



Resumen.

Este proyecto fue desarrollado en MATLAB, este implementa el algoritmo de búsqueda A estrella para encontrar la ruta más corta entre dos puntos en un mapa, el primer punto es el que se va a ir actualizando con las coordenadas obtenidas en el teléfono (tiempo real) y el segundo punto es el destino que ya está definido. El nodo objetivo lo puse en la salida de estacionamiento de ESCOM, y el nodo de inicio es al fondo y encontrara la ruta más corta haciendo uso del algoritmo. Este comienza definiendo un grafo que representa los posibles caminos entre los nodos en un mapa, donde cada nodo es una ubicación geográfica y las conexiones entre nodos son los posibles caminos. Entra en un bucle infinito que actualiza continuamente la ubicación actual del usuario y calcula la ruta más corta al destino. Utiliza la función encontrar Nodo Mas Cercano (), que calcula la distancia euclidiana entre la ubicación actual y cada nodo, devolviendo el nodo más cercano. Luego, se aplica el algoritmo A estrella a través de la función a _ estrella (), que busca la ruta más corta en el grafo desde el nodo inicial hasta el destino. Este algoritmo utiliza una función heurística para estimar el costo de viajar de un nodo a otro, y mantiene dos listas de nodos: una lista de nodos abiertos (aún no explorados) y una lista de nodos cerrados (ya explorados). El algoritmo continúa explorando los nodos hasta que encuentra el destino o hasta que no quedan nodos abiertos. Una vez que se encuentra el destino, la función reconstruir Ruta () toma la lista de padres de cada nodo (que se actualiza durante la ejecución del algoritmo A*) y reconstruye la ruta desde el destino hasta el nodo inicial. Finalmente, la ruta encontrada se grafica en el mapa y el código pausa durante 5 segundos antes de volver a calcular la ruta, permitiendo que la ubicación del usuario se actualice. En resumen, este código proporciona una implementación del algoritmo A* para la navegación en un mapa geográfico, útil para aplicaciones de navegación por GPS.

Palabras Clave.

Algoritmo A*, búsqueda de ruta, grafo, nodo, función heurística, distancia euclidiana



Introducción.

Este proyecto implementa el algoritmo de búsqueda A* para encontrar la ruta más corta entre dos puntos en un mapa. El algoritmo A* es un algoritmo de búsqueda informada que utiliza una función heurística para estimar el costo de viajar de un nodo a otro. Esto se hace con ayuda del teléfono para obtener la latitud y longitud donde se encuentra y así obtener el nodo desde donde se calculará la ruta más corta hacia el nodo objetivo.

Desarrollo.

Para desarrollar este programa elegí el lenguaje de programación de Matlab, esto por las funciones que ofrece para usar los sensores de mi dispositivo móvil, para esto realicé 3 funciones, una que es la principal que es donde se especifican los nodos y sus coordenadas, otra que calcula el nodo mas cercano comparando la distancia de donde se encuentra con cada uno de los nodos, el A* que calcula la ruta mas optima al destino, y otra que es la de reconstruir la ruta.

Definir un grafo que representa los posibles caminos entre los nodos en un mapa.

Entrar en un bucle infinito:

- Actualiza la ubicación actual del usuario.

- Utiliza la función encontrarNodoMasCercano() para encontrar el nodo más cercano a la ubicación actual.

- encontrarNodoMasCercano():

 - Para cada nodo en el mapa:

 - Calcular la distancia euclidiana entre la ubicación actual y el nodo

 - Devolver el nodo con la menor distancia

- Aplica el algoritmo A* para encontrar la ruta más corta desde el nodo inicial hasta el destino.

- A*:

 - Mientras la lista abierta no esté vacía:

 - Tomar el nodo con el menor costo f en la lista abierta

 - Si este nodo es el destino:

 - Terminar la búsqueda

 - Mover el nodo de la lista abierta a la lista cerrada

 - Para cada vecino del nodo:



Instituto Politécnico Nacional IPN
Escuela Superior de Computo ESCOM
Practica 1 Reporte
Torres Abonce Luis Miguel
luiseishon9@gmail.com
Si el vecino está en la lista cerrada:



Continuar al siguiente vecino

Si el vecino no está en la lista abierta o el nuevo camino es más corto:

Actualizar el costo g del vecino

Actualizar el costo f del vecino

Actualizar el padre del vecino

Si el vecino no está en la lista abierta:

Agregar el vecino a la lista abierta

Una vez que se encuentra el destino, utiliza la función `reconstruirRuta()` para reconstruir la ruta desde el destino hasta el nodo inicial.

`ReconstruirRuta()`:

Inicializar la ruta con el nodo destino

Mientras el nodo tenga un padre:

Agregar el padre a la ruta

Mover al nodo padre

Devolver la ruta

Grafica la ruta encontrada en el mapa.

