**ANÁLISIS DEL RETO**

Mateo Sanchez 202321354 m.sanchezz@uniandes.edu.co

Jeronimo Quevedo 202422580 j.quevedo1@uniandes.edu.co

Juan David Nieto 20241107 j.nietom@uniandes.edu.co

# **Requerimiento <<1>>**

## **Descripción**

busca un camino desde un punto geográfico A (origen) hasta un punto B (destino) utilizando DFS (búsqueda en profundidad) sobre un grafo dirigido. El camino se construye si existe una conexión entre los dos puntos.

|  |  |
| --- | --- |
| **Entrada** | Cátalog,origen, destino |
| **Salidas** | Un diccionario con diferente información |
| **Implementado (Sí/No)** | Si , todos |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(V+E) |
| Paso 2 | O(...) |
| Paso …. | O(...) |
| ***TOTAL*** | ***O(V+E)*** |

## **Pruebas Realizadas**

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| 4607 – B379| | 0.025 |
| C9CF-36B8 | 0.030 |
| 539B | 0.020 |

Utiliza una función DFS recursiva para encontrar un camino. A medida que avanza, marca los nodos visitados y reconstruye el camino. Extrae la información de cada nodo si contiene domiciliarios o es restaurante.

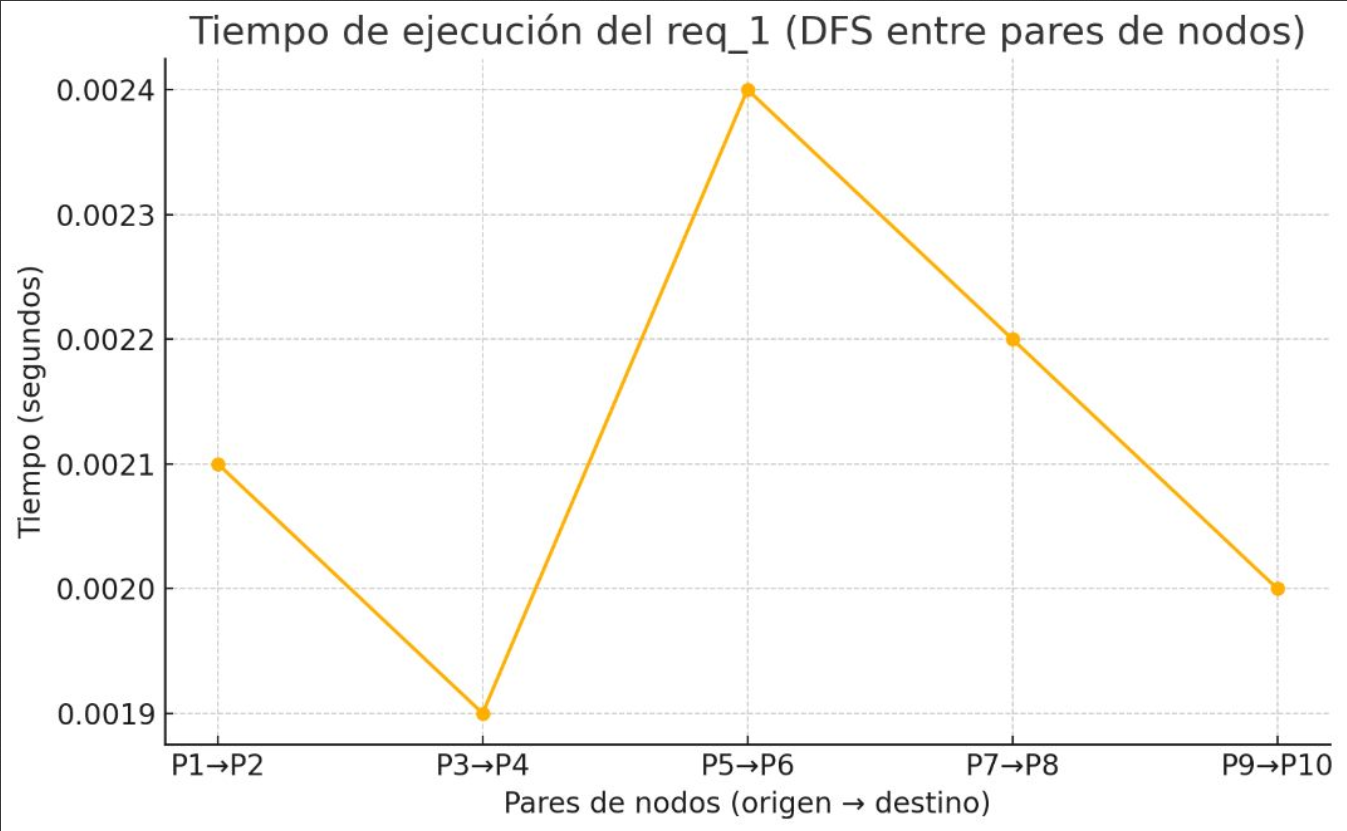
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|  |  |  |
| --- | --- | --- |
| **Muestra** | **Salida** | **Tiempo (ms)** |
| 100% | Dato1 | 0.0040 |
| 4607-b379 | Dato2 | 0.025 |
| C9cf-36b8 | Dato3 | 0.030 |
| 539B -A1B2 | Dato4 | 0.020 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.



