguientes aspectos:

Places API

Para obtener los restaurantes, se realiza una petición HTTP POST a la API Nearby Search (New), especificando los parámetros de ubicación, radio, tipo de establecimiento y otros datos adicionales. La solicitud devuelve una lista de hasta 20 establecimientos mejor evaluados por Google, ubicados dentro de un radio de 5 km desde la ubicación del turista. Además, se solicitará que los establecimientos devueltos incluyan cualquiera de los tipos primarios de comida y bebida presentes en la API de Google [47].

La petición al ser exitosa devuelve un objeto JSON que contiene una lista de restaurantes, filtrados para incluir solo aquellos que estén abiertos. Para cada restaurante, si la información está disponible, se incluirán los siguientes datos:

- ID
- Nombre
- Nombre de la foto
- Dirección
- Abierto ahora
- Nivel de precios
- Calificación
- Cantidad de opiniones
- Sitio web
- Ubicación

La Figura 13, muestra el cuerpo de la solicitud realizada en Postman para obtener la lista de restaurantes. Los parámetros necesarios incluyen la ubicación en coordenadas de latitud y longitud, así como el radio de búsqueda. La Figura 14, presenta la respuesta de la solicitud.

```
{
  "location": {
    "lat": -0.2526478,
    "lng": -78.5220977
  },
  "radio": 5000.0
}
```

Figura 13. Cuerpo de solicitud de restaurantes

```

"restaurants": [
  {
    "id": "ChIJLbbLbvCZ1ZERlGt79n4YI8I",
    "displayName": "Buffet Rancho Elicio",
    "shortFormattedAddress": "La mana, y, Quito",
    "openNow": true,
    "priceLevel": "PRICE_LEVEL_MODERATE",
    "rating": 4.6,
    "userRatingCount": 1842,
    "websiteUri": "",
    "location": {
      "latitude": -0.2695722,
      "longitude": -78.5467687
    },
    "photos": [
      {
        "name": "places/ChIJLbbLbvCZ1ZERlGt79n4YI8I/photos/"
      }
    ]
  }
]

```

Figura 14. Respuesta de los restaurantes dentro de un área

Para obtener más detalles de un restaurante, se realizará una petición HTTP GET a la API Place Details (New), utilizando el ID del lugar y los datos específicos que se deseen obtener. La API proporciona una gran cantidad de información, de la cual se utilizarán datos como horario de funcionamiento, ambientación, servicios, opciones de pago y opciones de accesibilidad, presente en la Figura 15. En el caso de obtener una foto, se realiza una petición HTTP GET a Place Photos (New) indicando la URL obtenida previamente.

```

"weekdayDescriptions": [
  "Monday: Closed",
  "Tuesday: Closed",
  "Wednesday: Closed",
  "Thursday: 12:00-10:00\u202fPM",
  "Friday: 12:00-10:00\u202fPM",
  "Saturday: 11:00\u202fAM-10:00\u202fPM",
  "Sunday: 10:00\u202fAM-7:00\u202fPM"
],
"paymentOptions": {
  "acceptsCreditCards": true,
  "acceptsDebitCards": true,
  "acceptsCashOnly": false,
  "acceptsNfc": false
},
"services": {
  "takeout": false,

```

Figura 15. Respuesta de los detalles de un restaurante

Actualización y enrutamiento de interfaces gráficas

Las interfaces gráficas implementadas se actualizaron para mostrar los datos provenientes del backend. Durante el proceso de recepción y gestión de los datos, se utilizó interpolación, lo que posibilitó la reutilización de los componentes creados para presentar la información de todos los restaurantes obtenidos. Además, se incorporó una interfaz para mostrar los detalles al hacer clic en un restaurante.

Para mejorar la navegación del sistema, se estableció un enrutamiento adecuado y se optimizó la obtención de la ubicación del turista al iniciar la pantalla principal.

Pruebas del lado del servidor

La Figura 16, muestra los escenarios definidos previamente en Gherkin para observar el comportamiento del sistema desde la perspectiva del turista en el lado del servidor. Luego, se implementaron pruebas utilizando Cucumber y Jest para validar estos escenarios, ajustándolas hasta lograr que los escenarios funcionaran correctamente. La Figura 17, muestra el código de la implementación.

```
Característica: Obtener restaurantes cerca de una ubicación

Esquema del escenario: Obtener restaurantes cerca de una ubicación
  Dado que se tiene la <ubicacion> del turista
  Cuando realiza la búsqueda de restaurantes
  Entonces se muestran restaurantes abiertos
  Y dentro de un radio de 5km de su ubicación
  Ejemplos:
    | ubicacion          |
    | -0.2553811, -78.491781 |
    | 40.7128, -74.0060      |
```

Figura 16. Escenario escrito en Gherkin para obtener un restaurante

```

when( stepMatcher: 'realiza la búsqueda de restaurantes', stepDefinitionCallback: async () : Promise<void>
  response = await request(app).get( url: `/api/restaurants/${location}`);
});

then( stepMatcher: 'se muestran restaurantes abiertos', stepDefinitionCallback: () : void => {
  expect(response.status).toBe( expected: 200);
  response.body.restaurants.forEach((restaurante: any) : void => {
    expect(restaurante.openNow).toBe( expected: true);
  });
});

and( stepMatcher: /^dentro de un radio de (\d+)km de su ubicación$/, stepDefinitionCallback: (arg0) : void
  response.body.restaurants.forEach((restaurante: any) : void => {
    const distanciaRestaurante : number = calcularDistancia(location, restaurante.location);
    expect(distanciaRestaurante).toBeLessThanOrEqual( expected: 5);
  });
});

```

Figura 17. Implementación de escenario para obtener un restaurante

Revisión

Pruebas de aceptación

La Tabla 11, detalla los criterios cumplidos en el Sprint 2. Se destaca que el criterio para la CAP02-02 no fue alcanzado en este Sprint.

Tabla 11. Pruebas de aceptación del Sprint 2

Historia de usuario	Criterios de aceptación	Cumplido	Comentario
CAP02-01	Dado que se requiere la ubicación del turista. Cuando el turista comparte su ubicación y se realiza la búsqueda de restaurantes. Entonces se muestran restaurantes abiertos y dentro de un radio de 5km de su ubicación.	Si	
CAP02-02	Dado que el turista ha establecido sus preferencias. Cuando realiza la búsqueda de restaurantes, entonces se recomiendan restaurantes que cumplen con las preferencias establecidas y que se encuentren abiertos y dentro de un radio de 5Km de la ubicación del turista.	No	El tiempo de investigación del algoritmo fue más largo de lo previsto, lo que impidió su implementación.
CAP02-03	Dado que tengo una lista de restaurantes. Cuando elija un restaurante. Entonces se muestra la información del restaurante elegido	Si	
CAP= COME AL PASO			

Adaptación del Product Backlog

Tras analizar los resultados, se concluyó que el algoritmo de recomendación se implementará en el tercer sprint. Por lo tanto, se ajustó el Product Backlog de la Tabla 12 para reflejar estas necesidades.

Tabla 12. Product Backlog actualizado en el Sprint 2

Product Backlog				
Historia épica	Historia de usuario			
	ID	Título	Estimación (Puntos de historia)	Estado
CAP01	CAP01-01	Interfaz de preferencias del turista	13	Terminado
	CAP01-02	Registro de información del turista en la aplicación	8	Terminado
	CAP01-03	Almacenar preferencias del turista	6	Terminado
CAP02	CAP02-01	Obtener restaurantes cerca de la ubicación geográfica del turista	8	Terminado
	CAP02-02	Filtrar restaurantes en base a las preferencias	10	En desarrollo
	CAP02-03	Obtener detalles de los restaurantes	13	Terminado
CAP03	CAP03-01	Implementación de ruta hacia el restaurante	13	Por implementar
CAP01	CAP01-04	Iniciar sesión en la aplicación	8	Por implementar
CAP02	CAP02-04	Analizar sentimiento en comentarios	8	Por implementar
CAP= COME AL PASO				

Gráfico Burndown

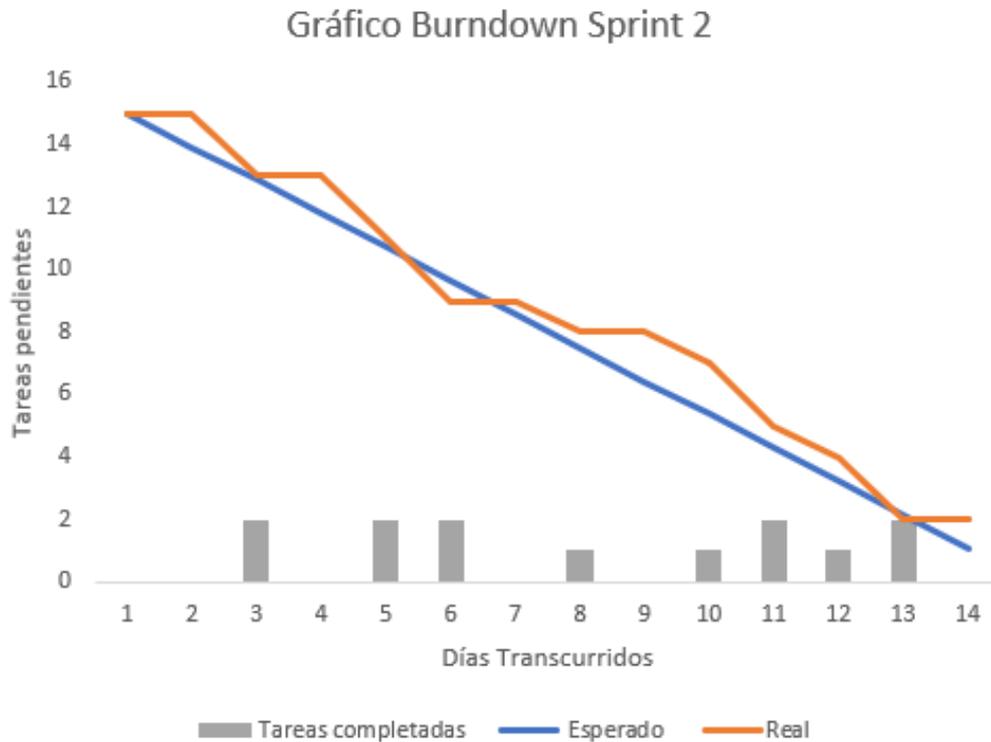


Figura 18. Gráfico Burndown del Sprint 2

En términos generales, este sprint fue más fluido en comparación con el anterior, ya que no se requirieron nuevas herramientas ni flujos de trabajo adicionales. La Figura 18, muestra que esta eficiencia permitió completar rápidamente dos historias de usuario. La velocidad del equipo en el Sprint 2 es de 1.4 tareas por día hábil, lo que permitió cumplir con las 14 tareas establecidas. Sin embargo, la tarea relacionada con la implementación del algoritmo de recomendación se trasladó al tercer sprint.

Retrospectiva

Durante la retrospectiva se obtuvieron las siguientes conclusiones:

Aspectos positivos:

1. Continuar mejorando la documentación del proyecto, asegurando que sea completa y esté actualizada en todo momento.
2. Mantener una comunicación continua con los miembros del equipo sobre el estado del proyecto y los posibles problemas identificados.

Aspectos negativos:

1. Mejorar la planificación y gestión del tiempo disponible, especialmente considerando factores externos como: tareas ajenas al proyecto o cortes de energía durante las tardes.
2. Reevaluar la priorización de tareas para enfocar más tiempo y recursos en aquellas historias que no presenten mayores problemas.

2.4 Sprint 3

Objetivo del Sprint

Filtrar los restaurantes en base a las preferencias, mostrar la ruta hacia el restaurante e iniciar sesión en la aplicación con Facebook o Gmail.

Planeación

Historias de usuario

La Tabla 13, contiene las historias de usuario usadas para el Sprint 3. Los detalles de las historias de usuario se encuentran en el Anexo IV.

Tabla 13. Historias de usuario para el Sprint 3

ID	Título	Estimación (Puntos de historia)	Estado
CAP02-02	Filtrar restaurantes en base a las preferencias	10	En desarrollo
CAP03-01	Implementación de ruta hacia el restaurante	13	Por implementar
CAP01-04	Iniciar sesión en la aplicación	8	Por implementar
CAP= COME AL PASO			

Sprint Backlog

El Sprint Backlog de la Tabla 14 muestra las tareas y estimación para las historias de usuario usadas en este sprint.

Tabla 14. *Sprint Backlog para el Sprint 3*

Sprint Backlog		
Historia de usuario	Tareas	Estimación
CAP02-02	Escribir la prueba para realizar las recomendaciones	2
	Implementar el código para obtener restaurantes en base a recomendaciones	5
	Actualizar la vista de restaurantes y mapa con las recomendaciones	3
CAP03-01	Implementar la prueba para conocer las rutas al restaurante	2
	Crear la ruta, controlador y servicio para realizar la petición de la ruta del restaurante	1
	Configurar la consulta a la API, para obtener la ruta del restaurante	8
	Presentar la ruta a pie y vehículo del restaurante seleccionado en el mapa	2
CAP01-04	Investigar el inicio de sesión con Facebook y Gmail	2
	Refactorizar el inicio de sesión con la nueva implementación	3
	Refactorizar el inicio de sesión y adaptar la interfaz creada	3
CAP= COME AL PASO		

Implementación

Durante el desarrollo del tercer Sprint se destaca lo siguiente:

Algoritmo de recomendación

El algoritmo de recomendación implementado utiliza un enfoque híbrido que combina filtrado colaborativo y filtrado basado en contenido para ofrecer las mejores sugerencias de restaurantes.

El filtrado colaborativo se aplica para obtener los restaurantes mejor valorados por Google. Esta plataforma promedia las evaluaciones de los usuarios y las presenta como una calificación global. No obstante, si se realiza una evaluación reciente, Google tarda dos semanas en actualizarla [48].

Para aplicar el filtrado basado en contenido, se recopilan las preferencias del turista y se contrastan con la información del restaurante. Las preferencias del turista se almacenan en una matriz de pesos, donde cada preferencia seleccionada suma un punto. De este modo, cuando el turista realiza búsquedas frecuentes, algunas preferencias adquieren mayor peso que otras, como se muestra en la Tabla 15.

Tabla 15. Matriz de pesos para la especialidad culinaria

Especialidad culinaria	Peso
Comida del Mar	10
Italiana	8
Mexicana	5
Bar	3
Comida rápida	2

El sistema de recomendación opera con dos tipos de turistas:

1. Turistas sin preferencias: En este caso, solo se aplica el filtrado colaborativo, mostrando los restaurantes mejor valorados y cercanos a la ubicación del turista.
2. Turistas con preferencias: Se aplica tanto el filtrado colaborativo como el filtrado basado en contenido. Las recomendaciones se basan en las preferencias almacenadas en la matriz de pesos, destacando las tres con mayor relevancia.

Rutas al restaurante

Para obtener la ruta al restaurante, se realiza una petición HTTP POST a la Routes API, especificando los parámetros descritos en la Tabla 16.

Tabla 16. Parámetros para obtener una ruta

Parámetro	Descripción
Origen	Ubicación del turista
Destino	ID restaurante seleccionado
Modo de viaje	Selección a pie o en vehículo
Unidad de medida	Metros

La Figura 19, muestra el cuerpo de la petición. Cuando la petición es exitosa, la API devuelve un objeto JSON que contiene la distancia en metros, la duración del viaje en segundos y la línea de la ruta desde el punto de origen hasta el destino. Estos resultados se muestran en la interfaz gráfica de la Figura 20.

```
.... "location": {  
.... "latLng": {  
.... |..... "latitude": -0.2526478,  
.... |..... "longitude": -78.5220977  
.... }  
.... },  
.... "placeId": "ChIJlbbLbvCZ1ZERlGt79n4YI8I",  
.... "travelMode": "WALK"
```

Figura 19. Cuerpo de petición para obtener una ruta

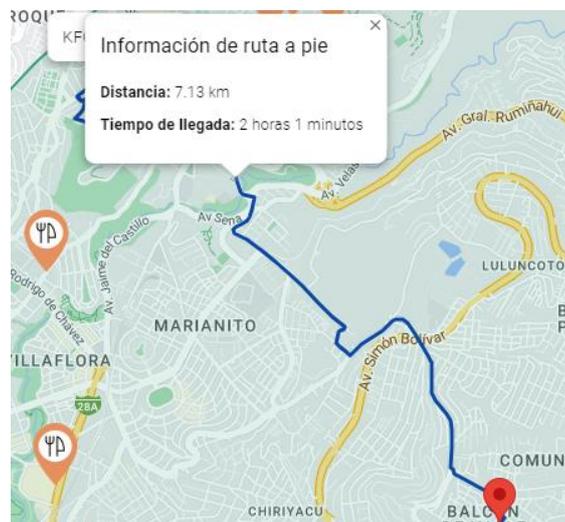


Figura 20. Interfaz gráfica de ruta desde la ubicación hasta el restaurante

Inicio de sesión

Para simplificar el proceso de inicio de sesión, se sugirió implementar la autenticación a través de redes sociales utilizando los servicios de autenticación de Firebase. Se habilitaron las opciones de inicio de sesión mediante Gmail y Facebook como se muestra en la Figura 21.

