ISSN 0188-7297

INSTITUTO MEXICANO DEL TRANSPORTE

SCT
SECRETARÍA DE COMUNICACIONES Y TRANSPORTES

Asistente automático para diseño de rutas de distribución.

José Alejandro Ascencio Laguna
Agustín Bustos Rosales
José Elías Jiménez Sánchez
José Alfonso Balbuena Cruz
Alma Rosa Zamora Domínguez

Publicación Técnica No. 538
Sanfandila, Qro, 2018

SECRETARÍA DE COMUNICACIONES Y TRANSPORTES

INSTITUTO MEXICANO DEL TRANSPORTE

Asistente automático para diseño de rutas de distribución.

Publicación Técnica No. 538
Sanfandila, Qro, 2018

Esta investigación fue realizada en la Coordinación de Integración del Transporte del Instituto Mexicano del Transporte, por José Alejandro Ascencio Laguna, Agustín Bustos Rosales, José Elías Jiménez Sánchez, José Alfonso Balbuena Cruz y Alma Rosa Zamora Domínguez.

Esta investigación es el producto final del proyecto de investigación interna **TI 04/18 Asistente Automático para el diseño de rutas de distribución**.

Contenido

Índice de figuras

Sinopsis

Dada la demanda de la optimización en los procesos de distribución de productos y en las necesidades particulares de los clientes potenciales, se ha detectado la necesidad de proponer una herramienta que permita la adecuación de modelos logísticos para el ruteo, buscando eliminar tiempos muertos, permitir una mejor planificación, perfeccionar los procedimientos y el establecimiento de indicadores de gestión.

La tecnología hoy en día ha permitido la generación de ventajas competitivas, en este contexto el conocimiento de la inteligencia artificial y la sistematización computacional ha potenciado los procesos logísticos, permitiendo ejecutar técnicas de explosión combinatoria para obtener soluciones factibles, sin embargo, es importante mencionar que los problemas de ruteo son complejos de resolver aun contando con poder de cómputo, es por eso que en este estudio se presenta una propuesta para el diseño de rutas de distribución a través del Heurístico CVRP (Problema de ruteo vehicular con capacidad) de **Google Optimization Tools**, una interfaz en un entorno Web y la obtención de distancias reales con el **Google Maps**.

Abstract

Given the demand for optimization in the processes of product distribution and the particular needs of potential customers, the need has been detected to propose a tool that allows the adaptation of logistic models for routing, seeking to eliminate downtime, allowing better planning, perfecting procedures and establishing management indicators.

Technology today has allowed the generation of competitive advantages, in this context the knowledge of artificial intelligence and computational systematization has enhanced the logistic processes, allowing combinatorial explosion techniques to be performed to obtain feasible solutions, however, it is important to mention that the routing problems are complex to solve even with computing power, that is why in this study we present a proposal for the design of distribution routes through the Google Optimization Tools CVRP Heuristic, an interface in a Web environment and Obtaining real distances with Google Maps.

Resumen ejecutivo

En el presente estudio se describe el desarrollo de una herramienta informática que da soporte a la toma de decisiones en el contexto de optimizar el plan de ruta de distribución entre el proveedor y sus clientes.

El OSRM (Open Source Routing Machine) de Google Maps ha permitido dar soluciones de reducción de costos en la operación del transporte, proponiendo las rutas origen-destino más convenientes y algoritmos de ruteo eficientes. Los heurísticos en este contexto han demostrado ser muy capaces en los problemas complejos de tiempo no polinomial y de explosión combinatoria; logrando generar soluciones óptimas en tiempos aceptables.

La plataforma propuesta en este estudio pretende cimentar las bases para fungir como una aplicación experimental para la adecuación de nuevos modelos de ruteo, proponiendo como trabajos a futuro el uso de la inteligencia artificial con sus más importantes técnicas como son el *Machine Learning*, *Redes Neuronales Artificiales*, *Genéticos* y *Heurísticos*.

Introducción

“La gestión logística se ha convertido en el elemento de carácter estratégico en el mundo empresarial de la actualidad, dentro de la misma se destaca, por su impacto en los clientes e importancia económica, el subsistema de distribución” (Reyes Chavez, Tamayo Garcia, & Leyva Zaldívar, 2011). En el contexto del proceso de aprovisionamiento y/o distribución, el diseño de rutas de una manera óptima ha generado ventajas competitivas, además de que ha motivado al investigador a proponer modelos de ruteo que mejoran el desempeño logístico.

El Problema del Ruteo de Vehículos (VRP, en inglés, Vehicle Routing Problem), busca la solución más óptima con diferentes restricciones como el número de vehículos, capacidad, orígenes-destino, la demanda de los clientes, etc., este tipo de problemas de optimización es de tipo combinatoria y por lo regular pertenece a la clase NP-Hard (Non deterministic Polynomial-Hard) lo que significa que no es posible resolverlos en tiempo polinomial.

La evolución del VRP, se remonta con el Problema del Agente Viajero (TSP, en inglés, Travelling Salesman Problem) por Flood en 1965, donde el agente busca determinar cuál ruta debe seguir para visitar cada ciudad-destino una sola vez y regresar al origen a una distancia mínima. En 1959, Dantzong y Ramser como variación al TSP, modeló el despacho de combustible a través de una flota de camiones a diferentes estaciones de servicio, esto dio pauta a los posteriores modelos que consideraban más variables y restricciones. En 1960 Miller, Tucker y Zemlin proponen el TSP múltiple o m-TSP, en el cual se tiene un depósito y un número determinado de vehículos; e1969 Tillman con el PTSP o TSP probabilístico, el cual busca encontrar el mínimo costo de recorrido esperado a través de nodos asociados a una probabilidad (Rocha Medina, González La Rota, & Orjuela Castro, 2011).

El m-TSP, puede ser considerado con un VRP, es decir, como un problema con restricción de capacidad de flota, a este problema actualmente se le denomina CVRP (Capacited VRP), donde a cada consumidor le corresponde una demanda determinística, todos los vehículos son de la misma capacidad y salen de un solo centro de distribución. A partir este tipo de soluciones han surgido un sinfín de variaciones, tales como VRPTW (VRP with time windows) en 1979, VRPMTW (VRP with multiple time windows) en 1986, VRPSTW (VRP witj soft time windows) en 1992, SDVRPTW (VRPTW with partial deliveries) en 1989, VRP Heurísticos y Metaheurísticos desde 1990 (Rocha Medina et al., 2011).

Con base a lo anterior y siendo uno de los objetivos del **Plan Sectorial de Comunicaciones y Transportes 2013-2018**, específicamente el Objetivo 6, “Desarrollar integralmente y a largo plazo al sector con la creación y adaptación de tecnología y la generación de capacidades nacionales”, alineado al **Plan Nacional de Desarrollo (PND) 2013-2018** con el objetivo de la Meta Nacional 3.5 “Hacer del desarrollo científico, tecnológico y la innovación pilares para el progreso económico y social sostenible” (Presidencia de la República, 2013a) y al quinto objetivo del **Programa para un Gobierno Cercano y Moderno (PGCM)**, “Establecer una Estrategia Digital Nacional que acelere la inserción de México en la Sociedad de la Información y del Conocimiento” (Presidencia de la República, 2013b), este proyecto pretende contribuir con la generación de nueva tecnología, desarrollando un aplicación que dé soporte al diseño de rutas de distribución a través de un mapa virtual para la determinación de origen-destinos y asignación de flota vehicular, además de un CVRP para resolver el problema de ruteo.

