**ACTIVIDAD LABORATORIO NO.1**

**ACTIVIDAD: PROPUESTA DE DESARROLLO Y DESPLIEGUE DE UN PROYECTO DE INTELIGENCIA ARTIFICIAL**

**PRESENTADO POR:**

**VIVIANA ANDREA BAUTISTA PULIDO**

**CARMEN EDILIA RICARDO HELVES**

**ALEJANDRO DE MENDOZA**

**PRESENTADO AL PROFESOR:**

**ING DIEGO OSORIO REINA**

**FUNDACIÓN UNIVERSITARIA INTERNACIONAL DE LA RIOJA**

**BOGOTÁ D.C.**

**24 DE NOVIEMBRE**

**2024**

**TABLA DE CONTENIDO**

[INTRODUCCIÓN 4](#_Toc184455858)

[TÍTULO DEL PROYECTO 4](#_Toc184455859)

[DESARROLLO DEL PROYECTO 4](#_Toc184455860)

[Este desarrollo está orientado a resolver problemas como: 4](#_Toc184455861)

[ Determinar las rutas más cortas entre un almacén y varios puntos de entrega. 4](#_Toc184455862)

[ Comparar el rendimiento de los algoritmos en términos de eficiencia al minimizar distancias. 4](#_Toc184455863)

[ Al limitar el modelo al criterio de distancia, el enfoque es más directo y permite un análisis claro de cómo los algoritmos manejan problemas de optimización. Esto también abre la puerta a futuros ajustes, como la incorporación de factores adicionales (tempos, capacidades, etc.), en sistemas más complejos. 4](#_Toc184455864)

[Requisitos Del Sistema 4](#_Toc184455865)

[Diseño De La Solución 5](#_Toc184455866)

[Recursos necesarios: 5](#_Toc184455867)

[Metodología De Desarrollo 6](#_Toc184455868)

[Metodología: 6](#_Toc184455869)

[Justificación: 6](#_Toc184455870)

[Roles: 6](#_Toc184455871)

[Perfiles De Recursos Humanos 7](#_Toc184455872)

[ Ingeniero de datos: 7](#_Toc184455873)

[ Científico de datos-Ingeniero Informático: 7](#_Toc184455874)

[ Arquitecto de sistemas – Ingeniero Informático: 7](#_Toc184455875)

[ Desarrollador Full Stack – Ingeniero Informático: 7](#_Toc184455876)

[ Especialista en logística: 7](#_Toc184455877)

[ Analista de seguridad – Ingeniero informático especialista en Ciberseguridad: 8](#_Toc184455878)

[Despliegue Del Sistema 8](#_Toc184455879)

[ Hardware: 8](#_Toc184455880)

[ Software: 8](#_Toc184455881)

[ Posibles problemas: 8](#_Toc184455882)

[Securización Y Anonimización De Datos 9](#_Toc184455883)

[ Anonimización: 9](#_Toc184455884)

[ Cifrado: 9](#_Toc184455885)

[ Regulaciones: 9](#_Toc184455886)

[Área De Investigación 10](#_Toc184455887)

[ Computación bioinspirada: 10](#_Toc184455888)

[Desarrollo De La Metodología 10](#_Toc184455889)

[Algoritmo Voraz 10](#_Toc184455890)

[Como introducción al algoritmo Voraz nos permitimos indicar que este es un algoritmo que tiene enfoque en la resolución de problemas donde construye una solución paso a paso, eligiendo una opción óptima que conlleva a una respuesta en general óptima. Su énfasis se basa en si la rapidez en encontrar una solución es primordial y no es necesaria de manera precisa la solución óptima. La distancia euclidiana suele representarse como una función heurística, denotada comúnmente por H(n), que es la estimación del costo mínimo para ir desde el nodo actual n hasta el objetivo. Es importante precisar que la heurística de la distancia euclidiana es una buena opción como H(n) cuando se busca una ruta entre puntos geográficos, ya que representa la línea recta (o el camino directo) y da una medida de la cercanía entre nodos. 10](#_Toc184455891)

[ALGORITMO A\* 15](#_Toc184455892)

[CONCLUSIÓN DE LA ACTIVIDAD 18](#_Toc184455893)

[BIBLIOGRAFÍA 19](#_Toc184455894)

# INTRODUCCIÓN

En el ámbito de la logística, encontrar rutas óptimas para el transporte de mercancías es crucial para reducir costos y mejorar la eficiencia operativa. Este proyecto tiene como objetivo diseñar un sistema de optimización de rutas basado en **algoritmos de búsqueda informada**, específicamente en el algoritmo Voraz y el algoritmo A\*, los cuales se aplicarán para minimizar las distancias totales recorridas en redes de distribución.

A diferencia de otros modelos que consideran múltiples factores como tiempos de entrega, tipos de vehículos o capacidades, este desarrollo se enfoca únicamente en optimizar el criterio de distancia entre los puntos de origen y destino. Esto simplifica el análisis y permite evaluar la eficacia de los algoritmos seleccionados para resolver problemas de búsqueda en grafos.

# TÍTULO DEL PROYECTO

Para el desarrollo de esta actividad elegimos como título del proyecto “Optimización de rutas logísticas mediante los Algoritmos Voraz y A\*.