Proveedor	Estado
 Google	 Habilitado
 Facebook	 Habilitado

Figura 21. Proveedores para inicio de sesión

El sistema permite al turista escoger entre estas dos opciones. Si el turista ya ha iniciado sesión previamente, accede directamente a la pantalla de recomendaciones; de lo contrario, se registra y es redirigido a la pantalla de recomendaciones. Además, el sistema verifica el estado de inicio de sesión, redirigiendo al turista a la pantalla correspondiente, ya sea la de inicio de sesión o la de recomendaciones.

Revisión

Pruebas de aceptación

La Tabla 17, detalla que todos los criterios fueron cumplidos en el Sprint 3.

Tabla 17. Pruebas de aceptación para el Sprint 3

Historia de usuario	Criterios de aceptación	Cumplido
CAP02-02	Dado que el turista ha establecido sus preferencias. Cuando realiza la búsqueda de restaurantes, entonces se recomiendan restaurantes que cumplen con las preferencias establecidas y que se encuentren abiertos y dentro de un radio de 5Km de la ubicación del turista.	Si
CAP03-01	Dado que el turista ha seleccionado un restaurante, cuando solicita la ruta hacia el restaurante, entonces se muestra la ruta en el mapa para llegar desde su ubicación hasta el restaurante seleccionado	Si
CAP01-04	Dado que el turista ingresa su correo y contraseña, cuando se verifique que las credenciales son correctas, entonces podrá acceder exitosamente a la aplicación.	Si
CAP= COME AL PASO		

Adaptación del Product Backlog

Las historias de usuario propuestas para este sprint se completaron exitosamente. Para el siguiente sprint se sugirió reforzar el algoritmo de recomendación con análisis de sentimiento de los comentarios. La Tabla 18, muestra el Product Backlog actualizado.

Tabla 18. Product Backlog actualizado en el Sprint 3

Product Backlog				
Historia épica	Historia de usuario			
	ID	Título	Estimación (Puntos de historia)	Estado
CAP01	CAP01-01	Interfaz de preferencias del turista	13	Terminado
	CAP01-02	Registro de información del turista en la aplicación	8	Terminado
	CAP01-03	Almacenar preferencias del turista	6	Terminado
CAP02	CAP02-01	Obtener restaurantes cerca de la ubicación geográfica del turista	8	Terminado
	CAP02-02	Filtrar restaurantes en base a las preferencias	10	Terminado
	CAP02-03	Obtener detalles de los restaurantes	13	Terminado
CAP03	CAP03-01	Implementación de ruta hacia el restaurante	13	Terminado
CAP01	CAP01-04	Iniciar sesión en la aplicación	8	Terminado
CAP02	CAP02-04	Analizar sentimiento en comentarios	8	Por implementar
CAP= COME AL PASO				

Burdown Chart

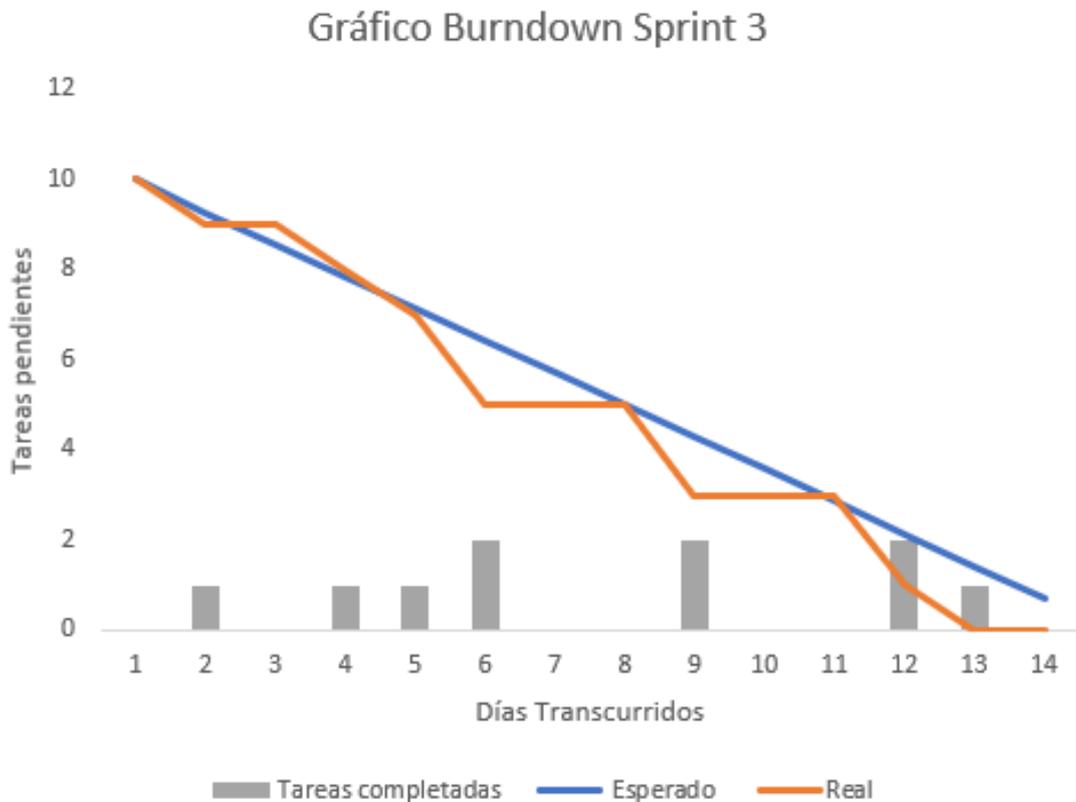


Figura 22. Gráfico Burndown en el Sprint 3

La Figura 22, muestra el gráfico Burndown del Sprint 3, donde se evidencia que todo el trabajo se completó con éxito, cumpliendo con el objetivo establecido para este sprint. La velocidad en este sprint, de 1.25 tareas por día hábil, muestra una disminución en comparación con el Sprint 2, atribuida tanto a la reducción en el número de tareas como a la disminución de los días hábiles en este sprint.

Retrospectiva

Durante este sprint, no se presentaron problemas significativos en el desarrollo. El equipo trabajó de manera fluida y eficiente, completando todas las historias de usuario propuestas y alcanzando los objetivos establecidos sin contratiempos. Por lo tanto, no hay sugerencias adicionales para futuros sprints.

2.5 Sprint 4

Objetivo del Sprint

Analizar sentimiento de los comentarios a los restaurantes y ordenar los restaurantes en base al promedio de los comentarios, desde los más positivos hasta los más negativos.

Planeación

Historias de usuario

Las historias de usuario para el Sprint 4 se encuentran en la Tabla 19. En el Anexo V se encuentran en detalle las historias de usuario del Sprint 4.

Tabla 19. Historias de usuario usadas en el Sprint 4

ID	Título	Estimación (Puntos de historia)	Estado
CAP02-04	Analizar sentimiento en comentarios	8	Media
CAP= COME AL PASO			

Sprint Backlog

La Tabla 20, muestra el Sprint Backlog con tareas y estimación por historia de usuario.

Tabla 20. Sprint Backlog del Sprint 4

Sprint Backlog		
Historia de usuario	Tareas	Estimación
CAP02-04	Investigar el análisis de sentimiento y obtener una credencial en IBM Watson	1
	Implementar un servicio para realizar el análisis de comentarios y agregar al sistema de recomendación	5
	Actualizar la recomendación usando los restaurantes mejor valorados y con los comentarios más positivos al negativo.	2
CAP= COME AL PASO		

Implementación

Análisis de sentimiento

Se recopiló información sobre el servicio de procesamiento de lenguaje natural de IBM, el cual ofrece acceso a su API para utilizar su analizador de texto. Esta herramienta asigna a los comentarios, indicando si son positivos como se muestra en la Figura 23 o negativos según la Figura 24. Para probar la API, se hizo uso de Postman, en el cuerpo de la petición, se indica el comentario y la funcionalidad de sentimiento.

```
1 { "text": "Wow! Where to start... this place is absolutely outstanding! The  
    ambiance is stunning, so many wonderful pictures you can take here. The  
    how delicious it was. It was all so fresh and authentic Italian food.",  
2 .. "features": {  
3   ... "sentiment": {  
4  
5  
pretty Raw Preview JSON v
```

```
2 "sentiment": {  
3   "document": {  
4     "score": 0.94528,  
5     "label": "positive"
```

Figura 23. Análisis de sentimiento para un comentario positivo (Solicitud en Postman)

```
1 { "text": "Interesting place downtown service was good but food was just ok.  
    Sandwiches needed something, and the water was not very available. I had  
    the Spanish torta (omelet with potatoes) traditional Spanish eggs.\n    The wife had the traditional calamari sandwich. they had no sauce no mayo  
    no nothing. But wanted to eat, and that is all we could find to eat.",  
2 .. "features": {  
3   ... "sentiment": {  
4  
5  
6  
7  
pretty Headers (19) Raw Preview JSON v
```

```
1 {  
2   "sentiment": {  
3     "document": {  
4       "score": -0.299739,  
5       "mixed": "1",  
6       "label": "negative"  
7     }  
}
```

Figura 24. Análisis de sentimiento para un comentario negativo (Solicitud en Postman)

La implementación del análisis de sentimiento se llevó a cabo al obtener la lista de restaurantes ordenados según las mejores evaluaciones de Google. Además, en la solicitud inicial para obtener los restaurantes, se incluyeron los comentarios correspondientes. El código de la implementación se encuentra en la Figura 25.

```
reviews: result.reviews?.map((review:any) : {review: any} =>
  ({review: review.originalText?.text}).filter(
    (comentario:any) : boolean => {
      const palabras = comentario.review?.split(' ');
      return palabras?.length >= 5;
    }) || [],
score: 0
```

Figura 25. Código para obtener los comentarios de los restaurantes

Una vez obtenida la lista ordenada de restaurantes y sus respectivos comentarios, se procedió a analizar cada uno de ellos utilizando la API de Watson. Se realiza una petición HTTP POST al servicio Natural Language Understanding, especificando parámetros como el comentario y la funcionalidad de sentimientos como se muestra en la Figura 26.

```
const parameters : {text: string, features: {sent... = {
  "text": comentario.review,
  "features": {
    "sentiment": {}
  }
};
```

Figura 26. Parámetros para realizar análisis de sentimiento

La solicitud devuelve un valor positivo o negativo para cada comentario, lo que permitió calcular un promedio de todas las opiniones asociadas a cada restaurante y asignarle un puntaje. Finalmente, se generó una lista con todos los restaurantes ordenados de mayor a menor puntuación como se muestra en la Figura 28. El código relacionado se puede consultar en la Figura 27.

```
restaurant.score = totalScore / restaurant.reviews.length;
}
return restaurants.sort( compareFn: (a : Restaurant , b : Restaurant ) => b.score - a.score);
```

Figura 27. Código de lista ordenada de restaurantes en base al análisis de sentimiento

```

"restaurants": [
  {
    "displayName": "La Vid Restaurante",
    "score": 0.9415124
  },
  {
    "displayName": "Cafe Mosaico",
    "score": 0.8767457999999999
  },
  {
    "displayName": "Hasta La Vuelta Señor",
    "score": 0.8338856
  },
  {
    "displayName": "Community Hostel",
    "score": 0.7607496
  },
  {
    "displayName": "Vista Hermosa (Itchimbia)",
    "score": 0.508972
  }
]

```

Figura 28. Lista ordenada de restaurantes según el puntaje de análisis de sentimiento

Revisión

Pruebas de aceptación

La Tabla 21, detalla que todos los criterios fueron cumplidos en el Sprint 4.

Tabla 21. Pruebas de aceptación del Sprint 4

Historia de usuario	Criterios de aceptación	Cumplido
CAP02-04	Dado que se obtiene los comentarios del restaurante. Cuando se realiza el análisis de sentimiento de cada comentario, entonces se obtiene una puntuación que permite ordenar los restaurantes con mejor puntuación.	Si
CAP= COME AL PASO		

Adaptación del Product Backlog

La Tabla 22, presenta el Product Backlog completo, que incluye todas las historias de usuario implementadas.

Tabla 22. Product Backlog actualizado en el Sprint 4

Product Backlog				
Historia épica	Historia de usuario			
	ID	Título	Estimación (Puntos de historia)	Estado
CAP01	CAP01-01	Interfaz de preferencias del turista	13	Terminado
	CAP01-02	Registro de información del turista en la aplicación	8	Terminado
	CAP01-03	Almacenar preferencias del turista	6	Terminado
CAP02	CAP02-01	Obtener restaurantes cerca de la ubicación geográfica del turista	8	Terminado
	CAP02-02	Filtrar restaurantes en base a las preferencias	10	Terminado
	CAP02-03	Obtener detalles de los restaurantes	13	Terminado
CAP03	CAP03-01	Implementación de ruta hacia el restaurante	13	Terminado
CAP01	CAP01-04	Iniciar sesión en la aplicación	8	Terminado
CAP02	CAP02-04	Analizar sentimiento en comentarios	8	Terminado
CAP= COME AL PASO				

Burdown Chart

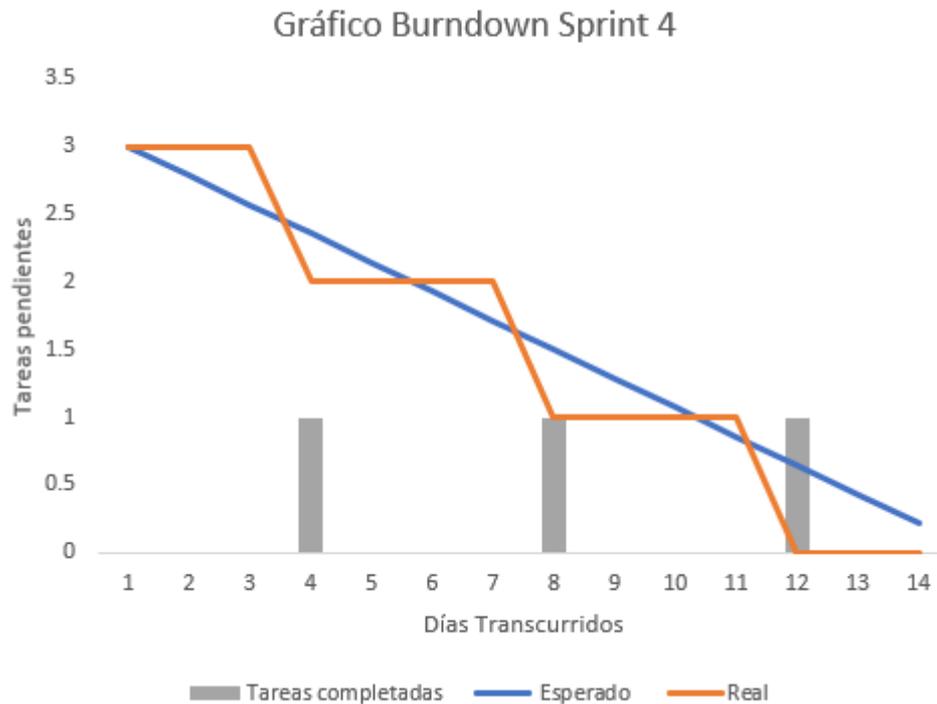


Figura 29. Gráfico Burndown del Sprint 4

Durante este sprint, se implementó una historia de usuario sin enfrentar complicaciones, logrando alcanzar con éxito los objetivos establecidos, como se muestra en la Figura 29. La velocidad del equipo disminuyó significativamente en comparación con los sprints anteriores, alcanzando solo 0.5 tareas por día hábil. Esta reducción se debe a que la historia de usuario incluyó únicamente 3 tareas.

