

UNIVERSIDAD CATÓLICA SANTO TORIBIO DE MOGROVEJO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN



**APLICACIÓN WEB MÓVIL PARA GESTIONAR LOS SERVICIOS
TÉCNICOS PARTICULARES EN LA CIUDAD DE CHICLAYO**

**TESIS PARA OPTAR EL TÍTULO DE
INGENIERO DE SISTEMAS Y COMPUTACIÓN**

AUTOR

JHON ANDERSON ZULOETA LOPEZ

ASESOR

KARLA CECILIA REYES BURGOS

<https://orcid.org/0000-0003-3520-5076>

Chiclayo, 2021

**APLICACIÓN WEB MÓVIL PARA GESTIONAR LOS
SERVICIOS TÉCNICOS PARTICULARES EN LA CIUDAD DE
CHICLAYO**

PRESENTADA POR
JHON ANDERSON ZULOETA LOPEZ

A la Facultad de Ingeniería de la
Universidad Católica Santo Toribio de Mogrovejo
para optar el título de

INGENIERO DE SISTEMAS Y COMPUTACIÓN

APROBADA POR

Mariana Chavarry Chankay
PRESIDENTE

Jury Yesenia Aquino Trujillo
SECRETARIO

Karla Cecilia Reyes Burgos
VOCAL

Dedicatoria

La presente investigación la dedico a mis padres y hermanos por el apoyo incondicional que me brindaron para lograr una de mis metas establecidas durante el periodo de mi carrera universitaria.

De igual modo a los ingenieros que formaron parte de este proceso, y gracias a eso puedo desenvolverme fácilmente en mi carrera profesional.

Agradecimientos

Agradezco a Dios por darme la oportunidad de estudiar y tener las capacidades suficientes para poder avanzar un paso más como persona y profesional. A mis padres y hermanos por apoyarme en este proceso y momentos difíciles como persona.

A la Ing. Karla Reyes Burgos, a quién me orientó en este arduo proceso, quien fue una de las personas que me brindó ánimos para culminar el proyecto. Agradezco su tolerancia, ánimos y conocimientos impartidos que me ayudaron en mi formación profesional

A los maestros de la Escuela de Ingeniería de Sistemas y Computación que me compartieron sus experiencias, conocimientos y me guiaron durante mi carrera profesional. A mi familia y amigos que han brindado su apoyo incondicional a lo largo de mi carrera universitaria.

Índice

Resumen.....	5
Abstract.....	6
Introducción	7
Revisión de literatura	8
Materiales y métodos.....	11
Resultados y discusión	12
Conclusiones	37
Recomendaciones.....	38
Referencias.....	39
Anexos.....	42

Resumen

Hoy en día el uso de las tecnologías de información ha revolucionado a toda la generación, lo cual permite que todos estemos más conectados e informados en el día a día. En la presente investigación se tomó como objeto de estudio a los trabajadores independientes, ya que las cifras de ellos han incrementado en cada año, siendo los principales dueños de su trabajo presentando diversos problemas como el financiamiento de su negocio y captación de clientes.

Así mismo a los ciudadanos ya que no cuentan con un acceso a la información para solicitar un servicio técnico o que estas se encuentren en las redes sociales, lo que ocasiona que no les sea suficientemente útil. En tal sentido fue necesario mejorar este hecho en obtener datos importantes de los trabajadores independientes que brindan servicios técnicos siendo relevantes para el usuario. El presente trabajo de investigación es la “APLICACIÓN WEB MÓVIL PARA GESTIONAR LOS SERVICIOS TÉCNICOS PARTICULARES EN LA CIUDAD DE CHICLAYO”, se desarrolló con el propósito de apoyar a los técnicos a gestionar sus servicios y poder captar clientes. Así mismo, mantener informados a los ciudadanos, permitiéndoles escoger cuál es el técnico más adecuado para poder contactarlo, ya sea por ubicación, calificación de otros usuarios, recomendación de la aplicación por nivel de valoración, precios de servicio, entre otros.

En el desarrollo del software se utilizó la metodología RUP dividido en cuatro iteraciones, la cual asegura la producción de software de alta calidad satisfaciendo las necesidades del cliente, mostrando como producto final el software para la gestión de información.

De esta manera se concluye, que con la metodología RUP se obtuvo un software que gestiona la información de los servicios de cada técnico integrando un sistema de recomendación, siendo de fácil uso para el usuario.

Palabras clave: Geolocalización, sistemas de recomendación, Aplicaciones móviles.

Abstract

Today, the use of information technologies has revolutionized the entire generation, allowing us to be more connected and informed on a day-to-day basis. In this research, independent workers are the object of study, since their numbers have increased each year, being the main owners of their work presenting various problems such as financing their business and attracting customers.

Likewise, citizens since they do not have access to information to request a technical service or are on social networks, which causes them not to be useful enough. In this sense, it is necessary to improve this fact in obtaining important data from independent workers who provide technical services that are relevant to the user. The present research work is the “MOBILE WEB APPLICATION TO MANAGE THE PARTICULAR TECHNICAL SERVICES IN THE CITY OF CHICLAYO”, it is treated with the purpose of supporting the technicians to manage their services and be able to attract clients. Likewise, keeping citizens informed, allowing them to choose the most suitable technician to be able to contact them, either by location, rating of other users, recommendation of the application by level of assessment, service prices, among others.

In the development of the software, the RUP methodology will be divided into four iterations, the qualification ensures the production of high-quality software satisfying the needs of the client, showing the software for information management as a final product.

In this way, it is concluded that with the RUP methodology, a software was certified that manages the information of the services of each technician, integrating a recommendation system, being user-friendly.

Keywords: Geolocation, recommendation systems, mobile applications.

Introducción

En el ámbito mundial según la Organización Internacional del Trabajo (OIT) demuestra que siete de cada diez trabajadores son independientes o trabajan en pequeñas empresas. Los datos recogidos de 99 países indican que, en conjunto que estas pequeñas unidades económicas representan el 70% del empleo total, siendo los que más empleo generan. Además, el 62% del empleo corresponde al sector informal, dado así las condiciones de trabajo son inferiores debido a la falta de seguridad social, salarios más bajos y deficiencias [1].

En Perú se ha demostrado que el trabajo independiente en el año 2015 representó el 35.1% de la población ocupada como no profesionales, técnicos o administrativos, sin embargo, en el año 2017 aumentó al 37%. Por lo que cada año las cifras van aumentando de personas que trabajan independientemente careciendo de beneficios propios como acceder a un sistema de seguridad social. Esto se debe a una reducida demanda laboral, y de los efectos limitados de la formalización de empleos formales preexistentes, además de la falta de empleo de calidad [2].

Por otro lado, se puede observar el problema de los usuarios al no poder contactar un servicio técnico particular o persona que trabaja independientemente. En el año 2018 según el Reporte Digital identificó que el 68% de la población equivalente a 22 millones de usuarios usan internet para las redes sociales, mas no el uso de aplicativos para la facilitación de acceso a la información. Cabe recalcar que el 45% son mujeres y 55% son hombres; además el 91% de usuarios se conecta a través de un smartphone [3].

El principal problema en la ciudad de Chiclayo radica desde la necesidad por algún servicio técnico, el no saber a qué establecimiento dirigirse por el difícil acceso a la información, ya sea por electricista, cerrajero, computación e informática, entre otros. La cual existen momentos imprevistos por los ciudadanos ocurriéndoles momentos trágicos como malograrse el carro en una carretera, y no poder contactar a alguien que solucione su problema.

La presente tesis denominada “APLICACIÓN WEB MÓVIL PARA GESTIONAR LOS SERVICIOS TÉCNICOS PARTICULARES EN LA CIUDAD DE CHICLAYO”, se inicia con el propósito de mantener informado a los usuarios o posibles clientes sobre la ubicación laboral de cada técnico, la cual le permitirá comparar acerca de qué técnico es recomendable para cumplir ciertas necesidades de negocio, ya sea por ubicación, servicios que brinda, precios de servicios o calificación de usuarios.

Ante esta realidad, es importante formular la siguiente pregunta ¿De qué manera una aplicación web móvil gestionará los servicios técnicos particulares en la ciudad de Chiclayo?

Frente a esta pregunta y la necesidad de profundizar el problema, se realizó la investigación de tipo aplicada cuya población fue de 15 749 personas que trabajan independientemente [4] y 165 397 de personas adulta joven en la ciudad de Chiclayo [5]. Para ello, se determinó implementar una aplicación web móvil para gestionar los servicios técnicos particulares integrando un sistema de recomendación de usuarios en la ciudad de Chiclayo, considerando las siguientes fases: inicio, elaboración, construcción y transición; es primer lugar, se tuvo que identificar una técnica de sistema de recomendación que permita la similitud de preferencias de usuario; luego determinar las APIS¹ necesarias de Google Maps para influir en la gestión de los servicios técnicos particulares; posteriormente desarrollar la aplicación web móvil en la que se integre las APIS determinadas y la técnica del sistema de recomendación de usuarios; luego, obtener una percepción positiva en la utilidad y facilidad de uso de la aplicación por parte de los técnicos.

¹ Conjunto de funciones, procedimientos y subrutinas que ofrece una biblioteca para ser utilizado por otro software.

Ver <https://web.archive.org/web/20190215165536/https://blog.pandorafms.org/es/para-que-sirve-una-api/>

Revisión de literatura

Para implementar la aplicación web móvil que gestione los servicios técnicos particulares integrando un sistema de recomendación de usuarios en la ciudad de Chiclayo se analizaron los antecedentes indicados a continuación:

En las investigaciones internacionales, Flores [6] narra la problemática de la dificultad de ubicar los restaurantes de acuerdo con las necesidades de los usuarios, logrando obtener la georreferenciación de restaurantes en la ciudad de la paz para la problemática del débil acceso a la información de cada restaurante. El valor agregado de esta investigación es aplicar el algoritmo que permite mostrar en un mapa todos los restaurantes ya sea por categoría, servicios, entre otros. Finalmente, el autor concluyó que la aplicación móvil desarrollada crea un espacio para el registro de restaurantes, la cual podrán ser localizadas de acuerdo con las necesidades de los usuarios donde ellos podrán criticar según el servicio que se le haya brindado. Por otra parte, Troya [7] describe la problemática por parte de los turistas al momento de localizar sitios turísticos más convenientes para visitar. En este caso la agencia de viaje “ECOMONTESTOUR” al no estar familiarizada con las nuevas tecnologías no puede ofrecer un servicio de calidad hacia los turistas al realizar tours, ya que no hay un punto de referencia para darle al ciudadano seguridad en donde está ubicado, lo que conlleva a pérdidas de clientes y bajos ingresos. Se aplicó la metodología cualitativa ya que se requiere de conocimientos para comprender la información de los lugares turísticos y cuantitativa para analizar los estudios analizados numéricamente, logrando obtener un sistema de geolocalización de sitios turísticos. la cual podrán ser localizadas de acuerdo con las necesidades de cada turista desde su posición actual. Mientras tanto, Tambaco [8] narra que hoy en día el uso de las tecnologías es muy común como en el caso de la geolocalización que nos brinda servicios esenciales permitiendo obtener la información que cualquier persona quiera conocer, ya sea por dirección de domicilio, empresa u otros de manera precisa. Se aplicó la metodología RUP para lograr una aplicación web interactiva para el desarrollo de un sistema de información de geolocalización para la visualización geográfica y multimedia en la universidad central del Ecuador con el fin de que los usuarios o posibles estudiantes puedan visualizar las instalaciones de la facultad de ingeniería ciencias físicas y matemática.

En las investigaciones nacionales, Quispe y Yallico [9] describen la problemática sobre la congestión vehicular, empezando por las avenidas principales, a esto se suma que a los conductores se les ve reflejado el desconocimiento de la ubicación de las cocheras en la ciudad de Huancayo, lo cual en el afán de buscar los choferes se demoran de la congestión vehicular generando estrés, pérdida de tiempo y gasolina; sobre todo al no encontrar un establecimiento dejan su vehículo en las calles, lo que los exponen a la delincuencia. En esta tesis se aplicó la metodología ágil SCRUM para el desarrollo de software, logrando obtener la solución de una aplicación web de geolocalización de cocheras. El valor agregado de esta investigación fue el uso de las técnicas de geolocalización, ya que permitieron saber la ubicación exacta de cada establecimiento. Finalmente, Quispe y Yallico concluyeron que la aplicación facilita el rápido acceso a la información de cada cochera que existen en la ciudad de Huancayo y sus espacios de estacionamiento para poder reservar, también ayuda a reducir el congestionamiento vehicular ahorrando la gasolina y disminuir el tiempo de búsqueda de alguna cochera. La razón por la que se consideró esta tesis es porque los tesisistas usaron tecnologías de geolocalización en la web, lo cual también se planteó usar en esta investigación. Por otro lado, Gamboa [10] describe que la empresa Nueva Ladrillera Huachipa no cuentan con tecnologías de información que gestionen el sistema de inventario, marcación de asistencia, de RR. HH, vigilancia, ventas y cotizaciones. La cual dentro del proceso de ventas uno de los problemas que se identificó es que los transportistas desconocen las direcciones de los clientes para entregar los artículos solicitados. Se aplicó la metodología RUP ya que está más enfocada a los procesos obteniendo

resultados rápidos, logrando obtener un sistema web para el proceso de ventas con monitoreo de mercadería por geolocalización para contar con tecnologías de información que impliquen mejor el rendimiento de la empresa. El valor agregado de esta investigación es la geolocalización de entrega de productos al cliente al módulo de ventas. Donde, Gamboa concluyó que la empresa mejoró el 61.9% de la administración de las ventas, el 70.53% en la gestión del despacho de mercadería, el 73.2% en la gestión de cotización y el proceso de ventas en el 68.54%. Por otra parte, Cueva [11] especifica que en los Laboratorios Siegfried S.A.C, se identificó el problema de la gestión de visitas, siendo en muchos casos el desplazamiento innecesario de los visitantes médicos, es decir desconocen la ubicación exacta de los clientes (Médicos, Farmacias, Clínicas). Así mismo el registro de cada doctor se lleva manera tradicional lo que conlleva al acumulación de hojas generando demoras en el procesamiento de información. Se aplicó la metodología Mobile-d para el desarrollo ya que se ajusta en tiempos cortos, logrando obtener una Aplicación móvil para la gestión de visitas médicas para gestionar la información de las visitas médicas y la ubicación exacta de cada médico, farmacia o clínica. El valor agregado de la investigación es que el tesista incluyó un módulo de geolocalización para el rápido acceso a la ubicación y así poder reducir las demoras en el tiempo de cada visitador. Finalmente, el autor concluyó que el sistema permitió reducir el tiempo de los informes al momento de registrar de 13.35 minutos a 1.9 minutos, también en el establecimiento de rutas de 10.08 minutos a 1.55 minutos.

