

Ambulancia McQueen: optimización de rutas para ambulancias a partir de la distancia entre el accidente y el centro de salud

Alexia Shariann Ríos Alarcón
as.riosa@alum.up.edu.pe

Carlos Felipe Mariños Oré
cf.marinoso@alum.up.edu.pe

Andrea Isabel Nepo Gamero
ai.nepog@alum.up.edu.pe

Juan David Amado Cárdenas Zúñiga
jd.cardenasz@alum.up.edu.pe

Víctor Andrés Ayma Quirita
va.aymaq@up.edu.pe

Universidad del Pacífico, Lima, Perú

Recibido: 9/8/2021 Aceptado: 28/9/2021
doi: <https://doi.org/10.26439/ciis2021.5583>

RESUMEN. Actualmente el transporte prehospitalario, como el que brindan las ambulancias para la atención médica de emergencia o soporte para atención primaria, enfrenta un gran problema en Lima: el tráfico, el cual además se ve afectado por el tiempo que transcurre desde que ocurre la emergencia hasta que el centro de salud envía una ambulancia para su atención. En ese sentido, es de suma importancia optimizar el tiempo de respuesta de este proceso para que la atención médica se brinde lo más rápido posible, ya que, durante una emergencia, el tiempo de atención es crucial. Por ello, este trabajo propone una metodología para optimizar la ruta y el tiempo de envío de una ambulancia para la atención de la emergencia tomando en consideración la distancia entre el centro de salud y el lugar de ocurrencia de la emergencia, específicamente, centrando nuestro piloto en el distrito de San Borja. Por otro lado, desde el punto de vista del paciente, se busca que este pueda pedir y acceder a una ambulancia lo más pronto posible a través de una única plataforma. Para ello se ha diseñado un programa de

código abierto, codificado en Python con las librerías OSMnx y Networks, el cual hace uso además del algoritmo de Dijkstra para la optimización de la ruta a seguir hasta el lugar de la emergencia. Las pruebas experimentales pudieron validar que mediante este proceso se logró optimizar el tiempo de acción requerido para la atención de estos casos al facilitar la logística necesaria para enviar las ambulancias, considerando, asimismo, que este tiempo ahorrado podría hacer posible salvar la vida de las personas involucradas en tales eventos.

PALABRAS CLAVE: teoría de grafos / atención de emergencias / optimización de rutas / OSMnx / OpenStreetMap, Dijkstra

MCQUEEN AMBULANCE: OPTIMIZATION OF ROUTES FOR AMBULANCES BASED ON THE DISTANCE BETWEEN THE ACCIDENT AND THE HEALTH CENTER

ABSTRACT. Currently, pre-hospital transport, such as that provided by ambulances for emergency medical care or support for primary care, faces a significant problem in Lima: traffic, which is also affected by the time that elapses from the emergency that occurs until the health center sends an ambulance for care. In this sense, it is essential to optimize the response time of this process so that medical attention is provided as quickly as possible, attention time is crucial during an emergency. Therefore, this work proposes a methodology to optimize the route and time of dispatch of an ambulance for emergency care, considering the distance between the health center and the place of occurrence of the emergency, specifically focusing our pilot on the district of San Borja. On the other hand, on the patient's side, it is sought that he can request and access an ambulance as quickly as possible through a single platform. For this, an open-source program has been designed and coded in Python with the OSMnx and Networks libraries, which also uses Dijkstra's algorithm to optimize the route to follow to the emergency site. The experimental tests were able to validate that through this process, it was possible to optimize the action time for the attention of cases by facilitating the logistics necessary to send the ambulances, also considering that this saved time could offer the possibility of saving the lives of the people involved in such events.