1 Arquitecturas y lenguajes de programación

En este apartado se describen las proyecciones de infraestructura de solución, además de los lenguajes de programación y librerías utilizadas. Dichas proyecciones representan gráficamente los módulos, métodos de programación y servicios web externos, además de la manera en la que estos se relacionan para cumplir con los objetivos del estudio.

1.1 Arquitectura de tres capas

Los diagramas de tres capas permiten separar las partes que componen una aplicación, esto con el objetivo de facilitar el desarrollo, el mantenimiento y la escalabilidad.

Se ha elegido esta arquitectura debido a que se requiere adecuar nuevos modelos de optimización, esta infraestructura permite separar la interfaz de usuario de la lógica de los modelos, con esto se consigue posibilitar el uso de más de un lenguaje de programación y no limitar el uso de algoritmos ya desarrollados en el mercado.

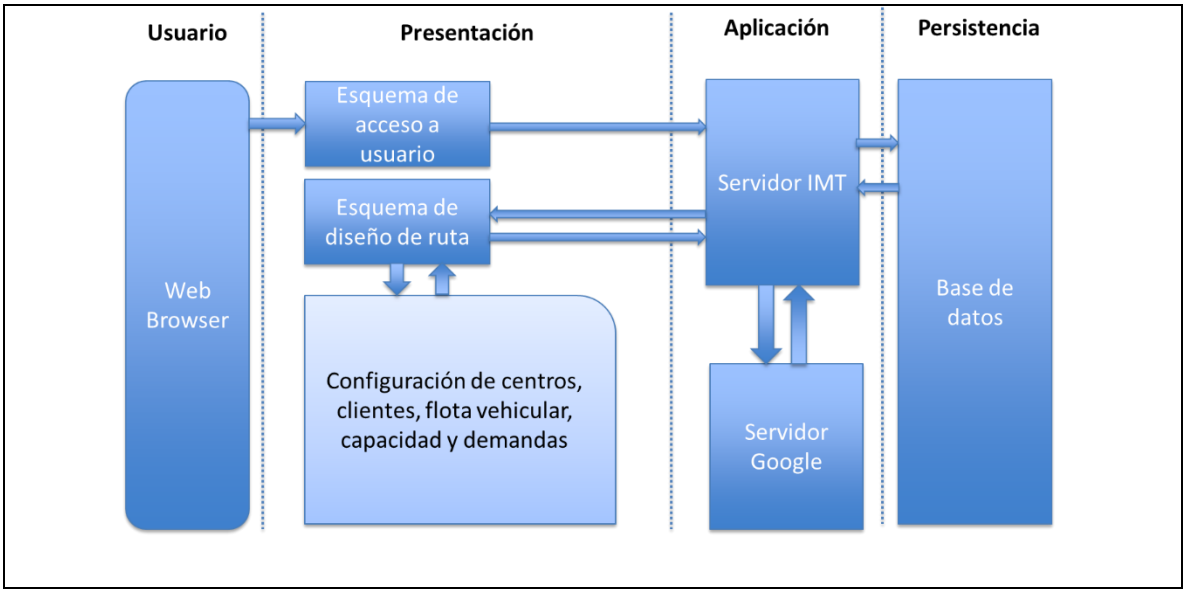

Figura 1.1 Diagrama de uso.

Se puede observar en la Figura 1.1 que el usuario accederá a la aplicación a través de un navegador Web, existen sólo dos interfaces, el *Esquema de acceso a usuario* para que el usuario se identifique y *Esquema de diseño de rutas* para establecer la configuración del problema, los dos esquemas acceden a la lógica del *Servidor IMT* que busca las credenciales del usuario en el servidor de *Base de datos* y resolver el problema de ruteo; por otro lado, el *Servidor IMT* accede a la lógica del *Servidor de Google* para obtener la matriz de distancias entre centro de distribución y los clientes.

1.2 Arquitectura Cliente-Servidor

La arquitectura cliente-servidor tiene como principal objetivo procesar la información de un modo distribuido, así los usuarios finales pueden estar dispersos en un área geográfica extensa (un edificio, una localidad, un país, etc.) y acceder a un conjunto de recursos compartidos. (Ruiz Aranda, 2013).

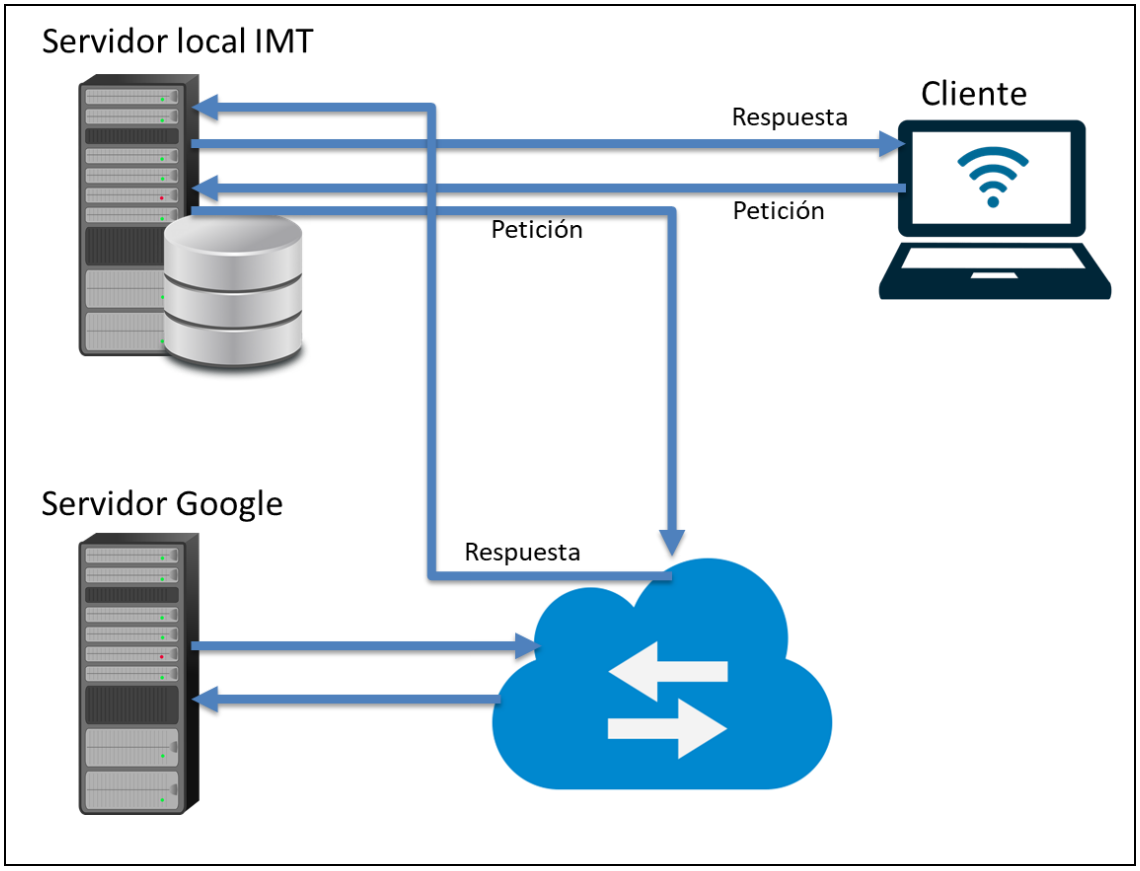

Figura 1.2 Diagrama Cliente-Servidor.