# DESARROLLO DEL PROYECTO

## Este desarrollo está orientado a resolver problemas como:

## Determinar las rutas más cortas entre un almacén y varios puntos de entrega.

## Comparar el rendimiento de los algoritmos en términos de eficiencia al minimizar distancias.

## Al limitar el modelo al criterio de distancia, el enfoque es más directo y permite un análisis claro de cómo los algoritmos manejan problemas de optimización. Esto también abre la puerta a futuros ajustes, como la incorporación de factores adicionales (tiempos, capacidades, etc.), en sistemas más complejos.

## Requisitos Del Sistema

• Problema: Optimizar las rutas de transporte para minimizar costos operativos y tiempos de entrega en una red de distribución de mercancías a través del cálculo de distancias óptimas basadas en kilómetros para el fácil transporte de mercancías utilizando distancias euclídeas y distancias tradicionales en kilómetros por carretera.

• Objetivo: Diseñar un sistema inteligente que calcule las rutas óptimas terrestres basadas en restricciones de kilómetros que impliquen reducciones en tiempos al recorrer menores distancias entre las diferentes ciudades del país Colombia.

• Beneficiarios: Empresas de logística y cadena de suministro que buscan eficiencia en sus operaciones en Colombia.

## Diseño De La Solución

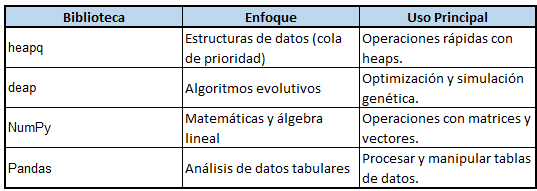
### Recursos necesarios:

• Hardware:

* Servidor con GPU para entrenar modelos y ejecutar algoritmos a gran escala.
* Dispositivos para visualización (laptops, tablets y celular, ya que como se tiene claridad Python es multiplataforma y sirve para la visualización en todos los tipos de plataforma Apple, Windows, Android, entre otros).

• Software:

* Lenguajes de programación: Python (bibliotecas como HEAPQ, DEAP, NumPy, Pandas). A continuación, una tabla con los enfoques y uso principal en el desarrollo de este laboratorio:



* Herramientas de visualización: Python, Anaconda-Spyder.
* Entorno para integración y pruebas: NetBeans y Anaconda.

• Técnicas empleadas:

* Algoritmos Voraz y Algoritmo a\* para optimización.
* Modelos heurísticos para validación inicial de resultados.
* Distancias euclídeas.
* Análisis geoespacial con librerías como Geopy o Folium.
* API de mapas como Google Maps para datos de kilómetros y distancias entre ciudades.

## Metodología De Desarrollo

### Metodología:

Para la metodología de desarrollo de este laboratorio nos vamos a enfocar en la metodología Scrum que se basa en un marco de trabajo ágil utilizado principalmente en el desarrollo de software, aunque también se aplica a otros tipos de proyectos. Está diseñado para gestionar proyectos complejos y cambiantes como es el caso de rutas cambiantes a diferentes ciudades en Colombia a través de una organización logística que necesita entregar o distribuir mercancías a lo largo de las diferentes ciudades que componen el país, fomentando la colaboración, la adaptabilidad y la entrega incremental de valor.

Ahora es preciso indicar que la metodología Scrum fomenta la autoorganización y la colaboración entre los miembros del equipo, es mas es uno de sus principios básicos y que mejor manera de gestionar una autoorganización en una empresa que dando a conocer las rutas optimas que deben seguir los procesos de distribución. Adicionalmente y otro de los principios de esta metodología es la retroalimentación constante que se basa en que después de cada sprint, se evalúan los resultados y se ajusta el plan según las necesidades del proyecto que en este caso es la distribución de mercancías por lo que al evaluar y poner en marcha si una ruta no es óptima se deben proceder a corregir los errores en el código, sobre todo en el bucle que recorre. Y por último frente a la adaptabilidad es importante indicar que Scrum acepta cambios en los requisitos, incluso en etapas avanzadas del desarrollo y pues en el ámbito de la logístico esto es algo necesario sobre todo si se quiere incluir tipos de vehículos, velocidades máximas y mínimas de reparto y tipos de mercancías a despachar por volumen, pesos volumétricos y pesos brutos.

### Justificación:

Como justificación es preciso indicar que la metodología Scrum fomenta la autoorganización y la colaboración entre los miembros del equipo, es mas es uno de sus principios básicos y que mejor manera de gestionar una autoorganización en una empresa que dando a conocer las rutas optimas que deben seguir los procesos de distribución. Adicionalmente y otro de los principios de esta metodología es la retroalimentación constante que se basa en que después de cada sprint, se evalúan los resultados y se ajusta el plan según las necesidades del proyecto que en este caso es la distribución de mercancías en Colombia por lo que al evaluar y poner en marcha si una ruta no es óptima se deben proceder a corregir los errores en el código, sobre todo en el bucle que recorre. Y por último frente a la adaptabilidad es importante indicar que Scrum acepta cambios en los requisitos, incluso en etapas avanzadas del desarrollo y pues en el ámbito de la logístico esto es algo necesario sobre todo si se quiere incluir tipos de vehículos, velocidades máximas, velocidades mínimas de reparto, y tipos de mercancías a despachar por volumen, pesos volumétricos y pesos brutos.