Retrospectiva

Durante los sprints anteriores, se adquirió valiosos aprendizajes que han permitido incorporar nuevas herramientas al componente de manera fluida. Esta integración sin complicaciones ha resultado en fortalecer la funcionalidad en el componente. En esta retrospectiva, se puede concluir que el trabajo realizado brinda una ventaja considerable para proyectos futuros. Cada sprint ha proporcionado nuevas lecciones y conocimientos, que me capacitan para abordar desafíos con mayor confianza.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados

Esta sección detalla los resultados obtenidos durante el desarrollo del sistema de recomendación de restaurantes, el cual está desplegado en los servidores de Netlify para el frontend y Railway para el backend. Incluye el análisis de la encuesta de usabilidad realizada a 20 participantes.

Producto Final

El producto final es un sistema de recomendación de restaurantes que incluye una pantalla para registrarse o iniciar sesión utilizando redes sociales como Facebook o Gmail como se muestra en la Figura 30. El turista puede elegir cualquiera de estas opciones para facilitar su acceso al sistema, evitando cuestionarios extensos, la necesidad de recordar contraseñas y demoras en el inicio de sesión. Si un turista ingresa por primera vez al sistema, este lo registra en la base de datos. Si el turista ya ha utilizado el sistema anteriormente, este lee la información registrada y las preferencias asociadas al perfil del turista para ofrecer recomendaciones de restaurantes en el área donde se encuentre.



Figura 30. Pantalla de registro e inicio de sesión

Una vez que el turista ingresa al sistema, se le presenta una pantalla de recomendaciones que muestra un mapa con su ubicación actual, un círculo de búsqueda con un radio de 5 km y las localizaciones de los restaurantes recomendados como se muestra en la Figura 31. Los restaurantes recomendados son aquellos mejor evaluados por Google y luego clasificados mediante un análisis de sentimiento a los comentarios de los visitantes del restaurante. Las recomendaciones se dividen en dos secciones: "Para ti" y "Cerca de ti".

La sección "Para ti" incluye restaurantes que coinciden con las preferencias del turista, mientras que la sección "Cerca de ti" presenta opciones que están dentro del área de búsqueda pero que no coinciden con las preferencias del turista.

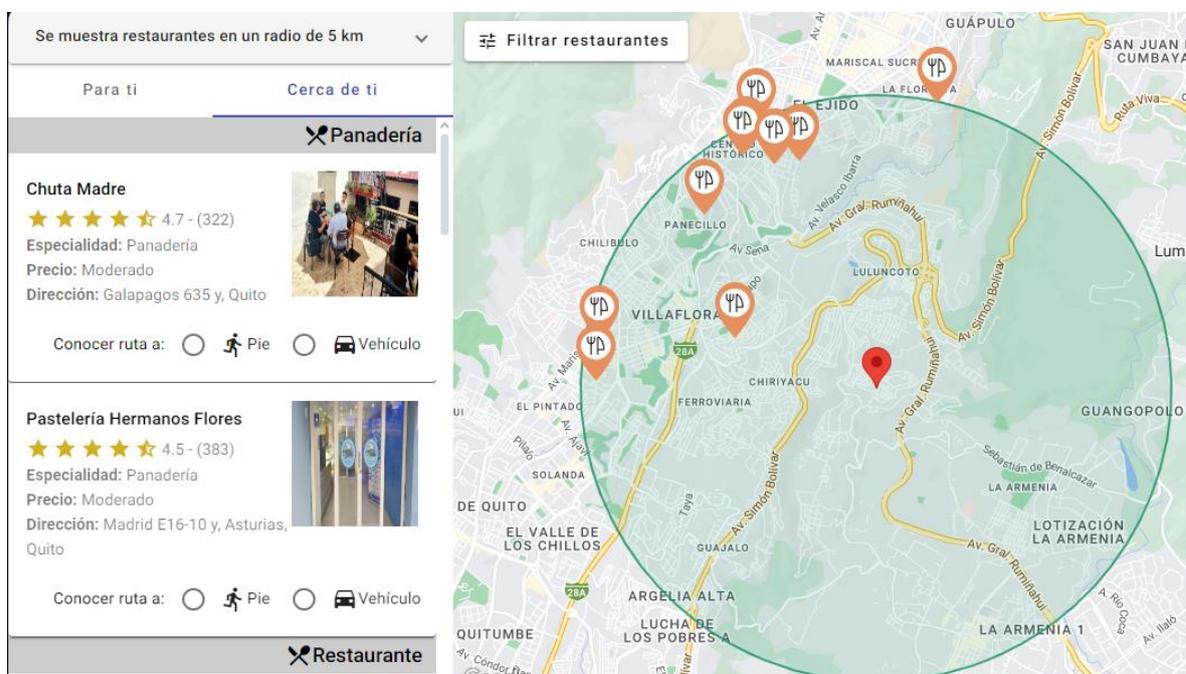


Figura 31. Pantalla de recomendaciones

La Figura 32, ilustra la sección "Cerca de ti" para un turista que está buscando restaurantes que se encuentran en el área de búsqueda. En el caso de un turista nuevo, la sección "Para ti" aún no muestra ningún restaurante, pero esto puede cambiar una vez que el turista empiece a filtrar las opciones de restaurantes. Cuando un turista ya ha indicado sus preferencias, el sistema adapta las secciones "Para ti", tal como se muestra en la Figura 33.

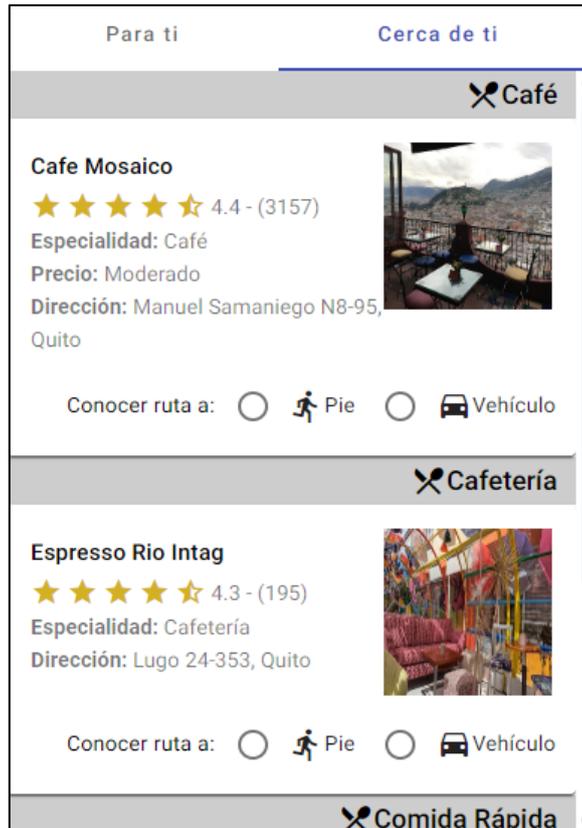


Figura 32. Pantalla de recomendaciones “Cerca de TI”

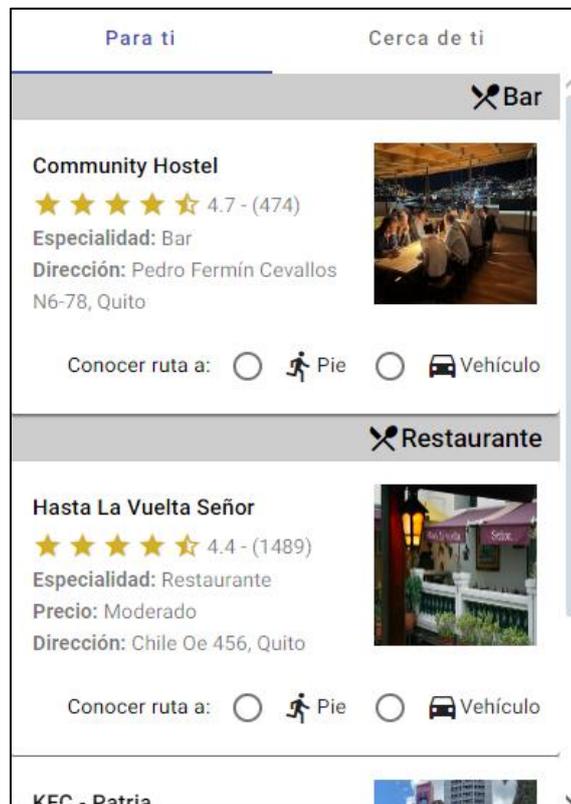


Figura 33. Pantalla “Para ti” cuando un turista tiene preferencias

El turista al seleccionar un restaurante puede acceder a información como descripción, servicios y el horario de funcionamiento como se muestra en la Figura 34.



Figura 34. Pantallas de información de un restaurante

El turista posee la opción de conocer la ruta hacia un restaurante de su elección. El sistema de recomendación le permite elegir entre la ruta a pie o en vehículo. Para ambas opciones, se dibuja la ruta en el mapa y se le indica la distancia y el tiempo estimado de llegada desde la posición hasta el restaurante seleccionado como se muestra en la Figura 35.

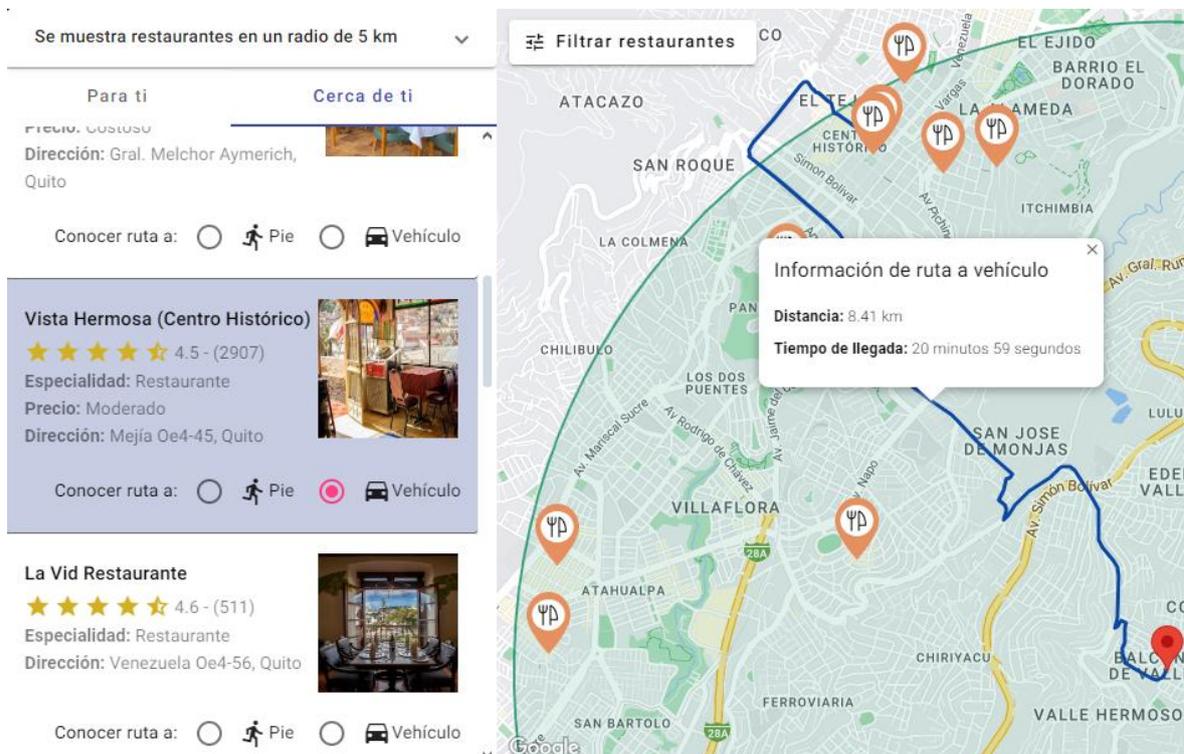


Figura 35. Ruta desde la posición del turista a un restaurante seleccionado

El turista tiene la posibilidad de filtrar los restaurantes mostrados según criterios de clasificación, nivel de precio o especialidad culinaria. Dependiendo de los restaurantes disponibles en el área, estos criterios se actualizarán automáticamente, como se muestra en la Figura 36. Los criterios seleccionados por el turista se almacenarán en la base de datos y se utilizarán para mostrar sus preferencias en futuras recomendaciones. Según los criterios elegidos, se presentarán los resultados correspondientes, tal como se ilustra en la Figura 37.



Figura 36. Diversas pantallas de filtros actualizados según el área de búsqueda

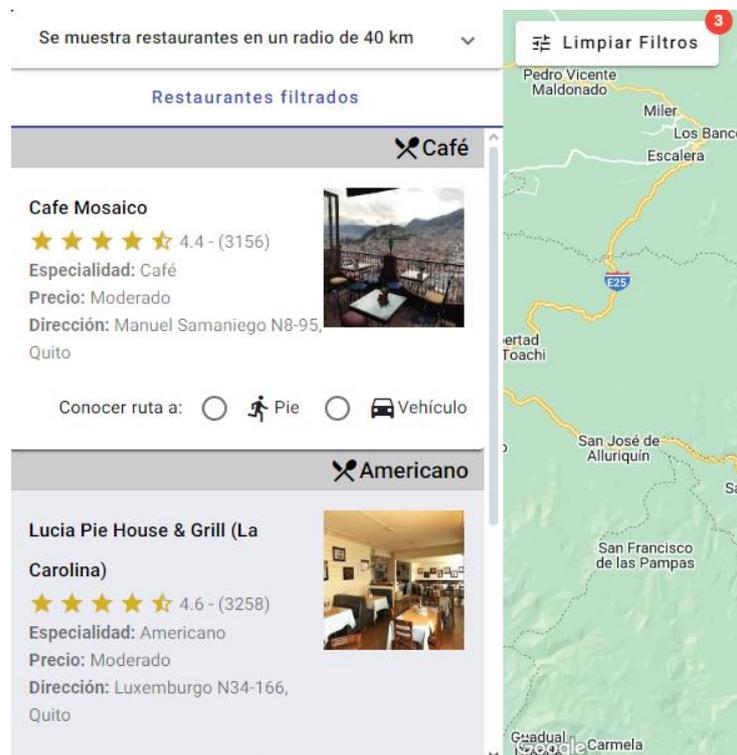


Figura 37. Resultados de restaurantes al filtrar opciones

El sistema de recomendación permite ampliar o disminuir el radio de búsqueda para obtener nuevas recomendaciones de restaurantes. Si el turista cambia el radio de búsqueda presente en la Figura 38, la lista de restaurantes recomendados se van a actualizar acorde al nuevo radio de búsqueda como se muestra en la Figura 39. Los nuevos restaurantes serán divididos en las secciones “Para ti” y “Cerca de ti” según las preferencias del turista.