En la Figura 1.2 se puede visualizar la relación entre los servidores y el cliente, la conexión entre *Cliente* y *Servidor IMT* es local, pero también en algún momento el *Servidor IMT* funge como cliente en una conexión de Internet con el *Servidor Google*.

1.3 Lenguajes de programación y librerías

Un lenguaje de programación, es una estructura sintáctica y semántica que sirve para declarar instrucciones a un programa de computadora (Pérez Porto & Merino, 2012). Dichas declaraciones tienen la capacidad de generar una interfaz gráfica para acceder a funcionalidades específicas de automatización de procesos.

Una librería o también llamada biblioteca, es un conjunto de implementaciones funcionales y codificadas en un lenguaje de programación que ofrece una interfaz bien definida y que se puede invocar con una sentencia determinada.

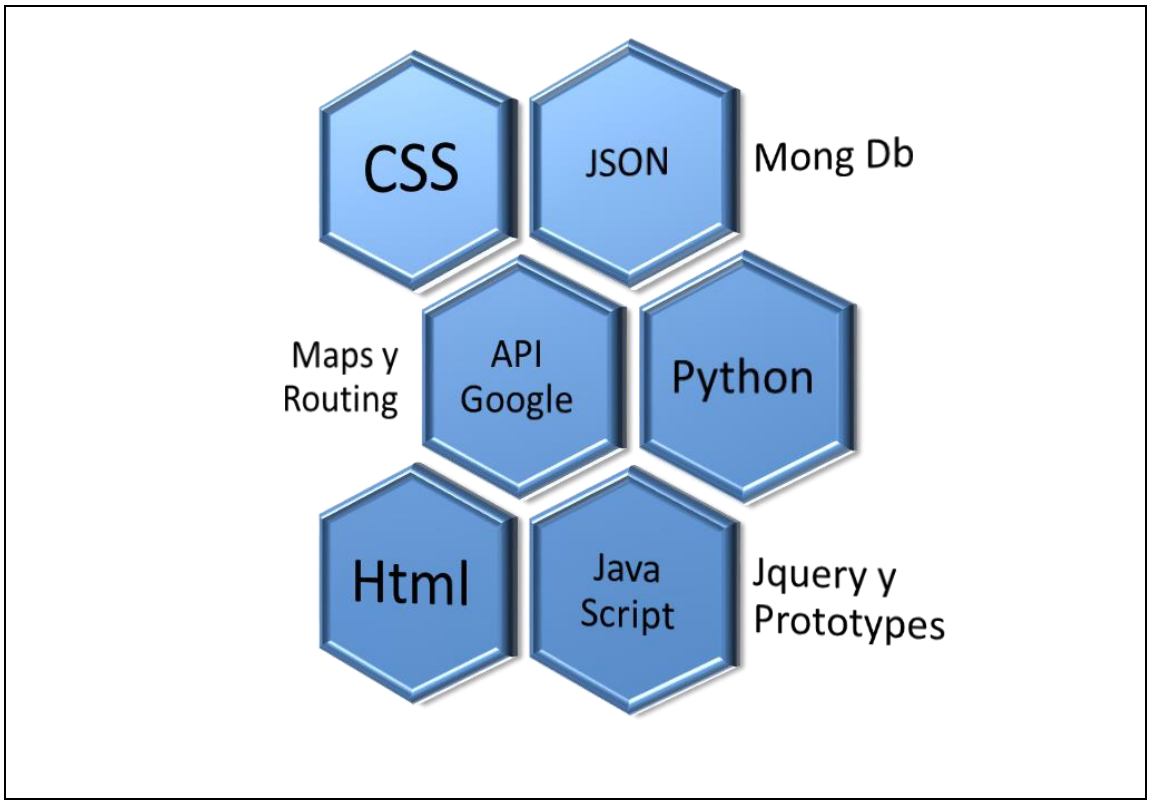

Figura 1.3 Lenguajes de programación

El desarrollo de esta aplicación ha exigido un conjunto de lenguajes de programación y librerías de tipo *freeware*(de distribución gratuita) esto con el objetivo de agilizar la implementación del mismo. En la Figura 1.3, se puede observar

la composición de herramientas utilizadas, a continuación, la clasificación en tres capas:

- Presentación: El uso de *Html (HyperText Markup Language)* para la declaración de objetos de interfaz y *CSS (Cascading Style Sheets)* para sus atributos de estilo, tales como color, tamaño, posición, etc. Por otro lado, lenguaje de programación *Java Script* con la librería *Jquery* para manipulación funcional de los elementos de interacción con el usuario.

- Aplicación: El lenguaje de programación *Python* como lenguaje principal para generar el *API (Application Programming interface)* que contiene todos los procedimientos funcionales de la interfaz y los accesos a las librerías externas, las *API's* de Google para calcular las matrices de distancias y la funcionalidad CVRP de ruteo, además del uso *Prototypes* de *Java Script* para la codificación de fácil reusabilidad, escalabilidad y herencia de atributos y métodos.

- Persistencia: El uso del formato *Json (Java Script Object Notation)* para el intercambio de datos entre los servidores y el modelo de persistencia que se encuentra en el administrador de base de datos *Mongo DB*.

2 Problema de enrutamiento de vehículos

En la denominación de problemas de ruteo de vehículos VRP (Vehicle Routing Problem), se engloban un conjunto de técnicas, variables y personalizaciones que generan resultados distintos, dichos procedimientos pueden ser muy sencillos o muy complejos, depende del número de restricciones, posibles soluciones y tiempos requeridos de respuesta.

Los problemas de ruteo de vehículos han sido de gran interés por ya más de cincuenta años, esto debido a que son considerados NP-Complejos y aún hay muchas oportunidades de mejora, además de su gran importancia como modelo de optimización (Orrego Cardozo, 2013)

2.1 CVRP

El CVRP (Capacitated Vehicle Routing Problem), se puede describir de manera sencilla como una flota de vehículos con capacidades homogéneas que tienen que satisfacer la demanda de un grupo de clientes que empiezan y finalizan su ruta en un centro de distribución al menor costo posible, además de identificar el orden de visita de los mismos (Orrego Cardozo, 2013).

Los métodos para la resolución de problemas de tipo CVRP se clasifican en dos grupos, los algoritmos exactos y lo aproximados: los primeros buscan la solución más óptima, pero tiene el inconveniente de requerir tiempos de ejecución elevados, pues valida y compara el conjunto de soluciones posibles; por otro lado, los segundos dan una solución suficientemente óptima (no la óptima) en un tiempo de ejecución razonable (Laporte, Toth, & Vigo, 2013).

Los algoritmos con métodos exactos se basan en su formulación, como un problema de propagación mixta, quiere decir que cuenta con variables enteras y otras binarias. Uno de los más populares, es ramificación y acotación (Chao, Lei-shan, Ti-xiang, & Ran, 1965). Una herramienta informática que resuelve problemas de optimización a través de métodos exactos es Xpress-Mosel, pero tiene como desventaja el costo de la licencia, pues es bastante costosa (Fico, 2018).

