



# Aplicación del algoritmo A\* con inteligencia colectiva para planificar rutas en sistema de transporte público

Arnulfo Castro Vásquez<sup>1</sup>, Raúl Ojeda Luna<sup>1</sup>, Israel Hernández Hernández<sup>1</sup>, Mayra V. Linares Gil<sup>4</sup>

<sup>1,2,3,4</sup>Universidad Autónoma de Ciudad Juárez, Avenida del

Charro 450 Norte, Ciudad Juárez,

Chihuahua, México,

C.P. 32310.



**Correo correspondiente**

arncastr@uacj.mx

## Resumen

En los últimos años, el transporte público de Ciudad Juárez ha crecido considerablemente ofreciendo el servicio a miles de usuarios cada día. El tiempo de traslado es importante para los usuarios que buscan llegar a su destino existiendo diversos factores que pueden retrasar su traslado. Este trabajo aplica el algoritmo A\* en combinación con la técnica de inteligencia colectiva para encontrar una ruta óptima y planificar rutas en el sistema de transporte público de la ciudad. Para esto, se construyó un software prototipo de aplicación móvil para sugerir la mejor ruta, considerando la participación de los ciudadanos. La idea es que los usuarios, desde su celular, aporten conocimiento útil en tiempo real sobre el estado de las rutas del transporte público, de manera que, otros usuarios puedan enterarse y puedan planificar su ruta para llegar a su destino en el menor tiempo posible. Para lograr lo anterior, fue necesario recolectar información de las diferentes rutas que componen el sistema de transporte público. La aplicación se configuró con un registro de 16 rutas y más de 250 nodos ubicados en la ciudad. La propuesta de este artículo se validó con diversos usuarios localizados en diferentes nodos, los cuales interactuaron con la aplicación para crear inteligencia colectiva y permitir a otros usuarios poder planificar sus rutas.

Se presentan los resultados obtenidos durante las pruebas, los cuales confirman que la aplicación del algoritmo A\* en combinación con la inteligencia colectiva es una alternativa efectiva en la planificación de rutas para los usuarios en un sistema de transporte público.

## Abstract

In recent years, public transportation in Ciudad Juarez has grown considerably, offering service to thousands of users every day. Travel time is important for users seeking to reach their destination and there are several factors that can delay their travel. This work applies the A\* algorithm in combination with the collective intelligence technique to find an optimal route and plan routes in the city's public transportation system. For this, a prototype mobile application software was built to suggest the best route, considering the participation of citizens. The idea is that users, from their cell phones, provide useful knowledge in real time about the status of public transport routes, so that other users can find out and plan their route to reach their destination in the shortest possible time. To achieve the above, it was necessary to collect information from the different routes that make up the public transportation system. The application was configured with a record of 16 routes and more than 250 nodes located in the city. The proposal of this article was validated with various users located in different nodes, whom interacted with the application to create collective intelligence and allow other users to plan their routes. The results obtained during the tests are presented, which confirm that the application of the A\* algorithm in combination with collective intelligence is an effective alternative in route planning for users in a public transportation system.

**Palabras clave:** Sistema de transporte público, inteligencia colectiva, aplicaciones móviles.

**Keywords:** Public transportation system, collective intelligence, mobile applications.

---

## Introducción

Las ciudades crecen día a día, por lo cual deben adaptarse y resolver problemas cotidianos que afectan a sus ciudadanos. Un problema recurrente se refiere al transporte público en donde cada ciudad posee diferentes rutas de

recorridos, que a su vez están compuestas por nodos (o estaciones) donde la gente sube y baja del transporte. El mayor reto consiste en realizar un traslado eficiente desde un nodo origen a un nodo destino en el menor tiempo posible. Existen factores que pueden afectar el tiempo de traslado de los usuarios, tales como el tráfico vehicular en determinadas horas del día, accidentes de tránsito, entre otros. En la ma-

yoría de las veces los usuarios desconocen estos factores, lo cual les ocasiona problemas para llegar a su destino y todo lo que ello implica.

En la literatura los algoritmos de trayectorias utilizados para aplicaciones informáticas orientadas al transporte público, iniciaron con el estudio de la trayectoria algebraica (Bellman, 1958) (Dijkstra, 1959). Este algoritmo es aplicable a múltiples problemas en la vida cotidiana, tales como: recorridos de vuelos entre ciudades, interacción en líneas de metro, entrega de paquetes, por mencionar algunos. El algoritmo de trayectoria aplicado al transporte utiliza: distancias, tráfico, tipo de transporte, urgencia de viaje, etc. para determinar la mejor ruta entre el punto de inicio del recorrido y el punto final.