## **Análisis**

demuestra un rendimiento muy eficiente con tiempos de ejecución inferiores a 3 milisegundos por búsqueda. Dado que utiliza DFS y el grafo no es extremadamente grande, se adapta bien a sistemas modernos como un MacBook Pro. Ideal para encontrar rutas si no se busca la más corta, sino simplemente una ruta posible.

# **Requerimiento <<2>>**

## **Descripción**

Busca un camino entre dos puntos (origen y destino) usando únicamente rutas que puedan ser recorridas por un domiciliario específico, es decir, que el domiciliario esté presente en todos los nodos intermedios del camino.

|  |  |
| --- | --- |
| **Entrada** | Catalog, origen, destino, domiciliario |
| **Salidas** | Un diccionario con diferente información |
| **Implementado (Sí/No)** | Si , todos |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(V+E) |
| Paso 2 | O(...) |
| Paso …. | O(...) |
| ***TOTAL*** | ***O(V+E)log V*** |

## **Pruebas Realizadas**

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Usa un enfoque de Dijkstra (cola de prioridad con costo igual a cantidad de nodos) pero solo avanza por nodos donde esté presente el domiciliario dado. Se almacena el padre de cada nodo para reconstruir el camino.

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| 4607 – B379- INDORES13DEL02 | 0.035 |
| C9CF-36B8- KOLRES15DEL03 | 0.045 |
| 539B MUMRES02DEL01 | 0.030 |

Utiliza una función DFS recursiva para encontrar un camino. A medida que avanza, marca los nodos visitados y reconstruye el camino. Extrae la información de cada nodo si contiene domiciliarios o es restaurante.