Para la resolución de problemas de tipo CVRP se usan heurísticas y metaheurísticas que logran solventar el problema de los elevados tiempos de ejecución, con estos métodos no se tiene la garantía de obtener la solución óptima, aunque sí una suficientemente buena.

Existe un conjunto de software comercial para la resolución de problemas con métodos aproximados (The Institute for Operations Research and the Management Sciences, 2018).

A continuación, la clasificación de los métodos aproximados:

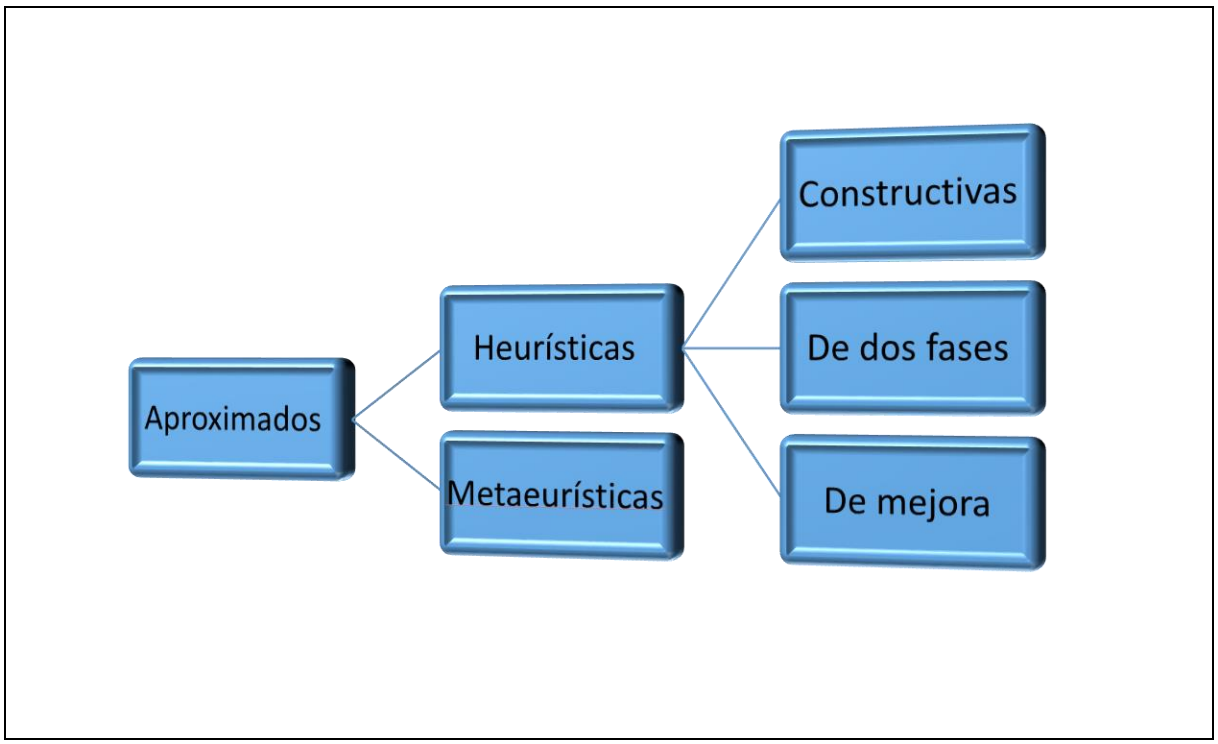

Figura 2.1 Métodos aproximados para la resolución de CVRP

A continuación, una breve descripción de la clasificación de métodos aproximados presentados en la Figura 2.1

- Los algoritmos heurísticos constructivos, son técnicas iterativas que gradualmente van creando una solución, donde en cada paso se agrega un nuevo elemento que se evalúa y se deshecha cuando no es mejor que otro. Un claro ejemplo es el método de *Ahorros de Clarke & Wright* (Alinaghian, Kaviani, & Khaledan, 2015).

- Los algoritmos heurísticos de dos fases que representan dos procesos: agrupación de vértices en rutas factibles y construcción de la ruta, de tal manera que el problema se resuelve de manera secuencial, un claro ejemplo es el método de Fisher y Jaikuman (Ramalhinho & Serra, 2002).

- Los algoritmos heurísticos de mejora realizan una serie de intercambios de vértices dentro de una misma ruta o entre distintas rutas para buscar la mejor solución. Está técnica consiste en búsquedas locales que parten de una solución completa y usando el concepto de vecindario recorren y evalúan

- parte del espacio para encontrar el óptimo local. Un claro ejemplo es, el método Swap (Lum, Chen, Wang, Golden, & Wasil, 2015).

- Los algoritmos metaheuristicos, son la combinación de distintas técnicas heurísticas para realizar la exploración del dominio de búsqueda de una manera más eficiente, se consideran seis tipos de metaheuristicas: recosido simulado, genéticos, redes neuronales, búsqueda tabú, colonia de hormigas y GRASP (Toth & Vigo, 2002).

2.2 Heurístico

Un heurístico es un modelo que encuentra una solución óptima entre conjunto total de soluciones factibles, pero no garantiza que la solución seleccionada sea la más óptima (Luna López, 2015).

2.3 Justificación del uso de algoritmo heurísticos

El uso de un heurístico en cualquier problema NP-Complejos se justifica en la búsqueda de espacios de soluciones locales y tratando de buscar soluciones óptimas en tiempos razonables y evitar el estancamiento en el universo de soluciones. Es posible expresar que en problemas de espacios de búsqueda muy grandes no sólo es práctico usar este tipo de técnicas, sino que es esencial, pues dicha formulación discrimina rutas de búsqueda no prometedoras, lo cual, si se plantea de manera objetiva, las soluciones siempre serán óptimas, cabe mencionar que dichos modelos sacrifican la validación del universo de soluciones y la detección de la solución más óptima, para generar una en un tiempo razonable.

2.4 Algoritmo de ahorros de Clark & Wright

El Algoritmo de ahorros de Clark & Wright, es una de las técnicas más populares para resolver VRP a través de heurísticas, consiste en el principio de combinar una solución de dos rutas diferentes para formar una nueva ruta donde se validen los ahorros.

El ahorro en la combinación de soluciones es obtenido con la siguiente fórmula (Benito Quintanilla, 2015):

$$s_{ij} = d_{i0} + d_{0j} - d_{ij} \quad (1)$$

donde,

s_{ij} es el valor del ahorro.

d_{i0} es el costo del arco i al punto 0 o depósito.

d_{0j} es el costo del arco 0 o depósito al punto j.

d_{ij} es el costo del arco i al punto j.