En Siuhi y Mwakalonge (2016), se describen aplicaciones para procesar datos y sugerir rutas, tales como: Easy Route Finder, Runtastic Road Bike Tracker, Voyager Route Planner, Journey Pro, Drivieng Route Finder.

Por su parte, Salcedo y Battistuti (2013), desarrolla el manejo de rutas de transporte aplicando un algoritmo discreto con dependencia heurística, este algoritmo aplica la selección de la mejor ruta considerando algunos de los siguientes aspectos: tiempo de intercambio de los servicios de transporte, promedio de velocidad de cada transporte, fecha y hora en la cual fue realizado un viaje específico y tiempo de espera para cada línea de transporte.

Otro enfoque que trata el manejo de rutas de transporte se aprecia en Sun et al. (2016), quien utiliza una aplicación móvil la cual accede a datos proporcionados mediante cámaras y sensores, que registran el estado del tráfico en las calles de la ciudad, procesa la información y notifica las mejores rutas a los usuarios para desplazarse de un punto a otro en la ciudad.

Por otro lado, Caggiani et al. (2017), presenta un sistema de navegación que obtiene las mejores rutas de transporte, basado en un modelo con sistema embebido de inferencia difusa, el cual es accesado mediante una aplicación móvil que ofrece las mejores rutas de destino a ciclistas y personas que caminan al aire libre. Así mismo, la aplicación permite guardar y compartir rutas.

Otra solución es presentada por Gallup et al. (2015), que hace uso de una aplicación móvil desarrollada para manejar rutas de transporte, lo que permite definir la mejor ruta, en base a sus necesidades y las problemáticas que se presentan en la zona de selección.

Por otro lado, Harriehausen-Muhlbauer (2017), desarrollo la aplicación denominada WheelScout la cual considera factores como: inconvenientes de rampas, superficies desiguales, obstáculos, etc. Esta aplicación está enfocada principalmente para personas discapacitadas, cuyos recursos pueden ser configurados para ser utilizados con una silla de ruedas.

Así mismo Dharani et al. (2019), desarrollo una aplicación que maneja diferentes tipos de sensores inteligentes y capas de abstracción de información, para aplicaciones que manejan sistemas de navegación en una ciudad. La aplicación funciona en tiempo real procesando la información de los sensores y genera sugerencias para transporte en la ciudad de México, esta aplicación es denominada "Transit Colectivo y Metro", maneja la información de los medios de transporte para ayudar a las aplicaciones móviles en el proceso de desplazamientos en la ciudad. Algunas de las fuentes de información que coadyuvan en esta aplicación son: Maps - Aplicaciones Android en Google Play, TranzMate: info de bus y tren, Aplicaciones Android en Google Play.

Otro aspecto de interés es la inteligencia colectiva, esta

se sustenta en la participación de los miembros de un entorno particular, aportando su conocimiento sobre un dominio que se trata de resolver, en algunos casos cada miembro participa aportando sus experiencias y vivencias, tal es el caso reportado en In-Soo (2020), en donde los médicos de una comunidad abordan, desde su perspectiva, la interpretación de electrocardiogramas realizados a algunos pacientes. En este contexto, cada médico responde a los cuestionamientos expuestos en una aplicación web, indicando su forma de interpretar los electrocardiogramas. Cada respuesta es presentada a la comunidad en un plazo relativamente pequeño de tiempo. La suma de las interpretaciones va generando inteligencia colectiva sobre el dominio.

Por su parte Guilbeault y Centola (2020), aplica el proceso de reflexión basado en inteligencia colectiva, en donde demuestra que los fumadores responden mejor a los tratamientos anti-fumar en base a compartir la experiencia a través de una red social. Específicamente las personas que participan activamente reducen el riesgo de fumar hasta 10 veces más, comparado con las personas que afrontan de manera personal el superar el no fumar. Por tanto, la capacidad del impacto de la inteligencia colectiva estructurada a través de la red social ha demostrado su efectividad en este contexto. Los investigadores han demostrado recientemente que el rendimiento de un grupo de individuos, al resolver una tarea en equipo, está correlacionado con la inteligencia colectiva, lo cual motiva su estudio en diferentes ámbitos (Krafft, 2019). La inteligencia colectiva se puede concebir como un resumen del rendimiento de un grupo o equipo en un conjunto de tareas heterogéneas.