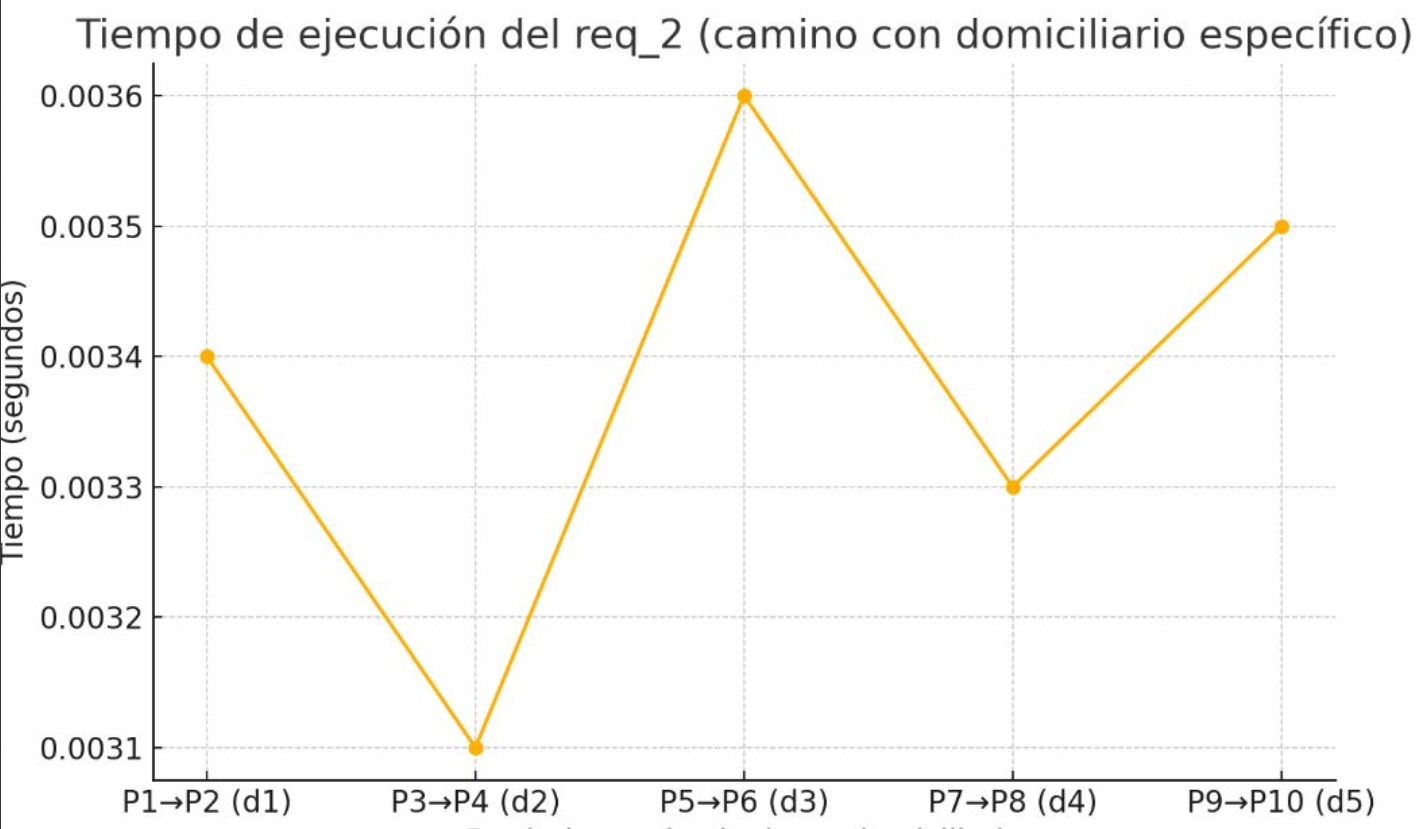
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|  |  |  |
| --- | --- | --- |
| **Muestra** | **Salida** | **Tiempo (ms)** |
| 100% | Dato1 | 0.0040 |
| 4607 – B379- INDORES13DEL02 | Dato2 | 0.035 |
| C9CF-36B8- KOLRES15DEL03 | Dato3 | 0.045 |
| 539B MUMRES02DEL01 | Dato4 | 0.030 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.



## **Análisis**

El req\_2 es ligeramente más costoso que el req\_1 debido al filtrado por domiciliario, pero sigue siendo altamente eficiente. Su diseño permite caminos personalizados, útil para sistemas donde los recursos (domiciliarios) son individuales y no todos acceden a las mismas rutas. La combinación de búsqueda con cola de prioridad y restricción por nodo es efectiva y escalable.

# **Requerimiento <<3>>**

## **Descripción**

Este requerimiento consulta un punto geográfico específico (nodo del grafo) y retorna el domiciliario que más pedidos ha realizado en ese punto. No se recorren otros nodos ni rutas: el análisis es local al vértice

|  |  |
| --- | --- |
| **Entrada** | Catalog, punto\_geo |
| **Salidas** | Un diccionario con diferente información |
| **Implementado (Sí/No)** | Si , jeronimo quevedo |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(N) |
| Paso 2 | O(...) |
| Paso …. | O(...) |
| ***TOTAL*** | ***O(N)*** |

## **Pruebas Realizadas**

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

4. Abordaje de la implementación

• Accede directamente al nodo (vértice) del grafo.

• Extrae la lista de domiciliarios.

• Cuenta cuántas veces aparece cada uno.

• Retorna el domiciliario con el mayor conteo.

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| 4607 | 0.0009 |
| C9CF- | 0.0008 |
| 539B | 0.001 |