KEYWORDS: graph theory / emergency care / route optimization / OSMnx / OpenStreetMap / Dijkstra

1. INTRODUCCIÓN

Un estudio realizado por la compañía holandesa TomTom colocó a Lima en el tercer lugar de las ciudades con más congestión vehicular en el 2018 (TomTom, 2018). Si bien el Ministerio de Transportes y Comunicaciones (MTC) creó la Autoridad de Transporte Urbano con el objetivo de organizar mejor el tránsito automovilístico en la ciudad, los cambios aún no han sido significativos.

Uno de los sectores más afectados por el tráfico en nuestro país es el Servicio de Atención Móvil de Urgencias y Emergencias (SAMU) (Reyna, 2018). En una entrevista con RPP Noticias, el médico de ambulancias, doctor Víctor Melgarejo, comentó:

La distancia y la falta de educación vial no nos permiten llegar en forma oportuna a una emergencia. Nos embarga la sensación de no poder ayudar en momentos en que queremos ayudar. Es por falta de conocimiento de la gente que no sabe que una unidad en camino puede salvar vidas.

El especialista menciona, además, que el tráfico vehicular puede convertir un viaje de 5 o 7 minutos en uno de 10 o más minutos (Reyna, 2018), lo cual es inaceptable en situaciones donde una pequeña demora en la atención puede definir la vida o la muerte de una persona, debido a la gran relación que existe entre el tiempo de atención y su resultado (Bürger *et al.*, 2018). Asimismo, es importante resaltar la importancia de la adecuada circulación de las ambulancias, porque según un estudio realizado por Enciso y Villanueva (2017), el 80 % de los pacientes con prioridad de atención I, que fallecieron hasta el momento, acudieron a un centro de salud con una unidad móvil particular, no médica.

Ante esta problemática, se desarrolló un sistema de código abierto, implementado en Python (<https://www.python.org/>), que, guiándose de la teoría de grafos y utilizando el algoritmo de Dijkstra, elige el hospital o clínica más adecuado dependiendo de la distancia a la cual se ubica respecto del paciente y del seguro al que este pertenezca; a su vez, nuestra plataforma encuentra la ruta óptima para enviarle una ambulancia. Es preciso mencionar que este piloto no es una aplicación móvil aún, pero su desarrollo está dentro de los trabajos futuros sobre esta investigación preliminar.

El presente trabajo se diferencia de las herramientas disponibles en otras plataformas, como Google Maps, en tres aspectos fundamentales. En primer lugar, la información empleada por nuestro sistema está actualizada y validada con datos reales de los hospitales y calles, a diferencia de otras herramientas que están implementadas con datos desfasados y poco confiables. En segundo lugar, el programa está diseñado específicamente para realizar una búsqueda directa entre los distintos centros de salud disponibles al momento de ocurrir la emergencia, a los que puede acudir un paciente, de modo que la selección se realiza de manera rápida y eficiente. Por otro lado, aunque muchas de las herramientas actualmente disponibles permiten realizar consultas similares, estas son más lentas, pues dependen de las habilidades de búsqueda y selección del usuario, hecho que es superado por nuestra propuesta. Finalmente, la herramienta

permite contactar al centro de salud directamente, en un mismo programa integrado, mientras que el uso de otros medios requiere de canales externos (Gobierno del Perú, 2021), como llamadas o mensajes de texto, para que el usuario pueda comunicarse con el centro de salud elegido; y es en consideración de estas características que nuestro trabajo hace referencia al personaje de Rayo McQueen, de la película *Cars*, por su rapidez de respuesta y eficacia en brindar una solución.

Por último, la estructura del artículo se dispone como sigue: en la sección 2 se exponen algunos de los fundamentos teóricos utilizados en este trabajo, la sección 3 describe la metodología desarrollada para implementar la solución; a continuación, en la sección 4 se presentan los resultados obtenidos y, finalmente, las conclusiones y recomendaciones del trabajo se describen en la sección 5.