En las investigaciones locales, Fernández y Querevalu [12] declaran que hoy en día el servicio de taxi en la ciudad de Chiclayo no se encuentra en un mercado ordenado, debido a la falta de regulación por parte del gobierno conllevando a la saturación del mercado, añadiendo la gran inseguridad ciudadana que se vive en la ciudad. Se aplicó la metodología Lean Canvas para analizar las preferencias y requerimientos de los clientes. Logrando así plantear una propuesta de negocio para el servicio de taxi en proporción a las preferencias de usuario. El valor agregado es que los tesisistas incluyeron ideas innovadoras de atención al cliente antes y después del servicio de un taxista, además propuestas tecnológicas de GPS para el monitoreo de cada chofer. Finalmente, el autor concluyó que la puntualidad, confianza, seguridad y buenas condiciones del vehículo son aspectos importantes al momento de solicitar un servicio de taxi. La razón por la que se consideró esta tesis es porque se planteó cubrir las necesidades de los clientes al momento de solicitar un servicio promoviendo la seguridad ciudadana, recalando que la mayoría de las personas utilizan su smartphone para contactarlos. Dicha mención será de utilidad para la presente investigación. Por otra parte, Rojas y Villalobos [13] narran que el Salón y Spa María Bonita cuentan con servicios de atención al cliente como pedicura, manicure, entre otros. Surgiendo como necesidad medir el nivel de calidad de servicios de la empresa. Se aplicó la metodología SERQUAL para la medición de la calidad y servicio como la fiabilidad, responsabilidad, seguridad, empatía y tangibles. Logrando así establecer los resultados en base a los niveles mostrados por la metodología. El valor agregado de la tesis es que las tesisistas evaluaron la calidad de los servicios que brinda la empresa para satisfacer las necesidades de los clientes. Finalmente se concluyó que la empresa presenta deficiencias respecto a los servicios que ofrece especialmente en la dimensión empatía, ya que no atiende las necesidades del cliente. Por otra parte, Carrasco y Díaz [14] describen que las personas que trabajan en limpieza de la Municipalidad provincial, recolectan un promedio diario de 350 toneladas de basura domiciliaria donde el área de desarrollo distribuyó en 32 zonas, atendidas por 8 compactadoras en dos turnos, la cual no cuenta con rutas que justifique operativamente el recojo de basura y que asegure eficientemente sus recursos como el combustible. Se adaptó la metodología de investigación de operaciones usando el modelo matemático de distribución para destinar un compactador adecuado según su capacidad. Logrando así aumentar la eficiencia del servicio de limpieza y minimizar las distancias, tiempos y costos; así mismo maximizar los recursos del empleado para realizar el trayecto de las diferentes rutas. El valor agregado de la

tesis es que los testistas evaluaron la determinación de rutas cortas para reducir costos y así poder dar un mejor servicio de limpieza abasteciendo a la población con la compactadora adecuada por cada zona. Finalmente, los autores concluyeron que se complementó con mejoras adicionales como la distribución de compactadora en cada zona aplicando un modelo de ruta más corta para ir de planta a zona.

1. Sistemas de Recomendación

Un sistema de recomendación busca predecir la preferencia que un usuario otorgaría a un producto o servicio en uso. Es decir que usuarios con actividades similares continúen compartiendo preferencias en el futuro [15].

1.1. Técnicas de filtrado

Existen diferentes técnicas de filtrado que utilizan algoritmos de recomendación: basado en contenido, filtrado colaborativo e híbridos.

a. Sistemas de recomendación basados en contenido

Este tipo de sistema de recomendación se apoya en la recuperación de información. Es decir, se centran en recomendar artículos que tengan textos de información, como documentos o sitios web.

En tal sentido se ajustan en elementos similares a los que un usuario le gustaba en el pasado [15].

b. Técnica de filtrado colaborativo

Es una técnica para hacer pronósticos acerca de las preferencias de un usuario, en función de la información a los diferentes gustos de cada persona. Existen dos formas del filtrado colaborativo [15].

✓ Enfoque basado en el usuario

La lista de elementos se crea en función de los clientes. Las características comunes entre los usuarios se utilizan para predecir los niveles de producto de preferencia. Este filtrado es muy fácil de utilizar ya que el procesamiento de la información es bastante rápido [15].

✓ Enfoque basado en elementos:

Se crea la lista basada en productos. La similitud se calcula en base a los elementos de cada producto para establecer los niveles de preferencia. Este filtrado es complicado y conlleva a una gran cantidad de operaciones debido a que cada usuario tiene diferentes preferencias [15].

c. Técnica de filtrado híbrido

Este tipo de filtrado combina las técnicas anteriores para conseguir una mejor optimización del sistema previniendo algunas limitaciones y dificultades de los sistemas de recomendación puros. La combinación de estas perspectivas se realiza de la siguiente manera: implementación separada de algoritmos y combinación del resultado, creando un sistema de recomendación unificado que brinde ambos puntos de vista [15].

2. Google Maps

Es un servicio de mapas al que se accede por diferentes entornos, donde según la ubicación geográfica, se pueden visualizar los mapas básicos o personalizados con su ubicación, información sobre negocios locales, datos de contacto e indicaciones sobre cómo llegar hasta el destino de cada uno de ellos [16]. A continuación, se consideran algunas de sus principales herramientas:

2.1. Mapas

Google Maps brinda experiencias simples y personalizadas para aproximar al mundo real a los usuarios mediante mapas dinámicos y estáticos, imágenes de Street View y vistas en 360°, por lo que proporciona unas APIS compatibles. [17]

Entre ellas son: API de inserción de mapas, API de mapas estáticos, API de mapas en JavaScript, API de publicación de Street View, el API estático de Street View, SDK de mapas para Android y SDK de mapas para iOS

2.1.1. Rutas

Google Maps brinda a los usuarios la mejor forma de dirigirse de un lugar a otro con indicaciones de calidad y actualizaciones en tiempo real. Por consiguiente, establece la ruta que recorre un vehículo para crear caminos más precisos [17]. Entre las APIS compatibles son: API de Instrucciones, API de Matriz de distancia y API de carreteras

2.1.2. Lugares

Google Maps apoya a los usuarios a encontrar el mundo con datos de ubicación completos sobre más de 100 millones de direcciones [17].

Las APIS compatibles son: API de lugares, API de geo codificación, API de geolocalización, API de mapas de elevación y el API de zona horaria

2.1.3. Juegos

Las APIS compatibles son: API de ubicaciones jugables y el API de mosaico semántico.

Materiales y métodos

Se debe explicar cómo se ejecutó la investigación: las muestras, técnicas planteadas, métodos de recolección, procedimientos y análisis empleados. Este apartado se redacta en tiempo pasado.

El tipo de investigación es aplicada ya que se construyó una aplicación web móvil para gestionar la información de los técnicos. El nivel de investigación fue preexperimental porque su grado de control es mínimo; por lo tanto, de acuerdo con la investigación que se desarrolló y según los estudios realizados se determinó que la investigación tiene un diseño Posprueba con un solo grupo, ya que fueron comprobado los resultados después de la implementación.

El diseño se diagrama como sigue:

G1 X O1

Donde:

G = Grupo de usuarios de Chiclayo

X = Aplicación web móvil integrando un sistema de recomendación

O1 = Nivel de percepción sobre la facilidad de uso y la utilidad de la herramienta

La población objeto del estudio estuvo constituida por:

Población 1 = 15 749 técnicos usuarios

La muestra se ha obtenido haciendo uso de la fórmula del *procedimiento para estimar el tamaño de la muestra representativa para una población finita* definida por Bernal [18].

$$n = \frac{Z^2 P Q N}{E^2 (N - 1) + Z^2 P Q}$$

La muestra representativa de la población 1 es de 375 técnicos.

Debido a que la población y muestra es extensa, se aplicó el muestreo no probabilístico de tipo casual o fortuito porque se analizaron a los técnicos voluntarios que se obtuvieron de forma casual [19], obteniéndose 21 técnicos en las redes sociales para medir el nivel de percepción sobre la facilidad de uso y la utilidad de la herramienta. Así mismo los 375 individuos que se tomaron en la muestra probabilística sirven como indicador para la cantidad de usuarios técnicos que soporta el sistema.

Los criterios de selección para la población de los usuarios técnicos se tomaron en cuenta según las estadísticas del ministerio de trabajo y promoción del empleo, lo cual en el año 2018 muestra que existen 15 749 trabajadores que prestan servicios tanto en el sector privado como en el sector público [4].

Las variables que se han utilizado como elementos básicos en el desarrollo de la hipótesis son la variable independiente siendo la aplicación web móvil y la variable dependiente fue los servicios técnicos particulares

Resultados y discusión

Para el desarrollo del producto se muestran las actividades más relevantes que se efectuaron en las diferentes fases de la metodología RUP.

Iteración #2: Modelado de Negocio

Se realizó el diagrama de caso de uso de negocio de servicio técnico (ver Fig. 1) y diagrama de caso de actividad de negocio Ofertar soluciones técnicas (ver Fig. 2)

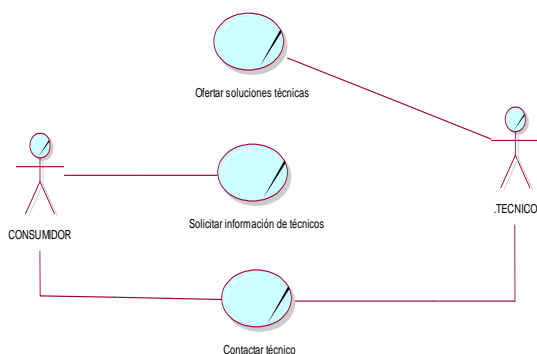


Fig. 1. Diagrama de caso de usos de negocio servicio técnico técnicas

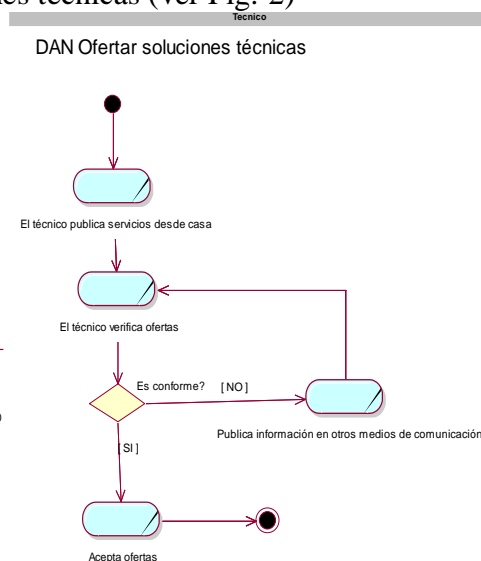


Fig. 2. Diagrama de actividad de negocio oferta soluciones

Iteración #3: Análisis Preliminar de Requerimientos

Modelado de casos de uso de Técnicos (ver Fig. 3) y diagrama de actividad gestionar eventos (ver Fig. 4).

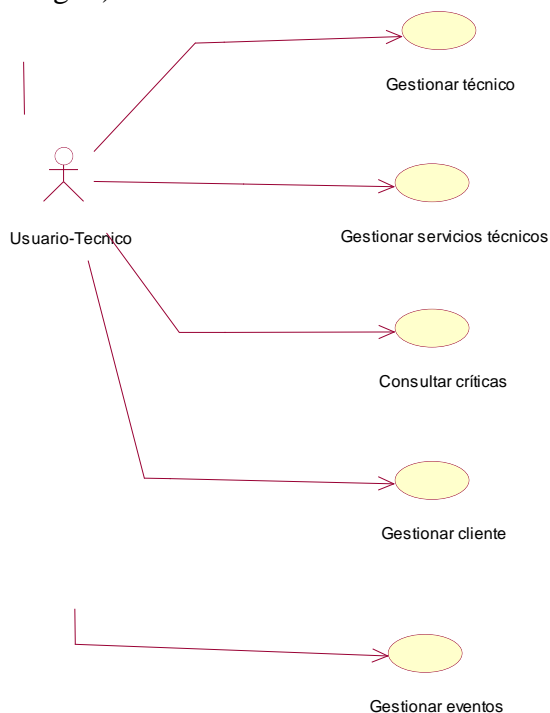


Fig. 3. Diagrama de modelo de casos de uso de técnicos

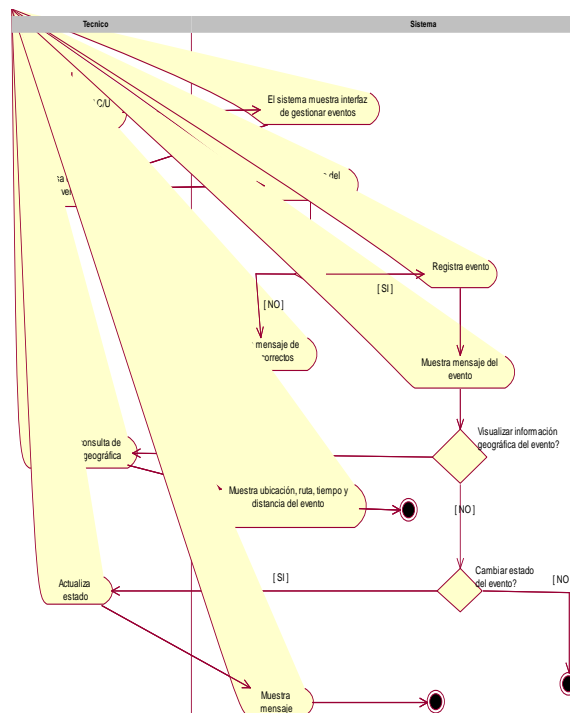


Fig. 4. Diagrama de actividad gestionar eventos

Modelado de casos de uso de Consumidor (ver Fig. 5), diagrama de actividad ubicación de técnicos (ver Fig. 6) y diagrama de actividad criticar técnico (ver Fig. 7)

