

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**IMPLEMENTACIÓN DE SISTEMA RECOMENDADOR TURÍSTICO
INDICANDO LAS RESTRICCIONES DE SEGURIDAD PARA EL
COVID-19.**

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

Patrick David Cabezas Tapia

patrick.cabezas@epn.edu.ec

DIRECTORA: Ing. Regina Maritzol Tenemaza Vera, MSc.

maritzol.tenemaza@epn.edu.ec

QUITO, diciembre 2022

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por PATRICK DAVID CABEZAS TAPIA, bajo mi supervisión.

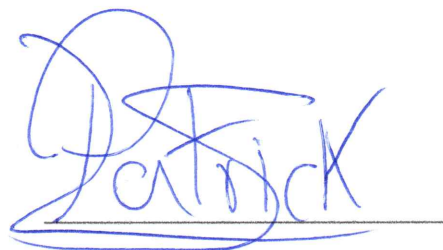
MSc. Maritzol Tenemaza

DIRECTOR DE PROYECTO

DECLARACIÓN

Yo PATRICK DAVID CABEZAS TAPIA, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

A handwritten signature in blue ink, appearing to read 'Patrick', written over a horizontal line.

PATRICK DAVID CABEZAS TAPIA

DEDICATORIA

A mi madre por su apoyo incondicional en todo momento.

A mi directora de tesis por su guía durante el desarrollo del proyecto.

A mis compañeros con los que compartí buenos momentos durante los años de la carrera.

A los profesores de la Escuela Politécnica Nacional por toda la sabiduría que me compartieron.

Patrick Cabezas

Tabla de contenido

1	INTRODUCCIÓN	1
1.1	Planteamiento del problema.....	1
1.2	Objetivos.....	1
1.2.1	Objetivo General.....	1
1.2.2	Objetivos Específicos.....	1
1.3	Marco Teórico.....	2
1.4	Herramientas, frameworks y librerías de desarrollo.....	3
2	METODOLOGÍA	6
2.1	Sprint 0.....	6
2.1.1	Arquitectura.....	6
2.1.2	Definición de roles.....	7
2.1.3	Definición de historias épicas.....	7
2.1.4	Product Backlog.....	8
2.1.5	Planificación del Release.....	9
2.2	Sprint 1.....	9
2.2.1	Sprint Planning.....	9
2.2.2	Ejecución del Sprint.....	10
2.2.3	Sprint Review.....	15
2.2.4	Medición de Velocidad de Avance del Proyecto.....	16
2.3	Sprint 2.....	17
2.3.1	Sprint Planning.....	17
2.3.2	Ejecución del Sprint.....	18
2.3.3	Sprint Review.....	21
2.3.4	Medición de Velocidad de Avance del Proyecto.....	22
2.4	Sprint 3.....	23
2.4.1	Sprint Planning.....	23
2.4.2	Ejecución del Sprint.....	24
2.4.3	Sprint Review.....	29
2.4.4	Medición de Velocidad de Avance del Proyecto.....	30
3	RESULTADOS Y DISCUSIÓN	32
3.1	Producto Final.....	32
3.1.1	Toma de Datos.....	32
3.1.2	Despliegue de los resultados.....	32
3.1.3	Despliegue de las Rutas.....	33

3.2	Pruebas	34
3.2.1	Usabilidad percibida	34
3.2.2	Pruebas de usabilidad basado en los principios de Jakob Nielsen ..	36
4	CONCLUSIONES Y RECOMENDACIONES	40
4.1	Conclusiones	40
4.2	Recomendaciones	41
5	REFERENCIAS BIBLIOGRÁFICAS	42
6	ANEXOS	44
6.1	Anexo I: Historias Épicas	44
6.2	Anexo II: Historias de Usuario	45
6.3	Anexo III: Resultados de las Encuestas Realizadas a los Usuarios Sobre Usabilidad de la Aplicación	47

ÍNDICE DE FIGURAS

Figura 1. Geolocalización satelital [6].....	2
Figura 2. Arquitectura de un Sistema de recomendación.....	2
Figura 3. Proceso de Scrum	4
Figura 4. Diagrama de flujo del algoritmo	6
Figura 5. Arquitectura MVC del proyecto	7
Figura 6. Cantidad de contagiados de Covid-19 en Quito por sector	11
Figura 7. Cantidad de contagiados de Covid-19 por cada sector.....	11
Figura 8. Cantidad de contagiados de Covid-19 en Quito de la fuente ModeMat.....	12
Figura 9 Google API Key.....	13
Figura 10. APIs de Lugares y Rutas.....	13
Figura 11. Script en Python.....	14
Figura 12. Archivo JSON obtenido.....	14
Figura 13. Servicios de Firebase utilizadas	14
Figura 14. Base de datos en Firebase	15
Figura 15. Gráfico de Burndown del Sprint 1.....	17
Figura 16. Versión de Android Studio.....	18
Figura 17. Versión de la API utilizada.....	19
Figura 18. Asignación de la API en Android Studio.....	19
Figura 19. Proyecto en GitHub	19
Figura 20. Maqueta de las pestañas de Registro, Logín y Menú	20
Figura 21. Maquetas del mapa con los POIs	20
Figura 22. Interfaces de usuario creadas en Android Studio	21
Figura 23. Gráfico de Burndown del Sprint 2.....	23
Figura 24. Código fuente de autenticación de los usuarios.....	24
Figura 25. Código fuente de registro de usuarios.....	25

Figura 26. Código fuente para cargar datos en el mapa.....	25
Figura 27. Lógica para cargar los marcadores de los POIs	25
Figura 28. Lógica para crear las rutas.....	26
Figura 29. Cambio de color del icono seleccionado.....	27
Figura 30. Dispositivo con Android 11.....	28
Figura 31. Dispositivo con Android 11.....	28
Figura 32. Emulador con Android 9.....	28
Figura 33. Gráfico de Burndown del Sprint 3.....	31
Figura 34. Menú de tomas de datos	32
Figura 35. Ejecución del mapa con los POIs y las rutas.....	33
Figura 36. Rutas entre los puntos	33
Figura 37. Promedio por pregunta de usabilidad percibida en el producto final.....	35
Figura 38. Puntaje de SUS [20].....	36
Figura 39. Resultados de las preguntas 1 sobre usabilidad.....	47
Figura 40. Resultados de las preguntas 2 sobre usabilidad.....	48
Figura 41. Resultados de las preguntas 3 sobre usabilidad.....	48
Figura 42. Resultados de las preguntas 4 sobre usabilidad.....	48
Figura 43. Resultados de las preguntas 5 sobre usabilidad.....	49
Figura 44. Resultados de las preguntas 6 sobre usabilidad.....	49
Figura 45. Resultados de las preguntas 7 sobre usabilidad.....	49
Figura 46. Resultados de las preguntas 8 sobre usabilidad.....	50
Figura 47. Resultados de las preguntas 9 sobre usabilidad.....	50
Figura 48. Resultados de las preguntas 10 sobre usabilidad.	50

ÍNDICE DE TABLAS

Tabla 1. Herramientas utilizadas durante el desarrollo del proyecto.	4
Tabla 2. Roles del equipo Scrum en el proyecto	7
Tabla 3. Historias épicas	8
Tabla 4. Product Backlog del Sprint 0.....	8
Tabla 5. Planificación del Release	9
Tabla 6. Historias de usuario del Sprint 1	9
Tabla 7. Sprint backlog del Sprint 1	10
Tabla 8. Niveles de contagio según la cantidad de casos confirmados.	12
Tabla 9. Pruebas de aceptación del Sprint 1	15
Tabla 10. Product Backlog adaptado del Sprint 1	16
Tabla 11. Historias de usuario del Sprint 2	17
Tabla 12. Sprint backlog del Sprint 2	18
Tabla 13. Pruebas de aceptación del Sprint 2.....	21
Tabla 14. Product Backlog adaptado del Sprint 2	22
Tabla 15. Historias de usuario del Sprint 3.....	23
Tabla 16. Sprint backlog del Sprint 3.....	23
Tabla 17. Pruebas de aceptación del Sprint 3.....	29
Tabla 18. Product Backlog adaptado del Sprint 3	30
Tabla 19. Resultado de la prueba de usabilidad SUS.....	35
Tabla 20. Historia Épica AH1	44
Tabla 21. Historia Épica AH2	44
Tabla 22. Historia Épica AH3	44
Tabla 23. Historia de Usuario AH1-01	45
Tabla 24. Historia de Usuario AH1-02	45
Tabla 25. Historia de Usuario AH2-01	46
Tabla 26. Historia de Usuario AH2-02	46
Tabla 27. Historia de Usuario AH3-01	46

RESUMEN

El trabajo de titulación presenta un sistema de recomendación turística en el que se identifican áreas con mayor nivel de contaminación de Covid-19 en la ciudad de Quito. Para analizar las áreas contaminadas se utilizaron datos publicados por el Ministerio de Salud Pública del Ecuador sobre número de personas contagiadas con Covid-19 por sector; Adicionalmente, el sistema de recomendación turística presenta Puntos de Interés (POIs) con mayor valoración turística, según el API de Google Maps. El desarrollo del Sistema de recomendación turística estuvo dirigido por el Framework SCRUM, usando por debajo Historias de Usuario traídas desde la Metodología Extreme Programming (XP).

Palabras Clave: Covid-19, POI, API de Google Maps, SCRUM, XP.

ABSTRACT

The degree work presents a tourism recommendation system that identifies areas with the highest level of Covid-19 contamination in the city of Quito. To analyze the contaminated areas, data published by the Ministry of Public Health of Ecuador on the number of people infected with Covid-19 by sector were used; Additionally, the tourist recommendation system presents Points of Interest (POIs) with higher tourist value, according to the Google Maps API.

The development of the tourist recommendation system was guided by the SCRUM Framework, using underneath User Stories brought from the Extreme Programming Methodology (XP).

Key words: Covid-19, POI, Google Maps API, SCRUM, XP.

1 INTRODUCCIÓN

1.1 Planteamiento del problema

La presencia del Covid 19 a nivel mundial, ha impuesto que las personas mantengan un distanciamiento entre ellos. A pesar de que en el momento actual la enfermedad ha comenzado a ceder en Ecuador, aún existen pequeños brotes en ciertas zonas de la ciudad de Quito. Cuando una persona requiere movilizarse a un determinado sitio turístico en las ciudades, necesita información que le permita conocer el grado de riesgo existente en esos lugares [1].

Actualmente existen varias aplicaciones con las cuales las personas pueden estar informadas sobre la cantidad de contagios en las diferentes ciudades del país, información administrada por el Ministerio de Salud Pública (MSP) [2] o el COE Metropolitano de Quito [3], donde se puede observar zonas con las respectivas cantidades de personas contagiadas. También la EPN ha implementado una herramienta (SALVAVIDAS) [4] con el fin de informar a las personas sobre el peligro y les permite estar alertas para evitar posibles contagios en esos lugares.

El presente proyecto propone el desarrollo de un sistema de recomendación turística en donde se expongan las rutas para llegar a sitios turísticos requeridos por el turista en cada Punto de Interés (POI) dicha información se tomará de los datos proporcionados por la API de Google Maps, y la información concerniente a las restricciones de seguridad para el COVID-19. Esto permitirá al turista facilitar la toma de decisiones.

1.2 Objetivos

1.2.1 Objetivo General

- Implementar un sistema de recomendación turística para la ciudad de Quito que indique información sobre las restricciones de seguridad para el COVID-19.

1.2.2 Objetivos Específicos

- Identificar servicios expuestos por la ciudad de Quito o por el COE Nacional, referentes a zonas o áreas de riesgo de contagio de Covid-19.
- Diseñar el sistema de recomendación turística considerando los servicios existentes y los puntos de interés de mayor importancia traídos desde Google de acuerdo con los intereses consultados al usuario.
- Probar el sistema de recomendación turística en la ciudad de Quito con al menos tres usuarios.
- Evaluar la usabilidad de la aplicación con los usuarios que participen en la prueba del sistema de recomendación turística.

1.3 Marco Teórico

Geolocalización

Es la capacidad de obtener la ubicación geográfica de un objeto que tenga acceso a internet [5] para logra obtener esta ubicación es necesario utilizar una serie de algoritmos de recopilación de datos, que generalmente utilizan direcciones de enrutamiento de red o dispositivos GPS (Global Positioning System: Sistema de Posicionamiento Global) integrados, los dispositivos hacen uso de una API para poder usar en una aplicación WEB [6].

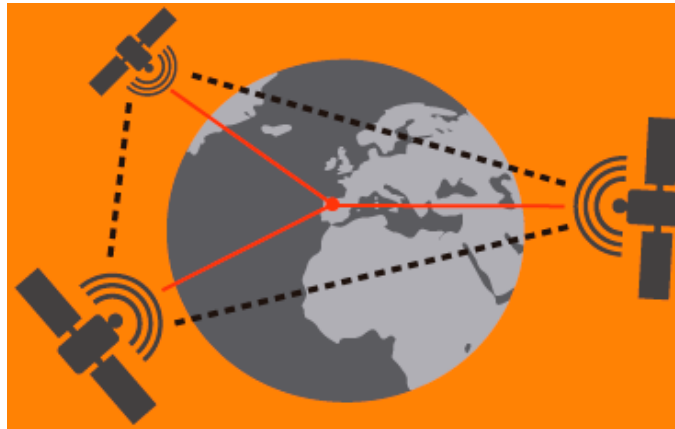


Figura 1. Geolocalización satelital [6].

Sistema de recomendación

Es un sistema de filtrado de información, el objetivo de este sistema es que los usuarios puedan obtener información filtrada en base algún tipo de condición previa [7], estos sistemas son útiles cuando el usuario debe elegir un producto entre una gran cantidad de elementos [8].

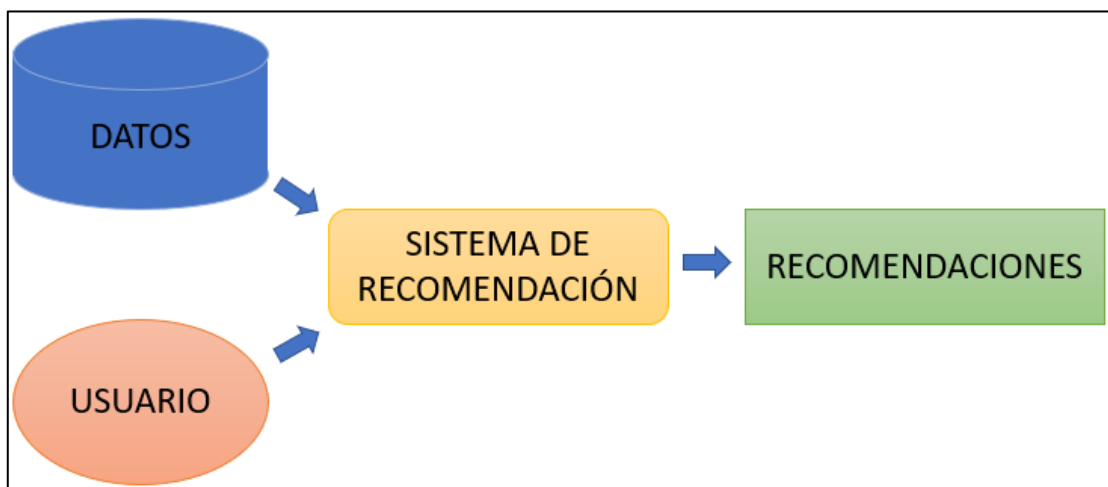


Figura 2. Arquitectura de un Sistema de recomendación

1.4 Herramientas, frameworks y librerías de desarrollo

Google Maps

Es el servicio de mapas de Google el cual ofrece una API para el consumo de varios servicios entre los que se encuentra la información de los POIs (puntos de interés) [9].