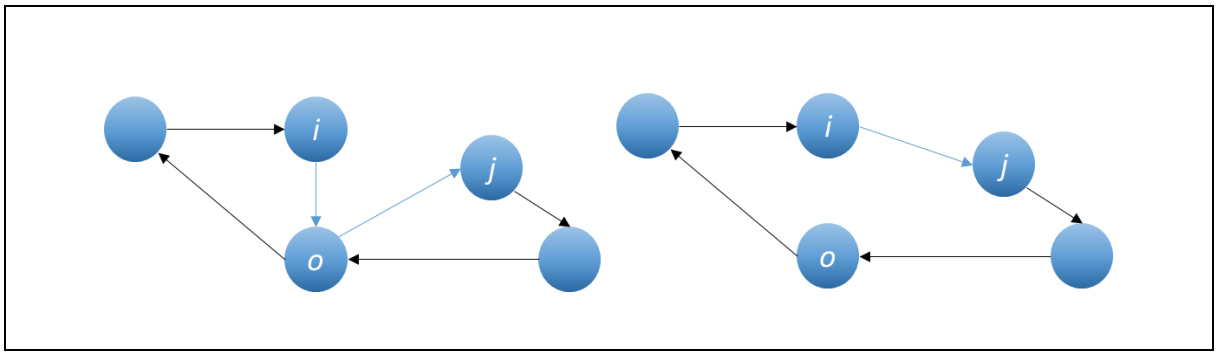

Figura 2.2 Dos rutas antes y después de ser unidas

Como podemos ver en la Figura 2.2 y contemplando la ecuación (1) podemos observar que la nueva ruta a generado un costo, pues el arco del depósito al punto ***j*** se ha eliminado y se ha creado uno nuevo del punto ***i*** al punto ***j***.

Como podemos observar la heurística antes expuesta logra resolver el problema sin validar al conjunto posible de soluciones, mientras que lo contrario es posible contemplar en la siguiente figura:

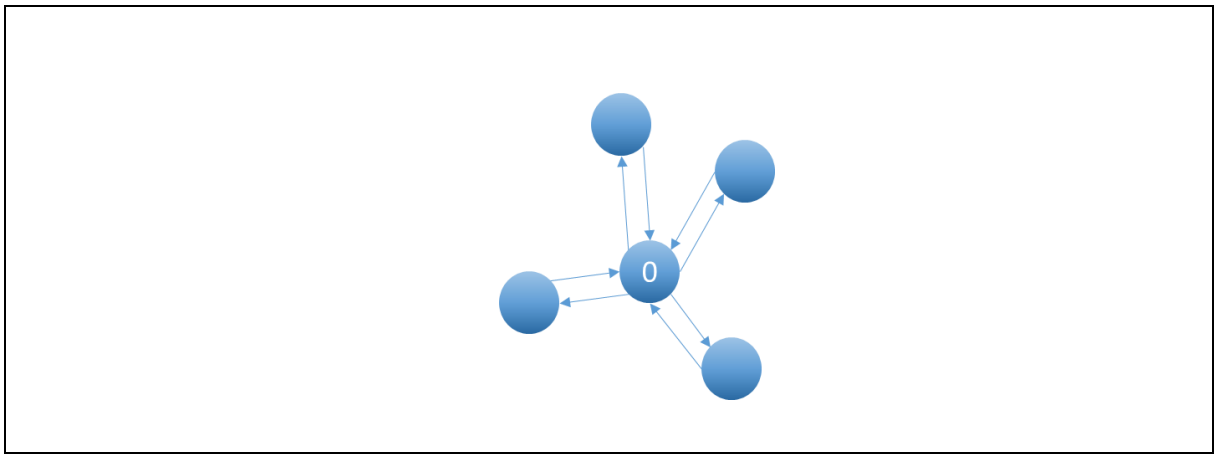

Figura 2.3 Validación de la mejor ruta con técnicas exactas

La variante del Algoritmo de ahorros de Clark & Wright para resolver CVRP presentada por (Benito Quintanilla, 2015) es la siguiente:

$$s_{ij} = d_{i0} + d_{0j} - \lambda d_{ij} - \mu\left|d_{0i} - d_{j0}\right| + \frac{v(dem_i + dem_j)}{demanda\ media} \quad (2)$$

donde,

- $\boldsymbol{\lambda}$ es el parámetro de forma de ruta para penalizar la unión de rutas con clientes lejanos con la restricción $\mathbf{0, 1} \leq \boldsymbol{\lambda} \leq \mathbf{2}$.
- $\boldsymbol{\mu}$ es el parámetro de peso que considera la asimetría de las distancias entre depósitos y cliente con la restricción $\mathbf{0} \leq \boldsymbol{\mu} \leq \mathbf{2}$.
- $\boldsymbol{v}$ es el parámetro de peso que considera el uso de la capacidad total de los vehículos con la restricción $\mathbf{0} \leq \boldsymbol{v} \leq \mathbf{2}$ y la filosofía de que en cuanto más larga sea la ruta, es mejor.

3 Google Directions API

Google Directions API es un servicio que calcula las direcciones entre localidades, es posible obtener las trayectorias por un conjunto de modos de transporte, tales como tránsito, conducción, caminar y bicicleta (Google Maps Platform, 2018). El principal objetivo de este recurso, es obtener el tiempo y distancia de arribo entre un punto y otro, también permite acceder a todas las posiciones de: la trayectoria, direcciones, elevaciones, entre otros atributos.

3.1 Políticas importantes

A continuación, se especifican las políticas Google de uso más importantes que se toman en cuenta en el desarrollo de esta aplicación (Google Maps Platform, 2018):

- Se pueden mostrar los resultados del API Google en Json Result o en un mapa, sin embargo, se restringe a usar el mapa de Google y ningún otro más.

- Se deben mostrar los créditos de Google cuando los resultados de dirección están en un mapa.

- Cuando los resultados publicados muestren detalles del tránsito, se deben mostrar las agencias proveedoras del recurso.

3.2 DirectionsResult Object

DirectionsResult Object, es un objeto resultado que se recibe al hacer la petición web a *DirectionsServide* de Google, dicha información contiene los siguientes atributos:

- Waypoints: Los puntos de planeación que permiten que la ruta de un origen a destino pase por algún punto en especial, sacrificando el trayecto más corto.

- Routes: Los atributos de la ruta origen-destino, tales como: la latitud y longitud, el nombre de la localidad, tiempo arribo, distancia de arribo, el conjunto de posiciones geográficas que componen la trayectoria, entro muchos otros más.

- Steps. Son las indicaciones de ruta origen-destino.

- Transit. Contiene los detalles de tránsito en el conjunto de indicaciones ruta origen-destino.

3.3 Google Directions API en el diseño de rutas.

Ya que uno de los principales objetivos en la resolución de problemas de ruteo con vehículos capacitados, es la minimización de costos: la distancia entre un punto y otro genera un costo de operación, ya sea en el contexto de tiempo o en el monetario, por lo tanto, uno de los parámetros de entra y configurable por el usuario final más importante para el modelo de solución, es la matriz de distancias de arribo, ya que con estas medidas es posible calcular los tiempos de recorrido y el consumo de combustible en un plan de distribución.

Google API también cuenta con una función para calcular la matriz de distancia directamente, sin embargo, se ha descartado la posibilidad de usar esta herramienta debido a las limitantes que genera en el propósito de cimentar la infraestructura informática para la adecuación de nuevos modelos de solución. A continuación, se describe dichas restrictivas (Google Maps Platform, 2018):

- La matriz de distancias permite sólo establecer 25 orígenes y 25 destinos, lo cual afecta la funcionalidad de la plataforma propuesta en este estudio, lo cierto es que un problema real por lo regular implica más de 25 destinos en una empresa mediana de transporte de carga.

- La matriz de distancias calcula la distancia entre origen-destino, pero no entre destino-destino, lo cual limitaría a modelos de resolución de problemas de ruteo donde un vehículo tiene la capacidad de atender a más de un cliente en un solo viaje.

