

Optimización del Transporte en la Logística de  
reforestación en línea de transmisión eléctrica  
Dominica - Charcas

Mandujano L., Fedra Fernanda  
A00835797@tec.mx

Priego B., Roberto  
A01285539@tec.mx

Gutiérrez V., Diego  
A01285421@tec.mx

Castro T., Juan Marco  
A01742821@tec.mx

Abril 2024

Instituto Tecnológico y de Estudios Superiores de Monterrey

# 1 Introducción

## 1.1 Introducción de la Problemática

La deforestación tiene un impacto significativo en la crisis del agua. Con la pérdida de bosques se modifica el sistema natural que asegura el flujo regular de agua, la limpieza del aire y la protección ante inundaciones y sequías, esto sucede porque el bosque actúa como una esponja que absorbe la lluvia y la libera en intervalos regulares de tiempo mientras retiene los suelos moderando los efectos destructivos de las inundaciones y la sequía. Al perderse esta capa protectora, el agua fluye muy rápido a los arroyos elevando los niveles de ríos exponiendo asentamientos humanos durante la época de lluvias. Por si fuera poco, el bosque aumenta la humedad local a través de su transpiración, contribuyendo a la precipitación local. (National Geographic, 2024)

La contribución al cambio climático es significativa ya que la deforestación disminuye la capacidad de los bosques para absorber dióxido de carbono, uno de los principales gases de efecto invernadero. La pérdida de árboles y la degradación de los ecosistemas forestales liberan grandes cantidades de carbono a la atmósfera, contribuyendo al calentamiento global.

Con respecto al impacto económico y social de la deforestación a largo plazo, la degradación de los recursos forestales puede afectar negativamente a las economías locales y nacionales al agotar los recursos naturales renovables y reducir la capacidad de generación de ingresos a través de actividades como el eco-turismo y la explotación sostenible de los recursos forestales. (Rhett Butler, 2009)

La deforestación también puede contribuir a la pobreza y la inestabilidad social al degradar los ecosistemas naturales que son vitales para el sustento de comunidades locales y al afectar la disponibilidad de recursos como el agua y los alimentos.

Para revertir los problemas ocasionados por este gran problema, se están implementando varias estrategias a nivel global, que incluyen la gestión cuidadosa de los recursos vegetales, la tala equilibrada y la reforestación, el control de la agricultura comercial, la reducción del consumo de productos asociados a la deforestación, la vigilancia y el control de las actividades tanto legales como ilegales que destruyen los bosques, así como la cooperación internacional puesto que este es un problema que afecta a todo el mundo. (Gómez, 2023)

Las medidas mencionadas buscan frenar la deforestación pero se debe también recuperar los bosques perdidos, y esto se logra a través de la reforestación y para ello primero se debe de decidir el tipo de plantas a colocar en los terrenos tomando en cuenta criterios como la adecuación al entorno, las condiciones del suelo, el clima, la disponibilidad del agua y la calidad de la planta. Con respecto a la distribución de especies, se planifica considerando el objetivo, la topografía, la cantidad de plantas disponibles y la distancia entre ellas aconsejando la plantación al inicio de la temporada de lluvias para aprovechar la humedad y dar tiempo a que las plantas arraiguen. (Gómez, 2023)

La deforestación en México ha experimentado cambios significativos desde

la última década. Un ejemplo de esto y según el Sistema Satelital de Monitoreo Forestal (SAMOF), en 2021, la tasa anual de deforestación se redujo en un 25.94% con respecto a 2019 por la implementación de acciones gubernamentales para reducir la deforestación bruta en el país, incluyendo la Estrategia Nacional para la Reducción de Emisiones por Deforestación y Degradación Forestal (ENAREDD+), programas de incentivos para actividades agropecuarias sustentables, y la promoción de sistemas agrosilvopastoriles. (C.N.F., 2022)

Además, se observa una disminución en la deforestación en áreas específicas como la Reserva de la Biosfera de la Mariposa Monarca, donde la tala clandestina se redujo significativamente gracias a la colaboración entre la Gendarmería Ambiental y la PROFEPA. (C.N.F., 2022)

La deforestación en México ha mostrado una tendencia a la baja debido a la implementación de programas gubernamentales, acciones de supervisión y vigilancia, así como a la colaboración con organizaciones internacionales y locales. (C.N.F., 2022)

## 1.2 Justificación del Problema

En la actualidad, el cambio climático representa una de las mayores amenazas globales que enfrenta nuestro planeta, afectando de manera significativa a México y sus ecosistemas. A pesar de la creciente conciencia sobre la importancia de mitigar el cambio climático, la deforestación continúa siendo un problema grave en México, contribuyendo al aumento de las emisiones de gases de efecto invernadero, pérdida de hábitats naturales, la degradación del suelo y la disminución de la biodiversidad.

La Comisión Nacional Forestal (CONAFOR) desempeña un papel crucial en la promoción y coordinación de actividades de reforestación en México. Sin embargo, la planificación y gestión logística de la siembra de especies forestales enfrentan desafíos relacionados con la asignación óptima de recursos, la distribución eficiente de plantas y la optimización de rutas de transporte. (C.N.F., 2022)

Por lo tanto, la necesidad de desarrollar un modelo matemático que permita al personal de CONAFOR planificar de manera eficiente la asignación y distribución de especies forestales a través de unidades de transporte se vuelve necesario. Este modelo no solo facilitará la toma de decisiones basada en datos y métodos matemáticos, sino que también contribuirá a reducir los tiempos de entrega, los costos operativos y a promover una distribución homogénea de las plantas a sembrar. (C.N.F., 2022)

### **1.3 Objetivo**

Desarrollar un modelo matemático eficiente para disminuir los costos asociados, la optimización de los tiempos de recorrido, la planificación, y gestión logística de la reforestación en México, con el fin de mejorar la asignación de especies y la distribución de plantas a través de unidades de transporte, considerando las restricciones y requerimientos específicos de siembra por polígono.

## 2 Trabajo relacionado

### 2.1 Investigación sobre la flora en el altiplano mexicano

El Altiplano mexicano es una de las zonas en las cuales se divide el país. Caracterizado por sus terrenos altos, que se extiende desde el río Bravo al norte, hasta Eje Neo-volcánico, en el sur (Gómez, 2023). Los estados de la república que conforman el altiplano son:

- Chihuahua
- Nuevo León
- Durango
- Zacatecas
- San Luis Potosí
- Aguascalientes
- Guanajuato
- Sonora
- Jalisco
- Querétaro
- Ciudad de México

Entre las especies mas reconocidas del país, tenemos la Mangolia mexicana (Yoloxochitl), una planta reconocida históricamente utilizada antes de la conquista como una planta aromática y curativa. La siguiente seria la Biznaga, un tipo de cactus bastante común en las zonas áridas y centro del país, caracterizado por tener un lento y longevo crecimiento. Por ultimo, se encuentra el Maguey, un cactus que tiene la forma de un candelabro. Además, caracterizado por dar frutos similares a las fresas durante la primavera. (Semarnat, 2002)

## 2.2 Diseño de plantación denominado tres bolillos

El método de tres bolillos se refiere a como es que se tiene que hacer la plantación con la forma de triángulo. El objetivo de este tipo de plantación es lograr distribución mas uniforme y eficiente en un campo. La técnica en si se basa en un patrón similar a la de un tablero de ajedrez, en las que las semillas se plantan de forma diagonal con una distancia similar, lo que permite aprovechar los espacios sin dejar ninguna área vacía entre las plantas y reduciendo el riego de plagas o enfermedades. (Gonzalez-L, 2015)

Este sistema de plantación puede ser usado por cualquier tipo de planta. Sin embargo, hay plantas que aprovechan más esto como lo puede ser a la hora de plantar lechugas o brócolis, plantas ideales porque requieren una buena exposición al sol y una siembra densa donde se encuentre al rededor de otras lechugas o brócolis.

## 2.3 Modelos o formas de resolver el problema de ruteo de vehículos

Esta clase de problemas en si son conocidos como Vehicle Routing Problem (VRP). Existen distintas formas de aplicar una solución a esta clase de problemas y las mas usadas y eficientes serian:

- Ruteo Dinámico (DVRP)

Es conocida principalmente por ser una solución algorítmica que hace planeación de entregas para el día siguiente, sin tener que esperar hasta el fin de una jornada (ciclo). Como tal, lo que se hace es optimizar pedidos en ejecución para poder evolucionar el resultado a medida que va pasando el tiempo. Esto afecta la gestión de vehículos utilizados para transportar y el tiempo de la entrega final del pedido. (Gonzalez-L, 2015)

- VRP con múltiples centros de distribución (MDVRP)

En este caso, se enfocan en buscar la ruta mas óptima de múltiples depósitos donde se encuentre tu producto a los grupos de personas. Su objetivo principal es claramente reducir las distancias de distribución considerando la capacidad de flota vehicular y ubicación de depósitos. (Gonzalez-L, 2015)

- VRP con recogidas y entregas (VRPPD)

Es una solución diseñada para agilizar y optimizar el flujo, rutas y organizar múltiples vehículos con restricciones de recoger y entregar. Esto modela una situación donde una flota de vehículos es asignada a cumplir un numero de ordenes en las que se especifican la ubicación de entrega. (Gonzalez-L, 2015)

- Técnicas de Clustering

Como tal, el clustering (o agrupamiento en español) es una técnica para agrupar clientes en grupos para luego encontrar las rutas más óptimas dentro de cada cluster. (Gonzalez-L, 2015)

- VRP estocástico (SVRP)

Esta técnica se basa totalmente en elementos aleatorios, donde existen múltiples variables que pueden cambiar como lo pueden ser demanda, el tiempo de viaje, condiciones de tráfico, etc. Todas estas variables pueden ser aleatorias o sujetas a cambios, lo que complica la planificación de rutas. (Gonzalez-L, 2015)

## 3 Definición del problema

### 3.1 Definir el problema

En el presente reto se busca atender el problema de la reforestación en México y proponer un modelo matemático que permita al personal de CONAFOR planear eficientemente la asignación de especies y cantidad de individuos a unidades de transporte de modo que cumplan con los requerimientos establecidos de siembra para cada polígono asignado. Se debe crear rutas de distribución entre el depósito y los diversos polígonos que se están reforestando, disminuyendo los tiempos de entrega, los costos de operación y creando una mejor distribución de las plantas a sembrar, buscando una mejor homogenización para evitar el monocultivo y por ende la infertilización del suelo. Con esto en mente se considerarán diversas restricciones propias del problema como pueden ser la jornada laboral, la capacidad de las unidades de transporte, demanda de plantas, entre otras.

