UNIVERSIDAD NACIONAL DEL SANTA

ESCUELA DE POSGRADO

Programa de Doctorado en Ingeniería de Sistemas e Informática

“Aplicación de la mineria de datos especiales basada en tecnicas de agrupamiento al congestionamiento del tráfico vehicular en la Ciudad de Trujillo, Perú”

Tesis para optar el grado académico de Doctor en Ingeniería de Sistemas e Informática

Autor:

Mg. Díaz Pulido, José Arturo

Asesor:

Dr. Aguilar Marín, Pablo
DNI. N° 18071385
Código ORCID: 0000-0001-6096-4010

Linea de investigación

Inteligencia Artificial

Nuevo Chimbote - PERÚ
2023

UNS
ESCUELA DE POSGRADO

CONSTANCIA DE ASESORAMIENTO DE TESIS

Yo, **Aguilar Marín, Pablo,** mediante la presente certifico mi asesoramiento de la Tesis Doctoral titulada: **Aplicación de la minería de datos espaciales basada en técnicas de agrupamiento al congestionamiento del tráfico vehicular en la Ciudad de Trujillo, Perú,** elaborada por el (la) magister **Diaz Pulido, José Arturo,** para obtener el Grado Académico de Doctor en **Ingeniería de Sistemas e Informática** en la Escuela de Posgrado de la Universidad Nacional del Santa.

Nuevo Chimbote, del 2023

.....

Dr. Aguilar Marín, Pablo
ASESOR
CODIGO ORCID: 0000-0001-6096-4010
DNI N°18071385

UNS
ESCUELA DE POSGRADO

CONFORMIDAD DEL JURADO EVALUADOR

Aplicación de la minería de datos espaciales basada en técnicas de agrupamiento al congestionamiento del tráfico vehicular en la Ciudad de Trujillo, Perú,

Tesis para optar el grado de
Doctor en Ingeniería de Sistemas e Informática

Revisado y Aprobado por el Jurado: Evaluador:

.....

Dr. Vega Moreno, Carlos Eugenio
PRESIDENTE
CODIGO ORCID: 0000-0003-2955-0674
DNI N°: 32937583

.....

Dra. Briones Pereyra, Lizbeth Dora
SECRETARIA
CODIGO ORCID: 0000-0003-0626-7227
DNI N°: 32960646

.....

Dr. Aguilar Marín, Pablo
VOCAL
CODIGO ORCID: 0000-0001-6096-4010
DNI N°: 18071385

UNS
ESCUELA DE POSGRADO

ACTA DE EVALUACIÓN DE SUSTENTACIÓN DE TESIS

A los dieciocho días del mes de setiembre del año 2023, siendo las 16:03 horas, en el aula multimedia N° 91 de la Escuela de Posgrado de la Universidad Nacional del Santa, se reunieron los miembros del Jurado Evaluador conformado por los docentes: Dr. Carlos Eugenio Vega Moreno (Presidente), Dra. Lisbeth Dora Briones Pereyra (Secretaria), Dr. Pablo Aguilar Marín (Vocal) ; designados mediante Resolución Directoral Nº 129-2023-EPG-UNS de fecha 07.06.2023, con la finalidad de evaluar la tesis titulada: "***APLICACIÓN DE LA MINERIA DE DATOS ESPACIALES BASADA EN TECNICAS DE AGRUPAMIENTO AL CONGESTIONAMIENTO DEL TRÁFICO VEHICULAR EN LA CIUDAD DE TRUJILLO, PERÚ***"; presentado por el tesista **Ms. José Arturo Díaz Pulido,** egresado del programa de **Doctorado en Ingeniería de Sistemas e Informática.**

Sustentación autorizada mediante Resolución Directoral Nº 218-2023-EPG-UNS de fecha 12 de setiembre de 2023.

El presidente del jurado autorizó el inicio del acto académico; producido y concluido el acto de sustentación de tesis, los miembros del jurado procedieron a la evaluación respectiva, haciendo una serie de preguntas y recomendaciones al tesista, quien dio respuestas a las interrogantes y observaciones.

El jurado después de deliberar sobre aspectos relacionados con el trabajo, contenido y sustentación del mismo y con las sugerencias pertinentes, declara la sustentación como: APROBADO asignándole la calificación de: BUENO (18).

Siendo las 17:20 horas del mismo día se da por finalizado el acto académico, firmando la presente acta en señal de conformidad.

Dr. Carlos Eugenio Vega Moreno
Presidente

Dra. Lizbeth Dora Briones Pereyra
Secretaria

Dr. Pablo Aguilar Marín
Vocal

turnitin

Recibo digital

Este recibo confirma quesu trabajo ha sido recibido por Turnitin. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega:	Jose Arturo Diaz Pulido
Título del ejercicio:	Revision Informe de Tesis Doctoral
Título de la entrega:	Revision Tesis Doctoral
Nombre del archivo:	Tesis_UNS_jadp.docx
Tamaño del archivo:	28.41M
Total páginas:	140
Total de palabras:	25,562
Total de caracteres:	146,395
Fecha de entrega:	25-oct.-2023 04:37a. m. (UTC-0500)
Identificador de la entre...	2206737510

UNS
POSGRADO

APLICACIÓN DE LA MINERIA DE DATOS ESPACIALES BASADA EN TECNICAS DE AGRUPAMIENTO AL CONGESTIONAMIENTO DEL TRÁFICO VEHICULAR EN LA CIUDAD DE TRUJILLO, PERÚ

Tesis para optar el grado de
Doctor en Ingeniería de Sistemas e Informática

Autor:
Mg. DIAZ PULIDO, JOSÉ ARTURO

Asesor:
Dr. PABLO AGUILAR MARÍN
DNI N°: 18071369
Código ORCID: 0000-0001-6090-4010

Línea de investigación
Inteligencia Artificial

NUEVO CHIMBOTE - PERÚ
2023

Revision Tesis Doctoral

INFORME DE ORIGINALIDAD

24%	24%	5%	%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	www.socialtechmonkeys.com Fuente de Internet	1 %
2	machinelearningparatodos.com Fuente de Internet	1 %
3	sites.google.com Fuente de Internet	1 %
4	repositorio.usmp.edu.pe Fuente de Internet	1 %
5	repositorio.ulima.edu.pe Fuente de Internet	1 %
6	rcs.cic.ipn.mx Fuente de Internet	1 %
7	inaoe.repositorioinstitucional.mx Fuente de Internet	1 %
8	cybertesis.unmsm.edu.pe Fuente de Internet	1 %
9	repositorio.upn.edu.pe Fuente de Internet	1 %

DEDICATORIA

A mis seres amados que reposan en la eternidad:
A mi célebre madre, doña María Pulido Soles, muestra de coraje, amor y valores para luchar.
A mi padre, don Marcelo Diaz Pulido, fuente de apoyo inagotable.
A mis recordados hermanos: Luciano, Josefa y ***Luis Fernando****.*

A las bendiciones más grandes que Dios me otorgo en esta vida, mis amados hijos: Sebástian y Joaquín.

A mi esposa Erika Quiroz Sánchez, por ser siempre mi cómplice en mis proyectos.

AGRADECIMIENTOS

Con especial aprecio a mi asesor el Dr. Pablo Aguilar Marín, que a través de su experiencia supo indicarme los parámetros exactos para culminar esta investigación.

A mis jurados por sus necesarias apreciaciones que conllevaron a la culminación de esta investigación.

INDICE GENERAL

Índice de Tablas

Índice de Figuras

RESUMEN

El presente estudio se basa en el análisis del tráfico vehicular en la red vial de la ciudad de Trujillo, con el propósito de detectar y/o diagnosticar el nivel de congestión en diversos puntos críticos con mayor afluencia de vehículos. Se empleó un diseño no experimental descriptivo, donde la técnica utilizada en la recolección de datos fue la observación directa, con fichas de registro alcanzadas por la oficina metropolitana de transporte de la municipalidad de Trujillo; realizándose con esto, el proceso de análisis de datos simulados con valores numéricos válidos y aleatorios en tiempo real, por medio de la construcción de un modelo computacional de aprendizaje no supervisado, empleando la metodología CRISP-DM, específica para gestión y análisis de minería de datos espaciales.

Para atender la simulación de la congestión de los diferentes puntos críticos se implementó un sistema de software, para determinar la clasificación de la congestión con los algoritmos k-means y dbscan;

Se hizo la comparación de los algoritmos de agrupamiento k-means y dbscan para determinar la fiabilidad y la tendencia de organización de grupos o clusters validando de esta manera el modelo computacional, para lo cual se consideró las técnicas del acodamiento y del promedio Silhouette respectivamente.

Con los resultados obtenidos desde el sistema de software implementado se logró clasificar diversos puntos críticos congestionados y la densidad del tráfico simulados en tiempo real.

Palabras Claves: densidad del tráfico, minería de datos espaciales, clustering, k-means, dbscan.

ABSTRACT

The present study is based on the analysis of vehicular traffic in the road network of the city of Trujillo, with the purpose of detecting and/or diagnosing the level of congestion at various critical points with the greatest influx of vehicles. A descriptive non-experimental design was employed, where the technique used in data collection was direct observation, with record cards obtained from the metropolitan transportation office of the municipality of Trujillo; thus, the process of analysis of simulated data with valid and random numerical values in real time was carried out, through the construction of an unsupervised learning computational model, using the CRISP-DM methodology, specific for spatial data mining management and analysis.

In order to simulate the congestion of the different critical points, a software system was implemented to determine the congestion classification with the k-means and dbscan algorithms;

A comparison of the k-means and dbscan clustering algorithms was made to determine the reliability and the organization tendency of groups or clusters, thus validating the computational model, for which the kink and Silhouette average techniques were considered, respectively.

With the results obtained from the implemented software system, it was possible to classify several congested critical points and the traffic density simulated in real time.

Keywords: traffic density, spatial data mining, clustering, k-means, dbscan.

1. INTRODUCCIÓN

La congestión vehicular es un desafío importante en el área de planificación del transporte vehicular en las redes viales de las diferentes ciudades del mundo, tornándose complejo cuando se produce la congestión vehicular.

La congestión vehicular generalmente se relaciona con un exceso de vehículos en una parte de la carretera en un momento determinado lo que resulta en velocidades más lentas, a veces mucho más lentas que las velocidades normales o de flujo libre. El tiempo de viaje y la densidad del tráfico son las medidas de tráfico más utilizadas para cuantificar la congestión en las carreteras.

La densidad del tráfico se define como el número de vehículos que ocupan una longitud determinada de la carretera. La densidad del tráfico se considera como la medida principal para cuantificar la congestión de carreteras que no sean intersecciones señalizadas. Como la densidad es difícil de medir, generalmente se adoptan métodos indirectos para estimar la densidad a partir de otros parámetros, como el flujo, la velocidad o la ocupación (UPCommons, 2000).

La congestión vehicular es un problema que afecta a muchas ciudades del mundo, generando pérdidas económicas, sociales y medioambientales. Para tener una proporción de lo que implican estas pérdidas, a modo de ejemplo, a Buenos Aires y la Ciudad de México la congestión les cuesta 2 y 2,3 veces lo que el gobierno local invierte en educación. La inversión de Sao Paulo en salud equivale a lo que le cuesta la congestión.

El aumento de la tasa de la urbanización y la ausencia de un plan eficiente del uso del suelo, han generado desafíos importantes para la movilidad urbana.

Otra determinante es la infraestructura vial y la asignación de prioridades en su uso que ha favorecido el transporte individual. Esto repercute en el incremento de la tasa de motorización y la reducción en el uso del transporte público. La tasa de crecimiento por cada 1.000 habitantes fue 4,7% en Latinoamérica los últimos 10 años (BID, 2019)

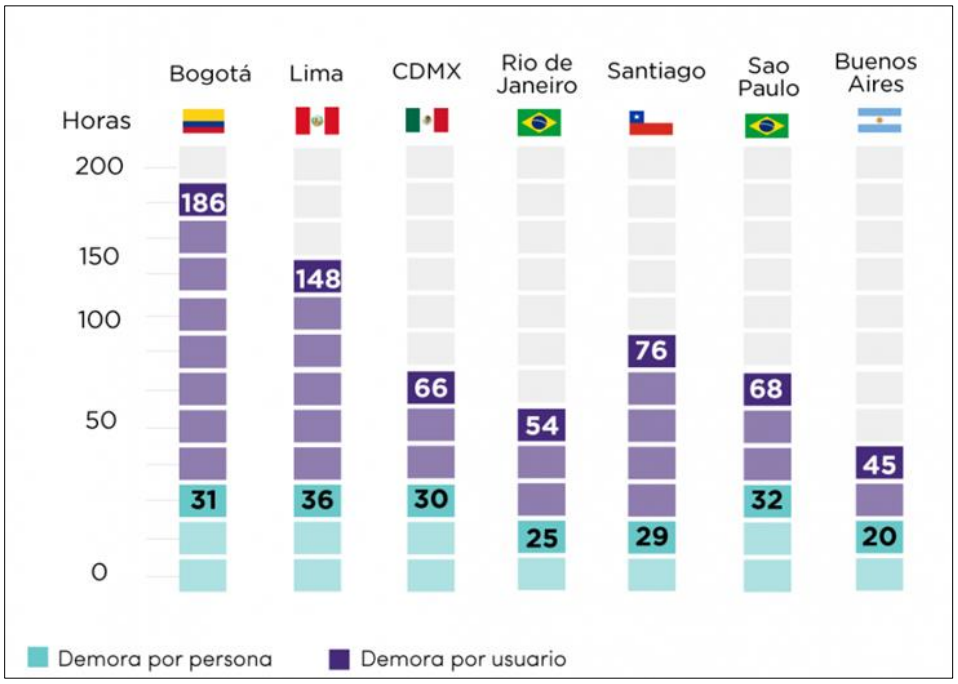

Figura 1. Demora por persona y usuario de vehículo particular en ciudades seleccionadas de Latinoamérica.

Según el Ranking de Congestión de Tráfico - Informe Anual 2023, indica que ciudades como Bogotá, Lima, Ciudad de México y Río de Janeiro se encuentran entre las más congestionadas del mundo, descrito en el portal web https://trafficindex.org/reports/annual-report-2023/. Tal como lo señala la figura 1, Lima ha sido catalogada como la ciudad con mayor congestión vehicular de América Latina y la octava de todo el mundo.

En varias ciudades de nuestro país, se presenta congestión vehicular por la informalidad y falta de cultura vial, a nivel de peatones, transporte público y privado, paraderos informales que se ubican en lugares públicos y privados, como: centros comerciales, colegios, hospitales, etc.

En esta tesis se consideró la información existente acerca de los 66 puntos críticos de rutas detectadas por la dirección metropolitana de Transporte de la ciudad de Trujillo. Luego, se seleccionaron y procesaron las fuentes de datos para el estudio de forma automática, almacenando las variables más relevantes, tales como: la velocidad que disminuye y el tiempo que aumenta en determinados momentos, cuando los vehículos se trasladan por las diferentes rutas de la ciudad de Trujillo. Las técnicas de “clustering” para obtener agrupamientos de puntos críticos o rutas congestionadas cuyas características fueran similares, se determinaron por valor de la medida de la velocidad.

Para llegar a resultados concretos se diseñó la base de datos espacial, partiendo de la abstracción de la información real de la problemática que se observa a diario en diferentes lugares o vías de la red vial de la ciudad de Trujillo, logrando con esto, la implementación un sistema de información basado en lenguaje de programación Python para simular el control y gestión de flujo del transporte vehicular en la ciudad de Trujillo, con datos extraídos de la información pública acerca de la congestión vehicular de la ciudad de Trujillo brindados por la oficina de TMT (Transporte Metropolitano de Trujillo) (Anexo A y Anexo D). Se uso el software aplicativo RapidMiner para comprobar el modelo de caracterización de patrones de tráfico vehicular a nivel de k-means y dbscan.

Se aplicó Minería de datos espaciales basada en técnicas de agrupamiento o clustering. Determinando la implementación de los algoritmos k-means y dbscan respectivamente; con el algoritmo k-means se logró obtener la mayor precisión de agrupamientos y con el algoritmo dbscan se consiguió tendencias de densidad en los diferentes puntos críticos simulados.

Esta tesis contiene 5 capítulos:

Capítulo I: en este capítulo se aborda el planteamiento y fundamentación del problema de investigación, describiendo el objeto de la investigación, la realidad genérica y sus características específicas del problema, formulación del problema, antecedentes de la investigación, delimitación del estudio, Justificación e importancia de la investigación y Objetivos de la investigación.

Capítulo II: en este capítulo se aborda los fundamentos teóricos de la investigación y el marco conceptual.

Capítulo III: en este capítulo se aborda la Hipótesis central de la investigación, variables e indicadores, métodos, diseño o esquema de la investigación, población y muestra, actividades del proceso investigativo, técnicas e instrumentos de la investigación, procedimiento para la recolección de datos, las técnicas de procesamiento y análisis de los datos.

Capítulo IV: en este capítulo se aborda los resultados y la discusión de Resultados.

Capítulo V: en este capítulo se aborda las conclusiones y las recomendaciones.

CAPÍTULO I

PROBLEMA DE INVESTIGACIÓN

1.1. Planteamiento y fundamentación del problema de investigación

1.1.1. Objeto de la investigación

Estuvo conformado por el análisis de la congestión del tráfico vehicular que ocurre sobre los diferentes nodos como rutas, calles, avenidas, jirones, óvalos, intersecciones viales, etc. de la red vial, que compromete indicadores como: tiempos, fechas, densidad vehicular, paraderos informales, semaforización, policías de tránsitos de la ciudad de Trujillo, Perú.

1.1.2. Realidad genérica del problema

El aumento del volumen de la población y de los vehículos motorizados causa congestión en las ciudades. La congestión del tráfico vehicular (exceso de vehículos a bajas velocidades en tramos de carreteras o calles en tiempos determinados) es uno de los mayores retos en el campo de la planificación del transporte urbano, así como en el manejo (management) del tráfico. La congestión conduce a efectos negativos como polución, pérdida de tiempo, dinero, combustible y otros. Para estimar la densidad del tráfico se suele recurrir a información sobre flujo, velocidad y ocupación de espacios por los vehículos. Por otro lado, vivimos en un mundo impulsado por los datos.

Los avances en la tecnología de generación, reunión y almacenamiento de datos han empoderado a las organizaciones para agrupar enormes cantidades de datos.

En países como Perú, las grandes bases de datos almacenados no se aprovechan para hacer estimaciones y pronósticos de la densidad del tráfico y problemas relacionados. En el país priman los aspectos políticos antes que soluciones técnicas. Como resultado del procesamiento adecuado de datos algunas medidas simples que se pueden explorar son: hacer respetar los paraderos por los peatones y los buses, cumplir normas de tránsito, enseñar respeto y etiqueta, habilitar paraderos segregados y planificados, contar con semaforización y señalética inteligente y adecuada, promover el transporte intermodal, evitando mayores números de carriles o pases a desnivel.

1.1.3. Características de la realidad específica

Actualmente en la ciudad de Trujillo, algunas de sus vías de tránsito vehicular e intersecciones de las mismas (figura 2), sufren de congestionamiento esto se debe a diversos factores como:

- Que estas vías concentran en sus entornos entidades referentes a actividades de diversa índole comercial de asidua concurrencia por el público consumidor como: bancos, instituciones educativas, entidades públicas como hospitales y otros.
- Sistema de transporte publico desordenado e informal, colectivos con paraderos indeterminados y no fiscalizados.
- Según reportes en la actualidad se estima una capacidad óptima de una vía urbana es considerada en 1,800 vehículos/hora/carril.
- Según el estudio realizado de vías saturadas, por el área de Transporte Metropolitano de Trujillo, se detectó que existen 66 puntos críticos de congestión vehicular en esta ciudad liberteña; siendo los hospitales, mercados e instituciones las zonas de más afluencia de transporte público.

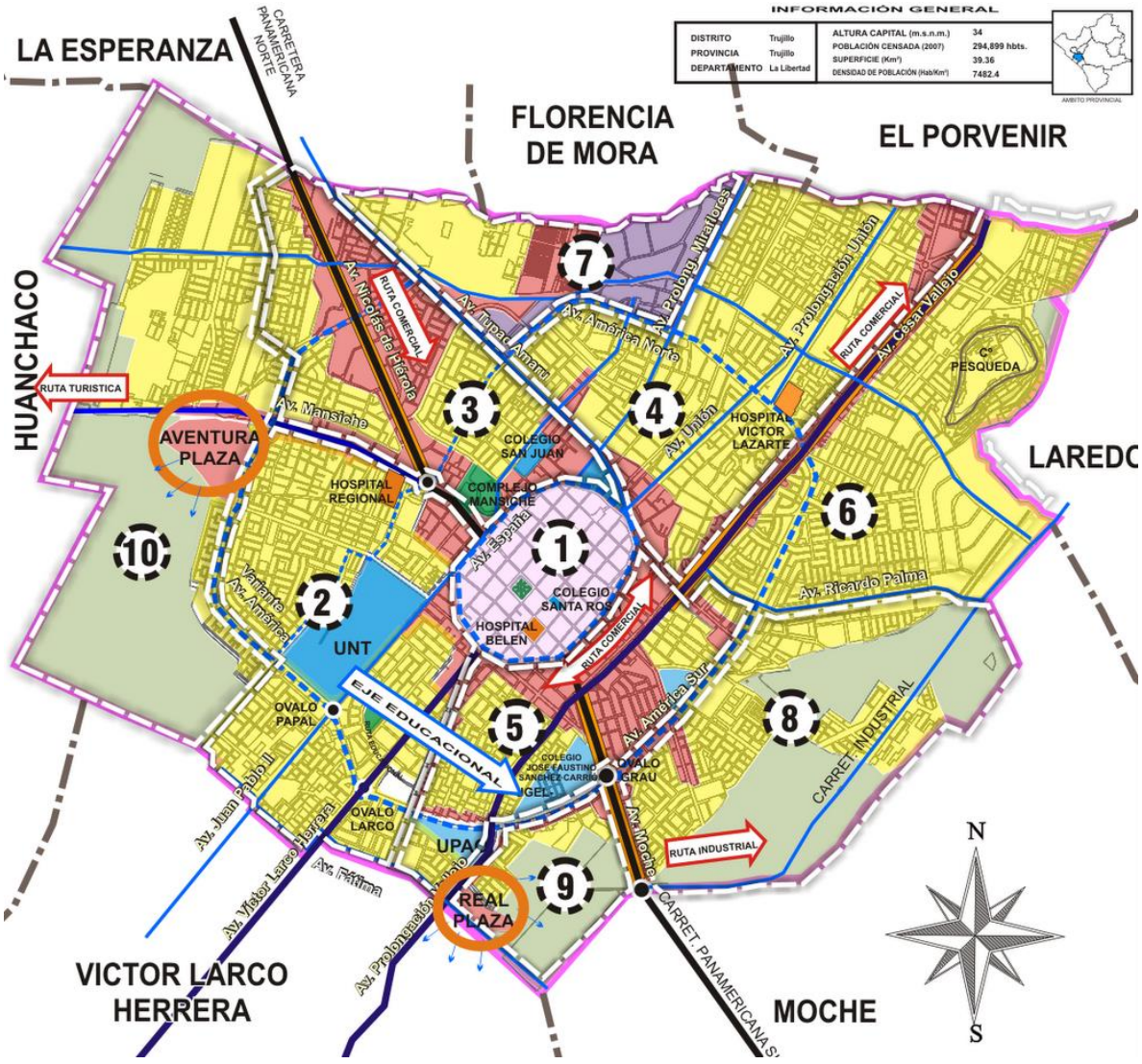

Figura 2. Estructura urbana de la provincia de Trujillo. Fuente: Escuela de Arquitectura, UPAO 2018.

1.2. Antecedentes de la investigación

La congestión del tráfico vehicular es un desafío importante en la gestion y control del transporte vehicular, se relaciona con un exceso de vehículos que sobrepasa la capacidad de la cantidad permitida en un momento determinado, lo que resulta el aletargo de las velocidades, y por ende el tiempo de recorrido aumenta mas en cada uno de los nodos, que hay en un flujo libre. La densidad del tráfico son las medidas de tráfico más utilizadas para cuantificar la congestión en los diferentes nodos de la red vial de la ciudad de Trujillo. El tiempo de viaje se define como el tiempo requerido para que los usuarios viajen de un nodo a otro. La densidad del tráfico se define como el almacenamiento anormal del número de vehículos que ocupan en un nodo. La densidad del tráfico se considera como la medida principal para cuantificar la congestión de nodos que no sean intersecciones señalizadas. Como la densidad es difícil de medir, generalmente se adoptan métodos indirectos para estimar la densidad a partir de parámetros, como el flujo, la velocidad, el tiempo, las horas punta y la semaforizacion.

Según Guerrero (2014), plantea un Modelo computacional de minería de datos, para un Sistema de Información Geográfico de monitoreo de vehículos, que permita la predicción de eventos peligrosos. Sostiene que, durante los últimos años, el auge de las Ciencias de la Información Geográfica, ha ido en aumento de una manera considerable, debido a diversos avances tecnológicos, tanto en Hardware, como en Software. Con la implementación de estos avances en el sector transporte, surgen varías inquietudes. Entre ellas, la manera en que el uso de estas tecnologías, permitan predecir y prevenir imprevistos, riesgos y demás factores

que afecten la operación de vehículos en la vía. En este trabajo, se propone un modelo computacional, que pueda ser instanciado en un sistema de información geográfico de monitoreo de vehículos, que capture información de eventos de riesgo y de esta manera, poder predecirlos, aplicando técnicas de minería de datos (algoritmos de clasificación) a partir de una muestra de datos. Palabras clave: SIG, Redes Bayesianas, Redes Neuronales Artificiales, Arboles de Decisión, clasificación, riesgos en vehículos, GPS, ciencias de la información geográfica.

