

**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 GELVES**

**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

<b>INTRODUCCIÓN .....</b>	<b>3</b>
<b>TÍTULO DEL PROYECTO .....</b>	<b>3</b>
<b>DESARROLLO DEL PROYECTO .....</b>	<b>3</b>
<b>Requisitos Del Sistema .....</b>	<b>3</b>
<b>Diseño De La Solución .....</b>	<b>3</b>
Recursos necesarios .....	3
<b>Metodología De Desarrollo .....</b>	<b>4</b>
Metodología .....	4
Justificación .....	4
Roles .....	4
Perfiles De Recursos Humanos .....	4
Despliegue Del Sistema .....	5
Securización Y Anonimización De Datos .....	5
Área De Investigación .....	6
<b>Desarrollo De La Metodología .....</b>	<b>6</b>
Algoritmo Voraz .....	6
<b>CONCLUSIÓN DE LA ACTIVIDAD .....</b>	<b>10</b>
<b>BIBLIOGRAFÍA .....</b>	<b>10</b>

## INTRODUCCIÓN

En el mercado actual, 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, el cual se aplicará 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 del algoritmo 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 El Algoritmo Voraz En Colombia”.

## DESARROLLO DEL PROYECTO

Este desarrollo está orientado a resolver problemas como: Determinar las rutas más cortas entre un almacén y varios puntos de entrega. Adicionalmente, comparar el rendimiento del algoritmo 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 el algoritmo maneja problemas de optimización. Esto también abre la puerta a futuros ajustes, como la incorporación de factores adicionales (tiempos, capacidades, etc.), en sistemas más complejos y la implementación de algoritmos más robustos, pero como este es un inicio vamos iniciar primero por algo no tan complejo como un algoritmo genético.

## 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 (directas entre ciudades) basadas en kilómetros tradicionales 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. Este es un servidor equipado con unidades de procesamiento gráfico, diseñadas originalmente para acelerar gráficos y renderizado (mapas de carreteras). Estos servidores son ideales para aplicaciones que requieren un alto rendimiento en cálculos paralelos, como el entrenamiento de modelos de inteligencia artificial, aprendizaje profundo, simulaciones científicas y renderizado 3D. Adicionalmente es importante contar con 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, etc.).
- **Software:** Python (bibliotecas como HEAPQ, DEAP, NumPy, Pandas). Herramientas de visualización: Python, Anaconda-Spyder. Entorno para integración y pruebas: NetBeans y Anaconda. A continuación, una tabla con los enfoques y uso principal de las bibliotecas en el desarrollo de este laboratorio:

Biblioteca	Enfoque	Uso Principal
heapq	Estructuras de datos (cola de prioridad)	Operaciones rápidas con heaps.
deap	Algoritmos evolutivos	Optimización y simulación genética.
NumPy	Matemáticas y álgebra lineal	Operaciones con matrices y vectores.
Pandas	Análisis de datos tabulares	Procesar y manipular tablas de datos.

- **Técnicas empleadas:** Algoritmo Voraz 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 enfocamos en la metodología Scrum frente a un marco de trabajo ágil utilizado principalmente en el desarrollo de software. Está diseñado para gestionar proyectos complejos y cambiantes como es el caso de rutas a diferentes ciudades en Colombia frente a 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.

### Justificación

Como justificación es preciso indicar que la metodología Scrum fomenta la autoorganización y la colaboración entre los miembros del equipo y que mejor manera de gestionar una autoorganización en una empresa que dando a conocer las rutas optimas que deben seguir sus procesos de distribución. Adicionalmente 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, 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 (revisar muy bien la heurística para que tenga un mayor enfoque en los datos de los kilómetros a recorrer entre cada ciudad). 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:** Persona es la encargada de definir los requisitos y asegura la alineación con 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

Los perfiles adecuados por parte de recursos humanos para la ejecución de este proyecto en una organización logística:

- **Ingeniero de datos:** Persona encargada de todo lo concerniente a la recopilación, limpieza y preparación de los datos.
- **Científico de datos-Ingeniero Informático:** Persona encargada de desarrollar y ajustar los algoritmos de búsqueda como en este caso el algoritmo Voraz, es claro que el proyecto se puede adaptar a algoritmos genéticos, pero pues no se van a implementar en este momento en el 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:** Persona que va a diseñar la infraestructura de HW y SW, que incluso al manejar tablas puede generar un acople de implementación en MySQL con Python utilizando bibliotecas específicas como mysql-connector-python, PyMySQL o SQLAlchemy si se necesitan más funcionalidades.
- **Desarrollador Full Stack – Ingeniero Informático:** Persona encargada de 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:** Persona encargada de definir las restricciones y escenarios reales del problema, tiene claridad de todos los procesos logísticos, manejo de WMS (Warehouse Managment System), manejo de MRP (Material Request Planning), manejo de MLP

(Management Logistics Processes), Lead Times, MPS (Management Production System), CRM, entre otros. Y da un enfoque claro de las posibles implementaciones en el desarrollo de este proyecto logístico.

- **Analista de seguridad – Ingeniero informático especialista en Ciberseguridad:** Persona encargada de asegurar la protección de los datos sensibles, que no 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:** Tener un servidor en la nube (AWS o Azure) para escalabilidad y procesamiento de datos, que permita la operabilidad en tiempo real, adicionalmente tener dispositivos IoT en vehículos para recolectar datos en tiempo real frente a sus distancias recorridas y tiempos de entregas.
- **Software:** 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. Asegurando que la aplicación funcione de manera consistente en cualquier entorno, ya sea en desarrollo, pruebas o producción. Es importante precisar que estos contenedores se utilizan 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. Por otra parte, CI/CD se espera utilizar Jenkins para actualizaciones, y dar claridad para integrar y desplegar cambios de software de manera continua y eficiente. Es importante indicar que Jenkins es una herramienta de integración continua que permite orquestar y automatizar las tareas necesarias para construir, probar y desplegar aplicaciones en diferentes entornos como es el caso de modificaciones en la estructura del backend al modificar los kilómetros por carretera de las ciudades con base en los entornos de prueba y campo, para ajustar.
- **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 que van a ir incluidas en el desarrollo del backend que en este caso no solo serían restricciones de vehículos y costos, sino también de horarios de recepción y entrega, y personal de carga y descarga.

## Securización Y Anonimización De Datos

- **Anonimización:** Enmascarar información personal de clientes. Se basa en 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 que se van a manejar en cuanto a clientes, proveedores y demás usuarios a los que haya lugar.
- **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 híbrido 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 (Ejemplo: clientes para entrega de mercancías en Colombia).

### Área De Investigación

El área de investigación de este laboratorio se basa en la Computación Bioinspirada en el uso de algoritmos para resolver problemas de optimización en entornos dinámicos que en este caso por simplicidad solo vamos a implementar el algoritmo de búsqueda que es el Algoritmo Voraz, donde se recalca que el objetivo es utilizar algoritmos más avanzados en el futuro como son los genéticos como es el caso del 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, pero está en visto hasta que no se pidan más pautas de desarrollo.

### Desarrollo De La Metodología

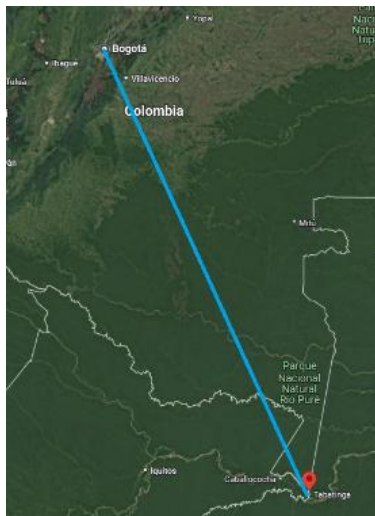
De acuerdo a lo planteado para el desarrollo de esta metodología es preciso indicar que un problema de búsqueda consiste en ejecutar una modelación de tal manera que la solución corresponda con un camino que nos lleve desde el estado inicial hasta un estado meta. Ahora con base en la actividad expuesta a continuación vamos a denotar un problema de búsqueda en el rubro del problema del camino más corto mediante búsqueda informada a través del Algoritmo Voraz.

### 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. 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. Entonces para este desarrollo primero que todo vamos al mapa brindado y determinamos la ruta a establecer **(las imágenes se pueden ampliar)**, en este caso vamos a tomar la ruta de Bogotá a Leticia.