Pausa durante 5 segundos antes de volver a calcular la ruta.

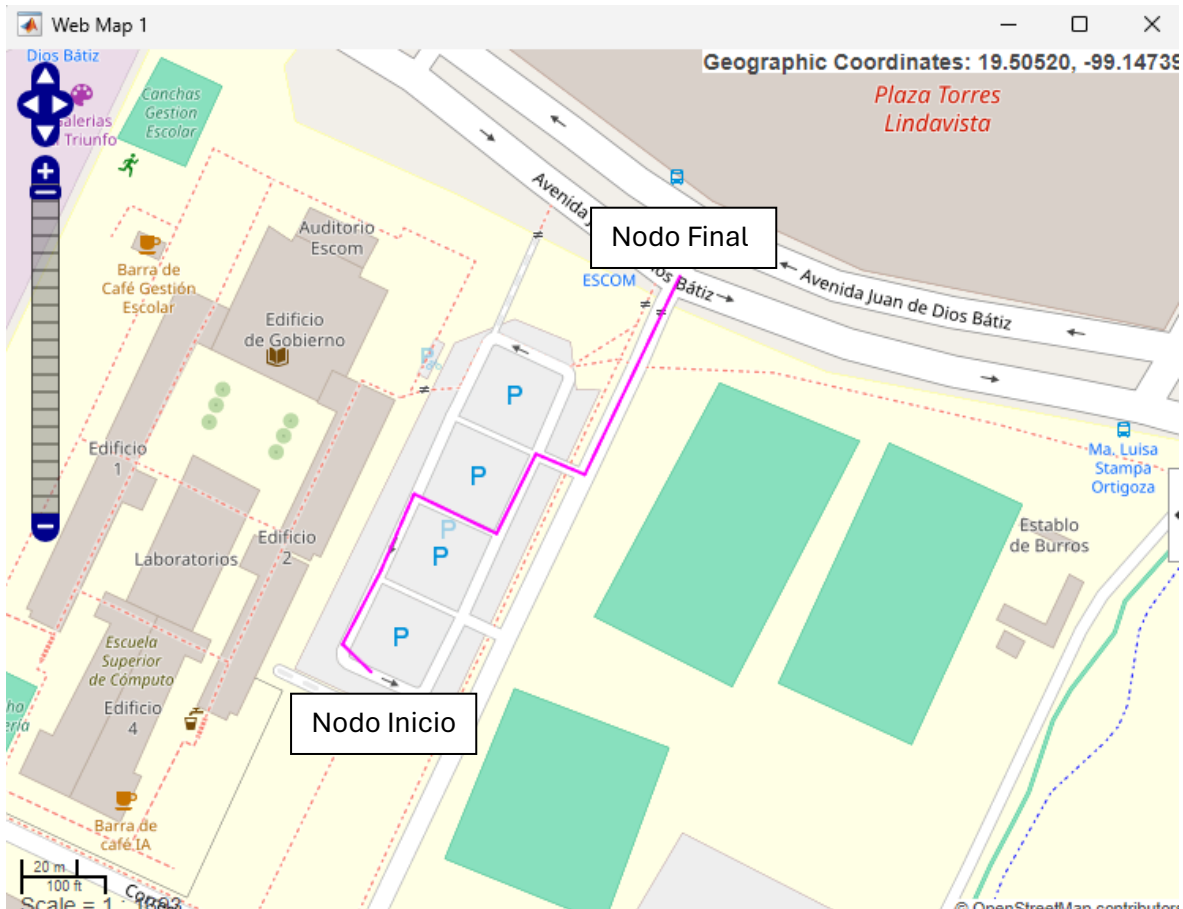
Análisis.

El algoritmo A* es eficiente y garantiza encontrar la ruta más corta si existe una. Sin embargo, la eficiencia del algoritmo depende en gran medida de la calidad de la función heurística utilizada. En este proyecto, se utiliza la distancia euclidiana como función heurística, que es una buena opción para problemas de navegación en un espacio geográfico.