Figura 38. Pantalla para modificar el radio de búsqueda de restaurantes

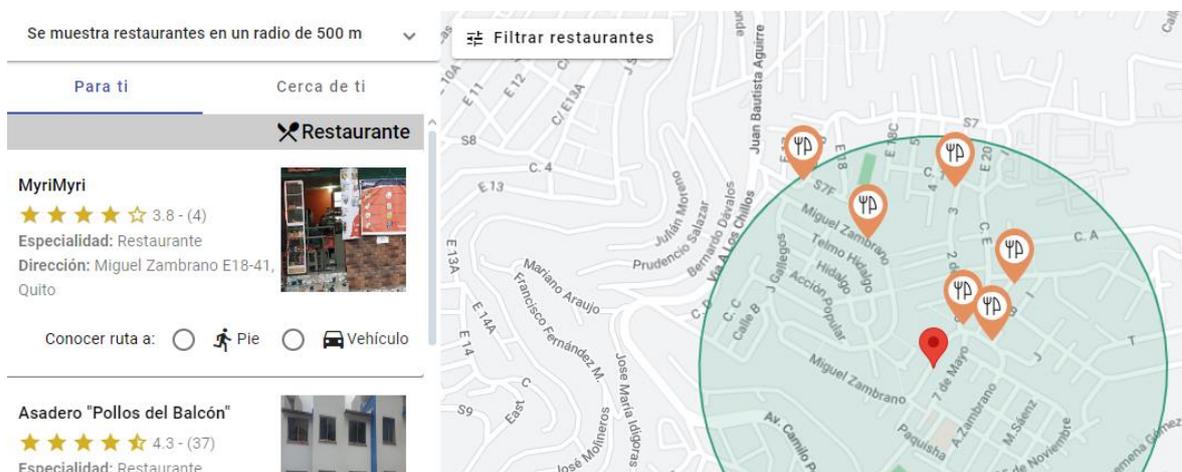


Figura 39. Actualización de recomendaciones al cambiar el radio de búsqueda

El sistema de recomendación de restaurantes desarrollado permite al turista obtener recomendaciones basadas en sus preferencias o descubrir nuevos restaurantes en el área donde se encuentra. A un turista nuevo le brinda la posibilidad de recibir recomendaciones iniciales y construir su perfil de preferencias a medida que usa el sistema. Las recomendaciones de restaurantes se refinan según el perfil del turista.

Encuesta de usabilidad

La prueba se diseñó para evaluar la usabilidad del sistema de recomendación de restaurantes. Antes de su realización, se estableció un protocolo de evaluación (Anexo VII) y un documento guía (Anexo VIII) que detallan el propósito del sistema, las tareas y las encuestas a realizar. Cada participante fue informado sobre la prueba y se les aseguró que recibirían el documento "Guía de Prueba" antes de la evaluación.

Participaron 20 personas residentes en la ciudad de Quito, con edades entre 21 y 26 años. La muestra estuvo compuesta por un 80% de hombres y un 20% de mujeres. En cuanto a la distribución geográfica, el 50% de los participantes vive en el norte de Quito, el 40% en el sur, y el 5% tanto en el centro como en los valles circundantes. Ninguno de los participantes había utilizado previamente el sistema de recomendación.

La usabilidad del sistema se medirá a través del cuestionario estándar SUS (System Usability Scale). Este cuestionario consta de 10 preguntas predefinidas, evaluadas mediante una escala de Likert de 5 puntos, donde 1 indica "totalmente en desacuerdo" y 5 indica "totalmente de acuerdo" [49]. La encuesta de usabilidad se encuentra en el Anexo IX, y las respuestas recopiladas se presentan en el Anexo X.

El resultado de la encuesta SUS fue calculado como se muestra a continuación [50]:

- I. Sea i el cálculo de las respuestas impares tal que:

$$i = \sum \text{Respuestas impares} - 5$$

Ecuación 1. Cálculo de preguntas impares

- II. Sea p el cálculo de las respuestas pares tal que:

$$p = 25 - \sum \text{Respuestas pares}$$

Ecuación 2. Cálculo de preguntas pares

- III. Sea R el resultado SUS tal que:

$$R = (i + p) * 2.5$$

Ecuación 3. Cálculo de resultado SUS

El puntaje SUS individual para los 20 participantes se muestra en la Tabla 23.

Tabla 23: Puntaje SUS individual de cada participante

Participantes	Puntaje SUS	Participantes	Puntaje SUS
Kevin Martinez	82.5	Jeimmy Eche	90
Byron Iñacasha	90	Danny Cabrera	62.5
Jorge Cordero	55	David Casnanzuela	75
Issac de la Cadena Bonilla	75	Glender Miranda	70
Gabriela Salazar	97.5	Sebastián Jiménez	85
Vickiann Jiménez	87.5	Fausto Román	80
Andrés Cantuña	97.5	Jimmy Quingatuña	77.5
Anthony Chamba	85	Joffre Córdor	97.5
Kevin Toasa	97.5	Rommel Masabanda	75
Alex Escobar	60	Jessica Morales	80

El puntaje SUS del sistema de recomendación de restaurantes se obtiene mediante el promedio de los puntajes individuales, resultando en un valor de 81. Este resultado sugiere que el sistema tiene una usabilidad aceptable, como se ilustra en la Figura 40.

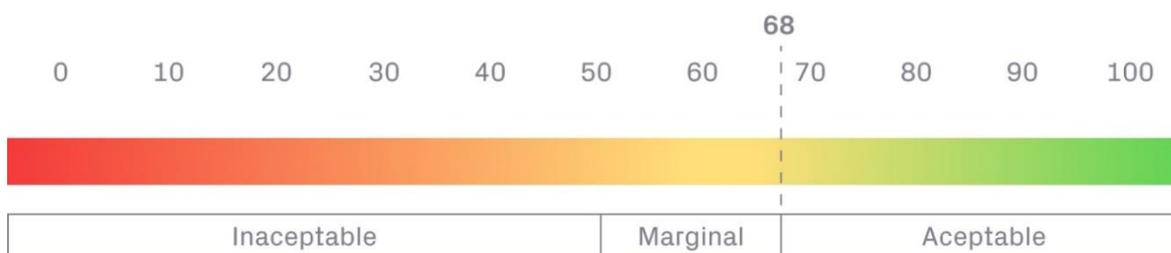


Figura 40: Representación de resultados SUS (tomada de [50])

Es importante destacar que los participantes eran jóvenes con un alto nivel de familiaridad con la tecnología. Además, no se puede asegurar que todos hayan usado el sistema con un interés genuino, lo cual podría afectar la precisión de los puntajes obtenidos. Por lo tanto, estos resultados no deben considerarse como indicativos de perfección, sino como un indicio de que se está avanzando en la dirección correcta para mejorar el sistema de recomendación de restaurantes.

3.2 Conclusiones

En este apartado se presentan las conclusiones obtenidas respecto a los objetivos específicos de este componente:

- **Respecto al objetivo específico 1:** Se logró crear una interfaz intuitiva que permite a los turistas recibir recomendaciones de restaurantes basadas en sus preferencias, las cuales se almacenan en una base de datos para futuros accesos. Las pruebas de usabilidad confirmaron que el sistema es fácil de usar y cumple su propósito de recomendación.
- **Respecto al objetivo específico 2:** Se implementaron algoritmos que extraen información de restaurantes desde la API de Google, asegurando que los establecimientos recomendados estén abiertos, dentro del área especificada y con las mejores evaluaciones de Google. Además, se realizó un análisis de sentimiento sobre los comentarios de los visitantes para clasificar los restaurantes. Las recomendaciones se presentan en la interfaz de usuario en dos secciones: "Para Ti", si el turista ha indicado preferencias, y "Cerca de Ti", mostrando los restaurantes disponibles en el área circundante.
- **Respecto al objetivo específico 3:** El sistema de recomendación de restaurantes proporciona rutas detalladas para llegar a los restaurantes seleccionados, ya sea a pie o en vehículo, según la preferencia del turista. Un mapa muestra la ruta desde la ubicación actual del turista hasta el restaurante, incluyendo información sobre el tiempo estimado y la distancia a recorrer.
- **Respecto al objetivo específico 4:** Se ofrece la posibilidad de ajustar el radio de búsqueda, permitiendo a los turistas ampliar o reducir el área para encontrar restaurantes que mejor se adapten a su ubicación y preferencias actuales.

3.3 Recomendaciones

Para trabajo futuro se recomienda:

- Ofrecer a los turistas la opción de cambiar su ubicación para explorar otras zonas y recibir recomendaciones de restaurantes en esas áreas, brindando una experiencia más amplia y personalizada.
- Ofrecer a los turistas la posibilidad de utilizar el sistema de recomendaciones y obtener sugerencias de restaurantes sin tener que registrarse o iniciar sesión.
- Agregar a la información del restaurante los comentarios realizados por otros visitantes y permitir a los turistas dejar sus propias opiniones después de visitar un restaurante.
- Añadir información sobre las rutas para llegar a los restaurantes en bicicleta y mediante transporte público, ofreciendo alternativas de transporte más ecológicas y accesibles para los turistas.
- Desarrollar filtros de búsqueda más avanzados y personalizables para que los turistas puedan encontrar restaurantes que se adapten mejor a sus preferencias específicas.
- Mejorar la accesibilidad del sistema de recomendaciones mediante la implementación de comandos de voz y permitir la visualización 3D de Google Maps, de modo que los turistas puedan ver los alrededores del restaurante y orientarse mejor.
- Mejorar la retroalimentación proporcionada por el sistema al realizar búsquedas o cambiar entre secciones, asegurando que el turista sepa qué está procesando el sistema. Además, agregar indicaciones visuales de carga, implementar un proceso de onboarding para nuevos turistas y optimizar los tiempos de carga de la aplicación para una experiencia más fluida y eficiente.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook," in *Recommender Systems Handbook*, Boston, MA: Springer US, 2011, pp. 1–35. doi: 10.1007/978-0-387-85820-3_1.
- [2] C. C. Aggarwal, *Recommender Systems*. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-29659-3.
- [3] H. Ko, S. Lee, Y. Park, and A. Choi, "A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields," *Electronics (Basel)*, vol. 11, no. 1, p. 141, Jan. 2022, doi: 10.3390/electronics11010141.
- [4] Z. Y. Jia, W. Gao, and Y. J. Shi, "An Agent Framework of Tourism Recommender System," *MATEC Web of Conferences*, vol. 44, p. 01005, Mar. 2016, doi: 10.1051/matecconf/20164401005.
- [5] J. P. Lucas, N. Luz, M. N. Moreno, R. Anacleto, A. A. Figueiredo, and C. Martins, "A hybrid recommendation approach for a tourism system," *Expert Syst Appl*, vol. 40, no. 9, pp. 3532–3550, 2013.
- [6] A. Moreno, L. Sebastiá, and P. Vansteenwegen, "Recommender Systems in Tourism," *IEEE Intelligent Informatics Bulletin*, vol. 16, no. 1, p. 1, 2015.
- [7] R. Alabduljabbar, "Matrix Factorization Collaborative-Based Recommender System for Riyadh Restaurants: Leveraging Machine Learning to Enhance Consumer Choice," *Applied Sciences*, vol. 13, no. 17, p. 9574, Aug. 2023, doi: 10.3390/app13179574.
- [8] P. Neupane, "Restaurant Recommendation System Based On Collaborative Filtering," Doctoral dissertation, Tribhuvan University, 2016.
- [9] A. A. Fakhri, Z. K. A. Baizal, and E. B. Setiawan, "Restaurant Recommender System Using User-Based Collaborative Filtering Approach: A Case Study at Bandung Raya Region," *J Phys Conf Ser*, vol. 1192, p. 012023, Mar. 2019, doi: 10.1088/1742-6596/1192/1/012023.
- [10] E. Asani, H. Vahdat-Nejad, and J. Sadri, "Restaurant recommender system based on sentiment analysis," *Machine Learning with Applications*, vol. 6, p. 100114, Dec. 2021, doi: 10.1016/j.mlwa.2021.100114.
- [11] T. Fahrudin and N. Wisna, "The Exploration of Restaurant Recommender System," *Journal of Computer Science*, vol. 18, no. 8, pp. 784–791, Aug. 2022, doi: 10.3844/jcssp.2022.784.791.
- [12] Y. chao Shi and U.-K. Lee, "The Impact of Restaurant Recommendation Information and Recommendation Agent in the Tourism Website on the Satisfaction, Continuous Usage, and Destination Visit Intention," *Sage Open*, vol. 11, no. 4, p. 215824402110469, Oct. 2021, doi: 10.1177/21582440211046947.
- [13] R. Alabduljabbar, M. Alshareef, and N. Alshareef, "Time-Aware Recommender Systems: A Comprehensive Survey and Quantitative Assessment of Literature," *IEEE Access*, vol. 11, pp. 45586–45604, 2023, doi: 10.1109/ACCESS.2023.3274117.

- [14] S. Bertan, "Impact of restaurants in the development of gastronomic tourism," *Int J Gastron Food Sci*, vol. 21, p. 100232, Oct. 2020, doi: 10.1016/j.ijgfs.2020.100232.
- [15] H. Amzad and K. Vijayalakshmi, "Tourism Recommendation System: A Systematic Review," *International Journal of Engineering Research & Technology (IJERT)*, vol. 10, Sep. 2021.
- [16] M. Hartanto and D. N. Utama, "Intelligent decision support model for recommending restaurant," *Cogent Eng*, vol. 7, no. 1, p. 1763888, Jan. 2020, doi: 10.1080/23311916.2020.1763888.
- [17] A. A. Munaji and A. W. R. Emanuel, "Restaurant Recommendation System Based on User Ratings with Collaborative Filtering," *IOP Conf Ser Mater Sci Eng*, vol. 1077, no. 1, p. 012026, Feb. 2021, doi: 10.1088/1757-899X/1077/1/012026.
- [18] M. I. Mahadi, N. Zainuddin, N. B. Shah, N. Naziron, and S. F. Rum, "E-halal restaurant recommender system using collaborative filtering algorithm," *Journal of Advanced Research in Computing and Applications*, vol. 12, no. 1, pp. 22–34, 2018.
- [19] F. F. Cao, "Eat-Smart: A Restaurant Recommendation Web Application," California State University, Northridge, 2018.
- [20] B. M. Lavanya, K. K. Kumar, H. S. Kayanath, and D. P. Bai, "A Machine Learning Model for Recommending Restaurants based on User Ratings," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 9, no. 1, pp. 732–736, May 2020, doi: 10.35940/ijrte.A1189.059120.
- [21] K. S. Rubin, *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley, 2012.
- [22] Scrum.org, "What is Scrum?" [Online]. Available: <https://www.scrum.org/learning-series/what-is-scrum/what-is-scrum>
- [23] C. Nance, *Typescript essentials*. Packt Publishing Ltd, 2014.
- [24] B. Dayley, B. Dayley, and C. Dayley, *Node.js, MongoDB and Angular Web Development: The definitive guide to using the MEAN stack to build web applications*. Addison-Wesley Professional, 2017.
- [25] G. LLC, "Angular Material - UI Component Library." 2024. [Online]. Available: <https://material.angular.io>
- [26] StrongLoop IBM and other expressjs.com contributors, "Express - Node.js Web Application Framework." 2024. [Online]. Available: <https://expressjs.com>
- [27] P. Inc., "Postman API Platform." 2024. [Online]. Available: <https://www.postman.com>
- [28] G. LLC, "Firebase." 2024. [Online]. Available: <https://firebase.google.com/?hl=es-419>
- [29] G. LLC, "Firestore." 2024. [Online]. Available: <https://firebase.google.com/products/firestore?hl=es-419>

- [30] G. LLC, "Firebase Authentication." 2024. [Online]. Available: <https://firebase.google.com/products/auth?hl=es-419>
- [31] G. Developers, "Places API." 2024. [Online]. Available: https://developers.google.com/maps/documentation/places/web-service?hl=es_419
- [32] G. Developers, "Routes API Overview." 2024. [Online]. Available: <https://developers.google.com/maps/documentation/routes/overview?hl=es-419#v2-enhancements>
- [33] I. B. M. Corporation, "IBM Watson Natural Language Understanding." 2024. [Online]. Available: <https://www.ibm.com/products/natural-language-understanding>
- [34] Cucumber, "Cucumber: BDD Testing & Collaboration." 2024. [Online]. Available: <https://cucumber.io>
- [35] Cucumber, "Gherkin Reference." 2024. [Online]. Available: <https://cucumber.io/docs/gherkin/reference/>
- [36] Jest, "Jest: Delightful JavaScript Testing." 2024. [Online]. Available: <https://jestjs.io>
- [37] NPM, "jest-cucumber." 2024. [Online]. Available: <https://www.npmjs.com/package/jest-cucumber>
- [38] Karma, "Karma - Spectacular Test Runner for JavaScript." 2024. [Online]. Available: <https://karma-runner.github.io/latest/index.html>
- [39] Jasmine, "Jasmine Documentation." 2024. [Online]. Available: <https://jasmine.github.io>
- [40] NPM, "karma-jasmine." 2024. [Online]. Available: <https://www.npmjs.com/package/karma-jasmine>
- [41] Netlify, "Netlify: Build, deploy, and scale modern web projects." 2024. [Online]. Available: <https://www.netlify.com>
- [42] Railway, "Railway: Infrastructure made simple." 2024. [Online]. Available: <https://railway.app>
- [43] JetBrains, "WebStorm: The Smartest JavaScript IDE." 2024. [Online]. Available: <https://www.jetbrains.com/webstorm/>
- [44] N. L. Inc., "Notion." 2024. [Online]. Available: <https://www.notion.so>
- [45] G. Inc., "GitHub: Let's Build from Here." 2024. [Online]. Available: <https://github.com>
- [46] F. Inc., "Figma: Design, Prototype, and Share." 2024. [Online]. Available: <https://www.figma.com>
- [47] G. Developers, "Tipos de lugar de la API de Places." 2024. [Online]. Available: <https://developers.google.com/maps/documentation/places/web-service/place-types?hl=es-419#food-and-drink>
- [48] G. Support, "Agregar o reclamar la ficha de tu empresa en Google My Business." 2024. [Online]. Available: <https://support.google.com/business/answer/4801187?hl=es-419>