Android Studio

Es el IDE (Integrated Development Environment: Entorno de desarrollo integrado) oficial para el sistema Android de Google, basado en IntelliJ IDEA de JetBrains [10], es el IDE utilizado para el desarrollo de la aplicación.

Kotlin

Es un lenguaje de programación compilado de código abierto [11], diseñado para interoperar con Java, es compatible con la programación multiplataforma y permite mantener la flexibilidad y los beneficios de la programación nativa [12]. Kotlin apunta principalmente a JVM, pero también permite ser compilado a JavaScript [13]. Es el lenguaje utilizado para el desarrollo de la aplicación

Firebase

Es un servicio de back-end cloud de Google, pero también permite crear aplicaciones WEB y móviles [14]. El servicio de back-end utilizado para el proyecto.

Python

Es un lenguaje de programación interpretado de alto nivel [15]. Es el lenguaje utilizado para el desarrollo del script

Scrum

Es un marco de trabajo para el desarrollo ágil de software, está diseñado para equipos que dividen el trabajo en objetivos que se pueden completar en iteraciones de tiempo limitado, llamados Sprints, el equipo evalúa el proyecto en reuniones diarias llamados daily scrums, al finalizar el Sprint, el equipo realiza las reuniones: sprint review (revisar el trabajo realizado por todos miembros y los interesados) y Sprint retrospective (se discute las mejoras para el siguiente Sprint) [16] [17].

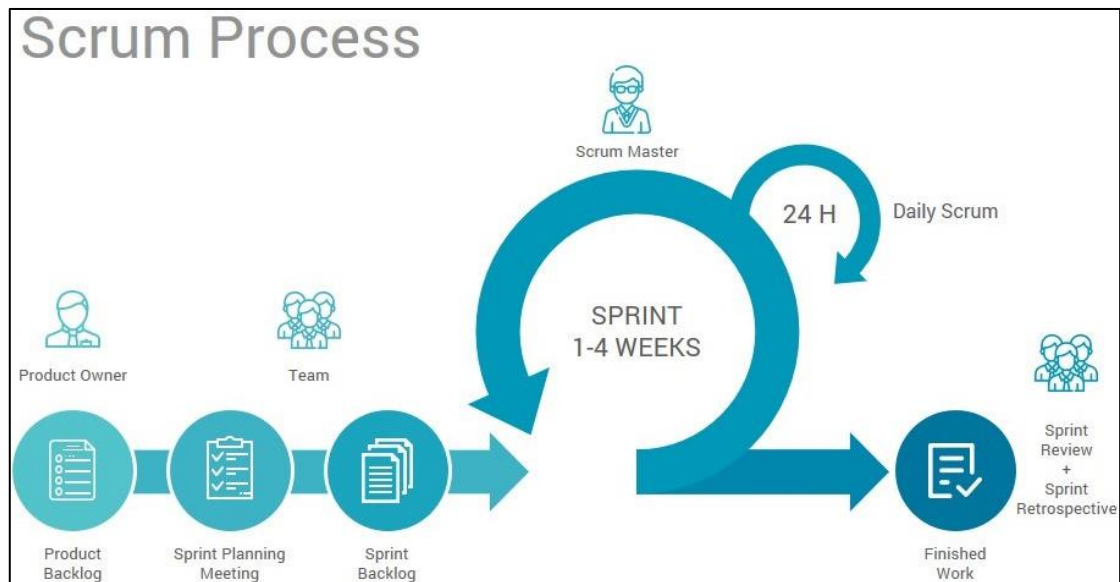









Figura 3. Proceso de Scrum

Figma

Es una herramienta para prototipos de diseño de interfaces colaborativo basado en la WEB, pero también cuenta con versiones para escritorio [18], en el caso del presente proyecto, esta herramienta será utilizada para realizar los Mockups de las interfaces de usuario.

Tabla 1. Herramientas utilizadas durante el desarrollo del proyecto.

Herramienta	Función	Logo
Google Maps	Consumo de APIs de lugares y rutas	
Android Studio	ID de desarrollo de la aplicación	
Kotlin	Lenguaje de programación de la aplicación	
Firebase	Base de datos en la nube	
Python	Lenguaje de programación para procesar los datos	

Scrum	Framework de desarrollo para el proyecto	
Figma	diseñar los prototipos de interfaces graficas.	

2 METODOLOGÍA

2.1 Sprint 0

El "Sprint 0" tuvo una duración aproximada de 1 mes, Donde se recopiló la información necesaria para iniciar el desarrollo del proyecto, principalmente en encontrar las fuentes de información publicada con los datos oficiales sobre el número de personas contagiadas con Covid-19 en la ciudad, también se creó la arquitectura del sistema, se definieron los roles del Scrum Team, las historias de usuario, el Product Backlog, la duración de los Sprints y se planificó el release.

2.1.1 Arquitectura

En la Figura 4 se visualiza el diagrama de flujo de la aplicación.

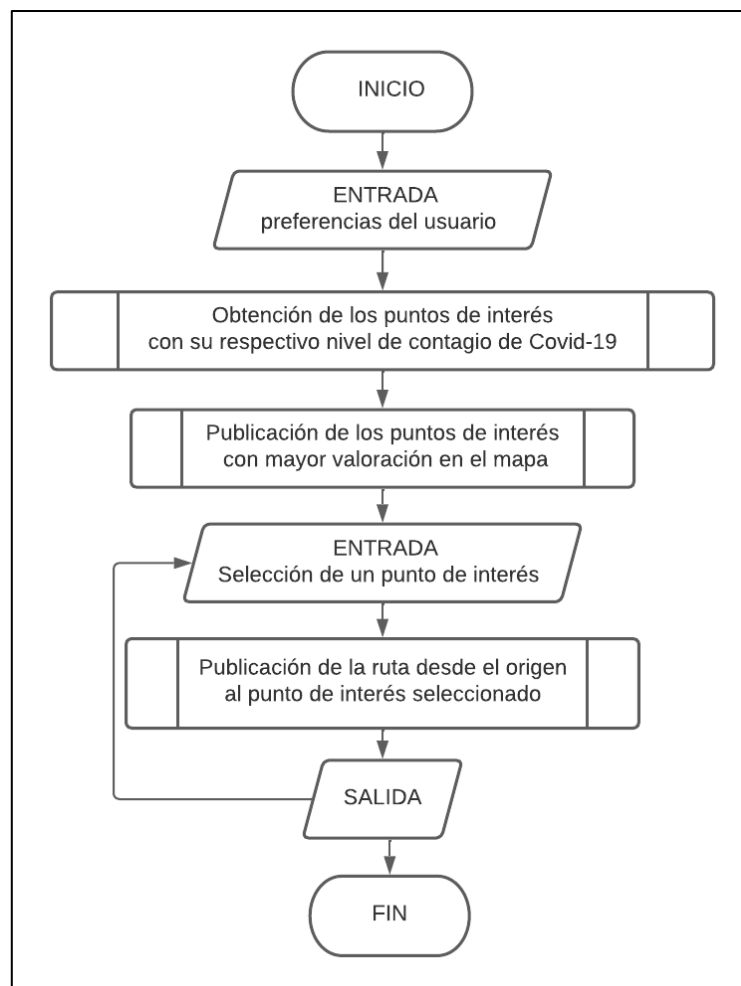


Figura 4. Diagrama de flujo del algoritmo

La Figura 5 representa la Arquitectura MVC (modelo vista controlador), donde el modelo corresponde a la base de datos en Firebase y la API de Google Maps, los sitios donde se obtiene la información de los POIs, por otro lado, el controlador representa a las

librerías WEB de java como Okhttp3 que permite realizar las peticiones al modelo. Por último, la vista es la Activity de la aplicación donde se despliega los resultados.

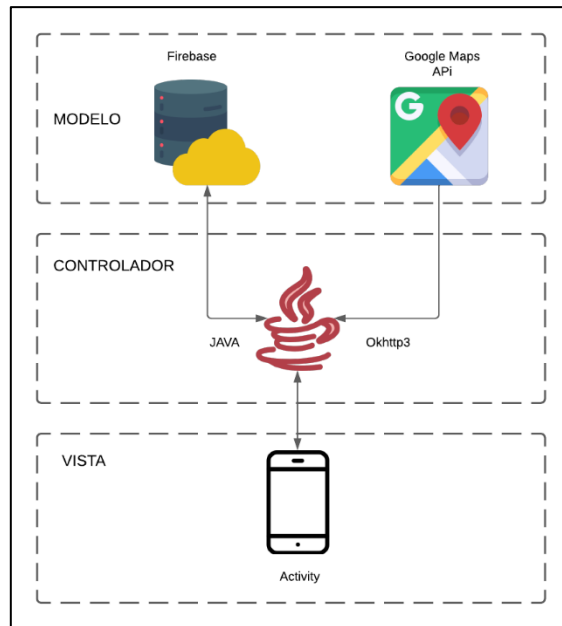


Figura 5. Arquitectura MVC del proyecto

2.1.2 Definición de roles

El presente proyecto está conformado de 2 integrantes, razón por la cual uno de los miembros de deberá cumplir con dos roles en el equipo Scrum, durante el desarrollo del proyecto como se muestra en la Tabla 2.

Tabla 2. Roles del equipo Scrum en el proyecto

Equipo Scrum	
Rol	Responsable
Product Owner	Maritzol Tenemaza
Scrum Master	Maritzol Tenemaza
Development Team	Patrick Cabezas

2.1.3 Definición de historias épicas

En la Tabla 3, se muestra las historias épicas identificadas para el desarrollo del proyecto. El detalle de cada una se puede visualizar en el Anexo I.

Tabla 3. Historias épicas

CÓDIGO	TÍTULO	PRIORIDAD
AH1	Identificar la cantidad de personas contagiadas de Covid-19 en los diferentes sectores de la ciudad.	ALTA
AH2	Integrar los datos del nivel de contagio de Covid-19 en cada sector, con los puntos de interés turístico de la ciudad.	ALTA
AH3	Desarrollo de la aplicación donde se despliegue el nivel de contagio de Covid-19 con los puntos de interés turísticos de cada sector en el mapa de la ciudad.	ALTA

2.1.4 Product Backlog

En la Tabla 4 se muestra el Product Backlog para el Sprint 0 a partir del desarrollo de las historias de usuario, en el Anexo II se muestran los detalles de cada historia de usuario.

Tabla 4. Product Backlog del Sprint 0

Product Backlog				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Esfuerzo	Prioridad
AH1	AH1-01	Identificar el número de personas contagiadas de Covid-19 en los diferentes sectores de la ciudad e integrar con los puntos de interés turístico.	8	Alta
	AH1-02	Cargar los datos del nivel de contagio de Covid-19 en los diferentes sectores de la ciudad con los puntos de interés turístico a la base de datos de Firebase.	3	Alta
AH2	AH2-01	Crear proyecto Android Studio e integrar con los servicios de Google Maps y Firebase.	5	Alta
	AH2-02	Diseño y desarrollo las interfaces de usuario.	8	Alta
AH3	AH3-02	Desarrollo de la lógica del formulario y el mapa de aplicación.	13	Alta

2.1.5 Planificación del Release

Después de definir las historias de usuario se ha decidido dividir el proyecto en 3 Sprints, donde cada Sprint tiene una duración de 14 días como se muestra en la Tabla 5.

Tabla 5. Planificación del Release

Release	Objetivo	Tiempo (Días)
Sprint 1	Identificar los datos sobre el número de personas contagiadas con Covid-19 en la ciudad de Quito, y crear los servicios necesarios para el funcionamiento del proyecto.	14
Sprint 2	Analizar y Construir el sistema de recomendaciones de sitios turísticos que integre información sobre el nivel de contagio de Covid-19 en áreas turísticas.	14
Sprint 3	Complementar los mapas en dos dimensiones para presentar la información del sistema de recomendaciones.	14

2.2 Sprint 1

2.2.1 Sprint Planning

Objetivo del Sprint

Identificar los datos sobre el número de personas contagiadas con Covid-19 en la ciudad de Quito, y crear los servicios necesarios para el funcionamiento del proyecto.

Historias de usuario

La Tabla 6 muestra las historias de usuario que se desarrollaran en el Sprint 1. En el Anexo II se encuentra el detalle de cada una.

Tabla 6. Historias de usuario del Sprint 1

ID	Título	Esfuerzo	Estado
AH1-01	Identificar el número de personas contagiadas de Covid-19 en los diferentes sectores de la ciudad e integrar con los puntos de interés turístico.	8	Por implementar
AH1-02	Cargar los datos del nivel de contagio de Covid-19 en los diferentes sectores de la ciudad con los puntos de interés turístico a la base de datos de Firebase.	3	Por implementar

Sprint Backlog

En la Tabla 7 se muestra el Sprint backlog con las tareas a realizar del Sprint 1.

Tabla 7. Sprint backlog del Sprint 1

Sprint backlog	
Historias de usuario	Tareas
AH1-01	Identificar las diferentes fuentes de información sobre el número de personas contagiadas con Covid-19 en cada sector de la ciudad.
	Recopilar los datos del número de personas contagiadas con Covid-19 en la ciudad de Quito.
	Reestructurar los datos del número de personas contagiadas con Covid-19 en cada sector de la ciudad acorde con el formato que va a ser utilizado en la aplicación.
AH1-02	Obtención de la API Key de los servicios de Google Maps Platform
	Habilitar las APIs de lugares y rutas de Google Maps Platform
	Crear el script de Python para estructurar los datos del nivel de contagio de Covid-19 en cada sector de la ciudad con los datos de los puntos de interés turístico.
	Generar la base de datos en tiempo real de Firebase
	Cargar los datos del nivel de contagio de Covid-19 en la ciudad con los datos de los puntos de interés turístico procesados a la base de datos de Firebase.

2.2.2 Ejecución del Sprint

Para la ejecución del Sprint 1 se detalla algunas de las tareas realizadas:

Identificación de los datos del número de personas contagiadas con Covid-19 en los diferentes sectores de la ciudad

Primero se identificó las fuentes de información oficiales sobre el número de contagios de Covid-19 en la ciudad de Quito, posteriormente se analizó los datos publicados, con lo que se decidió utilizar la información que proviene del Ministerio de salud pública del Ecuador como se muestra en la Figura 6, lo principal para que los datos puedan ser utilizados es que muestre la cantidad de personas contagiadas en cada sector de la ciudad.