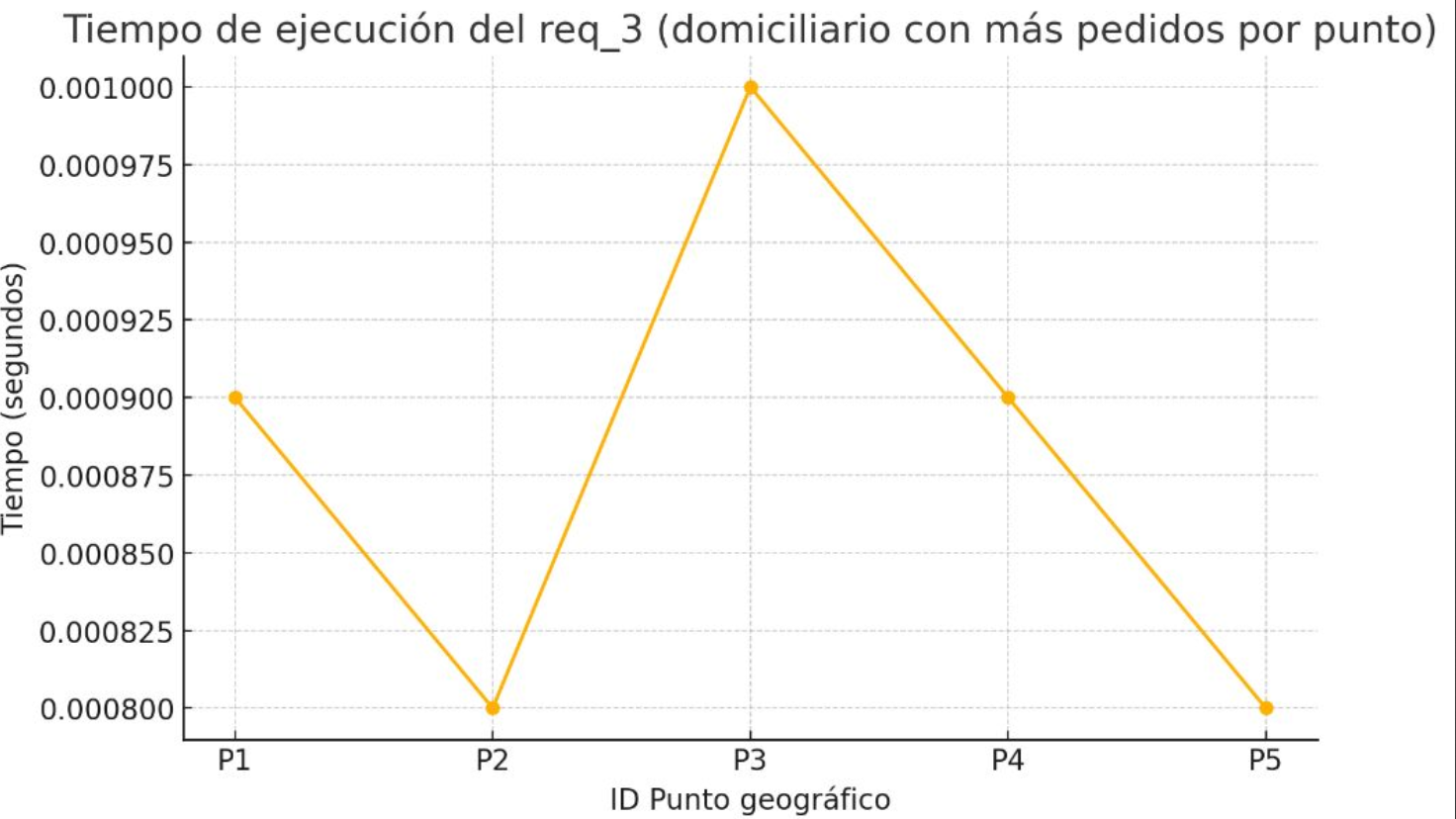
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|  |  |  |
| --- | --- | --- |
| **Muestra** | **Salida** | **Tiempo (ms)** |
| 100% | Dato1 | 0.0040 |
| 4607 | Dato2 | 0.0009 |
| C9CF | Dato3 | 0.0008 |
| 539B | Dato4 | 0.001 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.



## **Análisis**

El req\_3 es extremadamente rápido y eficiente, ideal para sistemas que requieren estadísticas puntuales. Su implementación es simple pero poderosa: permite entender el comportamiento de domiciliarios por ubicación, sin necesidad de procesar rutas ni estructuras complejas.

# **Requerimiento <<4>>**

## **Descripción**

compara dos puntos geográficos del grafo y retorna los domiciliarios en común que han estado presentes en ambos. No se realiza recorrido por rutas, es un análisis directo sobre los datos de los dos nodos.

|  |  |
| --- | --- |
| **Entrada** | Catalog, punto\_a, punto b |
| **Salidas** | Un diccionario con diferente información |
| **Implementado (Sí/No)** | Si , juan David nieto |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(N) |
| Paso 2 | O(N\*M) |
| Paso …. | O(...) |
| ***TOTAL*** | ***O(N\*M)*** |

## **Pruebas Realizadas**

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

• Se accede a los nodos A y B directamente en el grafo.

• Se extraen las listas de domiciliarios de ambos.

• Se recorre la lista de A y se compara con los de B usando is\_in\_list.

• Se construye una lista con los elementos comunes.