### Roles:

A continuación, vamos a denotar los roles adecuados para la ejecución de este proyecto en una organización logística:

#### Product Owner:

Esta persona es la encargada de definir los requisitos y asegura la alineación con los objetivos del cliente.

#### Scrum Master:

Supervisa el cumplimiento de los principios ágiles, es una persona experta en Scrum.

#### Equipo de desarrollo:

Este es el equipo que implementa los algoritmos, desarrolla la interfaz y va a realizar las pruebas de campo e implementación.

## Perfiles De Recursos Humanos

### Ingeniero de datos:

Esta es una persona encargada de todo lo concerniente a la recopilación, limpieza y preparación de los datos.

### Científico de datos-Ingeniero Informático:

Esta persona es la encargada de desarrollar y ajustar los algoritmos de búsqueda como en este caso el Algoritmo A\* y el algoritmo Voraz, es claro que el proyecto se puede adaptar a algoritmos genéticos, pero pues no se van a implementar en este proyecto por niveles de extensión en el mismo, aunque de nuestra parte si nos gustaría de pronto más adelante dar esa milla extra.

### Arquitecto de sistemas – Ingeniero Informático:

Es la persona que va a diseñar la infraestructura de HW y SW, que incluso al manejar tablas se puede generar de igual manera un acople de implementación en MySQL con Python utilizando bibliotecas específicas que permiten conectar Python con bases de datos MySQL. La más popular y recomendada es mysql-connector-python, aunque también se pueden usar otras como PyMySQL o SQLAlchemy si se necesitan más funcionalidades.

### Desarrollador Full Stack – Ingeniero Informático:

Es la persona encargada de toda la construcción de la interfaz de usuario y la integración con el sistema backend donde como es conocido el backend es el motor que alimenta la aplicación, mientras que el frontend es la interfaz visual que los usuarios ven y con la que interactúan.

### Especialista en logística:

Es la persona encargada de definir las restricciones y escenarios reales del problema, es una persona que tiene claridad de todos los proesos logísticos, manejo de WMS (Warehouse Managment System), manejo de MRP (Material Request Planning), manejo de MLP (Managment Logistis Processes), Lead Times, MPS (Managment Production System), entre otros. Y que da un enfoque claro de todas las posibles posibilidades a implementar en el desarrollo de este proyecto logístico.

### Analista de seguridad – Ingeniero informático especialista en Ciberseguridad:

Es la persona encargada de asegurar la protección de los datos sensibles, que o haya filtraciones y que se tengan los parámetros de permisos establecidos de información a los diferentes usuarios que operen la herramienta.

## Despliegue Del Sistema

Frente a la infraestructura de hardware y software vamos a utilizar la siguiente:

### Hardware:

Lo primero es tener un servidor en la nube (AWS o Azure) para escalabilidad y procesamiento de datos, que permita la operabilidad en tiempo real, adicionalmente se espera tener dispositivos IoT en vehículos para recolectar datos en tiempo real frente a sus distancias recorridas y tiempos de entregas.

### Software:

El objetivo es utilizar contenedores Docker para despliegue modular los cuales son una tecnología que permite empaquetar y desplegar aplicaciones y sus dependencias en un entorno portátil y aislado. Esto asegura que la aplicación funcione de manera consistente en cualquier entorno, ya sea en desarrollo, pruebas o producción. Es importante precisar que estos contenedores Docker se utilizan ampliamente para lograr un despliegue modular, donde cada contenedor puede representar un módulo o servicio independiente, como una base de datos en MySQL, una API o un frontend.

CI/CD con Jenkins para actualizaciones, y para dar claridad es una práctica automatizada para integrar y desplegar cambios de software de manera continua y eficiente. Jenkins es una herramienta de integración continua que permite orquestar y automatizar las tareas necesarias para construir, probar y desplegar aplicaciones en diferentes entornos.

### Posibles problemas:

En este caso como posibles desafíos tenemos el retardo en la transmisión de datos en zonas con mala cobertura, sobre todo en zonas donde la conexión web sea compleja por las zonas geográficas que Colombia presenta donde los niveles de cobertura sean escasos. Ejemplo: Si nos vamos a un campo Petrolero como Rubiales, la conexión genera fallos y en vista que hay torres de telecomunicaciones, estas no aseguran la cobertura total de la zona.

Y otro desafío fundamental que consideramos es el hecho de la escalabilidad del sistema si aumenta la cantidad de puntos de entrega, ya que con cada nuevo punto de entrega, la cantidad de rutas posibles crece exponencialmente (esto lo conocemos como el famoso problema del viajante) donde determinar la ruta óptima para cumplir con múltiples entregas se vuelve computacionalmente más costoso y a medida que aumenta el número de puntos, también crece la necesidad de coordinar horarios, prioridades y restricciones de entrega.

## Securización Y Anonimización De Datos

### Anonimización:

Enmascarar información personal de clientes. Para dar claridad es el proceso mediante el cual se transforman datos personales o sensibles para que no puedan ser asociados directa o indirectamente con una persona específica, garantizando la privacidad de los individuos. Este proceso es fundamental para cumplir con normativas de protección de datos.