- [49] B. Klug, "An Overview of the System Usability Scale in Library Website and System Usability Testing," *Weave: Journal of Library User Experience*, vol. 1, no. 6, Apr. 2017, doi: 10.3998/weave.12535642.0001.602.
- [50] U. I. F. Mars, "¿Cómo medir la usabilidad? ¿Qué es SUS?" 2024. [Online]. Available: <https://www.uifrommars.com/como-medir-usabilidad-que-es-sus/>

5 ANEXOS

ANEXO I. Historias Épicas

Tabla 24. Historia Épica CAP01

Historia de usuario épica	
Identificador: CAP01	Prioridad: Alta
Título: Diseño de la interfaz y registro de preferencias por parte del turista	
Descripción: Como turista necesito una interfaz para especificar mis preferencias con el objeto de buscar restaurantes	
CAP= COME AL PASO	

Tabla 25. Historia Épica CAP02

Historia de usuario épica	
Identificador: CAP02	Prioridad: Alta
Título: Desarrollo de algoritmo para filtrar los restaurantes	
Descripción: Como turista quiero obtener los mejores restaurantes de acuerdo con mis preferencias y mi ubicación	
CAP= COME AL PASO	

Tabla 26. Historia Épica CAP03

Historia de usuario épica	
Identificador: CAP03	Prioridad: Alta
Título: Implementar rutas para llegar al restaurante	
Descripción: Como turista necesito obtener la ruta para llegar al restaurante de mi preferencia.	
CAP= COME AL PASO	

ANEXO II. Historias de usuario del Sprint 1

Tabla 27. Historia de Usuario CAP01-01

HISTORIA DE USUARIO	
Identificador: CAP01-01	Estimación: 13 puntos de historia
Título: Interfaz de preferencias del turista	
Prioridad: Alta	Estado: Por implementar
Descripción: Como turista deseo definir mis preferencias en una interfaz para buscar restaurantes que respondan a ellas.	
Conversación:	
Pregunta	Respuesta
¿Cuáles son las preferencias que se establecen en la aplicación?	Las preferencias tienen que ver con el servicio, el nivel de precio y los tipos de comida
¿Qué tipos de comidas se tienen que mostrar?	Los tipos de comida van a estar establecido los que se encuentran en la API de Google
¿Cuál es el nivel de precio y rating que se tiene que mostrar?	El nivel de precio y rating se establece lo que se presenta en la API de Google
¿La aplicación tiene que ser responsive?	Si, la aplicación debe funcionar bien tanto para PC o dispositivos móviles
Criterios de aceptación	<p>Búsqueda de preferencias Dado que existe un formulario para elegir mis preferencias, Cuando selecciono todos los campos de preferencia y presiono el botón de búsqueda, Entonces se muestra restaurantes que cumplen con las preferencias seleccionadas.</p>
CAP= COME AL PASO	

Tabla 28. Historia de Usuario CAP01-02

HISTORIA DE USUARIO	
Identificador: CAP01-02	Estimación: 8 puntos de historia
Título: Registro de información del turista en la aplicación	
Prioridad: Alta	Estado: Por implementar
Descripción: Como turista requiero registrar mi información a través de un formulario para próximos usos de la aplicación.	
Conversación:	
Pregunta	Respuesta
¿Qué información debe ir en el formulario de registro?	Los datos que se piden en el formulario son el nombre y correo. No se pide contraseña. La aplicación no debe ser intrusiva, sino facilitar el acceso a la misma de manera rápida.
¿Qué sucede después de completar el registro?	La aplicación te va a permitir guardar las preferencias seleccionadas para usarlos en el futuro.
Criterios de aceptación	Registro Exitoso con Nombre y Correo Válidos Dado que existe un formulario de registro, Cuando ingreso mi nombre y correo válidos, Y hago clic en el botón registrarme, Entonces mi información se almacena en la base de datos correctamente, Y se muestra un mensaje de confirmación de registro exitoso.
	Mensaje de Datos Inválidos al Ingresar Información Incorrecta o campos vacíos Dado que existe un formulario de registro, Cuando ingreso un nombre o correo inválido, Y hago clic en el botón registrarme, Entonces aparece un mensaje indicando que los datos ingresados son inválidos.
	Mensaje de Correo Existente al Registrar un Correo ya Utilizado Dado que existe un formulario de registro, Cuando ingreso mi nombre y un correo que ya está registrado en la base de datos, Y hago clic en el botón registrarme, Entonces se muestra un mensaje informando que el correo ingresado ya ha sido registrado previamente.
CAP= COME AL PASO	

Tabla 29. Historia de Usuario CAP01-03

HISTORIA DE USUARIO	
Identificador: CAP01-03	Estimación: 6 puntos de historia
Título: Almacenar preferencias del turista	
Prioridad: Alta	Estado: Por implementar
Descripción: Como turista deseo que las preferencias seleccionadas se almacenen para poder usarlas en próximas búsquedas.	
Conversación:	
Pregunta	Respuesta
¿Cómo se guardan las preferencias?	Las preferencias se guardan en la base de datos, y están vinculadas al turista registrado.
¿Es necesario que este registrado para guardarlas?	Si, ya que las preferencias se vinculan a la cuenta creada
Criterios de aceptación	Almacenamiento de Todas las Preferencias en la Base de Datos Dado que he seleccionado mis preferencias, Cuando hago clic en buscar, Entonces las preferencias elegidas se guardan correctamente en la base de datos.
CAP= COME AL PASO	

ANEXO III. Historias de usuario del Sprint 2

Tabla 30. Historia de Usuario CAP02-01

HISTORIA DE USUARIO	
Identificador: CAP02-01	Estimación: 8 puntos de historia
Título: Obtener restaurantes cerca de la ubicación geográfica del turista	
Prioridad: Alta	Estado: Por implementar
Descripción: Como turista quiero conocer los restaurantes que se encuentren cerca de mi ubicación	
Conversación:	
Pregunta	Respuesta
¿A que distancia deben estar los restaurantes?	Los restaurantes deben estar a 5km de la posición del turista
¿Cuáles son las condiciones para mostrar el restaurante?	Estar abierto en esa hora, localizarse a 5km de la posición del turista y ser de tipo restaurante
Criterios de aceptación	Dado que se requiere la ubicación del turista. Cuando el turista comparte su ubicación. Entonces se muestran restaurantes abiertos y dentro de un radio de 5km de su ubicación.
CAP= COME AL PASO	

Tabla 31. Historia de Usuario CAP02-02

HISTORIA DE USUARIO	
Identificador: CAP02-02	Estimación: 13 puntos de historia
Título: Filtrar restaurantes en base a las preferencias	
Prioridad: Alta	Estado: Por implementar
Descripción: Como turista deseo filtrar los restaurantes de acuerdo con mis preferencias, para evitar la sobrecarga de información.	
Conversación:	
Pregunta	Respuesta
¿De dónde se obtienen las preferencias del turista?	Las preferencias se obtienen de la base de datos. Si el turista tiene preferencias registradas, se filtrarán los resultados de acuerdo con estas.

Si el turista tiene varias preferencias almacenadas, ¿qué preferencias tendrán prioridad?	Si el turista tiene múltiples preferencias, se mostrarán las tres opciones más buscadas por el turista.
Criterios de aceptación	Dado que el turista ha establecido sus preferencias. Cuando realiza la búsqueda de restaurantes, entonces se recomiendan restaurantes que cumplen con las preferencias establecidas y que se encuentren abiertos y dentro de un radio de 5Km de la ubicación del turista.
CAP= COME AL PASO	

Tabla 32. Historia de Usuario CAP02-03

HISTORIA DE USUARIO	
Identificador: CAP02-03	Estimación: 13 puntos de historia
Título: Obtener detalle de los restaurantes	
Prioridad: Alta	Estado: Por implementar
Descripción: Como turista deseo obtener información del restaurante para poder tomar una buena decisión	
Conversación:	
Pregunta	Respuesta
¿Qué información va a mostrar del restaurante?	Información como horario de operación, imágenes, página web, número de teléfono, dirección, ambiente y opciones de servicio
¿De dónde se va a obtener la información del restaurante?	Toda la información del restaurante será de la API places de Google
¿Qué sucede si un restaurante específico no contiene un dato que otros sí?	Si un restaurante no posee un dato, este dato no será mostrado en ese restaurante
Criterios de aceptación	Dado que tengo una lista de restaurantes. Cuando elija un restaurante. Entonces se muestra la información del restaurante elegido
CAP= COME AL PASO	

ANEXO IV. Historias de usuario del Sprint 3

Tabla 33. Historia de Usuario CAP02-02 actualizada para el Sprint 3

HISTORIA DE USUARIO	
Identificador: CAP02-02	Estimación: 10 puntos de historia
Título: Filtrar restaurantes en base a las preferencias	
Prioridad: Alta	Estado: En desarrollo
Descripción: Como turista deseo filtrar los restaurantes de acuerdo con mis preferencias, para evitar la sobrecarga de información.	
Conversación:	
Pregunta	Respuesta
¿De dónde se obtienen las preferencias del turista?	Las preferencias se obtienen de la base de datos. Si el turista tiene preferencias registradas, se filtrarán los resultados de acuerdo con estas.
Si el turista tiene varias preferencias almacenadas, ¿qué preferencias tendrán prioridad?	Si el turista tiene múltiples preferencias, se mostrarán las tres opciones más buscadas por el turista.
Criterios de aceptación	Dado que el turista ha establecido sus preferencias. Cuando realiza la búsqueda de restaurantes, entonces se recomiendan restaurantes que cumplen con las preferencias establecidas y que se encuentren abiertos y dentro de un radio de 5Km de la ubicación del turista.
CAP= COME AL PASO	

Tabla 34. Historia de Usuario CAP03-01

HISTORIA DE USUARIO	
Identificador: CAP03-01	Estimación: 13 puntos de historia
Título: Implementación de ruta hacia el restaurante	
Prioridad: Alta	Estado: Por implementar
Descripción: Como turista deseo obtener la ruta desde mi ubicación actual hasta llegar al restaurante seleccionado para facilitar mi llegada.	
Conversación:	
Pregunta	Respuesta

¿Qué ruta tiene que mostrar la aplicación?	Se debe mostrar la ruta a pie y a vehículo desde la ubicación del turista hacia el restaurante
¿Cuándo se muestra la ruta?	La ruta se muestra cuando el turista quiere conocer cómo llegar al restaurante. Únicamente cuando da clic en el botón de ruta.
¿De dónde se obtiene la ruta?	La ruta se va a obtener de la Routes API y tiene que ser la ruta a pie y a vehículo.
Criterios de aceptación	Dado que el turista ha seleccionado un restaurante, cuando solicita la ruta hacia el restaurante, entonces se muestra la ruta en el mapa para llegar desde su ubicación hasta el restaurante seleccionado
CAP= COME AL PASO	

Tabla 35. Historia de Usuario CAP01-04

HISTORIA DE USUARIO	
Identificador: CAP01-04	Estimación: 8 puntos de historia
Título: Iniciar sesión en la aplicación	
Prioridad: Media	Estado: Por implementar
Descripción: Como turista deseo ingresar a la aplicación usando Facebook o Gmail para poder buscar los restaurantes en base a mi preferencia.	
Conversación	
Pregunta	Respuesta
¿Cómo se debe iniciar sesión?	El inicio de sesión será a través de Facebook o Gmail. No se pedirá datos adicionales al turista
¿Se debe mantener iniciado la sesión?	Si, al iniciar sesión el turista permanece conectado hasta que cierre sesión.
Criterios de aceptación	Dado que el turista ingresa su correo y contraseña, cuando se verifique que las credenciales son correctas, entonces podrá acceder exitosamente a la aplicación.
CAP= COME AL PASO	

ANEXO V. Historias de usuario del Sprint 4

Tabla 36. Historia de Usuario CAP02-04

HISTORIA DE USUARIO	
Identificador: CAP02-04	Estimación: 8 puntos de historia
Título: Analizar sentimiento en comentarios	
Prioridad: Media	Estado: Por implementar
Descripción: Como turista deseo conocer los restaurantes con mejores comentarios, para evitar restaurantes con un mal servicio.	
Conversación:	
Pregunta	Respuesta
¿De dónde se van a obtener los comentarios?	Los comentarios se obtienen de la API de Google
¿Cómo se realizará el análisis de sentimiento?	El análisis de sentimiento se llevará a cabo utilizando la API de IBM, que proporciona un servicio para analizar el texto.
¿Cómo se clasificará los restaurantes?	Los restaurantes se clasificarán de mayor a menor puntuación, según el promedio de las calificaciones de sus comentarios.
Criterios de aceptación	Dado que se obtiene los comentarios del restaurante. Cuando se realiza el análisis de sentimiento de cada comentario, entonces se obtiene una puntuación que permite ordenar los restaurantes con mejor puntuación al peor.
CAP= COME AL PASO	

ANEXO VI. Mockup de las interfaces de usuario

Inicio de sesión



Figura 41. Diseño de la página de inicio (Realizado en Figma)

Página de preferencias



Figura 42. Diseño de la interfaz de preferencias (Realizado en Figma)

Página de recomendaciones

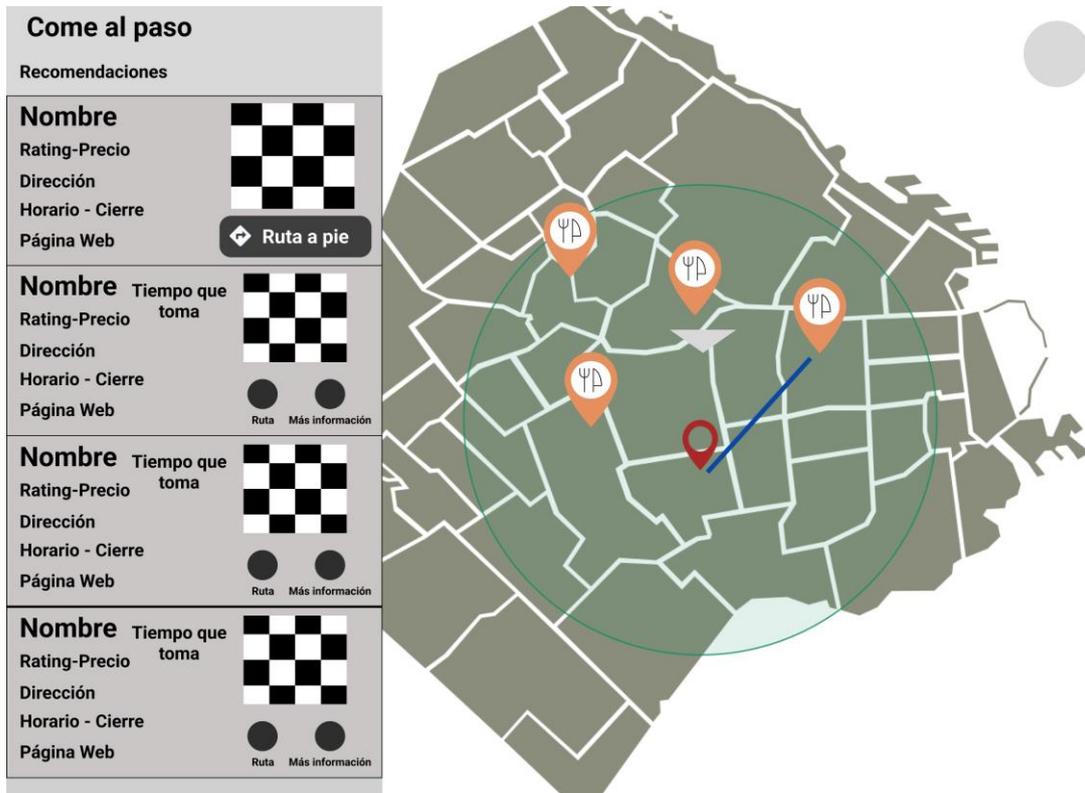


Figura 43. Diseño de la página de recomendaciones (Realizado en Figma)

ANEXO VII. Protocolo de evaluación



*ESCUELA POLITECNICA NACIONAL
INGENIERÍA DE SOFTWARE*

Protocolo de evaluación

Planificación de la evaluación

1. Propósito de la prueba

Evaluar la usabilidad por parte del turista al interactuar con el sistema de recomendación de restaurantes “Come al Paso”, diseñada para recomendar restaurantes basada en la ubicación e intereses del turista.