### 3.2 Identificación de las variables y posibles parámetros a tomar en cuenta para la solución del problema

#### 3.2.1 Variables

1.  $X_{ij}$ : Cantidad de planta "i" en polígono "j"
2.  $Y_i$ : Si se usa una ruta o no
3.  $R_i$ : Rutas que sigue la camioneta para plantar
4.  $T_j$ : Tiempo de viaje al recorrer una ruta

#### 3.2.2 Parámetros

1. Cantidad plantas de cada especie a llevar por viaje, por lo menos 524 por cada hectárea.
2. Orden de plantación en el polígono.
3. Cantidad de plantas de cada especie a llevar por viaje
4. Número de hectáreas por polígono.
5. Número de trabajadores
6. Jornada de 8 horas.
7. 30 minutos en vaciar la camioneta
8. 30 minutos en llenar la camioneta
9. Especies de plantas (12 tipos)



10. Número de plantas por hectárea
11. Porcentaje esperado por hectárea
12. Número de plantas esperado por las 75 hectáreas
13. Volumen de camioneta
14. Volumen de plantas
15. Distancia entre las plantas de 4 metros.
16. Una camioneta disponible
17. Cargas completas
18. Distancias euclidianas

### **3.3 Función objetivo a optimizar y restricciones. (solo menciónenlas y den una breve descripción)**

#### **3.3.1 Función Objetivo**

Minimizar el número de rutas

#### **3.3.2 Restricciones**

El objetivo del reto es la minimización de la cantidad de rutas, cumpliendo la proporción y cantidad de plantas en los polígonos. Tomando en cuenta que solo hay una camioneta disponible y en ella caben hasta 524 plantas por carga, se tiene que cargar completamente para no tener penalizaciones, esto es fundamental si consideramos que el terreno a plantar no es regular.

En cuanto a la forma de colocar a las plantas en el terreno se debe de dejar como mínimo 4 metros de distancia entre cada una de ellas. Considerando que existen 12 tipos de plantas en un ciclo específico.

Como consideraciones finales, el tiempo de carga y descarga del camión es de 30 minutos cada proceso, esto es importante a la hora de respetar las jornadas de trabajo que no deben de exceder las 8 horas.

## 4 Modelo Matemático

### 4.1 Conjuntos

- $i$ : Tipo de planta.
- $j$ : Número de polígono.

### 4.2 Variables

- $X_{ij}$ : cantidad de planta tipo  $i$  en polígono  $j$ .
- $Y_{ij}$ : variable binaria ya que es de un polígono  $i$  a otro polígono  $j$  (1 si se usa, 0 si no).
- $T_j$ : tiempo total utilizado para la ruta al polígono  $j$ , incluyendo tiempo de viaje, carga y descarga.
- $R$ : Número de rutas

### 4.3 Parámetros

- $N$ : número total de polígonos.
- $C$ : número exacto de plantas por camioneta por ruta (224 plantas).
- $D_{ij}$ : demanda de la planta tipo  $i$  en el polígono  $j$ .
- $M$ : número grande, usado en las restricciones para forzar la activación de la ruta.
- $T_{\max}$ : Tiempo máximo de trabajo por jornada (480 minutos).
- $T_{\text{carga}}$ : tiempo de carga de la camioneta (30 minutos).
- $T_{\text{descarga}}$ : tiempo de descarga de la camioneta (30 minutos).
- $T_{\text{(viaje)}}(j)$ : tiempo estimado de viaje ida y vuelta al polígono  $j$ .

### 4.4 Función Objetivo

$$\text{Min } (Z) = \sum_{j=1}^N Y_j T_j \quad (1)$$

## 4.5 Restricciones

- Porcentaje de plantas requeridas por polígono:

$$\sum_i X_{ij} \geq D_{ij} \quad \forall j \quad (2)$$

- Capacidad de transporte:

$$\sum_i X_{ij} \leq 524 \quad \forall j \quad (3)$$

- Jornada laboral:

$$T_j = T_{\text{viaje}j} + R \cdot T_{\text{carga}} + R \cdot T_{\text{descarga}} \leq T_{\text{max}} \cdot Y_j \quad \forall j \quad (4)$$

## 5 Propuestas de Soluciones

Para la solución de este reto, se realizaron tres maneras para resolver el planteamiento de rutas.

La primer propuesta se dividió los polígonos entre clústers, ignorando los polígonos ya planteados.

La segunda propuesta, se tomó en cuenta todos los polígonos sin importar que estén planteados o no.

Para éstas dos propuestas, se decidió plantear el problema en Excel para así darnos una idea sencilla de posibles resultados.

Cómo tercer y última propuesta se realizó un algoritmo en Python para dar solución a la matriz completa de la superficie total del problema.

### 5.1 Propuesta 1: Excel Clústers

Para darnos una idea de como iniciar, primero dividimos el mapa de la superficie total reforestada proporcionada en 4 partes, una para el polígono origen y otras tres para los polígonos a visitar.

Se decidieron 4 clústers porque notamos que hay ciertos polígonos que están juntos, y con el objetivo de aprovechar las posibles rutas se decidió dividir el mapa de ésta manera. Véase en la Figura 1.

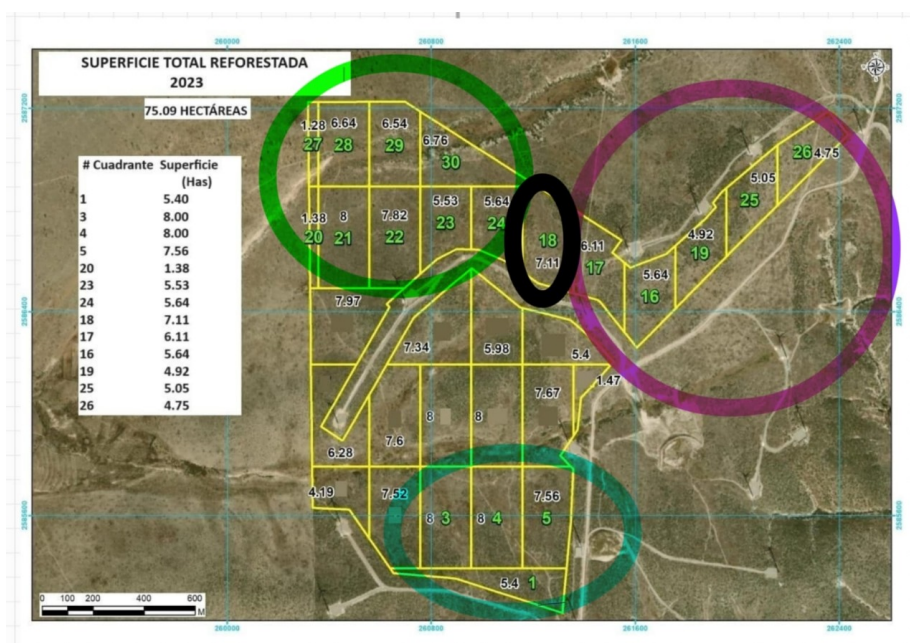


Figure 1: Superficie de hectáreas dividido en clústers

Teniendo un panorama un poco más conocido se decidió realizar un planteamiento de lógica usando Excel. Primero, queríamos hacer una tabla para poder calcular el número de plantas por polígono.

Para esto, se agregó una fila con las 10 especies de plantas y sus porcentajes por hectárea. Además de esto, incorporamos una columna de los 19 polígonos totales a plantar con sus números de hectáreas correspondientes.

Como siguiente paso se multiplicaron los porcentajes por las hectáreas de cada polígono, así obteniendo el número específico y total de plantas por cada polígono.

Nos dimos cuenta que en realidad no son 75 hectáreas totales, ya que la suma de todos los polígonos daban alrededor de 112 hectáreas, esto hizo que cambiaran algunos datos.

Debajo de la tabla también calculamos el número de cada planta que deberían de ir por camión, la suma dando 524 como capacidad máxima de plantas a transportar. Esto se ve en la Figura ??.