### Cifrado:

En este caso vamos a generar la implementación del cifrado AES-256 para proteger datos en tránsito y almacenamiento. Este método es uno de los métodos de cifrado más seguros y ampliamente utilizados para proteger datos. Es un algoritmo simétrico, lo que significa que utiliza la misma clave para cifrar y descifrar los datos, claro hay más opciones de cifrado que permiten una mayor seguridad, pero no se considera indispensable para este proyecto logístico, otro tema sería la ejecución de seguridad en un Banco en el cual podríamos utilizar un cifrado hibrido como (AES-256) con cifrado asimétrico, como RSA o Elliptic Curve Cryptography (ECC).

### Regulaciones:

En este caso las regulaciones se basan en el cumplimiento de estándares como los establecidos en el GDPR y CCPA que son regulaciones que establecen normas sobre la recopilación, procesamiento, almacenamiento y protección de datos personales. Aunque ambas están diseñadas para proteger la privacidad de los individuos (clientes para entrega de mercancías en Colombia).

## Área De Investigación

### Computación bioinspirada:

Uso de algoritmos para resolver problemas de optimización en entornos dinámicos que en este caso por simplicidad solo vamos a implementar dos algoritmos de búsqueda los cuales son el Algoritmo Voraz y el algoritmo A\*, se recalca que el objetivo es utilizar algoritmos más avanzados como son los genéticos y en este caso se espera utilizar el Algoritmo Genético para el Problema del Viajero (TSP, Traveling Salesman Problem). El cual es un algoritmo se adapta perfectamente para optimizar rutas de distribución al buscar la secuencia más eficiente para visitar múltiples ubicaciones.

## Desarrollo De La Metodología

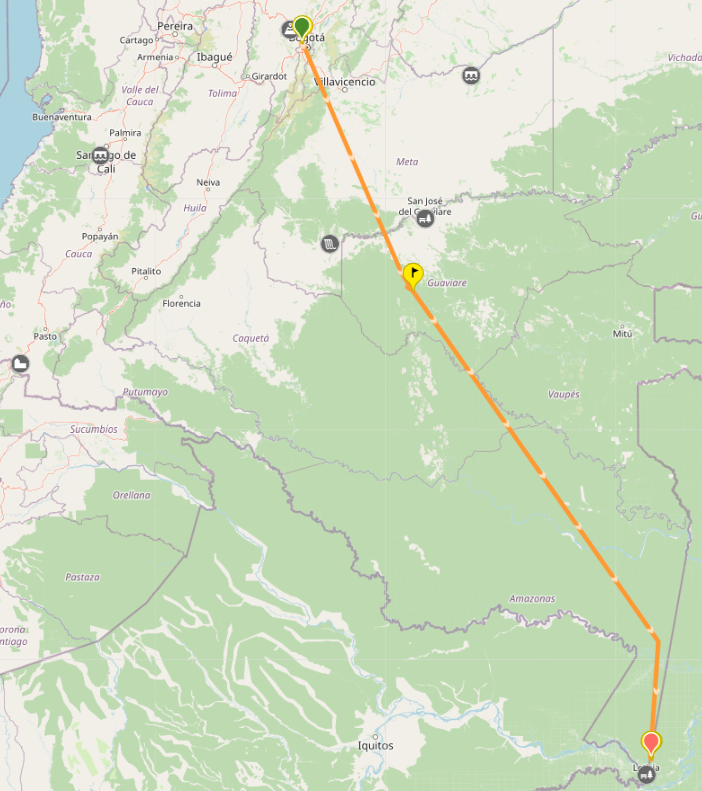
De acuerdo a lo planteado para el desarrollo de esta metodología es preciso primero que todo indicar que un problema de búsqueda consiste en ejecutar una modelación de tal manera que la solución corresponda con un camino que nos lleve desde el estado inicial hasta un estado meta, y en este caso se basa en un problema de búsqueda con el que existe una correspondencia muy cercana entre sus características reales y su modelización. Ahora con base en la actividad expuesta del laboratorio de inteligencia Artificial a continuación vamos a denotar un problema de búsqueda en el rubro del problema del camino más corto mediante búsqueda informada a través de dos algoritmos, el algoritmo Voraz y el algoritmo A\*.