2. FUNDAMENTOS

2.1 Grafos

Los grafos son tuplas compuestas por un conjunto finito de vértices, o nodos, y aristas (Moreno y Ramírez, 2012, p. 49). Usualmente se representa a los vértices con círculos y a las aristas con líneas:

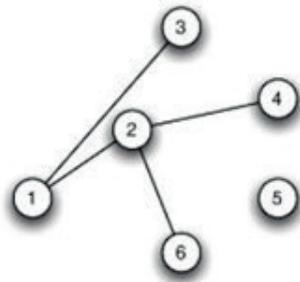


Figura 1. Grafo simple

Fuente: Moreno y Ramírez (2012)

Asimismo, los grafos que presentan un peso asociado a cada arista son conocidos como grafos ponderados. Por otro lado, cuando dos vértices se conectan mediante una arista, la relación es bidireccional. Si en el grafo se especifica solo un sentido de conexión, el grafo es conocido como grafo orientado o dígrafo. En este trabajo, a partir de la información del grafo del distrito de San Borja se crean las rutas para las ambulancias y se visualizan en el mismo grafo.

2.2 Algoritmo de Dijkstra

Diseñado por el físico y científico computacional Edsger Dijkstra (1930-2002), este algoritmo permite encontrar la ruta más corta entre un vértice de partida y otro de llegada. Según Salas (2008), el algoritmo de Dijkstra es voraz (algoritmo goloso), que genera uno a uno los caminos de un nodo al resto por orden creciente de longitud; usa un conjunto de vértices donde, a cada paso del algoritmo, se guardan los nodos que representan al camino mínimo desde el nodo de partida y devuelve un vector indexado por vértices, de modo que para cada uno de estos vértices podemos determinar el coste de un camino más económico (de peso mínimo) a tales vértices (p. 1). Este algoritmo forma parte del código para crear las rutas más cortas para las ambulancias a través de la función “nx.shortest path” de la librería NetworkX (Priyam, 2020).

2.3 OpenStreetMap

Esta plataforma es un proyecto colaborativo de nivel internacional que tiene como objetivo crear un mapa gratuito del mundo con todas las calles, nodos y etiquetas que describen la característica principal de un elemento como restaurantes o clínicas, y es de datos abiertos (Morales, 2021). Es necesario trabajar con esta plataforma para recolectar los datos sobre las coordenadas de los hospitales y clínicas en San Borja, y utilizar las funciones de librería OSMnx (Boeing, 2016) con el objetivo de mostrar el grafo y las rutas.

3. METODOLOGÍA

La problemática de las ambulancias en Lima puede aliviarse con la herramienta del presente trabajo porque permite encontrar la ruta con la menor distancia entre el paciente y el centro de salud más cercano en función del tipo de seguro con el que cuente el paciente.

En ese sentido, se siguieron cuatro pasos importantes para la creación de la herramienta utilizando el lenguaje de programación Python, tal como se muestra en la figura 2. El primero fue crear tablas con los datos relevantes de las clínicas y hospitales en San Borja. Una vez que se tuvieron esos datos, se modeló la data extraída para formar un grafo que permitirá un mejor manejo de la información. Esta data se utilizó en la aplicación del algoritmo de Dijkstra, para obtener la ruta más corta según la distancia. Por último, se creó una interfaz para que los usuarios puedan interactuar con el programa.

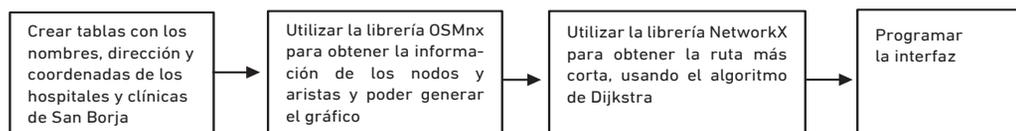


Figura 2. Procedimiento para construir la herramienta que permite obtener la ruta más corta
Elaboración propia

3.1 Descripción del lugar de estudio

San Borja es un distrito de la ciudad de Lima Metropolitana, tiene una extensión de 9,96 km², cuenta con una población aproximada de 112 048 habitantes y presenta una densidad poblacional de 11 249,8 habitantes/km². Asimismo, este distrito se caracteriza por la alta concentración de centros de salud (siete clínicas y cinco hospitales), por lo que fue el seleccionado para realizar la implementación de nuestro piloto.