Especie		Agave lechuguilla	Agave salmiana	Agave scabra	Agave striata	Opuntia cantabrigiensis	Opuntia engelmanni	Opuntia robusta	Opuntia streptacantha	Prosopis laevigata	Yucca filifera	
% POR HA		6.2977	29.9618	6.2977	6.2977	7.4427	5.7252	11.0687	9.7328	13.1679	4.0076	
Num. Poligonos	Num. Hectáreas	Numero de plantas por poligono										Plantas por Poligono
18.0000	7.1100	235	1116	235	235	277	213	412	363	491	149	3726
1.0000	5.4	178	848	178	178	211	162	313	275	373	113	2830
3.0000	8	264	1256	264	264	312	240	464	408	552	168	4192
4.0000	8	264	1256	264	264	312	240	464	408	552	168	4192
5.0000	7.96	249	1187	249	249	295	227	438	386	522	159	3961
17.0000	6.11	202	959	202	202	238	183	354	312	422	128	3202
16.0000	5.64	186	885	186	186	220	169	327	288	389	118	2955
19.0000	4.92	162	772	162	162	192	148	285	251	339	103	2578
25.0000	5.05	167	793	167	167	197	152	293	258	348	106	2646
26.0000	4.75	157	746	157	157	185	143	275	242	328	100	2489
24.0000	5.64	186	885	186	186	220	169	327	288	389	118	2955
23.0000	5.53	182	868	182	182	216	166	321	282	382	116	2898
22.0000	7.82	258	1228	258	258	305	235	454	399	540	164	4098
21.0000	8	264	1256	264	264	312	240	464	408	552	168	4192
20.0000	1.38	46	217	46	46	54	41	80	70	95	29	723
27.0000	1.28	42	201	42	42	50	38	74	65	88	27	671
28.0000	6.64	219	1042	219	219	259	199	385	339	458	139	3479
29.0000	6.54	216	1027	216	216	255	196	379	334	451	137	3427
30.0000	6.76	223	1061	223	223	264	203	392	345	466	142	3542
HA. Total:	112.1300											58756
Numero de plantas por Camión												Plantas total por camion
33	157	33	33	39	30	58	51	69	21			524

Figure 2: Tabla en Excel de numero de plantas por polígono

Se escaló la imagen proporcionada y se colocó en Geogebra para así obtener las distancias euclidianas entre polígonos. Se aplicó una regla de tres para convertir las distancias en metros a horas, tomando en cuenta una distancia promedio de 60 km/h del camión. Ya que los tiempos serían proporcionales. El cálculo de rutas sería igual sin importar la velocidad del camión. Una vez calculado esto, se elaboró una matriz en Excel con los tiempos en horas que tardaría en trasladar un camión de un polígono a otro. Esto se ve en la Figura 3.

	10	1	3	4	5	17	16	19	25	26	24	23	22	21	20	27	28	29	30
10	0	0.0298	0.01885	0.018015	0.01774	0.00714	0.00783	0.02928	0.01519	0.01701	0.00527	0.00646	0.00967	0.01232	0.01483	0.01	0.01426	0.0104	0.00938
1	0.0298	0	0.00614	0.00428667	0.00435	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
3	0.01885	0.00614	0	0.00323333	0.003	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
4	0.018015	0.00428667	0.00323333	0	0.00307	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
5	0.01774	0.00435	0.003	0.00307	0	0.00305	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
17	0.00714	0.001	0.001	0.001	0.001	0	0.00259	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239
16	0.00783	0.001	0.001	0.001	0.001	0.00259	0	0.00259	0.00259	0.00259	0.00259	0.00259	0.00259	0.00259	0.00259	0.00259	0.00259	0.00259	0.00259
19	0.02928	0.001	0.001	0.001	0.001	0.00239	0.00259	0	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239
25	0.01519	0.001	0.001	0.001	0.001	0.00239	0.00259	0.00239	0	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239
26	0.01701	0.001	0.001	0.001	0.001	0.00239	0.00259	0.00239	0.00239	0	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239
24	0.00527	0.001	0.001	0.001	0.001	0.00239	0.00259	0.00239	0.00239	0.00239	0	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239
23	0.00646	0.001	0.001	0.001	0.001	0.00239	0.00259	0.00239	0.00239	0.00239	0.00239	0	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239
22	0.00967	0.001	0.001	0.001	0.001	0.00239	0.00259	0.00239	0.00239	0.00239	0.00239	0.00239	0	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239
21	0.01232	0.001	0.001	0.001	0.001	0.00239	0.00259	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0	0.00239	0.00239	0.00239	0.00239	0.00239
20	0.01	0.001	0.001	0.001	0.001	0.00239	0.00259	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0	0.00239	0.00239	0.00239	0.00239
27	0.01426	0.001	0.001	0.001	0.001	0.00239	0.00259	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0	0.00239	0.00239	0.00239
28	0.0104	0.001	0.001	0.001	0.001	0.00239	0.00259	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0	0.00239	0.00239
29	0.00938	0.001	0.001	0.001	0.001	0.00239	0.00259	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0	0.00239
30	0.00938	0.001	0.001	0.001	0.001	0.00239	0.00259	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0.00239	0

Figure 3: Matriz de tiempos de rutas

Finalmente, consideramos que un viaje del camión podía cumplir una hectárea. Por lo tanto, hicimos una tabla nueva donde teníamos cada polígono, con sus viajes necesarios para cumplir sus hectáreas correspondientes, y el tiempo en horas que le tomaría al camión llegar desde el polígono de origen al polígono deseado. Ese tiempo se calculo multiplicando la matriz de tiempos de arriba por el numero de viajes requerido por el camión por polígono.

Una vez obtenido todo esto, se sumaron el tiempo de rutas para cada división creada previamente, y al final sumando el tiempo de carga y descarga, dándonos un resultado de 115 horas para realizar toda la ruta completa. Respetando la jornada laboral de 8 horas, el número de días a trabajar es de aproximadamente de 15 días. Esto se ve en la Figura 4.

Num. Poligonos	Num. Viajes	18			
18.00	8.0000	0	Azul	1.223386667	
1.00	6	0.17489333			
3.00	8	0.15077333	Rosa	1.060042667	
4.00	8	0.14412			
5.00	8	0.14190667	Verde	1.567866667	
17.00	7	0.05711867			
16.00	6	0.062608	Tiempo Total Ruta		
19.00	5	0.16870133		115.851296	
25.00	6	0.10549067			
26.00	5	0.13610267			
24.00	6	0.02617333			
23.00	6	0.05169333			
22.00	8	0.07737333			
21.00	8	0.10332			
20.00	2	0.11861333			
27.00	2	0.12802667			
28.00	7	0.11406667			
29.00	7	0.09122667			
30.00	7	0.07344			

Figure 4: Resultados obtenidos

## 5.2 Propuesta 2: Excel Completo

Para esta propuesta de todas los polígonos conectados entre sí, se hizo exactamente lo mismo como se explicó en la propuesta 1 pero ahora con todos los polígonos del mapa. Figura 5

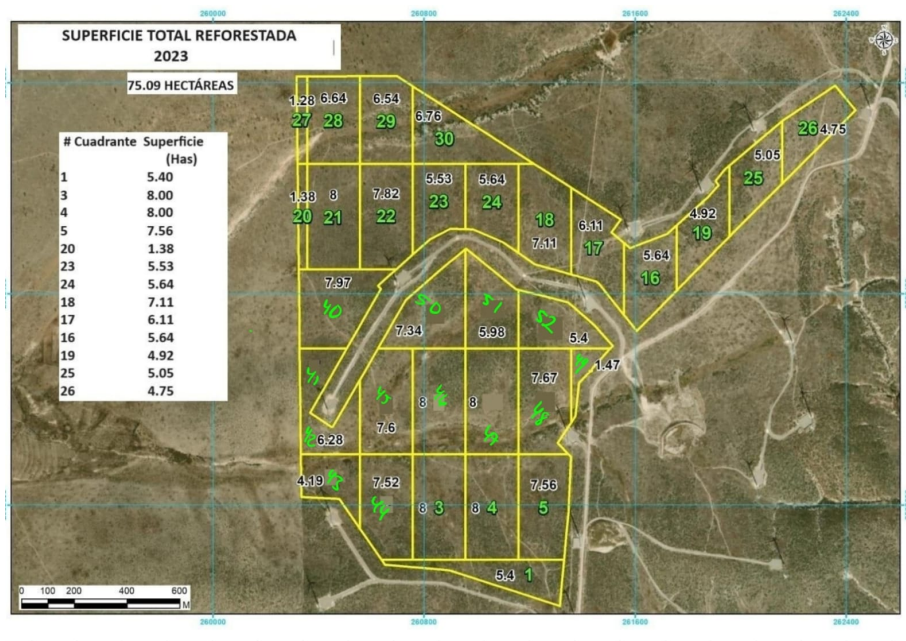


Figure 5: Nuevo mapa con todos los polígonos



Realizamos los mismos cálculos en Excel pero con la nueva matriz. Figura 6

Num. Polígonos	Num. Hectáreas	Numero de plantas por polígono										Plantas por Polígono	
18.0000	7.1100	235	1116	235	235	277	213	412	363	491	149	3726	
1.0000	5.4	178	848	178	178	211	162	313	275	373	113	2830	
3.0000	8	264	1256	264	264	312	240	464	408	552	168	4192	
4.0000	8	264	1256	264	264	312	240	464	408	552	168	4192	
5.0000	7.56	249	1187	249	249	295	227	438	386	522	159	3961	
17.0000	6.11	202	959	202	202	238	183	354	312	422	128	3202	
16.0000	5.64	186	885	186	186	220	169	327	288	389	118	2955	
19.0000	4.92	162	772	162	162	192	148	285	251	339	103	2578	
25.0000	5.05	167	793	167	167	197	152	293	258	348	106	2646	
26.0000	4.75	157	746	157	157	185	143	275	242	328	100	2489	
24.0000	5.64	186	885	186	186	220	169	327	288	389	118	2955	
23.0000	5.53	182	868	182	182	216	166	321	282	382	116	2898	
22.0000	7.82	258	1228	258	258	305	235	454	399	540	164	4098	
21.0000	8	264	1256	264	264	312	240	464	408	552	168	4192	
20.0000	1.38	46	217	46	46	54	41	80	70	95	29	723	
27.0000	1.28	42	201	42	42	50	38	74	65	88	27	671	
28.0000	6.64	219	1042	219	219	259	199	385	339	458	139	3479	
29.0000	6.54	216	1027	216	216	255	196	379	334	451	137	3427	
30.0000	6.76	223	1061	223	223	264	203	392	345	466	142	3542	
40.0000	7.97	263	1251	263	263	311	239	462	406	550	167	4176	
41.0000	2.51	83	394	83	83	98	75	146	128	173	53	1315	
42.0000	3.77	124	592	124	124	147	113	219	192	260	79	1975	
43.0000	4.19	138	658	138	138	163	126	243	214	289	88	2196	
44.0000	7.52	248	1181	248	248	293	226	436	384	519	158	3940	
45.0000	7.6	251	1193	251	251	296	228	441	388	524	160	3982	
46.0000	8	264	1256	264	264	312	240	464	408	552	168	4192	
47.0000	8	264	1256	264	264	312	240	464	408	552	168	4192	
48.0000	7.67	253	1204	253	253	299	230	445	391	529	161	4019	
49.0000	1.47	49	231	49	49	57	44	85	75	101	31	770	
50.0000	7.34	242	1152	242	242	286	220	426	374	506	154	3846	
51.0000	5.98	197	939	197	197	233	179	347	305	413	126	3134	
52.0000	5.4	178	848	178	178	211	162	313	275	373	113	2830	
HA. Total:	189.5500											Plantas total	99324

Figure 6: Resultados arrojados por el algoritmo

Con esta nueva matriz, obtuvimos de resultado un total de 190 hectáreas y 99,324 plantas totales a plantar. Esta diferencia a la primera propuesta se debe al incremento de polígonos que se tomaron en cuenta.