En el presente trabajo, se aplica el algoritmo  $A^*$  en combinación con la inteligencia colectiva para realizar dos funciones básicas, la primera: contar con la capacidad de obtener diferentes opciones de trayectoria en base a un punto de partida y un punto de destino. La segunda: modificar las

rutas de transporte en base a la aplicación de inteligencia colectiva, la cual está soportada por el comportamiento de los usuarios, estos pueden plantear modificaciones a las rutas y dichos cambios pueden ser votados por los mismos usuarios reflejando dichos cambios en las rutas en tiempo real.

El resto del artículo se estructura de la siguiente manera. La Sección II describe el algoritmo  $A^*$  en un entorno de planificación de rutas en el transporte público. La Sección III describe el desarrollo de la aplicación. La Sección IV muestra los resultados obtenidos y finalmente la Sección V presenta las conclusiones y trabajo a futuro.

## El Algoritmo $A^*$ en un entorno de planificación de rutas

La aplicación del algoritmo de búsqueda  $A^*$  funciona de manera óptima en problemas no lineales como el caso de búsqueda de nodos en rutas de transporte. Dada su eficiente búsqueda de interconexiones manipuladas por las distancias entre nodos y las interconexiones de estos en las rutas de transporte (Krawiec, 2012). El método de búsqueda  $A^*$  puede ser, dinámico no lineal y de tiempo continuo, y puede ser representado por la siguiente función:

$$\dot{x} = f(x, u) \quad (1)$$

Donde  $x$  es el vector de estado del sistema que contiene información de la configuración dinámica del sistema,  $u$  es el vector de entradas o acciones que se aplican al sistema. La función describe cómo  $x$  y  $u$  afectan a la dinámica del sistema, y  $\dot{x}$  es la derivada del vector de estado  $x$ , respecto al tiempo.

Aunque el modelo continuo (1) proporciona una representación compacta de un sistema, la planificación de una trayectoria en nodos sería bastante difícil debido a la natura-

leza discreta de los espacios implicados, las entradas parcialmente infinitas  $u \in U(x)$  que podrían aplicarse en el sistema en cualquier momento, y las integraciones que deben realizarse para cada transición de estado. Por lo tanto, el sistema puede ser representado en forma discreta como se muestra a continuación.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (2)$$

En esta ecuación,  $x$  sigue representando el vector de estados del sistema y contiene la colección de todos los estados posibles de  $x$ , de igual manera,  $u$  sigue representando una acción que puede aplicarse a un sistema.

Con esta formulación, es importante definir un espacio de estados o nodos  $X$  y un espacio de acciones  $U(x)$  para el sistema de interés. Lo cual puede ser aplicado en cualquier sistema de búsqueda, siempre que contenga todos los nodos del sistema. Por lo tanto, si nos referimos a la figura 1, el espacio de nodos sería  $X = \{1, 2, 3, 4\}$  y el espacio de acción en  $X(1) = 1$ , sería  $U(1) = \{R1, R2\}$ .

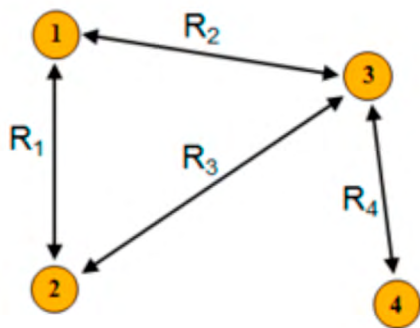


Figura 1. Ejemplo del problema del vendedor.

Si observamos en el ejemplo de un vendedor, el espacio de estados  $X$  no es más que una colección de lugares y el espacio de acción  $U(x)$  depende en gran medida del estado actual  $x$ . Desde el punto de vista computacional, se puede

implementar el proceso de búsqueda almacenando un conjunto de estado  $x \in X$ , que si el sistema es pequeño no habría problema, pero si es grande se vuelve complejo y difícil de manejar. Sería más eficiente un proceso de búsqueda, si un espacio de acciones pudiera ser independiente de  $x$  y se pudiera aplicar un conjunto común de acciones a cada estado.

En base a este argumento se puede formular un espacio de estado como una cuadrícula ordenada y el espacio de acción como independiente de  $x$ . En base a esta formulación se puede tener una matriz que permite almacenar fácilmente el espacio de estados en una matriz y sólo es necesario almacenar el conjunto de acciones.