### Algoritmo Voraz

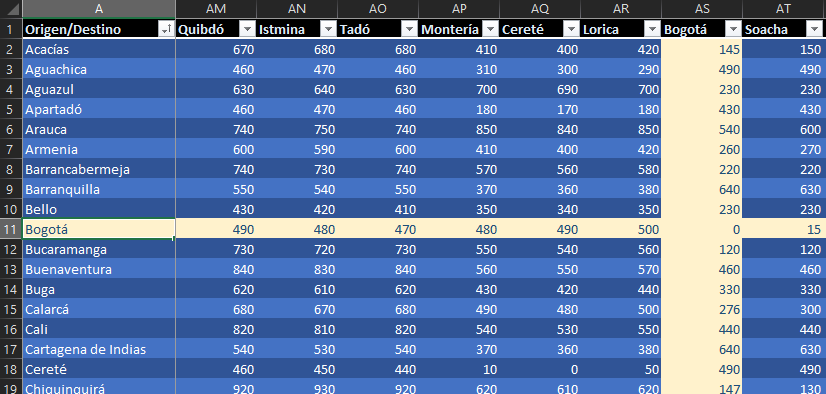
### Como introducción al algoritmo Voraz nos permitimos indicar que este es un algoritmo que tiene enfoque en la resolución de problemas donde construye una solución paso a paso, eligiendo una opción óptima que conlleva a una respuesta en general óptima. Su énfasis se basa en si la rapidez en encontrar una solución es primordial y no es necesaria de manera precisa la solución óptima. La distancia euclidiana suele representarse como una función heurística, denotada comúnmente por H(n), que es la estimación del costo mínimo para ir desde el nodo actual n hasta el objetivo. Es importante precisar que la heurística de la distancia euclidiana es una buena opción como H(n) cuando se busca una ruta entre puntos geográficos, ya que representa la línea recta (o el camino directo, en este caso el camino entre ciudades directo en Kilómetros) y da una medida de la cercanía entre nodos.

#### Desarrollo Algoritmo Voraz

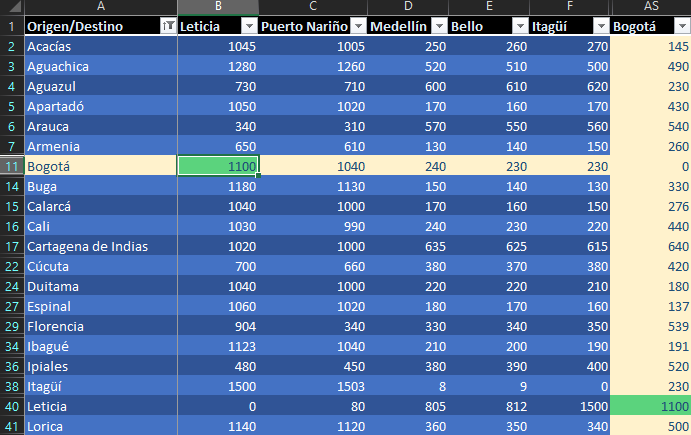
1. Primero que todo vamos al mapa brindado y determinamos la ruta a establecer (**las imágenes se pueden ampliar**).



2. Ahora, desarrollamos los Valores De Referencia Y Distancias Por Carretera, para esto entonces la distancia que tomamos de referencia va a ser la distancia que va a tomar la ciudad inicio-origen que es la ciudad de Bogotá. Entonces y para esto nos remitimos a la tabla de distancias y seleccionamos la columna de referencia de Bogotá.

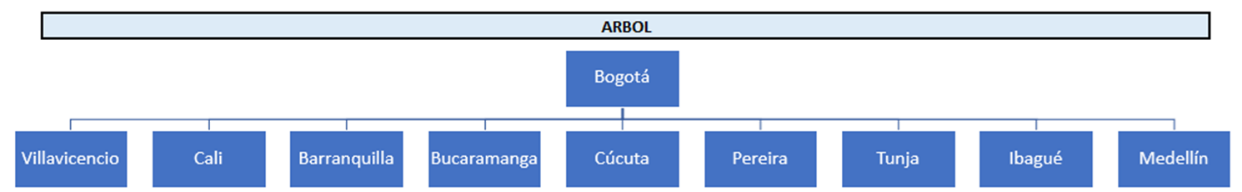


4. Ahora de Bogotá que es el origen a Leticia son **1100** kilómetros como lo podemos ver a continuación:



5. Procedemos a desarrollar el Algoritmo Voraz (lista abierta de estado), y como toma una sola función de referencia que en este caso se basa en que vamos a ir de Bogotá a Leticia, por lo que vamos a tener como referencia la distancia Euclídea (*distancia directa en Kilómetros para las ciudades*) como H(n), entonces para ir de Bogotá a Leticia primero verificamos las conexiones de Bogotá y son las siguientes (se muestra árbol):

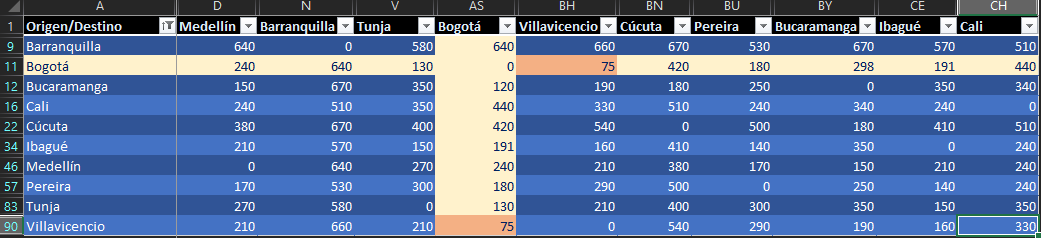
* Bogotá, Villavicencio.
* Bogotá, Cali.
* Bogotá, Barranquilla.
* Bogotá, Bucaramanga.
* Bogotá, Cúcuta.
* Bogotá, Pereira.
* Bogotá, Tunja.
* Bogotá, Ibagué.
* Bogotá, Medellín.