Aquí multiplicamos el tiempo que se tarda de ir del nodo origen a todos los polígonos totales. Sumamos esta cantidad, sumándole 180 horas que tardarían en cargar y descargar, y nos da un total de 194 horas para la ruta de toda la matriz completa, osea entre 8 a 9 días de trabajo.

Figura 7

Num. Polígonos	Num. Hectáreas	Horas por Hectáreas	Horas por Polígono
18.0000	7.1100	0	0
1.0000	5.4	0.025544	0.1379376
3.0000	8	0.0188467	0.1507733
4.0000	8	0.018015	0.14412
5.0000	7.56	0.0177383	0.1341018
17.0000	6.11	0.003856	0.0235602
16.0000	5.64	0.007826	0.0441386
19.0000	4.92	0.0097692	0.0480643
25.0000	5.05	0.0131863	0.066591
26.0000	4.75	0.0170128	0.080811
24.0000	5.64	0.0032717	0.0184522
23.0000	5.53	0.0064617	0.035733
22.0000	7.82	0.0096717	0.0756324
21.0000	8	0.012915	0.10332
20.0000	1.38	0.0148267	0.0204608
27.0000	1.28	0.0160033	0.0204843
28.0000	6.64	0.0142583	0.0946753
29.0000	6.54	0.0114033	0.0745778
30.0000	6.76	0.00918	0.0620568
40.0000	7.97	0.0139967	0.1115534
41.0000	2.51	0.01689	0.0423939
42.0000	3.77	0.01888	0.0711776
43.0000	4.19	0.02043	0.0856017
44.0000	7.52	0.019935	0.1499112
45.0000	7.6	0.01509	0.114684
46.0000	8	0.0131483	0.1051867
47.0000	8	0.01193	0.09544
48.0000	7.67	0.0113233	0.08685
49.0000	1.47	0.0091183	0.013404
50.0000	7.34	0.0092833	0.0681397
51.0000	5.98	0.0065633	0.0392487
52.0000	5.4	0.0065467	0.035352
<b>Tiempo Total Ruta</b>			<b>194.63816</b>

Figure 7: Resultados arrojados por el algoritmo

### 5.3 Propuesta 3: Código

El algoritmo diseñado encontró las rutas que recorrería el camión tomando en cuenta la necesidad de plantas por polígono, la oferta que tendría el camión por viaje y el tiempo recorrido. A continuación, se verán los pasos tomados para obtener un resultado aproximado usando Python:

1. Se calculó el número de plantas por polígono.

```
[ ] # Definición de las especies y sus características
    especies = {
        'Agave lechuguilla': {'plantas_por_HA': 33},
        'Agave salmiana': {'plantas_por_HA': 157},
        'Agave scabra': {'plantas_por_HA': 33},
        'Agave striata': {'plantas_por_HA': 33},
        'Opuntia cantabrigiensis': {'plantas_por_HA': 39},
        'Opuntia engelmanni': {'plantas_por_HA': 30},
        'Opuntia robusta': {'plantas_por_HA': 58},
        'Opuntia streptacantha': {'plantas_por_HA': 51},
        'Prosopis laevigata': {'plantas_por_HA': 69},
        'Yucca filifera': {'plantas_por_HA': 21}
    }

    # Hectáreas ordenadas por número de polígono
    hectares_polygons = {
        '18': 7.11, '1': 5.4, '3': 8, '4': 8, '5': 7.56, '17': 4.001, '16': 5.64, '19': 4.6547, '25': 5.05, '26': 4.75,
        '17.2': 2.1001, '19.2': 0.2653, '24': 5.64, '23': 5.53, '22': 7.82,
        '21': 8, '20': 1.38,
        '27': 1.28, '28': 6.64, '29': 6.54, '30': 6.76
    }

    # Función para calcular el número de plantas por polígono para una especie
    def calcular_plantas_por_especie(especie, hectares_polygons):
        plantas_por_ha = especie['plantas_por_HA']
        plantas_por_poligono = {poligono: plantas_por_ha * hectares for poligono, hectares in hectares_polygons.items()}
        return plantas_por_poligono
```

Figure 8: Código: Número de plantas por polígono

2. Imprimimos el número de plantas por polígono.

```
[ ] # 'Agave lechuguilla'
resultados_agave_lechuguilla = calcular_plantas_por_especie(especies['Agave lechuguilla'], hectares_polygons)

# Mostrar resultados
for poligono, cantidad in resultados_agave_lechuguilla.items():
    print(f'Poligono {poligono}: {cantidad:.2f} plantas de Agave lechuguilla')

Poligono 18: 234.63 plantas de Agave lechuguilla
Poligono 1: 178.28 plantas de Agave lechuguilla
Poligono 3: 264.00 plantas de Agave lechuguilla
Poligono 4: 264.00 plantas de Agave lechuguilla
Poligono 5: 249.48 plantas de Agave lechuguilla
Poligono 17: 132.03 plantas de Agave lechuguilla
Poligono 16: 186.12 plantas de Agave lechuguilla
Poligono 19: 153.61 plantas de Agave lechuguilla
Poligono 25: 166.65 plantas de Agave lechuguilla
Poligono 26: 156.75 plantas de Agave lechuguilla
Poligono 17.2: 69.30 plantas de Agave lechuguilla
Poligono 19.2: 8.75 plantas de Agave lechuguilla
Poligono 24: 186.12 plantas de Agave lechuguilla
Poligono 23: 182.49 plantas de Agave lechuguilla
Poligono 22: 258.06 plantas de Agave lechuguilla
Poligono 21: 264.00 plantas de Agave lechuguilla
Poligono 28: 45.54 plantas de Agave lechuguilla
Poligono 27: 42.24 plantas de Agave lechuguilla
Poligono 28: 219.12 plantas de Agave lechuguilla
Poligono 29: 215.82 plantas de Agave lechuguilla
Poligono 30: 223.08 plantas de Agave lechuguilla
```

Figure 9: Resultado: Número de plantas por polígono

3. Con ayuda de Geogebra se calcularon las distancias euclidianas y se plasmaron en una matriz en Python.

```
[ ] tiempos_df.head(21)

   0      1      2      3      4      5      6      7      8      9      ...    11    12    13    14    15    16    17    18    19    20
0  inf  0.030174  0.030357  0.027128  0.024095  0.003856  0.007826  0.000709  0.013186  0.017013  ...  0.011319  0.003272  0.006492  0.009672  0.012915  0.014827  0.016003  0.014258  0.011403  0.006180
1  0.030174  inf  0.005140  0.004282  0.004388  inf  inf  inf  inf  inf  ...  inf  inf  inf  inf  inf  inf  inf  inf  inf  inf
2  0.030357  0.005140  inf  0.003228  0.006295  inf  inf  inf  inf  inf  ...  inf  inf  inf  inf  inf  inf  inf  inf  inf  inf
3  0.027128  0.004282  0.003228  inf  0.003095  inf  inf  inf  inf  inf  ...  inf  inf  inf  inf  inf  inf  inf  inf  inf  inf
4  0.024095  0.004388  0.006295  0.003095  inf  inf  inf  inf  inf  inf  ...  inf  inf  inf  inf  inf  inf  inf  inf  inf  inf
5  0.003856  inf  inf  inf  inf  inf  0.003096  0.006037  0.010962  0.014802  ...  0.009027  inf  inf  inf  inf  inf  inf  inf  inf  inf
6  0.007826  inf  inf  inf  inf  0.003096  inf  0.004225  0.003013  0.012852  ...  0.007538  inf  inf  inf  inf  inf  inf  inf  inf  inf
7  0.000709  inf  inf  inf  inf  0.006037  0.004225  inf  0.004393  0.008042  ...  0.003440  inf  inf  inf  inf  inf  inf  inf  inf  inf
8  0.013186  inf  inf  inf  inf  0.010962  0.008613  0.004393  inf  0.004248  ...  0.001962  inf  inf  inf  inf  inf  inf  inf  inf  inf
9  0.017013  inf  inf  inf  inf  0.014802  0.012852  0.008042  0.004248  inf  ...  0.005787  inf  inf  inf  inf  inf  inf  inf  inf  inf
10 0.003284  inf  inf  inf  inf  0.002211  0.005301  0.006517  0.009938  0.013858  ...  0.008108  inf  inf  inf  inf  inf  inf  inf  inf  inf
11 0.011319  inf  inf  inf  inf  0.009027  0.007538  0.003440  0.001962  0.005787  ...  inf  inf  inf  inf  inf  inf  inf  inf  inf  inf
12 0.003272  inf  inf  inf  inf  inf  inf  inf  inf  inf  ...  inf  inf  0.003223  0.006478  0.009715  0.011825  0.012810  0.011115  0.008432  0.004958
13 0.006492  inf  inf  inf  inf  inf  inf  inf  inf  inf  ...  inf  0.003223  inf  0.003268  0.006493  0.008403  0.010003  0.008477  0.006352  0.004410
14 0.009672  inf  inf  inf  inf  inf  inf  inf  inf  inf  ...  inf  0.006478  0.003268  inf  0.003242  0.006154  0.007840  0.006756  0.009923  0.008247
15 0.012915  inf  inf  inf  inf  inf  inf  inf  inf  inf  ...  inf  0.009715  0.008493  0.003242  inf  0.001912  0.006207  0.005923  0.006750  0.008585
16 0.014827  inf  inf  inf  inf  inf  inf  inf  inf  inf  ...  inf  0.011825  0.008403  0.005154  0.001912  inf  0.005915  0.006233  0.007857  0.010335
17 0.016003  inf  inf  inf  inf  inf  inf  inf  inf  inf  ...  inf  0.012810  0.010003  0.007840  0.006207  0.005915  inf  0.001912  0.005153  0.006213
18 0.014258  inf  inf  inf  inf  inf  inf  inf  inf  inf  ...  inf  0.011115  0.008477  0.006756  0.006923  0.006233  0.001912  inf  0.003242  0.007323
19 0.011403  inf  inf  inf  inf  inf  inf  inf  inf  inf  ...  inf  0.008432  0.006352  0.005923  0.006750  0.007857  0.005153  0.003242  inf  0.004148
20 0.006180  inf  inf  inf  inf  inf  inf  inf  inf  inf  ...  inf  0.004958  0.004410  0.006247  0.008585  0.010335  0.006213  0.007323  0.004148

21 rows x 21 columns
```