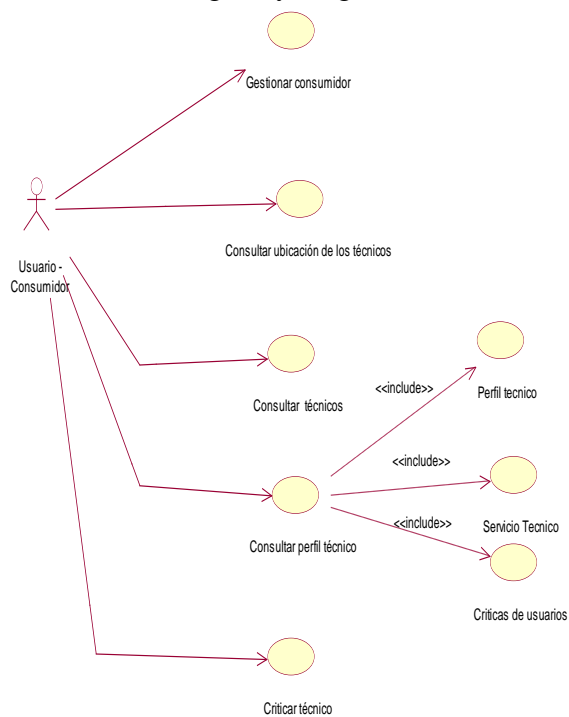


Fig. 5. Modelo de casos de uso de Consumidor

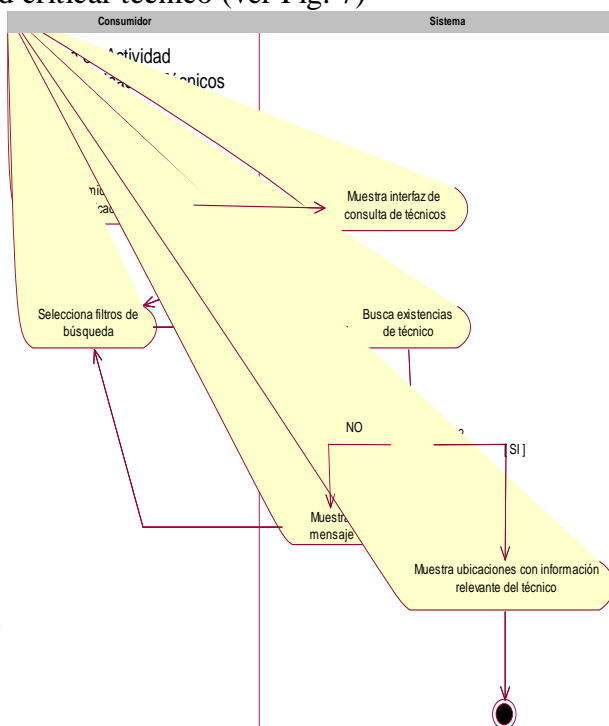


Fig. 6. Diagrama de actividad ubicación de técnicos

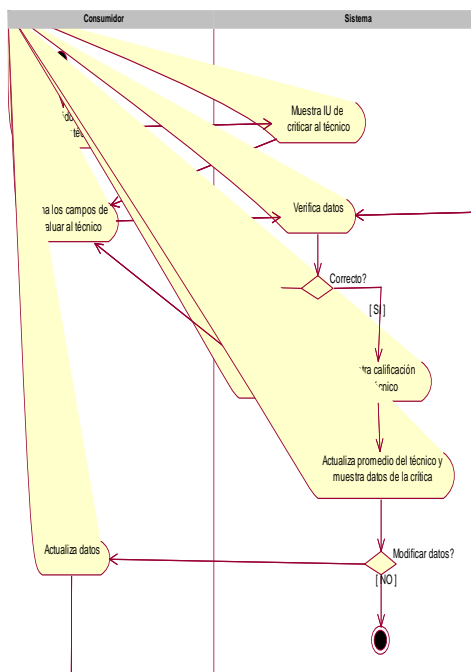


Fig. 7. Diagrama de actividad criticar técnico

Iteración #4: Análisis

Diagrama de realizaciones de casos de uso de análisis del técnico (ver Fig. 8) y diagrama de clases del análisis gestionar eventos (ver Fig. 9)

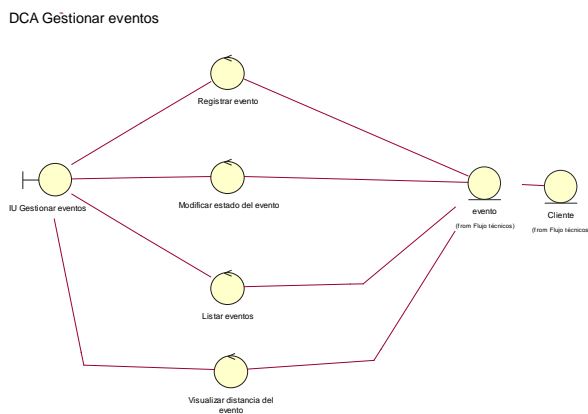
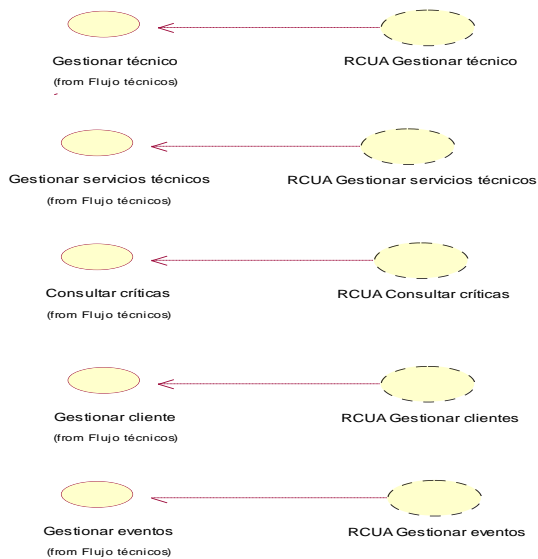


Fig. 8. Diagrama de realizaciones de casos de uso de análisis del técnico Fig. 9. Diagrama de clases del análisis gestionar eventos

Diagrama de realizaciones de casos de uso de análisis del consumidor (ver Fig. 10), diagrama de clases de análisis consultar ubicación de los técnicos (ver Fig. 11) y diagrama de clases de análisis criticar técnico (ver Fig. 12)

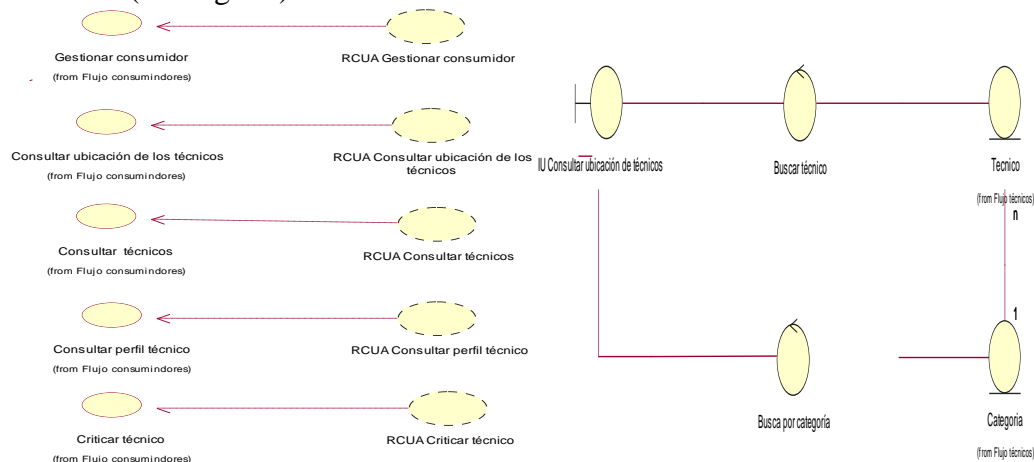


Fig. 10. Diagrama de realizaciones de casos de uso de análisis del consumidor Fig. 11. Diagrama de clases de análisis consultar ubicación de los técnicos

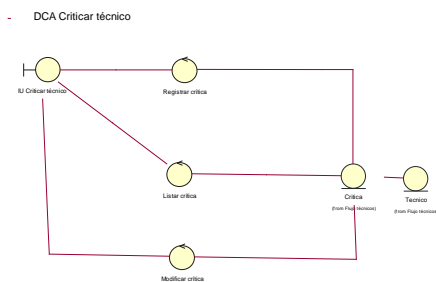


Fig. 12. Diagrama de clases de análisis criticar técnico

Iteración #5: Diseño

Diagrama de realizaciones de casos de uso de diseño técnicos (ver Fig. 13) y diagrama de secuencia Gestionar eventos (ver Fig. 14).

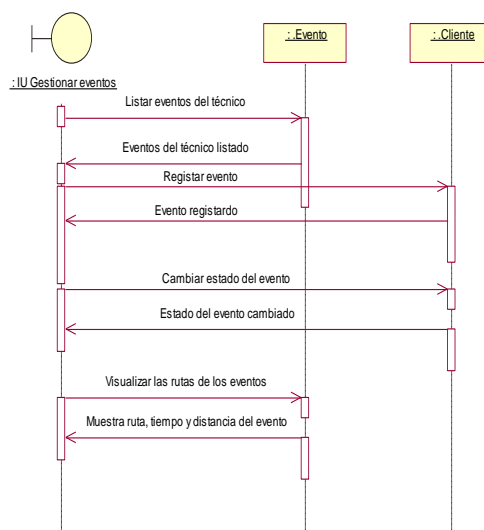
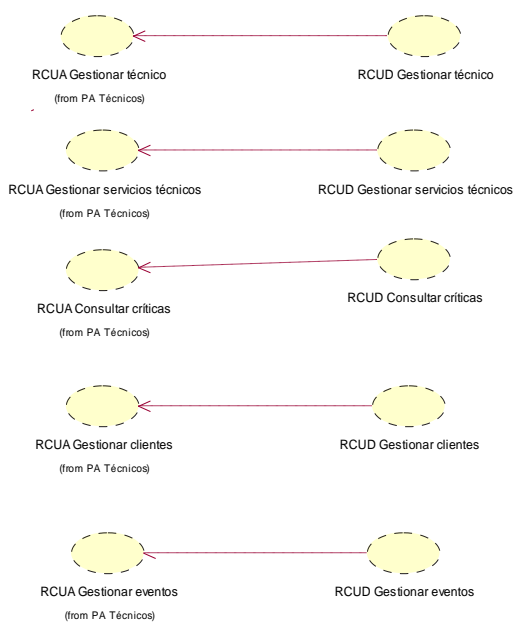


Fig. 13. Diagrama de realizaciones de casos de uso de diseño técnicos

Fig. 14. Diagrama de secuencia gestionar servicios técnicos

Diagrama de realizaciones de casos de uso de diseño consumidores (ver Fig. 15) y diagrama de secuencia diseño consultar ubicación de los técnicos (ver Fig. 16) y diagrama de secuencia diseño Criticar técnico (ver Fig. 17).

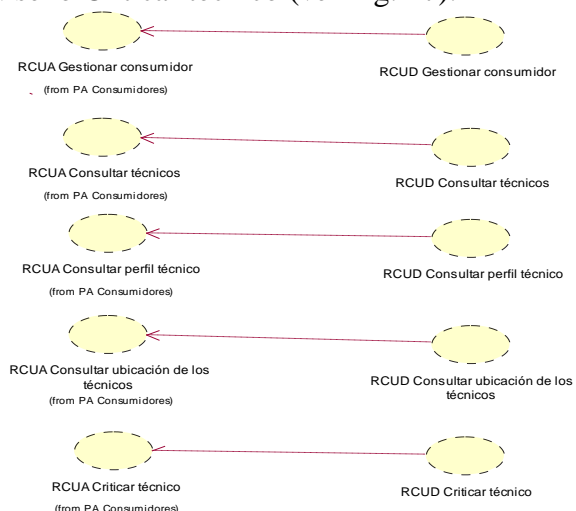


Fig. 15. Diagrama de realización de casos de uso de diseño consumidores

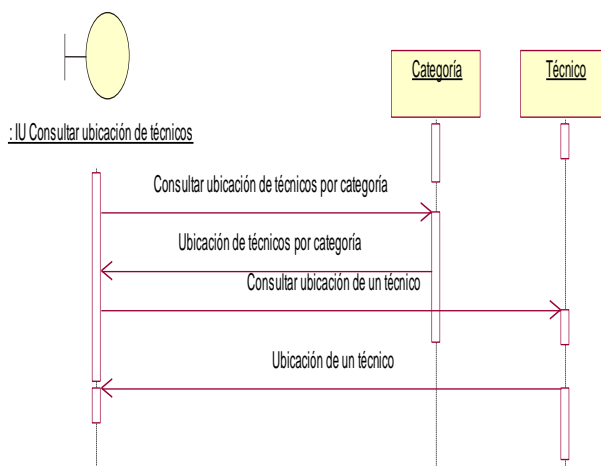


Fig. 16. Diagrama de secuencia de diseño consultar ubicación de los técnicos

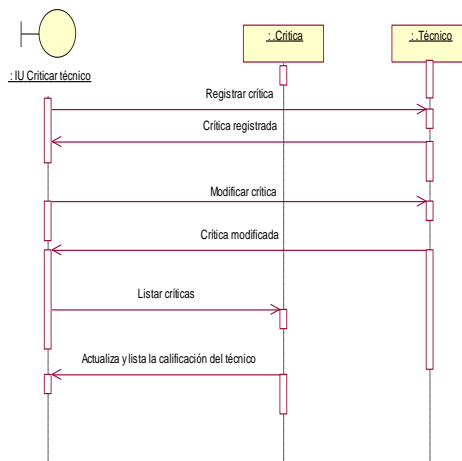


Fig. 17. Diagrama de secuencia de diseño criticar técnico

Diseño de base de datos del sistema (ver Fig. 18).