Según Kumar (2018), en su invetigacion denominada Estudio Comparativo de Algoritmos de Clasificación de Minería de Datos para Predecir la Gravedad de los Accidentes de Tránsito, argumentan que, según el informe de la Organización Mundial de la Salud, el número de muertes por accidentes de tráfico es de más de 1,25 millones de personas y cada año los accidentes no mortales afectan a más de 20 a 50 millones de personas. Varios factores contribuyen a la ocurrencia de un accidente de tránsito. En este estudio, se aplicaron técnicas de clasificación de minería de datos para establecer modelos (clasificadores) para identificar factores de accidentes y predecir la gravedad de los accidentes de tránsito utilizando datos de tránsito previamente registrados. Utilizando el árbol de decisión de minería de datos WEKA (Entorno de Waikato para el análisis del conocimiento) (J48, ID3 y CART) y los clasificadores Naïve Bayes se construyen para modelar la gravedad de la lesión. El rendimiento de clasificación de todos estos algoritmos se compara en función de sus resultados. El resultado experimental muestra que la precisión del clasificador J48 es mayor que la de otros.

Según Peña (2017), en su investigación denominada Modelo para la Caracterización del Delito en la Ciudad de Bogotá, Aplicando Técnicas de Minería de Datos Espaciales, argumenta que la seguridad ciudadana y el control del crimen se encuentran entre las mayores preocupaciones sociales no solo en Bogotá sino en todo el país. Se puede lograr una disminución en el índice de delincuencia mediante el uso de herramientas para describir el comportamiento delictivo. La minería de datos espaciales se utiliza para extraer conocimiento. Sus métodos pueden ser utilizados para explorar, descubrir relaciones entre datos espaciales y no espaciales, reorganizar datos espaciales en bases de datos y determinar sus características generales de manera simple. Hay muchos métodos diferentes para el

descubrimiento de datos espaciales, como: el método basado en la generalización, el método de reconocimiento de patrones, el método de uso de agrupamiento y el método de detección de correlación espacial. Mediante la aplicación de técnicas de minería de datos espaciales, se pretende caracterizar el comportamiento de los infractores patrimoniales que afectan a la ciudad de Bogotá. El objetivo de este trabajo fue crear un modelo característico del comportamiento delictivo para una zona de la ciudad metropolitana de Bogotá, mediante la aplicación de técnicas de agregación de datos espaciales. Para ello se trabajó con información obtenida de diversas entidades como: Infraestructura de Datos Espaciales del Área Metropolitana (IDECA), Cámara de Comercio de Bogotá, portales web de entidades oficiales como el Ayuntamiento de Bogotá, Policía Metropolitana de Bogotá y otros.

Según Padilla (2019), en su investigación Reducción de la dimensión de registros de evaluaciones académicas aplicando el algoritmo K-means. Sostiene que, en un ambiente educativo existe una gran cantidad de datos que pue-den ser analizados y utilizados en el proceso de la toma de decisiones. En la actualidad, debido al tamaño de su dimensión, los datos tienden a ser más complejos que los datos convencionales y requieren una reducción de su dimensión. La Minería de Datos Educacional permite utilizar técnicas de Minería de Datos para analizar información académica con el fin de identificar patrones que no son evidentes. Este artículo presenta resultados obtenidos en una investigación de un caso de estudio sobre el rendimiento académico de alumnos de Ingeniería del Centro Universitario UAEM Valle de México. En el análisis de los datos se utiliza el algoritmo K-means, el software WEKA y R Studio. Se propone utilizar el agrupamiento para reducir la dimensión de las variables académicas en función de los registros de las calificaciones obtenidas durante los últimos períodos cursados, luego se trabajará con una medida promedio para predecir el rendimiento académico de un alumno. Se utiliza R Studio para contrastar los grupos obtenidos por WEKA. Palabras clave: minería de datos educacional, reducción de la dimensión, agrupamientos, K-means.

Según Garrido (2017), analizó la organización espacial de la red de carreteras de Aragón, España, mediante la aplicación metodológica de la teoría de grafos. Utilizó las medidas de conectividad, accesibilidad y centralidad. El estudio se complementa con la integración de

nodos exteriores al grafo de Aragón y la correlación de la red de carreteras y el desarrollo socioeconómico.

Según Gonzáles (2013), en su investigación denominada La minería de datos espaciales y su aplicación en los estudios de salud y epidemiología, sostiene que la acumulación de información espacial producto del desarrollo de los sistemas informáticos, y en especial de los sistemas de información geográfica, propicia la aplicación de técnicas de minería de datos espaciales para la extracción de nuevos conocimientos que asistan a la toma de decisiones. Las áreas de salud y epidemiologia no han estado ajenas al desarrollo y utilización de estos sistemas; han revalorizado la importancia de la componente espacial en sus investigaciones y en el diseño de estrategias diferenciadas de prevención y control por área de salud. En este trabajo se describen los aspectos metodológicos y los conceptos asociados a la minería de datos espaciales. Se describen los principales algoritmos y herramientas existentes para la minería de datos espaciales y se muestran algunos trabajos, tendencias de su aplicación y potencialidades en las áreas de salud y epidemiología. Palabras clave: análisis espacial, epidemiología, salud, sistemas de información geográfica.

Según Gómez (2009), en su investigación denominada Desarrollo de un Modelo de Simulación Vehicular para la Mejora en la Sincronización de Semáforos. Sostiene que, el desarrollo de un simulador de flujo vehicular que incorpora un método para realizar la sincronización de un circuito de calles. El método de sincronización desarrollado en este trabajo se realiza por medio de varias teorías como son: teoría de colas, ecuaciones diferenciales parciales, teorías de ondas, las ondas de Shock; todas estas teorías conjuntadas proporcionan el tiempo de duración que debe de tener cada ciclo de los semáforos que conforman el circuito de calles. Por medio del concepto de offset se calculan los tiempos que deben transcurrir entre el inicio del ciclo verde de un semáforo y del semáforo que le sigue.

El producto de este trabajo ayuda en la toma de decisiones para aliviar la congestión de tráfico, reducir accidentes, disminuir contaminación, entre otros. Afirma que, se logró obtener un simulador de flujo vehicular, el cual puede representar la realidad con una diferencia menor a un auto, así como el hecho de que puede ser utilizado en computadoras con bajos recursos.

1.3. Formulación del problema de investigación

El examen de la situación actual del tráfico vehicular en la ciudad de Trujillo, Perú, revela que no se cuenta con un instrumento metodológico que permita evaluar los distintos sectores urbanos que requieran intervención. Esto nos permite formular la siguiente pregunta:

¿En qué medida la implementación de un modelo de caracterización mediante técnicas de minería de datos mejorará la eficiencia de la gestión y control del flujo vehicular en las diferentes vías de tránsito del centro histórico de la ciudad de Trujillo?

1.4. Delimitación del estudio

Esta investigación se llevó a cabo en la ciudad de Trujillo, y se tomaron datos referenciados geográficamente por la dirección de Transporte Metropolitano de la ciudad de Trujillo, La Libertad. El desarrollo de esta propuesta se llevó a cabo en el año 2021. Se considero las ordenanzas municipales que delimitan normas para el transporte vehicular de la ciudad de Trujillo. Se implementó un sistema de software basado en 2 técnicas de agrupamiento representadas por los algoritmos k-means y dbscan y usa aprendizaje no supervisado de tipo descriptivo partiendo de agrupaciones de clusters.

1.5. Justificación e importancia de la investigación

La minería de datos espacial puede describir los objetos espaciales que la forman a través de tres características básicas: atributos, localización y topología; en tiempo real. La información espacial ha estado asociada en forma directa con la cartografía, en el logro de objetivos específicos concernientes con operaciones de análisis y gestión de datos espaciales, en las cuales se representa dicha información con modelos que usan mapas y símbolos.

Por lo que, desde el punto de vista tecnológico, la minería de datos espacial permite extraer datos en tiempo real, como: datos y atributos geográficos, metadatos, métodos de búsqueda, de visualización y mecanismos para proporcionar acceso a los datos espaciales georreferenciados, involucrando para esto, técnicas de aprendizaje supervisado, no supervisado y reforzado de minería de datos, a través de algoritmos de agrupamiento obteniendo resultados óptimos de clasificación, diagnóstico, predicciones entre otros, para planificar nuevas estrategias y tomar decisiones (Cangrejo Aljure et al., 2012). Tal como indica la figura 3.

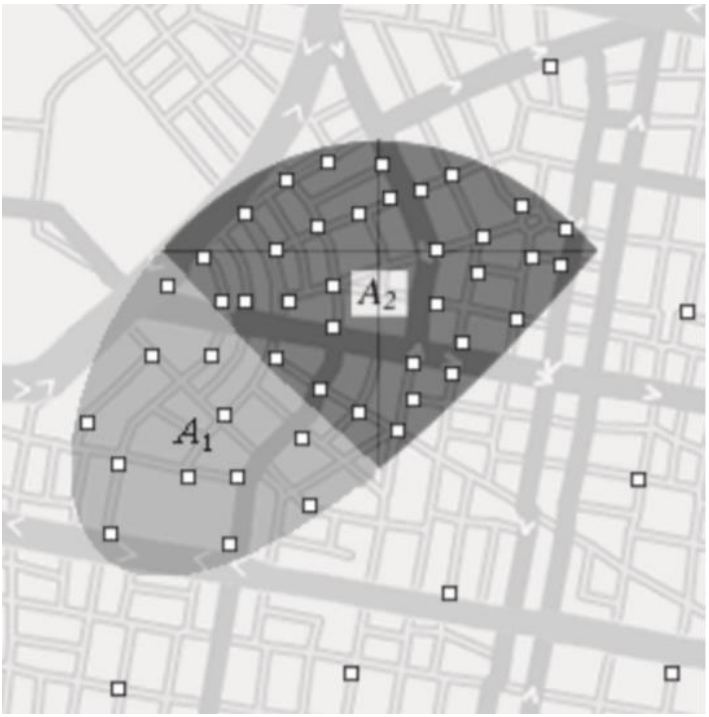

Figura 3. Incidencia de eventos y áreas de afectación (Cangrejo Aljure & Agudelo, 2011)

1.6. Objetivos de la investigación

1.6.1. Objetivo General

Determinar el nivel de congestión del tráfico vehicular a través de técnicas de agrupamiento en los diferentes nodos de la ciudad de Trujillo.

1.6.2. Objetivos específicos

- Desarrollar un sistema de software simulador para determinar el nivel del congestionamiento, aplicando técnicas de agrupamiento con los algoritmos kmeans y dbscan.
- Evaluar los algoritmos k-means y dbscan para clustering.

CAPÍTULO II

MARCO TEÓRICO

2.1. Fundamentos teóricos de la investigación

a. Congestión vehicular

El diccionario de la Lengua Española (Real Academia Española, 2017) define a la congestión como la "acción y efecto de congestionar o congestionarse" y congestionar significa "obstruir o entorpecer el paso, la circulación o el movimiento de algo".

Los autores Bull & Thomson (2002) definen la congestión vehicular como: "la condición en que existen muchos vehículos circulando y cada uno de ellos avanza lenta e irregularmente" (p.110). Ellos indican que tanto su definición como la del diccionario son de carácter subjetivo y no conllevan a una precisión suficiente. Hay que recalcar que es claro que ellos la definen de la manera en que las personas observan el tráfico vehicular.

Se puede llegar a una definición uniendo los significados que nos brindan estos autores, por lo cual, la congestión vehicular es considerada como una condición en la cual el flujo de vehículos en las vías es muy denso o está saturado, esto se debe principalmente, al exceso de demanda de las vías y por el número de vehículos en ellas. Todo esto, genera que los autos no puedan circular con fluidez por las diferentes vías de la ciudad.

El resultado de un embotellamiento es un accidente, aunque los vehículos no pueden ir a gran velocidad porque el conductor pierde los estribos al estar mucho tiempo parado en la calzada. Por otro lado, también conduce a la ira en la carretera, reduce la gravedad de los accidentes causados por vehículos que no circulan a velocidades apreciables, lo que provoca daños o lesiones más graves. Además, los vehículos desperdician combustible innecesariamente porque, según Iturra (2008): En general, entendemos la congestión vehicular como un exceso de vehículos en la vía, lo que conduce a un movimiento lento y errático de cada vehículo respecto a las condiciones normales de operación. Técnicamente, se puede decir que la congestión vehicular ocurre cuando los vehículos en la vía interfieren con el movimiento normal de otros vehículos, es decir, cuando se excede un cierto nivel de concentración y los vehículos comienzan a circular a una velocidad inferior a la de flujo libre. Sin embargo, lo anterior puede no coincidir con la importancia de la congestión del tráfico, porque para niveles ligeramente por encima de la concentración crítica, el retraso que hace un vehículo adicional a la carretera por cada otro vehículo es débil y podemos decir que la carretera sigue funcionando en condiciones normales. condiciones.

Por lo tanto, se puede concluir que cualquier definición de congestión vehicular debe incluir aspectos medibles o computables y las percepciones de los usuarios de la vía sobre el problema, que se pueden definir como: el tiempo que tarda un vehículo adicional. (Iturra, 2008)

Factores de la congestión vehicular

Cultura vial

La cultura del tráfico, desde un punto de vista antropológico, es la forma en que las personas viven, sienten, piensan y actúan en, desde y para los espacios cotidianos en movimiento y en movimiento. Contrariamente al concepto determinista, desde una perspectiva antropológica, es erróneo afirmar que una población, una comunidad o una sociedad "carece" de cultura vial. Entonces:

- Todas las comunidades y sociedades tienen diferentes formas de vivir, sentir, pensar y actuar en espacios de movimiento.
- La cultura vial es una expresión de cómo los miembros de una comunidad o sociedad interactúan en la vía.
- La cultura de la sociedad o la sociedad en sí misma no es mala ni buena, simplemente existe y existe.

Lo correcto es hablar de culturas en el camino, teniendo en cuenta las diferentes sociedades y comunidades. Esas maneras de relacionarse con y en las vías pueden aunarse a factores espaciales, pedagógicos, tecnológicos y mediáticos, para conformar un verdadero sistema de prevención de accidentes de tránsito y protección de la vida. O también pueden fomentar, propiciar o permitir que los accidentes ocurran (Figura 3). Sin embargo, la cultura de la carretera tiene la capacidad de modelar y moldear, facilitando la distribución del territorio, la circulación, el ritmo, el flujo de peatones y vehículos para proteger la salud y la vida. Además de crear entornos que se adapten al tráfico para reducir riesgos y permitir la diversión de los desplazamientos. (Camacho & Cabrera, 2009)

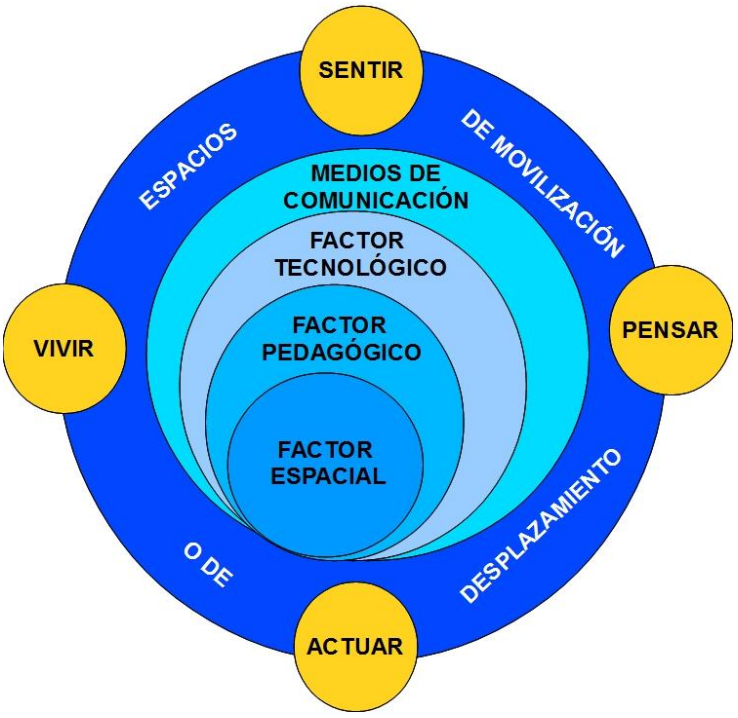

Figura 4. Cultura Vial. Tomado de: (Camacho-Cabrera, 2009)

Posibles causas de la congestión vehicular en La Libertad

Según el último censo del 2007 realizado por el Instituto Nacional de Estadísticas e Informática (INEI, 2007) la población de La Libertad a aumentado a 1’617,050 habitantes (Figura 5).

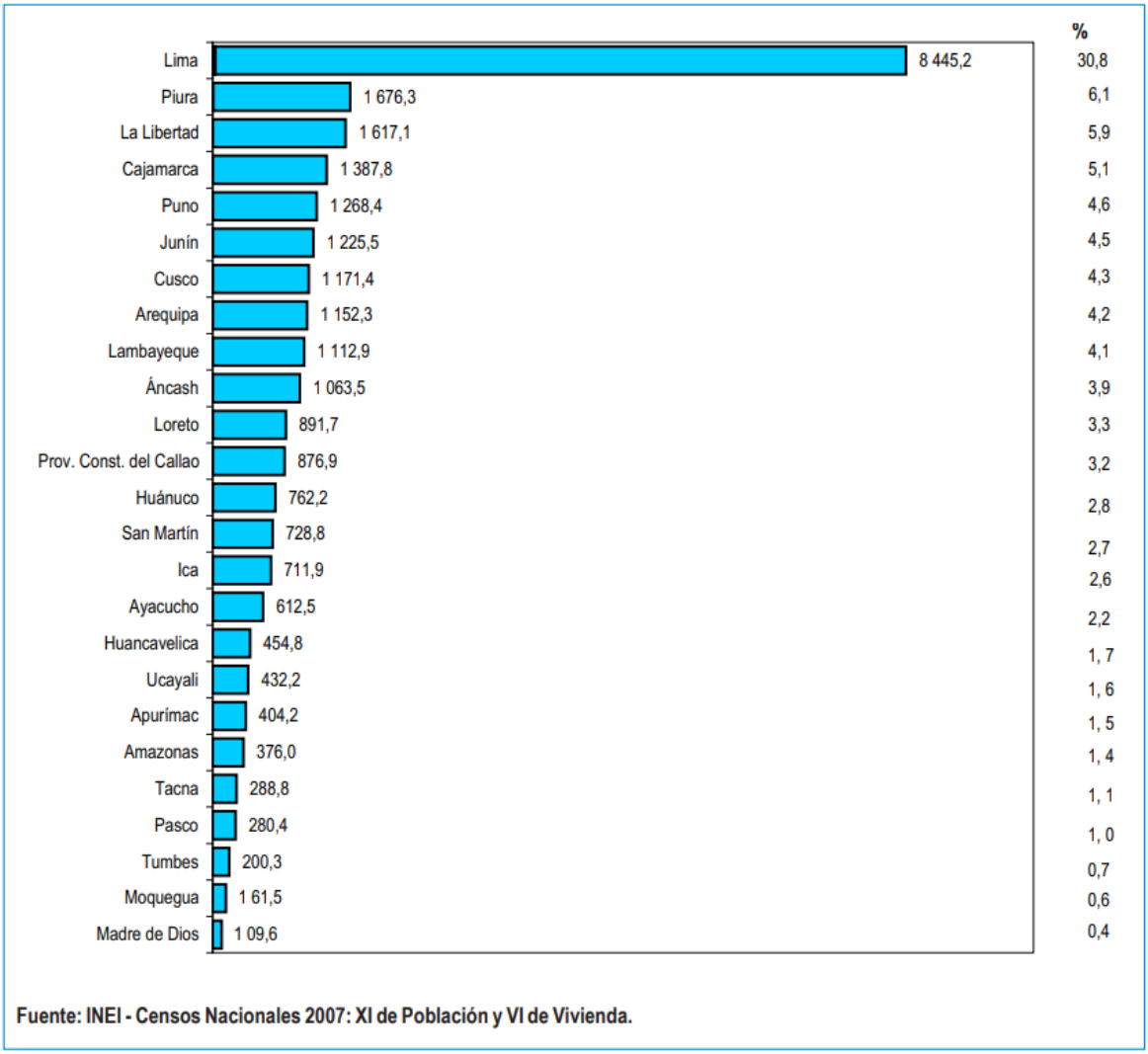

Figura 5. Perú: población censada, según departamento, 2007 (Miles)

Comportamiento por ámbito geográfico del Flujo Vehicular, enero 2023

Además, en La Libertad, el flujo vehicular total reportó un comportamiento negativo de 7,9%, influenciado por la interrupción del desplazamiento de vehículos en la carretera Panamericana Norte, en la provincia de Virú (Tabla 1).

Tabla 1. La Libertad: Índice del flujo vehicular total, 2021-2023.

Mes	2021	2022 P/	2023 P/	Variación Porcentual	
				Mensual [1/]	Anual [2/]
Ene.	234,3	249,2	229,6	-7,9	-0,3
Feb.	197,5	237,8			
Mar.	211,0	230,2			
Abr.	193,8	209,7			
May.	216,5	223,9			
Jun.	212,1	207,6			
Jul.	243,9	239,5			
Ago.	251,4	247,5			
Set.	235,5	225,4			
Oct.	252,6	241,1			
Nov.	234,8	224,4			
Dic.	264,5	236,9			
Promedio	**229,0**	**231,1**			

[1/] Respecto a similar mes del año anterior.

[2/] Últimos doce meses, respecto a similar periodo anterior.

Fuente: Ministerio de Transportes y Comunicaciones - PROVIAS Nacional.
Organismo Supervisor de la Inversión en Infraestructura de Transporte de Uso Público-OSITRAN
Elaboración: Instituto Nacional de Estadística e Informática - OTED.

b. **Modelo Computacional**

Un modelo computacional es una representación matemática y/o lógica de un sistema o proceso que se implementa y ejecuta en un computador. Estos modelos se diseñan para simular y analizar sistemas complejos, predecir comportamientos y realizar optimizaciones, entre otras aplicaciones.

En general, un modelo computacional consta de:

- **Entradas (inputs):** Datos o condiciones iniciales que se alimentan al modelo.
- **Procesos:** Conjunto de reglas, algoritmos, ecuaciones y lógica que el modelo utiliza para transformar las entradas en salidas.
- **Salidas (outputs):** Resultados generados por el modelo después de procesar las entradas.

Algunas características y aspectos clave de los modelos computacionales incluyen:

- **Abstracción:** Un modelo computacional no representa todos los detalles de un sistema real, sino que se enfoca en aquellos aspectos que son relevantes para el propósito del modelo.

- **Validación y verificación:** Es crucial validar y verificar un modelo computacional para asegurarse de que es una representación precisa del sistema real y que los resultados que produce son confiables.

- **Flexibilidad:** Dado que se basan en código y algoritmos, los modelos computacionales pueden ser adaptados y modificados según sea necesario para explorar diferentes escenarios o hipótesis.

- **Aplicaciones:** Se utilizan en una amplia variedad de campos, desde la física y la ingeniería hasta la biología, la economía y las ciencias sociales. Por ejemplo, en la meteorología, se usan modelos computacionales para predecir el clima; en la biología, para simular la interacción de proteínas; y en la economía, para modelar el comportamiento de los mercados.

- **Limitaciones:** A pesar de su utilidad, estos modelos tienen limitaciones inherentes debido a la simplificación de sistemas reales y la precisión de los datos de entrada. Por lo tanto, siempre es importante interpretar los resultados con precaución.

c. **Modelo de caracterización de patrones de tránsito vehicular**

La caracterización de patrones de tráfico vehicular se refiere al estudio y análisis de los flujos de tráfico en una red vial, con el objetivo de entender, predecir y gestionar el comportamiento y la demanda del tráfico. Un modelo de caracterización puede basarse en diversas métricas y técnicas dependiendo del objetivo del estudio.

Dentro de un modelo de caracterización de patrones de tráfico vehicular, se pueden considerar varios aspectos:

- **Intensidad del tráfico:** Se refiere al número de vehículos que pasan por un punto o segmento de la red en un periodo determinado, usualmente expresado en vehículos por hora (vph).
- **Distribución por tipo de vehículo:** Identifica la proporción de diferentes tipos de vehículos (automóviles, camiones, motocicletas, autobuses, etc.).
- **Velocidad y tiempo de viaje:** Estudia la velocidad a la que se desplazan los vehículos y el tiempo que tardan en recorrer un segmento.
- **Patrones temporales:** Analiza cómo varía el tráfico a lo largo del día, entre días laborables y fines de semana, o entre distintas estaciones del año.
- **Origen y destino:** Investigación sobre los puntos de inicio y término de los viajes, para entender las principales rutas y demandas de desplazamiento.
- **Condiciones de congestión:** Analiza los puntos o periodos donde el tráfico se vuelve más denso y se reduce la velocidad de circulación, lo que puede llevar a atascos.
- **Comportamiento del conductor:** Estudio sobre cómo se comportan los conductores en diferentes situaciones, como en zonas escolares, intersecciones, zonas de obras, entre otras.
- **Impacto de eventos o situaciones especiales:** Analiza cómo eventos específicos (como un concierto, un partido de fútbol o una construcción) pueden alterar los patrones normales de tráfico.