6. Determinamos las distancias de las ciudades conexión a Bogotá y nos da la siguiente tabla en Kilómetros:

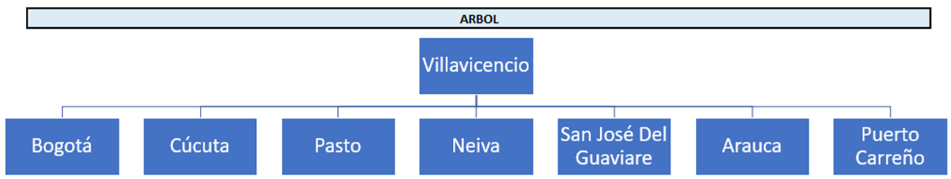


7. A partir de esta distancia tomamos la menor que en este caso es Villavicencio con 75 kilómetros:



8. Verificamos las conexiones de Villavicencio y son las siguientes (se muestra árbol):

* Villavicencio, Bogotá.
* Villavicencio, Cúcuta.
* Villavicencio, Pasto.
* Villavicencio, Neiva.
* Villavicencio, San José Del Guaviare.
* Villavicencio, Arauca.
* Villavicencio, Puerto Carreño.



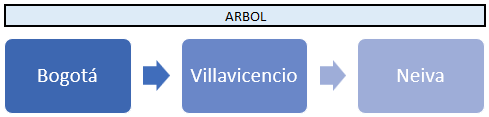
9. Determinamos las distancias de las ciudades conexión a Villavicencio y nos da la siguiente tabla en Kilómetros:



1. Y partir de esta distancia tomamos la menor que en este caso es Neiva con 170 kilómetros:

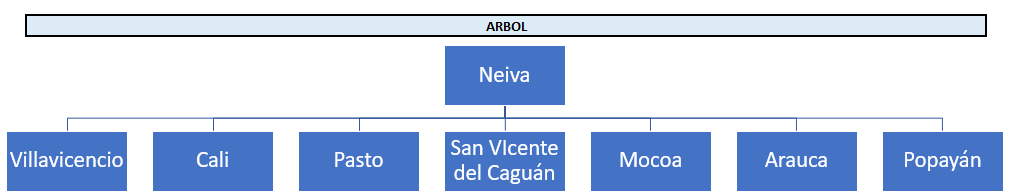


1. En este momento la ruta trazada como óptima es: Bogotá – Villavicencio, y Villavicencio – Neiva, cuya ruta la podemos ver en la imagen a continuación denotada con flechas verdes (se muestra árbol, se gira la imagen por espacio en el trabajo):

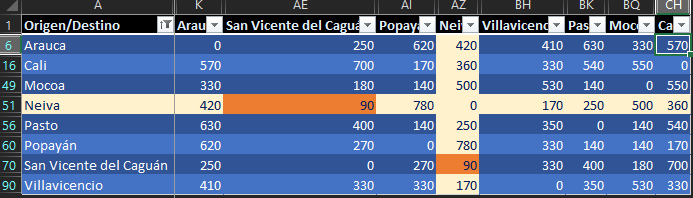


1. Ahora hacemos el mismo proceso por lo que verificamos las conexiones de Neiva y son las siguientes demarcadas en flechas rojas:

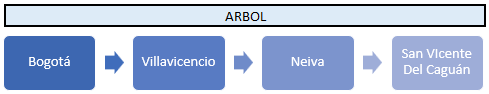
* Neiva, Villavicencio.
* Neiva, Cali.
* Neiva, Pasto.
* Neiva, San Vicente Del Caguán.
* Neiva, Mocoa.
* Neiva, Arauca.
* Neiva, Popayán.



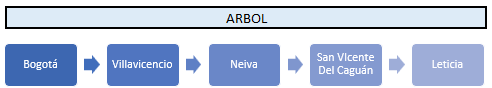
1. Determinamos las distancias de nuestras ciudades conexión a Neiva y nos da la siguiente tabla en Kilómetros:



1. A partir de esta distancia tomamos la menor que en este caso es San Vicente Del Caguán con 90 kilómetros y en este momento la ruta a seguir entonces es Bogotá – Villavicencio, y Villavicencio – Neiva, Neiva – San Vicente Del Caguán (se muestra árbol, se gira la imagen por espacio en el trabajo):

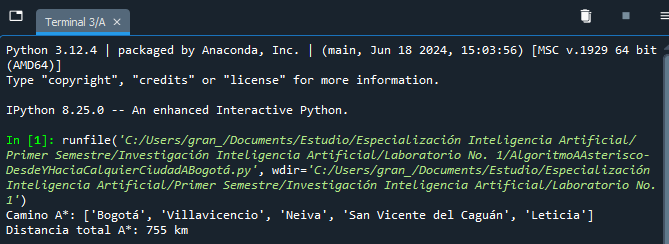


1. Ahora como San Vicente Del Caguán tiene ruta directa a Leticia entonces tomamos esa ruta y descartamos las demás para determinar la ruta óptima que en este caso es: Bogotá – Villavicencio, y Villavicencio – Neiva, Neiva – San Vicente Del Caguán – Leticia.
2. Ahora determinamos el kilometraje total de la ruta óptima determinada como paso final (lista cerrada de estado) terminando el algoritmo Voraz dando un total de 775 kilómetros por carretera, a continuación, el árbol.