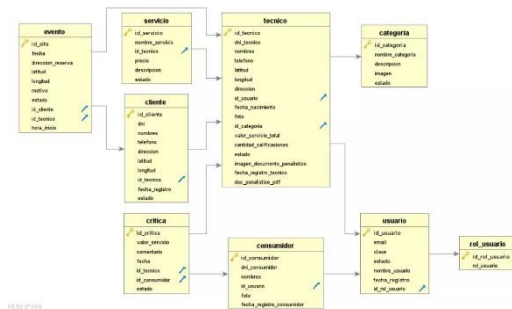


Fig. 18. Diseño físico de la base de datos

Diagramas de estado ubicación de técnicos (ver Fig. 19) y diagrama de estado criticar técnico (ver Fig. 20).

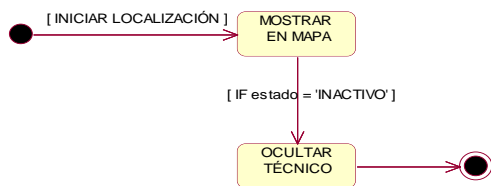


Fig. 19. Diagrama de estados ubicación de técnicos

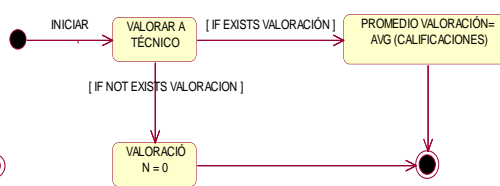


Fig. 20. Diagrama de estados criticar técnico

Diseño de interfaces

- Ubicación de los técnicos (Cliente):

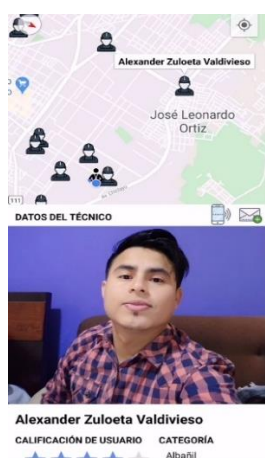


Fig. 21. Interfaz de ubicaciones de los técnicos – GEOTEC

- Visualizar ruta del evento

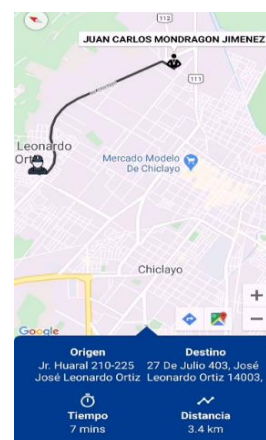


Fig.22. Interfaz de ruta del evento del técnico – GEOTEC

✓ Gestionar críticas (Cliente)



Fig. 23. Gestionar críticas – GEOTEC

Diagrama de despliegue

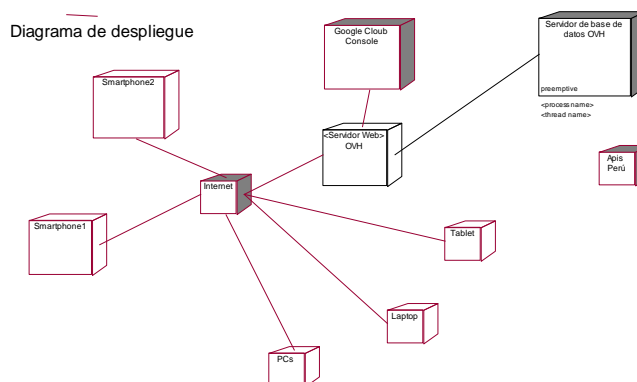


Fig. 24. Diagrama de despliegue – GEOTEC

En la presente arquitectura del sistema el servidor de base de datos enviará información mediante la petición de los demás usuarios. Así mismo, el servidor web procesará una aplicación (GEOTEC) para proveer datos donde los usuarios puedan usar como información relevante para realizar distintas acciones. Esto estará en una red que permite conectar a técnicos y clientes mediante sus dispositivos de forma global.

Apis Perú es una entidad peruana que ofrece Apis que se integren sencillamente en los sistemas de información como consultar datos de la reniec. Así mismo, la entidad de Google que provee Apis para integrar en los sistemas de información, como las Apis de geolocalización.

Smartphones

Al contar con Google Play permitirá descargar la aplicación para localizar los técnicos

Laptops, computadoras o tablets

Al contar con estos dispositivos deberá ingresar a la aplicación web para gestionar los servicios técnicos particulares

Iteración #6: Implementación y Prueba

i) Diagrama de componentes

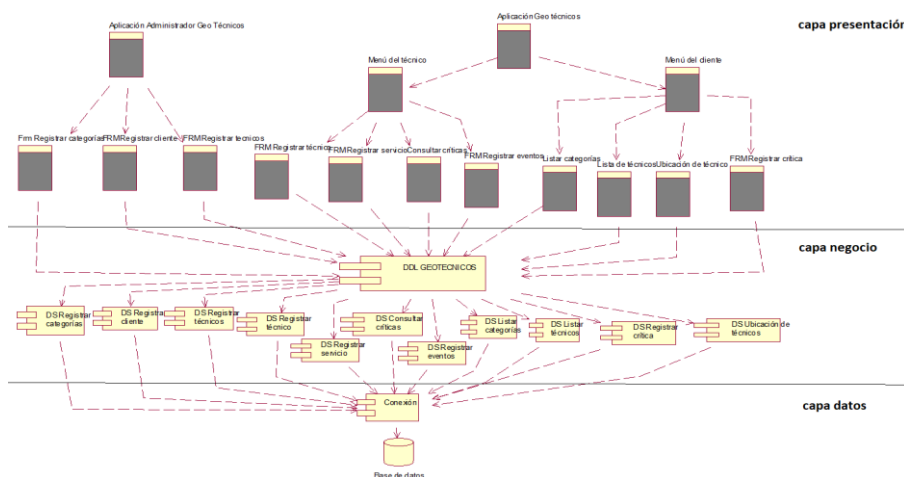


Fig. 25. Diagrama de componente – GEOTEC

Pseudocódigo

- ✓ Sistema de recomendación filtrado colaborativo basado en el usuario.
- Se optó por crear una función en el gestor de base de datos PostgreSQL para generar un sistema de recomendación de filtrado colaborativo basado en el usuario

```

-----Función registrar calificación tecnico-----
CREATE OR REPLACE FUNCTION fn_registrar_calificacion_Tecnico(fnnombreusuario character varying,fnvalor_servicio real,fncomentario character varying,fnid_tecnico int)
RETURNS boolean AS $$
Declare
codigo_consumidor int; codigo_critica int;cantidad_calificaciones_totales int;
cantidad_valor_servicio1 int;cantidad_valor_servicio2 int;cantidad_valor_servicio3 int;cantidad_valor_servicio4 int;cantidad_valor_servicio5 int;
Begin
select (c.id_consumidor) into codigo_consumidor from consumidor c inner join usuario u on c.id_usuario=u.id_usuario where nombre_usuario=fnnombreusuario;
select (case when max(id_critica) is null then 0 else max(id_critica) end) +1 into codigo_critica from critica;
insert into critica(id_critica,id_consumidor,valor_servicio,comentario,fecha,id_tecnico)
values(codigo_critica,codigo_consumidor,fnvalor_servicio,fncomentario,current_date,fnid_tecnico);
-----Cantidad de calificaciones por valor de servicios del consumidor -----
select count(valor_servicio) into cantidad_valor_servicio1 from critica cr inner join consumidor c on c.id_consumidor=cr.id_consumidor inner join tecnico t
on cr.id_tecnico=t.id_tecnico where t.id_tecnico=fnid_tecnico and valor_servicio=1;

select count(valor_servicio) into cantidad_valor_servicio2 from critica cr inner join consumidor c on c.id_consumidor=cr.id_consumidor inner join tecnico t
on cr.id_tecnico=t.id_tecnico where t.id_tecnico=fnid_tecnico and valor_servicio=2;

select count(valor_servicio) into cantidad_valor_servicio3 from critica cr inner join consumidor c on c.id_consumidor=cr.id_consumidor inner join tecnico t
on cr.id_tecnico=t.id_tecnico where t.id_tecnico=fnid_tecnico and valor_servicio=3;

select count(valor_servicio) into cantidad_valor_servicio4 from critica cr inner join consumidor c on c.id_consumidor=cr.id_consumidor inner join tecnico t
on cr.id_tecnico=t.id_tecnico where t.id_tecnico=fnid_tecnico and valor_servicio=4;

select count(valor_servicio) into cantidad_valor_servicio5 from critica cr inner join consumidor c on c.id_consumidor=cr.id_consumidor inner join tecnico t
on cr.id_tecnico=t.id_tecnico where t.id_tecnico=fnid_tecnico and valor_servicio=5;

SELECT COALESCE(count(valor_servicio),0) into cantidad_calificaciones_totales from critica c inner join consumidor cr on c.id_consumidor=cr.id_consumidor
inner join tecnico t on t.id_tecnico=c.id_tecnico where t.id_tecnico=fnid_tecnico;

update tecnico set cantidad_calificaciones=cantidad_calificaciones_totales where id_tecnico=fnid_tecnico;
update tecnico set valor_servicio_total=(((cantidad_valor_servicio1*1.0)+(cantidad_valor_servicio2*2.0)+(cantidad_valor_servicio3*3.0)+
(cantidad_valor_servicio4*4.0)+(cantidad_valor_servicio5*5.0))/(cantidad_calificaciones_totales)
where id_tecnico=fnid_tecnico;
Return true;
exception
when others then return false ;
end;
$$ language 'plpgsql';

select*from fn_registrar_calificacion_Tecnico('Jhon2L',1,'pésimo servicio',2);

```

Fig. 26. Función a crear en la base de datos

- Se crea la web service de la función creada en la base de datos

```

public function RegistrarCriticaTecnico()
try {
    $sql="select * from fn_registrar_calificacion_Tecnico(:fnnombreusuario,:fnvalor_servicio,:fncomentario,:fnid_tecnico);";
    $sentencia = $this->dblink->prepare($sql);
    $sentencia -> bindParam(":fnnombreusuario", $this->getNombre_usuario());
    $sentencia -> bindParam(":fnvalor_servicio", $this->getValor_servicio());
    $sentencia -> bindParam(":fncomentario", $this->getComentario());
    $sentencia -> bindParam(":fnid_tecnico", $this->getId_tecnico());
    $sentencia->execute();
    $resultado = $sentencia->fetch();
    return $resultado["fn_registrar_calificacion_Tecnico"];
} catch (Exception $exc)
{
    throw $exc;
}

$token = $_POST["token"];
if(!isset($_POST["token"])){
    Funciones::imprimeJSON(500, "Falta el token", "");
    exit();
}

try {
    require_once 'token.validar.php';

    if(validarToken($token)){//pregunta si el token es válido
        $objCritica = new critica();
        $objCritica->setNombre_usuario($_POST["fnnombreusuario"]);
        $objCritica->setValor_servicio($_POST["fnvalor_servicio"]);
        $objCritica->setComentario($_POST["fncomentario"]);
        $objCritica->setId_tecnico($_POST["fnid_tecnico"]);
        $resultado=$objCritica->RegistrarCriticaTecnico();
        if($resultado)
        {
            // Funciones::imprimeJSON(200, "Técnico registrado correctamente", $resultado);
            Funciones::imprimeJSON(500, "Error al registrar calificación", null);
        }
    } else {
        // Funciones::imprimeJSON(500, "Error al registrar técnico, usuario ya registrado", null);
        Funciones::imprimeJSON(200, "Calificación registrada correctamente", $resultado);
    }
}

} catch (Exception $exc) {
    Funciones::imprimeJSON(500, $exc->getMessage(), "");
}

```

Se ejecuta la
función creada

Fig. 27. Web service de la función creada en la base de datos

- En el Android Studio se llama a la web service para ejecutar las calificaciones que va dejando los usuarios en base a las experiencias por el servicio brindado.

```
private class AgregarCriticaTecnico extends AsyncTask<Void, Void, String> {
    ProgressDialog pDialog;
    Critica objcritica;
    public AgregarCriticaTecnico(Critica objcritica) {
        this.objcritica = objcritica;
    }
    @Override
    protected void onPreExecute() {
        pDialog = new ProgressDialog( context: CriticaTecnicoAgregarEditar.this);
        pDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        pDialog.setMessage("Grabando datos...");
        pDialog.setCancelable(false);
        pDialog.show();
        super.onPreExecute();
    }
    @Override
    protected String doInBackground(Void... params) {
        String resultado="";
        try {
            String urlListado = ServiciosWeb.URL_WS + "critica.registrarTecnico.php";
            HashMap parametros = new HashMap<String, String>();
            parametros.put("token", Sesion.TOKEN);
            parametros.put("fnombreusuario", Sesion.Usuario);
            parametros.put("fnvalor_servicio", String.valueOf(objcritica.getValorServicio()));
            parametros.put("fncomentario", objcritica.getComentario());
            parametros.put("fnid_tecnico", String.valueOf(AdaptadorTecnico.id_tecnico));

            resultado = new ServiciosWeb().openWebService(urlListado, parametros);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return resultado;
    }
    @Override
    protected void onPostExecute(String resultado) {
        if (!resultado.isEmpty()) {
            try {
                JSONObject json = new JSONObject(resultado);
                String mensaje = json.getString( name: "mensaje");
                Toast.makeText( context: CriticaTecnicoAgregarEditar.this, mensaje, Toast.LENGTH_LONG).show();
                pDialog.dismiss();
                CriticaTecnicoAgregarEditar.this.finish();
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else {
            Toast.makeText( context: CriticaTecnicoAgregarEditar.this, text: "Ha ocurrido un error al cargar los datos de la web service +12", Toast.LENGTH_SHORT).show();
        }
    }
}
```



Fig. 28. Ejecución de la web services

- ✓ Pseudocódigo de las APIS del proyecto GEOTEC.
Para hacer uso de las apis de geolocalización se debe crear la credencial de la API.