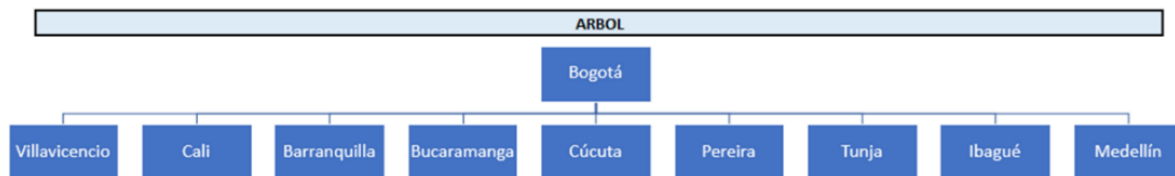
2. Ahora, desarrollamos los Valores De Referencia Y Distancias Por Carretera, para esto la distancia que tomamos de referencia va a ser la distancia que va a tomar la ciudad inicio-origen que es la ciudad de Bogotá. Entonces nos remitimos a la tabla de distancias y seleccionamos la columna de referencia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

	A	AM	AN	AO	AP	AQ	AR	AS	AT
1	Origen/Destino	Quibdó	Istmina	Tadó	Montería	Cereté	Lorica	Bogotá	Soacha
2	Acacias	670	680	680	410	400	420	145	150
3	Aguachica	460	470	460	310	300	290	490	490
4	Aguazul	630	640	630	700	690	700	230	230
5	Apartadó	460	470	460	180	170	180	430	430
6	Arauca	740	750	740	850	840	850	540	600
7	Armenia	600	590	600	410	400	420	260	270
8	Barrancabermeja	740	730	740	570	560	580	220	220
9	Barranquilla	550	540	550	370	360	380	640	630
10	Bello	430	420	410	350	340	350	230	230
11	Bogotá	490	480	470	480	490	500	0	15
12	Bucaramanga	730	720	730	550	540	560	120	120
13	Buenaventura	840	830	840	560	550	570	460	460
14	Buga	620	610	620	430	420	440	330	330
15	Calarcá	680	670	680	490	480	500	276	300
16	Cali	820	810	820	540	530	550	440	440
17	Cartagena de Indias	540	530	540	370	360	380	640	630
18	Cereté	460	450	440	10	0	50	490	490
19	Chiquinquirá	920	930	920	620	610	620	147	130

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

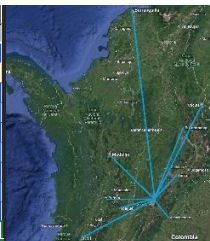
	A	B	C	D	E	F	AS
1	Origen/Destino	Leticia	Puerto Nariño	Medellín	Bello	Itagüí	Bogotá
2	Acacias	1045	1005	250	260	270	145
3	Aguachica	1280	1260	520	510	500	490
4	Aguazul	730	710	600	610	620	230
5	Apartadó	1050	1020	170	160	170	430
6	Arauca	340	310	570	550	560	540
7	Armenia	650	610	130	140	150	260
11	Bogotá	1100	1040	240	230	230	0
14	Buga	1180	1130	150	140	130	330
15	Calarcá	1040	1000	170	160	150	276
16	Cali	1030	990	240	230	220	440
17	Cartagena de Indias	1020	1000	635	625	615	640
22	Cúcuta	700	660	380	370	380	420
24	Duitama	1040	1000	220	220	210	180
27	Espinal	1060	1020	180	170	160	137
29	Florencia	904	340	330	340	350	539
34	Ibagué	1123	1040	210	200	190	191
36	Ipiales	480	450	380	390	400	520
38	Itagüí	1500	1503	8	9	0	230
40	Leticia	0	80	805	812	1500	1100
41	Lorica	1140	1120	360	350	340	500

4. Procedemos a desarrollar el Algoritmo Voraz, y como toma una sola función de referencia que en este caso es la ruta de Bogotá a Leticia, y teniendo como referencia la distancia Euclídea como  $H(n)$ , entonces para ir de Bogotá a Leticia primero verificamos las conexiones de Bogotá y son las siguientes (se muestra árbol); Bogotá a Villavicencio, Bogotá a Cali, Bogotá a Barranquilla, Bogotá a Bucaramanga, Bogotá a Cúcuta, Bogotá a Pereira, Bogotá a Tunja, Bogotá a Ibagué, y Bogotá a Medellín.