Estos modelos se utilizan en la planificación urbana y de transporte, diseño de infraestructuras, establecimiento de políticas de tráfico, entre otros. Por ejemplo, pueden ayudar a determinar dónde sería más efectivo construir un nuevo puente, ampliar una carretera o establecer una nueva línea de transporte público. Además, estos modelos

también son cruciales para desarrollar soluciones inteligentes de transporte, como sistemas de semáforos adaptativos o aplicaciones de navegación en tiempo real.

2.2. Marco conceptual

En esta sección se definen los conceptos básicos relacionados con la minería de datos espaciales. Una descripción general de las diferentes tareas de minería de datos enfocadas al caso espacial (Clasificación, Clustering, Reglas de Asociación, etc.), efectuando un contraste contra los métodos de minería de datos tradicionales. Por último, se describirán diferentes metodologías para la realización de un proceso de minería sobre conjuntos de datos geo-espaciales.

2.2.1. ¿Qué es la minería de datos?

Según Barry & Linoff (1997), la minería de datos es definida como "el proceso de extracción, exploración y análisis por medios automáticos o semiautomáticos de grandes cantidades de datos, para descubrir hechos, reglas y patrones útiles que se encuentren embebidos en los datos". Este proceso puede ser aplicado a diferentes casos de aplicación con el objetivo de apoyar la toma de decisiones y la planificación de nuevas estrategias.

Casos de aplicación de la minería de datos, donde es usada:

- En las áreas de créditos financiero y de seguros: índices de producción y costos, datos de tarjetas de crédito, detección de fraudes, mercadeo, geo-mercadeo.
- En salud: modelos de diagnóstico a partir de información almacenada en sistemas hospitalarios, gestión de tratamientos, diseño de campañas de prevención de enfermedades basadas en información geográfica, geo salud.
- En distribución comercial: análisis de compras, gestión de inventarios y planificación de transportes.
- En las ciencias: datos astronómicos, datos espaciales y biológicos.
- En análisis de textos (textmining): Internet, documentos multimedia.
- Geolocalización: localizar geográficamente algo como la congestión vehicular en un momento y lugar determinado.
- Aspectos climatológicos: predicción de tormentas, lluvias, etc.
- Determinación de niveles de audiencia de programas televisivos.

De los casos anteriores, resaltan algunos donde se combinan datos tradicionales con

datos espaciales. Por ejemplo, en un problema de Geolocalización, se puede ubicar problemas de congestionamiento vehicular en un lugar y tiempo real, considerando además que se podría hasta predecir en base a datos históricos el comportamiento de la congestión vehicular en el futuro relacionando estos datos con información de la vecindad espacial del área demográfica donde se produce.

La minería de datos aplicada a este tipo de problemas es tratada con un enfoque diferente, debido a la inclusión de la dimensión espacial en los diferentes análisis.

2.2.2. ¿Qué es la minería de datos espaciales?

La minería de datos espaciales es una extensión de la minería de datos que tiene que ver con la extracción de conocimiento, relaciones espaciales y otros patrones de interés que no están explícitamente almacenados en una base de datos espacial. En este tipo de minería, las implementaciones de los algoritmos deben ser adaptadas para trabajar sobre una tecnología de bases de datos espaciales, de tal forma que los datos puedan ser almacenados, consultados y analizados rápida y eficientemente.

En términos generales, la minería de datos espaciales se define como la búsqueda no trivial y automática de patrones embebidos en bases de datos espaciales, donde se involucran las siguientes tareas:

- Identificación de los objetos espaciales sobre los cuales se hace la búsqueda de los patrones.
- Identificación de atributos relevantes que caracterizan esos objetos.
- Identificación de relaciones espaciales insospechadas entre los objetos.
- Presentación de los resultados en una forma entendible, para apoyar análisis de toma de decisiones.

Identificar patrones comunes, asociaciones, reglas generales y nuevos conocimientos es una actividad altamente investigada, un proceso también conocido como descubrimiento de conocimiento en bases de datos (KDD).

La minería de datos y la minería de datos espaciales son el núcleo matemático del proceso KDD. Las tecnologías son parte de este proceso, incluidos los algoritmos de descubrimiento de datos, el desarrollo de modelos matemáticos y el desarrollo de modelos matemáticos.

Encuentre patrones significativos (implícitos o explícitos), que son intrínsecos. Conocimiento útil (Rokach & Maimon, 2010). Los patrones son las relaciones que existen entre los elementos de datos que se analizan. Los patrones son interesantes si son confiables, novedosos y útiles en términos del conocimiento que generan y su relevancia para los objetivos del análisis.

La minería de datos se define como "una técnica para extraer conocimiento útil y previamente desconocido de grandes cantidades de datos almacenados en varios formatos. En otras palabras, la tarea principal de la minería de datos. La minería de datos es encontrar patrones comprensibles en los datos" (Whitten et al., 2016). Técnicamente, la minería de datos es el proceso de encontrar correlaciones o patrones entre la información almacenada en una base de datos relacional. El crecimiento de los datos espaciales y el uso generalizado de bases de datos espaciales requieren procesos automatizados para identificar patrones válidos. La minería de datos espaciales es la técnica de encontrar, a través de varios métodos y herramientas, patrones interesantes y hasta ahora desconocidos, pero potencialmente útiles en bases de datos espaciales; Estos tipos de bases de datos no almacenan explícitamente patrones o reglas que definan relaciones espaciales entre objetos y ciertos atributos no espaciales (Shekhar et al., 2001). La complejidad de los datos espaciales y las relaciones espaciales intrínsecas limitan la utilidad de las técnicas tradicionales de minería de datos. Inicialmente, uno puede pensar que explorar datos espaciales está involucrado de formas similares utilizadas para descubrir datos tradicionales, debido a la complejidad de los datos espaciales, porque los objetos del espacio integrado no son solo las características generales expresadas en digital o texto, sino también espacial Características, como la ingeniería y su información tópica.

Aunque las técnicas y algoritmos de los datos tradicionales y satelitales son similares, deben usarse completamente de acuerdo con el problema; El enfoque tradicional varía según el enfoque espacial, de acuerdo con factores como:

i. La verdad es la primera en asumir las características de independencia actuales en la distribución de datos y la violación de la primera ley geográfica empacada por Tobler (1970) (Regtt & Lockwood, 2009). Todo está relacionado con todo lo demás, pero los próximos organismos se asocian principalmente con organismos remotos),
ii. datos complejos. y

iii. la presencia de la relación entre las propiedades espaciales. También debe procesarse con información almacenada a lo largo del tiempo, o la información actual puede considerarse una serie de eventos, como la aparición de una vez al mismo tiempo. Cierto hora o día de la semana. La minería de datos espaciales es similar a la minería de datos tradicional (Figura 7).

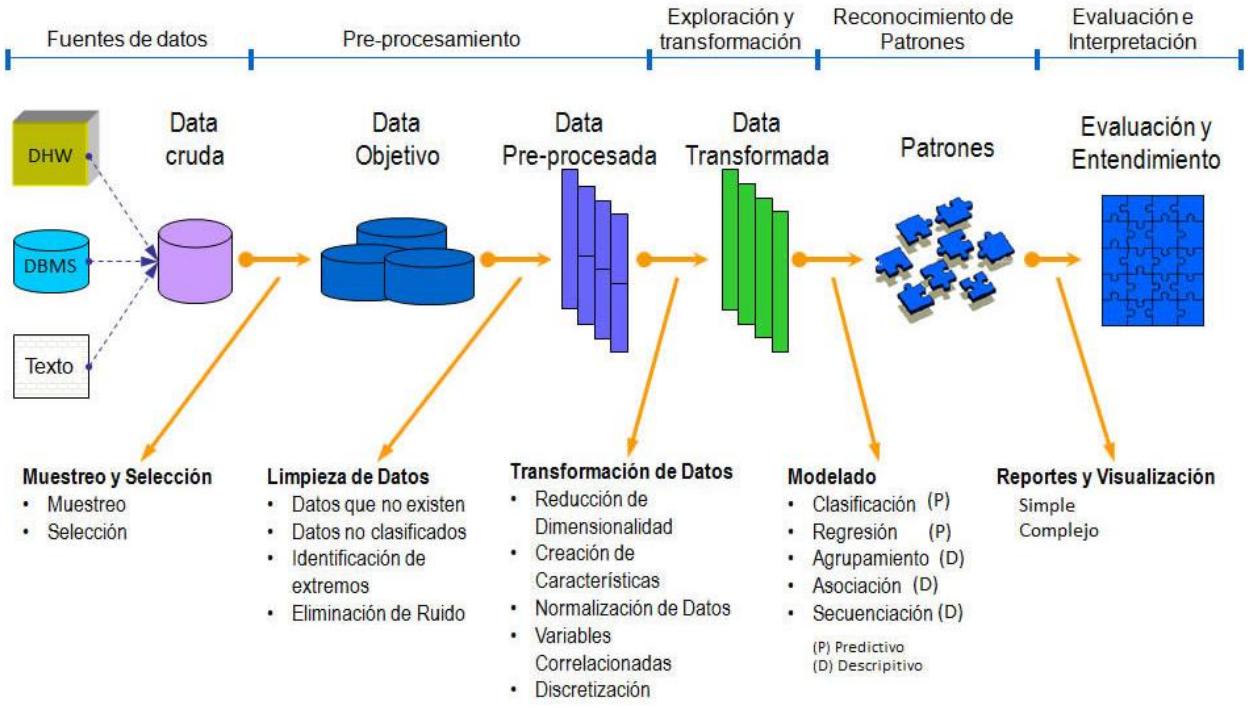

Figura 6. Proceso de minería de datos. Adaptado de Olmos-Pineda y González-Bernal (2007).

Un proceso típico de minería de datos implica los siguientes pasos generales (Hernández et al, 2007):

- **Selecciona conjuntos de datos**, tanto en términos de variables objetivo (aquellas pronosticadas, calculadas o inferidas), así como variables independientes (aquellas utilizadas en el cálculo o procesamiento), y puede muestrear de los registros disponibles.

- **Preparar datos**, incluidos gráficos, diagramas de dispersión, presencia de valores atípicos y datos faltantes (valores nulos).

- **Transformación del conjunto de datos de entrada**, que se realiza de diversas formas en base al análisis anterior, con el fin de preparar para aplicar la técnica de minería de datos que mejor se adapte a los datos y al problema. Este paso también se conoce como preprocesamiento de datos.

- **Selección y aplicación de técnicas de minería de datos**, construcción de modelos descriptivos o predictivos, y clasificación o segmentación.

- **Minería de conocimiento**, a través de la tecnología de minería de datos se obtiene un modelo cognitivo que representa los patrones de comportamiento observados en los valores de las variables problema o las relaciones entre las variables mencionadas anteriormente. También se pueden usar varias técnicas simultáneamente para generar diferentes modelos, aunque cada una generalmente requiere un procesamiento previo diferente de los datos.

- **Interpretar y evaluar modelos**, una vez que se dispone de un modelo hay que verificarlo, comprobando si las conclusiones a las que llega son válidas y suficientemente satisfactorias. En caso de que se obtengan varios modelos usando diferentes técnicas, los modelos deben compararse para encontrar el modelo que mejor se ajuste al problema. Si ninguno de los modelos produce los resultados esperados, se debe modificar uno de los pasos anteriores para crear nuevos modelos.

2.2.3. ¿Qué es un patrón espacial?

Un patrón espacial es un patrón cuya característica más relevante está relacionada con la ubicación espacial de los objetos involucrados. Generalmente se expresa como reglas, descripciones y tendencias encontradas en un conjunto de datos referenciados espacialmente. Ejemplos de algunos patrones:

• Zonas con población en ciertos rangos de edad y alta tasa de desempleo tienen una tasa de criminalidad superior al promedio.

• Osos de anteojos, generalmente se encuentran en bosques de niebla, cerca de fuentes de agua abierta.

• Los desplazamientos de población por causas de violencia desde las zonas selváticas del oriente y sur del país hacia el centro, hacen que enfermedades endémicas aparezcan en otras regiones.

• Zonas con alta densidad de población y sin puentes peatonales, presentan alta accidentalidad en ciertos tramos de carreteras (Zeitouni, 1999).

• El calentamiento inusual del océano pacifico (conocido como fenómeno del Niño), afecta el clima de otras regiones distantes (Shekhar, 2001).

2.2.4. ¿Qué es una base de datos espacial?

"Los datos espaciales están relacionados con los objetos que ocupan un espacio. Una base de datos espacial almacena datos representados por tipos de datos espaciales y relaciones entre tales objetos. Los datos espaciales llevan consigo información topológica o de distancia y son a menudo organizados por estructuras de índices espaciales y consultados por métodos de acceso espacial". (Koperski et al., 1997).

Una base de datos espacial es en general un sistema que maneja datos existentes en un espacio. Cuando se habla del concepto de espacio, existen diferentes tipos de datos que se pueden relacionar, por ejemplo: datos de la superficie de la tierra (geográficos), estructuras moleculares de ADN, planos de arquitectura, diseño de estructuras, etc.

Ejemplos de bases de datos espaciales:

• Bases de datos de Sistemas Geo-referenciados: Datos de censos, datos climáticos, datos de catastro, etc.

• Bases de datos de imágenes: Sistemas de censores remotos, Imágenes médicas, Imágenes de satélite, etc.

En este capítulo y en general a lo largo del presente trabajo, se utilizará una definición común para las bases de datos espaciales en el contexto de los Sistemas de Información Geográfica. Esta definición corresponde a un sistema de base de datos cuyo modelo ha sido extendido para manejar de manera eficiente tipos de datos geométricos, como: puntos, líneas y polígonos (ver figura 8). Estos tipos de datos son utilizados para representar las características de las capas o mapas temáticos1 en los Sistemas de Información Geográfica.

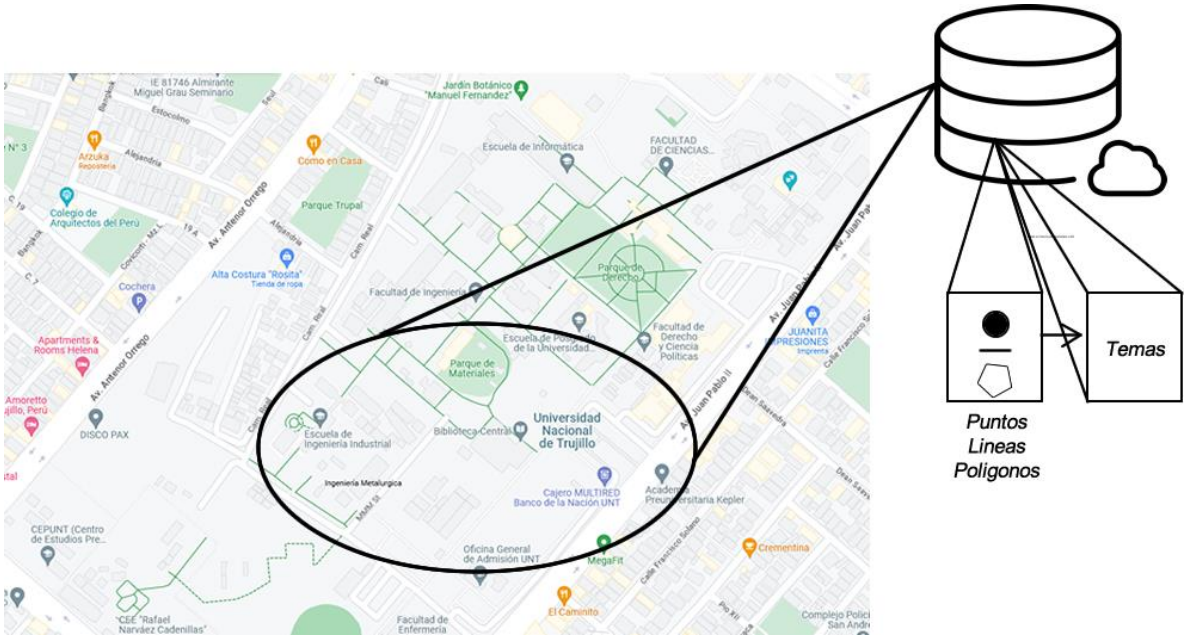

Figura 7. Representación de un ejemplo de una base de datos espacial.

2.2.5. Tareas de minería de datos espaciales

Las tareas de la minería de datos espaciales son una extensión de las aplicadas en la minería de datos tradicional y son llevadas a cabo usando diferentes métodos derivados de la estadística y del campo del aprendizaje de máquina. A continuación, se muestra una categorización con las tareas más comunes aplicadas a bases de datos espaciales (Tabla 2):

Tabla 2. Tareas para minería de datos espaciales. Tomado de: (Zeitouni, 2002)

Aproximación	Tareas de minería de datos espaciales	Métodos
Estadística	Análisis espacial estadístico (Geoestadística)	Análisis de correlación, Análisis factorial, Análisis Kriging, Regresión espacial etc.
Aprendizaje de Máquina	Descubrimiento de Patrones globales	Generalización, Reglas de caracterización
	Clasificación	Árboles de decisión SCART, etc.
	Clustering	Clustering basado en densidad DBSCAN, PAM, GAM etc.
	Asociaciones espaciales	Reglas de asociación espacial
	Tendencias y secuencias espacio-temporales	Reglas de tendencias, reglas de co-localización.

Las tareas de minería de datos se dividen en dos categorías (supervisadas y no supervisadas), como se muestra en la Figura 9.

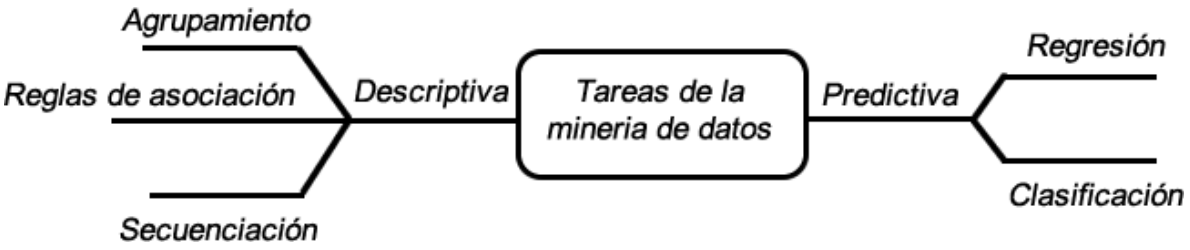

Figura 8. Tareas de la Minería de Datos.

2.2.5.1.*Predictiva*

El objetivo de este tipo de detección es predecir un valor particular de un atributo basado en otros atributos. El rasgo que se predice a menudo se denomina "categoría" o variable dependiente, mientras que los rasgos utilizados para predecir se denominan variable independiente. Permite predecir el valor de una variable desconocida (variable dependiente o variable objetiva) a partir de otras variables. Propiedades de la base de datos. (variables independientes) (Weiis & Indurkhya, 1998).

- **Clasificación:** El propósito de esta tarea es clasificar los datos en categorías específicas del dominio típico. Permite clasificar registros de categoría desconocida en categorías o categorías ya definidas en la base de datos (Tan et al., 2006).
- **Regresión:** predecir el valor de una variable con un valor continúo dado en base a los valores de otras variables, asumiendo un modelo dependiente lineal o no lineal. El objetivo es predecir los valores de una variable continua a partir del comportamiento de otra variable continua, el tiempo en general. Por ejemplo, está tratando de predecir la cantidad de clientes o pacientes, ingresos, llamadas, ganancias, costos, etc. A partir de los resultados de semanas, meses o años anteriores (Hernández et al., 2007).

2.2.5.2. *Descriptiva*

El objetivo de este tipo de minería es encontrar patrones (correlación, tendencia, conglomerado, ruta y anomalía) que resuman las relaciones en los datos. Es responsable de identificar modelos para describir los datos presentes (Han, Pei, & Kamber, 2011).

- **Agregación:** A partir de las categorías de un dominio particular se pueden obtener grupos o grupos en los que se combinan elementos similares (Riquelme, Ruiz, & Gilbert, 2006). Se permite la división en grupos cerrados y mutuamente excluyentes.
- **Regla de Asociación:** La asociación entre dos atributos ocurre cuando la frecuencia de ocurrencia de dos valores generalmente definidos para cada atributo es relativamente alta. Por ejemplo, en un supermercado se analiza si se compran juntos pañales y fórmula infantil (Hernández et al., 2007). Encuentra la relación entre dos o más características que ocurren con más frecuencia.
- **Secuenciación:** dado un conjunto de objetos, cada uno asociado con una línea de tiempo de eventos, encuentre reglas que predigan fuertes dependencias secuenciales entre diferentes eventos. Las reglas están formadas por los primeros patrones de detección. La ocurrencia de eventos en las muestras está sujeta a restricciones de tiempo (Hernández et al., 2007).

2.2.6. Agrupamiento (“Clustering”)

La agrupación en clústeres es una de las tareas principales en la minería de datos para descubrir clústeres e identificar distribuciones y características interesantes en los datos.

El proceso de agrupar un grupo de objetos físicos o abstractos en clases con objetos similares se denomina agrupación. El agrupamiento consiste en agrupar un conjunto dado de datos no etiquetados en un conjunto de grupos de manera que los objetos que pertenecen a un grupo sean idénticos entre ellos, mientras se busca la manera de que la diferencia de heterogeneidad entre los grupos sea lo más alta posible (Figura 9).

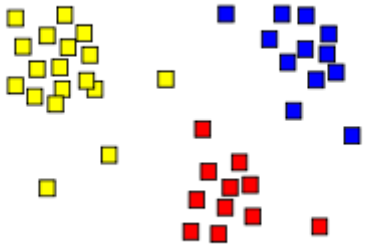

Figura 9. Formas de agrupamientos de componentes (clusters), diferenciados por colores.

Expresado en términos de varianza, hablaremos de minimizar la varianza dentro de los grupos para aumentar simultáneamente la varianza entre diferentes grupos. En el proceso de agregación, no hay clases predefinidas ni registros de muestra para ver las relaciones entre los datos, y puede considerarse como un proceso no supervisado. En este sentido, los clusters o grupos se crean en base a las características de los datos y no a la asignación de clases predefinidas, por lo que el clustering también se conoce como clasificación no supervisada.

2.2.6.1.Análisis de Clustering

Un problema de análisis de clustering, que forma parte de un grupo de instancias u objetos, cada uno de los cuales se caracteriza por una serie de variables.

A partir de esta información, el problema es obtener grupos de objetos, de manera que los objetos que pertenecen a un grupo son muy similares entre sí y, por un lado, la varianza entre diferentes grupos es muy alta. Aparece en términos de cambio, estamos hablando de reducir los cambios en los grupos para maximizar el cambio entre diferentes grupos (Figura 10).

$C_1 \dots C_i \dots C_k$
$O_1\ {x_1}^1 \dots {x_i}^1 \dots {x_k}^1$
$\dots\dots\dots\dots\dots$
$O_j\ {x_1}^j \dots {x_i}^j \dots {x_k}^j$
$\dots\dots\dots\dots\dots$
$O_n\ {x_1}^N \dots {x_i}^N \dots {x_k}^N$

Figura 10. Agrupación de N objetos en cada conjunto, k conjuntos.

Denotando por $O = O_1, \ldots, O_N$ al conjunto de N objetos, se trata de dividir O en k grupos o clúster, $C_1, \ldots, C_k$ de tal forma que: $\bigcup_{j=1}^{k} x_j = O$

Comenzando con una declaración del problema de agrupación, los procesos de análisis de grupo generalmente involucran los siguientes pasos:

1. **Representación de patrones**. Se refiere al establecimiento del número de clases, número de patrones, y el número, tipo y tamaño de las características disponibles para el algoritmo de clustering.
2. **Definición de proximidad**. La convergencia de modelos generalmente se mide como una función de la distancia entre un par de datos.
3. **Clustering**. se puede realizar de varias maneras. Se pueden usar grupos jerárquicos, divisionales y otros, incluidos métodos de probabilidad o teoría de grafos.
4. **Abstracción de datos**. Es el proceso de extraer una representación simple y compacta del conjunto de datos.
5. **Comprobación de resultados**. Consiste en validar el análisis de los grupos realizado mediante la evaluación de los resultados obtenidos.

2.2.7. Clustering espacial

El Clustering es una tarea de minería de datos de tipo descriptivo que busca agrupar de forma automática (no supervisada) un conjunto de datos, de acuerdo a una función de similaridad o distancia semántica.

Los métodos de Clustering para bases de datos espaciales no son muy diferentes comparados con los aplicados a datos tradicionales. De hecho, en las bases de datos espaciales es más natural el uso de la distancia Euclidiana para agrupar objetos vecinos.

El Clustering espacial es un proceso de agrupamiento de un conjunto de objetos en cluster de tal forma que los objetos dentro de ese cluster deben tener un alto grado de similaridad entre sí y ser diferentes a los objetos de otros clusters. El Clustering espacial es usado, por ejemplo, para determinar zonas críticas en análisis de criminalidad y seguimiento de enfermedades. En este tipo de análisis se busca encontrar clusters de eventos inusualmente densos a lo largo del tiempo y del espacio.

Para el caso espacial, se han desarrollado diversos algoritmos de Clustering, los cuales pueden ser divididos en cuatro categorías:

1. Métodos de Clustering Jerárquicos: Sucesivamente ejecutan un particionamiento o agrupamiento hasta encontrar un criterio de parada. Es por esto que los algoritmos de Clustering jerárquicos pueden ser divididos en métodos divisivos y aglomerativos.