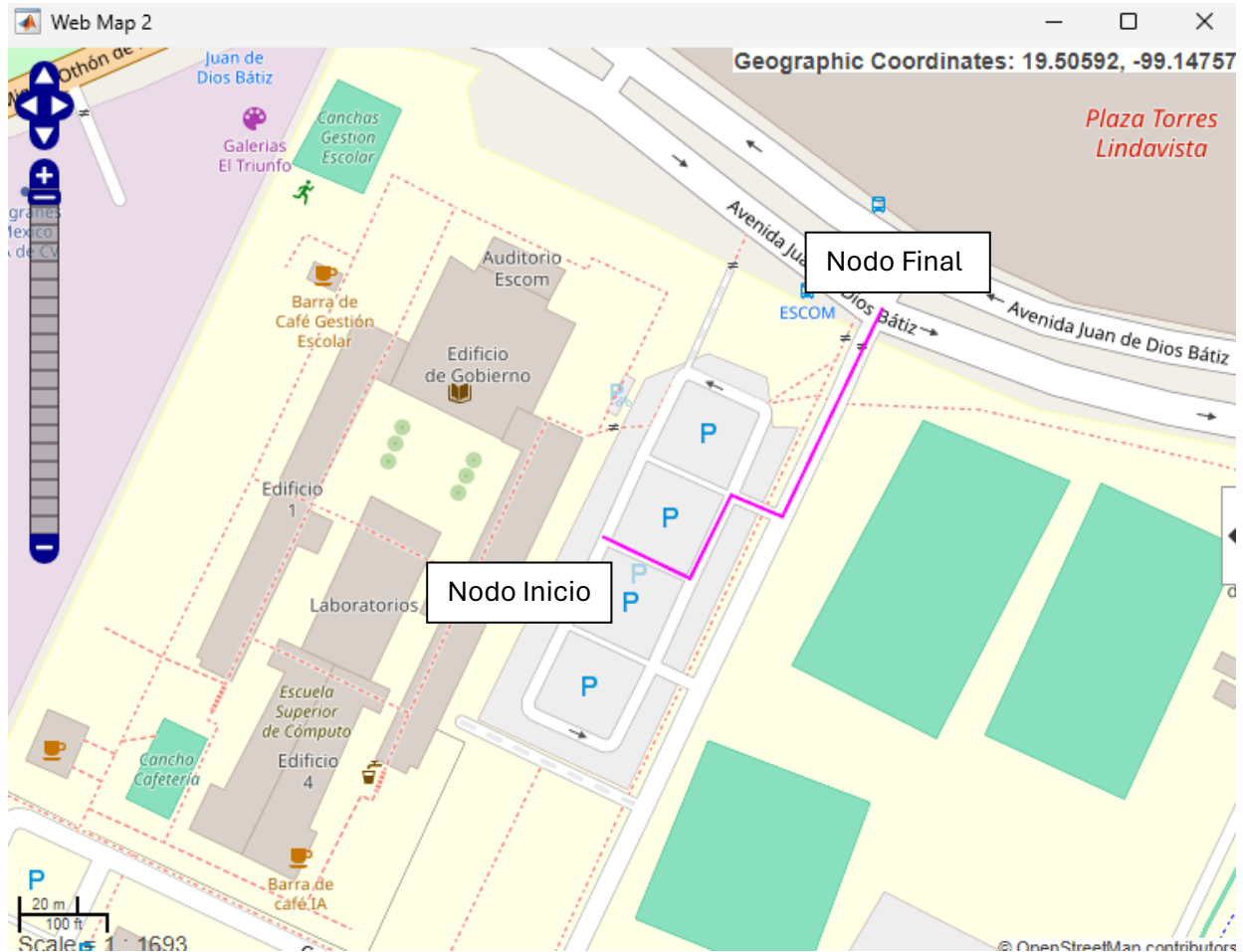
Resultados.

En esta captura de pantalla se puede observar que recorrió los nodos hasta el camino mas corto desde el nodo de inicio hasta el nodo final utilizando A*.





En esta otra captura se puede observar desde otro punto la ruta más optima.



Conclusión.

Este proyecto demuestra cómo se puede implementar el algoritmo A* para resolver problemas de navegación en un mapa geográfico. El código podría ser utilizado en una aplicación de navegación por GPS, por ejemplo. La parte que mas se me complico fue la de decidir en qué lenguaje de programación realizarlo ya que había elegido Python, pero a la hora de realizarlo se me dificulto el configurar el servidor, por lo que decidí mejor usar los recursos compartidos por el profesor y realizarlo en Matlab, lo complicado de esto fue el programar la parte del algoritmo a* ya que en un principio no actualizaba la ruta y daba en diagonal la ruta lo cual era incorrecto.



Instituto Politécnico Nacional IPN
Escuela Superior de Computo ESCOM
Practica 1 Reporte
Torres Abonce Luis Miguel
luiseishon9@gmail.com



Referencias.

Mapping toolbox. (s. f.). <https://la.mathworks.com/products/mapping.html>

Plot Geographic Data on a Map in MATLAB. (s. f.). MATLAB.

<https://la.mathworks.com/videos/plot-geographic-data-on-a-map-in-matlab1545831202291.html>

Crear mapas con datos de latitud y longitud - MATLAB & Simulink - MathWorks América

Latina. (s. f.). https://la.mathworks.com/help/matlab/creating_plots/plot-in-geographic-coordinates.html