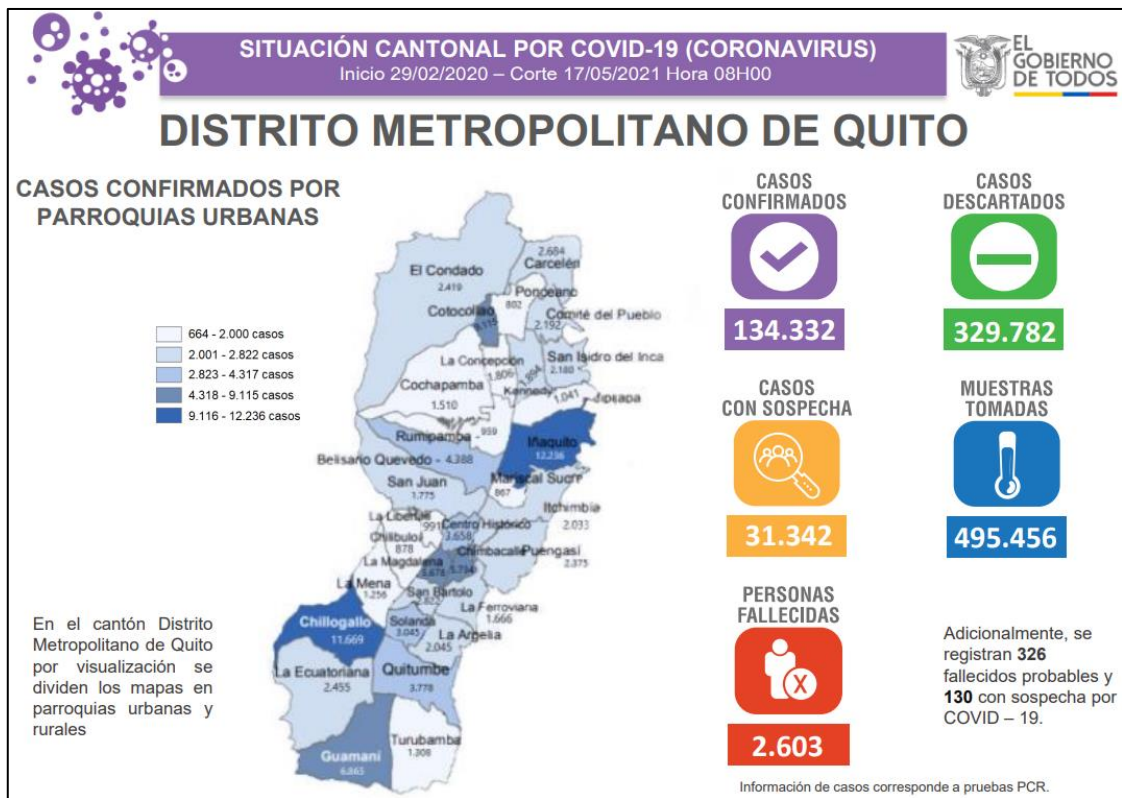


Figura 6. Cantidad de contagiados de Covid-19 en Quito por sector

Después se analizó los datos que se obtuvo de la fuente del Ministerio de Salud Pública del Ecuador [3], esta fuente no tiene una API o un servicio que permita extraer los datos de manera automática, por este motivo se extrajo los datos de manera manual, indicando el sector de la ciudad y la cantidad de casos confirmados de contagio de Covid-19 en una tabla, como se muestra gráficamente en la Figura 7.

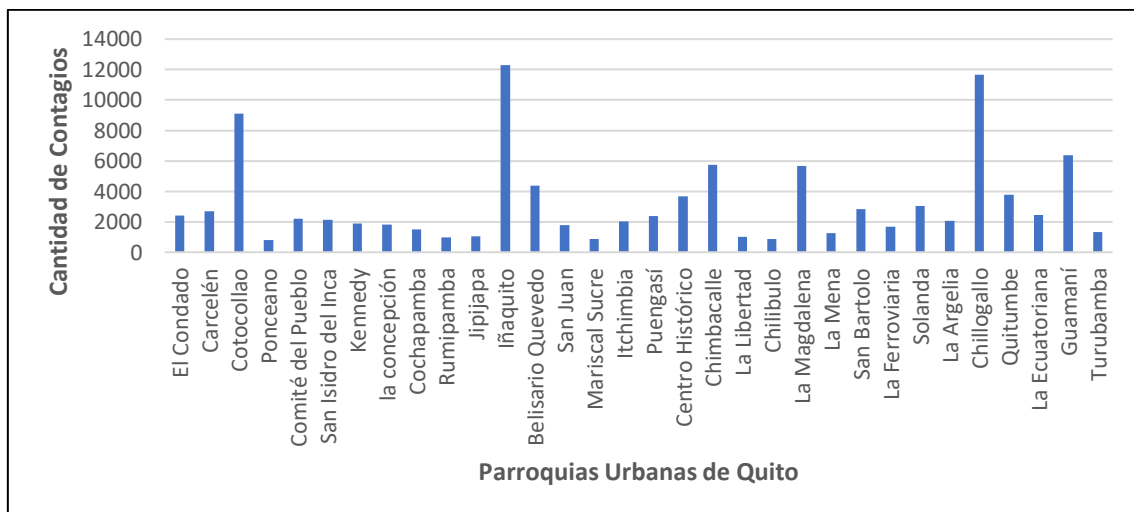


Figura 7. Cantidad de contagiados de Covid-19 por cada sector

A continuación, tomando como referencia los datos del Ministerio de salud pública [3] se optó por clasificar de la siguiente manera el nivel de contagio de cada sector, donde en los sectores que tienen menos de 2000 casos confirmados como nivel bajo, de 2001 a 4317 se considera nivel medio y los sectores con más de 4318 casos tiene un nivel alto de contagio, como se muestra en la Tabla 8.

Tabla 8. Niveles de contagio según la cantidad de casos confirmados.

Nivel de Contagio	Rango
Bajo	<2000
Medio	2001-4317
Alto	>4318

Además, se analizaron los datos del Centro de Modelización Matemática en Áreas Clave para el Desarrollo (ModeMat), que se encarga de desarrollar técnicas matemáticas y computacionales, para construir modelos matemáticos con el fin de resolver problemas [4]. La organización muestra un dato global de la ciudad de Quito como se muestra en la Figura 8, por el contrario, como se observaron los datos del Ministerio de Salud permiten observar los sectores con mayor riesgo de contaminación del Covid-19, estos datos son de mayor utilidad para el proyecto actual.

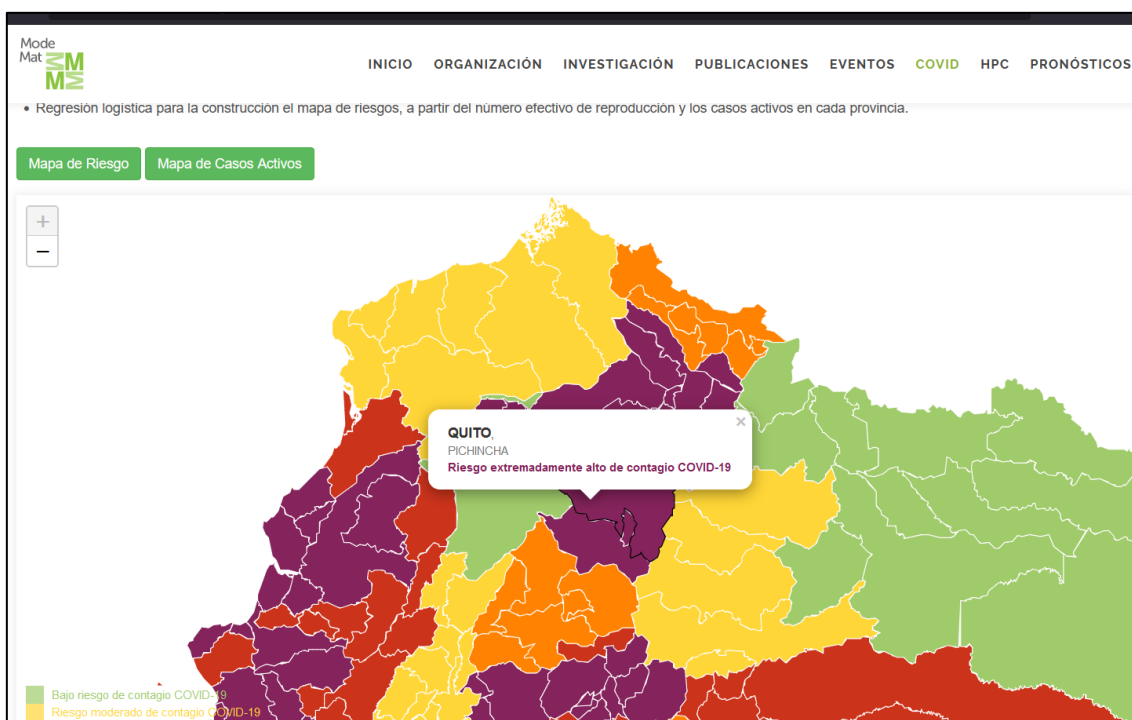


Figura 8. Cantidad de contagiados de Covid-19 en Quito de la fuente ModeMat

Google Maps API

Primero para utilizar el mapa de Google Maps fue necesario obtener el API KEY de Google Maps Platform para poder utilizar los servicios que ofrece, la API KEY se muestra en la Figura 9.



Figura 9 Google API Key

Posteriormente se habilitó las APIs necesarias para el funcionamiento de la aplicación, los cuales son los (API de lugares, y API de rutas) como se muestra en la Figura 10.

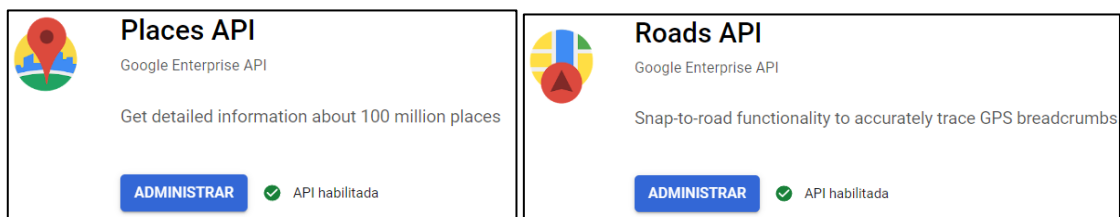


Figura 10. APIs de Lugares y Rutas

Para obtener los puntos de interés turístico (POIs) primero se seleccionó los tipos más interesantes, los cuales se muestran en la siguiente lista:

- Museos
- Parques
- Templos
- Estadios
- Iglesias
- Zoológicos
- Cementerios
- Bar
- Café

Script Python

A continuación, se desarrolló un script para poder asignar el nivel de contagio de cada sector, con cada punto de interés turístico. Para esto fue necesario obtener los puntos de interés de cada sector, luego con la información de cada punto de interés se los clasifica a los que tengan una valoración de los usuarios mayor o igual a cuatro con el

Posteriormente se carga a la base de datos de Firebase, para esto se utiliza el archivo Json que fue generado previamente, como se muestra en la Figura 14.

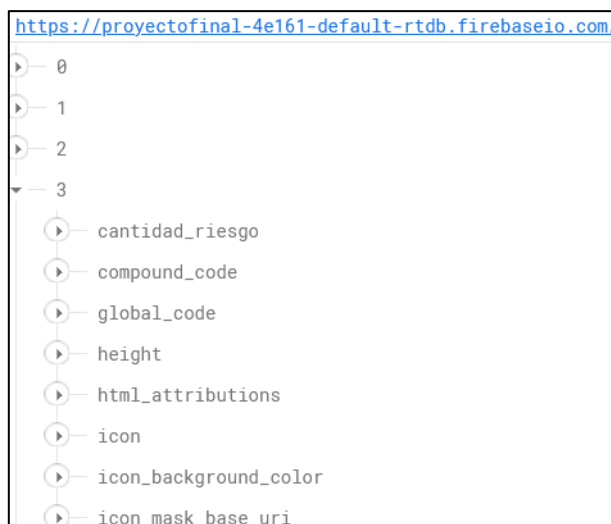


Figura 14. Base de datos en Firebase

2.2.3 Sprint Review

Todos los objetivos del Sprint 1 se cumplieron en su totalidad.

Pruebas de aceptación

En la Tabla 9 se muestra los criterios de aceptación para las historias de usuario del presente Sprint donde se cumplieron todos los criterios de aceptación.

Tabla 9. Pruebas de aceptación del Sprint 1

Historia de Usuario	Dado que	Cuando	Entonces	Cumplido
AH1-01	Se requiere conocer la información de los contagios en los sectores de la ciudad.	Se extrae los datos contagios de Covid-19 de las diferentes fuentes de información.	Se obtiene la información del número de personas contagiadas con Covid-19 por cada sector.	Sí
AH1-02	El usuario requiere identificar los puntos de interés	El usuario solicita un tipo de punto de interés.	Se obtiene como resultado los puntos de interés con el número de contagios de Covid-19 en el sector.	Sí

Adaptación del Product Backlog

Cumplido el tiempo de ejecución del Sprint 1 se analizó los avances y se actualizó la Tabla 10 del Product Backlog, los avances realizados muestran que los datos ya están disponibles para ser usados en futuros Sprints.

Tabla 10. Product Backlog adaptado del Sprint 1

Product Backlog				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Esfuerzo	Estado
AH1	AH1-01	Identificar el número de personas contagiadas de Covid-19 en los diferentes sectores de la ciudad e integrar con los puntos de interés turístico.	8	Terminado
	AH1-02	Cargar los datos del nivel de contagio de Covid-19 en los diferentes sectores de la ciudad con los puntos de interés turístico a la base de datos de Firebase.	3	Terminado
AH2	AH2-01	Crear proyecto Android Studio e integrar con los servicios de Google Maps y Firebase.	5	Por implementar
	AH2-02	Diseño y desarrollo las interfaces de usuario.	8	Por implementar
AH3	AH3-01	Desarrollo de la lógica del formulario y el mapa de aplicación.	13	Por implementar

2.2.4 Medición de Velocidad de Avance del Proyecto

En la siguiente Figura 15 se muestra la gráfica del Burndown donde se muestra el progreso del primer Sprint, en el cual se muestra que los avances en las tareas se realizaron en base a lo pronosticado.

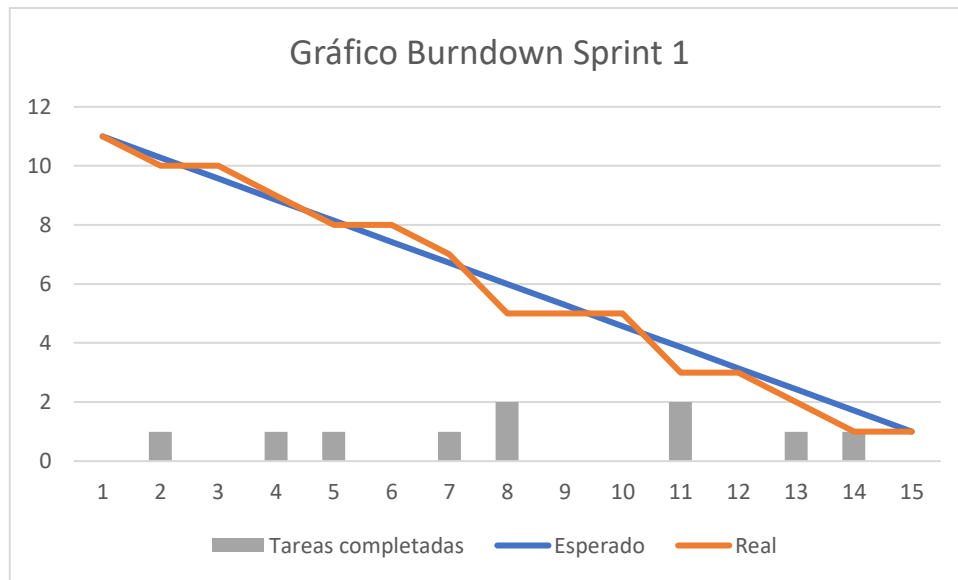


Figura 15. Gráfico de Burndown del Sprint 1

2.3 Sprint 2

2.3.1 Sprint Planning

Objetivo del Sprint

Analizar y Construir el sistema de recomendaciones de sitios turísticos que integre información sobre el nivel de contagio de Covid-19 en áreas turísticas.