Fig. 29. Credencial de la API creada

- APIS Seleccionadas en la aplicación web:

Para cargar el mapa en la web se debe tener habilitada la API de javascript

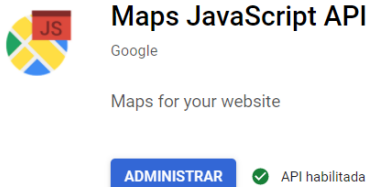


Fig. 30. Api de JavaScript

Ubicación de los técnicos locales

Consulta del lugar del trabajo de los técnicos locales.

```
<?php
class localizarTecnicos {
    private $id_tecnico;
    private $nombres;
    private $direccion;
    private $latitud;
    private $longitud;
    private $foto;
    //private $tableName = "tecnico";
    function setId_tecnico($id_tecnico) { $this->id_tecnico = $id_tecnico; }
    function getId_tecnico() { return $this->id_tecnico; }
    function setNombres($nombres) { $this->nombres = $nombres; }
    function getNombres() { return $this->nombres; }
    function setDireccion($direccion) { $this->direccion = $direccion; }
    function getDireccion() { return $this->direccion; }
    function setLatitud($latitud) { $this->latitud = $latitud; }
    function getLatitud() { return $this->latitud; }
    function setLongitud($longitud) { $this->longitud = $longitud; }
    function getLongitud() { return $this->longitud; }
    function setFoto($foto) { $this->foto = $foto; }
    function getFoto() { return $this->foto; }

    public function __construct() {
        require_once "conexion.php";
        $link = new Conexion;
        $this->link = $link->conectar();
    }

    public function getAllTecnicos() {
        $sql = "SELECT * FROM tecnico WHERE estado='A' and latitud IS NOT NULL AND longitud IS NOT NULL";
        $stmt = $this->link->prepare($sql);
        $stmt->execute();
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
}
}
```

Fig. 31. Ubicación de los técnicos locales

Se llama a la consulta creada `getAllTecnicos` para ejecutar la ubicación del local de cada técnico usando la API de javascript para así mismo cargar el mapa en base a la latitud y longitud.

```
<div class="panel-body">
    <div class="box-header with-border">
        <div class="row">
            <h3 class="box-title">Ubicación de los técnicos</h3>
            <div id="map"></div>
        </div>
        <?php
        require_once 'modelos/localizarTecnicos.php';
        $localizarTecnicos = new LocalizarTecnicos;

        $datosTecnicos = $localizarTecnicos->getAllTecnicos();
        $datosTecnicos = json_encode($datosTecnicos, true);

        echo '<div id="data">' . $datosTecnicos . '</div>';

        echo '<div id="allData">' . $datosTecnicos . '</div>';
        ?>

        <!--The div element for the map -->

        <script async defer
        src="https://maps.googleapis.com/maps/api/js?key=AIzaSyB2zIMA07_1-deZZfTNET2etpEhRcEQrsE&callback=loadMap">
        </script>

    </div>
</div>
```




Fig. 32. Consulta `getAllTecnicos`

En el javascript se carga el mapa la ubicación y una ventana de información(showAllTecnicos) de cada técnico de la consulta ejecutada.

```

var map;
var geocoder;
function loadMap() {
    var pune = {lat: -6.7719709, lng: -79.8374808};
    map = new google.maps.Map(document.getElementById('map'), {
        zoom: 13.5,
        center: pune
    });
    var cdata = JSON.parse(document.getElementById('data').innerHTML);
    geocoder = new google.maps.Geocoder();
    //codeAddress(cdata);
    var allData = JSON.parse(document.getElementById('allData').innerHTML);
    showAllTecnicos(allData)
}
function showAllTecnicos(allData) {
    var infoWind = new google.maps.InfoWindow;
    Array.prototype.forEach.call(allData, function(data){
        var content = document.createElement('div');
        var strong = document.createElement('strong');
        var p = document.createElement('p');
        var em = document.createElement('em');
        strong.textContent =(data.nombres) + ". Nro de celular: " +(data.telefono) ;
        content.appendChild(strong);
        p.textContent =(data.direccion);
        content.appendChild(p);
        var img = document.createElement('img');
        img.src = 'data:image/png;base64, '++(data.foto);
        img.style.height = '100px';
        content.appendChild(img);
        var marker = new google.maps.Marker({
            position: new google.maps.LatLng(data.latitud, data.longitud),
            icon: '../control_geo_tecnicos/vistas/img/plantilla/tecnico.png',
            map: map
        });
        if(window.matchMedia("(max-width:767px)").matches){//solo al pasar el mouse 'mouseover'
            marker.addListener('click', function(){
                infoWind.setContent(content);
                infoWind.open(map, marker);
            })
        }else{
            marker.addListener('mouseover', function(){
                infoWind.setContent(content);
                infoWind.open(map, marker);
            })
        }
    })
}
}

```

Fig. 33. Mapa de ubicación de los técnicos

- Capturar coordenadas geográficas de la ubicación del técnico (Local)

Para descifrar las coordenadas de un lugar se debe tener habilitada la API de geocodificación.



Fig. 34. Api de geocodificación

En el controlador se llama al caso “Consulta_API_GEOCODER”, donde se envía la dirección desde javascript para capturar las coordenadas geográficas (latitud y longitud). En tal sentido, el consumidor pueda localizar posteriormente el local de cada técnico.

```
<?php
$action = $_POST["accion"];
controlador($accion);
function controlador($accion){
    switch ($accion) {
        case 'CONSULTA_DNI_REGISTRAR':
            $dni = $_POST['Dni'];
            // $ws = "https://dni.optimizperu.com/api/persons/" . $dni . "?format=json";
            $ws = "https://dni.optimizperu.com/api/persons/" . $dni . "?format=json";
            $header = array();
            $ch = curl_init();
            curl_setopt($ch,CURLOPT_SSL_VERIFYPEER,1);
            curl_setopt($ch,CURLOPT_URL,$ws);
            curl_setopt($ch,CURLOPT_RETURNTRANSFER,true);
            curl_setopt($ch,CURLOPT_HTTPAUTH,CURLAUTH_ANY);
            curl_setopt($ch,CURLOPT_TIMEOUT,30);
            curl_setopt($ch,CURLOPT_HTTPHEADER,$header);
            //para ejecutar los procesos de forma local en windows
            //enlace de descarga del cacert.pem https://curl.haxx.se/docs/caextract.html
            curl_setopt($ch,CURLOPT_CAINFO,dirname(__FILE__)."/../cacert.pem");
            $datos = curl_exec($ch);
            curl_close($ch);
            echo $datos;
            break;
        case 'CONSULTA_API_GEOCODER':
            $direccion = urldecode($_POST['Direccion']);
            $ws = "https://maps.googleapis.com/maps/api/geocode/json?address=".$direccion."&key=AIzaSyBz2IMa07_1-deZZfTNET2etpEhRCeQsE&callback=";
            $result = file_get_contents($ws);
            echo ($result);
            //echo "<br />";
            //echo $long = $response_a->results[0]->geometry->location->lng;
            //echo json_encode($datos);
            break;
        default:
            # code...
            break;
    }
}
```

Fig. 35. Consulta api GEOCODER

Función que retorna la latitud, longitud y localización en base a la dirección que se envió al controlador

```
function localizar(elemento,direccion) {
    var geocoder = new google.maps.Geocoder();
    var map = new google.maps.Map(document.getElementById(elemento), {
        zoom: 18,
        scrollwheel: true,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    });
    $.ajax({
        method: "POST",
        url: 'controladores/controlador.php',
        data: {
            "accion": "CONSULTA_API_GEOCODER",
            "Direccion": $("#direccion").val()
        },
        success: function(respuesta) {
            json = JSON.parse(respuesta);
            console.log("estado",json["status"]);
            var estado=json["status"];
            if(estado=="OK"){
                $("#latitud").val(json["results"][0]["geometry"]["location"]["lat"]);
                $("#longitud").val(json["results"][0]["geometry"]["location"]["lng"]);
                map.setCenter(json["results"][0]["geometry"]["location"]);
                var marker = new google.maps.Marker({
                    map: map,
                    position: json["results"][0]["geometry"]["location"]
                });
            }
            else {
                var mensajeError = "";
                if (estado === "ZERO_RESULTS") {
                    mensajeError = "No hubo resultados para la dirección ingresada.";
                } else if (estado === "OVER_QUERY_LIMIT" || estado === "REQUEST_DENIED" || estado === "UNKNOWN_ERROR") {
                    mensajeError = "Error general del mapa.";
                } else if (estado === "INVALID_REQUEST") {
                    mensajeError = "Error de la web. Contacte con Name Agency.";
                }
                $("#direccion").val("");
                $("#latitud").val("");
                $("#longitud").val("");
                swal({
                    title: "Error ",
                    text: mensajeError,
                    type: "error",
                    confirmButtonText: "¡Cerrar!"
                });
            }
        }
    });
}
```

Fig. 36. Función GEOCODER

Se ejecuta la función localizar para cargar en el mapa la latitud y longitud de la dirección ingresada.

```
$.getScript("https://maps.googleapis.com/maps/api/js?key=AIzaSyAzbJzKp2uPSc0i-DfEqGnflg9mR_0pkTQ&callback=loadMap", function() {
    $("#buscar").click(function() {
        var direccion = $("#direccion").val();
        if (direccion !== "") {
            localizar("mapa-geocoder", direccion);
        }
    });
});
```

Fig. 37. Función ejecutada

- Android Studio

APIS Seleccionadas en la aplicación móvil:

Para hacer uso de la geolocalización en Android studio accedemos al enlace generado xml de google_maps_api.xml y agregamos la credencial creada en Google cloud console

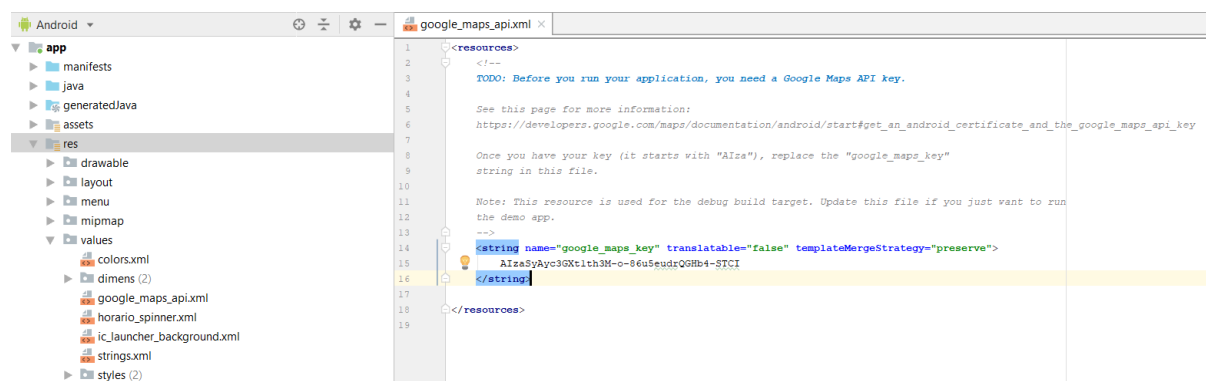


Fig. 38. Api de Maps en la aplicación móvil

Para cargar las direcciones dinámicamente en el mapa de la aplicación móvil se debe tener habilitada la API de Maps SDK para Android

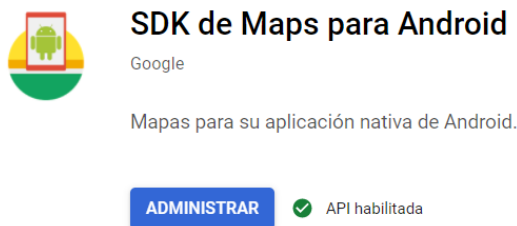


Fig. 39. SDK de Maps para Android

- Ubicación de los técnicos locales en la aplicación de Android Studio

Se crea la web services de la ubicación de los técnicos

```
public function listarTécnicosMapa() {
    try {
        $sql="select t.id_técnico, t.nombres,t.telefono,t.direccion,t.latitud,t.longitud,t.id_categoria,c.nombre_categoria,u.email,
            t.valor_servicio_total,t.foto from tecnico t inner join usuario u on t.id_usuario=u.id_usuario
            inner join categoria c on c.id_categoria=t.id_categoria where t.estado='A'";
        $sentencia = $this->dblink->prepare($sql);
        $sentencia->execute();
        $resultado = $sentencia->fetchAll(PDO::FETCH_ASSOC);
        RETURN $resultado;
    } catch (Exception $exc) {
        throw $exc;
    }
}

if(!isset($_POST["token"])){
    Funciones::imprimeJSON(500, "Falta el token", "");
    exit();
}

$token = $_POST["token"];
try {
    //validar el token
    require_once 'token.validar.php';
    if(validarToken($token)){//pregunta si el token es válido
        $objTecnico=new tecnico();
        $resultado=$objTecnico->listarTécnicosMapa();
        Funciones::imprimeJSON(200, "", $resultado);
    }
} catch (Exception $exc) {
    Funciones::imprimeJSON(500, $exc->getMessage(), "");
}
}
```

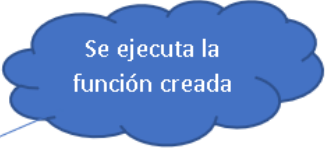


Fig. 40. Función creada de los técnicos en el mapa

En el Android Studio se inicializa los componentes de la interfaz del mapa de los técnicos con los datos específicos para cada técnico en el método onCreate

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // Inflate the layout for this fragment
    setContentView(R.layout.fragment_lista_puntos_técnicos);

    // Obtain the SupportMapFragment and get notified when the map is ready to be used.

    SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
        .findFragmentById(R.id.map);
    mapFragment.getMapAsync(new OnMapReadyCallback() {

        @Override
        public void onMapReady() {
            View headerLayout1 = findViewById(R.id.bottom$soft);
            imgmarker = (ImageView) findViewById(R.id.imgMarker);

            txtnombre_tecnico = (TextView) findViewById(R.id.txtNombreTecnicoPuntos);
            txtDireccion = (TextView) findViewById(R.id.txtDireccionPuntos);
            txtCategoria = (TextView) findViewById(R.id.txtCategoriaPuntos);
            rtValorServicioTotalPuntos = (RatingBar) findViewById(R.id.rtValorServicioTotalPuntos);
            tapactionlayout = (LinearLayout) findViewById(R.id.tap_action_layout);
            bottomSheet = findViewById(R.id.bottom$soft);
            mBottomSheetBehavior1 = BottomSheetBehavior.from(bottomSheet);
            mBottomSheetBehavior1.setPeekHeight(120);
            mBottomSheetBehavior1.setState(BottomSheetBehavior.STATE_COLLAPSED);
            redireccionar = (ImageView) findViewById(R.id.redireccionar);
            redireccionar.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    llamar();
                }
            });
        }
    });
}
```

Fig. 41. Método OnCreate

Se crea un método “ObtenerPuntos()” para hacer petición a la web services creada. Así mismo dentro de ese método se ejecuta el método “cargarPuntosAMapa” para capturar la ubicación de los técnicos provenientes de la web services.

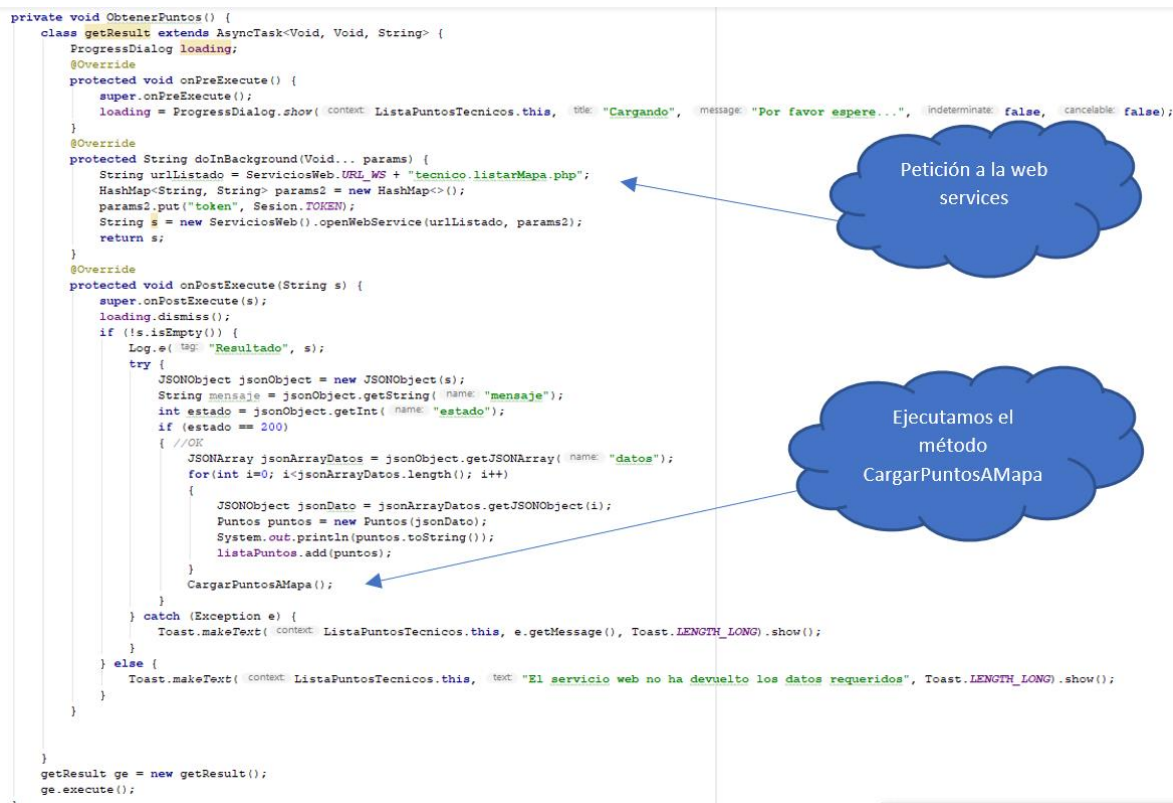


Fig. 42. Método Obtener puntos

En el método “cargarPuntosAMapa()” se adiciona los marcadores a cada ubicación de un técnico en base a la latitud y longitud obtenida de la web services.



Fig. 43. Método Cargar puntos mapa

En el método `onMarkerClick`, se carga la información obtenida de la web services de cada técnico seleccionado.

```

@Override
public boolean onMarkerClick(Marker marker) {
    try {
        Integer clickCount = (Integer) marker.getTag();
        return true;
    } catch (Exception ex)
    {
        Puntos info = new Puntos();
        info = (Puntos) marker.getTag();

        System.out.println(info.toString());
        nombre_punto=info.getNombres();
        foto_punto=info.getFoto();
        telefono_punto=info.getTelefono();
        imgmarker.setImageBitmap(Funciones.base64ToImage(info.getFoto()));
        txtnombre_tecnico.setText(info.getNombres());
        txtDireccion.setText(info.getDireccion());
        txtCategoria.setText(info.getNombreCategoria());
        rtValorServicioTotalPuntos.setRating((float)info.getvalorServicioTotal());

        mBottomSheetBehavior1.setState(BottomSheetBehavior.STATE_EXPANDED);
    }
    return false;
}

```

Fig. 44. Método `onMarkerClick`

En el `onMapReady` se llama al método “ObtenerPuntos()” para inicializar el mapa, donde se carga la ubicación de los técnicos con su respectiva información.

```

@Override
public void onMapReady(GoogleMap googleMap) {

    mMap = googleMap;
    mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
    mMap.getUiSettings().setZoomControlsEnabled(true);

    mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(1000);
    mLocationRequest.setFastestInterval(1000);
    mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    mLocationRequest.setSmallestDisplacement(5);

    startLocation();
    ObtenerPuntos();
    //
}

```

Fig. 45. Método `onMapReady`

Para predecir las direcciones ingresadas en la aplicación móvil se debe tener habilitada la API de lugares



API de lugares

Google

Obtenga información detallada sobre 100 millones de lugares

ADMINISTRAR

✓ API habilitada

Fig. 46. Interfaz de api de lugares

En el gradle se agrega la dependencia de los lugares de la API

```
implementation 'com.google.android.libraries.places:places:1.1.0'
```

Fig. 47. Gradle

En el diseño se agrega el componente Autocomplete

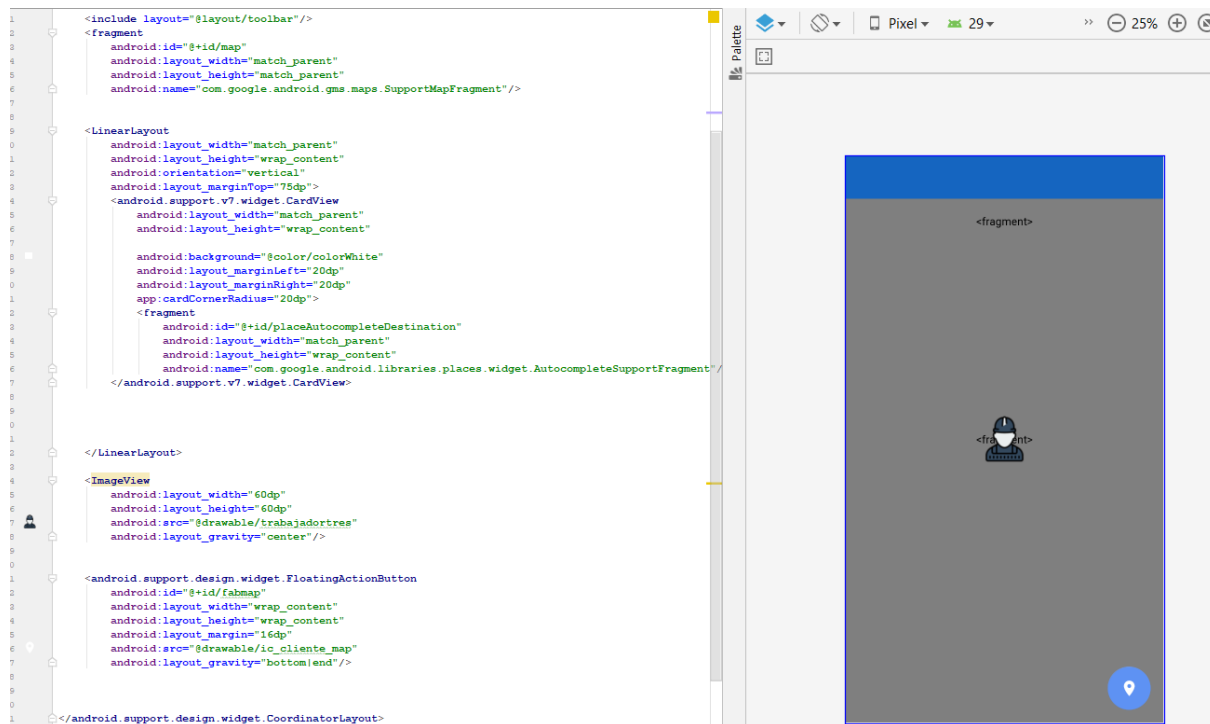


Fig. 48. Componente autocomplete

Inicializamos las propiedades

```

Toolbar mToolBar;
private GoogleMap mMap;
private SupportMapFragment mMapFragment;

FloatingActionButton fabMap;
private LocationRequest mLocationRequest;
private FusedLocationProviderClient mFusedLocation;

private final static int LOCATION_REQUEST_CODE = 1;
private final static int SETTINGS_REQUEST_CODE = 2;
private Marker mMarker;

private boolean mIsFirstTime = true;

private LatLng mCurrentLatLng;

private PlacesClient mPlaces;
private AutocompleteSupportFragment mAutocompleteDestination;

private String mDestination;
private LatLng mDestinationLatLng;

private GoogleMap.OnCameraIdleListener mCameraListener;

```

Fig. 49. Inicialización de las propiedades

En el onCreate inicializamos

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_mapa_cliente);
    mToolBar = findViewById(R.id.toolbar);
    setSupportActionBar(mToolBar);
    getSupportActionBar().setTitle("Selecciona dirección");
    mFusedLocation = LocationServices.getFusedLocationProviderClient(this);
    fabMap = (FloatingActionButton) findViewById(R.id.fabmap);
    fabMap.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            obtenerinformacion();
        }
    });
}

mMapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);
mMapFragment.getMapAsync(new OnMapReadyCallback() {
    @Override
    public void onMapReady() {
        if (!Places.isInitialized()) {
            Places.initialize(getApplicationContext(), "AIzaSyAyc3GXt1th3M-o-86u5eudrQGhb4-STCI");
        }
        mPlaces = Places.createClient(this);
        instanceAutocompleteOrigin();
        onCameraMove();
    }
});
```

Latitud, longitud
y dirección
obtenida del
Autocomplete de
la API Places

Fig. 50. Inicialización en el onCreate

Información obtenida que se envía a la interfaz de registro del cliente

```
private void obtenerinformacion() {
    if (mDestination != null) {
        Bundle parametros = new Bundle();
        parametros.putDouble("latitud", mDestinationLatLng.latitude);
        parametros.putDouble("longitud", mDestinationLatLng.longitude);
        parametros.putString("direccion", mDestination);

        Intent i = new Intent();
        i.putExtras(parametros);
        this.setResult(RESULT_OK, i);
        this.finish();
    } else {
        Toast.makeText(this, "Debe seleccionar la direccion del cliente", Toast.LENGTH_SHORT).show();
    }
}
}
```

Fig. 51. Registro cliente

LimitSearch(), se creó este método para que autocomplete las direcciones solo de Perú con una distancia de 5km en el norte y 5 km en el sur en base a la ubicación actual (GPS) instanceAutocompleteOrigin, se creó este método para capturar la dirección, latitud y longitud del técnico al seleccionar el autocomplete

```
private void limitSearch() {
    LatLng northSide = SphericalUtil.computeOffset(mCurrentLatLng, distance: 5000, heading: 0);
    LatLng southSide = SphericalUtil.computeOffset(mCurrentLatLng, distance: 5000, heading: 180);
    mAutocompleteDestination.setCountry("PER");
    mAutocompleteDestination.setLocationBias(RectangularBounds.newInstance(southSide, northSide));
}

private void instanceAutocompleteOrigin() {
    mAutocompleteDestination = (AutocompleteSupportFragment) getSupportFragmentManager().findFragmentById(R.id.placeAutocompleteDestination);
    mAutocompleteDestination.setPlaceFields(Arrays.asList(Place.Field.ID, Place.Field.LAT_LNG, Place.Field.NAME));
    mAutocompleteDestination.setHint("Lugar de recogida");
    mAutocompleteDestination.setOnPlaceSelectedListener(new PlaceSelectionListener() {
        @Override
        public void onPlaceSelected(@NonNull Place place) {
            mDestination = place.getName();
            mDestinationLatLng = place.getLatLng();
            Log.d("PLACE", "msg: " + mDestination);
            Log.d("PLACE", "msg: " + mDestinationLatLng.latitude);
            Log.d("PLACE", "msg: " + mDestinationLatLng.longitude);
        }
    });
}

@Override
public void onError(@NonNull Status status) {
}
}
}
}
```

Fig. 52. Métodos LimitSearch y InstanceAutocompleteOrigin

Para calcular la ruta, tiempo y distancia se debe tener habilitada la API de instrucciones.