3.2 Adquisición de datos

Para el primer paso de la solución, se requirió extraer información de los centros de salud disponibles en el distrito de San Borja, Lima. Estos datos se obtuvieron de Google Maps mediante una búsqueda simple. Luego, se contrastaron con la información disponible en OpenStreetMap (Morales, 2021).

Finalmente, se corroboró que los centros de salud preseleccionados cumplan con los requisitos mínimos para atender una emergencia, es decir, que cuenten con, al menos, una ambulancia para trasladar a los pacientes. Así, la información final generada para cada centro de salud verificado es la siguiente: nombre, dirección y referencia geográfica precisa (coordenadas X e Y).

3.3 Medición de las distancias en la red

Luego de haber identificado todos los centros de salud junto con sus coordenadas, continuamos con la generación del grafo; esto se logró utilizando la librería OSMnx (Boeing, 2016). La figura 4 permite apreciar la imagen del grafo $G(V, A)$, el cual está implementado sobre el mapa del distrito de San Borja; sus calles están representadas por las aristas del grafo, mientras que las esquinas por los vértices. Se utilizó todo el mapa del distrito para el grafo debido a que una emergencia podría suceder en cualquier parte de este, ergo, es necesario implementar toda su área para no tener que actualizar el grafo periódicamente. Finalmente se resaltó la ubicación de los hospitales, clínicas, así también como la ubicación de un punto de emergencia, aleatorio, para una mejor interpretación (revisar leyenda de la figura 4).