Por lo tanto, aunque el algoritmo  $A^*$  es aplicable a un espacio de estado discreto y su espacio de acción, en el formato basado en la matriz. Por lo tanto, el espacio de estados puede definirse como:

$$X = \{x \in R^n \mid \dot{x} - x = r, x \in X\}, \text{ donde } r_i = \begin{cases} 0, & \text{si } x_i = \dot{x}_i \\ \Delta x_i, & \text{si } x_i \neq \dot{x}_i \end{cases}, i = 1, 2, 3, \dots, n \quad (3)$$

En donde  $x_i$  y  $r_i$  representan las componentes  $i$ th de  $x$  y  $r$  respectivamente, y  $\Delta x_i$  representa la resolución de la matriz para la  $i$ th coordenada. Del mismo modo, el espacio de acción puede definirse como los movimientos a cada estado vecino en la cuadrícula.

$$U = \{u \in R^n \mid u_i \in \{-\Delta x_i, 0, \Delta x_i\}, i = 1, 2, 3, \dots, n\} - \{0\} \quad (4)$$

En la Figura 2, se muestra una matriz simple de  $3 \times 3$  que presenta el espacio de estado y el espacio de acción.

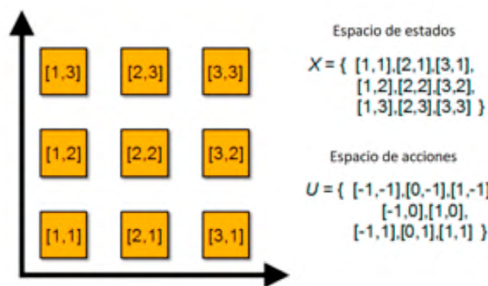


Figura 2. Ejemplo de matriz de estados y acción.

Con el espacio de estado y de acción basado en una cuadrícula, el problema de planificación del camino a través del espacio de estados puede ser implementado. La idea de la planificación de trayectorias o recorridos implica encontrar una trayectoria óptima a través del espacio de estados a partir de un estado inicial hacia un estado objetivo bajo la formulación  $xI \square X$  para lograr un estado meta  $xG \square X$  (Michel, 2006).

## Diseño de la aplicación

Se construyó un prototipo para aplicar el algoritmo  $A^*$  en combinación con la inteligencia colectiva para la planificación de rutas por parte de los usuarios. Para esto se utilizó la metodología en espiral, la cual se compone de diversas iteraciones y en cada iteración se va obteniendo un prototipo mejorado hasta llegar a la aplicación final. La Figura 3, muestra un diagrama general del prototipo, donde se observa del lado izquierdo a los usuarios que se conectan con la aplicación móvil, posteriormente el servidor que soporta la aplicación "Aplicación inteligencia colectiva de trazo de rutas", responde en base a lo que se programó para la navegación entre rutas de la ciudad.

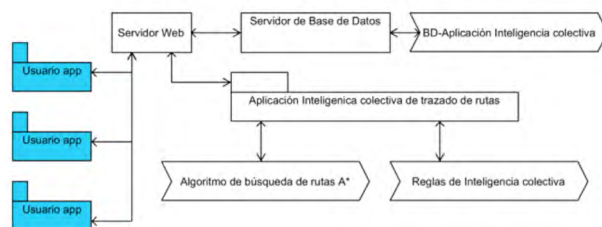


Figura 3. Diagrama principal de la aplicación de trazado de rutas con inteligencia colectiva.

La aplicación tiene embebida las acciones propias para administrar y manejar los datos de la aplicación, así mismo la interacción con el API de Google Maps. Con la combinación de estas plataformas se da respuesta a diferentes acciones que se tienen implementadas y son utilizadas por los usuarios mediante el uso de sus celulares, previa instalación de la aplicación web (App). Las diferentes acciones a las que responde la aplicación están documentadas en base a diagramas de clase, interfaces de usuario, diagrama relacional de base de datos y diagramas de flujos de datos entre los componentes que conforman la aplicación.

La aplicación se realizó en iteraciones en donde la primera iteración creo funcionalidad básica y planteo retos, la siguiente implementó los retos de la previa y creo nuevas funcionalidades, hasta terminar con un prototipo funcional en la última iteración. Para la primera iteración se analizaron los requerimientos, en base a estos, se creó una primera versión de base de datos para almacenar los datos básicos que hace uso la aplicación. En la Figura 4, se presentan las entidades básicas: usuario, ruta, polilinea, nodo y enlace. Algunas de las funcionalidades que se implementaron en esta interacción tienen que ver con las rutas y los nodos que forman dichas rutas, para cada uno de estos dos elementos se puede crear una nueva ruta o nodo, modificar, unir y eliminar una ruta o nodos, además también puede modificarse el cambio de dirección de las rutas. Cada una de estas funcionalidades repercute en el mapa de navegación de la aplicación.