   Algunos ejemplos de este tipo de Clustering son: BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies) propuesto en Zhang et al. (1997), CHAMELEON (Clustering Using Interconnectivity) descrito en Karypis et al. (1999), CURE (Clustering Using Representatives) Guha et al. (1998) y ROCK (Robust Clustering Using Links) Guha et al. (1999).

2. Algoritmo de agrupamiento particionales: comienzan con un número fijo de particiones arbitrarias y mueva iterativamente los puntos de datos hasta encontrar el criterio de parada. Estos métodos tienden a encontrar grupos esféricos. K-Means y K-Mediods son ejemplos conocidos de algoritmos de segmentación. Algunas implementaciones del algoritmo en esta categoría son: PAM (Partitioning Around Medoids), CLARA (Clustering LARge Applications) propuesta en Kaufman & Rousseeuw (1990) y otra denominada CLARANS (Clustering Large Applications Based On). La búsqueda estructurada), propuesta en Ng & Han (1994), es una combinación de PAM y CLARA para encontrar grupos de manera más eficiente, al mismo tiempo que permite detectar valores atípicos o puntos que no pertenecen a ningún grupo.

3. Algoritmos de Clustering basados en densidad: Intentan encontrar clusters basados en la densidad de los puntos de datos de una región. Estos algoritmos tratan a los clusters como regiones densas de objetos en el espacio de datos. Algunos ejemplos de implementaciones realizadas son: DBSCAN (Ester et al., 1997), OPTICS (Ordering Points to Identify Clustering Structure) descrito en Ankerst et al. (1999) y DENCLUE (Density Based Clustering) Hinneburg & Keim (1998).

4. Algoritmos de Clustering basados en mallas: Primero organizan el espacio de datos en un número finito de celdas y luego ejecutan las operaciones requeridas para organizar el espacio. Las celdas que contienen más de cierto número de puntos son

tratadas como densas. Las celdas densas son representadas al final como clusters. Los algoritmos basados en mallas son especialmente utilizados para analizar grandes conjuntos de datos espaciales. Algunos ejemplos de este tipo de métodos de Clustering son: STING (the Statistical Information Grid-Based Method) propuesto en Wang et al. (1997), CLIQUE (Clustering in-Quest) descrito en Agrawal et al. (1998) y GAM (The Geographical Analysis Machine) descrito en: http://www.ccg.leeds.ac.uk/SMART/GAM/.

2.2.8. Técnicas de Minería de Datos

Las técnicas de minería de datos permiten realizar tareas predictivas y descriptivas utilizando algoritmos de minería de datos. Dependiendo del propósito del análisis de datos, los algoritmos utilizados se clasifican en supervisados y no supervisados (Weiis & Indurkhya),

- **Aprendizaje supervisado (o predictivo):** predice el valor de un atributo (etiqueta) de un conjunto de datos, previamente desconocido, a partir de otros atributos conocidos (atributo descriptivo). A partir de los datos con etiquetas conocidas, se establece una relación entre dicha etiqueta y un conjunto de otros atributos. Estas relaciones se utilizan para hacer predicciones sobre datos con etiquetas desconocidas. Esta forma de trabajar se denomina aprendizaje supervisado y tiene lugar en dos fases: entrenamiento (construir un modelo a partir de un subconjunto conocido de los datos etiquetados) y prueba (probar el modelo en el resto de los datos).

- **Aprendizaje no supervisado (o descubrimiento de conocimiento):** se descubren patrones y tendencias en los datos. El descubrimiento de esta información se utiliza para tomar acciones y obtener beneficios (científicos o comerciales). La Tabla 3 muestra algunas técnicas de minería de datos.

Tabla 3. Clasificación de las técnicas de minería de datos.

Supervisados	No supervisados
Arboles de decisión	Detección de desviaciones
Inducción neuronal	Segmentación
Regresión	Agrupamiento (clustering)

Series Temporales	Reglas de Asociación
	Patrones Secuenciales

A continuación, se describe brevemente un conjunto seleccionado de tareas y técnicas relacionadas, incluidas las reglas de asociación, la clasificación (aprendizaje supervisado), la visualización geográfica multivariada y, más adelante, un enfoque en las técnicas de agrupación (clasificación no supervisada).

2.2.8.1. *Reglas de Asociación Espacial*

Aunque las reglas de asociación tienen muchas ventajas en la minería de datos espaciales, es difícil identificar predicados espaciales y establecer valores en diferentes niveles de reglas de asociación. También puede tener diferentes resultados según el método de entrada para los atributos no espaciales. Al igual que la extracción de reglas de asociación en bases de datos relacionales, las reglas de asociación espacial en bases de datos espaciales se pueden extraer teniendo en cuenta atributos y predicados espaciales (Ding et al., 2008).

El algoritmo utilizado en las reglas de asociación es PARM, que se basa en el clásico algoritmo Apriori. El algoritmo Apriori usa un enfoque jerárquico para generar la iteración completa de un conjunto de elementos, comenzando con frecuencia 1 de conjuntos de elementos. Basado en el hecho de que, si un conjunto de elementos es iterativo, entonces todos sus conjuntos secundarios también deben Sea iterativo, el algoritmo Apriori genera conjuntos de candidatos (k+1) de elementos repetidos a partir de k grupos de elementos y luego calcula el soporte para cada candidato (K+1) un grupo de elementos para formar un grupo regular (k+1) elementos.

2.2.8.2.*Clasificación Espacial*

Por lo general, en la clasificación espacial, los objetos se clasifican teniendo en cuenta tanto las características espaciales como las no espaciales. La clasificación espacial también utiliza árboles de decisión (Koperski & Han, 1998). Esta técnica utiliza predicados sobre relaciones entre objetos espaciales como criterio de decisión. En el primer paso, las propiedades espaciales se representan como predicados espaciales y luego se extrae la utilidad potencial de los predicados utilizando el algoritmo RELIEF. Para el segundo paso, se construye un árbol de decisión usando predicados (Koperski & Han, 1998).

Las ventajas de esta técnica son:

- El árbol de decisión se crea después de eliminar los predicados ineficaces y el costo de crear el árbol de decisión se reduce considerablemente.
- Se puede realizar una clasificación rápida y precisa mediante el uso de reglas simples mediante la construcción de un árbol de decisión binario y la reducción del costo computacional de poda por el algoritmo RELIEF.

2.2.8.3. *Caracterización Espacial*

Las propiedades espaciales extraen un esquema global de clases de datos para una región espacial utilizando las características espaciales de la región. Se proporciona información abstracta simple y clara sobre el área (Ester et al., 2001). Las propiedades espaciales evalúan si ciertas características de los objetos espaciales se extienden cerca de una región. Para ello, los objetos se identifican como vecinos de otros objetos teniendo en cuenta la distancia o la orientación. La información del vecindario se obtiene mediante el uso de una tabla de vecindario. La región operada por la característica espacial puede expandirse mediante un algoritmo de expansión espacial usando una tabla de vecindad (Guo & Gahegan, 2006).

2.2.9. Agrupamiento o Clustering

El análisis de conglomerados es un conjunto de técnicas multivariadas utilizadas para clasificar un grupo de individuos en grupos homogéneos. El análisis de conglomerados se aplica cuando no se sabe a qué grupo pertenecen los datos y se quiere encontrar dichos grupos, esta técnica agrupa los objetos en base a la información de los datos que describe los objetos y su relación. El objetivo es que los elementos de un grupo sean similares (o relacionados) entre sí y diferentes (o no relacionados) con los elementos de otros grupos. Cuanto mayor es la similitud (u homogeneidad) dentro de un grupo, y mayor la divergencia entre los grupos, más diferenciado es el grupo (Pascual et al., 2007). En la Figura 11, hay tres grupos, donde los elementos que pertenecen a cada grupo son similares y diferentes o no están relacionados con los elementos de los otros grupos.

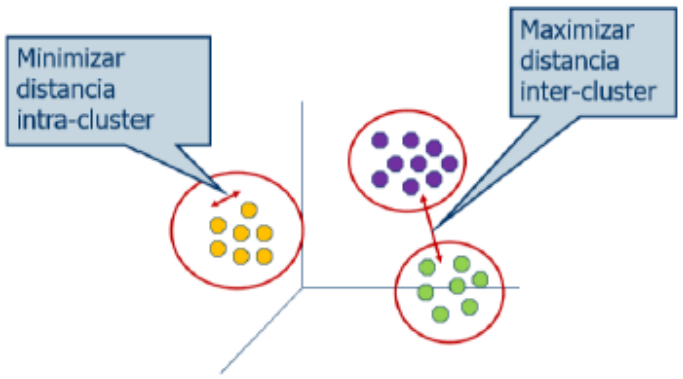

Figura 11. Análisis de Clúster. Tomado de (Tan et al., 2006)

La agrupación o agrupación en clústeres es una de las técnicas más útiles para encontrar conocimiento oculto en conjuntos de datos. Actualmente, el análisis de conglomerados en minería de datos es ampliamente utilizado en diversos campos, tales como: reconocimiento de patrones, análisis de datos espaciales, procesamiento de imágenes, computación y multimedia, análisis médico, economía, bioinformática y biometría (principalmente Han et al., 2011).

La operación de agrupamiento consta de los siguientes pasos (Jain 1999, Hernández, 2006).

- **Representación de patrones**: se refiere a la cantidad de clústeres, la cantidad de esquemas disponibles y la cantidad, el tipo y el tamaño de las funciones disponibles para el algoritmo de agrupación.
- **Definición de proximidad**: la proximidad del patrón suele medirse mediante una función de distancia definida; Esta función utiliza medidas de distancia como: euclidiana, Manhattan, Chebyshev y Minkowski. (Gibert & Nonell, 2005).
- **Clustering o agrupación**: Esto se puede hacer de varias maneras. Se pueden utilizar algoritmos de agrupamiento jerárquico distribuido y otras técnicas, incluidos métodos probabilísticos o teóricos de grafos.
- **Abstracción de datos**: Es el proceso de extraer una representación simple y compacta del conjunto de datos.
- **Verificación de resultados**: Consiste en validar el análisis de clustering realizado evaluando los resultados obtenidos.

2.2.10. Técnicas de Clustering

Los algoritmos de clustering difieren entre sí de acuerdo con las pautas que usan y el tipo de aplicación para la que está diseñada. La mayoría de ellos dependen del uso sistemático de distancias entre vectores (organismos al grupo), así como entre grupos que se formaron durante el ensamblaje. Las propiedades básicas sobre las que se pueden clasificar los algoritmos de agrupamiento (Hernández, 2006):

- El tipo de datos que están tratando (numéricos, categóricos y/o mixtos).
- Criterios utilizados para medir la similitud entre puntos.
- Conceptos y técnicas de agrupamiento utilizados (ej. lógica difusa y estadística).

En la literatura existe un gran número de técnicas de clustering que difieren según la arquitectura que se utilice (Jain et al., 1999). La clasificación común divide los algoritmos en: agrupación jerárquica, segmentación de grupos y agrupación basada en la densidad (Figura 13).

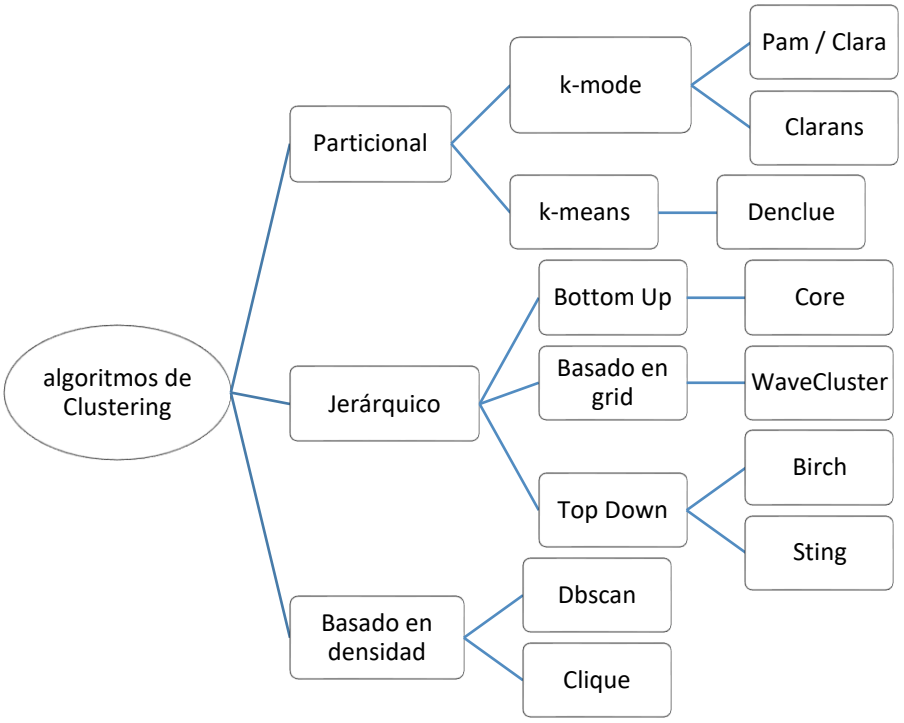

Figura 12. Algoritmos de Clustering. Tomado de (Tan et al., 2006)

2.2.10.1.*Clustering jerárquico*

Una colección de grupos anidados se organiza en un árbol jerárquico (Tan et al., 2006) (ver Figura 14). El método jerárquico produce un análisis jerárquico del conjunto de datos, formando un programa piezoeléctrico (árbol) que divide iterativamente el conjunto de datos en grupos cada vez más pequeños (Jain, 1999). El método descentralizado que se puede clasificar es el acercamiento o la división, dependiendo de la formación de descentralización (Han et al., 2011).

- **Aglomerativo**, también conocido como Ascendente, comienza cada objeto para formar una colección separada. Los objetos o grupos cerrados se unen consecutivamente, hasta que todos los grupos se fusionan en un solo grupo (el nivel más alto de la jerarquía) o hasta que se cumple la condición de terminación.
- **Divisivo**, también conocida como de arriba hacia abajo, comienza con todos los objetos en el mismo grupo. En cada iteración sucesiva, el bloque se divide en bloques más pequeños, hasta que se satisfacen todos los objetos del bloque o condición de terminación.

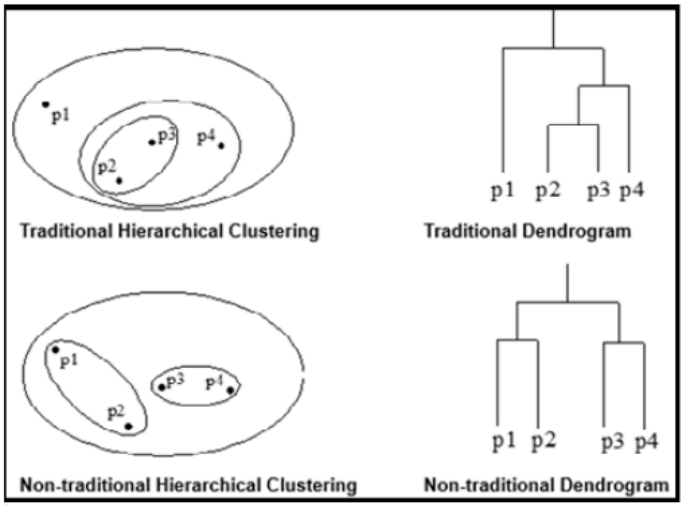

Figura 13. Clustering Jerárquico (Tan et al., 2006)

El clustering jerárquico no se recomienda para grandes bases de datos con millones de registros, porque la cantidad de espacios a calcular será mayor y la construcción del dendograma será complicada.

2.2.10.2.*Clustering particional*

El agrupamiento de particiones es la partición de objetos de datos en subconjuntos (clusters) que no se superponen, de modo que cada objeto de datos esté exactamente en un subconjunto (Tan et al., 2006) (consulte la Figura 15).

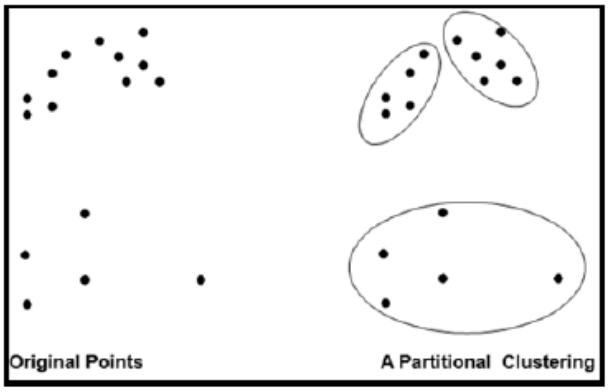

Figura 14. Ejemplo Clustering Particional (Tan et al, 2006)

Dado un conjunto de n objetos, el método de partición generará k particiones de datos, donde cada partición representa un grupo y $k \leq N$; Es decir, divide los datos en k grupos de manera que cada grupo debe contener al menos un objeto. En otras palabras, los métodos de partición realizan un nivel de partición en el conjunto de datos. Los métodos básicos de partición generalmente aplican la separación de bloques; Es decir, todo objeto debe pertenecer exactamente a un grupo (Han et al, 2011).

La agrupación particional se puede utilizar para grandes cantidades de datos, encontrando grupos mutuamente excluyentes de forma circular y en función de la distancia.

2.2.10.3.*Clustering basado en densidad*

Un clúster es una región densa de puntos, separados por regiones de baja densidad de otras regiones de alta densidad. Se utiliza cuando los agrupamientos son irregulares o entrelazados y cuando hay ruido (Tan et al, 2006).

La mayoría de los métodos de agrupación agrupan objetos en función de la distancia entre los objetos. Dichos métodos solo pueden encontrar cúmulos globulares y tienen dificultades para detectar cúmulos de forma arbitraria.

Se han desarrollado otros métodos de agrupamiento basados en el concepto de densidad. Su idea general es mantener el crecimiento de una población dada mientras la densidad (la cantidad de objetos o puntos de datos) en el "vecindario" exceda un cierto umbral. Por ejemplo, para cada punto de datos en un grupo dado, la vecindad de un radio dado debe contener la menor cantidad de puntos. Este método se puede utilizar para filtrar valores atípicos y detectar grupos de forma arbitraria (Han et al, 2011).

2.2.11. Algoritmos de Clustering

2.2.11.1. *Clustering jerárquico aglomerativo*

El algoritmo de agrupamiento aglomerativo es la técnica de agrupamiento jerárquico más popular. Los algoritmos jerárquicos tradicionales utilizan matrices de similitud o distancia (Tan et al, 2006).

Este enfoque de agrupamiento combina una colección de métodos de agrupamiento estrechamente relacionados que producen un agrupamiento jerárquico, comenzando cada punto como un agrupamiento de un solo elemento y agrupando repetidamente con los dos agrupamientos siguientes hasta que queda un solo punto (Flores, 2014). Los algoritmos de agrupamiento jerárquico son costosos en términos de requisitos de computación y almacenamiento y no se recomiendan para grandes cantidades de datos. El hecho de que todos los clústeres se combinen al final también puede causar problemas con datos ruidosos o de alta dimensión (Flores, 2014).

El algoritmo básico es sencillo:

Algoritmo de Clustering Aglomerativo.

1. Calcular la matriz de proximidad
2. Dejar que cada punto de datos sea un clúster
3. Repeat
4. Combinar los dos clústeres más cercanos
5. Actualizar la matriz de proximidad
6. Until solamente queda un solo clúster

Tomado de: (Tan et al., 2006).

2.2.11.2. Algoritmo de agrupamiento K-means

Esta técnica se basa en el agrupamiento dividido, que intenta encontrar un número de clústeres (K) especificado por el usuario, representado por sus centroides. El algoritmo básico se describe a continuación.

Primero, se seleccionan K centroides iniciales. donde K es un parámetro especificado por el usuario y corresponde al número deseado de clústeres.

Cada punto está asociado con el centroide más cercano, y cada colección de puntos asociados con un centroide representa un grupo. El centroide de cada clúster se actualiza en función de la asignación de puntos a los clústeres. El procedimiento de asignación y actualización se repite hasta que los puntos en los clusters no cambien o los centroides no cambien (Flores, 2014). (Figura 15).

El algoritmo básico de K-means consta de los siguientes pasos (Tan et al, 2006):

1. Seleccionar K puntos iniciales como centroides
2. Repetir
3. Formar K cluster asignando cada punto a su centroide más cercano
4. Recalcular el centroide de cada cluster
5. hasta que los centroides no cambien.

Para ejecutar este algoritmo se requiere elegir a priori el valor K (no se sabe cuántos grupos puede haber), esto se puede hacer de dos formas:

- Se puede utilizar un método jerárquico sobre una muestra de los datos (por eficiencia) para estimar el valor de K.
- Usar un valor de K alto, ver los resultados y ajustar.
    - Siempre que se aumente el valor de K disminuirá el valor de la suma de los cuadrados dentro de cada grupo (WCSS).
    - Lo normal es ir probando con varios valores de K y comprobar cuanto no hay de una mejora significativa en SSE.

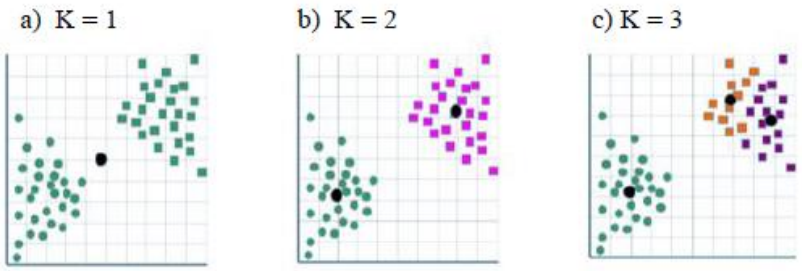

Figura 15. Algoritmo K-means. Tomado de (Tan et al, 2006)

Ejemplo: para encontrar tres clusters en datos de prueba, a partir de tres centroides definidos inicialmente, los clústeres finales se encuentran en cuatro iteraciones de asignación-actualización

Las limitaciones de K-means según (Tan et al., 2006) son:

- K-means tiene problemas cuando los clústeres son de diferente tamaño, densidades, que no tengan forma esférica.
- K-means tiene problemas cuando los datos contienen outliers.

La ventaja de K-means es ser un algoritmo simple, efectivo para pequeñas y medianas cantidades de datos. Utiliza el promedio para representar los centros de los clusters.

2.2.11.3. Algoritmo de agrupamiento DBSCAN

Es un método de clustering basado en densidad. La idea es hacer crecer un clúster siempre y cuando la densidad en el entorno del objeto exceda de un umbral. Este tipo de método permite la detección de clústeres de forma arbitraria, sirviendo además para filtrar datos ruidosos.

DBSCAN (agrupación espacial basada en la densidad de aplicaciones con ruido) detecta objetos centrales. H. Objetos adyacentes al alcance de la mano. Un objeto central y sus vecinos están conectados para formar regiones densas similares a grupos (Han et al. 2011). Este es un algoritmo de agrupamiento basado en la densidad que produce agrupamientos particionados en los que el algoritmo determina automáticamente el número de agrupamientos. El algoritmo no produce un agrupamiento perfecto, ya que los puntos dispersos se clasifican como ruido y se omiten (Tan et al., 2006).

Este algoritmo utiliza un enfoque de densidad basado en el centro. La densidad se calcula para un punto determinado contando el número mínimo de puntos (MinPts) necesarios para formar un grupo que se ajuste al radio de vecindad máximo (eps). Esto nos permite clasificar los puntos como aquellos que se encuentran dentro de la región densa (puntos centrales), aquellos que se encuentran en el borde de la región densa (puntos de borde) y aquellos que están dispersos (puntos de ruido o de fondo) (Tan et. al., 2006) (Flores, 2014).

- **Puntos de núcleo (core):** Puntos que tienen más de MinPts vecinos dentro de su vecindario de Radio Eps.
- **Puntos de borde (border):** Son los puntos que tienen menos de MinPts vecinos dentro de su vecindario de radio Eps, pero están en la vecindad de un punto de núcleo.
- **Puntos de ruido (noise):** Son aquellos puntos que no caen en ninguna de las dos categorías anteriores.

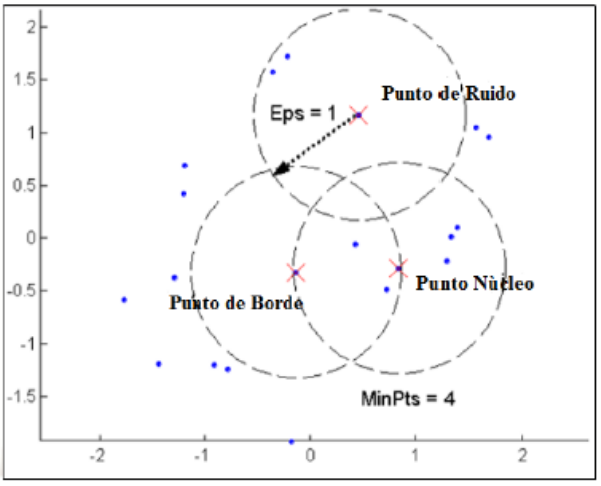

Figura 16. Puntos de Núcleo, Borde y Ruido (Considerando un Eps de valor 1 y un MinPts de valor 4). Tomado de (Tan et al., 2006).