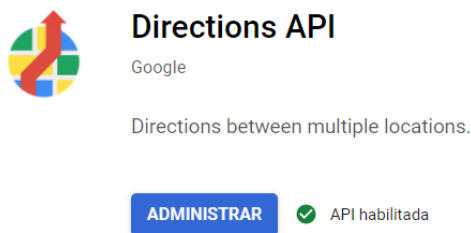


Fig. 53. Api de instrucciones

API de instrucciones

```
public class GoogleApiProvider {

    private Context context;

    public GoogleApiProvider(Context context) {
        this.context = context;
    }

    public Call<String> getDirections(LatLng originLatLng, LatLng destinationLatLng) {
        String baseUrl = "https://maps.googleapis.com";
        String query = "/maps/api/directions/json?mode=driving&transit_routing_preferences=less_driving&"
            + "origin=" + originLatLng.latitude + "," + originLatLng.longitude + "&"
            + "destination=" + destinationLatLng.latitude + "," + destinationLatLng.longitude + "&"
            + "departure_time=" + (new Date().getTime() + (60*60*1000)) + "&"
            + "traffic_model=best_guess&"
            + "key=" + "AIzaSyAyc3Gxt1th3M-o-86u5eudrQGHb4-STCI";
        return RetrofitClient.getClient(baseUrl).create(IGoogleApi.class).getDirections(Uri.parse(baseUrl + query));
    }
}
```

Fig. 54. Clase GoogleApiProvider

Decodificación de la distancia de un origen y destino

```
public class DecodePoints {

    public static List decodePoly(String encoded) {

        List poly = new ArrayList();
        int index = 0, len = encoded.length();
        int lat = 0, lng = 0;

        while (index < len) {
            int b, shift = 0, result = 0;
            do {
                b = encoded.charAt(index++) - 63;
                result |= (b & 0x1f) << shift;
                shift += 5;
            } while (b >= 0x20);
            int dlat = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
            lat += dlat;

            shift = 0;
            result = 0;
            do {
                b = encoded.charAt(index++) - 63;
                result |= (b & 0x1f) << shift;
                shift += 5;
            } while (b >= 0x20);
            int dlng = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
            lng += dlng;

            LatLng p = new LatLng((((double) lat / 1E5)),
                (((double) lng / 1E5)));
            poly.add(p);
        }

        return poly;
    }
}
```

Fig. 55. Decodificación de distancia de origen y destino

Dibujar la línea del origen y destino del técnico y cliente

```
private void drawRoute() {
    mGoogleApiProvider.getDirections(mOriginLatLng, mDestinationLatLng).enqueue(new Callback<String>() {
        @Override
        public void onResponse(Call<String> call, Response<String> response) {
            try {
                JSONObject jsonObject = new JSONObject(response.body());
                JSONArray jsonArray = jsonObject.getJSONArray( name: "routes");
                JSONObject route = jsonArray.getJSONObject( index: 0);
                JSONObject polyLines = route.getJSONObject("overview_polyline");
                String points = polyLines.getString( name: "points");
                mPolylineList = DecodePoints.decodePoly(points);
                mPolylineOptions = new PolylineOptions();
                mPolylineOptions.color(Color.DKGRAY);
                mPolylineOptions.width(13f);
                mPolylineOptions.startCap(new SquareCap());
                mPolylineOptions.jointType(JointType.ROUND);
                mPolylineOptions.addAll(mPolylineList);
                mMap.addPolyline(mPolylineOptions);

                JSONArray legs = route.getJSONArray( name: "legs");
                JSONObject leg = legs.getJSONObject( index: 0);
                JSONObject distance = leg.getJSONObject("distance");
                JSONObject duration = leg.getJSONObject("duration");
                String distanceText = distance.getString( name: "text");
                String durationText = duration.getString( name: "text");
                mTextViewTime.setText(durationText);
                mTextViewDistance.setText(distanceText);
            } catch (Exception e) {
                Log.d( tag: "Error", msg: "Error encontrado " + e.getMessage());
            }
        }
    });
}
```

Fig. 56. DrawRoute

Se carga la distancia del técnico al cliente en el mapa dibujando la ruta en el método `onMapReady()`.

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
    mMap.getUiSettings().setZoomControlsEnabled(true);

    mMap.addMarker(new MarkerOptions().position(mOriginLatLng).title(""+AdaptadorReserva.nombre_cliente).icon(BitmapDescriptorFactory.fromResource(R.drawable.iconoconsumidor)));
    mMap.addMarker(new MarkerOptions().position(mDestinationLatLng).title("Tu").icon(BitmapDescriptorFactory.fromResource(R.drawable.trabajadortres)));

    mMap.animateCamera(CameraUpdateFactory.newCameraPosition(
        new CameraPosition.Builder()
            .target(mOriginLatLng)
            .zoom(14f)
            .build()
    ));