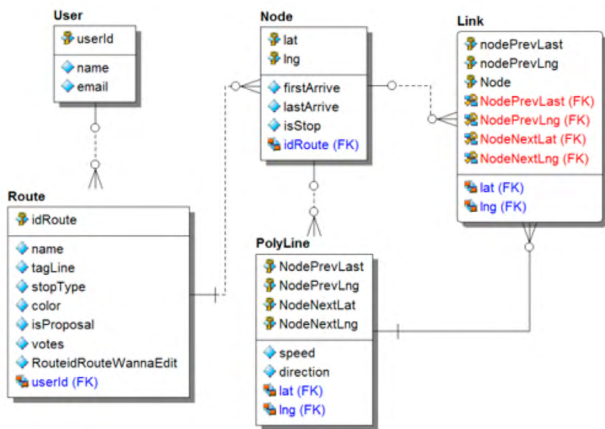


Figura 4. Esquema lógico de rutas.

Para el manejo de la funcionalidad de la aplicación en el lado del servidor, así como implementar las funcionalidades, ya antes mencionadas, se desarrollaron las siguientes clases: `NewRouteAttributesActivity`, `DrawOnMapActivity`, `Route`, `EditOBusPolyDialog`, `PostCaller`, `ToolButton` y `ObusPolyline`. También se realizaron diferentes interfases de usuarios para interactuar con la aplicación montada en el servidor web. La primera interfaz creada para interactuar con la aplicación se presenta en la Figura 5, en esta se pueden visualizar dos pantallas, en la primera se registra un nuevo recorrido o ruta en donde se le asignan sus datos básicos y en la segunda se traza la ruta en base a los nodos que representan las paradas de la ruta que se está registrando.



Figura 5. Representación de un recorrido y sus paradas.

La funcionalidad de la aplicación se basa en el envío de datos y procesamiento de los mismos, incluyendo la interacción con la base de datos y respuesta del servidor a los usuarios. Para el manejo del flujo de datos en la aplicación se desarrollaron diferentes diagramas de secuencia, la figura 6 presenta el diagrama de secuencias de la primera iteración, en donde se aprecian las diferentes validaciones que se realizan por parte de la aplicación.

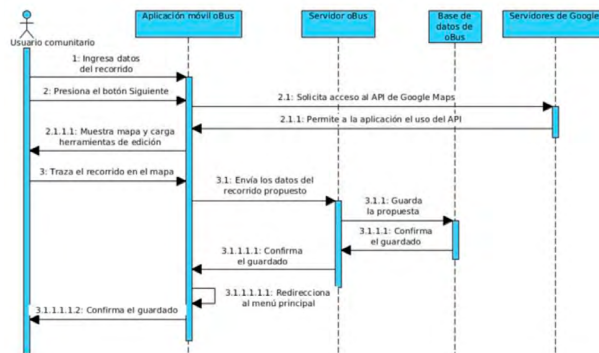


Figura 6. Diagrama de secuencia de la primera iteración.

En el diagrama de secuencia de la primera iteración se pueden apreciar las diferentes acciones que se implementaron en la aplicación para crear la funcionalidad "registro de recorrido" propuesto por el usuario con privilegios de modificación. Para cada una de las funcionalidades se tiene un diagrama de secuencia, un diagrama de clases e interfaces de usuario, que soportan dicha funcionalidad.

Para la segunda iteración, se implementó la clase `SignUpRequestSender`, mientras que en la base de datos se agregó la entidad `User`, `polyline` y `WalkingLink` con sus atributos básicos. Se creó la forma `SignUpRequest.php` que permite la inserción del usuario en el sistema y le asigna un rol básico. Se crearon los procesos que permiten crear cambios en las rutas de polilíneas, los cuales permiten la interacción entre dos rutas con un nodo de intersección. Estas nuevas inser-



ciones en la aplicación pueden provocar cambios en algunas de las clases, así como la creación de los distintos diagramas de secuencia que describen el comportamiento de los datos entre los usuarios y la aplicación web, además de la interacción con la base de datos.

En la tercera iteración, se creó el proceso para cambiar el estatus de los usuarios en base a reglas creadas por tiempo, participación y veracidad confirmada por otros usuarios. Se modificó la clase que controla los usuarios agregando la funcionalidad LoginActivity, la cual permite controlar las acciones que realiza cada usuario en su interacción con la aplicación y determinar posteriormente su capacidad para ser cambiado de nivel básico a nivel con permisos de generar sugerencias de cambio. Los usuarios podrán modificar su estatus o nivel en base a las votaciones de otros usuarios, estas votaciones son registradas por la clase y almacenadas en la base de datos, una vez registradas son utilizadas por un proceso que las contabiliza y evalúa; cuando cumplen una normativa de veracidad cambian el estatus del usuario y puede pasar a nivel editor. Las votaciones también pueden ser modificadas por el administrador directamente.