Historias de Usuario

Historias de usuario del Sprint 2, el detalle de cada historia se encuentra en el Anexo II.

Tabla 11. Historias de usuario del Sprint 2

ID	Título	Esfuerzo	Estado
AH2-01	Crear proyecto Android Studio e integrar con los servicios de Google Maps y Firebase.	5	Por implementar
AH2-02	Diseñar y desarrollar las interfaces de usuario.	8	Por implementar

Sprint Backlog

Tabla 12. Sprint backlog del Sprint 2

Sprint backlog	
Historias de usuario	Tareas
AH2-01	Crear proyecto en Android Studio
	Asignar las credenciales de Google Maps Platform en el proyecto de Android Studio
	Integrar la base de datos de Firebase en Android Studio
	Crear el repositorio en Github.
	Integrar el proyecto de Android Studio al repositorio en Github
AH2-02	Crear los Mockups de las interfaces de usuario
	Desarrollar las interfaces de login, registro y menú principal en Android Studio

2.3.2 Ejecución del Sprint

Durante el desarrollo del presente Sprint se realizaron las siguientes actividades.

Android Studio

Se creó el proyecto de en Android Studio que es el ID que va a ser utilizado para el desarrollo de la aplicación, la versión con la que se va a trabajar es la 11.0.12 como se muestra en la Figura 16.

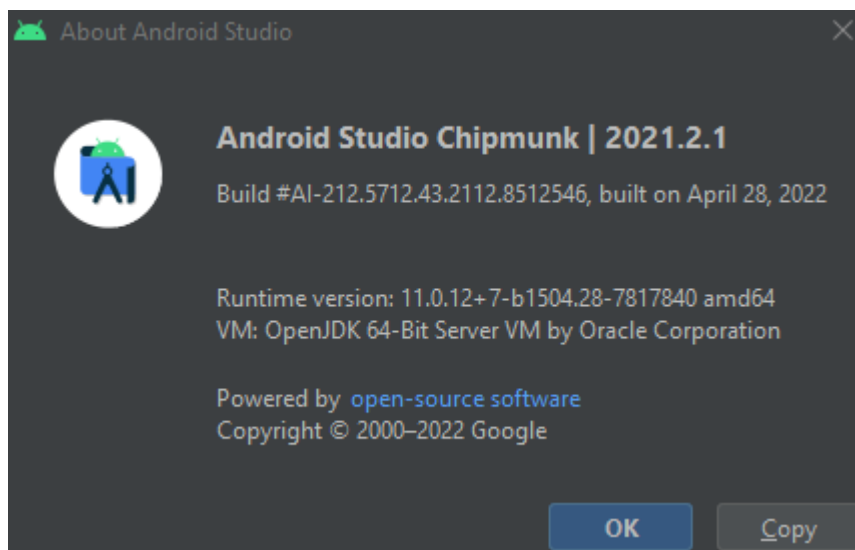


Figura 16. Versión de Android Studio

El SDK que va a ser utilizado va a ser la API 23 con soporte mínimo para Android 6 como se muestra en la Figura 17, y el lenguaje de programación con el que se va a desarrollar el proyecto es Kotlin.

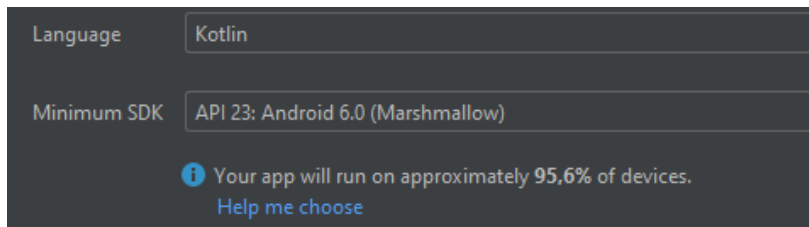


Figura 17. Versión de la API utilizada

Después se agrega la clave de la API de Google Maps al proyecto en Android Studio como se muestra en la Figura 18, para que funcionen los servicios como el mapa y las rutas en la aplicación.

```
<meta-data
  android:name="com.google.android.geo.API_KEY"
  android:value="AIzaSyDvn6DilnJELEJrcDURg49w0TM_9nUE_FI" />

<activity
  android:name=".MapsActivity"
  android:exported="true"
  android:label="@string/title_activity_maps">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
```

Figura 18. Asignación de la API en Android Studio

Por último, se integran al proyecto de Android Studio con la base de datos de Firebase, para que la aplicación pueda utilizar los datos que ya fueron cargados en el Sprint 1.

Github

A continuación, se inicializó un proyecto en la página WEB de Github para poder hacer el control de versiones de los cambios del proyecto, posteriormente se agregó la dirección que se muestra en la Figura 19 a Android Studio, de esta forma se tiene control directamente en el ID.

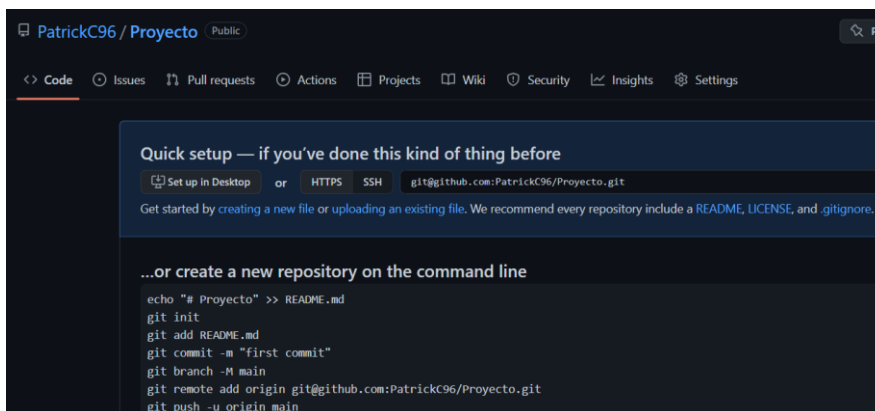


Figura 19. Proyecto en GitHub

Interfaces de usuario

Después de la reunión con los miembros del equipo se realizó el maquetado de las interfaces en la herramienta Figma y el resultado de estas se puede ver en las Figura 20 y 21, donde se muestra las pantallas de Registro, Bienvenida, Menú, y el Mapa.



Figura 20. Maqueta de las pestañas de Registro, Logín y Menú

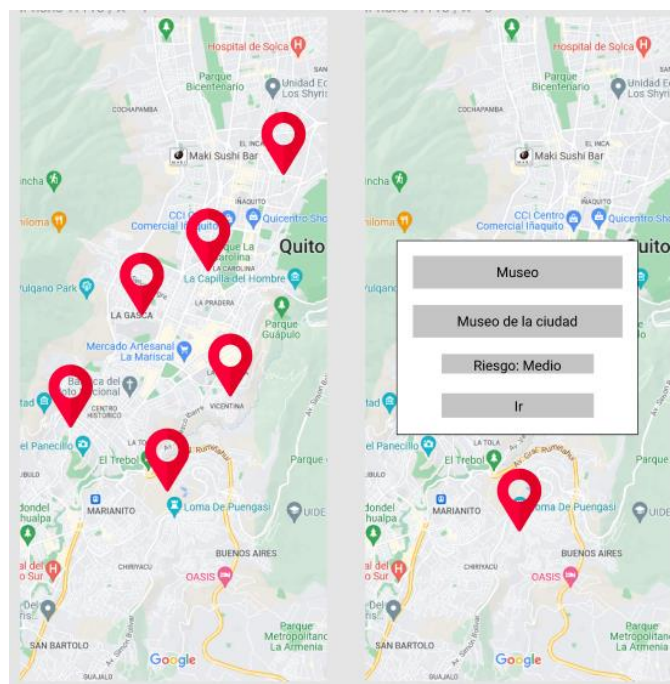


Figura 21. Maquetas del mapa con los POIs

Por último, en el ID de desarrollo Android Studio se realizó la programación de las interfaces de usuario como logín, registro, y menú, como se muestra en la Figura 22.



Figura 22. Interfaces de usuario creadas en Android Studio

2.3.3 Sprint Review

Pruebas de aceptación

Después de revisar las actividades realizadas en el Sprint 2 se puede verificar el cumplimiento de los criterios de aceptación de las historias de usuario en su totalidad como se muestra en la Tabla 13.

Tabla 13. Pruebas de aceptación del Sprint 2

Historia de Usuario	Dado que	Cuando	Entonces	Cumplido
AH2-01	Se configura Android Studio para el desarrollo del proyecto	El usuario utiliza un dispositivo Android.	La aplicación funcionará en dispositivos Android	SÍ
AH2-02	El usuario acceda a la pantalla	cuando el usuario desee realizar una acción en la aplicación	Se mostrará la pantalla correspondiente en base a los datos ingresados.	SÍ

Adaptación del Product Backlog

Todas las tareas de las historias fueron culminadas por ende se marca en el Product Backlog los cambios realizados como se muestra en la Tabla 14.

Tabla 14. Product Backlog adaptado del Sprint 2

Product Backlog				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Esfuerzo	Estado
AH1	AH1-01	Identificar el número de personas contagiadas de Covid-19 en los diferentes sectores de la ciudad e integrar con los puntos de interés turístico.	8	Terminado
	AH1-02	Cargar los datos del nivel de contagio de Covid-19 en los diferentes sectores de la ciudad con los puntos de interés turístico a la base de datos de Firebase.	3	Terminado
AH2	AH2-01	Crear proyecto Android Studio e integrar con los servicios de Google Maps y Firebase.	5	Terminado
	AH2-02	Diseño y desarrollo las interfaces de usuario.	8	Terminado
AH3	AH3-01	Desarrollo de la lógica del formulario y el mapa de aplicación.	13	Por implementar

2.3.4 Medición de Velocidad de Avance del Proyecto

En la Figura 23 se muestra el Grafico Burndown del Sprint 2, donde se muestra que las tareas fueron culminadas dentro del tiempo establecido.

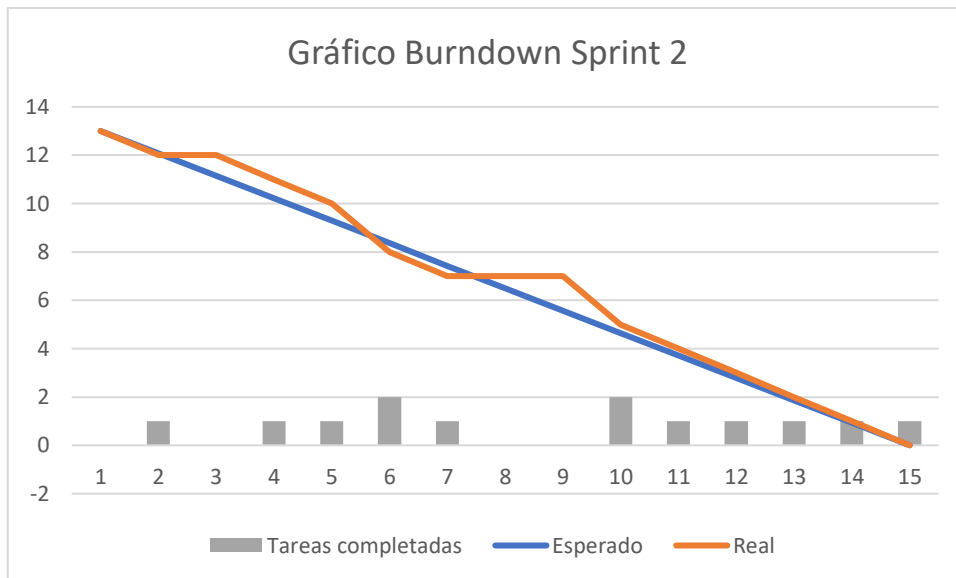


Figura 23. Gráfico de Burndown del Sprint 2

2.4 Sprint 3

2.4.1 Sprint Planning.

Objetivo del Sprint.

Complementar los mapas en dos dimensiones para presentar la información del sistema de recomendaciones.

Historias de Usuario

En la Tabla 15 se muestra la historia de usuario planificada para el desarrollo en el Sprint 3.

Tabla 15. Historias de usuario del Sprint 3

ID	Título	Esfuerzo	Estado
AH3-01	Desarrollo de la lógica del formulario y el mapa de aplicación.	13	Por implementar

Sprint Backlog

Tabla 16. Sprint backlog del Sprint 3

SPRINT BACKLOG	
HISTORIA DE USUARIO	TAREAS
AH3-01	Desarrollar la lógica del logín y registro de usuarios con Firebase

	Desarrollar la lógica del menú que exija al usuario seleccionar al menos una opción
	Desarrollar la lógica del mapa para que se muestren las recomendaciones de los puntos de interés en la zona respectiva.
	Desarrollar el algoritmo para crear las rutas a los puntos de interés en el mapa.