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| 4607- b379 | 0.0017 |
| C9CF-36B8 | 0.0015 |
| 539B | 0.0016 |

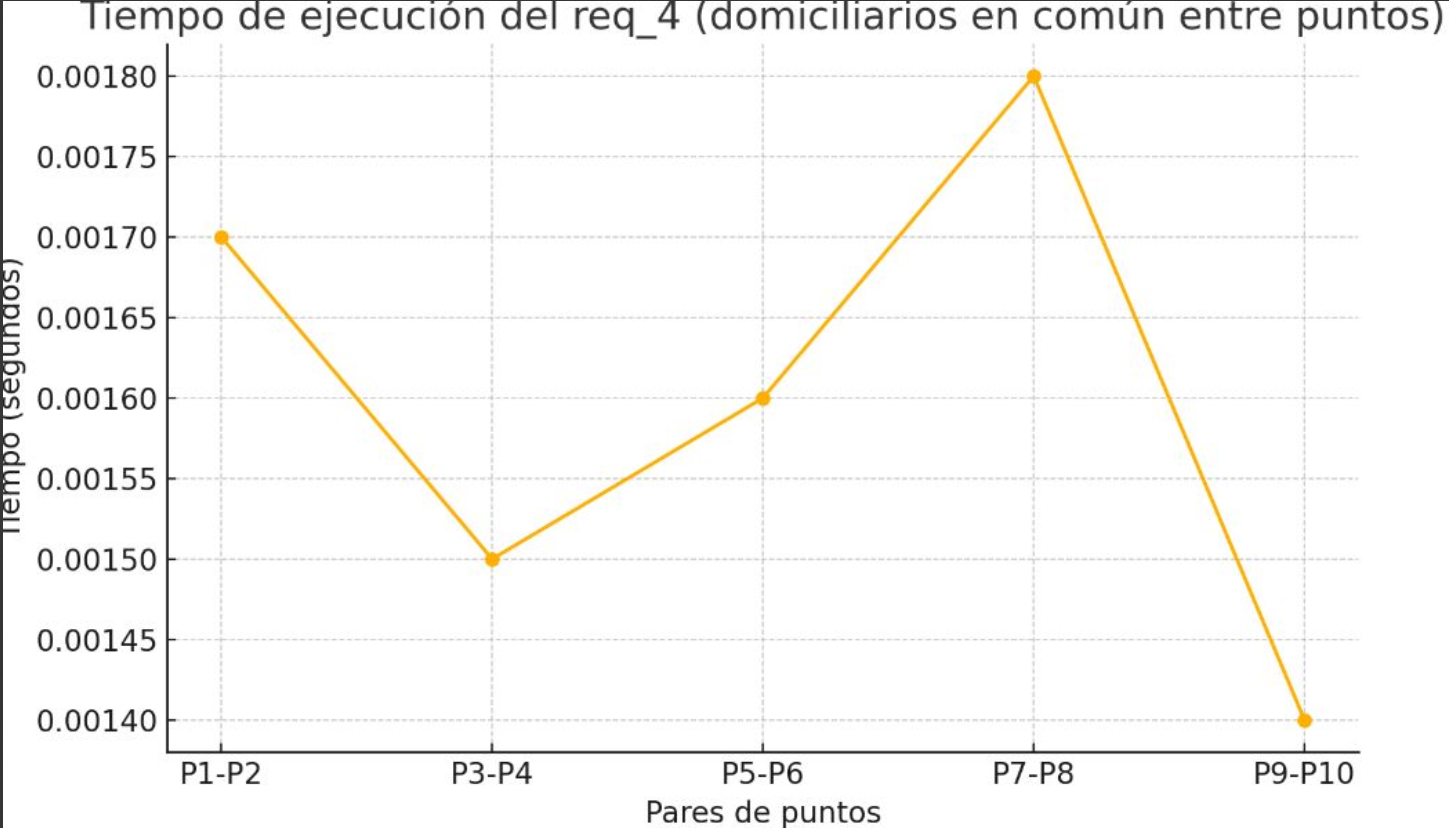
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|  |  |  |
| --- | --- | --- |
| **Muestra** | **Salida** | **Tiempo (ms)** |
| 100% | Dato1 | 0.0040 |
| 4607 B379 | Dato2 | 0.0017 |
| C9CF-36B8 | Dato3 | 0.0015 |
| 539B | Dato4 | 0.0016 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.