Para la cuarta iteración, se agregan funcionalidades a la clase de control de recorridos en base al nuevo rol de usuario que podrá: proponer, editar y borrar un recorrido, incluyendo los flujos de acciones a realizar para cada cambio. Algunas de las acciones tienen su repercusión en las interfaces de usuarios que se presentan en la Figura 7.



Figura 7. Interfaz de modificaciones a ruta por usuario no básico.

En esta iteración se realizaron cambios en las clases: Route, DrawOnMapActivity, ViewOfficialRoutesActivitiy y PostCaller, además se crearon los archivos: getOfficialRoutes.php, getRouteInfo.php, saveRequest.php y getNodesAndPolysFromRoutes.php, que agregan funcionalidad a la aplicación web.

Para la quinta iteración, se presenta a los usuarios una lista de propuestas de modificaciones de recorridos y se pueden generar votaciones de aceptación o rechazo a dichos cambios. Cada uno de los posibles cambios tiene la posibilidad de edición, creación y eliminación de un recorrido. La aplicación responde informando sobre las votaciones en base a la colectividad de los usuarios. Las principales clases que participaron en esta iteración fueron: Route, ViewRequestActivity, GetRequestSender, RoutesViewAdapter, DeleteRouteDetailsActivity, EditRouteDetailsActivity NewRouteAttributesActivity, EditNodeDialog y EditOBusPolyDialog. Estas clases fueron procesadas con los siguientes archivos: getRequestRoutes.php, getNodesAndPolysFromRoutes.php y SendVote.php. De igual forma se desarrollaron los diferentes diagramas de comportamiento y secuencia de actividades para cada una de las funcionalidades implementadas.

La sexta iteración realiza las acciones para manejar las consultas de los viajes, en donde se plantean los nodos de inicio y fin de recorrido y estos datos se procesan por el algoritmo A\*. El algoritmo toma como base los pesos de los recorridos y las intersecciones entre nodos de las rutas participantes. Cuando el usuario elige un nodo de inicio y un nodo de destino, se realiza la búsqueda entre ambos puntos determinando la mejor opción de ruta, dentro de un grupo de opciones de ruta.

Al elegir uno de los recorridos, la aplicación responderá dibujando en el mapa dicha ruta. Para esta interacción par-



ticiparon las clases: Node, ObusPolyline y SetMarkerDialog. Aunadas a las demás clases que forman toda la aplicación y que son: DrawOnMapActivity, EditObusPoliDialog, newRouteAttributesActivity, EditNodeDialog, PostCaller, toolButton, SignUpActivity, SignUpRequestSender, LoginRequestSender, GetRequestRouteSender, Route, UserAccount, OfficialRoutesListAdapter, OfficialRoutesDetailsActivity, NewRouteDetailsActivity, EditRouteAttributeActivity, LookOfficialRouteActivity, DeleteRouteDetailsActivity y EditRouteDetailsActivity.

Las peticiones a la aplicación fueron procesadas por los siguientes archivos: takeATrip.php y getNodesAndPolysFromRoutes.php. En la figura 8 se presenta el diagrama de secuencia que expresa el comportamiento de la funcionalidad consultar viaje. El Usuario Comunitario ingresa una consulta a la interfaz gráfica, la cual responde y realiza una solicitud de acceso al API de Google Maps, este responde y regresa una vista parcial del mapa a la interfaz en donde presenta la ruta trazada del recorrido, la cual es mostrada al usuario final. Posteriormente, el usuario ingresa datos de inicio y fin de recorrido, estos son utilizados para realizar el proceso de cálculo de trayecto en la parte de oBus, la cual obtiene todos los nodos de conexión almacenados en la base de datos de cada ruta que participa en dicha solicitud.

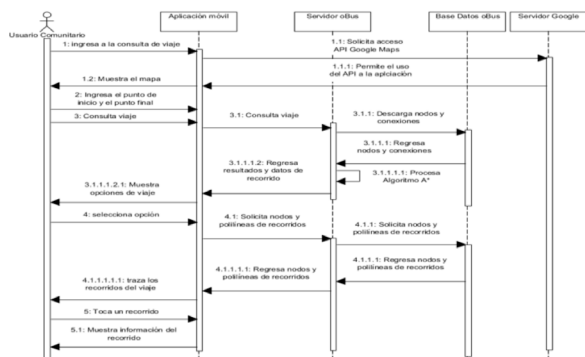


Figura 8. Diagrama de secuencias para consultar viaje.

En la Figura 9, se presenta una secuencia de imágenes que describen el proceso de consulta de una ruta.