2. Descripción del problema

El sistema de recomendación de restaurantes, parte de un proyecto de titulación, se ha concebido con el propósito de ofrecer sugerencias gastronómicas personalizadas a los turistas, tomando en cuenta tanto su ubicación como sus preferencias individuales. Con esta aplicación, los turistas podrán establecer un radio de búsqueda desde su ubicación y refinar las opciones presentadas según la especialidad culinaria, el rango de precios o las puntuaciones otorgadas por Google. Los establecimientos recomendados serán aquellos mejor evaluados por Google y posteriormente clasificados mediante un análisis de los comentarios de sus visitantes, asegurando así una selección de calidad. Además, la aplicación tiene la capacidad de ser utilizada en cualquier lugar del mundo, lo que garantiza que los restaurantes sugeridos se encuentren abiertos y que la información presentada esté siempre actualizada. Los turistas tendrán la libertad de explorar las recomendaciones basadas en sus intereses o de descubrir nuevas opciones dentro de la zona en la que se encuentren.

Es crucial, por tanto, evaluar la aplicación en su conjunto para comprender la satisfacción del turista en cuanto a su usabilidad. Este análisis proporcionará insights valiosos que servirán como base para trabajo futuro.

3. Perfil de participantes

Grupo de usuario

Turista: Persona que visita una ciudad y requiere identificar restaurantes de su interés dentro de un área geográfica.

Geografía: Personas que residen en Quito y que poseen un computador con acceso a internet

Demografía: Edad comprendida entre 20 a 30 años

Comportamiento: Personas que usan la tecnología de forma natural; usan aplicaciones que les permiten acceder a nuevas experiencias.

Preferencias: Encontrar nuevos tipos de comida dentro del área donde se encuentran.

Necesidades: Tener recomendaciones de restaurantes que sean adecuadas a sus gustos y experiencias previas.

4. Metodología de la prueba

La prueba se llevará a cabo con participantes que dispongan de un ordenador con acceso a internet. Estos participantes realizarán una serie de tareas paso a paso. Una vez completadas las tareas, se les solicitará que respondan a un cuestionario compuesto por 10 preguntas utilizando una escala Likert de 5 puntos, basadas en el Cuestionario de Usabilidad del Sistema (SUS). Además, se les pedirá que completen un segundo cuestionario donde podrán proporcionar comentarios adicionales para futuras mejoras.

5. Lista de tareas

Los participantes seguirán las tareas que se encuentran a continuación y que forman parte del protocolo de evaluación.

Escenario: Eres una persona que quiere conocer los restaurantes que se encuentran dentro del área donde te encuentras. Para ello vas a usar un sistema de recomendación de restaurantes llamado "Come al Paso", tu objetivo es navegar por

la aplicación y obtener recomendaciones en base a tus gustos y el lugar en donde te encuentras. Para cumplir con tu objetivo vas a realizar las siguientes tareas.

Tarea 1: Ingresar a la página web

- Ingresar al enlace de la página web
- Compartir la ubicación

Tarea 2: Registro e inicio de sesión

- Ingresar a la aplicación usando Facebook o Gmail.

Tarea 3: Exploración de restaurantes cercanos

- Navegar la lista de restaurantes
- Seleccionar un restaurante de su preferencia
- Explorar la información del restaurante
- Conocer la ruta a pie y en vehículo hacia el restaurante

Tarea 4: Filtrado de restaurantes según preferencias

- Dirigirse al filtro de restaurantes
- Seleccionar cualquier opción de nivel de precio, clasificación y especialidades
- Filtrar la oferta de restaurantes con las opciones escogidas
- Explorar las opciones de restaurantes filtrados

Tarea 5: Exploración de nuevas recomendaciones en otra área

- Dirigirse al apartado de buscar restaurantes
- Hacer la búsqueda en un radio a elección
- Explorar el apartado “Para ti” y “Cerca de ti” en la nueva área

Tarea 6: Cierre de sesión

- Cerrar sesión en la aplicación

6. Entorno de prueba

La evaluación se llevará a cabo en sesiones individuales según la disponibilidad de los participantes, con fecha límite hasta el 21 de mayo de 2024. Se estima que la prueba tenga una duración de 10 a 20 minutos por participante.

7. Datos a recolectar

Se recogerán puntuaciones del cuestionario SUS, así como la percepción general de los participantes sobre la aplicación.

8. Contenido del informe

El informe presentará los resultados de las evaluaciones realizadas por cada participante, así como el puntaje general del SUS para la aplicación. Se incluirá un apartado de consideraciones futuras basadas en los comentarios de los participantes para posibles mejoras.

9. Recursos de evaluación

El encargado de llevar a cabo la prueba tendrá que tener en cuenta las siguientes actividades antes de comenzar la evaluación:

- Informar al participante de la evaluación a realizar
- Preparar la página web para la evaluación
- Enviar el documento "Guía de prueba". El documento contiene el enlace a la aplicación, las encuestas y tareas a realizar
- Recordar al participante de realizar la evaluación
- Indicar al participante de poseer las tareas a la mano

Los enlaces para acceder a las pruebas son los siguientes:

- Prueba de usabilidad: <https://forms.gle/mUHx3BgY98LRvTHC6>
- Comentarios para trabajo futuro: <https://forms.gle/feDzanp28ye9qQrTA>

ANEXO VIII. Guía de prueba



ESCUELA POLITECNICA NACIONAL INGENIERÍA DE SOFTWARE

Antecedentes

El sistema de recomendación de restaurantes, parte de un proyecto de titulación, se ha concebido con el propósito de ofrecer sugerencias gastronómicas personalizadas a los turistas, tomando en cuenta tanto su ubicación como sus preferencias individuales. Con esta aplicación, los turistas podrán establecer un radio de búsqueda desde su ubicación y refinar las opciones presentadas según la especialidad culinaria, el rango de precios o las puntuaciones otorgadas por Google. Los establecimientos recomendados serán aquellos mejor evaluados por Google y posteriormente clasificados mediante un análisis de los comentarios de sus visitantes, asegurando así una selección de calidad. Además, la aplicación tiene la capacidad de ser utilizada en cualquier lugar del mundo, lo que garantiza que los restaurantes sugeridos se encuentren abiertos y que la información presentada esté siempre actualizada. Los turistas tendrán la libertad de explorar las recomendaciones basadas en sus intereses o de descubrir nuevas opciones dentro de la zona en la que se encuentren.

Es crucial, por tanto, evaluar la aplicación en su conjunto para comprender la satisfacción del turista en cuanto a su usabilidad. Este análisis proporcionará insights valiosos que servirán como base para trabajo futuro.

Guía para la realización de la prueba

A continuación, se presenta los pasos para poder probar el sistema de recomendación de restaurantes “Come al paso”.

Escenario: Eres una persona que quiere conocer los restaurantes que se encuentran dentro del área donde te encuentras. Para ello vas a usar un sistema de recomendación de restaurantes llamado “Come al Paso”, tu objetivo es navegar por la aplicación y obtener recomendaciones en base a tus gustos y el lugar en donde te encuentras. Para cumplir con tu objetivo vas a seguir los siguientes pasos.

Paso 1: Ingresar a la página web: <https://timely-wisp-c6585f.netlify.app/>

Al ingresar a la página web por primera vez, encontrará un cuadro de diálogo similar al de la Ilustración 1, que **solicita permisos** para acceder a su **ubicación**. Es importante aceptar este permiso, ya que el sistema necesita conocer su ubicación para funcionar correctamente.

1
AL VISITAR LA WEB ACEPTA LA PARTICIPACIÓN EN LA PRUEBA



ESCUELA POLITECNICA NACIONAL
INGENIERÍA DE SOFTWARE

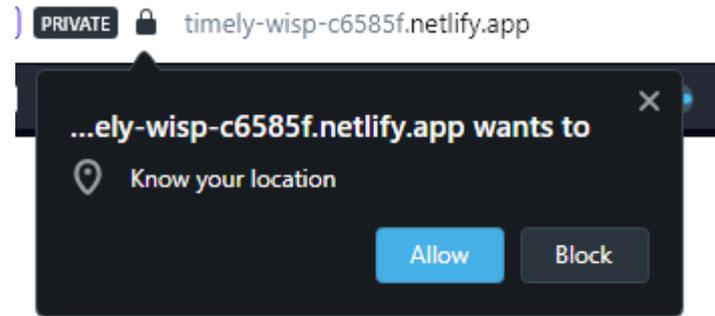


Ilustración 1. Cuadro de diálogo para solicitar permisos de ubicación

Paso 2: Registro e inicio de sesión

Como nuevo turista del sistema, deberá registrarse. La aplicación ofrece la opción de registrarse con su cuenta de Facebook o Gmail como se muestra en la Ilustración 2. Una vez seleccionada la opción deseada y completados los datos requeridos, será redirigido a la página principal.

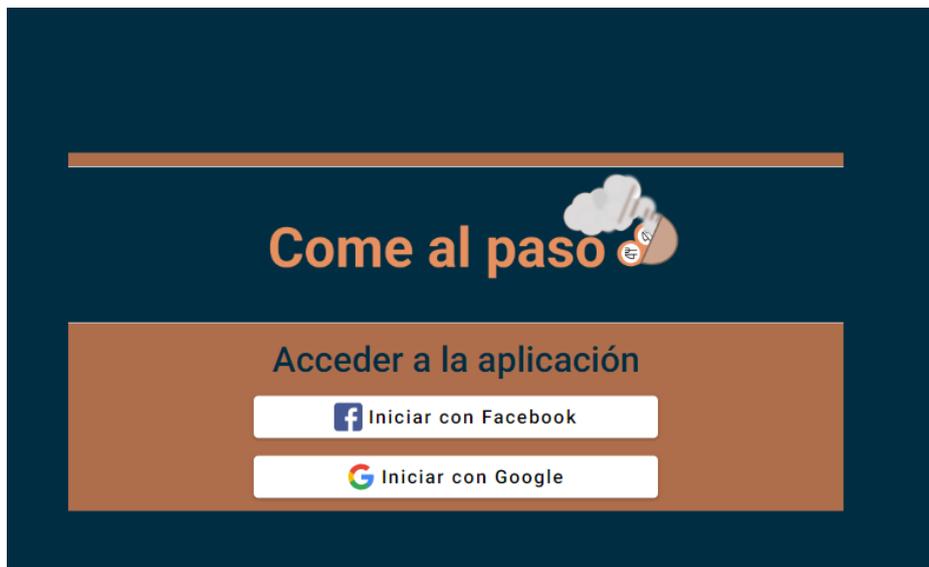


Ilustración 2. Pantalla para realizar registro e inicio de sesión



ESCUELA POLITECNICA NACIONAL INGENIERÍA DE SOFTWARE

Paso 3: Exploración de restaurantes cercanos

Una vez dentro del sistema, comenzará la búsqueda de restaurantes en un radio predeterminado de 5 km desde su ubicación, como se muestra en la Ilustración 3. Espere a que la búsqueda finalice.

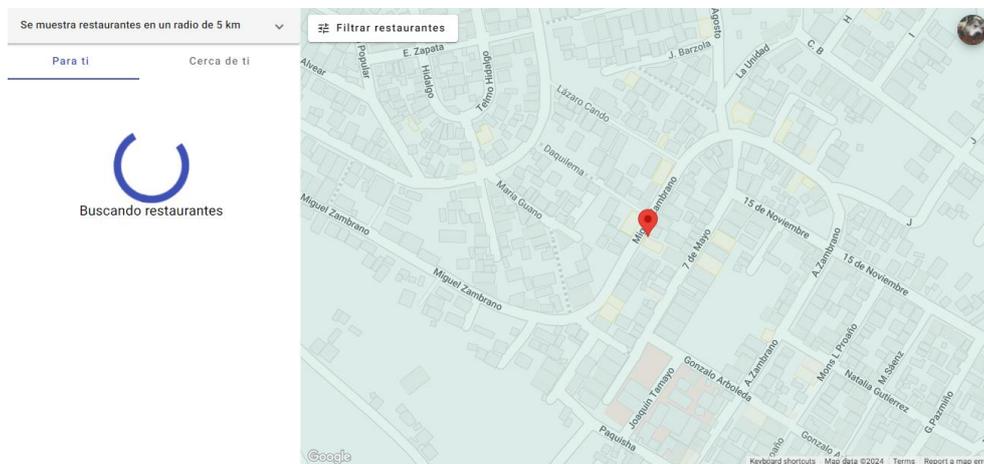


Ilustración 3. Búsqueda inicial de restaurantes

Dado que es un turista nuevo, no verá resultados en la pestaña "Para ti" (Ilustración 4). En su lugar, diríjase a la pestaña "Cerca de ti".

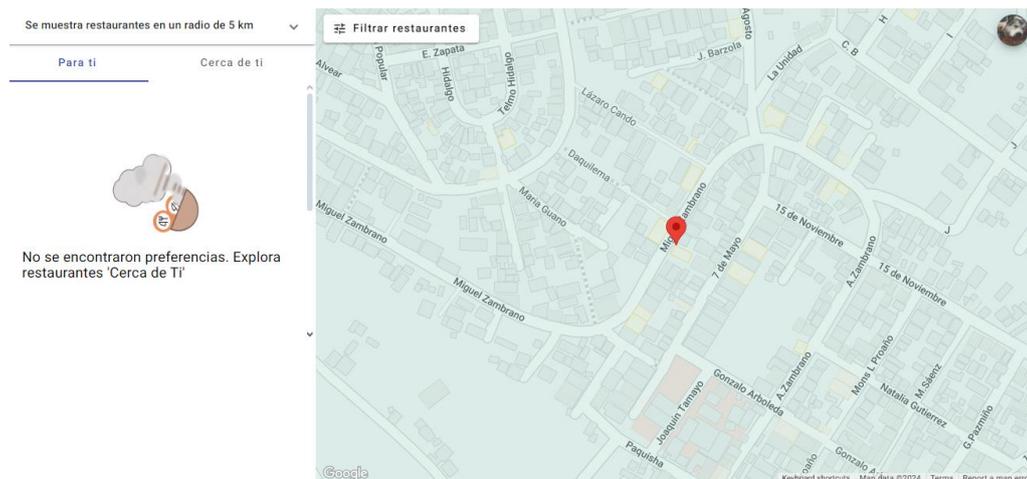


Ilustración 4. Estado de la aplicación para un turista nuevo



ESCUELA POLITECNICA NACIONAL INGENIERÍA DE SOFTWARE

En la pestaña "Cerca de ti", encontrará una selección de los mejores restaurantes evaluados por Google y posteriormente clasificados mediante un análisis de los comentarios de sus visitantes, abiertos en ese momento y dentro de un radio de 5 km. La Ilustración 5 muestra la oferta de restaurantes en el mapa.

