
Documentación del Proyecto: Algoritmo A* para Rutas Óptimas

1. Descripción del proyecto

Este proyecto busca implementar el algoritmo A* para encontrar rutas óptimas en una malla de cuadros que simula las distintas áreas de una ciudad. El mapa puede configurarse estableciéndose un punto de origen y un punto final. Además, se pueden bloquear rutas y visualizar otras rutas.

2. Características principales

- *Uso del Algoritmo A*:* El algoritmo encuentra la ruta más corta considerando movimientos en 8 direcciones, puede moverse en vertical, horizontal, y diagonal.
- **Interfaz gráfica:** Permite configurar el mapa de forma sencilla con varios clicks.
- **Visualización:** Se muestran distintos colores y animaciones para diferenciar inicio, metas, bloqueos y ruta.

3. Requisitos del sistema

- **Python**
- Dependencias:
 - Flask
 - Librerías estandar (heapq, json...)

4. Instalación

Tras descargar el proyecto, seguir los siguientes pasos:

- **Windows:** Ejecutar como administrador *install.bat* para instalar Python y Flask.
- **macOS/Linux:** El instalador no se ha codificado todavía. Mientras tanto:
apt install python
pip install flask

5. Ejecución

Forma automática:

Ejecutar como administrador el archivo *RutasA.bat*.

Forma manual:

1. Abrir un terminal/cmd (Linux/Windows) en la carpeta del proyecto.
2. Escribir: *python app.py*
3. Abrir el navegador y dirigirse a *localhost:5000* o *http://127.0.0.1:5000*.

6. Uso de la aplicación

Cuadrícula: Representa el mapa de la ciudad.

- Punto verde: Punto de inicio.
- Punto rojo: Punto de meta.
- Punto negro: Zonas bloqueadas.
- Punto amarillo: Ruta óptima.

Botones: Configuran y ejecuta la aplicación.

- Establecer inicio: Selecciona el punto de partida.
- Establecer meta: Selecciona el destino.
- Bloquear celda: Bloquea zonas para que la aplicación lo use.
- Calcular ruta: Ejecuta el algoritmo A*.
- Reiniciar todo: Limpia el mapa.
- Deshacer: Retrocede una acción hacia atrás (bloquear).
- Rehacer: Avanza una acción hacia adelante (una vez se ha deshecho).

7. Estructura del proyecto

/ProyAlgoritmoA

```
├── data/                # Archivos de instalación
|   └── python-3.13.0-amd64.exe
├── instalar_dependencias.bat # Script para instalar dependencias (Windows)
├── Rutas.bat            # Script para ejecutar la aplicación (Windows)
├── app.py               # Backend (Flask y algoritmo A*)
├── templates/          # Plantillas HTML
|   └── index.html
├── static/              # Archivos estáticos
|   ├── styles.css      # Estilos CSS
|   └── scripts.js       # Lógica del frontend
├── doc/                 # Documentación del proyecto
|   ├── Dependencias.md  # Detalles de las dependencias
|   ├── Memoria.md       # Explicación técnica del proyecto
|   └── documentación.pdf # Documentación en formato PDF
```

8. Detalles técnicos

*Algoritmo A:**

- Heurística: Se ha utilizado la distancia de Chebyshev, que permite calcular la distancia entre dos puntos mediante 8 direcciones distintas.
- Estructuras de datos: Se han utilizado distintas estructuras para guardar datos entre frontend y backend.
 - Heap: Gestiona nodos abiertos.
 - Diccionarios: Almacenan rutas y el coste del movimiento.

Flask

Rutas:

- **/:*** Interfaz principal.
- ***/configurar:** Recibe la configuración del mapa.
- **/calcular_ruta:** Ejecuta el algoritmo A*.
- **/reset:** Reinicia el estado del mapa.

9. Posibles mejoras

El algoritmo realiza el cálculo buscando la ruta más corta mediante el número de cuadros. Tras encontrar el menor número de cuadros, prioriza el camino de forma vertical.

Es decir, si establecemos un inicio y meta en la misma línea horizontal (no en el mismo recuadro), el algoritmo calculará el camino más rápido saliendo de la línea.

Por tanto, una posible mejora sería realizar los caminos más sencillos, es decir, con menos movimientos de líneas.

10. Contacto

Alejandro R. González Jurado

([Gmail](#))

([GitHub](#))