Figure 10: Matriz de tiempo

4. Se diseñó el grafo que representa las conexiones entre todos los polígonos.

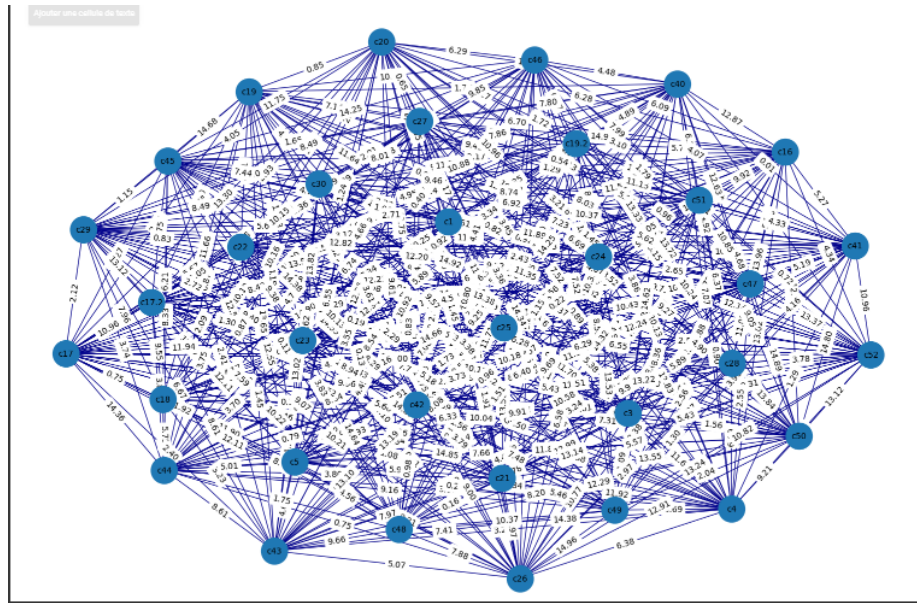


Figure 11: Grafo que representa las conexiones entre polígonos

Aquí se puede observar la distribución de los polígonos en los 3 clusters, con el cuarto siendo el polígono 18 de origen.

5. Se imprimieron las demandas de los polígonos. Figura 12

```
Aqui tenemos en una lista la cantidad de plantas de cada especie que debe de tener cada polígono.

[ ] import math
# Demanda
lista_resultados_agave_lechuguilla = list(resultados_agave_lechuguilla.values())
lista_resultados_agave_salsiana = list(resultados_agave_salsiana.values())
lista_resultados_agave_scabra = list(resultados_agave_scabra.values())
lista_resultados_agave_striata = list(resultados_agave_striata.values())
lista_resultados_Opuntia_cantabrigiensis = list(resultados_Opuntia_cantabrigiensis.values())
lista_resultados_Opuntia_engelmanni = list(resultados_Opuntia_engelmanni.values())
lista_resultados_Opuntia_roburta = list(resultados_Opuntia_roburta.values())
lista_resultados_Opuntia_straptacanta = list(resultados_Opuntia_straptacanta.values())
lista_resultados_Prosopis_laevigata = list(resultados_Prosopis_laevigata.values())
lista_resultados_Yucca_filifera = list(resultados_Yucca_filifera.values())

lista_total_plantas = [
    lista_resultados_agave_lechuguilla,
    lista_resultados_agave_salsiana,
    lista_resultados_agave_scabra,
    lista_resultados_agave_striata,
    lista_resultados_Opuntia_cantabrigiensis,
    lista_resultados_Opuntia_engelmanni,
    lista_resultados_Opuntia_roburta,
    lista_resultados_Opuntia_straptacanta,
    lista_resultados_Prosopis_laevigata,
    lista_resultados_Yucca_filifera
]

poligonos = {
    'c18':[], 'c1':[], 'c3':[],
    'c4':[], 'c5':[], 'c17':[],
    'c16':[], 'c19':[], 'c25':[],
    'c26':[], 'c17.2':[], 'c19.2':[],
    'c24':[], 'c23':[], 'c22':[], 'c21':[],
    'c20':[], 'c27':[], 'c28':[], 'c29':[], 'c30':[]
}

counter = 0
for key,value in poligonos.items():
    for i in lista_total_plantas:
        value.append(math.ceil(i[counter]))
        counter+=1
poligonos

{'c18': [235, 1117, 235, 235, 278, 214, 413, 363, 491, 150],
 'c1': [179, 848, 179, 179, 211, 162, 314, 276, 373, 114],
 'c3': [264, 1256, 264, 264, 312, 240, 464, 408, 552, 168],
 'c4': [264, 1256, 264, 264, 312, 240, 464, 408, 552, 168],
 'c5': [250, 1187, 250, 250, 295, 227, 439, 386, 522, 159],
 'c17': [133, 629, 133, 133, 157, 121, 233, 205, 277, 85],
 'c16': [187, 886, 187, 187, 220, 170, 328, 288, 390, 119],
 'c19': [154, 731, 154, 154, 182, 140, 270, 238, 322, 98],
 'c25': [167, 793, 167, 167, 197, 152, 293, 258, 340, 107],
 'c26': [157, 746, 157, 157, 186, 143, 276, 243, 328, 100],
 'c17.2': [70, 330, 70, 70, 82, 64, 122, 108, 145, 45],
 'c19.2': [9, 42, 9, 9, 11, 8, 16, 14, 19, 6],
 'c24': [187, 886, 187, 187, 220, 170, 328, 288, 390, 119],
 'c23': [183, 869, 183, 183, 216, 166, 321, 283, 382, 117],
 'c22': [259, 1228, 259, 259, 305, 235, 454, 399, 540, 165],
 'c21': [264, 1256, 264, 264, 312, 240, 464, 408, 552, 168],
 'c20': [46, 217, 46, 46, 54, 42, 81, 71, 96, 29],
 'c27': [43, 201, 43, 43, 50, 39, 75, 66, 89, 27],
 'c28': [220, 1043, 220, 220, 259, 200, 386, 339, 459, 140],
 'c29': [216, 1027, 216, 216, 256, 197, 380, 334, 452, 138],
 'c30': [224, 1062, 224, 224, 264, 203, 393, 345, 467, 142]}
```

Figure 12: Demandas de cada polígono

## 6. Funcionamiento del algoritmo 16

- Se buscó el polígono más cerca, se descarga el cargamento de plantas y si la demanda todavía no es satisfecha el camión se regresa a 18.
- Cuando satisface la demanda del polígono, si todavía tiene carga va al polígono más cercano, sino se regresa al 18.
- Este proceso se repite hasta que las demandas de todos los polígonos son satisfechas

Como mencionamos anteriormente, las matriz fue creadas en Excel y al agregar esta matriz al código hará lo siguiente:

- El nombre de nuestra función es:

`find_best_route`

- Agregamos una función donde se encuentren las restricciones mínimas para cada tipo de planta.
- Inicializamos con una carga vacía, buscamos la ruta más cercana cargamos un conjunto de polígonos donde vienen todos los nodos, este está separado por un archivo CSV del mismo tamaño de columnas (simplemente por comodidad y mayor practicidad al trabajar).
- Entramos en un bucle donde se empiezan a visitar poco a poco los nodos.
  - En caso de que la demanda de uno de los nodos se cumple y quedan oferta, estas se plantan en el nodo 18.
- Luego tenemos otro bucle donde volvemos a cargar los camiones cuando estos son vaciados y vuelve a empezar desde cero buscando el siguiente nodo más cercano.