- La matriz de distancias no retorna los detalles de la ruta, por ejemplo, las direcciones textuales o los detalles de tránsito, lo cual es importante para problemas de ruteo que considere rutas prohibidas y peligrosas.

- La matriz de distancias no permite la modificación de la ruta propuesta por su servidor (puntos de planeación), lo cual es importante para el transportista que de manera empírica conoce las rutas más peligrosas y decide sacrificar la más corta para optar por una con más seguridad.

4 Google OR-Tools

Google OR-Tools es un framework (entorno de trabajo) de código abierto para la automatización, está diseñado para resolver problemas complejos de enrutamiento de vehículos, flujos, programación lineal, entera y de restricciones (Google OR-Tools, 2018).

4.1 Principales características de OR-Tools

Esta herramienta puede ser utilizada en cuatro lenguajes de programación, C++, PYTHON, Java y C#, además de poder usar cualquiera de sus seis principales herramientas de optimización GUROBI, CPLEX, SCIP, GLPK, GLOP y CP-SAT.

A continuación, sus principales soluciones (Google OR-Tools, 2018):

- Programación lineal: Cómputo para encontrar la mejor solución a problemas modelados como conjunto de relaciones lineales.
- Optimización de restricciones: También denominada programación de restricciones para identificar soluciones factibles de un gran grupo de cantidades donde el problema puede ser modelado en término de restricciones.
- Optimización entera: También denominada como programas lineales de enteros mixtos, que requieren de variables enteras, pero no del todo.
- Ruteo: Cómputo para la resolución de problemas con el objetivo de encontrar caminos eficaces para transportar artículos a través de una red compleja.
- Embalaje: Tiene como objetivo encontrar el conjunto de objetos para empacar en contenedores, considerando volumen y capacidades vehiculares.
- Flujos de red: Tiene como objetivo representar flujos de red, como un flujo ferroviario, considerando arcos y ciudades.
- Asignación: Esta herramienta busca asignar al trabador el conjunto de tareas distintas a otros con el objetivo de minimizar el costo total de proceso.
- Programación: La herramienta busca minimizar la cantidad total de tiempo traducido a costo requerido para completar las tareas de un proceso completo.

4.2 Ruteo

A pesar de que el framework OR-Tools, es una herramienta informática muy completa para la resolución de problemas de optimización y que se relaciona con el contexto del sector transporte, en este estudio nos concentraremos en el módulo de ruteo para resolver el problema de diseño de rutas.

Los problemas de ruteo se dividen en dos principales problemas: ruteo de nodos y ruteo de arco, el primero se concentra en la expresión de ubicaciones y el segundo en los bordes que los conectan. El problema de ruteo de arco busca la ruta más corta que atraviesa cada calle en una región asignada, al contrario del ruteo de nodo que considera además el costo por viajar por cierta ruta.

OR-Tools contiene un conjunto de librerías para resolver varios tipos de problemas de ruteo de nodo:

- **Traveling Salesman Problems (TSP)**. Se considera un solo vehículo para la distribución de productos.
- **Vehicle Routing Problems (VRP)**. Es el TSP pero que considera más de un vehículo.
- **Capacitated Vehicle Routing Problems (CVRP)**. Donde los vehículos tienen limitaciones en cargas máximas.
- **Vehicle Routing Problems with Time Windows (VRPTW)**. Los vehículos deben comenzar y terminar el proceso de distribución en un tiempo determinado.
- **Vehicle Routing Problems with Resource Constraints (VRPRC)**. Considera restricciones de recursos, por ejemplo, el depósito tiene limitaciones de vehículos o los vehículos requieren carga de combustible.
- **Vehicle Routing Problems with Pickup and Delivery (VRPPD)**. Considera la recolección de los productos antes de entregarlos al cliente.

5 Desarrollo de la plataforma

Como primera fase para este estudio se desarrolla una plataforma con arquitectura web que contempla la interfaz de configuración de problemas de ruteo y la implementación del módulo CVRP de OR-Tools de google. En este apartado se describen los principales objetivos del desarrollo de la aplicación, los requerimientos detectados, las restricciones y los problemas a los que se enfrentaron en la fase de codificación, al final se muestran las interfaces de usuario que ejemplifican el funcionamiento último de la plataforma desarrollada.

5.1 Objetivos

El principal objetivo del desarrollo es generar una herramienta informática para el diseño de rutas de distribución y posibilitar la adecuación de otros algoritmos como trabajos a futuro.

Objetivos específicos:

- Implementar CVRP de OR-Tools de Google.
- Diseño y codificación de las interfaces de configuración del problema.
- Obtención de la solución óptima de ruteo.

5.2 Requerimientos

Los requerimientos detectados fueron los siguientes:

- Requerimientos del negocio:
    - Una herramienta automática que le de soporte al investigador con la configuración de rutas óptimas.
    - Infraestructura con alta capacidad de escalabilidad para la adecuación de nuevos modelos de solución.
- Requerimientos de usuario:
    - Interfaz gráfica intuitiva y amigable.
    - Interfaz que permita establecer al conjunto de centros de distribución.
    - Interfaz que permita establecer al conjunto de clientes.
    - Interfaz que permita establecer capacidades vehiculares y demandas.
    - Interfaz que permita visualizar la solución óptima.

- Requerimientos de sistema:
    - Organizar la información geográfica de las configuraciones del problema de ruteo y establecer una estructura de datos que pueda manipularse con eficiencia.
    - Obtener las distancias reales entre el centro de distribución y los clientes con Google Directions API.
    - Generar las clases (objetos de programación) que le den el formato a los datos de tal manera que sean reconocidos por las librerías de ruteo.
    - Formatear las respuestas de OR-Tools y presentar los resultados.
    - Permitir el acceso multiusuario a la herramienta en un entorno Web.
- Requerimientos de software:
    - Funcionales.
        - Autocompletado de posiciones geográficas a través de los nombres de las localidades para encontrar de una manera más rápida el punto exacto.
        - Permitir al usuario agregar el centro de distribución a través de un mapa de Google, además de poder editar el nombre y etiqueta de los puntos geográficos requeridos.
        - Obtener la matriz de distancias reales entre el centro de distribución y los clientes.
        - Permitir al usuario establecer la flota vehicular y sus capacidades, además de las demandas de los clientes.
        - Visualizar el resultado del algoritmo de ruteo.
    - No funcionales.
        - Configuración de un servidor Web, en primera instancia con acceso local.
- Requerimientos de hardware:
    - Un servidor con sistema operativo Windows Server de 64 bits dedicado a la aplicación.
    - Acceso a internet con al menos 4 Mb. de ancho de banda.

5.3 Alcances y limitaciones

A continuación, la detección de los alcances y limitaciones:

- Alcances:
    - El sistema genera una solución de ruteo a través del algoritmo heurístico de tipo CVRP con el framework OR-Tolls de Google.
    - Permite la agregación, modificación y eliminación de puntos geográficos, tanto en centros de distribución como en clientes.
    - Permite la configuración de capacidades vehiculares y las demandas de los clientes.
- Limitaciones:
    - Un solo centro de distribución.

- Capacidades homogéneas para la flota vehicular.
- Supone que los centros de distribución soportan la demanda de los clientes.
- Las carreteras prohibidas no son consideradas.
- No es posible visualizar las rutas.
- No es posible modificar las rutas generadas por Google Maps.
- No considera ventanas de tiempo.