El algoritmo comienza eligiendo un punto arbitrario p. Si p es un punto central, entonces comienza la formación de grupos y todos los objetos que se encuentran cerca de p se colocan en ese grupo. Si p no es el punto central, se visita otro objeto en el conjunto de datos. El procesamiento continúa hasta que se hayan procesado todos los objetos. Los puntos fuera del grupo formado se denominan puntos de ruido, y los puntos que no son ni ruido ni centro se denominan puntos de borde (Pascual et al., 2007).

Todos los pares de puntos centrales que están lo suficientemente cerca entre sí (menos de la distancia Eps) se asignan al mismo grupo. De manera similar, los puntos de borde lo suficientemente cerca del punto central se colocan en el mismo grupo que el punto central y los puntos de ruido se descartan (Tan et al., 2006).

El algoritmo DBSCAN se describe como sigue:

Algoritmo DBSCAN (Tan et al., 2006)

1. Etiquetar todos los puntos como núcleo, borde o ruido
2. Eliminar puntos de ruido.
3. Poner un borde entre todos los puntos de núcleo que están dentro de Eps de cada uno de otros
4. Convierta cada grupo de puntos centrales conectados en un clúster separado.
5. Asignar cada punto de borde a uno de los clústeres de sus puntos de núcleo asociados.

DBSCAN puede encontrar grupos de forma arbitraria. Sin embargo, es posible que no produzca un agrupamiento perfecto porque los puntos dispersos se consideran ruido y se filtran.

2.2.12. Método del codo

Se utiliza en minería de datos, para el caso de k-means, y ayuda a determinar, el número de clústeres que se deben elegir (MEDINA-VELOZ, 2016).

La idea básica de los algoritmos de clustering es la minimización de la varianza intra clúster y la maximización de la varianza inter clúster. Es decir, queremos que cada observación se encuentre muy cerca a las de su mismo grupo y los grupos lo más lejos posible entre ellos.

El método del codo utiliza la distancia media de las observaciones a su centroide. Es decir, se fija en las distancias intra clúster. Cuanto más grande es el número de clústeres k, la varianza intra clúster tiende a disminuir. Cuanto menor es la distancia intra clúster mejor, ya que significa que los clústeres son más compactos. El método del codo busca el valor k que satisfaga que un incremento de k, no mejore sustancialmente la distancia media intra clúster. (Gonzalo, s.f.) (Figura 18)

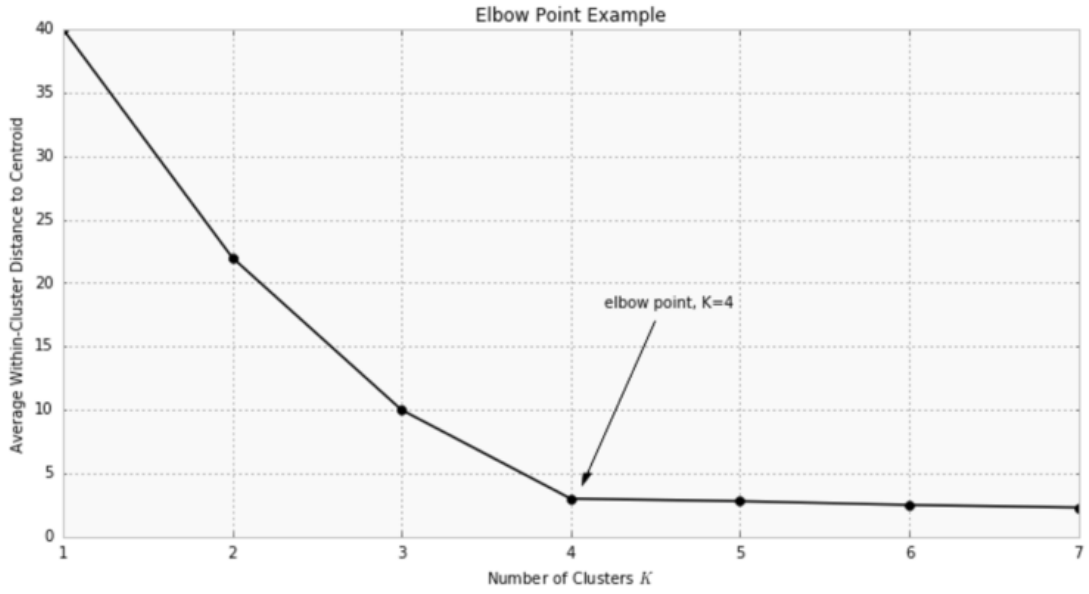

Figura 17. Ejemplo de gráfico, codo de Jambú (Gonzalo, s.f.)

2.2.13. Método de Coeficiente de Silhouette

El análisis de la silueta mide la calidad del agrupamiento o clustering. Mide la distancia de separación entre los clústeres. Nos indica como de cerca está cada punto de un clúster a puntos de los clústeres vecinos. Esta medida de distancia se encuentra en el rango [-1, 1]. Un valor alto indica un buen clustering.

Los coeficientes de silueta cercanos a +1 indican que la observación se encuentra lejos de los clústeres vecinos. Un valor del coeficiente de 0 indica que la observación está muy cerca o en la frontera de decisión entre dos clústeres. Valores negativos indican que esas muestras quizás estén asignadas al clúster erróneo.

El método de la silueta calcula la media de los coeficientes de silueta de todas las observaciones para diferentes valores de k. El número óptimo de clústeres k es aquel que maximiza la media de los coeficientes de silueta para un rango de valores de k.

El coeficiente de la silueta es calculado como:

$$S = \frac{b - a}{max(a, b)}$$

siendo a la distancia media intra clúster y b la distancia media a las observaciones del clúster más cercano. (Gonzalo, s.f.)

2.2.14. Datos espaciales

Un modelo de geodatos es una abstracción del mundo real que utiliza un conjunto de objetos de datos para admitir la visualización, consulta, manipulación y análisis de mapas. Los datos geográficos presentan información en términos subjetivos a través de mapas y símbolos que representan la geografía como formas geométricas, redes, superficies, lugares e imágenes, cada uno atribuido a sus respectivos atributos que los definen y describen.

Un dato espacial es una variable asociada a una posición en el espacio. Normalmente se utilizan datos vectoriales, que pueden representarse mediante tres tipos de objetos espaciales.

2.2.14.1. Puntos o Nodos

Se refieren a las coordenadas terrestres medidas por latitud y longitud (las cuales están dadas en medidas angulares medidas desde el centro de la Tierra). (Alonso Fernandez-Coppel, 2021)

2.2.15. METODOLOGÍA CRISP-DM

El proceso CRISP-DM fue desarrollado originalmente por un esfuerzo de consorcio formado por DaimlerChrysler, SPSS y NCR. CRISP-DM (Proceso Estándar Interindustrial para Minería de Datos).

Consta de un ciclo de seis fases:

1. **Entender el negocio**: Esta primera fase implica entender los objetivos y requerimientos del proyecto desde una perspectiva de negocio y aplicar estos conocimientos a la minería de datos, se centra en definir problemas y traducirlos en problemas preliminares. Transforma tus planes para alcanzar tus metas.
2. **Comprensión de datos**: la fase de comprensión de datos comienza con la recopilación inicial de datos, la familiarización con los datos, la identificación de problemas de calidad de los datos, el descubrimiento de datos tempranos, la identificación de subconjuntos interesantes y la formación de hipótesis sobre información oculta.
3. **Preparación de datos**: La fase de preparación de datos incluye todas las actividades para construir el conjunto de datos final a partir de los datos sin procesar iniciales.
4. **Modelado**: Esta fase selecciona y aplica varias técnicas de modelado y ajusta sus parámetros a valores óptimos.
5. **Evaluación**: En esta fase, el modelo recibido se vuelve a evaluar y los pasos tomados para crear el modelo se revisan para garantizar que los objetivos comerciales se han logrado adecuadamente.
6. **Implementación**: La creación de un modelo no suele ser el final de un proyecto. El propósito del modelo es aumentar nuestro conocimiento de los datos, pero los conocimientos obtenidos deben organizarse y presentarse para que los use el cliente.

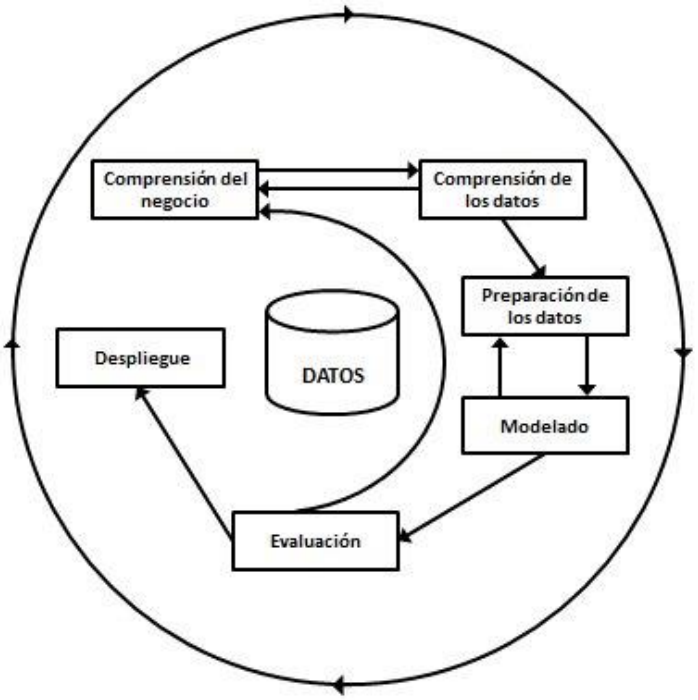

Figura 18. Esquema de la metodología CRISP.

2.2.16. Tareas y salidas de las fases de la metodología CRISP - DM

Fase 1. Comprensión del negocio

(Vallalta Rueda, 2016a) menciona: "El objetivo de esta fase es alinear los objetivos del proyecto de data mining con los objetivos del negocio. Tratando así de evitar embarcarnos en este proyecto de data mining que no produzca ningún efecto real en la organización." (párr. 5)

a) Establecer los objetivos de negocio.

b) Evaluar la situación actual.

c) Fijar los objetivos a nivel de minería de datos.

d) Plan de proyecto generado.

Fase 2. Comprensión de los datos

(Vallalta Rueda, 2016b) menciona que en estas fases deben ser capaces de:

a) Ejecutar procesos de captura de datos.

b) Proporcionar una descripción del juego de datos.

c) Tarea de explotar los datos

d) Gestionar la calidad de datos identificando problemas y proporcionando soluciones.

Fase 3. Preparación de los datos

(Vallalta Rueda, 2016b) menciona que en esta fase debemos ser capaces de:

a) Establecer el universo de datos con los que trabajar.

b) Realizar tareas de limpieza de datos.

c) Construir un juego de datos apto para ser usado en modelos de minería de datos.

d) Integrar datos de fuentes heterogéneas si es necesario.

Fase 4. Modelado

(Vallalta Rueda, 2016b) menciona que en esta fase debemos ser capaces de:

a) Seleccionar las técnicas de modelado más adecuadas para nuestro juego de datos y nuestros objetivos.

b) Fijar una estrategia de velicación de la calidad del modelo.

c) Construir un modelo a partir de la aplicación de las técnicas seleccionadas sobre el juego de datos.

d) Ajustar el modelo evaluando su habilidad y su impacto en los objetivos anteriormente establecidos.

Fase 5. Evaluación del Modelo

(Vallalta Rueda, 2016b) menciona que en esta fase debemos ser capaces de:

a) Evaluar el modelo o modelos generados hasta el momento.

b) Revisar todo el proceso de minería de datos que nos ha llevado hasta este punto.

c) Establecer los siguientes pasos a tomar, tanto si se trata de repetir fases anteriores como si se trata de abrir nuevas líneas de investigación.

Fase 6. Despliegue

(Vallalta Rueda, 2016b) menciona que en esta fase debemos ser capaces de:

a) Diseñar un plan de despliegue de modelos y conocimiento sobre nuestra organización.

b) Realizar seguimiento y mantenimiento de la parte más operativa del despliegue.

c) Revisar el proyecto en su globalidad con el objetivo de identificar lecciones aprendidas.

CAPÍTULO III

MARCO METODOLÓGÍCO

3.1. Hipótesis central de la investigación

La minería de datos espaciales basada en técnicas de agrupamiento con los algoritmos k-means y dbscan, permite medir el nivel de congestión del tráfico vehicular en la red vial de la ciudad de Trujillo.

3.2. Variables e indicadores de la investigación

Tabla 4. Operacionalización de variables.

Variable	Definición conceptual	Dimensiones	Indicador
Dependiente Congestionamiento del Tráfico vehicular	Es un fenómeno urbano y de transporte que ocurre cuando la cantidad de vehículos en una carretera o área excede su capacidad de flujo, provocando una disminución de la velocidad, aumentos en los tiempos de viaje y la acumulación de vehículos.	Factores de congestión vehicular	Densidad del tráfico
			Velocidad vehicular
			Tiempo de Tránsito
Independiente Minería de Datos Espaciales	La minería de datos espaciales se refiere al proceso de descubrir y extraer patrones y conocimientos ocultos de grandes conjuntos de datos espaciales. Estos datos espaciales suelen estar asociados con objetos geográficos (por ejemplo, coordenadas, polígonos, mapas, imágenes satelitales) y pueden tener información sobre la ubicación y/o la forma de los objetos.	Técnicas de Agrupamiento	Algoritmo k-means
			Algoritmo DBSCAN

3.3. Métodos de la investigación

3.3.1. Tipo de Investigación

La investigación desarrollada es de tipo aplicada descriptiva.

3.4. Diseño de la investigación

Diseño de investigación no experimental transversal Explicativa.

3.5. Población y Muestra

3.5.1. Población

Para este estudio se consideró como población todas las vías de la ciudad de Trujillo.

3.5.2. Muestra

Se considero la muestra por conveniencia de los 66 puntos (lugares) de congestionamiento proporcionados por la dirección de Transporte Metropolitano de la ciudad de Trujillo (ver Figura 19).

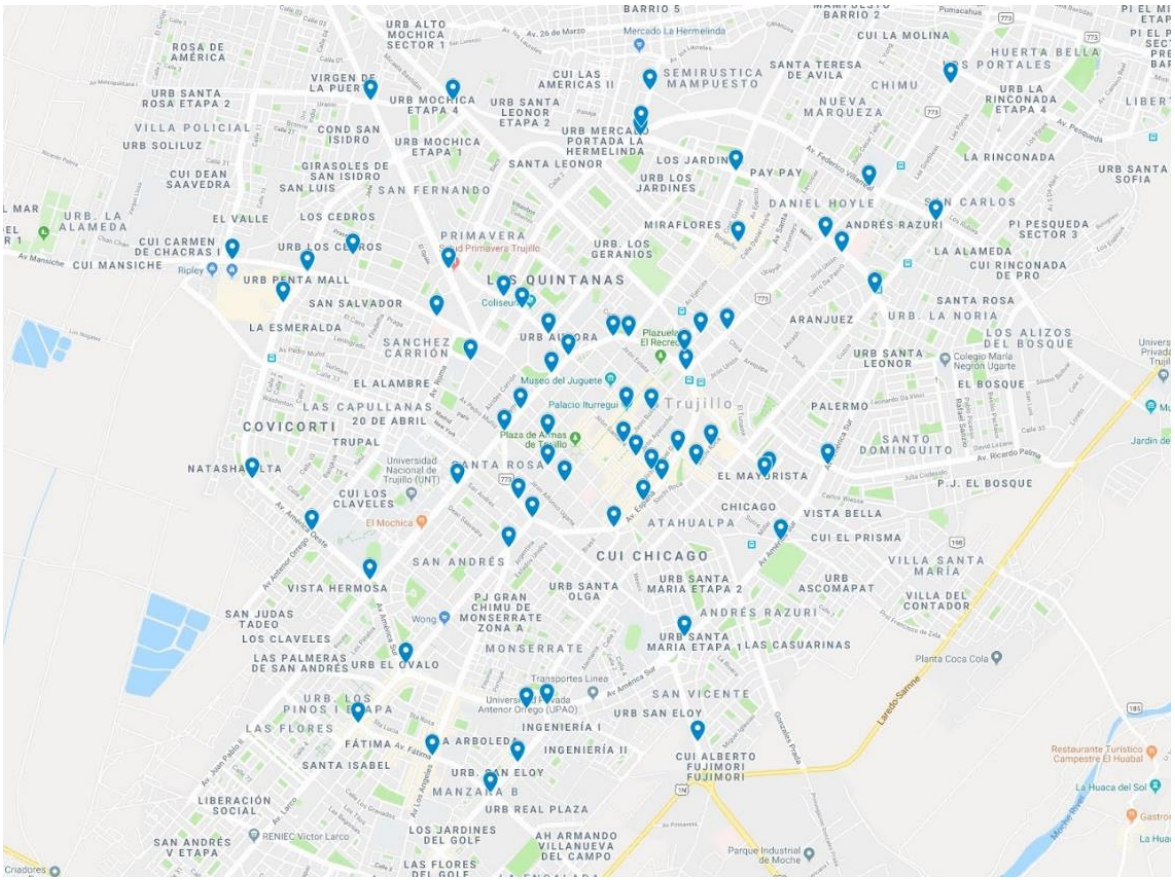

Figura 19. Puntos críticos detectados por la dirección metropolitana de transporte de la ciudad de Trujillo.

3.6. Actividades del proceso investigativo

Se desarrolló un modelo computacional que explica el proceso del análisis de la congestión vehicular permitiendo ingresar características del flujo de vehículos, explicando su comportamiento por ubicación, fecha y hora de ocurrencia, grado de congestión vehicular en determinadas horas pico, etc. Para llevar a cabo la ejecución de este modelo se recopilan datos y luego, mediante la aplicación de técnicas de minería de datos, se analizan y se encuentran patrones de comportamiento del tráfico vehicular.

También se desarrolló un diagrama de flujo (Figura 20), como hoja de ruta para indicar el orden secuencial donde se describe las fases de desarrollo para la formalización de un modelo de caracterización del flujo vehicular en la ciudad de Trujillo, así como la aplicación de los algoritmos kmeans y dbscan. El diagrama de flujo se basa en la metodología CRISP-DM (figura 20).

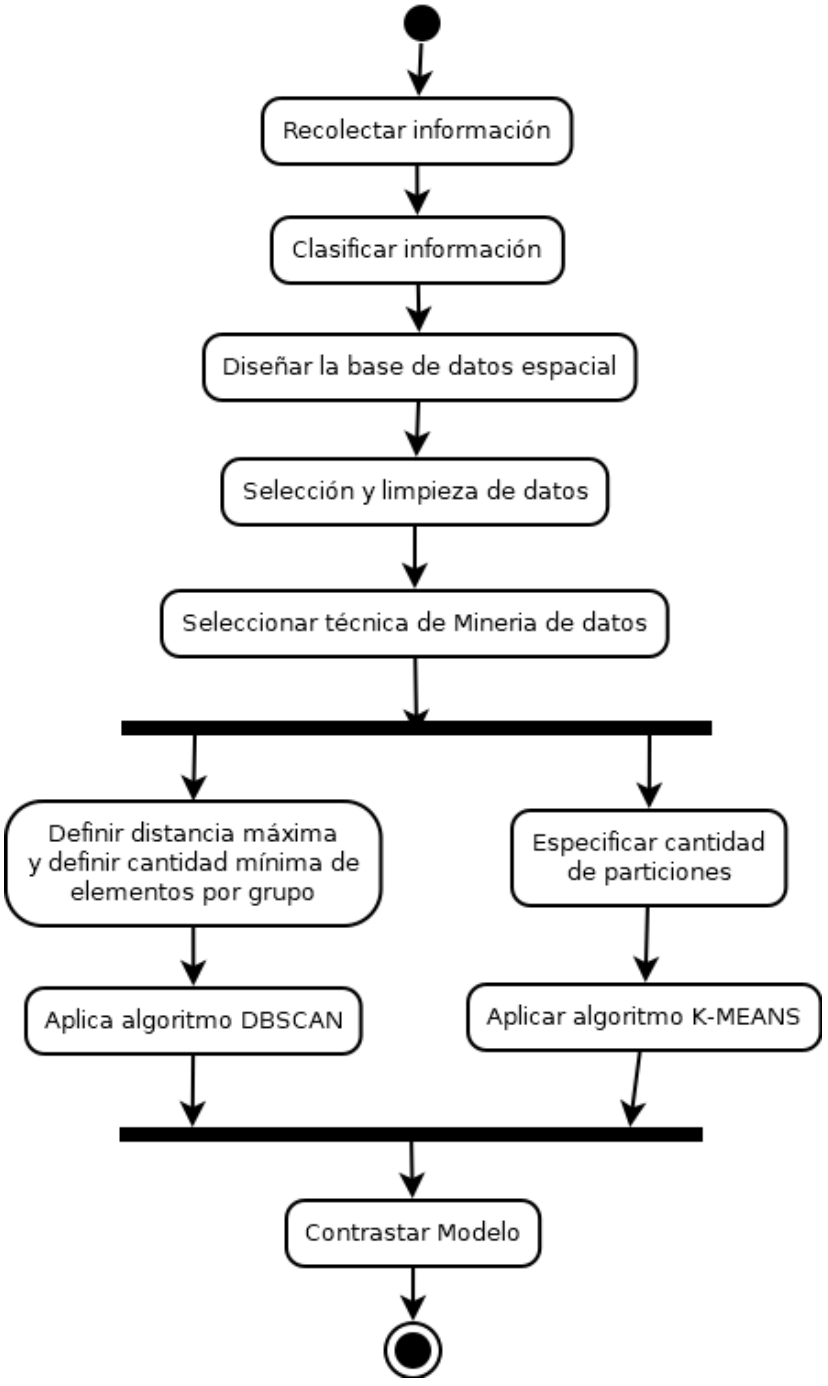

Figura 20. Fases de modelo de caracterización del flujo vehicular basado en la Metodología CRISP-DM.

Se recolectó información sobre los diversos puntos (nodos) críticos congestionables detectados por la oficina metropolitana de transporte de la ciudad de Trujillo.

De la información recolectada se seleccionó los datos relevantes requeridos para construir una base de datos espacial y un sistema de software para la gestión y control del tráfico de la ciudad de Trujillo.

Se diseñó la base de datos espacial utilizando del diagrama de flujo (figura 21) y luego generamos el modelo relacional (figura 22) en el software gestor de base de datos MySQL.

Para referenciar cada punto geográfico (nodo, vía o ruta) se seleccionaron las cantidades: 1) latitud, 2) longitud, 3) semáforos, 4) horas punta (turnos de mañana, tarde y noche), 5) cantidad de vehículos que hay en cada vía, 6) cantidad tope de vehículos, 7) velocidad permitida para los vehículos según lo dispuesto por ordenanza municipal, 8) tiempo permitido vehículos según lo dispuesto por ordenanza municipal, 9) velocidad aproximada (promedio) de los vehículos que están en el momento en la vía, 10) tiempo aproximado (promedio) de los vehículos que están en el momento en la vía.

La limpieza de datos, se hizo tomando en cuenta que, para ejecutar los algoritmos k-means y dbscan necesitamos un conjunto de datos sólidos, donde no haya datos nulos, no datos textuales, tampoco redundancia de datos.
Se seleccionó la técnica de minería de datos a emplear en esta investigación: agrupamiento.
Se seleccionó el algoritmo considerando que, el conjunto de datos de este estudio está compuesto, en parte, por datos geográficos, lo que influyó en la selección del algoritmo a aplicar en la etapa de minería, por lo cual se escogió dbscan como el más apropiado. DBSCAN es eficiente incluso para grandes bases de datos espaciales. Este algoritmo (densidad clustering espacial basado en aplicaciones con ruido) está diseñado para descubrir los clústeres y el ruido en un espacio espacial (Ester et al., 1996).

Para optimizar el número de clústeres a considerar en los algoritmos K-means (agrupamiento particional) y dbscan (agrupamiento basado en localidades), se usó el método del codo de Jambú y el método de coeficiente de Silhouette.

3.7. Técnicas, Herramientas e instrumentos de la investigación

En esta investigación se aplicaron diversas técnicas e instrumentos para la recolección de información, como:

- Técnicas: Metodología CRISP-DM, diagramas de Flujos de Datos (Figura 21), Modelo relacional de base de datos (Figura 21) y generación de la base de datos numérico (dataset de datos) considerando el formato de la muestra que está conformada por los 66 puntos críticos de la ciudad de Trujillo.
  La técnica de minería de datos espacial, empleada de: agrupamiento, basada en los algoritmos k-means (Clasificación) y dbscan (agrupamiento).
- Herramientas computacionales:
  - Softwares de aplicación (Rapidminer, weka), para atender el modelo de datos.
  - Software Dia, MySQL, para tratar el análisis y diseño de la base de datos.
  - Lenguajes de programación Python, para la programación del sistema de software.
- Instrumentos para la recolección de información: Revisión de documentación y hojas de cálculo (Anexo A, Anexo D, Anexo E).

3.8. Procedimiento para la recolección de datos

Para la recolección de los datos, se hizo una entrevista al personal de la Dirección de Transporte Metropolitano de Trujillo (anexo D y anexo E).

Una vez recolectada la información de los puntos críticos (que presentan congestión, según la oficina metropolitana de transporte de la ciudad de Trujillo) está se analizó y clasificó, se descartan aquellos datos que son irrelevantes o innecesarios, se trata la presencia de datos faltantes o perdidos y se detecta la presencia de valores que no se ajustan al comportamiento general de los datos. Otra actividad que se realiza dentro de esta fase es georreferenciar (localización espacial) las coordenadas geográficas (latitud y longitud), de la ubicación de los puntos críticos de la ciudad de Trujillo.