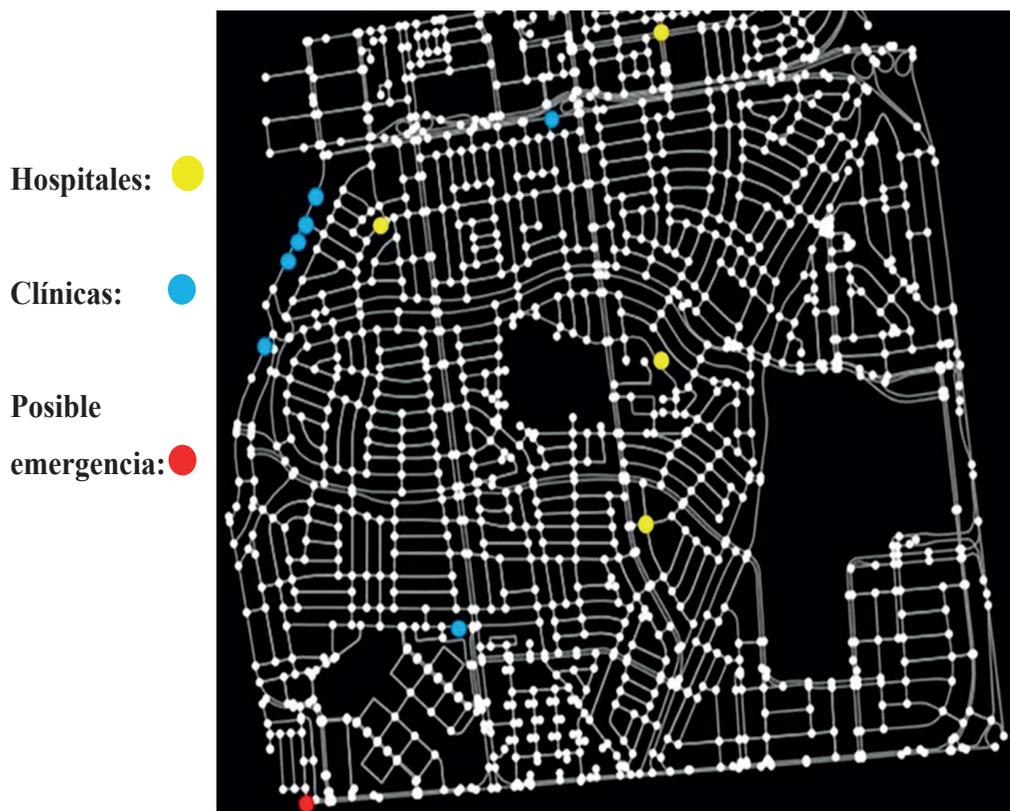


Figura 3. Grafo de San Borja con centros de salud resaltados

Elaboración propia

3.4 Proceso del desarrollo del programa

Para la creación de la interfaz, se siguieron distintos pasos, tal como se presenta en la figura 2. El primero fue definir funciones relevantes para el programa como las que evitan errores por los datos que se ingresan en el menú, otra que escoge el centro de salud más cercano al usuario dependiendo del seguro del paciente, y la función principal que permite generar la ruta con menor distancia a partir de las coordenadas del centro de salud y el usuario. Luego se definieron las clases, las cuales son Usuario, CentroSalud, Ambulancias, Hospitales y Clínicas.

Se crearon las listas de las clínicas y hospitales que se tomarían en cuenta, es decir, los que se encuentran en el distrito de San Borja y cuenten con servicio de ambulancias. En estas listas también se toma en cuenta el tipo de seguro del usuario, ya sea público o privado. Finalmente, se creó un menú para que los usuarios puedan utilizar el programa de manera sencilla.

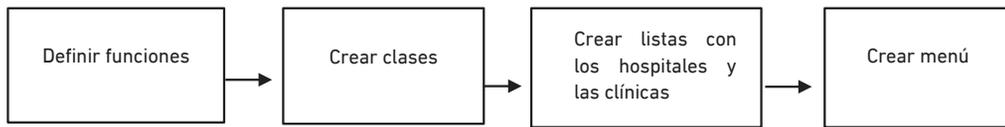


Figura 4. Creación de la interfaz

Elaboración propia

Para el correcto funcionamiento del código se crearon varias clases: Usuario y CentroSalud (a partir de esta se crearon dos clases hijas, que son Clínicas y Hospitales). La clase Usuario sirve para identificar a los clientes, tomando datos como su DNI, nombre, celular, correo electrónico, referencia geográfica precisa (coordenadas X e Y) y seguro al que pertenece.

En el caso de la clase CentroSalud, se toman datos como el nombre, la dirección, los seguros que son aplicables y sus coordenadas de OpenStreetMap. Luego, se crean listas que se utilizarán más adelante para relacionar al cliente con su centro de salud y que sirven para separar las clases Hospitales y Clínicas.