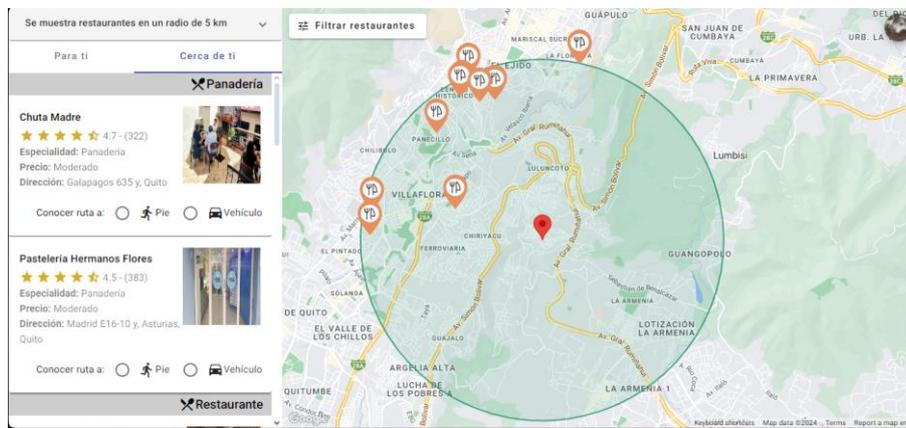


Ilustración 5. Restaurantes cerca de la ubicación del turista

Seleccione uno de los restaurantes mostrados para obtener más información, como se muestra en la Ilustración 6, incluyendo descripción, servicios y horarios de atención.

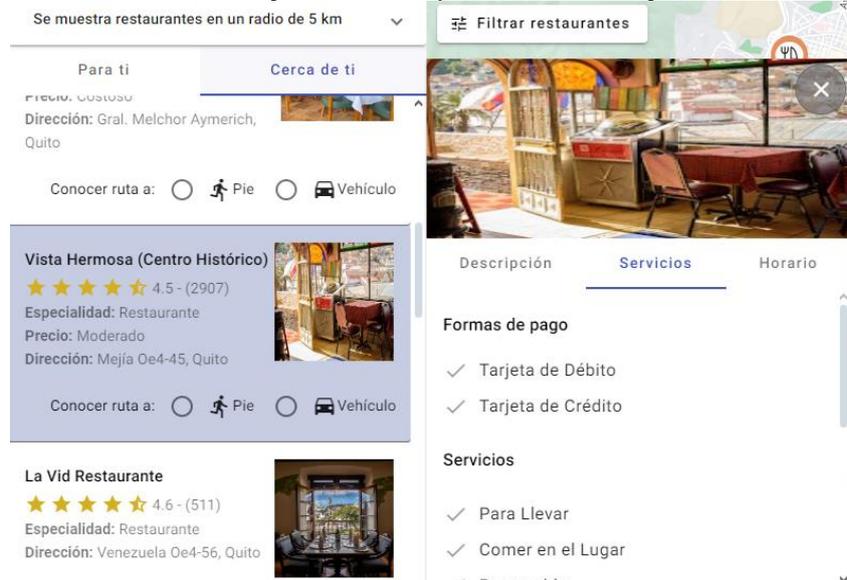


Ilustración 6. Información de un restaurante seleccionado



ESCUELA POLITECNICA NACIONAL INGENIERÍA DE SOFTWARE

Además, podrá obtener indicaciones para llegar al restaurante, ya sea a pie o en vehículo, como se muestra en la Ilustración 7, con detalles como distancia y tiempo estimado.

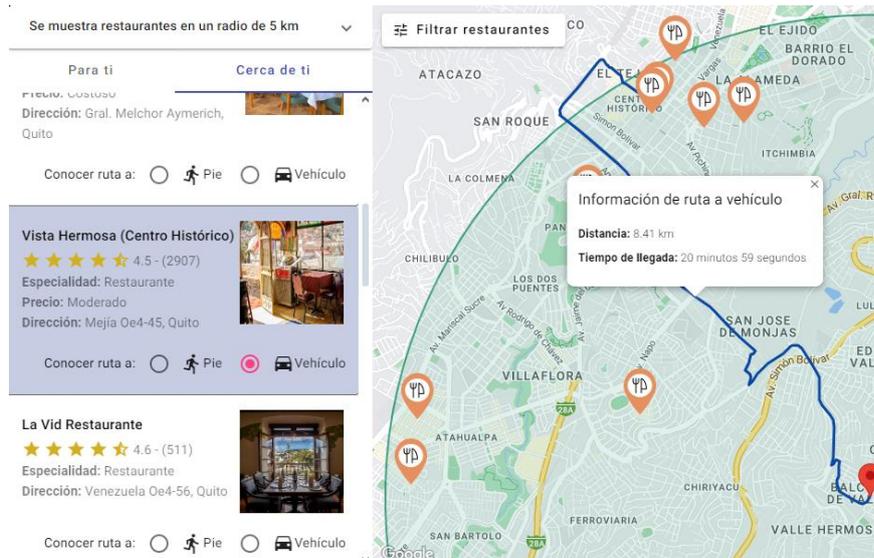


Ilustración 7. Conocer la ruta hacia un restaurante

Paso 4: Filtrado de restaurantes según preferencias.

El sistema le ofrece la opción de filtrar los restaurantes según sus preferencias, marcando el área de filtrado indicada en la Ilustración 8 con un recuadro rojo.

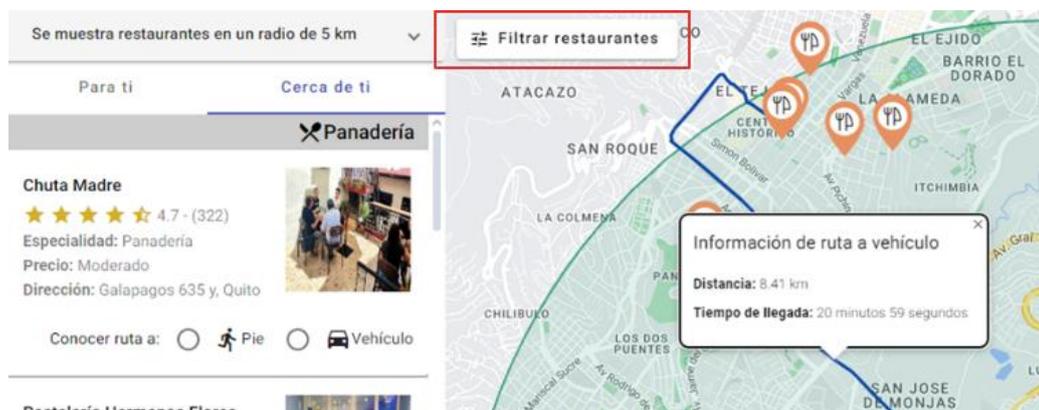


Ilustración 8. Dirigirse al apartado de filtros



ESCUELA POLITECNICA NACIONAL INGENIERÍA DE SOFTWARE

Se abrirá una ventana con varias opciones de filtrado, como clasificación, nivel de precio y especialidades (Ilustración 9). Una vez seleccionadas las opciones deseadas, realice la búsqueda.

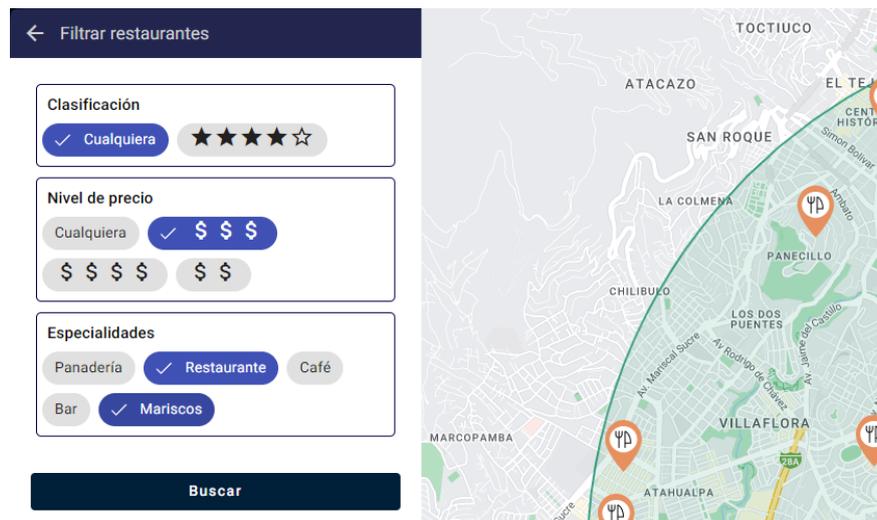


Ilustración 9. Filtros para buscar restaurantes

El nuevo listado de restaurantes basado en los filtros aplicados se mostrará, como se ve en la Ilustración 10. Puede ajustar o limpiar los filtros según sea necesario para obtener recomendaciones más precisas.

Para pasar al siguiente paso limpiar los filtros aplicados.

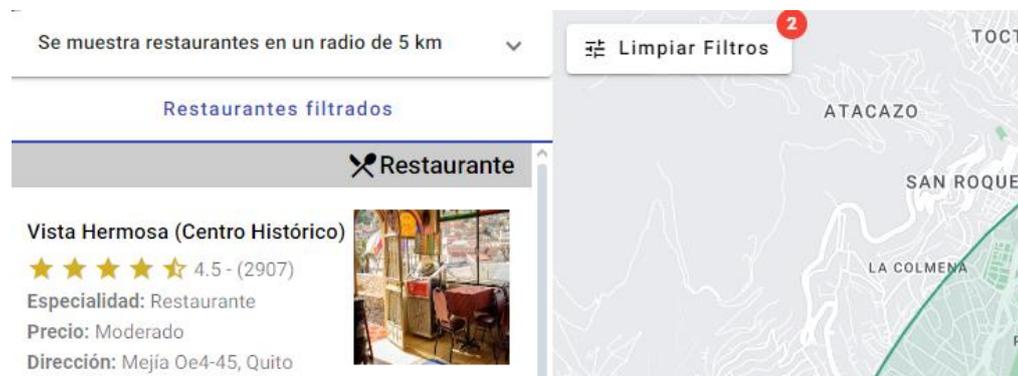


Ilustración 10. Limpiar los filtros aplicados



ESCUELA POLITECNICA NACIONAL INGENIERÍA DE SOFTWARE

Paso 5: Exploración de nuevas recomendaciones en otra área

Para buscar restaurantes en un área diferente, diríjase al apartado señalado en la Ilustración 11 con un recuadro rojo.

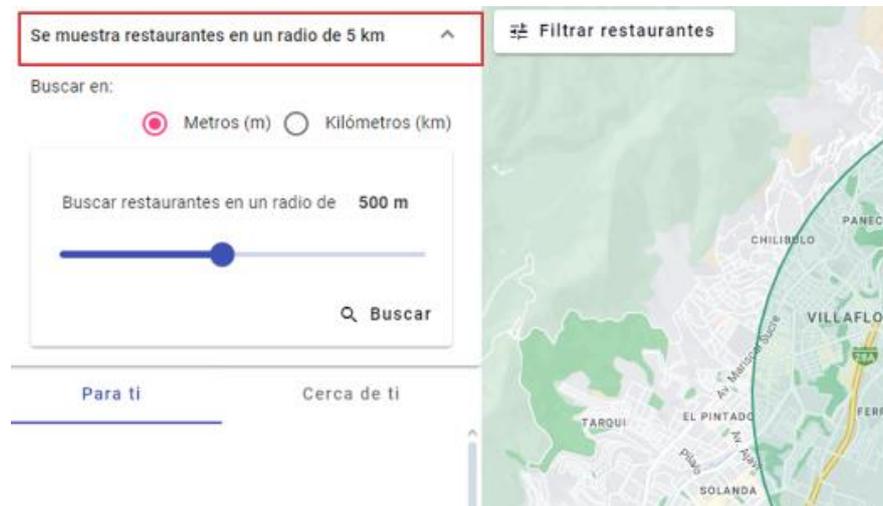


Ilustración 11. Acceder a la búsqueda en área

Al realizar una nueva búsqueda en un radio diferente, la oferta de restaurantes se actualizará, e incluso la sección "Para ti" ofrecerá opciones basadas en su historial previo, como se muestra en la Ilustración 12. Explore estas recomendaciones y evalúe su experiencia con la aplicación.

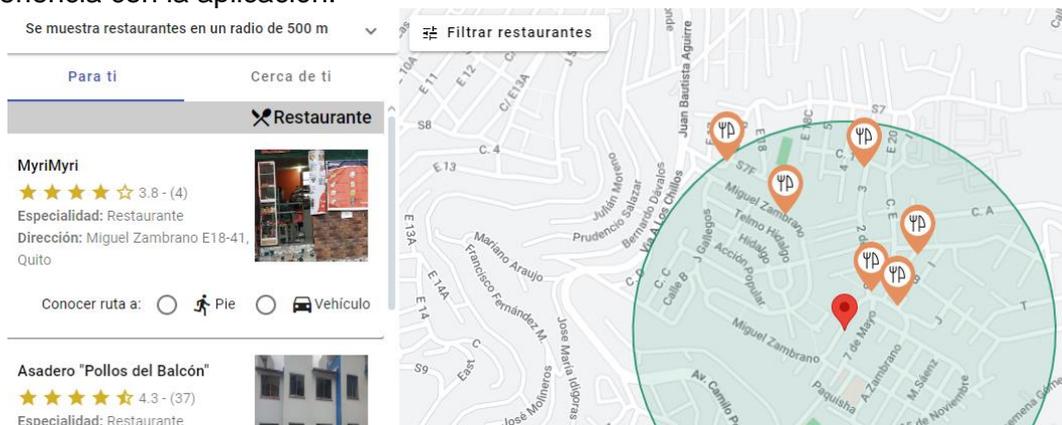


Ilustración 12. Restaurantes encontrados en una nueva área



*ESCUELA POLITECNICA NACIONAL
INGENIERÍA DE SOFTWARE*

Paso 6: Cierre de sesión

Después de explorar las recomendaciones y la interfaz, puede cerrar sesión seleccionando la opción indicada en la Ilustración 13, lo que le llevará de vuelta a la página de inicio.



Ilustración 13. Opción para cerrar sesión

Cierre de la prueba

Una vez completadas las tareas, le invitamos a compartir su experiencia con el sistema mediante dos encuestas:

- **Encuesta 1:** <https://forms.gle/mUHx3BqY98LRvTHC6>
- **Encuesta 2:** <https://forms.gle/feDzanzp28ye9qQrTA>

Agradecemos sinceramente su participación en esta prueba

ANEXO IX. Encuesta de usabilidad

Datos del participante

Ingrese su nombre y apellido *

Tu respuesta _____

Edad *

Tu respuesta _____

Seleccione su sexo *

Femenino

Masculino

¿Sector en el que se encuentra? *

Quito Sur

Quito Centro

Quito Norte

Valles (Cumbayá, Los Chillos)

Figura 44. Encuesta datos demográficos

Prueba de usabilidad

1. Creo que me gustaría utilizar este sistema con frecuencia *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

2. Encontré el sistema innecesariamente complejo *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

3. Pensé que el sistema era fácil de usar *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

5. Encontré que las diversas funciones de este sistema estaban bien integradas *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

Figura 45. Preguntas de 1 al 5 encuesta usabilidad

6. Pensé que había demasiada inconsistencia en este sistema *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

7. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

8. Encontré el sistema muy complicado de usar *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

9. Me sentí muy seguro usando el sistema *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

10. Necesitaba aprender muchas cosas antes de empezar con este sistema *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

Figura 46. Preguntas 6 al 10 encuesta usabilidad

ANEXO X. Respuestas encuesta de usabilidad

Datos demográficos

Ingrese su nombre y apellido

20 respuestas

Gabriela Salazar
Vickiann Jiménez
Andrés Cantuña
Anthony Chamba
Kevin Toasa
Alex Escobar
Jimmy Eche
Danny Cabrera

Figura 47. Respuestas del nombre de los participantes

Edad

20 respuestas

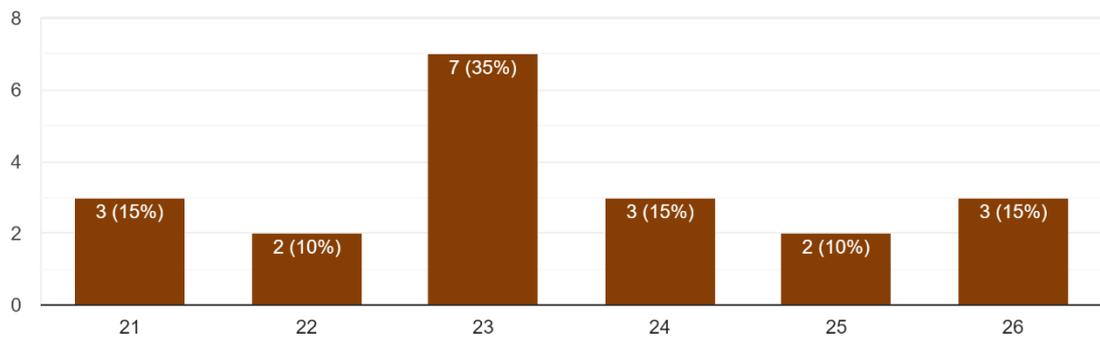


Figura 48. Respuestas de la edad de los participantes

Seleccione su sexo

20 respuestas

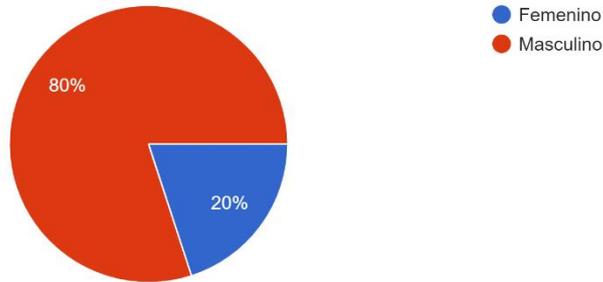


Figura 49. Respuesta al sexo de los participantes

¿Sector en el que se encuentra?

