



 Planeta Formación y Universidades

CORPORACION UNIVERSITARIA IBEROAMERICANA

FACULTAD

INGENIERIA DE SOFTWARE

INTELIGENCIA ARTIFICIAL

PROFESOR:

JORGE CASTAÑEDA

AUTOR:

JEFREE MATEO HERNANDEZ RAMIREZ

100118344 - jhern133@ibero.edu.co

JUAN DAVID NEIRA BELLO

100112092 - jneirabe@ibero.edu.co

1. Introducción	2
-----------------------	---

2. Objetivo	3
3. Configuración del Entorno.....	3
4. Pruebas Realizadas	3
Prueba 1: Ruta desde A hasta E	3
Entrada:	3
Salida esperada:.....	4
Salida obtenida:	4
Prueba 2: Ruta desde B hasta D	4
Entrada:	4
Salida esperada:.....	4
Salida obtenida:	4
Prueba 3: Ruta desde C hasta A	5
Entrada:	5
Salida esperada:.....	5
Salida obtenida:	5
5. Conclusiones.....	5
6. Instrucciones para la Ejecución	5
7. Recomendaciones Futuras	6

1. Introducción

Este documento presenta las pruebas realizadas sobre el sistema inteligente de búsqueda de la mejor ruta en un sistema de transporte masivo. La implementación utiliza el algoritmo A* para encontrar el camino óptimo entre dos puntos dentro de una red de estaciones conectadas.

2. Objetivo

El objetivo de este sistema es calcular la mejor ruta entre dos estaciones en un sistema de transporte masivo, minimizando la distancia o el costo del trayecto mediante una estrategia de búsqueda heurística.

3. Configuración del Entorno

- **Lenguaje:** Python 3.12
- **Librerías utilizadas:** heapq
- **Plataforma:** Windows 10/11, Linux, macOS
- **Editor recomendado:** Visual Studio Code / PyCharm / Jupyter Notebook
- **Formato de entrada:** Grafo representado con diccionarios en Python
- **Formato de salida:** Lista con la secuencia óptima de estaciones

4. Pruebas Realizadas

A continuación, se presentan distintas pruebas del sistema para evaluar su funcionalidad y precisión.

Prueba 1: Ruta desde A hasta E

Entrada:

Red de transporte representada como un grafo:

A - B (1), A - C (4)

B - C (2), B - D (5)

C - D (1)

D - E (3)

Inicio: A Destino: E

Salida esperada:

Ruta óptima: ['A', 'B', 'C', 'D', 'E']

Salida obtenida:

Mejor ruta encontrada: ['A', 'B', 'C', 'D', 'E']

✅ **Prueba exitosa: La ruta óptima coincide con la esperada.**

Prueba 2: Ruta desde B hasta D

Entrada:

Inicio: B Destino: D

Salida esperada:

Ruta óptima: ['B', 'C', 'D']

Salida obtenida:

Mejor ruta encontrada: ['B', 'C', 'D']

✅ **Prueba exitosa: La ruta óptima es correcta.**

Prueba 3: Ruta desde C hasta A

Entrada:

Inicio: C Destino: A

Salida esperada:

Ruta óptima: ['C', 'B', 'A']

Salida obtenida:

Mejor ruta encontrada: ['C', 'B', 'A']

✅ **Prueba exitosa: Se encuentra la mejor ruta inversa.**

5. Conclusiones

El sistema ha demostrado ser efectivo para encontrar la mejor ruta en distintos escenarios. La implementación con el algoritmo A* garantiza una búsqueda eficiente y óptima. Se pueden ampliar las pruebas agregando más nodos y condiciones como tiempos de espera o costos adicionales.

6. Instrucciones para la Ejecución

Para ejecutar el código, siga estos pasos:

1. **Instalar Python 3.12** (o versión compatible)
2. **Guardar el código en un archivo** llamado transporte.py
3. **Abrir la terminal o consola** y ubicarse en la carpeta donde guardó el archivo.
4. **Ejecutar el siguiente comando:**

```
python transporte.py
```

5. **Analizar la salida:** Se imprimirá la mejor ruta calculada.

7. Recomendaciones Futuras

- Extender el sistema para considerar **tiempos de espera** y **diferentes costos por trayecto**.
- Implementar una interfaz gráfica para visualizar las rutas.
- Optimizar el rendimiento con estructuras de datos avanzadas.