5. Ahora determinamos las distancias de las ciudades conexión a Bogotá y nos da la siguiente tabla en Kilómetros:

	A	D	N	V	AS	BH	BN	BU	BY	CE	CH
1	Origen/Destino	Medellín	Barranquilla	Tunja	Bogotá	Villavicencio	Cúcuta	Pereira	Bucaramanga	Ibagué	Cali
9	Barranquilla	640	0	580	640	660	670	530	670	570	510
11	Bogotá	240	640	130	0	75	420	180	350	191	440
12	Bucaramanga	150	670	350	350	190	180	250	0	350	340
16	Cali	240	510	350	440	330	510	240	340	240	0
22	Cúcuta	380	670	400	420	540	0	500	180	410	510
34	Ibagué	210	570	150	191	160	410	140	350	0	240
46	Medellín	0	640	270	240	210	380	170	150	210	240
57	Pereira	170	530	300	180	290	500	0	250	140	240
83	Tunja	270	580	0	130	210	400	300	350	150	350
90	Villavicencio	210	660	210	75	0	540	290	190	160	330



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

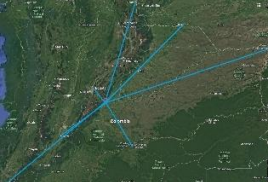
	A	D	N	V	AS	BH	BN	BU	BY	CE	CH
1	Origen/Destino	Medellín	Barranquilla	Tunja	Bogotá	Villavicencio	Cúcuta	Pereira	Bucaramanga	Ibagué	Cali
9	Barranquilla	640	0	580	640	660	670	530	670	570	510
11	Bogotá	240	640	130	0	75	420	180	350	191	440
12	Bucaramanga	150	670	350	350	190	180	250	0	350	340
16	Cali	240	510	350	440	330	510	240	340	240	0
22	Cúcuta	380	670	400	420	540	0	500	180	410	510
34	Ibagué	210	570	150	191	160	410	140	350	0	240
46	Medellín	0	640	270	240	210	380	170	150	210	240
57	Pereira	170	530	300	180	290	500	0	250	140	240
83	Tunja	270	580	0	130	210	400	300	350	150	350
90	Villavicencio	210	660	210	75	0	540	290	190	160	330

7. Verificamos las conexiones de Villavicencio y son (se muestra árbol); Villavicencio a Bogotá, Villavicencio a Cúcuta, Villavicencio a Pasto, Villavicencio a Neiva, Villavicencio a San José Del Guaviare, Villavicencio a Arauca, Villavicencio a Puerto Carreño.



8. Ahora determinamos las distancias de las ciudades conexión a Villavicencio y nos da la siguiente tabla en Kilómetros:

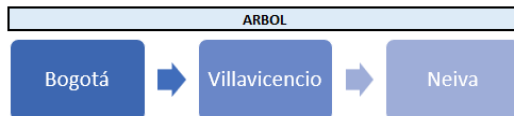
	A	K	AS	AY	AZ	BH	BK	BN	CN
1	Origen/Destino	Arauca	Bogotá	San José del Guaviare	Neiva	Villavicencio	Pasto	Cúcuta	Puerto Carreño
6	Arauca	0	540	150	420	410	630	210	701
11	Bogotá	540	0	460	304	75	460	420	1211
22	Cúcuta	210	420	540	390	540	160	0	700
51	Neiva	420	304	380	0	170	250	390	951
56	Pasto	630	460	580	250	350	0	160	901
62	Puerto Carreño	701	1211	766	951	1051	901	700	0
69	San José del Guaviare	150	460	0	380	285	580	540	766
90	Villavicencio	410	75	285	170	0	350	540	1051



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

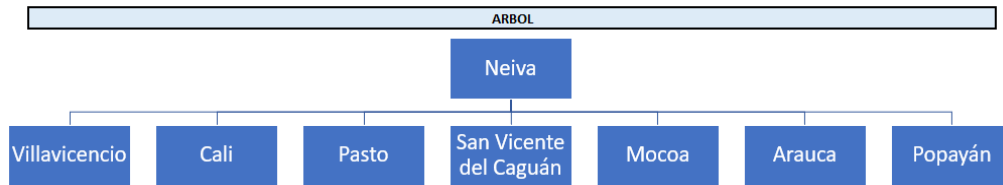
	A	K	AS	AY	AZ	BH	BK	BN	CN
1	Origen/Destino	Arauca	Bogotá	San José del Guaviare	Neiva	Villavicencio	Pasto	Cúcuta	Puerto Carreño
6	Arauca	0	540	150	420	410	630	210	701
11	Bogotá	540	0	460	304	75	460	420	1211
22	Cúcuta	210	420	540	390	540	160	0	700
51	Neiva	420	304	380	0	170	250	390	951
56	Pasto	630	460	580	250	350	0	160	901
62	Puerto Carreño	701	1211	766	951	1051	901	700	0
69	San José del Guaviare	150	460	0	380	285	580	540	766
90	Villavicencio	410	75	285	170	0	350	540	1051