2.4.2 Ejecución del Sprint.

Lógica de la aplicación

La Figura 24 muestra el código desarrollado para el funcionamiento de la función de logín del usuario, se implementó la función que ofrece Firebase de validar si un usuario esta registrado y si las credenciales son correctas. Para esto en el botón **ACCEDER** se genera la acción para que el algoritmo tome los datos del nombre de usuario y contraseña, y se envía a Firebase con la función de **signInWithEmailAndPassword**, que retorna verdadero si las credenciales son correctas de lo contrario retorna falso, si la autenticación es correcta se ingresa al menú de la aplicación.

```

val email=emailEditText?.text.toString()
val password=passEditText?.text.toString()
if (!TextUtils.isEmpty(email) && !TextUtils.isEmpty(password)) {
    auth.signInWithEmailAndPassword(email,password).addOnCompleteListener { task ->
        if(task.isSuccessful){
            val intent= Intent( packageContext: this,MainActivity::class.java)
            startActivity(intent)
            Log.d(TAG, msg: "acceso exitoso")
            finish()
        }
    }.addOnFailureListener { exception ->
        Toast.makeText(applicationContext,exception.localizedMessage, Toast.LENGTH_LONG).show()
        Log.d(TAG, msg: "acceso erroneo")
    }
}else {
    Toast.makeText( context: this, text: "Ingrese Los Datos Correctos", Toast.LENGTH_SHORT).show()
}

```

Figura 24. Código fuente de autenticación de los usuarios

A continuación, en la Figura 25 se muestra código fuente para registrar a los usuarios en la aplicación, para esto el algoritmo solicita el correo electrónico que va a ser usado como nombre de usuario, la contraseña y la confirmación de la contraseña, si esto datos son iguales se procede a enviar los datos con la función **createUserWithEmailAndPassword**, que retorna verdadero si los datos son válidos y se abre la pantalla de logín.

```

fun register(){
    val email= emailEditText?.text.toString()
    val password= passEditText?.text.toString()
    val password2= passEditText2?.text.toString()
    if (!TextUtils.isEmpty(email) && !TextUtils.isEmpty(password) && !TextUtils.isEmpty(password2)) {
        if(password.equals(password2)){
            auth.createUserWithEmailAndPassword(email,password).addOnCompleteListener { task ->
                if(task.isSuccessful){
                    val intent= Intent( packageContext: this,AuthActivity::class.java)
                    startActivity(intent)
                    Log.d(TAG, msg: "Registro Exitoso")
                    Toast.makeText( context: this, text: "Registro Exitoso", Toast.LENGTH_SHORT).show()
                    finish()
                }
            }.addOnFailureListener { exception ->
                Toast.makeText(applicationContext,exception.localizedMessage,Toast.LENGTH_LONG).show()
                Log.d(TAG, msg: "registro erronea")
            }
        }
    }
}

```

Figura 25. Código fuente de registro de usuarios

Luego, en la Figura 26 se muestra el código fuente que obtiene los datos del menú donde el usuario selecciona que tipo de puntos de interés desea visitar y sector de la ciudad, este algoritmo verifica que el usuario marque al menos un tipo de punto de interés y un sector, después abre y envía los datos a la pantalla del mapa.

```

if(verificarParametros()){
    Toast.makeText( context: this, text: " Ir al MAPA ",Toast.LENGTH_SHORT).show()
    obtenerUbicacion()
    val intentExplicito = Intent( packageContext: this, MapsActivity::class.java)
    val lista = obtenerDatos()
    for (posicion in lista.indices){
        intentExplicito.putExtra(posicion.toString(),lista.get(posicion).toString())
    }
    intentExplicito.putExtra( name: "ubicacion",this.selccionUbicacion.toString())
    this.startActivity(intentExplicito)
}

```

Figura 26. Código fuente para cargar datos en el mapa

A continuación, en la lógica del mapa que recibe los datos de las preferencias del usuario, procede a realizar la selección de los puntos de interés obteniendo la información de la base de datos en Firebase y posterior mente presenta cada punto de interés en forma de marcadores en mapa, el código fuente se muestra en la Figura 27.

```

database.child(valor.toString()).addListenerForSingleValueEvent(object : ValueEventListener {
    override fun onDataChange(snapshot: DataSnapshot) {
        if (snapshot.exists()){
            for (elemento in elementos) {
                if (!elemento.equals("")){
                    if (snapshot.child( path: "type_place").value?.equals(elemento) == true){
                        if (snapshot.child( path: "ubicacion_punto").value?.equals(LugarUbicacion.toString()) == true){
                            Log.i( tag: "datos", msg: "el dato es: "+snapshot.child( path: "name").value)
                            longitud = snapshot.child( path: "Lng").value.toString()
                            latitud = snapshot.child( path: "Lat").value.toString()
                            nombre = snapshot.child( path: "name").value.toString()
                            riesgo = snapshot.child( path: "nivel_riesgo").value.toString()
                            Log.i( tag: "datos", msg: "el dato es: "+longitud+" : "+latitud)
                            agregar_marcadores(googleMap,LatLng(latitud.toFloat(), longitud.toFloat()), nombre, riesgo, elemento)
                        }
                    }
                }
            }
        }
    }
})

```

Figura 27. Lógica para cargar los marcadores de los POIs

Además, se procede a realizar la lógica para marcar la ruta desde la posición del usuario hasta el punto de interés, como se muestra en la Figura 28, esto se lo realiza utilizando el servicio de rutas Google Platforms, para lo cual el algoritmo debe procesar la respuesta en formato Json, y obtener la información de cada punto por su longitud y latitud para marcar una línea a siguiente punto hasta llegar al destino.

```
val poly = ArrayList<LatLng>()
var index = 0
val len = encoded.length
var lat = 0
var lng = 0
while (index < len) {
    var b: Int
    var shift = 0
    var result = 0
    do {
        b = encoded[index++].code - 63
        result = result or (b and 0x1f shl shift)
        shift += 5
    } while (b >= 0x20)
    val dlat = if (result and 1 != 0) (result shr 1).inv() else result shr 1
    lat += dlat
    shift = 0
    result = 0
    do {
        b = encoded[index++].code - 63
        result = result or (b and 0x1f shl shift)
        shift += 5
    } while (b >= 0x20)
    val dlng = if (result and 1 != 0) (result shr 1).inv() else result shr 1
    lng += dlng
    val latLng = LatLng((lat.toDouble() / 1E5), (lng.toDouble() / 1E5))
    poly.add(latLng)
}
```

Figura 28. Lógica para crear las rutas

También, se implementó la funcionalidad que permite a los usuarios obtener información detallada sobre los puntos de interés turístico que les interesan. Ahora, cuando un usuario selecciona un icono en el mapa, este cambia de color para indicar que ha sido seleccionado como se muestra en la Figura 29, y se despliega la ventana de información. Con esta funcionalidad se cambio el color del icono seleccionado, permitiendo que la experiencia de usuario sea más intuitiva y fácil de identificar sobre que icono se seleccionó, de esta manera los usuarios no se van a confundir cuando el mapa está lleno de iconos alrededor, por último, el icono regresa a su color original cuando el usuario sale de la venta flotante.

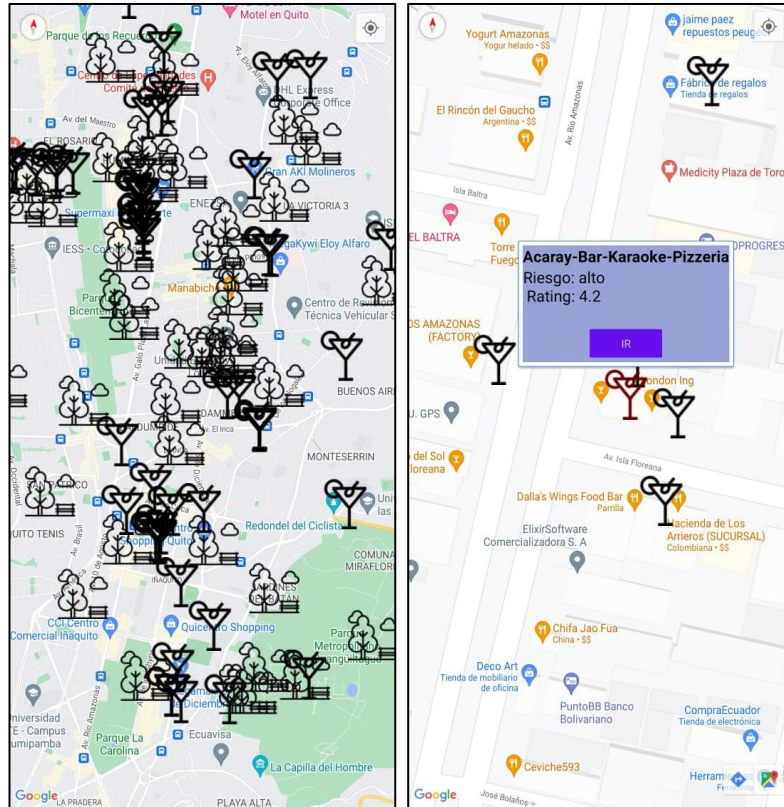


Figura 29. Cambio de color del icono seleccionado

Por último, la aplicación fue probada en varios dispositivos Android tanto como físicos como emuladores durante el desarrollo del proyecto, en la Figura 30 y 31 se muestra la información de los dispositivos físicos con Android 11 y la Figura 32 es un dispositivo emulador con Android 9.

Dispositivo físico con versión de Android 11.

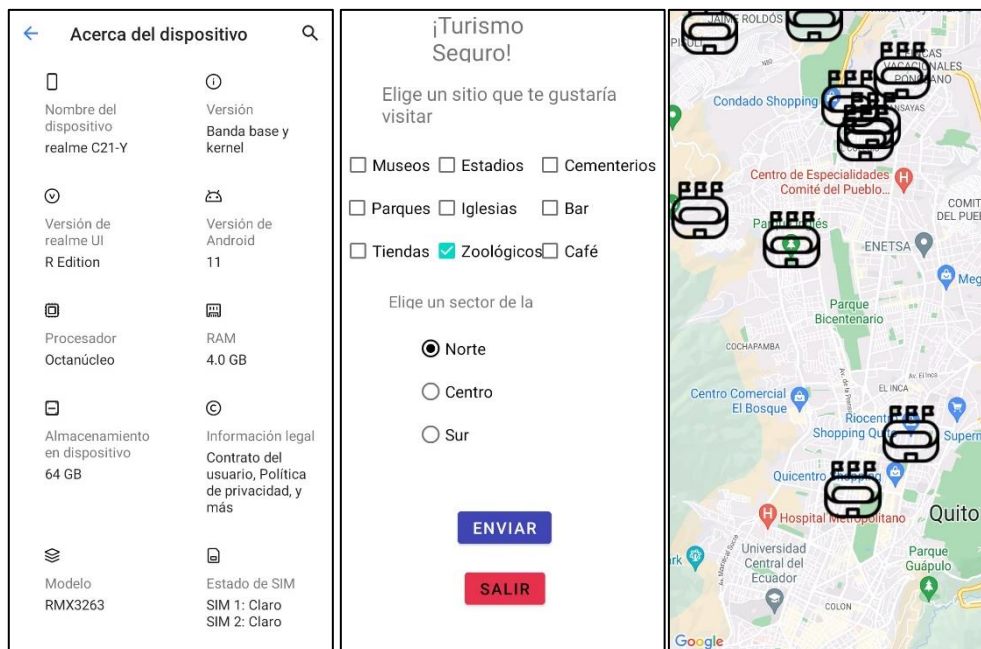


Figura 30. Dispositivo con Android 11.

Dispositivo físico con versión de Android 11.

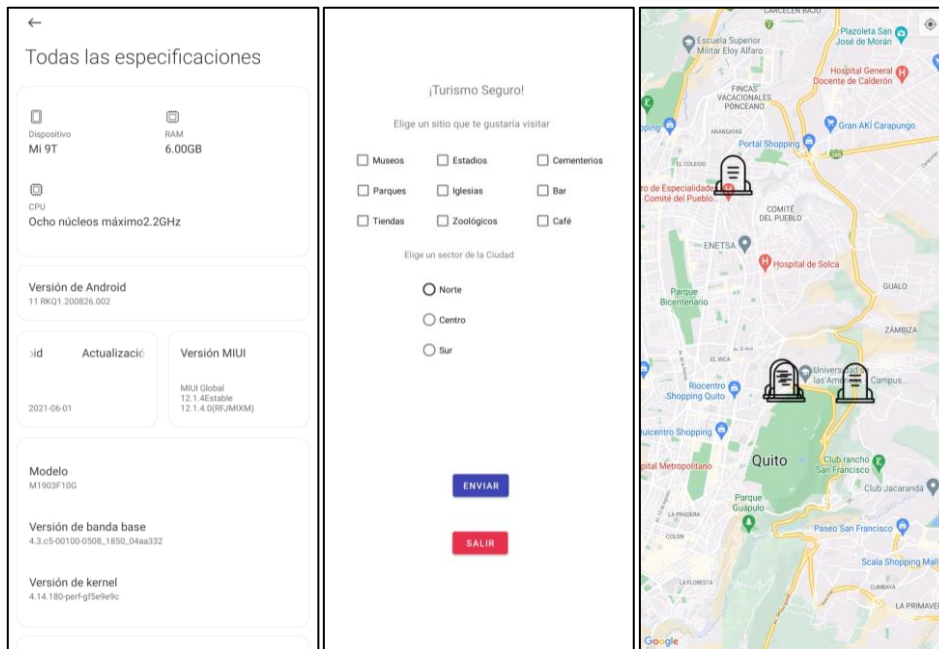


Figura 31. Dispositivo con Android 11.

Emulador con versión de Android 9.

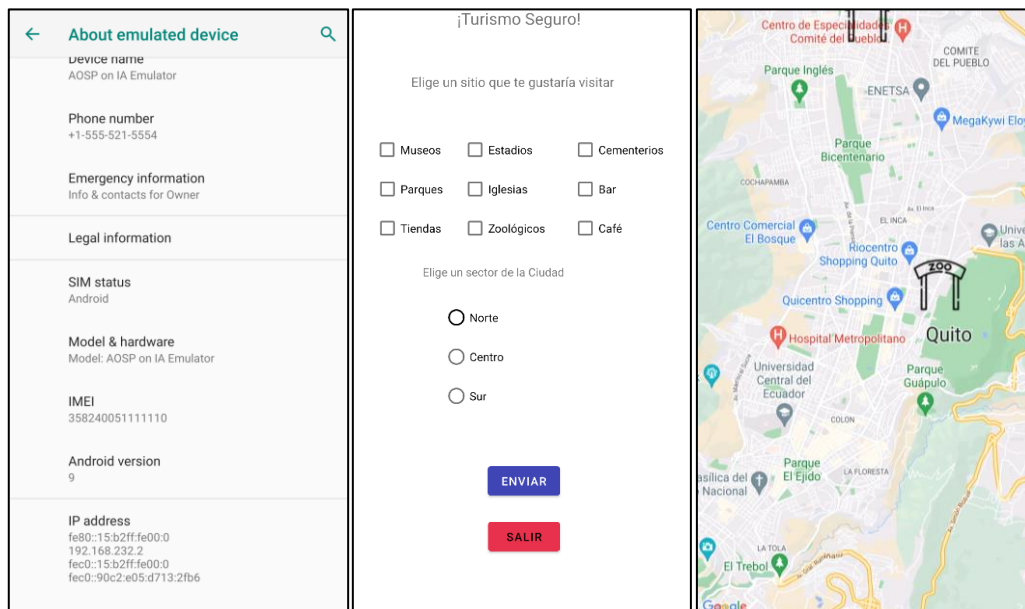


Figura 32. Emulador con Android 9.

2.4.3 Sprint Review.

Pruebas de aceptación

En el Sprint 3 se completaron todas las tareas en el tiempo acordado.

Tabla 17. Pruebas de aceptación del Sprint 3

Historia de Usuario	Dado que	Cuando	Entonces	Cumplido
AH3-01	El usuario accede a la pantalla de inicio	El usuario inicia la aplicación	Se despliega la pantalla donde debe ingresar el usuario y contraseña.	SÍ
	El usuario accede a la pantalla de registro	El usuario no se ha registrado en la aplicación	Se despliega la pantalla donde el usuario ingresa la información para registrarse en la aplicación	SÍ
	El usuario accede a la pantalla del menú.	El usuario allí iniciado de manera correcta	Se muestra las opciones que el usuario debe elegir para buscar los puntos de interés.	SÍ
	El usuario accede a la pantalla del mapa	El usuario selecciona las opciones de los puntos de interés que fueron requeridos.	Se despliega los puntos de interés en base a los requerimientos del usuario.	SÍ
	El usuario seleccionó un punto de interés (POI)	El usuario selecciona en ir al punto de interés	Se muestra la ruta para llegar al punto de interés desde el punto donde se encuentra.	SÍ

Adaptación del Product Backlog

Todas las tareas de las historias fueron culminadas por ende se marca en el Product Backlog los cambios realizados como se muestra en la Tabla 18.