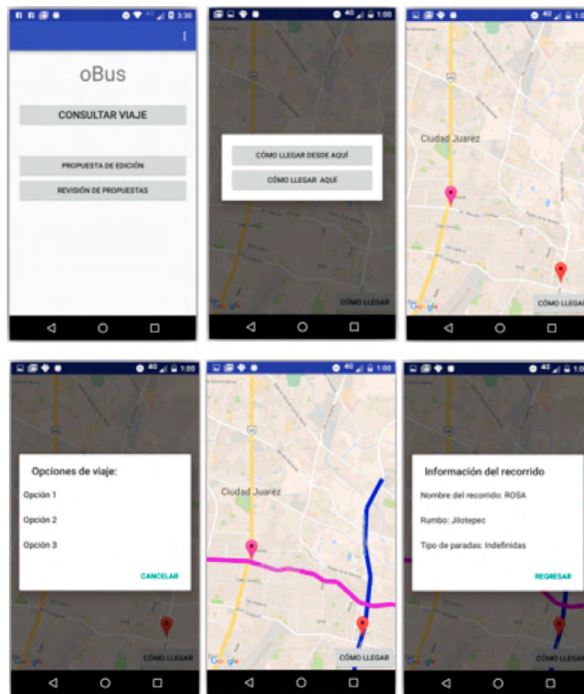


Figura 9. Secuencia de pantallas de aprobación o rechazo de propuesta de recorrido.

En la Figura 9, se aprecia la consulta de un viaje, la secuencia se presenta en imágenes de izquierda a derecha, además se pueden registrar y modificar rutas en tiempo real. Así mismo, se presenta la ubicación actual del usuario en el mapa y la distancia a la ruta. Cuando se registra el inicio del viaje, se presentan las opciones generadas por el algoritmo A\* y una vez elegida una opción, la aplicación presenta el trazado de la ruta elegida y el proceso de avance de la misma del autobús en la ruta.

## Resultados y discusión

Se obtuvo una aplicación móvil con acceso al API de Google utilizando una metodología basada en iteraciones, que la hacen funcional y permite registrar rutas de transpor-

te público con la interacción de usuarios previamente registrados. Se implementó el algoritmo A\* para la funcionalidad de la aplicación, el cual se utiliza del lado del servidor para dar respuesta a los usuarios en el proceso de selección de rutas y generar opciones de destino, en base a los registros de nodos que se tienen para cada una de las rutas registradas.

La aplicación móvil es capaz de generar cambios en las rutas registradas, en base a sugerencias de los usuarios en tiempo real. Mediante un proceso de votación de usuarios registrados aplicando la inteligencia colectiva se generan los cambios en base a las sugerencias proporcionadas por los usuarios, las cuales se reflejan en la Interfaz Gráfica de la aplicación móvil.

Se probó la aplicación con un registro de cinco rutas considerando que las rutas se intersectan en algunos nodos, tomando como base el inicio y fin del recorrido. Se generaron resultados esperados en cuanto a la respuesta de las posibles sugerencias proporcionadas por el algoritmo A\*, el cual generó sugerencias de recorrido entre los puntos de partida y destino.

Se procedió a modificar una ruta por parte de un usuario con permisos, quien cambió la posición de un nodo, y dicho cambio fue validado por el administrador. Una vez realizado el cambio se aplicó nuevamente la búsqueda de la ruta, y se generaron los mismos resultados de inicio y fin de la ruta.

Se realizó y aplicó una encuesta piloto a estudiantes, docentes y ciudadanos, usuarios de cinco distintas rutas establecidas en la ciudad, las preguntas estuvieron enfocadas en la funcionalidad de la aplicación, la usabilidad del software y si harían uso de la aplicación si estuviera en función, cuyos resultados arrojan un índice de satisfacción del 85% para los temas encuestados.

La aplicación permite que los usuarios, desde su celular, aporten conocimiento útil en tiempo real sobre el estado del transporte público, tal que, otros usuarios puedan enterarse y puedan planificar su ruta para llegar a su destino en el menor tiempo posible. La aplicación considera las rutas representadas por nodos y verifica si estas tienen puntos de intersección una vez que se hace una solicitud de un punto de partida a un punto de destino, el algoritmo A\* toma estos puntos y los aproxima al nodo más cercano, estos nodos serán el punto inicial y el punto final que el algoritmo toma para realizar los cálculos de los recorridos, las distancias entre nodo y nodo representan los pesos o coste de recorrido. Con los datos de búsqueda registrados se eligen las mejores opciones y se generan las sugerencias de recorrido, si existe una o más sugerencias se presentan en el mapa de google en la aplicación del celular. Los pesos o coste se determinan por la distancia de separación y las sugerencias generadas por los usuarios en sus reportes de colectividad.