Diseño de la Base de Datos. Una vez se ha recopilado la información, se diseña el diagrama de Flujo de datos de la base de datos (Figura 21).

3.9. Aplicación de la metodología CRISP-DM

3.9.1.1. Comprensión del negocio

- **Determinar los objetivos del negocio**

- ✓ Determinar mediante el aforo/ volumen simulado de vehículos, el nivel del congestionamiento, tal como lo indica en su información la oficina Metropolitana de transportes de la ciudad de Trujillo (ver Anexo A).
- ✓ Extraer conocimiento, el cual puede ser en forma de relaciones, patrones o reglas inferidas de los datos y previamente desconocidos, o bien en forma de una descripción más concisa. Estas relaciones o resúmenes constituyen el modelo de los datos analizados.
- ✓ Diagnosticar a través de un modelo descriptivo, las rutas congestionadas utilizando técnicas de agrupamiento de datos espaciales basadas en los algoritmos K-Means y dbscan.

- **Evaluar la situación**

  Debido a que se cuenta con información descriptiva del fenómeno observado, se hace un análisis de los datos en un diagrama de flujo de datos (figura 21), a partir del cual se diseña una base de datos espacial (figura 22).

  - **Análisis de los datos**

    De la figura 21 se determina a cada punto de referencia (calle, jirón, avenida, esquina, etc.) como un nodo (el cual tiene espacialmente una ubicación geográfica a través de su longitud y latitud), siendo así reconocido hacia adelante en los datos de la figura 22.

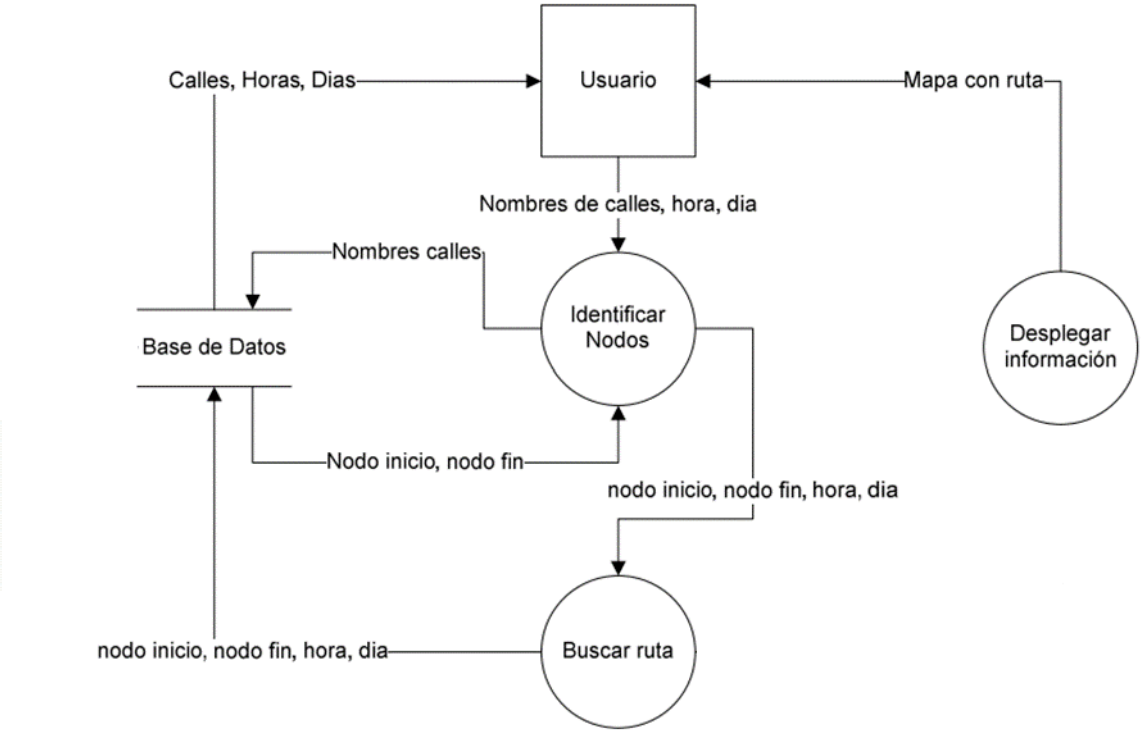

Figura 21. Diagrama de Flujo de Datos de la situación actual del tráfico en la ciudad de Trujillo.

- **Diseño de la Información**

De acuerdo a los datos analizados y a la información proporcionada por la oficina metropolitana de transporte de la ciudad de Trujillo, se hace el diseño de una base de datos espacial a medida para almacenar los datos identificados en la congestión en los determinados puntos de referencia o nodos.

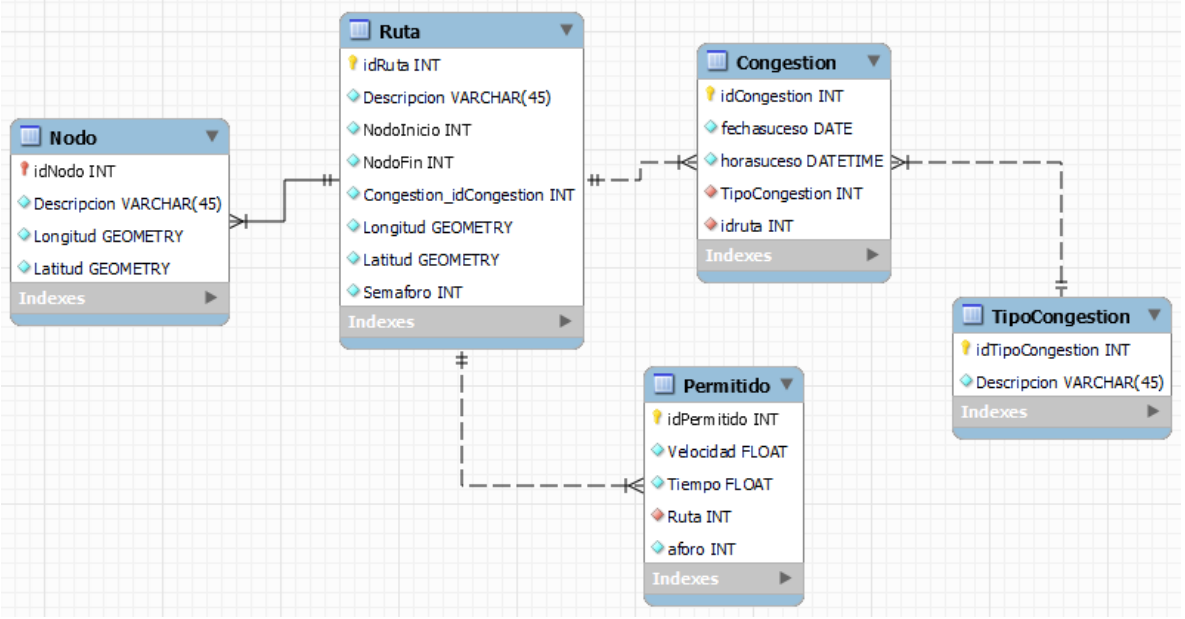

Figura 22. Modelo relacional de la base de datos espacial, para almacenar y determinar la congestión y sus niveles en la ciudad de Trujillo.

- **Descripción de las Variables:**

De acuerdo al análisis elaborado de la información brindada por la oficina Metropolitana de Transporte de la ciudad de Trujillo, se determinó el modelo relacional de la base de datos de la figura 21.

Tabla 5. Tabla Nodo que almacena los puntos de referencias del sistema vial del tráfico vehicular.

idNodo	Clave primaria.
Descripción	Descripción o nombre de un punto geográfico (ovalo, esquina de calles, etc.)
Longitud	Determina la ubicación de la dirección o coordenada geográfica, perteneciente a la ruta.
Latitud	

Tabla 6. Tabla Ruta, almacena la descripción del nodo.

idRuta	Clave primaria.
Descripción	Descripción o nombre de la ruta (calle, avenida, jirón, pasaje, etc.)
NodoInicio	Identifica el nodo inicial (lugar de origen) desde donde inicia una ruta.
NodoFin	Identifica el nodo final (lugar destino) donde finaliza una ruta.
Longitud	Determina la ubicación de la dirección o coordenada geográfica, perteneciente a la ruta.
Latitud	

Tabla 7. Tabla Congestión, almacena los momentos de las ocurrencias del fenómeno.

idCongestion	Clave primaria.
Fechasuceso	Descripción de la fecha (dia/mes/año), cuando ocurre la congestión.
Horasuceso	Descripción de la hora (hora:minuto:segundo), cuando ocurre la congestión.
TipoCongestion	Identifica el nivel de congestión.

Tabla 8. Tabla Tipocongestion, almacena los niveles de congestión**.**

idTipoCongestion	Clave primaria.
Descripción	Descripción del nivel de congestión.

Tabla 9. Tabla Permitido, almacena los datos asignados por la oficina metropolitana de transporte.

idPermitido	Clave primaria.
Velocidad	Cantidad de velocidad permitida (según norma establecida por la municipalidad provincial de Trujillo)

Tiempo	Cantidad de tiempo permitido (según norma establecida por la municipalidad provincial de Trujillo)
Aforo	Cantidad de vehículos permitidos para cada ruta (según norma establecida por la municipalidad provincial de Trujillo)

- **Determinar objetivos de minería de datos**

  Los objetivos de minería de datos en nuestra investigación son las siguientes:

  a) Procesar toda la información recopilada, para que así pueda ser manejada por los métodos clustering, estos dos métodos solo trabajan con valores cuantitativos.

  b) Hallar la distancia de los datos, en cuestión de acercamiento entre unidades vehiculares, esto medirá cuanta es la diferencia del nivel de congestionamiento en cada punto de la ciudad donde se produzca la aglomeración.

- **Producir plan de proyecto**

  Para producir el plan de proyecto de esta investigación se procedió a estimar el tiempo de realización:

  - Etapa 1: Datos analizados, estructurados y recopilación de la información.

    Estimación: 1 mes.

  - Etapa 2: Realizar consultas para adquirir muestras que puedan representar los datos.

    Estimación: 2 semanas.

  - Etapa 3: Fase 3 Preparación de los datos, seleccionar, limpiar, conversión y el formateo si en caso fuera necesario, para facilitar el análisis de estos datos.

    Estimación: 1 mes.

  - Etapa 4: Seleccionar la técnica que se utilizara para la ejecución de los modelos sobre los datos procesados.

    Estimación: 2 semanas.

  - Etapa 5: Realizar un análisis general de cómo se trabajó la metodología incluyendo la etapa anterior.

    Estimación: 2 semanas.

- Etapa 6: Generar informes acerca de los avances hechos, para una futura referencia si se tiene que replantear algún criterio de evaluación.

  Estimación: 2 semanas.

- Etapa 7: Mostrar los resultados

  Estimación: 2 semanas.

3.9.1.2. Comprensión de los datos

En esta fase de CRISP-DM se procederá a recolectar todos los datos iniciales para poder establecer un primer contacto con el problema, conocer los datos y averiguar su calidad.

✓ **Recolectar datos Iniciales**

Los datos utilizados en este tema de investigación son datos asociados a las velocidades, tiempos, distancias de rutas, semaforizaciones, latitudes y longitudes de lugares de referencia a puntos geográficos estos datos serán recolectados de manera ordenada en una hoja de cálculo en MS Excel, el cual se procedió a ordenar de manera que cada se convierta en una variable numérica.

✓ **Describir los datos**

Se trabajó con 66 puntos y con los algoritmos que comparen estos datos. Cada punto crítico hace referencia a:

Los campos seleccionados para cada punto crítico de referencia son:

- Latitud: viene hacer la distancia en grados, minutos y segundos con respecto al ecuador (0º).
- Longitud: Es la distancia en grados, minutos y segundos que hay con respecto al meridiano de Greenwich (0º).
- Semáforo: determina si en el punto crítico de referencia puede existir un semáforo.
- HoraPuntaM: determina la hora punta por la mañana en que se produce congestión vehicular en un determinado punto crítico.
- HoraPuntaT: determina la hora punta por la tarde en que se produce congestión vehicular en un determinado punto crítico.
- HoraPuntaN: determina la hora punta por la noche en que se produce congestión vehicular en un determinado punto crítico.

- Cantidadvehiculos: determina el número de vehículo que actualmente se encuentren en el punto crítico.
- Tope: se refiere al número limite que debe existir en cualquier punto crítico.
- VelocidadPermtida: se refiere a la velocidad impuesta según ordenanza municipal que debe existir para cruzar la ruta o punto crítico.
- TiempoPermitido: se refiere al tiempo impuesto según ordenanza municipal que debe existir para cruzar la ruta o punto crítico.
- VelocidadAprox: se refiere a la velocidad actual con la que se encuentra una unidad móvil al cruzar la ruta o punto crítico.
- TiempoAprox: se refiere al tiempo actual con la que se encuentra una unidad móvil al cruzar la ruta o punto crítico.

✓ **Exploración los datos**

Tal como indica la figura 23 se muestra los diferentes puntos críticos (ubicaciones geográficas) con relación a las diferentes horas puntas. Se puede apreciar que depende mucho de las horas puntas en que se encuentra congestionado los puntos críticos para que la relación varié, no todos los puntos cuentan con la misma hora punta.

✓ **Verificar la calidad de los datos**

Después de hacer la exploración inicial de los datos se puede afirmar que estos son completos. Los datos cubren los casos requeridos para la obtención de los resultados necesarios para poder cumplir los objetivos del proyecto. Los datos no contienen errores y tampoco se encuentran valores fuera de rango ya que son datos generados y controlados respectivamente de manera automática por el software simulador, por lo que no hay riesgo de ruido en el proceso de la minería de datos.

3.9.1.3. Preparación de los datos

En esta fase de la metodología CRISP – DM, se procederá a mostrar los pasos para alistar la data, previamente a ser procesada.

✓ **Seleccionar datos**

Se utilizó sólo los datos numéricos, ya que se utilizarán los métodos de clustering, estos dos métodos tienen como requisito utilizar datos numéricos para ser comparados. Sin embargo, hay datos que se pueden prescindir.

Los campos seleccionados para cada punto crítico de referencia son:

- Latitud: viene hacer la distancia en grados, minutos y segundos con respecto al ecuador (0º).
- Longitud: Es la distancia en grados, minutos y segundos que hay con respecto al meridiano de Greenwich (0º).
- Semáforo: determina si en el punto crítico de referencia puede existir un semáforo.
- HoraPuntaM: determina la hora punta por la mañana en que se produce congestión vehicular en un determinado punto crítico.
- HoraPuntaT: determina la hora punta por la tarde en que se produce congestión vehicular en un determinado punto crítico.
- HoraPuntaN: determina la hora punta por la noche en que se produce congestión vehicular en un determinado punto crítico.
- Cantidadvehiculos: determina el número de vehículo que actualmente se encuentren en el punto crítico.
- Tope: se refiere al número limite que debe existir en cualquier punto crítico.
- VelocidadPermtida: se refiere a la velocidad impuesta según ordenanza municipal que debe existir para cruzar la ruta o punto crítico.
- TiempoPermitido: se refiere al tiempo impuesto según ordenanza municipal que debe existir para cruzar la ruta o punto crítico.
- VelocidadAprox: se refiere a la velocidad actual con la que se encuentra una unidad móvil al cruzar la ruta o punto crítico.
- TiempoAprox: se refiere al tiempo actual con la que se encuentra una unidad móvil al cruzar la ruta o punto crítico.

El motivo de inclusión o exclusión de algunos campos, depende de la importancia de los datos para llegar al objetivo de la investigación.

✓ **Limpiar datos**

Nuestra base de datos no necesita una limpieza a profundidad, porque se manejó los criterios básicos para construir la información, no permitir datos nulos, no permitir

datos textuales, como también la redundancia de los datos, esto provocaría una pérdida de tiempo al momento de analizar los datos de forma ordenada.

- ✓ **Construir datos**

Atributos derivados

En esta parte solo se puede destacar la transformación de los datos numéricos de los diferentes puntos críticos porque al momento de utilizar los métodos de clustering para que puedan ser comparados.

Registros generados

Todo registro generado será un gráfico, para entender mejor la data, adicionalmente si el grafico es demasiado grande para ser entendible, se procedió a desmenuzar este.

- ✓ **Integrar datos**

Fue necesario la fusión de las tablas para producir el dataset que es procesado por el sistema de software construido con el lenguaje de programación Python.

Formatear datos

Se aseguraron que los datos estén en un formato adecuado, como fechas y datos numéricos para el método de clustering.

3.9.1.4. Modelado

En esta fase de la metodología CRISP-DM se procedió a generar los modelos a realizar, porque la data obtenida debe ser representada de manera gráfica para un mejor entendimiento.

3.9.1.4.1. Seleccionar técnicas de modelado

Se utilizó el algoritmo k-means, conocido como un algoritmo de clasificación no supervisada (cauterización), este agrupa objetos en k grupos basándose en sus características, consta de tres pasos:

- Inicialización: Escogido el número de grupos, se establece en K centroides en el espacio de los datos.

- Asignación de objetos a los centroides: cada objeto es asignado a su centroide más cercano.
- Actualización de centroides: se actualiza dependiendo de cada grupo tomado como nuevo centroide.

Se utilizó la técnica de modelado dbscan, conocido como un algoritmo de agrupación en clústeres, dbscan se basa en la aglomeración de agrupaciones determinada por la densidad que está diseñada para descubrir agrupaciones de forma arbitraria. dbscan requiere solo un parámetro de entrada y ayuda al usuario a establecer un valor apropiado para él. En los puntos de un mismo clúster, su k-ésimo vecino debería estar más o menos a la misma distancia. En los puntos de ruido, su k-ésimo vecino debería estar más lejos. dbscan está diseñado para descubrir los clústeres y el ruido en una base de datos espacial (Han, Kamber y Pei, 2012). Idealmente, se deberían conocer los parámetros apropiados Eps (radio de vecindad) y MinPts (como vecinos mínimos para considerar un punto como punto central) de cada grupo y al menos un punto del grupo respectivo. Luego, se podrían recuperar todos los puntos que son de densidad alcanzable desde el punto dado usando los parámetros correctos. Pero no hay una manera fácil de obtener esta información por adelantado para todos los grupos de la base de datos. Sin embargo, hay una forma simple y efectiva: la heurística para determinar los parámetros Eps y MinPts. Por tanto, dbscan usa valores globales para Eps y MinPts, es decir, los mismos para todos los grupos. Los parámetros de densidad del grupo "más delgado" son buenos candidatos para estos valores de parámetros globales que especifican la densidad más baja que no se considera ruido.

El conjunto de datos de este estudio está compuesto, en parte, por datos geográficos, lo que influyó en la selección del algoritmo a aplicar en la etapa de minería de datos, por lo cual se escogió dbscan como el más apropiado. El algoritmo de clustering dbscan requiere solo un parámetro de entrada y ayuda al usuario a determinar un valor apropiado para ello. dbscan es eficiente incluso para grandes bases de datos espaciales. Este algoritmo (densidad clustering espacial basado en aplicaciones con ruido) está diseñado para descubrir los clústeres y el ruido en un espacio espacial (Ester, Kriegel, Jorg y Xu, 1996).

3.9.1.4.2. Generar el plan de prueba

El procedimiento que se empleó para probar la calidad y validez del modelo será la comparación de los algoritmos k-means y dbscan. Estas medidas determinaran el modelo más optimo.

CAPÍTULO IV

RESULTADOS Y DISCUSION

4.1. Exploración de los datos

La figura 23 muestra las diversas intersecciones de los nodos, según una determinada hora punta, tal como lo indica el anexo A.

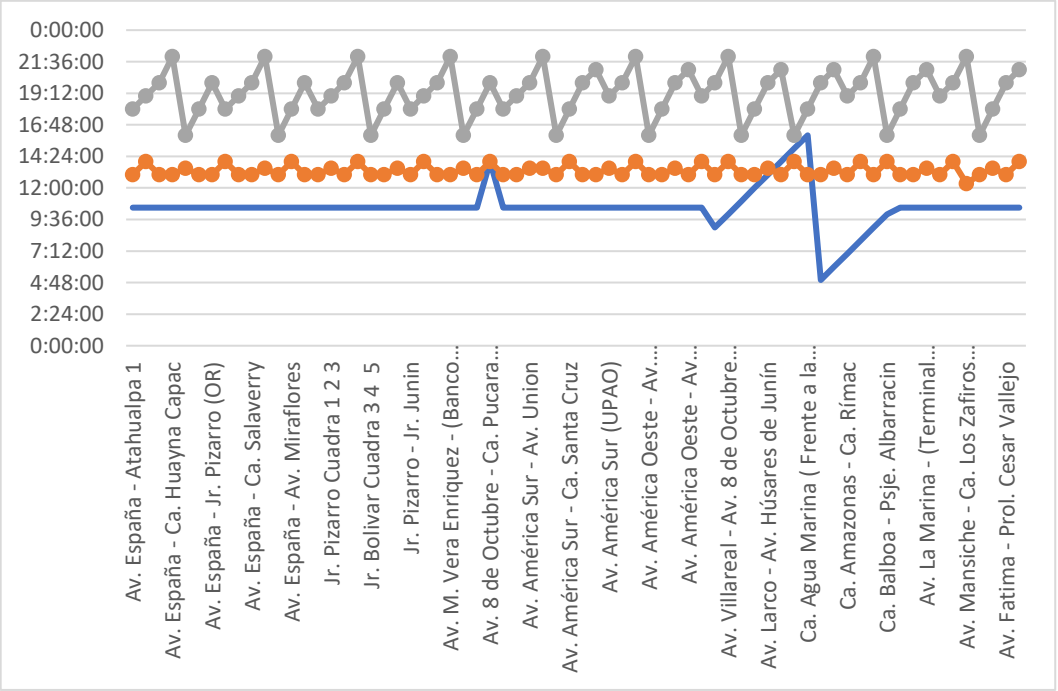

Figura 23. Exploración de los datos en general

4.2. Construcción del modelo

Para el presente trabajo, según lo expuesto por (Epstein, 2008), los modelos desarrollados son modelos explicativos, no modelos predictivos, sino modelos explícitos cuyos resultados son evaluados por otros después de la implementación. La intención de este modelo es explicar, no predecir. El grado de congestión puede explicarse a partir del lugar de ocurrencia y la fecha y hora de ocurrencia, pero es imposible predecir la hora y el lugar de ocurrencia. Los datos primero se recopilan y luego se analizan mediante la aplicación de técnicas de minería de datos para descubrir patrones de comportamiento de la congestión. La información oculta se puede encontrar aplicando técnicas de minería de datos y puede generar nuevos problemas para desarrollar aún más el modelo.

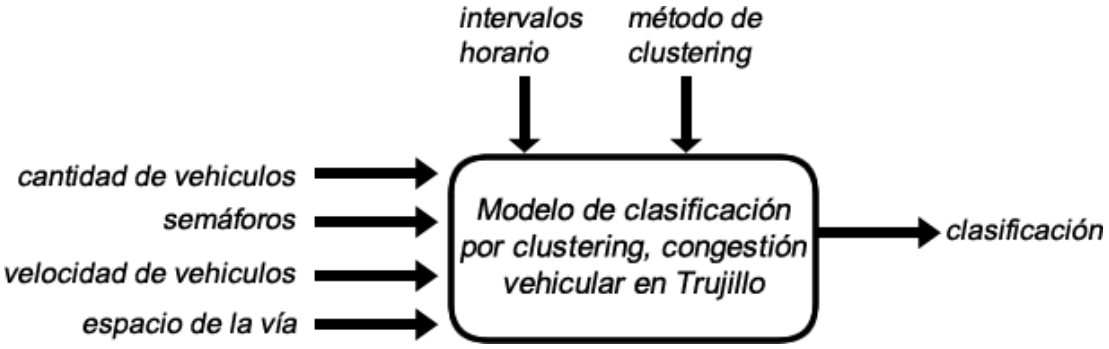

Figura 24. Prototipo del modelo de caracterización de congestión vehicular.

Para ejecutar el modelo elegido sobre los datos de entrenamiento se utilizó el software RapidMiner, donde se pudo definir y construir el tipo de modelado. El análisis se realizó utilizando la técnica de minería de datos, clustering. Tal como se muestra en la figura 25.

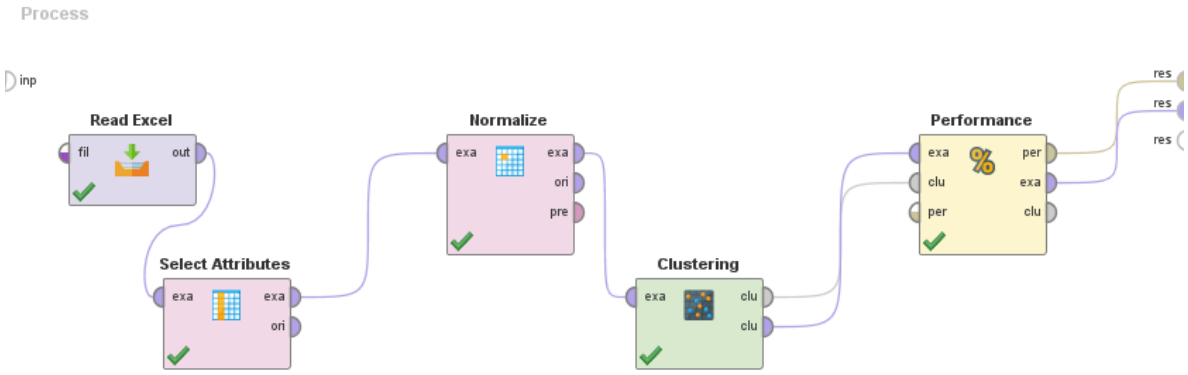

Figura 25. Diseño del modelo en software RapidMiner

Se implementaron 2 algoritmos de minería de datos: k-means y DBSCAN. Estos algoritmos son de tipo no supervisado lo que significa que no necesitan de una variable de clasificación y pueden ser utilizados en grandes cantidades de datos (Moreno García & López Batista, n.d.)