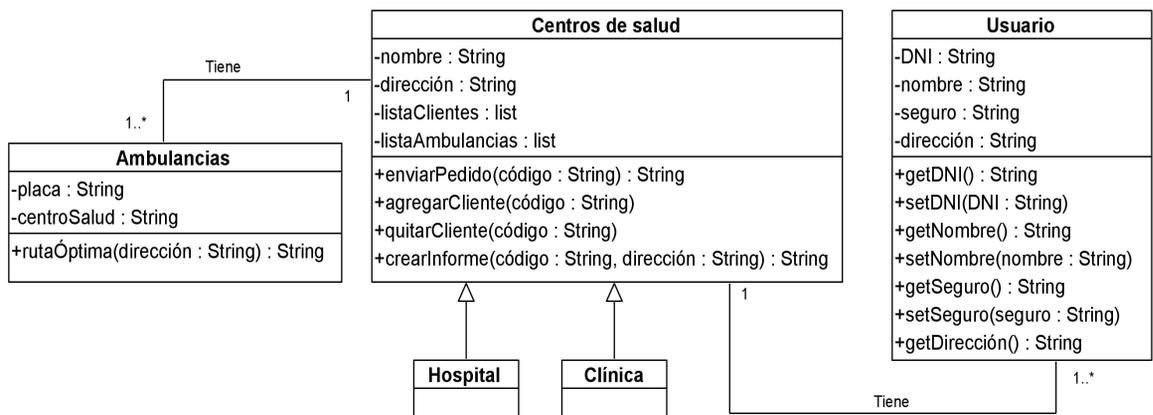


Figura 5. Diagrama de clases

Elaboración propia

Posteriormente, para implementar la solución, se programaron algunas funciones que nos permiten obtener la ruta óptima hacia el paciente de acuerdo con sus requerimientos específicos. Entre todas las funciones existen tres destacables.

La primera función es “conseguirCentroSalud”. Esta función se encarga de evaluar el tipo de seguro del paciente y asociarlo al centro de salud más cercano donde este se pueda aplicar y al cual será trasladado. La segunda función es “conseguirRuta”, la cual es empleada para obtener la ruta óptima entre el punto de emergencia y el centro de salud más cercano. Para este fin, la

función emplea el algoritmo de Dijkstra, que evalúa dinámicamente cada nodo de modo que al final se obtiene el camino más corto. Este proceso se desarrolla mediante la librería NetworkX (Priyam, 2020) que ya cuenta con el algoritmo anteriormente mencionado, el cual utiliza el grafo obtenido a partir de las calles y esquinas del distrito de San Borja.

Para ordenar y utilizar las diferentes funciones del programa, se implementó un menú con cinco opciones distintas, donde:

- La primera es “Registrarse”, y le permitirá al usuario ingresar los datos necesarios para usar las otras funciones (DNI, nombre, coordenadas en X e Y, correo electrónico, número de celular y el seguro con el que cuente). Esta opción también comprobará que el celular y el DNI ingresados sean válidos.
- La segunda opción, “Editar información”, le pedirá al usuario que ingrese su DNI y, después de comprobar que sea válido, permitirá que el usuario seleccione el dato que desee modificar.
- Al utilizar la tercera opción, “Pedir una ambulancia”, el programa pedirá al usuario ingresar su DNI y mostrará la ruta al hospital o clínica más cercano (esta ruta será utilizada por el conductor de la ambulancia), junto con la distancia en metros.
- Usar la opción 4, “Mostrar información”, le pedirá al usuario ingresar su DNI, y mostrará la información ingresada en la opción 1. En caso de que el usuario haya cambiado algún dato usando la opción 2, le mostrará la información actualizada.
- Por último, la opción 5, “Salir”, terminará el programa automáticamente.

4. RESULTADOS

En primer lugar, conforme se presenta en la figura 6, un resultado del proyecto es el mapa georreferenciado donde se resaltan los hospitales y las clínicas seleccionadas en el distrito de San Borja, el cual fue implementado gracias a la librería Folium (Story, 2015).



Figura 6. Visualización del mapa de San Borja con hospitales y clínicas
Elaboración propia

En esta figura se pueden observar doce centros de salud en total, de los cuales cinco son hospitales que envían ambulancias a personas que están afiliadas a EsSalud y al SIS, y los siete restantes son clínicas para personas con seguros privados como Rimac, Pacífico, Mapfre, La Positiva y Sanitas.

A continuación, la figura 7 presenta la dinámica de los menús de operación del programa desarrollado, mostrando la interacción con el usuario y dando un lugar determinado de un accidente dentro del distrito de San Borja. Además, la columna de la izquierda de la figura muestra cómo funciona el programa con una persona que cuenta con un seguro privado como Mapfre y en la columna de la derecha se presenta el mismo ejemplo, pero con una persona que cuenta con un seguro público como el SIS.