* De Bogotá a Villavicencio son: 75 kms.
* De Villavicencio a Neiva son: 170 kms.
* De Neiva a San Vicente Del Caguán son: 90 kms.
* De San Vicente Del Caguán a Leticia son: 420 kms.
* Total, kilómetros recorridos: 75+170+90+420 = 755 kms.

1. A continuación, el resultado del desarrollo en Python (para ver el desarrollo completo del código se adjunta archivo de Python en la entrega, donde se puede seleccionar las ciudades orígenes y destinos que se consideren):



#### Análisis Algoritmo Voraz

Para el análisis del resultado primero selecciono la distancia euclídea, tomo la distancia en carretera, determino cual es la ruta óptima para el algoritmo voraz que toma como información referente la distancia euclídea, determinamos la ruta óptima y en este caso el resultado de la ruta óptima es la ruta: Bogotá – Villavicencio, y Villavicencio – Neiva, Neiva – San Vicente Del Caguán – Leticia, que nos da un total de 775 kilómetros de carretera.

#### Programa Algoritmo Voraz

El desarrollo se ejecutó en la plataforma Python – Anaconda – Spyder, que se adjunta en documentos aparte de acuerdo a lo indicado, se denota que se desarrolló el algoritmo en adición para las ciudades de Medellín y Cali, pero no se adjuntan en este trabajo en PDF o de lo contrario la actividad sobrepasa la extensión máxima del trabajo a desarrollar, pero si se adjuntan en archivos por separado.

### ALGORITMO A\*

Como introducción el Algoritmo A\* es un algoritmo de búsqueda que tiene en cuenta el estado inicial, el estado final y las conexiones que tenemos. Es un algoritmo preferente en el que se utiliza f como función heurística y una función h aceptable. La función de evaluación se basa en un costo uniforme que toma dos valores f(n) = el algoritmo heurístico g(n) + el costo directo-euclídea h(n). Ahora los dos valores que vamos a tomar son: H(n) del punto inicial al final, es la euclídea, y, G(n) va a ser la distancia en carretera, Sumamos los dos valores y a partir de esta suma hallamos la menor distancia.

#### Desarrollo del Algoritmo A\*

A continuación, los pasos para el desarrollo del algoritmo en esta actividad:

1. (Lista abierta de estado), establecemos tabla de origen y destino desde Bogotá a sus respectivas conexiones (se muestra árbol):

2. Determinamos los kilómetros por carretera y la fórmula de f(n) = h(n) + g(n), la tabla queda de la siguiente manera:

3. A partir de esta distancia Euclídea tomamos la menor que en este caso es Soria con 411 kilómetros (se muestra árbol, se gira la imagen por espacio en el trabajo).

6. Establecemos tabla de origen y destino desde Soria a sus respectivas conexiones (se muestra árbol):

8. Determinamos los kilómetros por carretera y la fórmula de f(n) = h(n) + g(n), la tabla queda de la siguiente manera:

9. A partir de esta distancia Euclídea tomamos la menor que en este caso es Soria con 440 kilómetros (se muestra árbol, se gira la imagen por espacio en el trabajo).

11. Establecemos tabla de origen y destino desde Guadalajara a sus respectivas conexiones (se muestra árbol):

13. Determinamos los kilómetros por carretera y la fórmula de f(n) = h(n) + g(n), la tabla queda de la siguiente manera:

14. A partir de esta distancia Euclídea tomamos la menor que en este caso es Soria con 300 kilómetros (se muestra árbol, se gira la imagen por espacio en el trabajo).

16. Ahora como Cuenca tiene ruta directa a Valencia entonces tomamos esa ruta y descartamos las demás.

18. Por último, determinamos la ruta completa y/o final que en este caso sería: Logroño – Soria, Soria – Guadalajara, Guadalajara – Cuenca y Cuenca - Valencia. A continuación, el árbol de ruta, se gira la imagen por espacio en el trabajo:

19. Determinamos el kilometraje total de la ruta como paso final terminando el algoritmo A\* dando un total de 605 kilómetros por carretera (lista cerrada de estado).

#### Análisis Algoritmo A\*

Para el análisis del resultado, selecciono la distancia euclídea, tomo la distancia en carretera, determino cual es la ruta óptima para el algoritmo que toma como información referente f(n) que es la función de evaluación, h(n) hace referencia a la distancia euclídea y g(n) hace referencia al costo de la ruta desde su origen hasta algún nodo n del grafo que en este caso es la distancia en carretera, a continuación, una imagen explicativa:

Determinamos la ruta óptima que es la ruta: Logroño–Soria, Soria–Guadalajara, Guadalajara–Cuenca y Cuenca–Valencia que nos da un total de 605 kilómetros de carretera.

#### Programa Algoritmo A\*

El desarrollo se ejecutó en la plataforma Python – Anaconda – Spyder, que se adjunta en documentos aparte de acuerdo a lo indicado, se denota que se desarrolló el algoritmo en adición para las ciudades de Medellín y Cali, pero no se adjuntan en este desarrollo o de lo contrario la actividad sobrepasa la extensión máxima del trabajo a desarrollar.