10. En este momento la ruta trazada como óptima es: Bogotá a Villavicencio, y Villavicencio a Neiva, cuya ruta la podemos ver en la imagen a continuación denotada con flechas azules en el mapa y en el árbol:



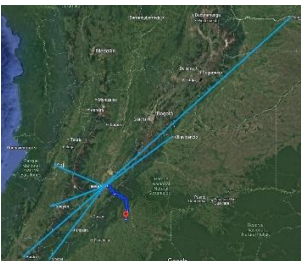
11. Ahora determinamos las conexiones de Neiva (en este caso se denota a San Vicente del Caguán con una línea azul oscura y un punto de ubicación rojo para la fácil denotación de la ubicación). A continuación, las conexiones y árbol: Neiva a Villavicencio, Neiva a Cali, Neiva a Pasto, Neiva a San Vicente Del Caguán, Neiva a Mocoa, Neiva a Arauca, y Neiva a Popayán.





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

	A	K	AE	AI	AZ	BH	BK	BQ	CH
1	Origen/Destino	Ar	San Vicente del Caguán	Popayán	Nei	Villavicencio	Pas	Moc	Ca
6	Arauca	0	250	620	420	410	630	330	570
16	Cali	570	700	170	360	330	540	550	0
49	Mocoa	330	180	140	500	530	140	0	550
51	Neiva	420	90	780	0	170	250	500	360
56	Pasto	630	400	140	250	350	0	140	540
60	Popayán	620	270	0	780	330	140	140	170
70	San Vicente del Caguán	250	0	270	90	330	400	180	700
90	Villavicencio	410	330	330	170	0	350	530	330

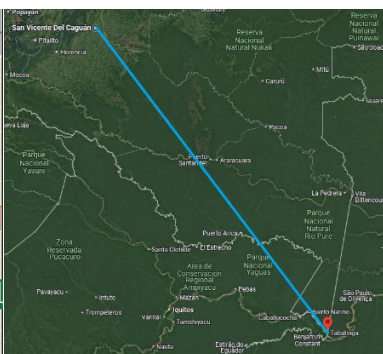


13. A partir de esta distancia tomamos la menor que en este caso es San Vicente Del Caguán con 90 kilómetros y la ruta a seguir entonces es Bogotá a Villavicencio, y Villavicencio a Neiva, Neiva a San Vicente Del Caguán:

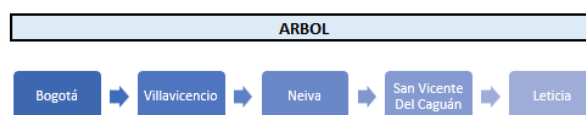


14. Y como San Vicente Del Caguán tiene ruta directa a Leticia entonces tomamos esa ruta y descartamos las demás para determinar la ruta óptima que en este caso es: Bogotá a Villavicencio, y Villavicencio a Neiva, Neiva a San Vicente Del Caguán, San Vicente Del Caguán a Leticia. A continuación de distancia de San Vicente Del Caguán a Leticia con 420 kilómetros.