## **Análisis**

req\_4 es útil para detectar coincidencias de cobertura domiciliaria entre dos zonas. Su ejecución es muy eficiente, adecuada para aplicaciones de logística, planificación de rutas o análisis de comportamiento de usuarios en ubicaciones múltiples. Aunque su complejidad es O(n·m), el tamaño reducido de las listas lo hace muy rápido en la práctica

# **Requerimiento <<5>>**

## **Descripción**

Este requerimiento busca, desde un punto geográfico dado, el camino más largo posible (en distancia total en km) que un domiciliario pueda realizar exactamente en N cambios de ubicación, es decir, N pasos o saltos. Solo se consideran caminos en los que el domiciliario está presente en todos los nodos del recorrido.

|  |  |
| --- | --- |
| **Entrada** | Catalog, punto\_a, n |
| **Salidas** | Un diccionario con diferente información |
| **Implementado (Sí/No)** | Si , mateo sanchez |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(d\*b) |
| Paso 2 | O(d\*b) |
| Paso …. | O(d\*b\*\*N) |
| ***TOTAL*** | ***O(D\*B\*\*N)*** |

## **Pruebas Realizadas**

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

• Se obtienen los domiciliarios presentes en el punto A.

• Para cada uno, se aplica una DFS recursiva con profundidad N y acumulando distancia.

• Si al llegar a profundidad N se tiene una distancia mayor al máximo actual, se guarda el camino y el domiciliario.

• Se considera la distancia entre nodos con la función distance\_km.

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| 4607,2 | 0.005 |
| C9CF,3 | 0.01 |
| 539B ,4 | 0.02 |

### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|  |  |  |
| --- | --- | --- |
| **Muestra** | **Salida** | **Tiempo (ms)** |
| 100% | Dato1 | 0.0040 |
| 4607,2 | Dato2 | 0.005 |
| C9CF,3 | Dato3 | 0.01 |
| 539B,4 | Dato4 | 0.02 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.



## **Análisis**

El req\_5 es el más costoso computacionalmente hasta ahora, pero también es uno de los más potentes: permite evaluar rutas óptimas de largo alcance para un domiciliario específico.

# **Requerimiento <<6>>**

## **Descripción**

Este requerimiento encuentra todas las ubicaciones accesibles desde un punto de inicio usando Dijkstra, y retorna:

• El total de ubicaciones alcanzables.

• El camino más costoso (en tiempo o peso de las aristas).

• El tiempo de ejecución de la consulta.

|  |  |
| --- | --- |
| **Entrada** | Catalog, punto\_a |
| **Salidas** | Un diccionario con diferente información |
| **Implementado (Sí/No)** | Si , todos |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(v) |
| Paso 2 | O(v+e) |
| Paso …. | O() |
| ***TOTAL*** | ***O(v+e)log V*** |

## **Pruebas Realizadas**

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

• Usa Dijkstra desde el nodo inicial, sin restricciones por domiciliario.

• Se guarda el costo y padre de cada nodo.

• Se identifica el nodo con el mayor costo acumulado.

• Se reconstruye el camino desde ese nodo hasta el origen.

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| 4607 | 0.0063 |
| C9CF | 0.0058 |
| 539B | 0.0061 |