4.3. Resultados Computacionales

Los resultados obtenidos en este estudio permitieron comprender cómo se organiza y gestiona la información espacial en las bases de datos espaciales según sus propiedades (representación, relaciones y operaciones).

De igual forma, sobre estas bases de datos se puede realizar el proceso de preparación y transformación de conjuntos de datos de entrada para su posterior aplicación de técnicas de minería de datos.

- Evaluación del Modelo.
- Determinar el nivel de Congestionamiento.
- Influencia de la implantación del sistema mediante satisfacción de los Usuarios

4.3.1. Evaluación del Modelo

Los algoritmos de agrupamiento k-means y dbscan funcionan de manera diferente, pero k-means calcula aleatoriamente la cantidad de grupos o particiones para recuperar y la distancia entre los objetos dentro de cada grupo. En dbscan, el radio de distancia entre objetos y el número mínimo de objetos en cada grupo. Allí se entendió la utilidad, funcionalidad y aporte a esta investigación de los algoritmos de agrupamiento. Porque permiten un análisis espacial preciso de elementos aleatorios sin un patrón común.

Se consideró la ejecución de las técnicas de agrupamiento utilizando algoritmos de minería de datos k-means y dbscan, desarrollando un software simulador que permita mostrar los niveles de congestión vehicular.

Para detectar la mayor cantidad de concentraciones y ubicar aquellos puntos de la ciudad de mayor tráfico recurrente, se tomaron en cuenta un dataset identificando la información de los puntos de la ciudad brindados por la oficina metropolitana de Transporte de Trujillo.

4.3.1.1. Análisis de datos

Se aplicó para los métodos de clustering K-Means y dbscan el mismo dataset respectivamente.
Para el algoritmo K-Means, se realizó la técnica de acodamiento para encontrar el valor más adecuado para el número de agrupamientos (clúster) en una primera instancia.

Para el algoritmo K-Means se aplicó el método del coeficiente Silhouette para medir la calidad de los agrupamientos (clusters) encontrados. Este método va a permitir determinar qué tan bien cada objeto se encuentra dentro de su agrupación. El número óptimo de clusters k es el que maximiza la silueta promedio en un rango de posibles valores de k.

Para el algoritmo dbscan, se utilizó la técnica KNN (k-nearest neighbors), para hallar el valor más adecuado de la distancia épsilon. El objetivo de su aplicación es calcular el promedio de las distancias de todos los puntos a sus k vecinos más cercanos. El valor de k será incremental en un ciclo y se corresponde con MinPts.

A continuación, con la distancia óptima calculada con la técnica KNN, se realizaron varias iteraciones para calcular los agrupamientos obtenidos con el algoritmo dbscan, con un valor de minPts de 2 a 8. El valor de minPts óptimo se lo asociara con número máximo de clusters determinado, luego de la ejecución de todas las iteraciones.

El algoritmo DBSCAN provee una clusterización comparativamente mejor con otros métodos, además es apropiado para el tratamiento de datos geográficos, con lo cual es posible establecer gran cantidad de clústeres en la zona de estudio.

Se comparó los resultados de los algoritmos K-Means y dbscan.
Con la ejecución del algoritmo K-Means se obtuvieron resultados más precisos, mientras que con el algoritmo dbscan se obtuvo información de tipo aglomerativo.

4.3.1.2. Algoritmo K-means

Para encontrar un número óptimo de clústeres, que no involucren agrupaciones heterogéneas (pocos clústeres); o datos que siendo muy similares unos a otros los agrupemos en clústeres diferentes (muchos clústeres). Por lo que, se consideraron 2 métodos para dicha elección: la técnica de acodamiento y el método del coeficiente Silhouette.

En la figura 26, se ejecutó la técnica de acodamiento, donde se puede observar que el número sugerido de clústeres K, como parámetro para el algoritmo K-Means está en 3.

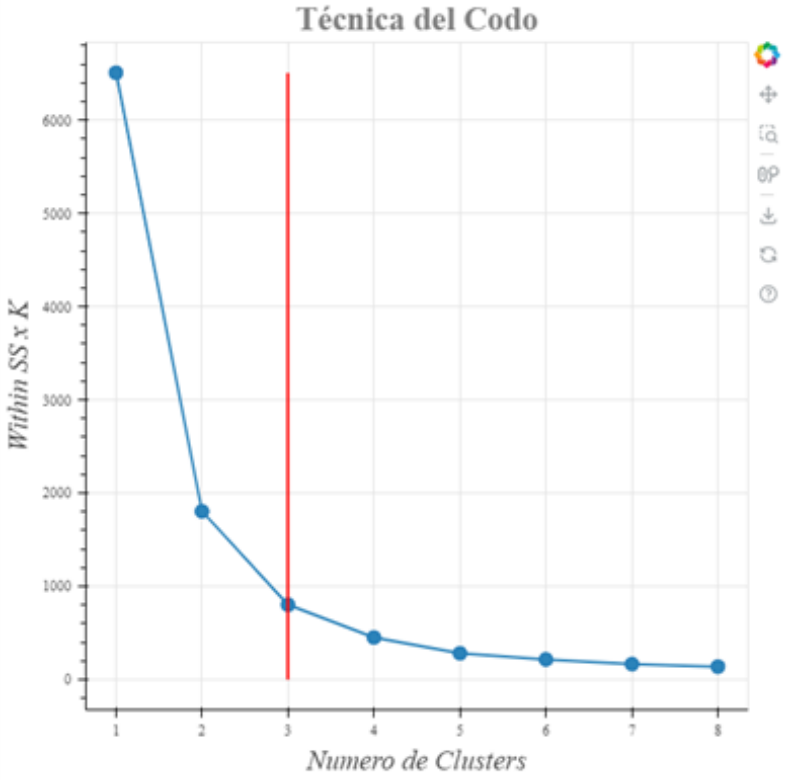

Figura 26. Técnica del codo en K-Means

Explicación:

Para comprender el cálculo de WCSS, hay que considerar en que para un punto del eje X de la gráfica (cuales indican la cantidad de clúster) se hagan cálculos sobre 3 puntos (A, B y C), siendo B el punto analizado, A el punto que está inmediatamente atrás y C el punto inmediatamente adelante, ya que lo que se busca es ver el punto donde se forma el codo para lo cual necesitamos al menos 3 puntos de referencia. Entonces:

Calcular la diferencia de WCSS (eje Y) entre A y B (D1)

Calcular la diferencia de WCSS (eje Y) entre B y C (D2)

Dividir D1/D2

El cociente indica cuantas veces entra D2 en D1, esto permite el puntaje óptimo del codo. Para nuestro caso, se calcula la suma de errores cuadráticos dentro del clúster para diferentes valores de K y se elige la K para la cual la suma de errores cuadráticos comienza a disminuir. Esto es visible como un codo.

En la figura 27, se ejecutó el método del coeficiente Silhouette, en un rango de entre 2 a 8 particiones, el resultado sugiere que el número óptimo de agrupamientos (clusters), para el dataset es 2.

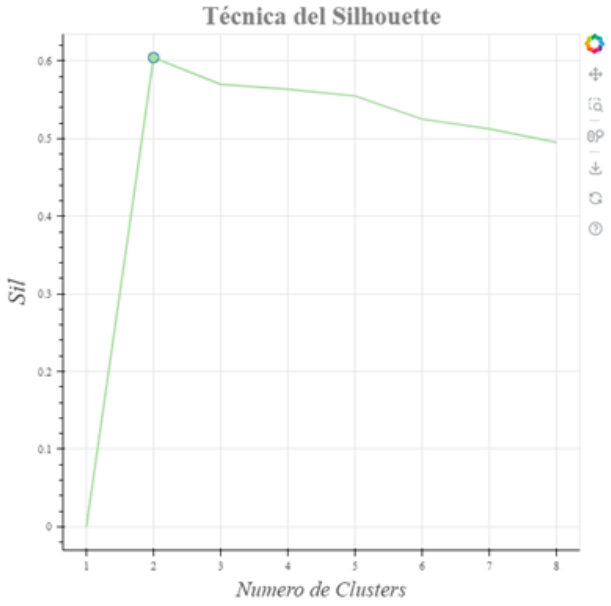

Figura 27. Técnica Promedio Silhouette para algoritmo K-Means.

A continuación, en la figura 28 se procesa el algoritmo K-Means, con parámetro K =2, y se obtiene el siguiente resultado:

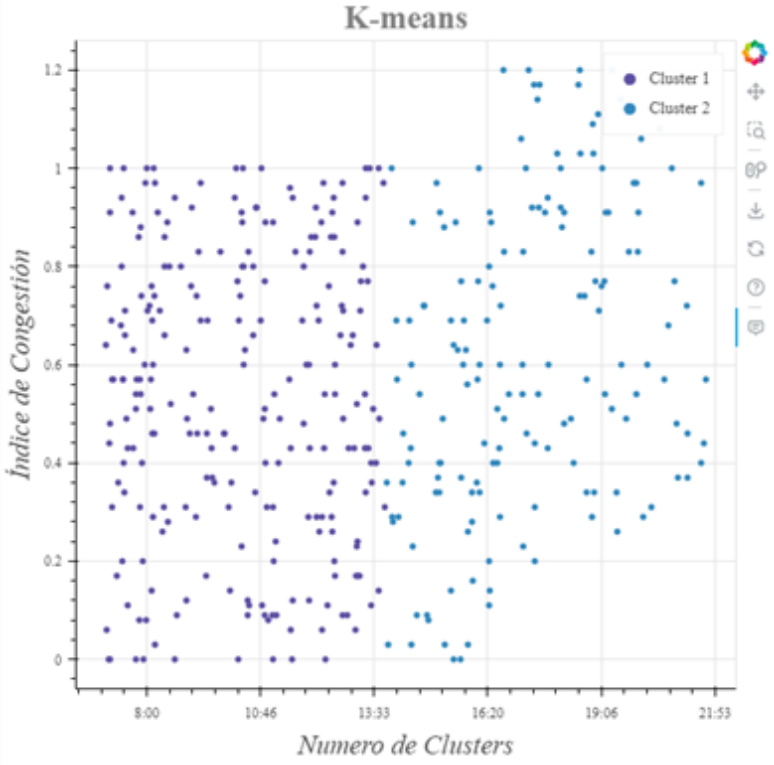

Figura 28. Aplicación del algoritmo K-Means. Modelo k=2.

El objetivo principal es particionar datos en K clústeres (para un K dado). Se contrasta el modelo comprobando las particiones de clústeres, por medio de k. Ejemplo: k=2.

4.3.1.3. Algoritmo DBSCAN

En la figura 29, se muestra la aplicación de la técnica KNN para evaluar distancias de 0 a 1.5, para el dataset en cuestión y se determinó que con un número de MinPts = 37, la distancia épsilon optima es de 0.4.

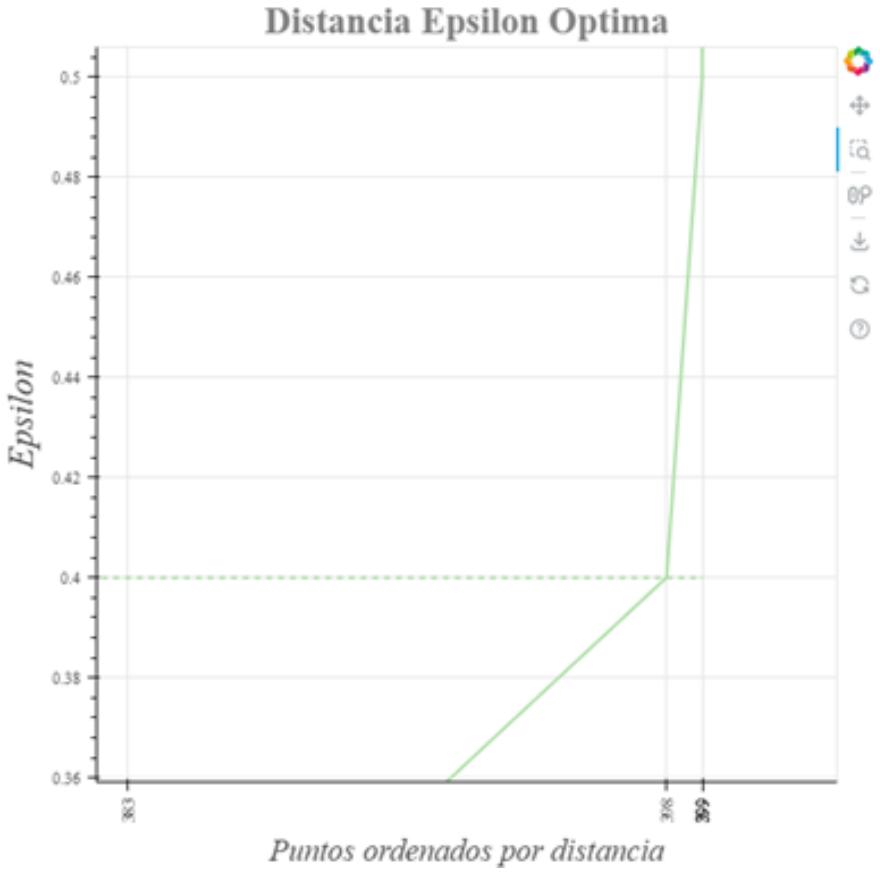

Figura 29. Determinación de distancia épsilon optima.

A continuación, en la figura 30, luego de las iteraciones con un valor del parámetro MinPts de 3 a 8, se puede observar el número común de clústeres calculado es de 2.

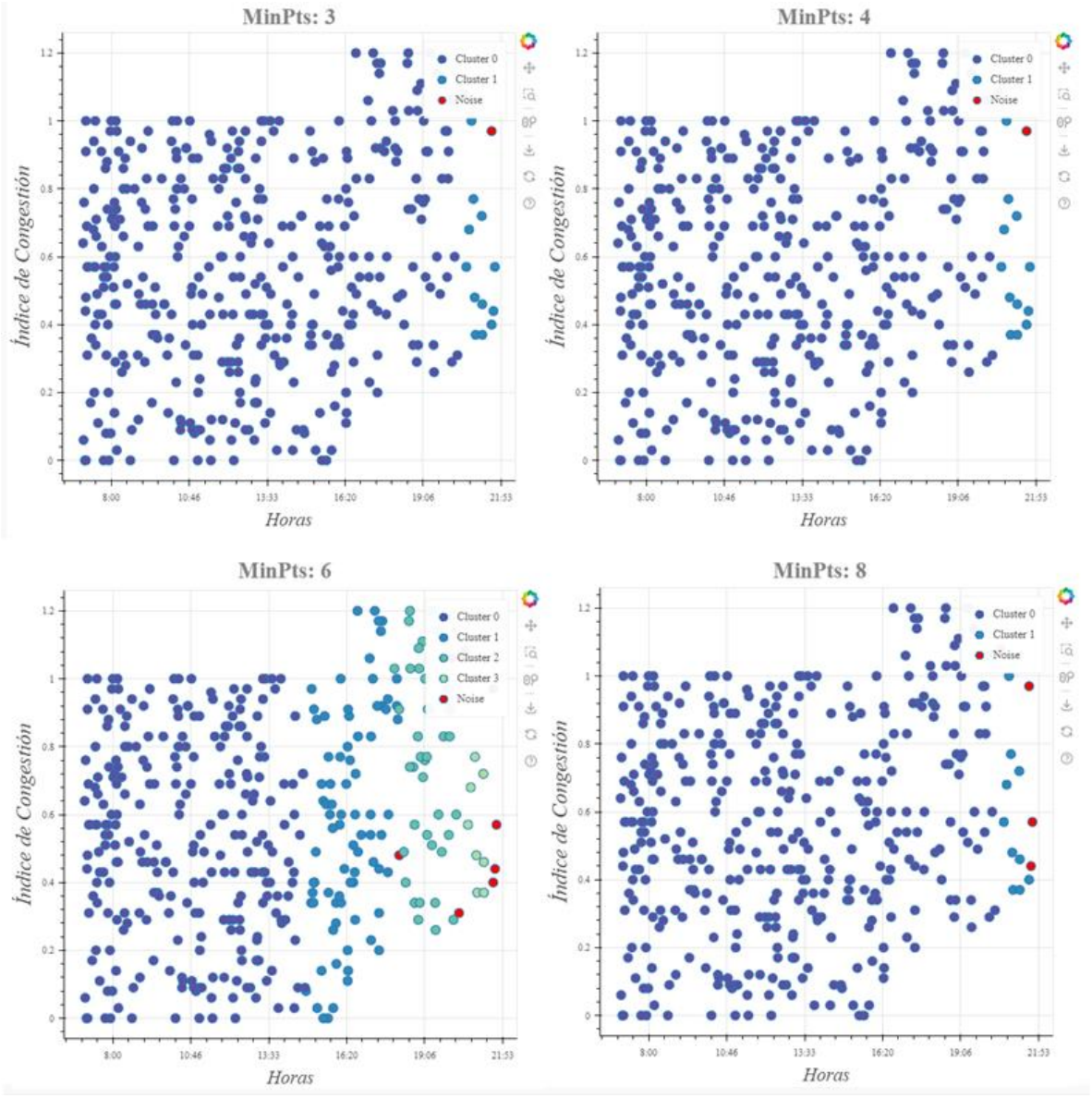

Figura 30. Comparación de Clústeres en relación al número de puntos mínimos en dbscan.

4.3.2. Determinar el nivel de Congestionamiento

Se desarrollo un software basado en las técnicas de agrupamiento k-means y dbscan. El sistema muestra la información referente al nivel de congestión en el que se encuentra cada nodo según 6 horarios aleatorios que se ubican 2 por cada hora punta, por Ejemplo: Figura 31.

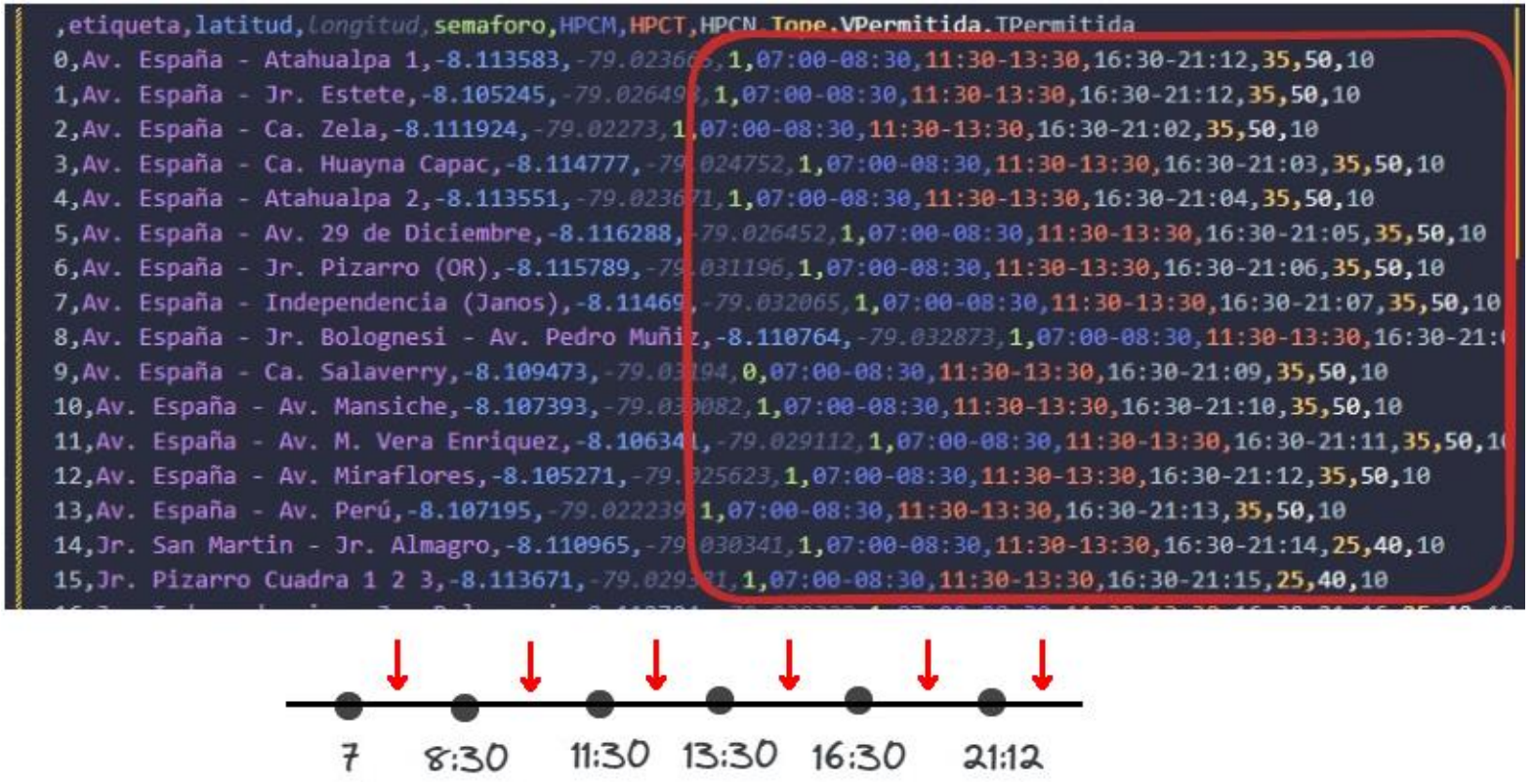

Figura 31. Ejemplo de clasificación de horarios aleatorios.

En la figura 32, se puede observar la generación de horarios para determinar el índice / nivel de congestión en que se encuentra aleatoriamente cada nodo, adecuando para esto una data preprocesada (Anexo E).

	Etiqueta	Hora	Ind. de Congestion	Descripción
1	Av. España - Jr. Estete	07:59	0.34	Fluida
2	Av. España - Jr. Estete	10:01	1.00	Inestable,Congestionada/Intolerable
3	Av. España - Jr. Estete	11:45	0.43	Fluida
4	Av. España - Jr. Estete	14:48	0.54	Fluida
5	Av. España - Jr. Estete	19:15	0.43	Fluida
6	Av. España - Jr. Estete	19:44	0.49	Fluida
7	Av. España - Ca. Zela	07:45	0.60	Fluida
8	Av. España - Ca. Zela	11:05	0.66	Estable/Ligera
9	Av. España - Ca. Zela	11:34	0.91	Inestable,Congestionada/Intolerable
10	Av. España - Ca. Zela	15:01	0.20	Fluida
11	Av. España - Ca. Zela	20:35	0.31	Fluida
12	Av. España - Ca. Zela	11:42	0.89	Pre-inestable/Tolerable
13	Av. España - Ca. Huayna Capac	08:09	0.03	Fluida
14	Av. España - Ca. Huayna Capac	11:24	0.20	Fluida
15	Av. España - Ca. Huayna Capac	13:09	0.80	Estable/Aceptable
16	Av. España - Ca. Huayna Capac	15:38	0.77	Estable/Aceptable
17	Av. España - Ca. Huayna Capac	20:21	0.29	Fluida
18	Av. España - Ca. Huayna Capac	18:23	0.40	Fluida
19	Av. España - Atahualpa 2	07:52	0.83	Pre-inestable/Tolerable
20	Av. España - Atahualpa 2	08:37	0.37	Fluida

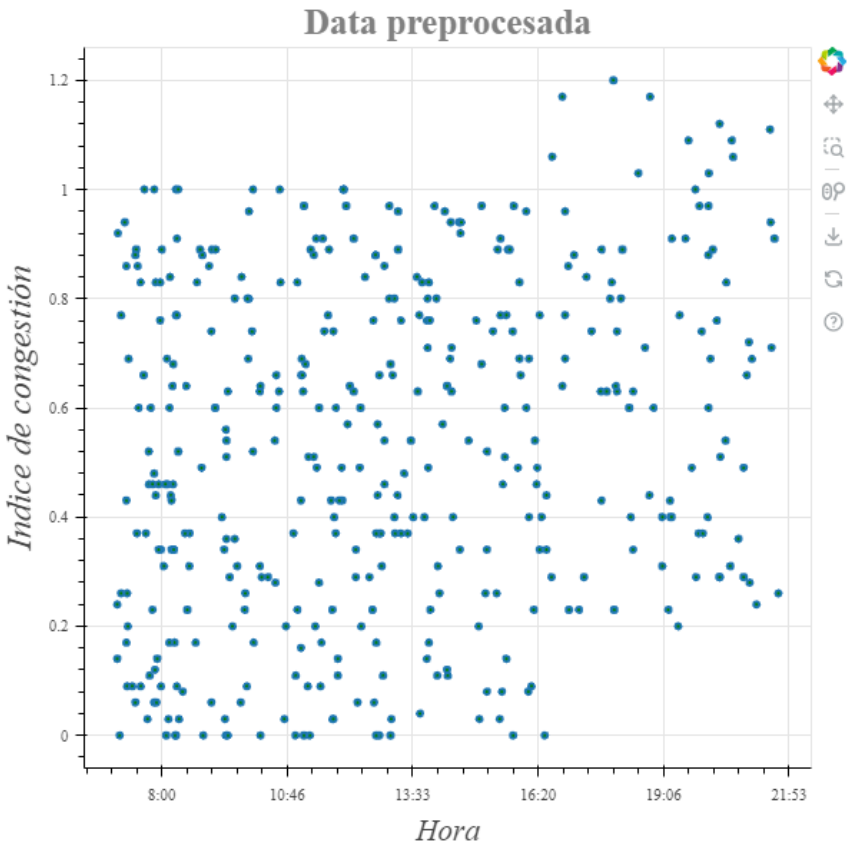

Figura 32. Índice de congestión de los nodos procesados aleatoriamente.