5.4 Complicaciones en el desarrollo

A continuación, se describen algunas de las complicaciones que se lograron resolver en el desarrollo de la plataforma y la adecuación en las librerías y herramientas mencionadas anteriormente:

- El algoritmo de OR-Tools para CVRP calcula distancias euclidianas, fue necesario usar Google Directions API para encontrar las distancias reales.
- Google Maps no permite peticiones web de distancia simultaneas con licencia freeware.
- OR-Tools imprime los resultados a nivel consola.
- OR-Tools no recibe parámetros de entrada, es configurable a nivel programación.
- Entender y modificar el algoritmo de OR-Tools fue lo más complejo.

5.5 Resultados del desarrollo

A continuación, los resultados gráficos obtenidos en el desarrollo:

- Acceso a usuarios

Figura 5.1 interfaz acceso a usuarios

- Captura de centros de distribución

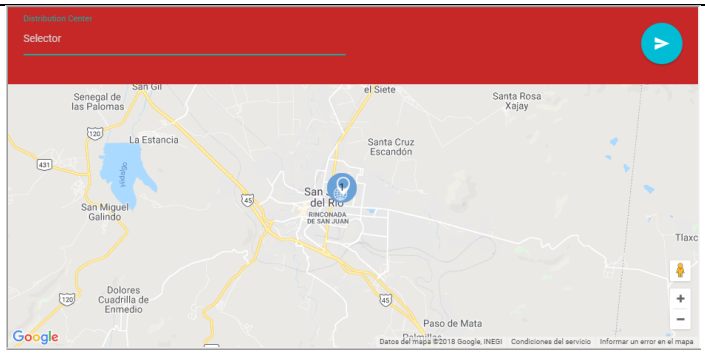

Figura 5.2 interfaz captura de centros de distribución

- Captura de clientes

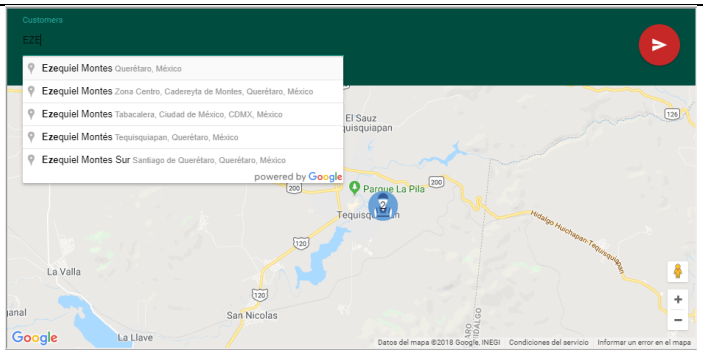

Figura 5.3 interfaz captura de clientes

- Obtención de tiempos de arribo todos contra todos

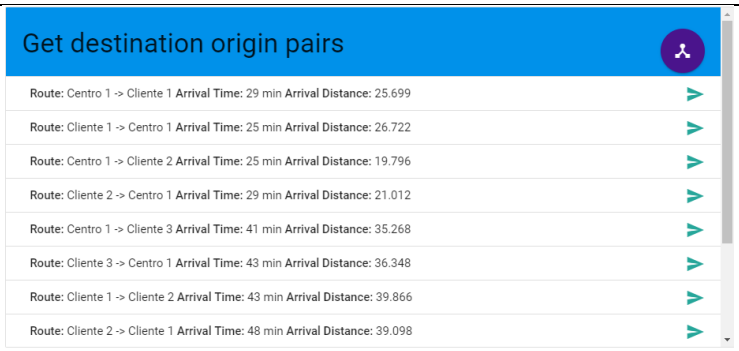

Figura 5.4 interfaz tiempos de arribo

- Configuración del problema

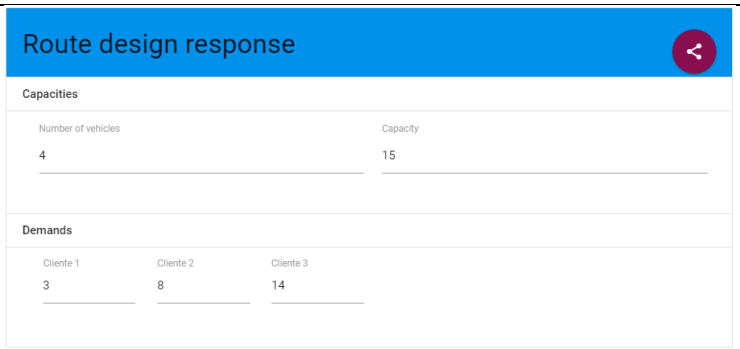

Figura 5.5 interfaz configuración del problema

- Resultados

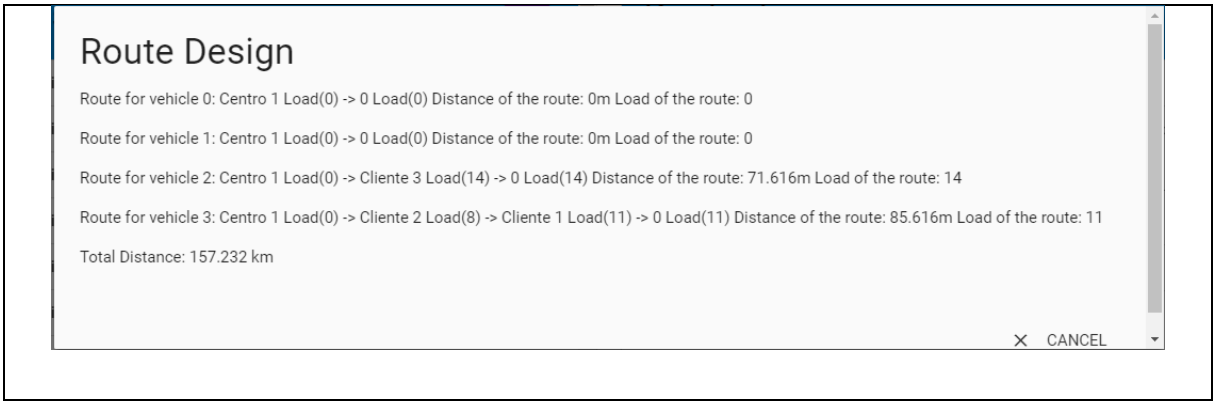

Figura 5.6 interfaz resultados

- Sistema integral

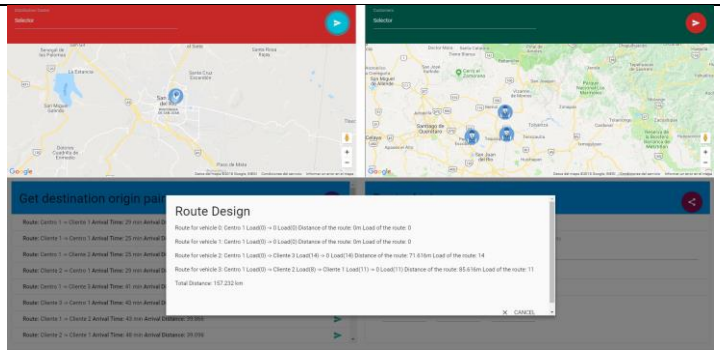

Figura 5.7 interfaz sistema integral

Como se puede observar en la Figura 5.7 los resultados generados por la aplicación, son las rutas propuestas para cada vehículo configurado, la distancia a recorrer en orden para el menor costo y el número de unidades a entregar a cada cliente, al final una leyenda de la distancia total recorrida por todos los vehículos con el fin de servir de insumo al cálculo de los costos de operación. Es importante mencionar que la decisión tomada por el algoritmo considera vehículos sin ruta, ya que, el problema puede resolverse con menos vehículos.