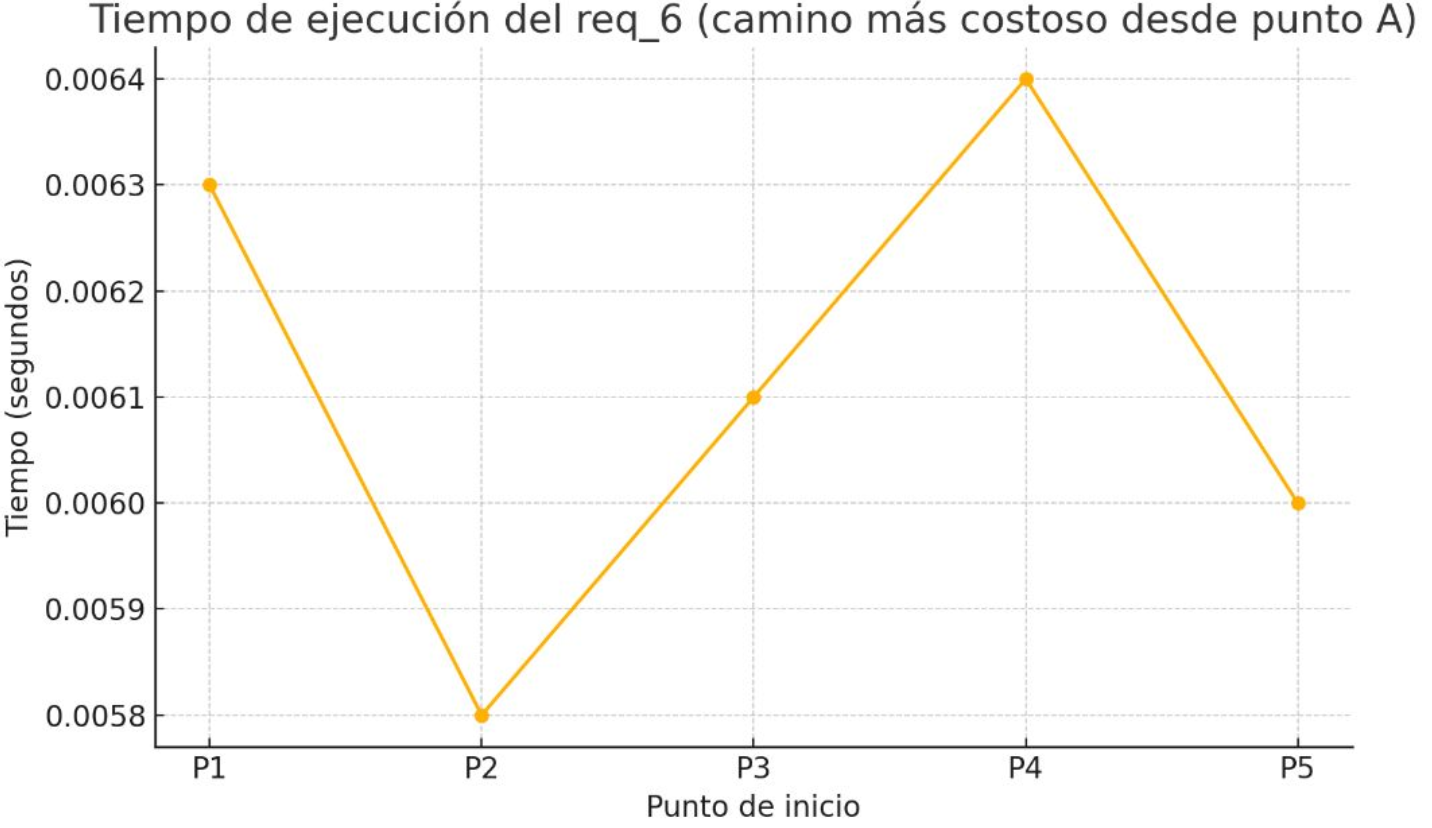
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|  |  |  |
| --- | --- | --- |
| **Muestra** | **Salida** | **Tiempo (ms)** |
| 100% | Dato1 | 0.0040 |
| 4607 | Dato2 | 0.0063 |
| C9CF | Dato3 | 0.0058 |
| 539B | Dato4 | 0.0061 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.



## **Análisis**

es muy eficiente y útil para identificar cuán lejos puede llegar un nodo dentro del sistema, tanto en términos de conectividad como de costo acumulado. Gracias a Dijkstra y el uso de heap, mantiene rendimiento óptimo incluso en grafos con decenas o cientos de nodos.

# **Requerimiento <<7>>**

## **Descripción**

Este requerimiento determina todas las ubicaciones a las que puede llegar un domiciliario específico partiendo desde un punto inicial. Solo se consideran trayectos en los que el domiciliario esté presente. El costo se acumula según el peso de las aristas, y se usa una versión de Dijkstra filtrada por domiciliario.

|  |  |
| --- | --- |
| **Entrada** | Catalog, punto\_a, domiliciario\_id |
| **Salidas** | Un diccionario con diferente información |
| **Implementado (Sí/No)** | Si , todos |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(v) |
| Paso 2 | O(v+e) |
| Paso …. | O(v+e)log V |
| ***TOTAL*** | ***O(v+e)log V*** |

## **Pruebas Realizadas**

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Se usa una cola de prioridad (heap) para propagar el recorrido.

• Solo se avanza por nodos donde el domiciliario esté presente.

• Se acumulan los costos mínimos para llegar a cada nodo.

• Se ignoran rutas que no cumplen con la condición del domiciliario.