RESPUESTAS A LAS DOS PREGUNTAS

 ¿Es la heurística escogida adecuada? Justifica tu respuesta.

En el desarrollo de esta actividad como se pudo denotar, se tomó como heurística la Distancia Euclidiana, la cual se considera es una heurística adecuada para este tipo de algoritmos como es el Voraz y el A\*, ya que se basa en un contexto que en este caso es la ruta óptima entre dos ciudades y adicionalmente se basa en las características de las regiones como podemos ver en los kilómetros frente a las distancias en las cuáles incluso dependen de la topografía. Adicionalmente es una heurística admisible que desde mi punto de vista ya que garantiza que el algoritmo va a encontrar una solución óptima y no sobreestima el costo. Además, es al ser simple permite que el algoritmo funcione efectivamente, reduciendo los largos tiempos de procesamiento del computador, lo cual algunas veces genera bastante estrés. Ahora como es visto en esta actividad y en problemas donde los movimientos pueden ejecutarse en cualquier dirección, esta heurística remite esa distancia esperada entre dos puntos, que es lo esperado en temas como la navegación de mapas como es el caso. Y por último es preciso indicar que esta heurística ayuda a predecir el comportamiento durante el proceso de búsqueda de la solución óptima por lo que definitivamente se considera la heurística adecuada en este entorno.

 ¿Cuál consideras que ha sido el algoritmo que mejor se ha comportado para resolver este problema? Explica tu respuesta.

Frente a la resolución de esta actividad donde se debe encontrar la ruta más corta o yo diría la ruta óptima entre ciudades, desde mi punto de vista considero que el algoritmo A\* se comporta mejor que el algoritmo Voraz y esto, frente al hecho que el Algoritmo Voraz indico menos kilómetros en la ruta óptima. Pero también está el hecho que se debe velar por la optimización ya que por algo nos dan la distancia euclídea y la distancia por carretera, que esta última se basa en registros topográficos, lo que abarca una mayor distancia y de igual manera el algoritmo A\* abarca una mayor exactitud y fortaleza. Ahora mi análisis es que este mejor comportamiento se da en el algoritmo A\* ya que este algoritmo evalúa tanto el costo de llegar a un nodo como una estimación del costo hasta el objetivo. El algoritmo voraz, en cambio, solo se basa en la heurística, lo que puede llevar a soluciones que sinceramente no son del todo óptimas, y sinceramente desde mi punto de vista eso es algo que me hace dudar, yo soy más del tipo de persona que se basa más en cifras y datos que me lleven a una solución más efectiva y por ende más confiable, y no es el hecho de obtener un resultado, es el hecho de determinar qué tan confiable es ese resultado. Ahora en situaciones donde hay múltiples caminos y costos variables, A\* tiende a manejar mejor la complejidad y las interacciones entre diferentes rutas, lo que a menudo resulta en soluciones más confiables, como es el hecho de la cantidad de kilómetros en ciertas rutas por carretera frente a su estado topográfico como es el hecho de la ruta Guadalajara–Teruel con 244 kilómetros cuya geografía según lo indicado en clase es compleja. Para culminar quiero indicar que, en la mayoría de los casos, A\* es preferido por su capacidad para encontrar la solución óptima, mientras que el algoritmo voraz puede ser útil en situaciones donde se requiere rapidez y no se necesita una solución óptima, lo cual es contraproducente en el desarrollo de esta actividad ya que estamos buscando la solución óptima de una ruta a tomar entre dos ciudades. Y en el caso que se obtenga una mejor solución de manera consistente con el algoritmo voraz, es importante considerar el contexto y los criterios específicos del problema, ya que desde mi punto de vista puede que no sea la solución más fiable y por ende toca entrar a analizar muy bien, lo que implica más tiempo para el desarrollador y pues está el riesgo de una mayor incertidumbre en los resultados obtenidos.

### CONCLUSIÓN DE LA ACTIVIDAD

Para concluir este trabajo queremos indicar que, utilizando tanto el Algoritmo Voraz como el Algoritmo A\*, ofrecen métodos avanzados para resolver problemas de caminos y optimización en IA. Entre las ventajas del Algoritmo Voraz podemos denotar su velocidad por lo que es útil cuando se requiere una rápida solución. Por otra parte, el Algoritmo A\*, es un algoritmo que equilibra el costo acumulado lo que permite encontrar el camino óptimo. De nuestra parte es importante precisar que en problemas donde el rendimiento es una prioridad sobre la exactitud, el Algoritmo Voraz puede ser una buena opción. Sin embargo, para la mayoría de las aplicaciones de IA donde se requiere una solución óptima y precisa, el algoritmo A\* nos dio a entender cómo la combinación de la heurística y el costo acumulado generan una optimización frente a la toma de decisiones en problemas que para nuestro gusto son de una envergadura más compleja como es el caso de determinar una ruta óptima que puede recorrer todo un país de una ciudad a otra. Queremos finalizar indicando que esta es una actividad que nos deja mucho análisis y aplicación en cuanto a la temática de la búsqueda informada ya que nos brinda dos herramientas esenciales en los procesos de la inteligencia artificial, que son efectivos desde la optimización del problema que busquemos solucionar y su nivel de prioridad.

### BIBLIOGRAFÍA