6 Conclusiones

En la actualidad existe un sin fin de herramientas que dan soporte en la optimización de recursos, algunas de ellas especializadas en la gestión de distribución de mercancías, sin embargo, es importante mencionar que la mayoría son de alto costo o no considera tiempos de arribo en distancias reales, además de que la mayoría de veces las aplicaciones son diseñadas con base en los atributos específicos de la zona done se encuentran, por lo tanto, sigue habiendo una amplia área de oportunidad (Juan, Mejía, & Arroyo, 2017).

Con base en la anterior se ha detectado que el área de oportunidad se concentra en la creación de modelos de solución que considere las rutas prohibidas para determinado tipo de vehículo, zonas de alta accidentabilidad y seguras en el sentido de la delincuencia, esto implica no sólo el desarrollo de nuevas heurísticas, sino que también el uso de los paradigmas de **Inteligencia Artificial**, específicamente *Machine Learning*, *Big Data* e *Internet de las cosas (IoT)*.

El poder hacer una sinergia entre las empresas transportistas a través de *IoT* para la comunicación entre los dispositivos de posición geográfica de los transportistas y *Big Data* para generar las capacidades de cómputo en el manejo de grandes cantidades de información, se pueden obtener los insumos necesarios para la detección de patrones y generación de inteligencia con *Machine Learning*; por ejemplo obtener las actualización de datos sobre el tráfico actual o poder pronosticar accidentes en tiempo real generaría ventajas competitivas, esto con el objetivo no sólo de generar una herramienta de optimización, sino, agentes inteligentes que nos lleven a alcanzar el título de Smart City en México.

A continuación, se presentan los trabajos a futuro.

- Adecuar todos los modelos de solución por OR-Tools de Google.
- Desarrollar un heurístico que considere rutas prohibidas, de alta accidentabilidad e inseguras.
- Una propuesta de infraestructura IoT para un clúster de transportistas y la generación de los requerimientos para una ciudad inteligente en el transporte de carga.
- Mejorar la visualización de los resultados en la plataforma, mostrando un mapa de solución.
- Considerar más de un centro de distribución en el CVRP.
- Permitir como primera instancia la definición de rutas prohibidas y zonas de alta accidentabilidad, validando la factibilidad de calcularlo u obtenerlo a través de un servicio web. Como segunda instancia, es desarrollar modelos

- de *Machine Learning* para la detección y pronóstico de rutas prohibidas y zonas inseguras.

Bibliografía

Alinaghian, M., Kaviani, Z., & Khaledan, S. (2015). A novel heuristic algorithm based on Clark and Wright Algorithm for Green Vehicle Routing Problem. *International Journal of Supply and Operations Management*, *2*(2), 784–797.

Benito Quintanilla, A. (2015). *Problemas de Rutas de Vehículos: modelos, aplicaciones logísticas y métodos de resolución*. Universidad de Valladolid. Retrieved from http://uvadoc.uva.es/handle/10324/13287

Chao, L., Lei-shan, Z., Ti-xiang, Y., & Ran, C. (1965). A branch and bound algorithm for the exact solution of three machine scheduling problems. *Mathematical Problems in Engineering*, *16*(1), 89–100.

Fico. (2018). Xpress Optimization. Retrieved from https://www.fico.com/en/products/fico-xpress-optimization

Google Maps Platform. (2018). Directions API. Retrieved from https://developers.google.com/maps/documentation/directions/start

Google OR-Tools. (2018). Google AI. Retrieved from https://developers.google.com/optimization/

Juan, G., Mejía, C., & Arroyo, P. (2017). Análisis de las herramientas para la distribución. Retrieved September 21, 2018, from Análisis de las herramientas para la distribución

Laporte, G., Toth, P., & Vigo, D. (2013). Vehicle routing: historical perspective and recent contributions. *EURO Journal on Transportation and Logistics*, *2*(1–2), 1–4. https://doi.org/10.1007/s13676-013-0020-6

Lum, O., Chen, P., Wang, X., Golden, B., & Wasil, E. (2015). A Heuristic Approach for the Swap-Body Vehicle Routing Problem. *14th INFORMS Computing Society Conference*, *11*(13), 172–187.

Luna López, L. C. (2015). *Localización de paradas y diseño óptimo de rutas para transporte personal*. Universidad Autónoma de Nuevo León.

Orrego Cardozo, J. P. (2013). *Solución al problema de ruteo de vehículos con capacidad limitada "CVRP" a través de la heurística de barrido y la implementación del algoritmo genético de Chu-Beasley*. *Tesis*. Universidad tecnológica de Pereira.

Pérez Porto, J., & Merino, M. (2012). Lenguaje de programación. Retrieved from https://definicion.de/lenguaje-de-programacion/

Presidencia de la República. (2013a). Plan Nacional de Desarrollo 2013-2018. *Gobierno de La República Mexicana*, 183. Retrieved from http://pnd.gob.mx/

Presidencia de la República. (2013b). Programa para un Gobierno Cercano y Moderno. *Diario Oficial De La Federacion, Acuerdo 650*, (30 de agosto de 2013), 53–110. Retrieved from http://www.dgespe.sep.gob.mx/public/normatividad/acuerdos/acuerdo_650.pdf:%5Cn%5Cn

Ramalhinho, H., & Serra, D. (2002). Heurísticas Adaptativas para el Problema de Asignación Generalizada. *Proceedings of The First Spanish Congress in Evolutive and Bioinspired Algorithms*, 267–275.

Reyes Chavez, E., Tamayo Garcia, Y., & Leyva Zaldívar, M. (2011). Procedimiento para el diseño de redes de distribución logística. *Universidad de Holguín "Oscar Lucero Moya" Ave. XX*.

Rocha Medina, L. B., González La Rota, E. C., & Orjuela Castro, J. A. (2011). Una Revisión al Estado del Arte del Problema de Ruteo de Vehículos: Evolución Histórica Y Métodos De Solución. *Ingeniería*, *16*(2), 35–55. Retrieved from http://revistas.udistrital.edu.co/ojs/index.php/reving/article/view/3832

Ruiz Aranda, P. (2013). Arquitectura cliente/servidor. Retrieved from http://somebooks.es/arquitectura-clienteservidor/

The Institute for Operations Research and the Management Sciences. (2018). Informs. Retrieved from https://pubsonline.informs.org/magazine/orms-today

Toth, P., & Vigo, D. (2002). *The Vehicle Routing Problem*. (2002 Society for Industrial and Applied Mathematics, Ed.) (Ilustrada).

Km 12+000 Carretera Estatal 431 “El Colorado-Galindo”
Parque Tecnológico San Fandila
Mpio. Pedro Escobedo, Querétaro, México
CP 76703
Tel +52 (442) 216 9777 ext. 2610
Fax +52 (442) 216 9671

publicaciones@imt.mx

http://www.imt.mx/

Esta publicación fue desarrollada en el marco de un sistema de gestión de calidad certificada bajo la norma ISO 9001:2015