Esta aplicación se probó en Ciudad Juárez, Chihuahua, con un registro de 16 rutas hipotéticas y más de 250 nodos. Se simularon pruebas de modificación de pesos entre nodo y nodo simulando accidentes y bloqueos de paso, modificando la ruta con alteración de recorrido, se obtuvieron resultados favorables de comportamiento en el software ya que al realizar consultas de recorrido estas alteraciones cambian la sugerencia de recorrido en base a las modificaciones.

## Conclusiones

Este trabajo explora el uso del algoritmo A\*, el cual permite encontrar la ruta óptima en combinación con la inteligencia colectiva, para que un usuario pueda planificar su ruta en el sistema de transporte público de la ciudad. Para esto, se construyó una aplicación móvil para sugerir la mejor ruta,

considerando la participación ciudadana con aporte de sugerencias. Los resultados indican que el uso del algoritmo A\* junto con la inteligencia colectiva son una buena alternativa en la planificación de rutas en un sistema de transporte público de una ciudad.

Como trabajo futuro se espera instalar la aplicación en trabajo real y probar con rutas y usuarios reales registrando las funcionalidades de la aplicación. Además, se propone aplicar procedimientos de usabilidad, mejoras de rendimiento, y diseño de la interfaz gráfica de la aplicación móvil, la cual se implementó con los elementos mínimos necesarios para su funcionamiento básico.

## Referencias

- Bellman, R. (1958). On a routing problem. *Quarterly of Applied Mathematics*, 16(1), 87–90. <https://doi.org/10.1090/qam/102435>
- Caggiani, L., Camporeale, R., Ottomanelli, M., y Bari, P. (2017). Model for a Bike-Sharing Mobile Application. 645–650.
- Dharani, A., Vijaylakshmi, M. N., y Narasimha Swamy, S. (2019). Mobile as a sensor in intelligent transportation system for street route. 2018 International Conference on Computing, Power and Communication Technologies, GUCON 2018, 138–141. <https://doi.org/10.1109/GUCON.2018.8674986>
- Dijkstra, E. W. (1959). A Note on Two Problems in Connexion with Graphs. 271, 269–271.
- Gallup, J., Lamothe, S. N., y Gallup, A. (2015). Enhancing Transportation Education Using Mobile Devices and Applications. *TEACHING Exceptional Children*, 48(1), 54–61. <https://doi.org/10.1177/0040059915580027>
- Guilbeault, D., y Centola, D. (2020). Networked collective intelligence improves dissemination of scientific information regarding smoking risks. In *PLoS ONE* (Vol. 15, Issue 2). <https://doi.org/10.1371/journal.pone.0227813>
- Harriehausen-Muhlbauer, B. (2017). Communicating with Wheelscout via voice: Speech technology in a mobile navigation app computing barrier-free routes. *FTC 2016 - Proceedings of Future Technologies Conference*, December, 488–493. <https://doi.org/10.1109/FTC.2016.7821652>
- In-Soo, kim y M. (2020). Mobile Application Can Now Assist to Diagnose Arrhythmias with collective intelligence. *Nature*, 388, 1–14.
- Krafft, P. M. (2019). A Simple Computational Theory of General Collective Intelligence. *Topics in Cognitive Science*, 11(2), 374–392. <https://doi.org/10.1111/tops.12341>
- Krawiec, B. M. (2012). A\*-Based Path Planning for an Unmanned Aerial and Ground Vehicle Team in a Radio Repeating Operation [Virginia Polytechnic Institute]. <https://vtechworks.lib.vt.edu/handle/10919/32545>
- Michel, S. (2006). *Planning Algorithms* (1st ed.). Cambridge University Press.
- Salcedo, J. S., y Battistuti, O. C. (2013). Heuristic for unscheduled public transport navigation system. *Proceedings of the European Conference on Knowledge Management, ECKM*, 2, 607–615.
- Siuhi, S., y Mwakalonge, J. (2016). Opportunities and challenges of smart mobile applications in transportation. *Journal of Traffic and Transportation Engineering (English Edition)*, 3(6), 582–592. <https://doi.org/10.1016/j.jtte.2016.11.001>
- Sun, N., Han, G., Duan, P., y Tan, J. (2016). A Global and Dynamic Route Planning Application for Smart Transportation. *Proceedings - 2015 1st International Conference on Computational Intelligence Theory, Systems and Applications, CCITSA 2015*, 203–208. <https://doi.org/10.1109/CCITSA.2015.43>