Tomemos en cuenta que cada iteración que se hace, el código va guardando registro sobre la posición actual, posición a la que se dirige, la demanda que necesita cubrir y su tiempo de desplazo. Todos estos datos son calculados en base a los datos del reporte original del socio formador, el cual nos ha brindado las distancias y el número de plantas esperado por hectáreas. Al terminar el código, nos suelta los siguientes resultados.

```
{
  "action": "load",
  "time": 30,
  "current_node": "18.0",
  "next_node": null,
  "demand": 3725.64
}, {
  "action": "load",
  "time": 30,
  "current_node": "18.0",
  "next_node": null,
  "demand": 3725.64
}, {
  "action": "travel",
  "time": 0.19630000000000003,
  "current_node": "18.0",
  "next_node": "24.0",
  "demand": 3725.64
}, {
  "action": "unload",
  "time": 30,
  "current_node": "18.0",
  "next_node": "24.0",
  "demand": 3725.64
}, {
  "action": "travel",
  "time": 0.19630000000000003,
  "current_node": "24.0",
  "next_node": "18.0",
  "demand": 2955.36
}, {
  "action": "load",
  "time": 30,
  "current_node": "18.0",
  "next_node": null,
  "demand": 3725.64
}, {
  "action": "travel",
  "time": 0.19630000000000003,
  "current_node": "18.0",
  "next_node": "24.0",
  "demand": 3725.64
}, {
  "action": "travel",
  "time": 0.19630000000000003,
  "current_node": "18.0",
  "next_node": "24.0",
  "demand": 3725.64
}, {
  "action": "travel",
  "time": 0.19630000000000003,
  "current_node": "24.0",
  "next_node": "18.0",
  "demand": 2955.36
}, {
  "action": "load",
  "time": 30,
  "current_node": "18.0",
  "next_node": null,
  "demand": 3725.64
}, {
  "action": "travel",
  "time": 0.19630000000000003,
  "current_node": "18.0",
  "next_node": "24.0",
  "demand": 3725.64
}, {
  "action": "unload",
  "time": 30,
  "current_node": "18.0",
  "next_node": "24.0",
  "demand": 3725.64
}, {
  "action": "travel",
  "time": 0.19630000000000003,
  "current_node": "24.0",
  "next_node": "18.0",
  "demand": 2955.36
}, {
  "action": "load",
  "time": 30,
  "current_node": "18.0",
  "next_node": null,
  "demand": 3725.64
}, {
  "action": "travel",
  "time": 0.19630000000000003,
  "current_node": "18.0",
  "next_node": "24.0",
  "demand": 3725.64
}
```

Figure 13: Salida del código al correr la matriz completa.

Podemos observar que el código nos da los datos que hemos mencionado anteriormente, incluso, al terminar todos los ciclos como viene descrito en el grafo, podemos imprimir: el número de viajes, horas que tardaría y el número de cargas que hizo.

```
Total time: 206.36423349999996 hours
```

Figure 14: Resultado horas totales con la matriz completa

```
Total trips: 398
```

Figure 15: Numero de viajes realizados con la matriz completa

```
Total loads: 186
```

Figure 16: Numero de cargas hechas con la matriz completa



## 6 Experimentos

### 6.1 Características de la PC y software

Modelo de la PC: Custom Build

- Sistema Operativo: Windows 11
- Capacidad de disco duro: 1TB de almacenamiento, 250 GB de SSD para Sistema operativo
- Memoria Ram y capacidad: 16 GB DDR4 2400 MHz
- Tipo de procesador: Intel i7-8700 (3.30 Ghz)
- Número de núcleos: 6 núcleos - 12 hilos
- Software utilizado y versión: Python 3.11

### 6.2 Tamaño del problema

- Número de nodos:  
El número de polígonos es de 35 más 3 nodos que se colocaron para conveniencia de la resolución
- Número de arcos: 625 debido, esto puede comprobarse a través de la tabla de Excel hecha que tiene una dimensión de 35 x 35.
- Número de variables consideradas:
  - $Y_{ij}$ : variable binaria ya que el viaje es de un polígono  $i$  a otro polígono  $j$  (1 si se usa, 0 si no).
  - $T_j$ : tiempo total utilizado para la ruta al polígono  $j$ , incluyendo tiempo de viaje, carga y descarga.
- Número de parámetros:
  - Cantidad plantas de cada especie a llevar por viaje, 524 por cada hectárea, calculada por los porcentajes ya establecidos.
  - Número de hectáreas por polígono.
  - Jornada de 8 horas.
  - 30 minutos en vaciar la camioneta
  - 30 minutos en llenar la camioneta
  - Especies de plantas (12 tipos)
  - Una camioneta disponible
  - Distancias euclidianas

### 6.3 Datos

Para esta tercera parte del reto, se nos pidió realizar experimentos con nuevas matrices. Un total de 5 matrices chicas (15x15), 5 matrices medianas (20x20), y 5 matrices grandes (25x25), todas aleatorias. Este proceso se realizó en Excel y se agregaron al Colab donde realizamos los nuevos algoritmos con las nuevas matrices.

A continuación se mostraran una matriz de cada tamaño:

1		18.0	1.0	4.0	17.0	19.0	26.0	19.2	23.0	21.0	27.0	29.0	40.0	42.0	44.0	46.0
2	18.0	0	15.33	10.81	2.31	5.86	10.21	6.79	3.88	7.75	9.6	6.84	8.4	11.33	11.96	7.89
3	1.0	15.33	0	2.57	12.24	14.49	19.45	16.54	13.69	14.72	18.41	17.34	11.88	8.43	5.34	6.8
4	4.0	10.81	2.57	0	10.18	12.89	18	14.89	11.15	12.18	15.86	14.77	9.43	6.51	3.85	4.25
5	17.0	2.31	12.24	10.18	0	3.98	8.88	5.42	6.03	9.8	11.89	9.15	9.92	12.04	12.06	8.13
6	19.0	5.86	14.49	12.89	3.98	0	5.19	2.06	9.74	13.61	15.27	12.32	13.9	15.8	15.41	11.71
7	26.0	10.21	19.45	18	8.88	5.19	0	3.47	13.87	17.74	18.64	15.55	18.59	20.89	20.59	16.86
8	19.2	6.79	16.54	14.89	5.42	2.06	3.47	0	10.54	14.44	15.64	12.59	15.14	17.45	17.27	13.46
9	23.0	3.88	13.69	11.15	6.03	9.74	13.87	10.54	0	3.9	6	3.81	5.28	9.36	10.98	7.21
10	21.0	7.75	14.72	12.18	9.8	13.61	17.74	14.44	3.9	0	3.72	4.05	3.28	8.16	10.72	7.93
11	27	9.6	18.41	15.86	11.89	15.27	18.64	15.64	6	3.72	0	3.09	6.92	11.75	14.42	11.61
12	29.0	6.84	17.34	14.77	9.15	12.32	15.55	12.59	3.81	4.05	3.09	0	7.11	11.91	14.09	10.66
13	40.0	8.4	11.88	9.43	9.92	13.9	18.59	15.14	5.28	3.28	6.92	7.11	0	4.88	7.52	5.32
14	42.0	11.33	8.43	6.51	12.04	15.8	20.89	17.45	9.36	8.16	11.75	11.91	4.88	0	3.2	4.22
15	44.0	11.96	5.34	3.85	12.06	15.41	20.59	17.27	10.98	10.72	14.42	14.09	7.52	3.2	0	4.07
16	46.0	7.89	6.8	4.25	8.13	11.71	16.86	13.46	7.21	7.93	11.61	10.66	5.32	4.22	4.07	0

Figure 17: Ejemplo de una matriz chica (15x15) aleatoria.

1		18.00	17.00	16.00	19.00	25.00	26.00	17.20	19.20	24.00	23.00	22.00	21.00	20.00	27.00	28.00	29.00	30.00	40.00	41.00	42.00
2	18.00	0.00	2.31	4.70	5.86	7.91	10.21	1.97	6.79	1.96	3.88	5.80	7.75	8.90	9.60	8.55	6.84	5.51	8.40	10.13	11.33
3	17.00	2.31	0.00	2.39	3.98	6.40	8.88	1.33	5.42	4.20	6.03	7.87	9.80	10.93	11.89	10.86	9.15	6.80	9.92	11.28	12.04
4	16.00	4.70	2.39	0.00	2.54	5.17	7.71	3.18	4.52	6.59	8.40	10.20	12.10	13.23	14.28	13.25	11.53	9.15	11.93	13.01	13.41
5	19.00	5.86	3.98	2.54	0.00	2.64	5.19	3.91	2.06	7.81	9.74	11.66	13.61	14.75	15.27	14.17	12.32	9.83	13.90	15.21	15.80
6	25.00	7.91	6.40	5.17	2.64	0.00	2.55	5.96	1.18	9.77	11.69	13.65	15.59	16.73	16.82	15.69	13.76	11.29	16.22	17.68	18.37
7	26.00	10.21	8.88	7.71	5.19	2.55	0.00	8.31	3.47	11.97	13.87	15.83	17.74	18.87	18.64	17.49	15.55	13.14	18.59	20.14	20.89
8	17.20	1.97	1.33	3.18	3.91	5.96	8.31	0.00	4.86	3.90	5.84	7.77	9.72	10.86	11.41	10.34	8.53	6.09	10.28	11.87	12.85
9	19.20	6.79	5.42	4.52	2.06	1.18	3.47	4.86	0.00	8.62	10.54	12.50	14.44	15.58	15.64	14.51	12.59	10.12	15.14	16.66	17.45
10	24.00	1.96	4.20	6.59	7.81	9.77	11.97	3.90	8.62	0.00	1.93	3.89	5.83	6.98	7.69	6.67	5.06	2.97	6.84	8.85	10.37
11	23.00	3.88	6.03	8.40	9.74	11.69	13.87	5.84	10.54	1.93	0.00	1.96	3.90	5.04	6.00	5.09	3.81	2.65	5.28	7.53	9.36
12	22.00	5.80	7.87	10.20	11.66	13.65	15.83	7.77	12.50	3.89	1.96	0.00	1.95	3.09	4.70	4.05	3.55	3.75	3.82	6.28	8.43
13	21.00	7.75	9.80	12.10	13.61	15.59	17.74	9.72	14.44	5.83	3.90	1.95	0.00	1.15	3.72	3.55	4.05	5.21	3.28	5.75	8.16
14	20.00	8.90	10.93	13.23	14.75	16.73	18.87	10.86	15.58	6.98	5.04	3.09	1.15	0.00	3.55	3.74	4.71	6.20	3.46	5.72	8.21
15	27.00	9.60	11.89	14.28	15.27	16.82	18.64	11.41	15.64	7.69	6.00	4.70	3.72	3.55	0.00	1.15	3.09	5.53	6.92	9.26	11.75
16	28.00	8.55	10.86	13.25	14.17	15.69	17.49	10.34	14.51	6.67	5.09	4.05	3.55	3.74	1.15	0.00	1.95	4.39	6.84	9.29	11.72
17	29.00	6.84	9.15	11.53	12.32	13.76	15.55	8.53	12.59	5.06	3.81	3.55	4.05	4.71	3.09	1.95	0.00	2.49	7.11	9.62	11.91
18	30.00	5.51	6.80	9.15	9.83	11.29	13.14	6.09	10.12	2.97	2.65	3.75	5.21	6.20	5.53	4.39	2.49	0.00	7.55	9.95	11.93
19	40.00	8.40	9.92	11.93	13.90	16.22	18.59	10.28	15.14	6.84	5.28	3.82	3.28	3.46	6.92	6.84	7.11	7.55	0.00	2.51	4.88
20	41.00	10.13	11.28	13.01	15.21	17.68	20.14	11.87	16.66	8.85	7.53	6.28	5.75	5.72	9.26	9.29	9.62	9.95	2.51	0.00	2.51
21	42.00	11.33	12.04	13.41	15.80	18.37	20.89	12.85	17.45	10.37	9.36	8.43	8.16	8.21	11.75	11.72	11.91	11.93	4.88	2.51	0.00