Para ello, el primer paso se realiza a través de la cuarta opción, donde se imprime la información del usuario; luego, se visualiza la ruta óptima junto con el nombre de la clínica y la distancia en metros al seleccionar la tercera opción. Conforme se ha mencionado previamente, en ambos ejemplos los usuarios tienen la misma referencia geográfica, esto con el fin de mostrar, de manera simultánea, el criterio de selección bajo la consideración de la menor distancia hacia el evento y el tipo de seguro asociado al paciente.

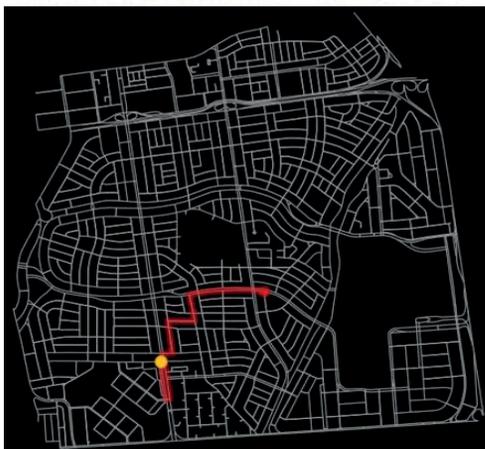
- Menú
1. Registrarse
 2. Editar información
 3. Pedir una ambulancia
 4. Mostrar información
 5. Salir

Elija una opción:

Elija una opción: 4
Ingrese su DNI: 25572978

Nombre: Claudia Bravo
DNI: 25572978
Coordenada x: -76.99392
Coordenada y: -12.10052
Email: cla284@gmail.com
Celular: 968513991
Seguro: Mapfre

Elija una opción: 3
Ingrese su DNI: 25572978
Centro de salud: Clínica Especialidades Medicas
Distancia en metros: 1911.14
La ruta ha sido enviada a la ambulancia asignada :)



Elija una opción: 4
Ingrese su DNI: 21668087

Nombre: Alfredo Perez
DNI: 21668087
Coordenada x: -76.99392
Coordenada y: -12.10052
Email: alf487@gmail.com
Celular: 904859614
Seguro: SIS

Elija una opción: 3
Ingrese su DNI: 21668087
Centro de salud: Policlinico Naval San Borja
Distancia en metros: 1508.41
La ruta ha sido enviada a la ambulancia asignada :)



Figura 7. Ejemplos del funcionamiento de la herramienta con usuarios inventados
Elaboración propia

Estos resultados muestran que incluso con el mismo lugar de partida, representado por el punto rojo en la figura, el programa realiza el filtro por el tipo de seguro y busca el centro de salud más cercano al paciente, haciendo este proceso eficiente y de respuesta óptima para la correcta atención del transporte hacia la emergencia.

5. CONCLUSIONES

Luego de extraer información de OpenStreetMap, implementarla en un grafo que representa a San Borja y desarrollar una interfaz con funciones que utilizan la librería OSMnx y Networks, fue posible emplear el algoritmo de Dijkstra para hallar la ruta mínima entre el centro de salud más cercano y el lugar de la emergencia.

Mediante este proceso se logra optimizar el tiempo requerido de atención de la emergencia en estos casos, logrando facilitar la logística necesaria para enviar las ambulancias, la cual generalmente toma minutos valiosos, que podrían ser vitales para conseguir salvar una vida.

En cuanto a las recomendaciones y trabajos futuros que se pueden originar de esta investigación, podemos mencionar que el trabajo fue elaborado a partir de los nodos y aristas del distrito de San Borja con la información de OpenStreetMap; sin embargo, es posible ampliarlo a más distritos utilizando la misma metodología e incluso con la información de otras aplicaciones como Google Maps.

Asimismo, se plantea implementar mayores criterios para elaborar las posibles rutas a seguir por la ambulancia, tomando en consideración la información de posibles accidentes de tránsito, el tiempo que toma la ruta en función de la densidad de tránsito actual, o identificar la cantidad de ambulancias disponibles en el centro de salud más cercano. Además, considerando que el piloto de este trabajo fue desarrollado en una interfaz de escritorio, se tiene planeado, como trabajo futuro, desarrollar una aplicación móvil que permita conectar el dispositivo del usuario directamente con los centros de salud asociados.