Tabla 18. Product Backlog adaptado del Sprint 3

Product Backlog				
HISTORIA ÉPICA	HISTORIA DE USUARIO			
	ID	Título	Esfuerzo	Estado
AH1	AH1-01	Identificar el número de personas contagiadas de Covid-19 en los diferentes sectores de la ciudad e integrar con los puntos de interés turístico.	8	Terminado
	AH1-02	Cargar los datos del nivel de contagio de Covid-19 en los diferentes sectores de la ciudad con los puntos de interés turístico a la base de datos de Firebase.	3	Terminado
AH2	AH2-01	Crear proyecto Android Studio e integrar con los servicios de Google Maps y Firebase.	5	Terminado
	AH2-02	Diseño y desarrollo las interfaces de usuario.	8	Terminado
AH3	AH3-01	Desarrollo de la lógica del formulario y el mapa de aplicación.	13	Terminado

2.4.4 Medición de Velocidad de Avance del Proyecto

En la última reunión del Sprint 3 se concluyó que todas las tareas fueron culminadas en el tiempo estimado como se muestra en la Figura 33.

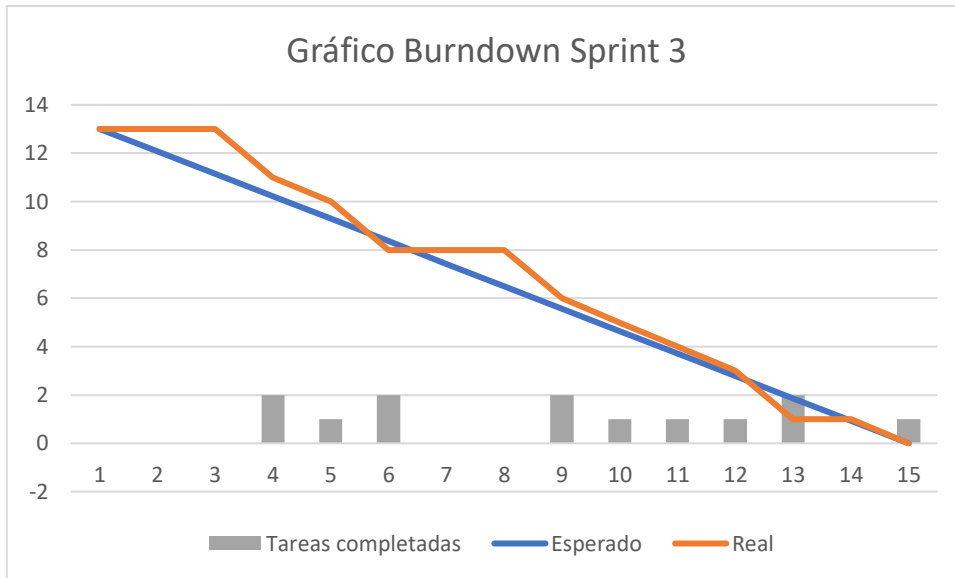


Figura 33. Gráfico de Burndown del Sprint 3

3 RESULTADOS Y DISCUSIÓN

3.1 Producto Final

El Producto final del actual proyecto está constituido por tres partes: la primera parte consiste en la toma de datos del usuario, la segunda parte donde la aplicación selecciona los establecimientos que cumpla los requerimientos del usuario y retorna los puntos de interés turísticos, y la última que despliega las rutas para llegar a cada punto de interés.

3.1.1 Toma de Datos

En la pantalla de toma de datos el usuario puede seleccionar al menos un tipo de punto de interés, también seleccionar el sector que se desee explorar, como se muestra en la Figura 34.

Sure Tourist Trip!
Sitios que te gustaría visitar

<input type="checkbox"/> Museos	<input checked="" type="checkbox"/> Estadios	<input type="checkbox"/> Cementerios
<input type="checkbox"/> Parques	<input type="checkbox"/> Iglesias	<input type="checkbox"/> Bar
<input type="checkbox"/> Tiendas	<input type="checkbox"/> Zoológicos	<input type="checkbox"/> Café

Sector de la Ciudad

Norte
 Centro
 Sur

ENVIAR

SALIR

Figura 34. Menú de tomas de datos

3.1.2 Despliegue de los resultados

La pantalla de despliegue de resultados consiste en mostrar el mapa con los puntos de interés que estén disponibles en el sector que fue seleccionado por el usuario, al marcar sobre cada punto de interés mostrar un mensaje indicando el nivel de contagio de ese sitio, como se muestra en la Figura 35.

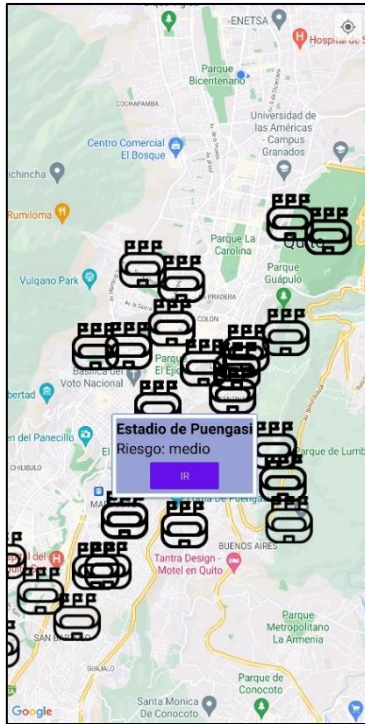


Figura 35. Ejecución del mapa con los POIs y las rutas

3.1.3 Despliegue de las Rutas

Por último, en el despliegue de las rutas, el usuario puede agregar más puntos de interés con su respectiva ruta después de crear la ruta a cada punto de interés, como se muestra en la Figura 36.

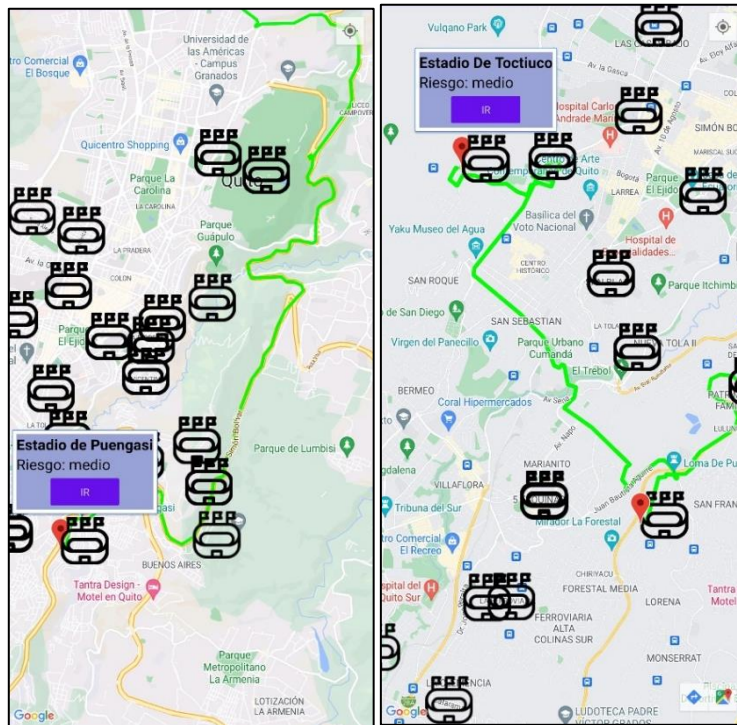


Figura 36. Rutas entre los puntos

3.2 Pruebas

Se realizó pruebas del producto final con 20 personas, donde se evaluó la usabilidad percibida del producto usando el test de usabilidad SUS, de las siglas en inglés (System Usability Scale: Sistema de Escala de usabilidad) es una herramienta metodológica que permite medir la usabilidad de dispositivos y aplicaciones, es un método rápido, pero no ofrece resultados totalmente limpios, el test consiste en 5 preguntas positivas 5 preguntas negativas, que se utilizan para calcular los resultados [19].

3.2.1 Usabilidad percibida

La prueba consiste en 10 preguntas con 5 opciones de respuesta como se muestra a continuación.

Lista de preguntas:

1. Creo que me gustaría utilizar este sistema frecuentemente.
2. El sistema me resultó innecesariamente complejo.
3. Creo que el sistema es bastante fácil de utilizar.
4. Creo que necesitaría el soporte de un técnico para poder utilizar este sistema.
5. Creo que las diferentes funciones del sistema se encuentran muy bien integradas.
6. Opino que hubo demasiada inconsistencia en el sistema.
7. Imagino que la mayoría de las personas aprendería a utilizar el sistema rápidamente.
8. Me sentí algo incómodo al utilizar este sistema.
9. Me sentí muy seguro al utilizar este sistema.
10. Necesito aprender muchas otras cosas antes de poder utilizar correctamente el sistema.

Se usa la escala de Likert, donde los extremos 1 y 2 así como 4 y 5, indica el valor de la respuesta:

- Totalmente en desacuerdo: 1 punto
- En desacuerdo: 2 puntos
- Neutral: 3 puntos
- De acuerdo: 4 puntos
- Totalmente de acuerdo: 5 puntos

A cada respuesta se le asigna un valor, en este caso Totalmente en desacuerdo tendrá un valor de 1 y la respuesta Totalmente de acuerdo tendrá un valor de 5 puntos, al concluir la prueba SUS se procede a calcular el valor promedio de cada pregunta, luego a las preguntas impares se resta un punto y las preguntas pares se calcula con la resta

de cinco y el valor promedio obtenido en cada pregunta, con eso se obtuvo los valores de la Tabla 19.

Tabla 19. Resultado de la prueba de usabilidad SUS.

Pregunta	1	2	3	4	5	6	7	8	9	10
Promedio	4,2	1,8	4,3	1,8	4,2	1,9	4,4	1,5	4,2	1,7
Calificación	3,2	3,3	3,3	3,2	3,2	3,2	3,4	3,5	3,2	3,3

Con esto datos se realizó el siguiente Figura 37 que indica la variación de los valores de cada pregunta.

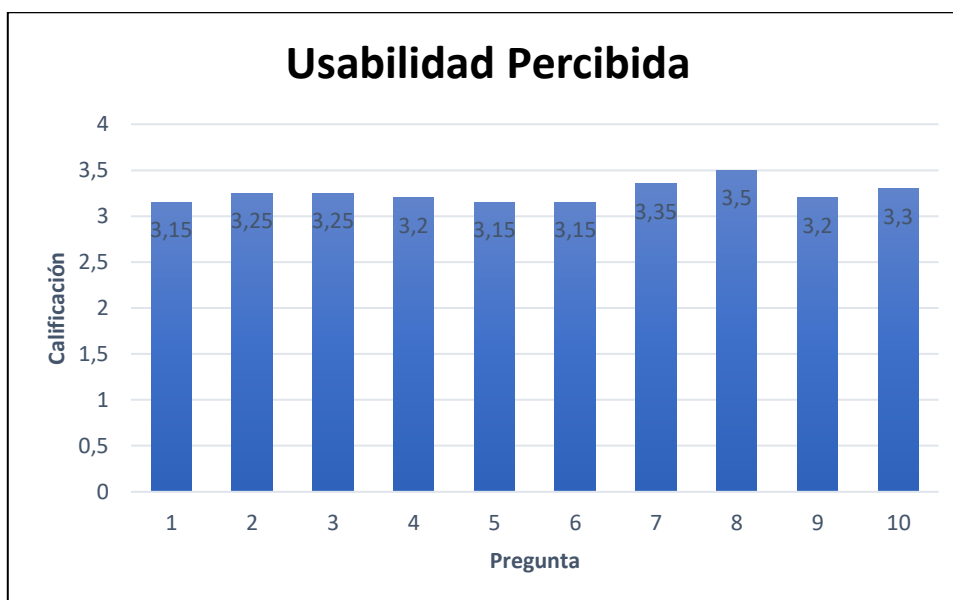


Figura 37. Promedio por pregunta de usabilidad percibida en el producto final

La pregunta 8 tiene la calificación más alta con un 3,5 lo cual indica que el sistema no fue incómodo para utilizar para los usuarios en su mayoría, por otro lado, la pregunta 1 indica que el sistema tuvo una aceptación entre los usuarios, este no es un sistema que utilizarían con frecuencia.

Para obtener la calificación total obtenida de la usabilidad percibida, es necesario sumar los valores de cada pregunta y multiplicar por 2.5, con eso se obtiene un valor de 81,25 en la prueba SUS.

Resultado de la prueba SUS:

$$(3,15+3,25+3,25+3,2+3,15+3,15+3,35+3,5+3,2+3,3)*2.5=81,25$$

Para interpretar los resultados es necesario convertirlo a un rango de percentil a través de la normalización, para generar los rangos de percentiles y calificaciones con letras de A a F [20], como se muestra en la Figura 38.

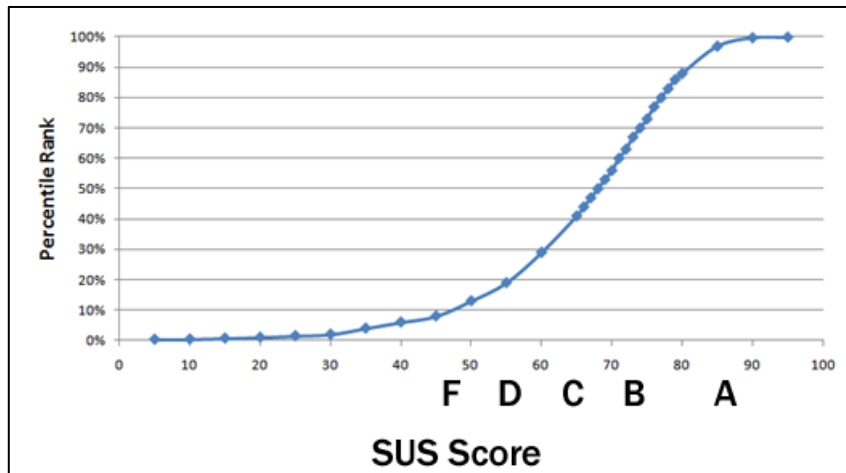


Figura 38. Puntaje de SUS [20].

Basado en esto se establece que las aplicaciones que tengan una puntuación superior a 80% tienen una buena aceptación por los usuarios, tomando como referencia el estudio realizado en la investigación realizada en el estudio de páginas WEB, solo el 10% de estas tienen una valoración superior a 80% [20].