20 respuestas

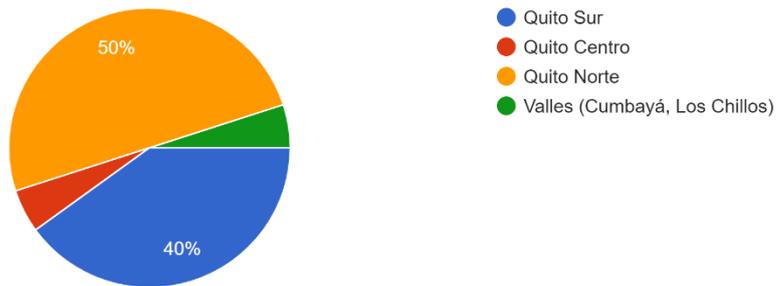


Figura 50. Respuesta al sector donde viven los participantes

Respuestas encuesta de usabilidad

1. Creo que me gustaría utilizar este sistema con frecuencia

20 respuestas

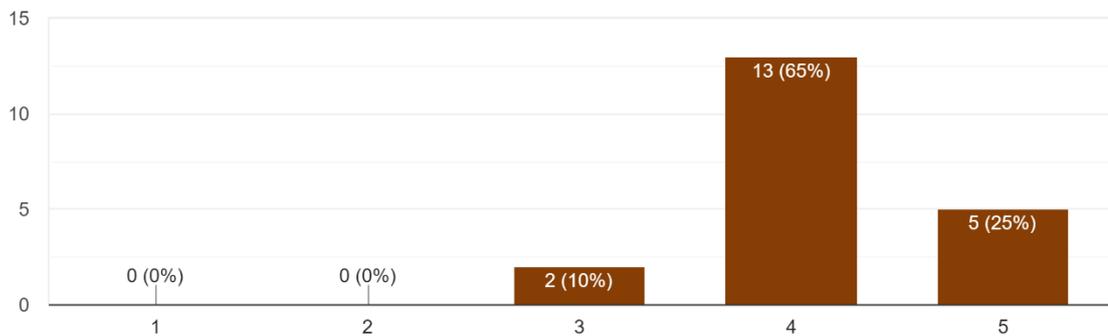


Figura 51. Respuesta de la pregunta 1 para usabilidad

2. Encontré el sistema innecesariamente complejo

20 respuestas

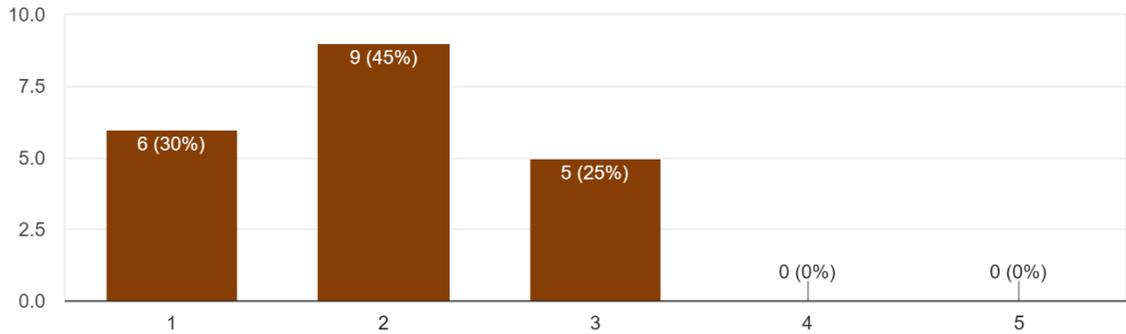


Figura 52. Respuesta de la pregunta 2 para usabilidad

3. Pensé que el sistema era fácil de usar

20 respuestas

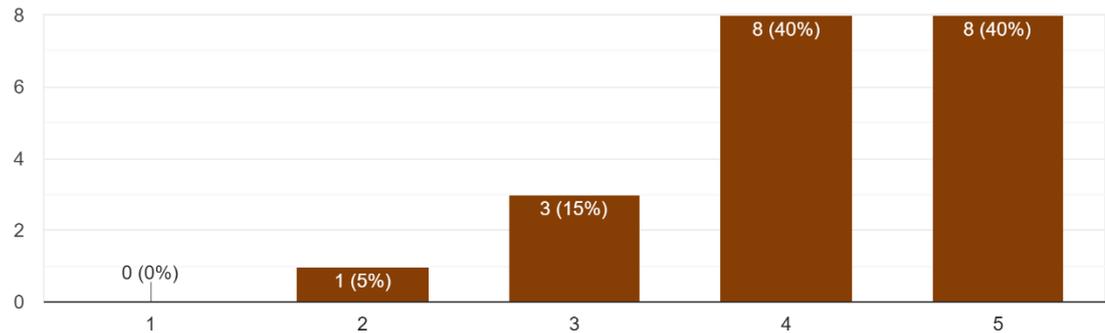


Figura 53. Respuesta de la pregunta 3 para usabilidad

4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema

20 respuestas

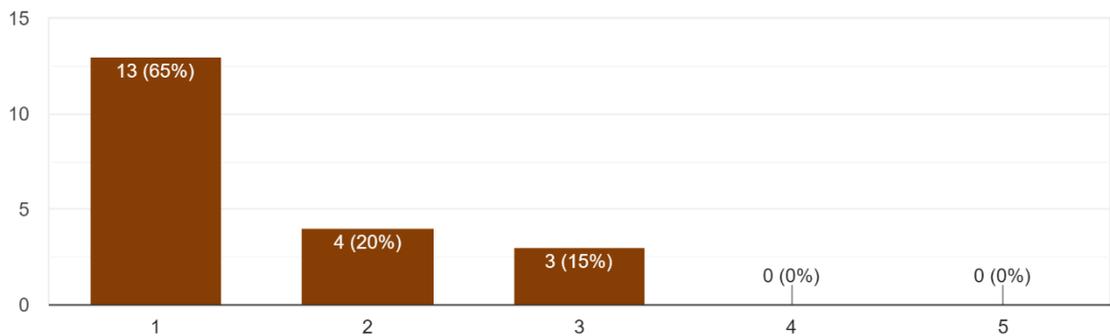


Figura 54. Respuesta de la pregunta 4 para usabilidad

5. Encontré que las diversas funciones de este sistema estaban bien integradas

20 respuestas

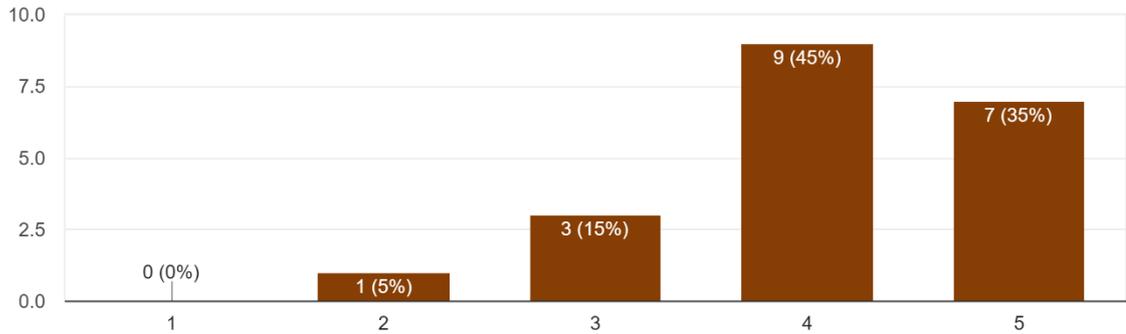


Figura 55. Respuesta de la pregunta 5 para usabilidad

6. Pensé que había demasiada inconsistencia en este sistema

20 respuestas

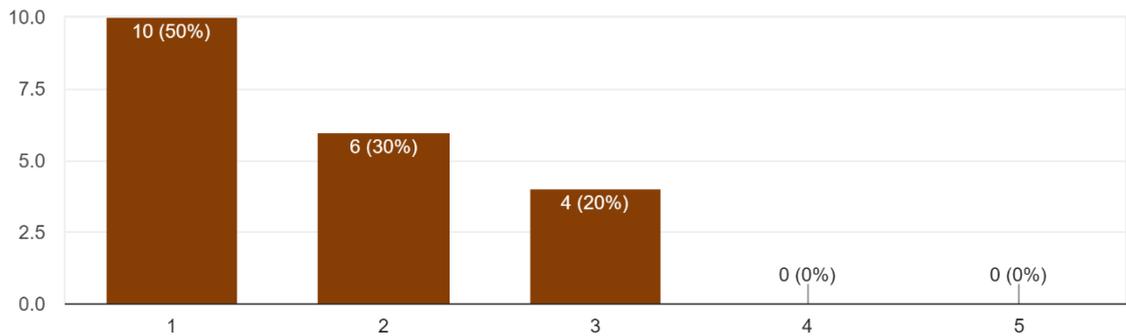


Figura 56. Respuesta de la pregunta 6 para usabilidad

7. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente

20 respuestas

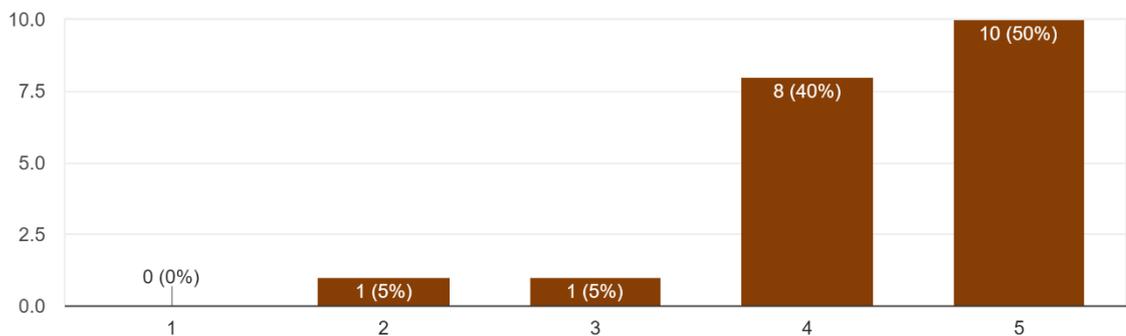


Figura 57. Respuesta de la pregunta 7 para usabilidad

8. Encontré el sistema muy complicado de usar

20 respuestas

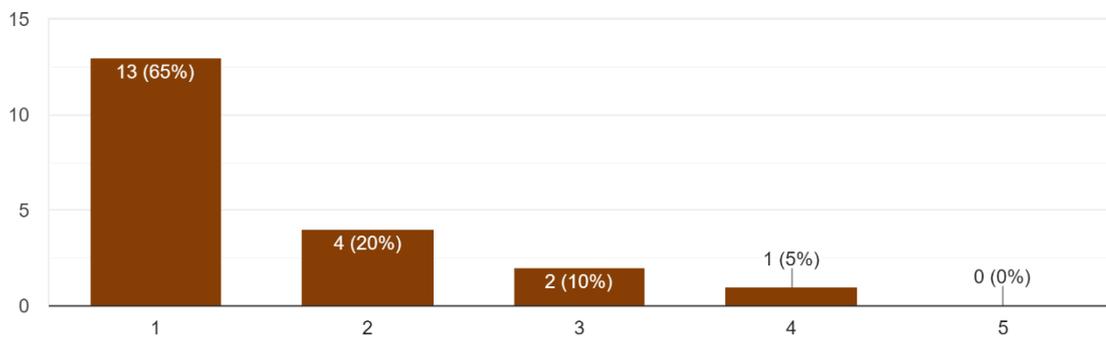


Figura 58. Respuesta de la pregunta 8 para usabilidad

9. Me sentí muy seguro usando el sistema

20 respuestas

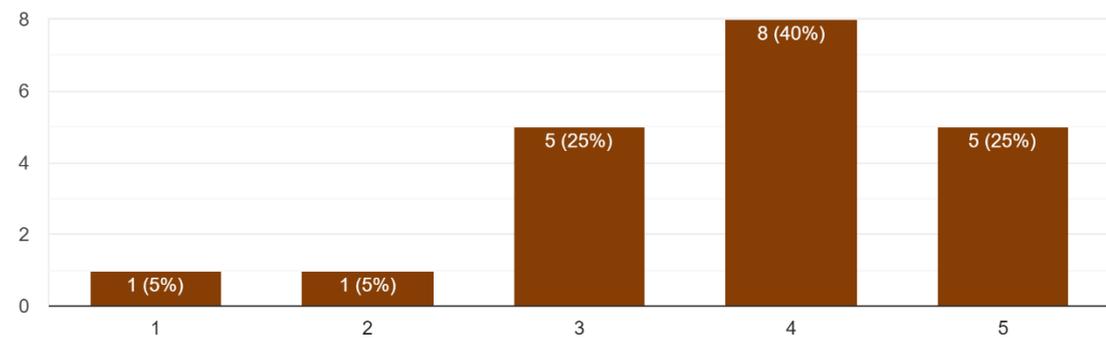


Figura 59. Respuesta de la pregunta 9 para usabilidad

10. Necesitaba aprender muchas cosas antes de empezar con este sistema

20 respuestas

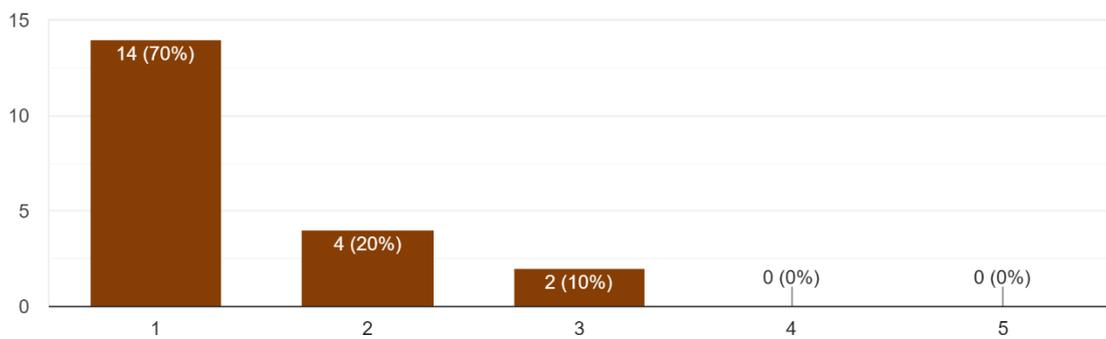


Figura 60. Respuesta de la pregunta 10 para usabilidad