Figure 18: Ejemplo de una matriz mediana (20x20) aleatoria.

		18.00	3.00	4.80	17.00	19.00	25.00	24.00	22.00	21.00	20.00	27.00	28.00	30.00	40.00	41.00	42.00	43.00	44.00	45.00	46.00	48.00	49.00	50.00	51.00	52.00	
1	1	18.00	0.00	11.31	10.81	2.31	5.86	7.91	1.96	5.80	7.75	6.90	9.60	8.55	5.51	6.40	10.13	11.33	12.26	11.96	9.05	7.89	6.79	5.47	5.57	3.94	3.93
2	2	3.00	11.31	0.00	1.94	11.08	14.13	16.77	11.17	10.86	11.37	11.81	15.10	14.76	13.58	8.57	6.75	4.78	4.12	1.91	4.19	3.76	5.48	7.54	7.04	7.46	7.96
3	3	4.80	10.81	1.94	0.00	10.18	12.89	15.49	11.01	11.38	12.18	12.76	15.86	15.49	13.67	9.43	8.19	6.51	6.01	3.85	5.39	4.25	4.32	6.33	7.42	7.21	7.12
4	4	17.00	2.31	11.08	10.18	0.00	3.98	6.40	4.20	7.87	9.80	10.93	11.89	10.86	6.80	9.92	11.28	12.04	12.77	12.06	9.66	8.13	5.89	4.09	6.65	4.48	3.10
5	5	19.00	5.86	14.13	12.89	3.98	0.00	2.64	7.81	11.66	13.61	14.75	15.27	14.17	9.83	13.90	15.21	15.80	16.41	15.41	13.40	11.71	8.67	6.60	10.61	8.37	6.58
6	6	25.00	7.91	16.77	15.49	6.40	2.64	0.00	9.77	13.65	15.59	16.73	16.82	15.69	11.29	16.22	17.68	18.37	19.02	18.04	15.98	14.32	11.30	9.23	13.94	10.87	9.11
7	7	24.00	1.96	11.17	11.01	4.20	7.81	9.77	0.00	3.89	5.83	6.96	7.69	6.67	2.97	6.94	8.85	10.37	11.44	11.51	8.27	7.49	7.43	6.57	4.60	3.79	4.87
8	8	22.00	5.80	10.86	11.38	7.87	11.66	13.65	3.89	0.00	1.95	3.89	4.70	4.05	3.75	3.82	6.28	8.43	9.72	10.54	6.99	7.19	9.01	9.00	3.97	5.20	7.27
9	9	21.00	7.75	11.37	12.18	9.80	13.61	15.59	5.83	1.95	0.00	1.15	3.72	3.55	5.21	3.28	5.75	8.16	9.51	10.72	7.24	7.83	10.38	10.63	5.06	6.77	9.07
10	10	20.00	6.90	11.81	12.76	10.93	14.75	16.73	6.96	3.89	1.15	0.00	3.55	3.74	6.20	3.46	5.72	8.21	9.56	10.90	7.62	8.54	11.26	11.63	5.90	7.77	10.11
11	11	27.00	9.60	15.10	15.86	11.89	15.27	16.82	7.69	4.70	3.72	3.55	0.00	1.15	5.53	6.92	9.26	11.75	13.09	14.42	10.96	11.61	13.71	13.63	8.57	9.89	11.96
12	12	28.00	8.55	14.76	15.40	10.86	14.17	15.69	6.67	4.05	3.55	3.74	1.15	0.00	4.39	6.94	9.29	11.72	13.06	14.23	10.71	11.17	11.01	12.79	8.01	9.13	11.10
13	13	30.00	5.51	13.58	13.67	6.80	9.83	11.29	2.97	3.75	5.21	6.20	5.53	4.39	0.00	7.55	9.95	11.93	13.15	13.63	10.17	9.83	10.34	9.55	6.60	6.54	7.84
14	14	40.00	8.40	8.57	9.43	9.92	13.90	16.22	6.94	3.82	3.28	3.46	6.92	6.94	7.55	0.00	2.51	4.88	6.22	7.52	4.17	5.32	8.58	9.43	3.52	5.82	8.19
15	15	41.00	10.13	6.75	8.19	11.28	15.21	17.68	8.85	6.28	5.75	5.72	9.26	9.29	9.95	2.51	0.00	2.51	3.84	5.51	2.88	4.69	8.54	9.87	4.64	6.84	8.99
16	16	42.00	11.33	4.78	6.51	12.04	15.80	18.37	10.37	8.43	8.16	8.21	11.75	11.72	11.93	4.88	2.51	0.00	1.35	3.20	2.40	4.22	8.87	9.78	5.81	7.59	9.32
17	17	43.00	12.26	4.12	6.01	12.77	16.41	19.02	11.44	9.72	9.51	9.56	13.09	13.06	13.15	6.22	3.84	1.35	0.00	2.29	3.21	4.70	8.30	10.15	6.85	8.41	9.91
18	18	44.00	11.96	1.91	3.85	12.08	15.41	18.04	11.51	10.54	10.72	10.96	14.42	14.23	13.63	7.52	5.51	3.20	2.29	0.00	3.55	4.87	6.88	8.89	7.05	8.02	9.82
19	19	45.00	9.05	4.19	5.39	9.66	13.40	15.98	8.27	6.99	7.24	7.62	10.96	10.71	10.17	4.17	2.88	2.40	3.21	3.55	0.00	1.95	5.88	7.46	3.68	5.23	6.93
20	20	46.00	7.80	3.76	4.25	8.13	11.71	14.52	7.49	7.19	7.93	8.54	11.61	11.17	9.83	5.32	4.69	4.22	4.70	4.87	1.95	0.00	3.93	5.56	3.29	3.95	5.21
21	21	48.00	6.79	5.48	4.32	5.89	8.67	11.50	7.43	9.01	10.38	11.26	13.71	11.01	10.34	8.58	8.54	8.07	8.30	6.88	5.88	3.93	0.00	2.06	5.36	3.91	2.91
22	22	49.00	5.47	7.54	6.33	4.09	6.60	9.23	6.57	9.80	10.63	11.63	13.63	12.79	9.55	9.43	9.87	9.78	10.15	8.89	7.46	5.56	2.06	0.00	5.94	3.86	1.72
23	23	50.00	5.57	7.04	7.42	6.65	10.61	13.04	4.60	3.97	5.06	5.90	8.57	8.01	6.60	3.52	4.64	5.81	6.85	7.05	3.68	3.29	5.36	5.94	0.00	2.32	4.67
24	24	51.00	3.94	7.46	7.21	4.48	8.37	10.87	3.79	5.20	6.77	7.77	9.89	9.13	6.54	5.82	6.94	7.59	8.41	8.82	5.23	3.95	3.91	3.86	2.32	0.00	2.39
25	25	52.00	3.93	7.96	7.12	3.10	6.50	9.11	4.87	7.27	9.07	10.11	11.96	11.10	7.84	8.19	8.99	9.32	9.91	9.82	6.93	5.21	2.91	1.72	4.67	2.39	0.00

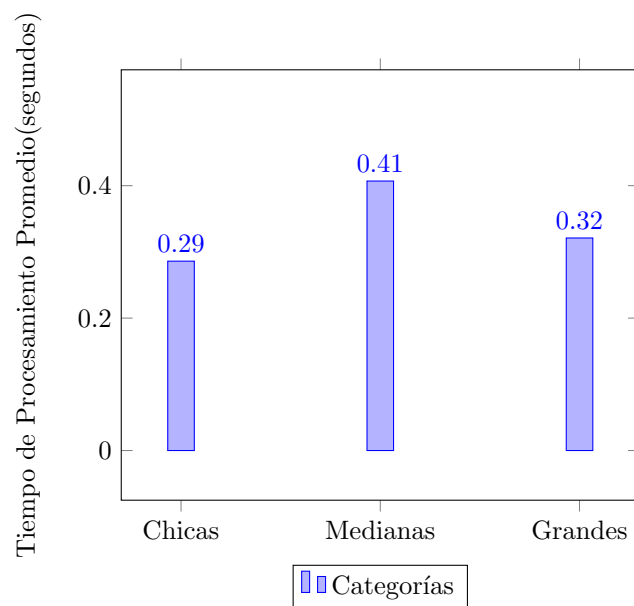
Figure 19: Ejemplo de una matriz grande (25x25) aleatoria.