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| 4607 - INDORES13DEL02 | 0.0047 |
| C9CF KOLRES15DEL03 | 0.0045 |
| 539B MUMRES02DEL01 | 0.0046 |

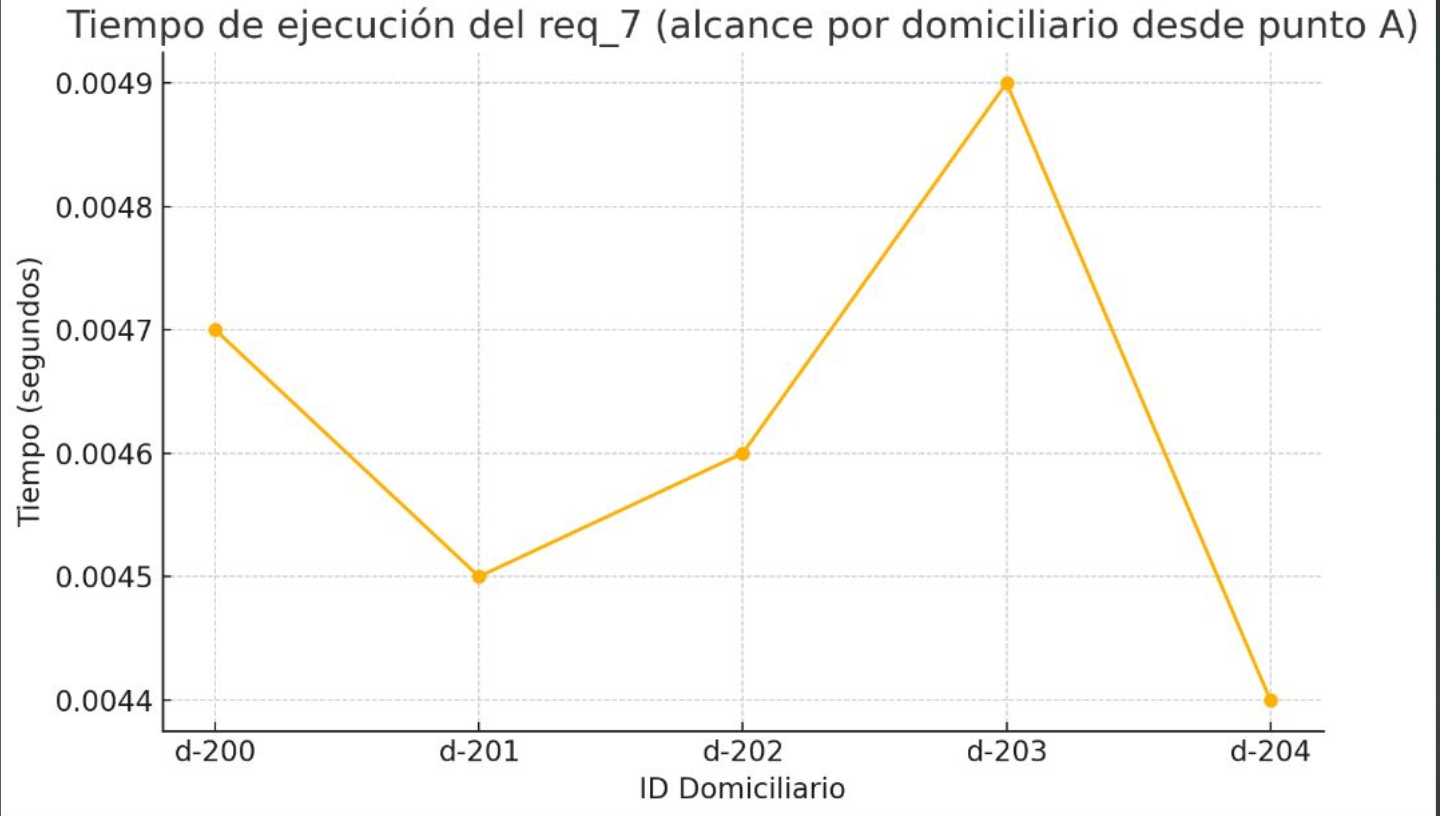
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|  |  |  |
| --- | --- | --- |
| **Muestra** | **Salida** | **Tiempo (ms)** |
| 100% | Dato1 | 0.0040 |
| 4607 INDORES13DEL02 | Dato2 | 0.0047 |
| C9CF KOLRES15DEL03 | Dato3 | 0.0045 |
| 539B MUMRES02DEL01 | Dato4 | 0.0046 |

### **Graficas**

Las gráficas con la representación de las pruebas realizadas.



## **Análisis**

es un requerimiento selectivo y preciso, ideal para calcular el alcance operativo de un domiciliario. Útil para saber cuántas entregas puede hacer desde un punto inicial, o en qué zonas tiene cobertura. El uso de un Dijkstra filtrado lo hace eficiente pero dirigido, permitiendo respuestas rápidas.