3.2.2 Pruebas de usabilidad basadas en los principios de Jakob Nielsen

La prueba de usabilidad de Jakob Nielsen es un método para evaluar la usabilidad de una aplicación y mejorar la experiencia del usuario. A diferencia de otras técnicas de evaluación, este se enfoca en la observación directa de la interacción del usuario con la aplicación en tiempo real, lo que lo hace más efectivo para identificar problemas concretos y desarrollar soluciones específicas para mejorar la eficacia y eficiencia del sistema. [21]

Objetivos de la prueba de usabilidad de la aplicación móvil

1. Evaluar la facilidad de uso de la aplicación móvil para que los usuarios puedan identificar fácilmente los puntos de interés turísticos que indiquen el nivel de contagio de Covid-19 en el sector.
2. Identificar las áreas de la aplicación móvil que generan confusiones y dificultades en su uso, a fin de realizar mejoras en la navegación y diseño de la interfaz y asegurar que la información sobre los destinos turísticos se presente de manera clara y accesible.
3. Medir la satisfacción de los usuarios con la experiencia de uso de la aplicación móvil, para asegurar que se sientan seguros y confiados al utilizar la aplicación y que la información que se presenta sea relevante y útil.

4. Identificar los problemas técnicos y errores que puedan presentarse en la aplicación móvil, para poder corregirlos y mejorar el rendimiento general de la aplicación y garantizar que los usuarios puedan acceder a la información.

Introducción

La aplicación móvil ayuda a las personas interesadas en identificar el nivel de contagio de Covid-19 de cada sector de la ciudad en tiempos de pandemia. Para evaluar la eficacia y facilidad de uso de la aplicación, se realizó una prueba de usabilidad con 5 usuarios de diferentes edades. Este informe presenta los resultados de la evaluación, utilizando como referencia los principios de usabilidad de Jakob Nielsen.

Tareas:

1. Registrarse en aplicación utilizando un correo electrónico y una contraseña en la pantalla de registro.
2. Loguearse en la aplicación utilizando el correo electrónico y la contraseña.
3. En el menú de la aplicación seleccionar un sitio y un sector cualquiera, posteriormente presionar el botón 'ENVIAR'.
4. En el mapa seleccionar cualquier icono y presionar sobre el mismo, se despliega la venta flotante, después tocar cualquier parte del mapa, para que la ventana flotante desaparezca.
5. Seleccionar otro icono para que se despliegue la ventana flotante, y el botón ir para que se grafique la ruta en el mapa.
6. Regresar al menú con el botón de retroceso del dispositivo.
7. Seleccionar al menos 2 sitios para visitar y un sector diferente al selecciona previamente y presionar el botón enviar.
8. Seleccionar un icono cualquiera y presionar el botón ir, cuando se allá marcado la ruta seleccionar otro icono y presionar el botón ir para que se muestre la nueva ruta.

Preguntas:

- ¿Qué le pareció la interfaz de usuario de la aplicación?
- ¿Pudo encontrar la información que buscaba con facilidad?
- ¿Encontró algún problema al utilizar la aplicación? En caso afirmativo, ¿puede explicarlo?
- ¿Considera que la información proporcionada es clara y precisa?
- ¿Le resultó fácil elegir los puntos de interés turístico?
- ¿Pudo encontrar información sobre el nivel de contagio del punto de interés? ¿Le pareció suficiente?
- ¿Hubo alguna funcionalidad que le pareció innecesaria o confusa?
- ¿Le gustaría que se agregara alguna funcionalidad adicional a la aplicación?

¿Recomendaría esta aplicación a alguien que quiera visitar algún sector o sitio turísticos en tiempos de pandemia? ¿Por qué?

Respuestas

¿Qué le pareció la interfaz de usuario de la aplicación?

21 años: fácil de usar.

25 años: La encontré un poco abrumadora al principio, pero después de unos minutos la entendí.

36 años: La interfaz es clara y fácil de usar.

37 años: Simple

58 años: Me pareció un poco confusa, especialmente al principio.

¿Pudo encontrar la información que buscaba con facilidad?

21 años: Sí.

25 años: Sí, la encontré fácilmente.

36 años: Sí, encontré la información que buscaba rápidamente.

37 años: Sin ningún problema.

58 años: Sí encontré la información después que me indicaran como hacerlo.

¿Encontró algún problema al utilizar la aplicación? En caso afirmativo, ¿puede explicarlo?

21 años: No.

25 años: No, no tuve ningún problema.

36 años: No, todo funcionó sin problemas.

37 años: No, pero las indicaciones no son totalmente claras.

58 años: Sí, tuve dificultades para elegir los puntos de interés en el mapa.

¿Considera que la información proporcionada es clara y precisa?

21 años: Sí, sin mayor problema.

25 años: Sí, encontré la información clara y fácil de entender.

36 años: En su mayoría, sí. A veces sentí que faltaban detalles importantes.

37 años: Sí, la información proporcionada me pareció clara y precisa.

58 años: En general, sí.

¿Le resultó fácil elegir los puntos de interés turístico?

21 años: Sí, muy fácil.

25 años: En general, sí.

36 años: Sí, me resultó fácil elegir los puntos de interés.

37 años: Sí, sin mayor problema

58 años: Sí, pero a veces encontré difícil seleccionar un punto de interés en el mapa.

¿Pudo encontrar información sobre el nivel de contagio del punto de interés? ¿Le pareció suficiente?

21 años: Sí, encontré la información fácilmente y me pareció suficiente.

25 años: Sí, pude encontrar la información que buscaba.

36 años: Sí, encontré la información que buscaba y me pareció suficiente.

37 años: En general sí, pero me pareció un poco incompleta.

58 años: No necesariamente ya que, si muestra la información, pero no es preciso o cloro el hecho que voy a hacer con esa información.

Resultados

Esta prueba de usabilidad de la aplicación se realizó a grupo de 5 personas en un rango de edades que va desde los 21 a los 58 años, con el fin de analizar la facilidad de uso y las experiencias de los usuarios al utilizar la aplicación para obtener información del nivel de contagio de Covid-19 en algún punto de un interés turístico en tiempos de pandemia.

La prueba se llevó a cabo basada en los principios de usabilidad de Jakob Nielsen, durante la prueba, a los usuarios se les indicó varias a tareas realizar relacionadas con las principales funciones de la aplicación que el hecho de obtener información de los puntos de interés turístico.

Después de llevar a cabo las tareas, los usuarios fueron entrevistados para recopilar información sobre su experiencia al utilizar la aplicación, con el fin de identificar cualquier problema o dificultad que hayan encontrado, para posteriormente analizar los resultados y agregar recomendaciones de mejoras que se podrían implementar.

Conclusiones:

En general, la prueba de usabilidad de la aplicación permitió identificar algunos aspectos relevantes sobre la usabilidad y la experiencia de uso de la aplicación por parte de los usuarios. Se observó que la mayor parte de los problemas se presentaron cuando los usuarios seleccionaron puntos de interés en el mapa. A pesar de estos problemas, se puede concluir que la aplicación tiene un funcionamiento sencillo y que los usuarios en general pudieron realizar las tareas propuestas sin mayores dificultades.

Sin embargo, es importante destacar que el usuario de mayor edad (58 años) presentó una mayor confusión al momento de utilizar la aplicación, especialmente en lo que se refiere a la navegación y al uso de algunas funcionalidades. Esto sugiere que podrían ser necesarias algunas mejoras en la interfaz de usuario para mejorar la experiencia de los usuarios que no estén acostumbrados al uso de este tipo de aplicaciones, que puede ser agregando mensajes para guiar a los usuarios.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- En este trabajo se implementó un sistema de recomendaciones turísticas para la ciudad de Quito que indica información sobre el nivel de contagios de Covid-19. Lo más importante del sistema de recomendaciones turísticas es informar a los usuarios de los sitios turísticos más interesantes, indicando información sobre el nivel de contagio de Covid-19 en el sector donde se encuentra el punto de interés.
- Se utilizó la información publicada del Ministerios de Salud Pública del Ecuador, ya que es una fuente con datos oficiales, los datos utilizados para la aplicación es el número de personas contagiadas con Covid-19 en cada sector de la ciudad de Quito.
- Fue importante el uso de las valoraciones de los usuarios que han visitado el punto de interés turístico previamente para poder generar las recomendaciones a los usuarios que utilizan la aplicación.
- Para poder generar las rutas se requirió que el usuario otorgue los permisos de ubicación en el dispositivo de lo contrario la aplicación toma un punto preestablecido y de ese modo poder generar la ruta en el mapa hasta el punto de interés seleccionado.
- Se utilizó los servicios de Google Maps para obtener la información de los puntos de interés turísticos que se encuentran disponibles en cada sector de la ciudad y el servicio para obtener las rutas hasta el punto de interés turístico.
- Con Firebase se realizan las tareas de registro y logín de los usuarios en la aplicación utilizando el correo electrónico, también permite almacenar la información de los puntos de interés con los datos de contagios de Covid-19.
- Se realizó un Script en Python para poder procesar los datos del número de contagios de Covid-19 en cada sector con la información de los puntos de interés, este lenguaje de programación permitió procesar los datos de una manera muy simple, dado que este lenguaje conocido en el campo de análisis de datos.
- Para determinar el nivel de contagio de Covid-19 en cada sector sea este bajo, medio y alto, se toma como referencia el número de contagiados de cada sector y se hace un promedio, el cual fue utilizado para crear el rango del nivel de contagio.

- Se cumplieron todos los objetivos planteados, iniciando por identificar los servicios expuestos sobre las zonas de riesgo de contagio de Covid-19, que se obtuvo por medio de los datos del número de personas contagiadas en cada sector de la ciudad de Quito, por otro lado se diseñó e implementó el sistema de recomendación turística utilizando los servicios de Google Maps con lo referente a los puntos de interés (POI) con mayor valoración, también se probó el sistema de recomendación turística con varios usuarios y por último se midió la usabilidad con la opinión de los usuarios que probaron la aplicación.

4.2 Recomendaciones

Para trabajos futuros se recomienda:

- Implementar el registro de los usuarios mediante otras plataformas como las cuentas Facebook y Google, y agregar una confirmación para validar que el correo electrónico es real.
- Implementar una función para poder calificar los establecimientos que el usuario ya visitó, para poder mejorar las futuras recomendaciones a otros usuarios.
- Desarrollar la aplicación para otros sistemas operativos como la plataforma IOS para poder llegar a un mayor número de usuarios.
- Implementar una funcionalidad para poder obtener la información del número de personas contagiadas con Covid-19 en sector de manera automática.
- Implementar la funcionalidad para determinar los puntos de interés turístico que se encuentren abiertos en el momento que se está utilizando la aplicación, y de ese modo asegurar que los usuarios puedan ir a los sitios en la noche.
- Implementar el seguimiento por GPS mientras el usuario se dirige a los diferentes puntos de interés turísticos.
- Implementar la funcionalidad para poder gestionar los puntos de interés, que permita quitar los puntos de interés que el usuario ya no desee visitar.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] «Tecnologías para luchar contra la pandemia Covid-19: geolocalización, rastreo, big data, SIG, inteligencia artificial y privacidad,» E-prints in library & information science, 2020. [En línea]. Available: <http://eprints.rclis.org/40350/>. [Último acceso: 12 12 2020].
- [2] «Actualización de casos de coronavirus en Ecuador,» MSP, [En línea]. Available: <https://www.salud.gob.ec/actualizacion-de-casos-de-coronavirus-en-ecuador/>. [Último acceso: 2020 12 12].
- [3] «Situación Pichincha por Covid-19,» COE Provincial Pichincha, 2020. [En línea]. Available: <https://coe-pichincha.senescyt.gob.ec/>. [Último acceso: 12 12 2020].
- [4] «SALVAVIDAS,» EPN, 2020. [En línea]. Available: <https://covidmodemat.epn.edu.ec/#/>. [Último acceso: 20 12 2020].
- [5] osi.es, «Geolocalización: virtudes y riesgos,» Oficina de Seguridad del Internauta, 20 09 2016. [En línea]. Available: <https://www.osi.es/es/actualidad/blog/2016/09/20/geolocalizacion-virtudes-y-riesgos#:~:text=La%20geolocalizaci%C3%B3n%20consiste%20en%20obtener,GPS%20de%20nuestro%20tel%C3%A9fono%20m%C3%B3vil..> [Último acceso: 15 04 2022].
- [6] thesite.org, «How to Locate a Mobile Using Geolocation,» TheSite, 22 12 2021. [En línea]. Available: <https://www.thesite.org/locate-mobile-using-geolocation/>.. [Último acceso: 15 04 2022].
- [7] W. Shi, «Recommendation Systems: A Review,» towardsdatascience, 23 2 2020. [En línea]. Available: <https://towardsdatascience.com/recommendation-systems-a-review-d4592b6caf4b>. [Último acceso: 15 04 2022].
- [8] L. R. & B. S. Francesco Ricci, «Recommender Systems: Techniques, Applications, and Challenges,» 22 11 2021. [En línea]. Available: https://link.springer.com/chapter/10.1007/978-1-0716-2197-4_1. [Último acceso: 15 04 2022].
- [9] developers.google, «Maps SDK for Android Quickstart,» Google Maps Platform, [En línea]. Available: <https://developers.google.com/maps/documentation/android-sdk/start>. [Último acceso: 15 04 2022].
- [10] T. N. K. C. Xavier Ducrohet, «Android Studio: An IDE built for Android,» Android Developers Blog, 15 05 2013. [En línea]. Available: <https://android-developers.googleblog.com/2013/05/android-studio-ide-built-for-android.html>. [Último acceso: 15 04 2022].
- [11] developer.android, «Develop Android apps with Kotlin,» Android Developers, [En línea]. Available: <https://developer.android.com/kotlin>. [Último acceso: 20 04 2022].

- [12] «Kotlin Programming Language,» JetBrains, [En línea]. Available: <https://github.com/JetBrains/kotlin>. [Último acceso: 20 04 2022].
- [13] «Promote and advance the development of the Kotlin programming language,» Kotlin Foundation, [En línea]. Available: <https://kotlinfoundation.org/>. [Último acceso: 20 04 2022].
- [14] F. Lardinois, «Google Acquires Firebase To Help Developers Build Better Real-Time Apps,» <https://techcrunch.com/>, 21 10 2014. [En línea]. Available: <https://techcrunch.com/2014/10/21/google-acquires-firebase-to-help-developers-build-better-realtime-apps/>. [Último acceso: 20 04 2022].
- [15] D. Kuhlman, A Python Book: Beginning Python, Advanced Python, and Python Exercises, Platypus Global Media, 2011.
- [16] scrumguides.org, «The 2020 Scrum Guide,» Scrum Guides, 2020. [En línea]. Available: <https://scrumguides.org/scrum-guide.html>. [Último acceso: 21 04 2022].
- [17] J. W. Tomasz Wykowski, «Lessons learned: Using Scrum in non-technical teams,» Agile Alliance, 05 2018. [En línea]. Available: <https://www.agilealliance.org/resources/experience-reports/lessons-learned-using-scrum-in-non-technical-teams>. [Último acceso: 21 04 2022].
- [18] «Creative tools meet the internet.,» www.figma.com, [En línea]. Available: <https://www.figma.com/about/>. [Último acceso: 26 06 2022].
- [19] J. Brooke, «SUS - A quick and dirty usability scale,» uifrommars.com, 1995. [En línea]. Available: https://www.researchgate.net/publication/228593520_SUS_A_quick_and_dirty_usability_scale. [Último acceso: 10 08 2022].
- [20] P. Jeff Sauro, «Measuring Usability with the System Usability Scale (SUS),» Measuring U, 3 02 2011. [En línea]. Available: <https://measuringu.com/sus/>. [Último acceso: 10 8 2022].
- [21] J. Nielsen, «Usability 101: Introduction to Usability,» www.nngroup.com, 1 3 2012. [En línea]. Available: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>. [Último acceso: 14 2 2023].