## 6.4 Resultados

La siguiente tabla muestra los resultados de las distintas matrices y el numero de prueba, de igual manera, se utilizo el mismo código que en el de la matriz completa.

	Prueba 1	Prueba 2	Prueba 3
Chica	Tiempo: 84.73 hours Total trips: 159 Total loads: 78	Tiempo: 95.9 hours Total trips: 184 Total loads: 88	Tiempo: 84.52 hours Total trips: 161 Total loads: 77
Mediana	Tiempo: 124.55 hours Total trips: 237 Total loads: 113	Tiempo: 125.23 hours Total trips: 239 Total loads: 114	Tiempo: 104.58 hours Total trips: 200 Total loads: 94
Grande	Tiempo: 154.61 hours Total trips: 296 Total loads: 141	Tiempo: 147.07 hours Total trips: 283 Total loads: 132	Tiempo: 161.64 hours Total trips: 313 Total loads: 146

	Prueba 4	Prueba 5
Chica	Tiempo: 73.82 hours Total trips: 139 Total loads: 66	Tiempo: 88.89 hours Total trips: 171 Total loads: 81
Mediana	Tiempo: 123.64 hours Total trips: 239 Total loads: 113	Tiempo: 124.36 hours Total trips: 237 Total loads: 113
Grande	Tiempo: 142.25 hours Total trips: 275 Total loads: 128	Tiempo: 153.98 hours Total trips: 298 Total loads: 139



Cómo puede verse en la gráfica de Tiempos de Procesamiento Promedio, las matrices medianas tienden a tardarse más tiempo que las matrices pequeñas y grandes, con esto podemos deducir dos posibles causas, los polígonos elegidos de manera aleatoria o algún tipo de problema de procesamiento computacional.

## 7 Conclusiones

La mejor ruta encontrada es sin duda la que satisface todas las necesidades de los polígonos, con nuestra matrices originales, las que tienen todos los polígonos y sus respectivas distancias.

Nuestro algoritmo heurístico presentó un desempeño medianamente medio al hacer experimentos puesto que se implementó una matriz diferente con la que se diseñó. Este tipo de comportamiento es normal puesto que el algoritmo trata de adaptarse a la información nueva que se colocó.

### 7.1 Conclusiones individuales

#### 7.1.1 Roberto Priego

Los resultados que obtuvimos durante el transcurso de esta actividad fueron bastante interesantes. Considerando que hicimos una solución heurística y tomando en cuenta los datos que nos brindaron durante el reto, siento que nuestra solución está bastante acertada a la realidad y no necesariamente con resultados calculados con fórmulas no vistas en clase. Si bien, me hubiera gustado desarrollarlo un poco más, por el momento, considerando que me gustaría seguir trabajando este proyecto por parte de la iniciativa que nos ha mencionado el Profesor Fernando.

Por último, en cuanto a entrega de código, no me terminó de gustar mucho la forma en la que quedo, realmente es un código bastante normal y no optimizado para utilizarse como base diaria o herramienta de trabajo. Hay muchas cosas que podemos cambiar como hacerlo en C++ o en algún lenguaje de programación más óptimo y sobretodo que no sea tan lineal, aunque es un poco complicado considerando que estamos viendo programación lineal.

#### 7.1.2 Fedra Fernanda Mandujano

Este proyecto me hizo trabajar de manera inteligente, explotar al máximo los datos accesibles y convertirlos en una nueva fuente de información que da valor a los resultados obtenidos por el método heurístico. Por fin pude entender la razón por la que las ciencias de datos tienen matemáticas, no sólo basta saber programar algoritmos, sino que se debe, además tener cierta certeza de que los resultados que arroja el agente sean aproximados, y esto se logra con cálculos y lógica no siempre arrojada con los datos introducidos al algoritmo, sino con métodos creativos que arrojen, de manera más sencilla, resultados que pueden ser contrastados.

### **7.1.3 Juan Marco Castro Trinidad**

Este reto fue un problema de optimización el cual me llevo a pensar de una manera creativa y eficiente para poder encontrar posibles soluciones para esta problemática. La complejidad en la cual se puede plantear un modelo de optimización puede llegar a ser muy complicado como nos dimos cuenta al momento de hacer el algoritmo para la búsqueda de las rutas óptimas. Es por eso, que para contrarrestar los problemas que tuvimos al realizar el código, buscamos optar por usar excel para así poder visualizar los cálculos de una manera mas sencilla, ya que usaríamos los datos que se nos entregaron, y simplemente calcular distancias y tiempos con métodos euclidianos. A pesar de todo esto, toda esta experiencia me hizo dar cuenta de lo importante y necesario que son los modelos de optimización determinista para resolver problemáticas necesarias.

### **7.1.4 Diego Gutiérrez**

Este proyecto me enseñó lo que es que la Ciencia de Datos puede hacer en diferentes áreas y una diferente aproximación a los problemas de la vida real. Antes de entrar a esta clase, no sabía sobre la programación lineal ni como era utilizado, sin embargo, ahora sé que puede ser una de las herramientas más versátiles para tener una solución factible y como es que con observar los reportes dados con GAMS tiene un gran potencial de información para decisiones informadas. Finalmente, quisiera recalcar como es que este fue un proyecto fue un cruce de distintos temas como el VRP y el TSP y restricciones lineales en la carga del camión, lo cuál a mi parecer fue una manera buena para integrar los temas vistos y al mismo tiempo ayudando al medio ambiente.



## 8 Anexos

### 8.1 Mapa escalado

<https://www.geogebra.org/calculator/wrgsxf2h>

### 8.2 Excel

[https://docs.google.com/spreadsheets/d/1DZzvI2BUipJbOAoettbLRW0DgemEI8ML/edit?usp=drive\\_in](https://docs.google.com/spreadsheets/d/1DZzvI2BUipJbOAoettbLRW0DgemEI8ML/edit?usp=drive_in)  
 $106314606468962714795rtpof = truesd = true$

### 8.3 Google Colaboratory

Experimentos y código limpio:

<https://drive.google.com/file/d/1A6qTloDxU6ikZaaI0UZIXzGFqEjgbTT/view?usp=sharing>

## 9 Referencias

- [1] BIODIVERSIDAD. 2000. URL: [https://paot.org.mx/centro/ine-semarnat/informe02/estadisticas\\_2000/informe\\_2000/06\\_Biodiversidad/6.1\\_Diversidad/index.htm#:~:text=La%20flora%20mexicana%20consta%20de,especies%20\(43%25%20end%C3%A9mic](https://paot.org.mx/centro/ine-semarnat/informe02/estadisticas_2000/informe_2000/06_Biodiversidad/6.1_Diversidad/index.htm#:~:text=La%20flora%20mexicana%20consta%20de,especies%20(43%25%20end%C3%A9mic).
- [2] Rhett Butler. *Impacto de la deforestación - Efectos a nivel local y nacional*. 2009. URL: <https://global.mongabay.com/es/rainforests/0902.htm>.
- [3] Rhett Butler. *Impacto de la deforestación—Pérdida de recursos renovables, conflicto con la fauna silvestre*. 2009. URL: <https://global.mongabay.com/es/rainforests/0905.htm>.
- [4] Comisión Nacional Forestal. *Se reduce en 26% la tasa anual de deforestación*. 2022. URL: <https://www.gob.mx/conafor/prensa/se-reduce-en-26-la-tasa-anual-de-deforestacion>.
- [5] Gobierno de México and Comisión Nacional Forestal (CONAFOR). *QUE ES LA DEFORESTACIÓN?* 2020. URL: <https://snmf.cnf.gob.mx/deforestacion/>.
- [6] Thelma Gómez. *Los desafíos ambientales de México en 2022: detener deforestación, proteger áreas naturales y valorar a comunidades forestales*. Mar. 22, 2023. URL: <https://es.mongabay.com/2022/01/desafios-ambientales-de-mexico-en-el-2022/>.
- [7] Elsa Cristina Gonzalez-L, Wilson Adarme-Jaimes, and Javier Arturo Orjuela Castro. “Stochastic mathematical model for vehicle routing problem in collecting perishable products”. In: *Dyna* 82.189 (Feb. 21, 2015), pp. 199–206. DOI: 10.15446/dyna.v82n189.48549. URL: [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S0012-73532015000100025](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0012-73532015000100025).
- [8] Lifeder. *Altiplanicie mexicana: características, flora, fauna, relieve*. Oct. 28, 2020. URL: <https://www.lifeder.com/altiplanicie-mexicana/>.
- [9] National Geographic. *El último medio siglo ha supuesto el mayor coste de deforestación de la historia de la humanidad, arrasando un 15% de la superficie mundial de vegetación, equivalente al territorio de España, Portugal y Francia*. Mar. 20, 2024. URL: <https://www.nationalgeographic.es/medio-ambiente/deforestacion>.
- [10] Procuraduría Federal De Protección Al Ambiente. *IMPLEMENTAN CONAFOR y PROFEPA ACCIONES PARA COMBATIR LA DEFORESTACIÓN EN MÉXICO*. URL: <https://www.gob.mx/profepa/prensa/implementan-conafor-y-profepa-acciones-para-combatir-la-deforestacion-en-mexico>.

- [11] Secretaría De Medio Ambiente Y Recursos Naturales. *Qué hacer antes de reforestar*. URL: <https://www.gob.mx/semarnat/articulos/que-hacer-antes-de-reforestar#:~:text=Se%20recomienda%20elegir%20especies%20propias,deben%20ser%20de%20buena%20calidad>.