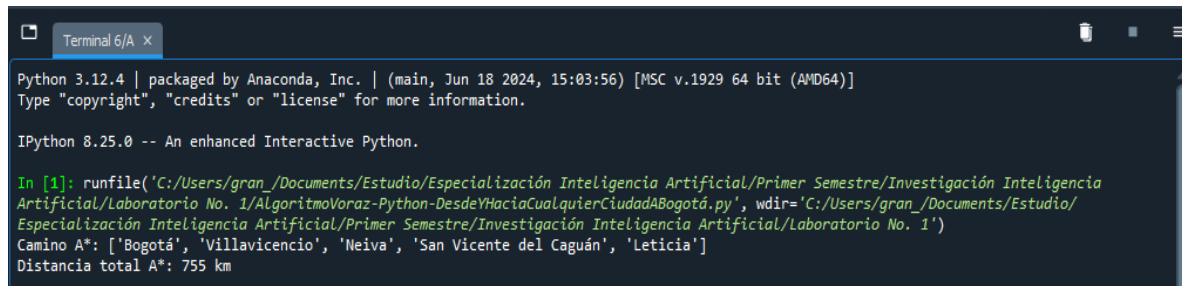
	A	B	AE
1	Origen/Destino	Leticia	San Vicente del Caguán
9	Barranquilla	1050	320
11	Bogotá	1100	350
12	Bucaramanga	1080	600
16	Cali	1030	700
22	Cúcuta	700	510
34	Ibagué	1123	580
40	Leticia	0	420
46	Medellín	805	460
57	Pereira	1000	560
70	San Vicente del Caguán	420	0
83	Tunja	1070	630
90	Villavicencio	1040	330



15. Ahora determinamos el kilometraje total de la ruta óptima determinada como paso final terminando el algoritmo Voraz y dando un total de 755 kilómetros por carretera, a continuación, las mediciones de las carreteras y 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. Por ende, el total de kilómetros recorridos son:  $75+170+90+420 = 755$  kms.



16. 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):



```
Python 3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:03:56) [MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.25.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/gran/Documents/Estudio/Especialización Inteligencia Artificial/Primer Semestre/Investigación Inteligencia Artificial/Laboratorio No. 1/AlgoritmoVoraz-Python-DesdeYHaciaCualquierCiudadABogotá.py', wdir='C:/Users/gran/Documents/Estudio/Especialización Inteligencia Artificial/Primer Semestre/Investigación Inteligencia Artificial/Laboratorio No. 1')
Camino A*: ['Bogotá', 'Villavicencio', 'Neiva', 'San Vicente del Caguán', 'Leticia']
Distancia total A*: 755 km
```

### Análisis Algoritmo Voraz

Para este análisis seleccionamos las distancias directas en carretera en kilómetros, determinamos cual es la ruta óptima para el Algoritmo Voraz que toma como información referente esas distancias, determinamos la ruta óptima y en este caso el resultado de la ruta óptima es la ruta: Bogotá a Villavicencio, Villavicencio a Neiva, Neiva a San Vicente Del Caguán, San Vicente Del Caguán a Leticia, que nos da un total de 755 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 en caso de requerir pruebas del código.

### CONCLUSIÓN DE LA ACTIVIDAD

Para concluir este trabajo queremos indicar que, utilizando el Algoritmo Voraz podemos obtener 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. Ahora, es importante precisar que el Algoritmo Voraz es una técnica de búsqueda informada que selecciona, en cada paso, el nodo que parece más prometedor según una heurística y es idóneo para problemas de rutas óptimas en kilómetros cuando la heurística empleada es efectiva, ya que selecciona siempre el nodo que parece más cercano al destino esto lo hace rápido y eficiente para encontrar rutas en escenarios donde las distancias son bien representadas por la heurística. Ahora es preciso indicar que si se requiere un mayor detalle o exactitud podemos utilizar el algoritmo A\* donde se hace una referencia tanto de la distancia Euclídea como la distancia real en kilómetros, pero en un país como Colombia esto puede ser complejo por su geografía, por esto se procedió a generar una heurística más efectiva y se tomó el Algoritmo Voraz. Para culminar queremos indicar que el Algoritmo Voraz es idóneo para encontrar rutas óptimas en kilómetros en problemas donde la heurística (como la distancia en línea recta al destino) refleja con precisión los costos reales. Su enfoque de seleccionar siempre el nodo más cercano al objetivo permite calcular rutas rápidamente, lo que lo hace especialmente útil en sistemas donde la velocidad es prioritaria.

### BIBLIOGRAFÍA

A continuación, la bibliografía implementada en la búsqueda de información:

- Tema 7. Gestión de proyectos de inteligencia artificial. Aula Virtual Investigación en Inteligencia Artificial (COEINTAR) octubre 2024 PER 12153.
- Clases virtuales con el profesor Ing. Diego Osorio Reina.
- API de mapas en Google Maps.