6 ANEXOS

6.1 Anexo I: Historias Épicas

Tabla 20. Historia Épica AH1

HISTORIA DE USUARIO ÉPICA	
ID: AH1	Prioridad: Alta
Título: Identificar la cantidad de personas contagiadas de Covid-19 en los diferentes sectores de la ciudad	
Descripción: Como usuario, necesito obtener información del nivel de contagio de Covid-19 en cada sector de la ciudad.	

Tabla 21. Historia Épica AH2

HISTORIA DE USUARIO ÉPICA	
ID: AH2	Prioridad: Alta
Título: Integrar los datos del nivel de contagio de Covid-19 en cada sector, con los puntos de interés turístico de la ciudad.	
Descripción: Como usuario, necesito conocer los puntos de interés según mis requerimientos de que tipo de punto de interés y el nivel de contagio de Covid-19.	

Tabla 22. Historia Épica AH3

HISTORIA DE USUARIO ÉPICA	
ID: AH3	Prioridad: Alta
Título: Desarrollo de la aplicación donde se despliegue el nivel de contagio de Covid-19 con los puntos de interés turísticos de cada sector en el mapa de la ciudad.	
Descripción: Como usuario, necesito una interfaz de usuario intuitiva, de tal forma que pueda visualizar los puntos de interés conforme a mis requerimientos.	

6.2 Anexo II: Historias de Usuario

Historias de Usuario del Sprint 1

Tabla 23. Historia de Usuario AH1-01

HISTORIA DE USUARIO		
ID: AH1-01	Esfuerzo: 8	
Título: Identificar el número de personas contagiadas de Covid-19 en los diferentes sectores de la ciudad e integrar con los puntos de interés turístico.		
Sprint N°: 1	Prioridad: Alta	Estado: Por implementar
Descripción: como usuario, deseo conocer el nivel de contagio de Covid-19 que tiene cada punto de interés turístico de la ciudad.		
Criterios de aceptación		
Dado que	Cuando	Entonces
Se requiere conocer la información de los contagios en los sectores de la ciudad.	Se extrae los datos del número de contagiados de Covid-19 de las diferentes fuentes de información.	Se obtiene la información del número de personas contagiadas con Covid-19 por cada sector.

Tabla 24. Historia de Usuario AH1-02

HISTORIA DE USUARIO		
ID: AH1-02	Esfuerzo: 3	
Título: Cargar los datos del nivel de contagio de Covid-19 en los diferentes sectores de la ciudad con los puntos de interés turístico a la base de datos de Firebase.		
Sprint N°: 1	Prioridad: Alta	Estado: Por implementar
Descripción: como usuario, deseo ver los puntos de interés que estén disponibles en la ciudad		
Criterios de aceptación		
Dado que	Cuando	Entonces
El usuario requiere identificar los puntos de interés	El usuario solicita un tipo de punto de interés.	Se obtiene como resultado los puntos de interés con el número de contagios de Covid-19 en el sector.

Historias de Usuario del Sprint 2

Tabla 25. Historia de Usuario AH2-01

HISTORIA DE USUARIO		
ID: AH2-02	Esfuerzo: 5	
Título: Crear proyecto Android Studio e integrar con los servicios de Google Maps y Firebase.		
Sprint N°: 2	Prioridad: Alta	Estado: Por implementar
Descripción: como usuario, deseo utilizar una aplicación móvil para dispositivos Android.		
Criterios de aceptación		
Dado que	Cuando	Entonces
Se configura Android Studio para el desarrollo del proyecto	El usuario utiliza un dispositivo Android.	La aplicación funcionará en dispositivos Android

Tabla 26. Historia de Usuario AH2-02

HISTORIA DE USUARIO		
ID: AH3-01	Esfuerzo: 8	
Título: Diseñar y desarrollar las interfaces de usuario.		
Sprint N°: 2	Prioridad: Alta	Estado: Por implementar
Descripción: como usuario, deseo utilizar una interfaz donde pueda ingresar mis preferencias y visualizar los resultados.		
Criterios de aceptación		
Dado que	Cuando	Entonces
El usuario acceda a la pantalla	cuando el usuario desee realizar una acción en la aplicación	Se mostrará la pantalla correspondiente en base a los datos ingresados.

Historias de Usuario del Sprint 3

Tabla 27. Historia de Usuario AH3-01

HISTORIA DE USUARIO		
ID: AH3-02	Esfuerzo: 13	
Título: Desarrollo de la lógica del formulario y el mapa de aplicación.		
Sprint N°: 3	Prioridad: Alta	Estado: Por implementar
Descripción: como usuario, deseo ver los puntos de interés y las rutas para llegar a cada uno de allá seleccionado.		

Criterios de aceptación		
Dado que	Cuando	Entonces
El usuario accede a la pantalla de inicio	El usuario inicia la aplicación	Se despliega la pantalla donde debe ingresar el usuario y contraseña.
El usuario accede a la pantalla de registro	El usuario no se ha registrado en la aplicación	Se despliega la pantalla donde el usuario ingresa la información para registrarse en la aplicación
El usuario accede a la pantalla del menú.	El usuario allá iniciado de manera correcta	Se muestra las opciones que el usuario debe elegir para buscar los puntos de interés.
El usuario accede a la pantalla del mapa	El usuario selecciona las opciones de los puntos de interés que fueron requeridos.	Se despliega los puntos de interés en base a los requerimientos del usuario.
El usuario seleccionó un punto de interés (POI)	El usuario selecciona en ir al punto de interés	Se muestra la ruta para llegar al punto de interés desde el punto donde se encuentra.

6.3 Anexo III: Resultados de las Encuestas Realizadas a los Usuarios Sobre Usabilidad de la Aplicación

Pregunta 1

Creo que me gustaría utilizar este sistema frecuentemente.

20 respuestas

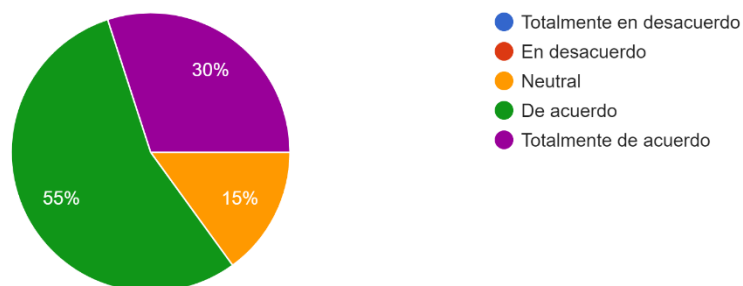


Figura 39. Resultados de las preguntas 1 sobre usabilidad.

Pregunta 2

El sistema me resultó innecesariamente complejo.
20 respuestas

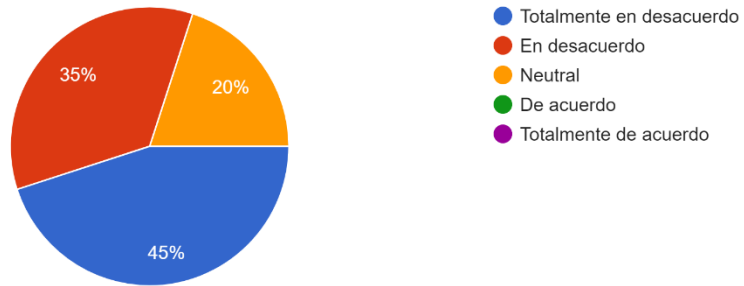


Figura 40. Resultados de las preguntas 2 sobre usabilidad.

Pregunta 3

Creo que el sistema es bastante fácil de utilizar.
20 respuestas

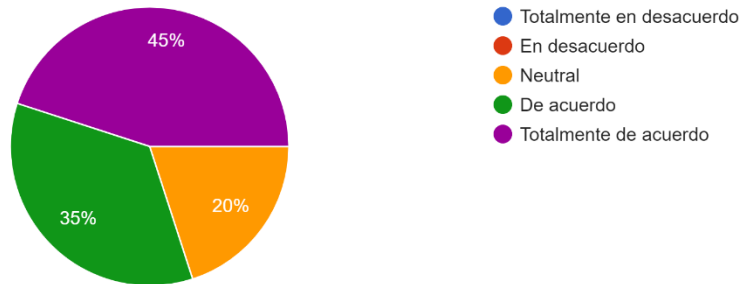


Figura 41. Resultados de las preguntas 3 sobre usabilidad.

Pregunta 4

Creo que necesitaría el soporte de un técnico para poder utilizar este sistema.
20 respuestas

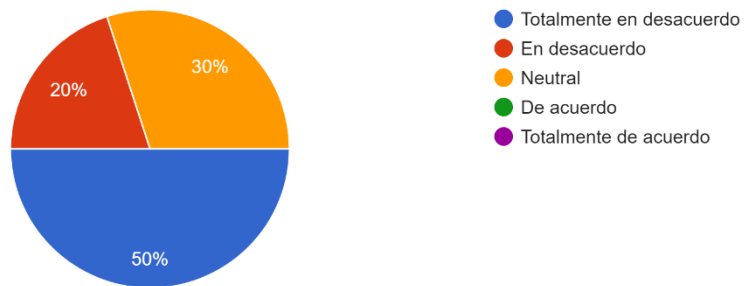


Figura 42. Resultados de las preguntas 4 sobre usabilidad.

Pregunta 5

Creo que las diferentes funciones del sistema se encuentran muy bien integradas.
20 respuestas

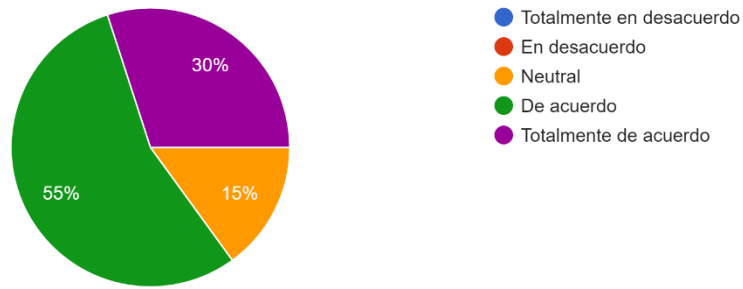


Figura 43. Resultados de las preguntas 5 sobre usabilidad.

Pregunta 6

Opino que hubo demasiada inconsistencia en el sistema.
20 respuestas

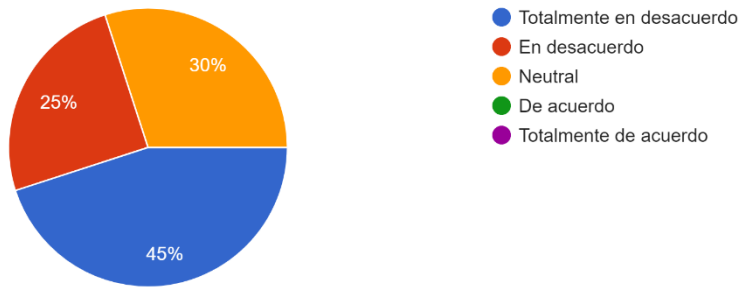


Figura 44. Resultados de las preguntas 6 sobre usabilidad.

Pregunta 7

Imagino que la mayoría de las personas aprendería a utilizar el sistema rápidamente.
20 respuestas

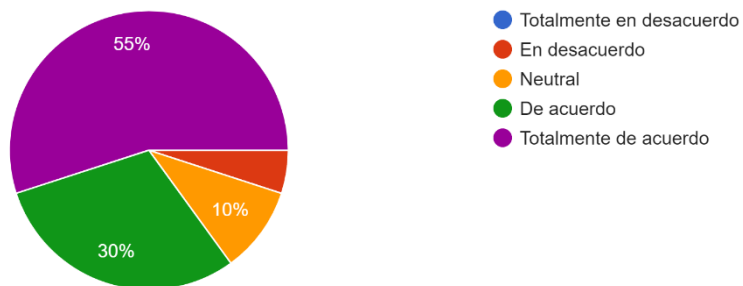


Figura 45. Resultados de las preguntas 7 sobre usabilidad.

Pregunta 8

Me sentí algo incómodo al utilizar este sistema.
20 respuestas

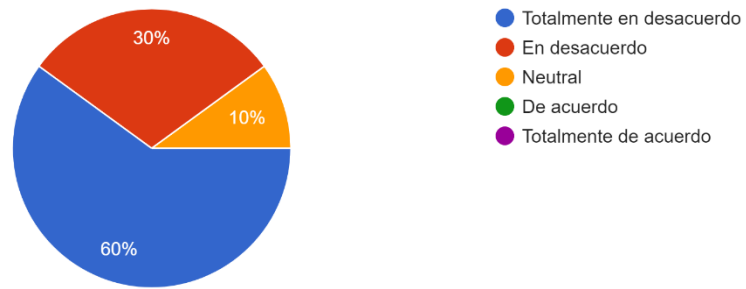


Figura 46. Resultados de las preguntas 8 sobre usabilidad.

Pregunta 9

Me sentí muy seguro al utilizar este sistema.
20 respuestas

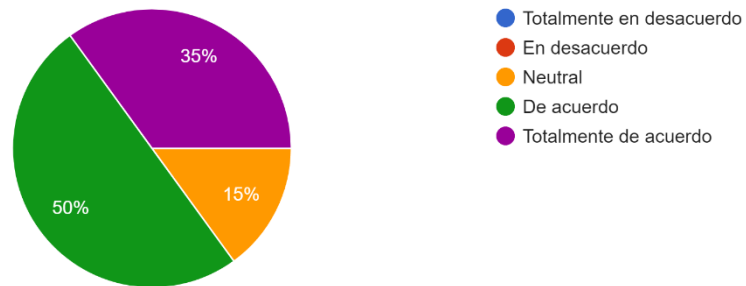


Figura 47. Resultados de las preguntas 9 sobre usabilidad.

Pregunta 10

Necesito aprender muchas otras cosas antes de poder utilizar correctamente el sistema.
20 respuestas

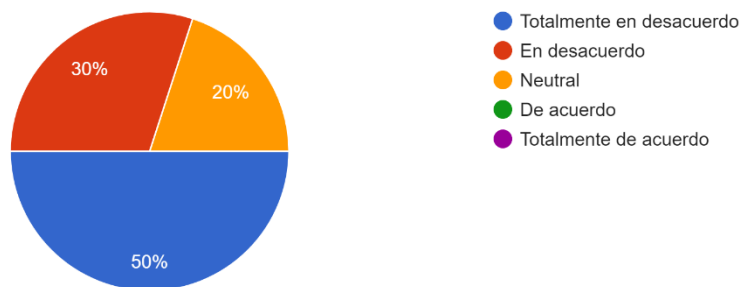


Figura 48. Resultados de las preguntas 10 sobre usabilidad.