    drawRoute();
}
```

Fig. 57. Método onMapready

- Pruebas

Se realizaron diferentes pruebas de software al aplicativo móvil “GEOTEC” para asegurar la calidad del sistema. Se consideraron las pruebas unitarias, de rendimiento y de usabilidad.

- Pruebas Unitarias:

Se realizó la ejecución de pruebas unitarias del aplicativo móvil para comprobar el correcto funcionamiento de cada unidad de código.

- Pruebas de rendimiento

Se realizaron las pruebas de rendimiento para evaluar el tiempo que le toma en procesar la información en el servidor web, concluyendo que el servidor web procesa la información en tiempos óptimos.

En base a los objetivos de la investigación

- ✓ **Identificar una técnica de sistema de recomendación que permita la similitud de preferencias de usuario.**

Entre las técnicas de sistema de recomendación se determinaron tres: recomendaciones basadas en contenido, de filtrado colaborativo (basado en el usuario y basado en elementos) e híbrida (la cual combina a ambas).

Los sistemas de recomendación que se vieron en un inicio tienen diferentes características según las técnicas de filtrado. En la tabla I se puede ver el análisis sistémico de las tres técnicas mencionadas; suponemos que I es el conjunto de objetos sobre los cuales podemos hacer recomendaciones, U es el conjunto de usuarios cuyas preferencias son conocidas, U es el usuario para el que queremos generar recomendaciones, e I es un objeto para el que nos gustaría predecir las preferencias de U , $I \in I'$ con $I' \subseteq I$.

TABLA I
ANÁLISIS SISTEMICO DE LAS TÉCNICAS DE SISTEMA DE RECOMENDACIÓN

Técnicas		Características			Técnica utilizada
Basados en contenido		Características de los objetos de “ I ”	Valoraciones de objetos de “ I ”	Genera un clasificador que ajusta el comportamiento de las valoraciones de “ U ” y lo usa en “ I ”	
Filtrado colaborativo	Colaborativo basado en usuario	Valoración por parte de “ U ” de los objetos de “ I ”	Valoraciones por parte de “ U ” de los objetos de “ I ”	Identifica usuarios en “ U ” similares a “ U ” y predice la valoración de “ I ” a partir de las valoraciones de esos usuarios.	✓

	Colaborativo basado en elementos	Características de los objetos de I	Una función de utilidad sobre los objetos en I que describen las preferencias de u	Aplica una función a los objetos y determina la ordenación de i	
Hibrido		Valoración por parte de “U” y características de los objetos de “i”	Valoraciones por parte de “U” de los objetos de “I” y valoraciones de los objetos de “I”. Asimismo tiene función de utilidad sobre los objetos en “I” que describen las preferencias de “U”.	Reconoce a los usuarios en “U” semejantes a “U” para luego pronosticar la valoración de “I” a partir de las valoraciones de esos usuarios, produciendo un clasificador que acopla la conducta de las valoraciones de “U” y la utiliza en “I”.	

Se eligió la segunda técnica que es filtrado colaborativo basado en el usuario, debido a que se ajusta más a los requerimientos del sistema, ya que en la encuesta realizada a 60 personas en la ciudad de Chiclayo, el 80% de las personas encuestadas asume que la manera de cómo contactan al técnico es por recomendación de las personas de su entorno, por lo que este sistema permite calcular un promedio en base a las calificaciones que han ido dejando otros clientes, ya sea por la experiencia de los servicios que han realizado los técnicos.

✓ **Determinar las API necesarias de Google Maps para influir en la gestión de servicios técnicos particulares.**

La plataforma de Google en la nube provee a los desarrolladores 17 APIS de desarrollo para conectar las aplicaciones con geolocalización en diferentes entornos, ya sea Android, iOS o Web.

A continuación, se observa un cuadro descriptivo de las APIS que provee Google Maps.

TABLA II
SELECCIÓN DE LAS APIS PARA EL SISTEMA DE RECOMENDACIÓN

Apis de mapas	Descripción	Api seleccionado
API de JavaScript de Maps	Este API sirve para mostrar un mapa a un sitio web, lo cual proporciona imágenes y datos locales de la misma fuente de Google maps.	✓

Maps SDK para Android	Este API sirve para mostrar un mapa basado en datos de Google maps para el sistema operativo Android.	✓
Instrucciones API	Este API accede a las rutas de conducción, ciclismo, caminata y transporte utilizando solicitud HTTP.	✓
API de Lugares	Este API provee la información de más de 100 millones de lugares.	✓
API de Geo codificación	Esta API convierte las direcciones en coordenadas geográficas y viceversa, lo cual se puede utilizar para colocar o posicionar uno o más marcadores en un mapa.	✓
Maps SDK para IOS	Permite agregar mapas y gráficos, tomando la información de la misma base de datos que usa Google Maps para el sistema operativo IOS.	
API estático de MAPS	Permite mostrar una imagen de Google Maps fija en una página web, sin Java Script.	
API estático de Street View	Permite incrustar una imagen panorámica estática del lugar en una página web, sin Java Script.	
API de Geolocalización	Permite conocer la ubicación y la radio de precisión, basados en información de torres de telefonía celular.	
API de zona horaria	Permite solicitar la zona horaria de una ubicación, la diferencia horaria de UTC y de verano en una superficie de la tierra.	
API de carreteras	Lo que hace esta API es identificar las carreteras por las que recorre el vehículo brindando así información útil sobre esas calles.	
API de matriz de distancia	Esta API lo que hace específicamente es calcular tiempo y distancia de viaje o trayecto	
API de mosaico semántico	Esta API brinda acceso a datos geográficos de muy buena calidad que se basa en datos de Google Maps como también proporciona herramientas de desarrollo que expanden el entorno de desarrollo de juegos en Unity para crear juegos móviles	
API de publicación de Street view	Este API crea imágenes reales y panorámicas en 360 grados en la página.	
API de elevación de MAPS	Este API visualiza todas las ubicaciones de la superficie de tierra incluyendo las ubicaciones del fondo del océano.	
API de inserción MAPS	Esta API coloca un mapa interactivo Street View en un sitio web.	
API de ubicación jugables	Lo que hace esta API es seleccionar cada ubicación jugable en función a la capacidad de cada jugador basados en ubicación como puntos de generación para cosas como depósitos de reparación y premios de juegos.	

Se seleccionó cinco APIS a utilizar según los requerimientos funcionales del software (sistema operativo, ubicación, rutas, GPS y direcciones). Entre ellas son:

- ✓ Maps SDK para Android

Este API sirve para mostrar un mapa basado en datos de Google Maps para el sistema operativo Android. Se seleccionó esta API debido a que el sistema de los usuarios técnicos y clientes fue implementado en ese entorno. Así mismo poder cargar los datos de Google maps en tiempo real y mostrar los técnicos de la ciudad de Chiclayo.

- ✓ Instrucciones API

Este API accede a las rutas de conducción, ciclismo, caminata y transporte utilizando solicitud HTTP. Fue necesario seleccionar esta API para que el técnico le permita saber cuál es el tiempo que le demora en ir a realizar un servicio al cliente.

- ✓ API de JavaScript de Maps

Este API sirve para mostrar un mapa a un sitio web, lo cual proporciona imágenes y datos locales de la misma fuente de Google Maps. Se seleccionó esta API para el administrador del sistema web, por lo que tendrá el control de los técnicos ubicados en la ciudad de Chiclayo.

- ✓ API de Geo codificación

Esta API convierte las direcciones en coordenadas geográficas y viceversa, lo cual se puede utilizar para colocar o posicionar uno o más marcadores en un mapa. Se seleccionó esta API para geo codificar las direcciones que ingresa el administrador del sistema para gestionar la información de los técnicos.

- ✓ API de lugares

Este API provee la información de más de 100 millones de lugares. Se seleccionó esta API para reconocer a través de la dirección escogida calcular la latitud y longitud de esa ubicación, con la facilidad de ingresar la dirección a registrar.

- ✓ **Desarrollar una aplicación web móvil en la que se integre las API determinadas y la técnica del sistema de recomendación de usuarios**

Se desarrolló la aplicación web móvil considerando la metodología RUP, así mismo haciendo uso de las cinco APIS determinadas de Google Maps y la técnica de sistema de recomendación de filtrado colaborativo basado en el usuario, por lo que el sistema es capaz de procesar la información que se registra y así poder mostrar resultados.

- ✓ **Obtener una percepción positiva en la utilidad y facilidad de uso percibidas de la aplicación por parte de los técnicos**

La aplicación desarrollada fue evaluada por los técnicos para medir la intencionalidad de su uso basada en el modelo TAM [20], mediante los factores de utilidad percibida (UP) y facilidad de uso percibida (FUP). Se encuestó a 21 técnicos que trabajan independientemente, a quienes se les proporcionó un video explicativo de la aplicación para visualizarlo y, en base a ello, dar respuesta a un cuestionario en línea para registrar sus percepciones. Se calculó el promedio de sus calificaciones en base a dichos factores, obteniéndose un valor resultante de 6.5 para el primero (UP), representado el 93% del máximo posible; mientras que en el segundo (FUP), el promedio de las calificaciones es de 6.2, representando el 88% del máximo posible, según la escala establecida en dicho instrumento. Dicho esto, los técnicos consideran que es extremadamente probable que la aplicación aumente la rapidez de sus tareas, así como su productividad, utilidad, desempeño y efectividad en el trabajo; y que es bastante probable que la aplicación GEOTEC sea fácil y entendible de usar, debido a su flexibilidad e interactividad.

DISCUSIÓN

A partir de los hallazgos encontrados, se aceptó la hipótesis general que establece que la aplicación web móvil gestionará los servicios técnicos particulares en la ciudad de Chiclayo integrando un sistema de recomendación. En el presente capítulo se realiza una comparación de los hallazgos obtenidos con los de otros autores mencionados en los antecedentes del problema.

A diferencia de las investigaciones [6] y [9] que solamente utilizaron las características de la geolocalización en su desarrollo, en el aplicativo móvil de la presente investigación propone el desarrollo de un módulo que permita gestionar los servicios técnicos particulares mediante la geolocalización, por lo cual el usuario tiene acceso a las ubicaciones de cada técnico incluyendo su posición actual (GPS) para poder determinar cuáles son los establecimientos más cercanos para su posible elección. Así mismo se tiene un control por el administrador del sistema, disponiendo de 2 APIs que le permite saber la ubicación de cada técnico y poder registrar la ubicación de ellos en caso se presente inconvenientes para registrarse por la aplicación móvil.

Por otra parte, en la investigación [7] y [10], se propuso el uso de la geolocalización únicamente en función de las rutas en la plataforma web, sin embargo en el aplicativo móvil de la presente investigación se propuso la inclusión de un cálculo de rutas más personalizado haciendo uso de la API de instrucciones y la API de lugares para poder calcular las rutas en base a las direcciones ingresadas por el usuario en el mismo sistema, por lo que le permite mostrar al técnico el tiempo y la distancia que le tomará en desplazarse hacia el cliente para realizar el servicio.

Así mismo en la investigación [6], se propuso la técnica de filtrado colaborativo en base a usuarios. En tal sentido el aplicativo móvil de la investigación realizada se plantea la misma técnica de filtrado, ya que permitirá recomendar a los usuarios del sistema en base a las opiniones cualitativas que ha ido teniendo otros usuarios según los servicios que se le haya brindado. Esto presenta ciertas limitaciones, debido a que la funcionalidad de la aplicación depende directamente del usuario, por lo que sino ingresan sus valores de entrada (calificación cualitativa) por parte del usuario, el sistema no podrá dar alguna recomendación.

Conclusiones

Se concluye que la aplicación web móvil, a partir de la información registrada y valoraciones que proporcionan los usuarios es capaz de geolocalizar y recomendar a los técnicos asignándoles una calificación.

1. Se analizaron las diferentes técnicas de sistema de recomendación y se identificó que la técnica de filtrado colaborativo en base al usuario permite calificar de manera cualitativa a los servicios que brinda un técnico, sirviendo como referencia para otros usuarios; siendo esta, una característica muy actual en la elección de un producto o servicio.
2. Se analizaron las APIS de Google Maps, determinándose que cinco de ellas se ajustaban más a los requerimientos del sistema porque presentaban características tales como: mostrar ubicaciones, convertir direcciones en coordenadas geográficas, calcular rutas, tiempo y distancia, GPS, y buscar direcciones.
3. Se desarrolló la aplicación web móvil apoyada de la técnica de sistema de recomendación de filtrado colaborativo basado en el usuario y las APIS de geo codificación, JavaScript de mapas, de lugares, instrucciones y mapas SDK para Android.
4. Se obtuvo una percepción positiva de la aplicación en cuanto a la utilidad y facilidad de uso por parte de los técnicos.

Recomendaciones

1. Se recomienda que la aplicación se desarrolle en otros sistemas operativos como por ejemplo iOS para que la toda la localidad pueda hacer uso de esta aplicación.
2. Se recomienda que la aplicación cuente por lo menos dos idiomas, por ser el inglés más recomendable.
3. Se recomienda que la aplicación pueda ejecutar de manera offline y no dependa en su totalidad de conexión a internet.

Referencias

- [1] GINEBRA , «Organización Internacional del Trabajo,» 2019 octubre 2019. [En línea]. Disponible en: https://www.ilo.org/global/about-the-ilo/newsroom/news/WCMS_723414/lang--es/index.htm.
- [2] K. Maza, «OIT: Trabajadores independientes cobran mayor fuerza en el Perú,» El Comercio, 19 diciembre 2018. [En línea]. Disponible en: <https://elcomercio.pe/economia/peru/oit-trabajadores-independientes-cobran-mayor-fuerza-peru-noticia-589177>.
- [3] E. Vizcarra , «Perú en Redes Sociales 2018,» La República, 16 febrero 2018. [En línea]. Disponible en: <https://larepublica.pe/sociedad/1198456-peru-en-redes-sociales-2018-parte-i/>.
- [4] Ministerio de Trabajo y Promoción del Empleo, «Estadístico sectorial Anuario 2018,» Mayo 2019. [En línea]. Disponible en: <https://www.gob.pe/institucion/mtpe/informes-publicaciones/279944-anuario-estadistico-2018>.
- [5] Instituto Nacional de Estadística e Informática (INEI), «Sistema de información regional para la toma de decisiones,» [En línea]. Disponible en: <http://systems.inei.gob.pe:8080/SIRTOD/app/consulta>. [Último acceso: 18 mayo 2020].
- [6] R. Flores Ibañez, «Sistema de georreferenciación de restaurantes en la ciudad de la Paz mediante el uso de smartphones», Tesis de fin de grado, Dpto: Facultad de Ingeniería, Universidad Regional Autónoma de los Andes, La Paz, 2016. [En línea]. Disponible en: <https://repositorio.umsa.bo/xmlui/bitstream/handle/123456789/10097/T.3195.pdf?sequence=1&isAllowed=y>
- [7] S. O. Troya Carbajal, «Implementación de un sistema de geolocalización de sitios turísticos mediante tecnología GPS - móvil para la agencia de viajes “Ecomontestour” de la ciudad de Otavalo», Tesis de fin de grado, Dpto: Facultad de Ingeniería, Universidad Regional Autónoma de los Andes, Ambato, 2016. [En línea]. Disponible en: <https://dspace.uniandes.edu.ec/bitstream/123456789/5263/1/PIUASIS010-2016.pdf>
- [8] G. M. Tambaco Tipantiza, «Desarrollo de un sistema de información de geolocalización interactiva (sigi) para la visualización geográfica y multimedia de las carreras de la universidad central del Ecuador», Tesis de fin de grado, Dpto: Facultad de Ingeniería, ciencias físicas y matemática, Universidad Central del Ecuador, Quito, 2016. [En línea].

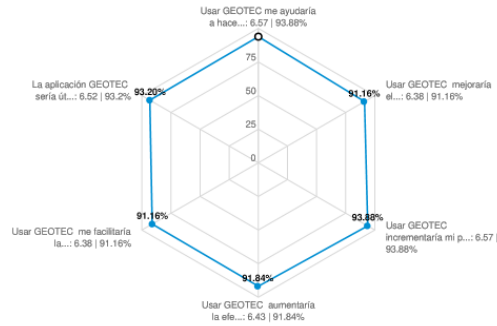
- Disponible en: <http://www.dspace.uce.edu.ec/bitstream/25000/6504/1/T-UCE-0011-264.pdf>
- [9] J. Quispe Limaylla y R. G. Yallico Lazo, «Desarrollo de una aplicación web para la reserva del servicio de estacionamiento mediante geolocalización en la provincia de Huancayo», Tesis de fin de grado, Dpto: Facultad de Ingeniería, Universidad Continental, Huancayo, 2018. [En línea]. Disponible en: https://repositorio.continental.edu.pe/bitstream/20.500.12394/5161/1/IV_FIN_103_TE_Quispe_Yallico_2018.pdf
- [10] I. E. Gamboa Castillo, «Análisis, diseño e implementación de un sistema web para el proceso de ventas con monitoreo de mercadería por geolocalización en la empresa NIH 2007 S.A», Tesis de fin de grado, Dpto: Facultad de Ingeniería, Universidad Alas Peruanas, Lima, 2018. [En línea]. Disponible en: <http://repositorio.ulasamericas.edu.pe/handle/upa/443>
- [11] J. Y. Cueva Huaman, «Aplicación móvil con geolocalización, mediante la metodología Mobile-D, para la gestión de visitas médicas en la empresa Laboratorios Siegfried S.A.C.», Tesis de fin de grado, Dpto: Facultad de Ingeniería, Universidad Cesar Vallejo, Lima, 2018. [En línea]. Disponible en: <http://repositorio.ucv.edu.pe/handle/20.500.12692/37415>
- [12] D. F. S. Fernandez Miranda y R. J. Querevalu Céspedes, «Propuesta de negocio para el servicio de taxi en la ciudad de Chiclayo teniendo en cuenta distintos elementos con base tecnológica», Tesis de fin de grado, Dpto: Facultad de Ciencias Empresariales, Universidad Católica Santo Toribio de Mogrovejo, Chiclayo, 2019. [En línea]. Disponible en: <http://tesis.usat.edu.pe/handle/20.500.12423/2288>
- [13] A. E. Rojas Vera y L. d. C. Villalobos Olivera, «Niveles de calidad de servicio del salón Spa Maria Bonita, en la ciudad de Chiclayo, 2016», Tesis de fin de grado, Dpto: Facultad de Ciencias Empresariales, Universidad Católica Santo Toribio de Mogrovejo, Chiclayo, 2018. [En línea]. Disponible en: <http://tesis.usat.edu.pe/handle/20.500.12423/1383>
- [14] E. Carrasco Peña y G. Y. M. Díaz Toro, «Optimización de las rutas recolectoras de residuos sólidos en el distrito de Chiclayo, para mejorar la eficiencia del servicio de limpieza. Chiclayo-2017», Tesis de fin de grado, Dpto: Facultad de Ingeniería,

- Universidad Señor de Sipán, Chiclayo, 2017. [En línea]. Disponible en: <https://repositorio.uss.edu.pe/handle/20.500.12802/4234>
- [15] P. Gómez , T. Guarda, J. Cedeño, A. Benavides, C. Alejandro, G. Mosquera, T. Garcia y V. Benavides , «Sistemas de Recomendación: un enfoque a las técnicas de filtrado,» *RISITI*, p. 9, 05 diciembre 2018.
- [16] Gerencia Operativa Incorporación de Tecnologías (InTec), «Google Maps y Street View,» Buenos Aires, 2020.
- [17] Google, «Google Maps Platform,» [En línea]. Disponible en: <https://developers.google.com/maps/documentation/?hl=es-419>.
- [18] C. A. Bernal, Metodología de la investigación, tercera ed., O. Fernández Palma, Ed., Pearson, 2010, p. 322.
- [19] R. Pimienta Lastra, «Encuestas probabilísticas vs. no probabilísticas,» de *Politica y Cultura;* , 2000, pp. 263-276.
- [20] J. Cabero Almenara, J. Barroso Osuna y M. d. C. Llorente Cejudo, «Technology acceptance model & realidad aumentada: estudio en desarrollo,» *LA SALLISTA DE INVESTIGACIÓN*, vol. 13 No. 2, p. 9, 2016.

Anexos

ANEXO N° 01. RESULTADOS DE LA ENCUESTA REALIZADA A LOS TÉCNICOS

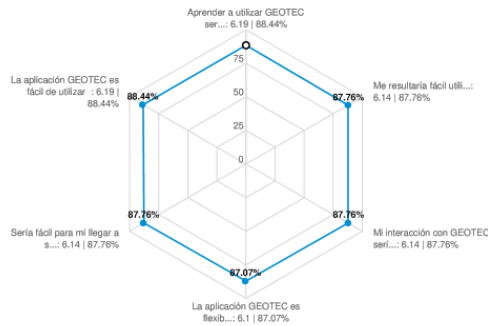
Utilidad percibida (UP)



Powered by AI

Question	Count	Score	Extremadamente improbable	Bastante improbable	Ligeramente improbable	Ninguno	Ligeramente probable	Bastante probable	Extremadamente probable
Usar GEOTEC me ayudaría a hacer mis tareas más rápido	21	6.57							
Usar GEOTEC mejoraría el desempeño de mi trabajo	21	6.38							
Usar GEOTEC incrementaría mi productividad	21	6.57							
Usar GEOTEC aumentaría la efectividad en mi trabajo	21	6.43							
Usar GEOTEC me facilitaría la realización de mi trabajo	21	6.38							
La aplicación GEOTEC sería útil en mi trabajo independiente	21	6.52							
Average		6.47							

Facilidad de uso percibida (FUP)



Powered by AI

Question	Count	Score	Extremadamente improbable	Bastante improbable	Ligeramente improbable	Ninguno	Ligeramente probable	Bastante probable	Extremadamente probable
Aprender a utilizar GEOTEC sería fácil para mí	21	6.19							
Me resultaría fácil utilizar GEOTEC para hacer lo que yo quiero que haga	21	6.14							
Mi interacción con GEOTEC sería clara y entendible	21	6.14							
La aplicación GEOTEC es flexible para interactuar con él	21	6.1							
Sería fácil para mí llegar a ser un experto en el uso de la aplicación GEOTEC	21	6.14							
La aplicación GEOTEC es fácil de utilizar	21	6.19							
Average		6.15							

ANEXO N° 02. ANÁLISIS DEL TURNITIN

feedback studio | Jhon Anderson ZULOETA LOPEZ | artículo científico

APLICACIÓN WEB MÓVIL PARA GESTIONAR LOS SERVICIOS TÉCNICOS PARTICULARES EN LA CIUDAD DE CHICLAYO
 Zuloeta, López, Jhon, Anderson*
 *Universidad Católica Santo Toribio de Mogrovejo, Facultad de Ingeniería

RESUMEN
 Hoy en día el uso de las tecnologías de información ha revolucionado a toda la generación, lo cual permite que todos estemos más conectados e informados en el día a día. En la presente investigación se tomó como objeto de estudio a los trabajadores independientes, ya que las cifras de ellos han incrementado en cada año, siendo los principales dueños de su trabajo presentando diversos problemas como el financiamiento de su negocio y captación de clientes. Así mismo a los ciudadanos ya que no cuentan con un acceso a la información para solicitar un servicio técnico o que estas se encuentren en las redes sociales, lo que ocasiona que no les sea suficientemente útil. En tal sentido fue necesario realizar esta investigación para obtener datos importantes de los trabajadores

Página: 2 de 19 | Número de palabras: 5032 | Text-only Report | High Resolution | Activado

Resumen de coincidencias

19 %

Nº	Match	Source	Percentage
1	www.risti.xyz	Fuente de Internet	3 %
2	hdl.handle.net	Fuente de Internet	2 %
3	Entregado a Universida...	Trabajo del estudiante	2 %
4	Entregado a Universida...	Trabajo del estudiante	1 %
5	tesis.usat.edu.pe	Fuente de Internet	1 %
6	repositorio.unasam.ed...	Fuente de Internet	1 %
7	Entregado a Pontificia ...	Trabajo del estudiante	1 %
8	Entregado a Universida...	Trabajo del estudiante	1 %
9	inforfydocen.blogspotc...	Fuente de Internet	1 %