REFERENCIAS

- Agencia EFE. (20 de junio del 2019,). Lima se enfrenta a una de sus peores caras: el tráfico caótico. *Gestión*. <https://gestion.pe/peru/lima-enfrenta-peores-caras-trafico-caotico-270759-noticia/>
- Boeing, G. (2016). *OSMnx: Python for Street Networks*. Geoff Boeing. <https://geoffboeing.com/2016/11/osmnx-python-street-networks/>
- Bürger, A., Wnent, J., Bohn, A., Jantzen, T., Brenner, S., Lefering, R., Seewald, S., Gräsner, J. T., y Fischer, M. (2018). The Effect of Ambulance Response Time on Survival Following Out-of-Hospital Cardiac Arrest. *Deutsches Arzteblatt international*, 115(33-34), 541-548. <https://doi.org/10.3238/arztebl.2018.0541>
- Enciso, C., y Villanueva, J. (2017). *Sobrevida de pacientes asignados como prioridad de atención I y II, en emergencia de adultos de un Hospital General de Lima, según el tipo de transporte prehospitalario utilizado* [Tesis de licenciatura, Universidad Peruana Cayetano

- Heredia]. Universidad Peruana Cayetano Heredia, Repositorio Institucional. <https://hdl.handle.net/20.500.12866/660>
- Gobierno del Perú. (8 de julio del 2021). *Solicitar atención médica en caso de emergencia, SAMU*. <https://www.gob.pe/1013-solicitar-atencionmedica-en-caso-de-emergencia-samu>
- Johnsonbaugh, R. (2005). *Matemáticas discretas* (6.^a ed.). Pearson Educación.
- Morales, A. (2021). *OpenStreetMap: la plataforma de mapas libre más grande del mundo*. MappingGIS. <https://mappinggis.com/2021/04/openstreetmap-la-plataforma-de-mapas-libre-mas-grande-del-mundo/>
- Moreno, E., y Ramírez, H. (2012). *Grafos: fundamentos y algoritmos*. J. C. Sáez Editor. <https://elibro.net/es/ereader/bibliotecaup/68438?page=50>
- Priyam, A. (2020). *Find and Plot your Optimal Path Using OSM, Plotly and NetworkX in Python*. Medium. <https://towardsdatascience.com/find-and-plot-your-optimal-path-using-plotly-and-networkx-in-python-17e75387b873>
- Reyna, R. (2018). *Un día con el SAMU: El tráfico de Lima es el principal problema ante una emergencia*. RPP Noticias. <https://rpp.pe/vital/salud/un-dia-con-el-samu-el-trafico-de-lima-es-el-principal-problema-ante-una-emergencia-noticia-1125527>
- Rodríguez, L. Y., Solis, T., y Saromo, V. (2020). *Limpieza y desinfección de ambulancias luego de transportar un paciente confirmado con COVID-19*. IETSI. <https://docs.bvsalud.org/biblioref/2020/05/1096333/desinfeccion-de-ambulancia-ietsi-2020.pdf>
- Salas, A. (2008). *Acerca del algoritmo de Dijkstra*. arXiv. <https://arxiv.org/abs/0810.0075>
- Story, R. (2015). *Folium*. Pypi.org. <https://pypi.org/project/folium/0.1.5/>
- Tejeda, A. (2018). Plan de Gobierno Municipal San Borja 2019-2021. <http://peruvotoinformado.com/descargas/pg/plan-de-gobierno-de-carlos-alberto-tejada-noriega.pdf>
- TomTom. (2018). *Lima Traffic Report*. https://www.tomtom.com/en_gb/trafficindex/lima-traffic/
- Torrubia, G., y Terrazas, V. (2001). Algoritmo de Dijkstra: un tutorial interactivo. *VII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2001)*.