4.4. DISCUSIÓN DE RESULTADOS

4.4.1. Contrastación del Modelo

Utilizando el algoritmo k-means, el valor del promedio Silhouette, se maximiza con 0.6043 con 2 clústers.

Tabla 10. Resultados de evaluación de algoritmos de k-means y dbscan.

Clúster 2	Clúster 3	Clúster 4	Clúster 5	Clúster 6	Clúster 7
0.6043	0.5697	0.5635	0.5548	0.5251	0.5125

Con respecto al algoritmo DBSCAN, el número común de clústers calculado con una distancia épsilon =0.4, es de 2.

El número de clases (grupos) para el dataset evaluado es de 3 (número de horarios punta por turnos: HPM, HPT, HPN), por lo que se puede concluir que los algoritmos K-Means y DBSCAN, han presentado alto grado de bondad y una calidad aceptable en lo que a medidas de similitud se refiere, con respecto al conjunto de datos en lo que fueron aplicados.

4.4.2. Determinar los índices / niveles de congestión

El sistema propuesto permitió mostrar los índices / niveles de congestión considerados en la tabla 11. Estos índices son generados como resultados de las condiciones de las variables como: velocidad permitida, tiempo permitido, semáforo, hora punta mañana, hora punta tarde, hora punta noche, cantidad permitida en cada nodo.
En el sistema gestiona y controla el flujo vehicular reportando por cada nodo un nivel de congestión, usando las técnicas de agrupamiento kmeans y dbscan.

Tabla 11. Descripción y valores de los índices de congestión.

Definición Circulación / Demora	**Índice de Congestión**
Fluida	<0.60
Estable / Ligera	0.61-0.70

Estable / Aceptable	0.71-0.80
Pre-Inestable / Tolerable	0.81-0.90
Inestable, Congestionada / Intolerable	0.91-1.00
Forzada, Congestión Total	>1.00

Fuente: Dirección Metropolitana de Transporte de la ciudad de Trujillo.

- Con el uso del sistema de software se logró determinar la congestión, mediante la clasificación de agrupamientos con los algoritmos k-means y dbscan. Tal como lo indican las figuras 44 y 45, respectivamente.
- Con el uso del sistema de software se logró medir el nivel de congestión. Para lo cual, según reporte de las figuras 44 y 45, se obtiene el punto de referencia 18, tal como indica la figura 34.

Figura 33. Punto de referencia 18 (Jr. Gamarra – Jr. Grau) del sistema vial de la ciudad de Trujillo, situado por el sistema de software.

Fuente: Google Maps.

Figura 34. Nivel de congestión detectado en el punto de referencia 18 (Jr. Gamarra – Jr. Grau) en dataset analizado por el sistema de software.

Explicación de la figura 33:

Para determinar en qué nivel de congestión se encuentra el nodo 18, se usó los datos de la tabla 17, cuyo índice se calcula mediante la cantidad de vehículos que están ocupando la ruta sobre la cantidad máxima permitida (Tope). En el caso de la figura 34, se obtuvo el nivel de congestión:

$$nivel\ de\ congestion = \frac{Cantidad\ vehiculos}{Tope} = \frac{23}{25} = 0.92$$

Ubicándose este valor en la tabla 11, en el siguiente ítem:

Inestable, Congestionada / Intolerable	0.91-1.00

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Se desarrolló un sistema de software (ver anexo F) como interface para interactuar con el modelo computacional.

- La aplicación de modelo computacional en machine learning, a través de los algoritmos kmeans y dbscan permitió la clasificación de los agrupamientos o clústeres para determinar el nivel de índices de congestionamiento vehicular en los diferentes nodos de la ciudad de Trujillo. Tal como se demuestra en las figuras 44 y 45 respectivamente, y se específica en la figura 34.

- Se pudo determinar que, el modelo computacional genera un agrupamiento por cada uno de los 6 niveles de congestión (Tabla 11).

- Se hizo la comparación respectiva de los algoritmos k-means y dbscan. Tal como lo indican los resultados de la tabla 10. Demostrándose que ambos algoritmos permiten un grado de similitud en sus resultados, notando que ambos algoritmos kmeans y dbscan, coinciden cuando tienen en común 2 clústers.

RECOMENDACIONES

Es ideal considerar para trabajos futuros:

- Crear una aplicación colaborativa para dispositivos portables en vehiculos de transporte para monitorear en tiempo real las rutas congestionadas y descongestionadas.

- Se recomienda al Transporte Metropolitano de Trujillo hacer énfasis en realizar capacitaciones a los encargados del análisis del comportamiento vial de las redes más importantes de la ciudad, con respecto al uso específico de este tipo de herramientas simuladoras para proyectos públicos; y al municipio de la ciudad se le hace de conocimiento la necesidad de contar con normativa exclusiva para la planificacion mas optima del trafico vehicular, usando herramientas tecnologicas e inteligentes.

- Si usamos K-means espacial, podríamos segmentar la ciudad en 'K' regiones, y para cada región, tener una idea promedio del volumen y velocidad del tráfico. Esto puede ayudar a las autoridades a tomar decisiones específicas para cada región, como reconfigurar señales de tráfico o establecer zonas de prioridad para el transporte público.

REFERENCIAS BIBLIOGRAFICAS

- Alonso Fernandez-Coppel, I. (13 de Febrero de 2021). Las Coordenadas Geograficas. https://iotsensores.com/libros/coordenadas.pdf

- Cangrejo Aljure, D., & Agudelo, J. (2011). Mineria de Datos Espaciales. Avances en Sistemas e Informatica.

- Cangrejo Aljure, D., Sc, M., & Agudelo, J. G. (2012). Minería de datos espaciales Spatial data mining – An overview. Revista Avances en Sistemas e Informática, 88(3).

- Charme, J., Russo, C., Piergallini, M., Guasch, M., Torriggino, A., & Smail, A. (2016). APLICACIÓN DE MINERIA DE DATOS ESPACIAL EN EL AREA DE SALUD EN LA ZONA DE INFLUENCIA DE LA UNNOBA. SEDICI, p. 135-138. http://sedici.unlp.edu.ar/bitstream/handle/10915/52839/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y

- Epstein, J. (2008). Why Model? Journal of Artificial Societies and Social Simulation, 11(4). http://jasss.soc.surrey.ac.uk/11/4/12.html

- Gonzalo, Á. (s.f.). MACHINE LEARNING, DATA SCIENCE Y ANALÍTICA AVANZADA. https://machinelearningparatodos.com/segmentacion-utilizando-k-means-en-python/#:~:text=Los%20coeficientes%20de%20silueta%20cercanos,est%C3%A9n%20asignadas%20al%20cl%C3%BAster%20err%C3%B3neo.

- Gutiérrez, J. A. (2016). Líneas de investigación en MD en ciencias e ingeniería: Estado del arte y perspectivas. 17.

- Hareesh, B., Lelitha, D., & Jithin, R. (2016). Aplicación de técnicas de minería de datos para la estimación de densidad de tráfico y predicción. ELSEVIER, 321-330.

- Han, J., Kamber, M., & Pei, J. (2012). Data Mining Concepts and Techniques. In J. Han, M. Kamber, & J. Pei (Eds.), Data Mining (Third Edition). Morgan Kaufmann. https://doi.org/https://doi.org/10.1016/B978-0-12-381479-1.00001-0

- Hasperué, W., Estrebou, C. A., Camele, G., López, P., Jimbo Santana, P. R., Reyes Zambrano, G., Lanzarini, L. C., & Fernández Bariviera, A. (2021). Procesamiento inteligente de grandes volúmenes de información y de flujos de datos. XXIII Workshop de Investigadores En Ciencias de La Computación (WICC 2021, Chilecito, La Rioja). http://sedici.unlp.edu.ar/handle/10915/120089

- MEDINA-VELOZ, G. L.-R.-A. (2016). Calibración y selección del modelo de aprendizaje no supervisado K-Medias, de una encuesta sobre factores de riesgo en el consumo de drogas entre estudiantes. Revista de Análisis Cuantitativo y Estadístico, 9. https://www.ecorfan.org/bolivia/researchjournals/Analisis_Cuantitativo_y_Estadistico/vol3num7/Revista_de_Analisis_Cuantitativo_V3_N7_1.pdf

- Otaegui, J., & Santa María, C. (2014). Técnicas de data mining aplicadas a datos de transporte público. XVI Workshop de Investigadores en Ciencias de la Computación (págs. 213-217). Florencio Varela 1903 San Justo Pcia. de Buenos Aires : SEDICI.
- Palacios, J. G. (1995). La organización espacial de la red de carreteras en Aragón aplicación metodológica de la teoría de grafos. Dialnet, 83-102.
- Palomino, E. M. (2016). ANÁLISIS COMPARATIVO DE LA EFICACIA ENTRE LA MEDIDA DE RESTRICCIÓN VEHICULAR POR NÚMERO DE. Lima, Peru.
- Silberschatz , A., Korrth, H. F., & Sudarshan, S. (2002). *Fundamentos de Bases de Datos.* Madrid, España: McGraw-Hii/Interamericana de España.
- Tan , P.-N., Steinbach , M., & Kumar , V. (2006). Classification: basic concepts, decision trees, and model evaluation. Introduction to data mining.
- UPCommons. (2000). Estudio de tráfico de carreteras: densidad, intensidad y velocidad. https://www.ingartek.com/es/tres-claves-para-el-estudio-de-trafico-de-carreteras-la-densidad-la-intensidad-y-la-velocidad/
- Koperski K. Geominer(2003). A knowledge discovery system for spatial databases and geographic information systems. Universidad Simon Fraser, Canadá. 1999. http://db.cs.sfu.ca/GeoMiner/.
- Koperski K. & Han J. (1995) Discovery of Spatial Association Rules in Geographic Information Databases, In Advances in Spatial Databases (Proc. 4th Symp. SSD'95), Portland (USA).
- Koperski K., Han J., Adhikary J. (1997), Spatial Data Mining: Progress and Challenges. Escuela de ciencias de la computación. Universidad Simon Fraser.
- Koperski, K., Han, J., & Stefanovic, N. (1998). An Efficient Two-Step Method for Classification of Spatial Data, In Proc. International Symposium on Spatial Data Handling (SDH'98), pp. 45- 54.
- Kumar, D., Kebede, T., & Tessfaw. (2018). Estudio Comparativo de Algoritmos de Clasificación de Minería de Datos para Predecir la Gravedad de los Accidentes de Tránsito. https://ieeexplore.ieee.org/document/8473265/
- Shekhar S. (2001). Mining for Spatial Patterns. Departamento de ciencias de la computación. Universidad de Minnesota.
- Shekhar S, Huang Y, Wu W, Lu C.T & Chawla S. (2001). What's Spatial about Spatial Data Mining: Three Case Studies. Departamento de ciencias de la computación. Universidad de Minnesota.

- Shekhar S & Chawla S. (2002) “Spatial Databases: A Tour”. Prenticell Hall.

- Vallalta Rueda, J. F. (2016a). CRISP-DM: una metodología para minería de datos en salud - healthdataminer.com.

- Vallalta Rueda, J. F. (2016b). CRISP-DM: una metodología para minería de datos en salud - healthdataminer.com.

- Zeitouni, K. (1999). Extraction de connaissances des bases de données spatiales en accidentologie routière, Laboratorio PRiSM, Universidad de Versalles.

- Zeitouni K. A (2002). Survey on Spatial Data Mining Methods Databases and Statistics Point of Views, Capítulo 13 en “Data Warehousing and Web Engineering”, Shirley Becker Editor, IRM Press, p 229-242.

- Zeitouni K, Yeh L, Aufaure M. (2001) Join indices as a tool for spatial data mining. Laboratorio PRiSM, Universidad de Versalles.

- Zhang T., Ramakrishnan R., Livny M. (1997) .BIRCH: A New Data Clustering Algorithm and Its Applications. Data Mining and Knowledge Discovery 1. pp. 141-182.

ANEXOS

ANEXO A. PUNTOS CRÍTICOS DE CONGESTIONAMIENTO EN EL TRÁNSITO VEHICULAR E INTERSECCIONES SEMAFORIZADAS.

PUNTO	INTERSECCIÓN CON CONGESTIONAMIENTO VEHICULAR	INTERSECCIÓN SEMAFORIZADA	HORA PUNTA CONGESTIONAMIENTO		
			MAÑANA	TARDE	NOCHE
0	Av. España - Atahualpa	Av. España - Atahualpa	07:00 A 08:30	11:30 A 13:30	16:30 A 21:012
1	Av. España - Jr. Estete	Av. España - Jr. Estete	07:00 A 08:30	11:30 A 13:30	16:30 A 21:012
2	Av. España - Ca. Zela	Av. España - Ca. Zela	7:00 A 08:30	11:30 A 13:30	16:30 A 21:02
3	Av. España - Ca. Huayna Cápac	Av. España - Ca. Huayna Cápac	07:00 A 08:30	11:30 A 13:30	16:30 A 21:03
4	Av. España - Atahualpa	Av. España - Atahualpa	07:00 A 08:30	11:30 A 13:30	16:30 A 21:04
5	Av. España - Av. 29 de diciembre	Av. España - Av. 29 de diciembre	07:00 A 08:30	11:30 A 13:30	16:30 A 21:05
6	Av. España - Jr. Pizarro (OR)	Av. España - Jr. Pizarro (OR)	07:00 A 08:30	11:30 A 13:30	16:30 A 21:06
7	Av. España - Independencia (Janos)	Av. España - Independencia (Janos)	07:00 A 08:30	11:30 A 13:30	16:30 A 21:07
8	Av. España - Jr. Bolognesi - Av. Pedro Muñiz	Av. España - Jr. Bolognesi, Av. Pedro Muñiz	07:00 A 08:30	11:30 A 11:30	16:30 A 21:08
9	Av. España - Ca. Salaverry		07:00 A 08:30	11:30 A 13:30	16:30 A 21:09
10	Av. España - Av. Mansiche	Av. España - Av. Mansiche	07:00 A 08:30	11:30 A 13:30	16:30 A 21:010
11	Av. España - Av. M. Vera Enriquez	Av. España - Av. M. Vera Enriquez	07:00 A 08:30	11:30 A 13:30	16:30 A 21:11
12	Av. España - Av. Miraflores	Av. España - Av. Miraflores	07:00 A 08:30	11:30 A 13:30	16:30 A 21:12
13	Av. España - Av. Perú	Av. España - Av. Perú	07:00 A 08:30	11:30 A 13:30	16:30 A 21:13
14	Jr. San Martin - Jr. Almagro	Jr. San Martin - Jr. Almagro	07:00 A 08:30	11:30 A 13:30	16:30 A 21:14
15	Jr. Pizarro Cuadra 1, 2 y 3	Jr. Pizarro Cuadra 1, 2 y 3	07:00 A 08:30	11:30 A 13:30	16:30 A 21:15
16	Jr. Independencia - Jr. Bolognesi	Jr. Independencia - Jr. Bolognesi	07:00 A 08:30	11:30 A 13:30	16:30 A 21:16
17	Jr. Gamarra - Jr. Grau	Jr. Gamarra - Jr. Grau	07:00 A 08:30	11:30 A 13:30	16:30 A 21:17
18	Jr. Bolívar Cuadra 3, 4 y 5		07:00 A 08:30	11:30 A 13:30	16:30 A 21:18
19	Jr. Gamarra - Jr. Bolívar	Jr. Gamarra - Jr. Bolivar	07:00 A 08:30	11:30 A 13:30	16:30 A 21:19
20	Jr. Gamarra - Jr. Ayacucho	Jr. Gamarra - Jr. Ayacucho	07:00 A 08:30	11:30 A 13:30	16:30 A 21:20
21	Jr. Pizarro - Jr. Junín		07:00 A 08:30	11:30 A 13:30	16:30 A 21:21
22	Av. Nicolás de Piérola - Av. Valcárcel	Av. Nicolás de Pierola - Av. Valcárcel	07:00 A 08:30	11:30 A 13:30	16:30 A 21:22
23	Av. América Norte - Av. M. Vera Enríquez	Av. América Norte - Av. M. Vera Enriquez	07:00 A 08:30	11:30 A 13:30	16:30 A 21:23
24	Av. M. Vera Enríquez - (Banco de la Nación)		07:00 A 08:30	11:30 A 13:30	16:30 A 21:24
25	Av. M. Vera Enríquez - Av. 9 de octubre	Av. M. Vera Enriquez - Av. 9 de Octubre	07:00 A 08:30	11:30 A 13:30	16:30 A 21:25
26	Av. América Norte - 8 de octubre (La Hermelinda)	Av. America Norte - 8 de Octubre (las Hermelindas)	07:00 A 08:30	11:30 A 13:30	16:30 A 21:26
27	Av. 8 de octubre - Ca. Pucara (La Hermelinda)		06:30 A 12:30	12:30 A 14:30	
28	Av. América Norte - Av. Miraflores	Av. America Norte - Av. Miraflores	07:00 A 08:30	11:30 A 13:30	16:30 A 21:27
29	Av. América Norte - Av. Perú	Av. América Norte - Av. Perú	07:00 A 08:30	11:30 A 13:30	16:30 A 21:28
30	Av. América Sur - Av. Union	Av. América Sur - Av. Union	07:00 A 08:30	11:30 A 13:30	16:30 A 21:29

31	Av. America Sur - Av. Vallejo	Av. America Sur - Av. Vallejo	07:00 A 08:30	11:30 A 13:30	16:30 A 21:30
32	Av. América Sur - Av. Ricardo Palma	Av. América Sur - Av. Ricardo Palma	07:00 A 08:30	11:30 A 13:30	16:30 A 21:31
33	Av. America Sur - Ca. Santa Cruz	Av. America Sur - Ca. Santa Cruz	07:00 A 08:30	11:30 A 13:30	16:30 A 21:32
34	Av. América Sur - Ca Túpac Yupanqui		07:00 A 08:30	11:30 A 13:30	16:30 A 21:33
35	Av. América Sur - Av. Costa Rica	Av. América Sur - Av. Costa Rica	07:00 A 08:30	11:30 A 13:30	16:30 A 21:34
36	Av. América Sur (UPAO)	Av. América Sur (UPAO)	07:00 A 08:30	11:30 A 13:30	16:30 A 21:35
37	Av. America Sur (Ovalo Larco)		07:00 A 08:30	11:30 A 13:30	16:30 A 21:36
38	Av. América Oeste - Av. Juan Pablo II		07:00 A 08:30	11:30 A 13:30	16:30 A 21:37
39	Av. América Oeste - Av. Antenor Orrego	Av. América Oeste - Av. Antenor Orrego	07:00 A 08:30	11:30 A 13:30	16:30 A 21.38
40	Av. América Oeste (Frente a la Corte)	Av. América Oeste (Frente a la Corte)	07:00 A 08:30	11:30 A 13:30	16:30 A 21:39
41	Av. America Oeste ((Frente al Mall Saga Falabella)		07:00 A 08:30	11:30 A 13:30	16:30 A 21:40
42	Av. América Oeste - Av Mansiche	Av. América Oeste - Av Mansiche	07:00 A 08:30	11:30 A 13:30	16:30 A 21:41
43	Av. Túpac Amaru - Av. Indoamérica		07:00 A 08:30	11:30 A 13:30	16:30 A 21:42
44	Av. Nicolás de Pierola - Av. Indoamérica	Av. Nicolás de Pierola - Av. Indoamérica	07:00 A 08:30	11:30 A 13:30	16:30 A 21:43
45	Av. Villareal - Av. 8 de octubre (Mercado las Hermelindas)	Av. Villareal - Av. 8 de octubre (Mercado las Hermelindas)	07:00 A 08:30	11:30 A 13:30	16:30 A 21:44
46	Av. Villareal - Av. Peru		07:00 A 08:30	11:30 A 13:30	16:30 A 21:46
47	Av. Villareal - Av. Vallejo	Av. Villareal Av. Vallejo	07:00 A 08:30	11:30 A 13:30	16 30 A 21:47
48	Av. Larco - Av. Húsares de Junín	Av. Larco - Av. Húsares de Junín	07:00 A 08:30	11:30 A 13:30	16:30 A 21:48
49	Av. Fátima - Av. Húsares de Junín	Av. Fátima - Av. Húsares de Junín	07:00 A 08:30	11:30 A 13:30	-16:10 A 21:49
50	Av. Fátima - Av. Larco	Av. Fátima - Av. Larco	07:00 A 08:30	11:30 A 13:30	16.30 A 21 50
51	Ca. Agua Marina (Frente a la ULADECH)		07:00 A 08:30	11:30 A 13:30	16:30 A 21:51
52	Av. Perú Cdra. 1, 2, 3 y 4	Av. Perú Cdra. 1, 2, 3 y 4	07:00 A 08:30	11:30 A 13:30	16:30 A 21 52
53	Ca. Rimac (Mercado Unión)		07:00 A 08:30	11:30 A 13:30	16:30 A 21:53
54	Ca. Amazonas - Ca. Rimac		07:00 A 08:30	11:30 A 13:30	16:30 A 21:54
55	Av. Ejército (Frente a la UPN)		07:00 A 08:30	11:30 A 13:30	16:30 A 21:55
56	Av. José María Eugeren - Ca. Balboa		07:00 A 08:30	11:30 A 13:30	16:30 A 21:56
57	Ca. Balboa - Psje. Albarracin		07:00 A 08:30	11:30 A 13:30	16:30 A 21:57
58	Ca. Sinchi Roca (Mercado Mayorista)		07:00 A 08:30	11:30 A 13:30	16:30 A 21:58
59	Ca. Sinchi Roca - Ca. Zela		07:00 A 08:30	11:30 A 13:30	16:30 A 21:59
60	Av. La Marina - (Terminal América Expresa)		07:00 A 08:30	11:30 A 13:30	16:30 A 21.60
61	Av. Prolg. Unión (Plaza Vea)		07:00 A 08:30	11:30 A 13:30	16:30 A 21:61
62	Av. Mansiche Mall Aventura Plaza	Av. Mansiche Mall Aventura Plaza	07:00 A 08:30	11:30 A 13:30	16:30 A 21:61
63	Av. Mansiche - Ca. Los Zafiros (Hospital Regional)	Av. Mansiche Ca. Los Zafiros (Hospital Regional)	07:00 A 08:30	11:30 A 13:30	16:30 A 21:63
64	Av. Nápoles - Ca. Salaverry	Av. Nápoles - Ca. Salaverry	07:00 A 08:30	11:30 A 13:30	16:30 A 21:64
65	Av. Juan Pablo II (Puerta Principal UNT)	Av. Juan Pablo II (Puerta Principal UNT)	07:00 A 08:30	11:30 A 13:30	16:30 A 21:65

Fuente: Observatorio de la Movilidad de Transporte de Trujillo.

ANEXO B. DATA PREPROCESADA

	Etiqueta	Hora	Ind. Congestion
1	Av. España - Jr. Estete	07:53	0.46
2	Av. España - Jr. Estete	09:48	0.83
3	Av. España - Jr. Estete	13:24	0.80
4	Av. España - Jr. Estete	13:52	0.63
5	Av. España - Jr. Estete	19:29	0.60
6	Av. España - Jr. Estete	12:08	0.49
7	Av. España - Ca. Zela	08:05	0.77
8	Av. España - Ca. Zela	10:23	0.97
9	Av. España - Ca. Zela	12:58	0.23
10	Av. España - Ca. Zela	16:16	0.89
11	Av. España - Ca. Zela	18:51	0.46
12	Av. España - Ca. Zela	13:38	0.51
13	Av. España - Ca. Huayna Capac	08:07	0.11
14	Av. España - Ca. Huayna Capac	11:12	0.37
15	Av. España - Ca. Huayna Capac	12:42	0.14
16	Av. España - Ca. Huayna Capac	14:14	0.37
17	Av. España - Ca. Huayna Capac	19:54	1.00
18	Av. España - Ca. Huayna Capac	18:43	0.94
19	Av. España - Atahualpa 2	07:03	0.43
20	Av. España - Atahualpa 2	08:56	0.23
21	Av. España - Atahualpa 2	12:59	0.49
22	Av. España - Atahualpa 2	14:36	0.37
23	Av. España - Atahualpa 2	19:08	0.74
24	Av. España - Atahualpa 2	11:00	0.80
25	Av. España - Av. 29 de Diciembre	08:29	0.83
26	Av. España - Av. 29 de Diciembre	10:50	1.00
27	Av. España - Av. 29 de Diciembre	13:07	0.91
28	Av. España - Av. 29 de Diciembre	13:49	0.00
29	Av. España - Av. 29 de Diciembre	18:28	0.60
30	Av. España - Av. 29 de Diciembre	08:58	0.89
31	Av. España - Jr. Pizarro (OR)	07:28	0.43
32	Av. España - Jr. Pizarro (OR)	10:12	0.17
33	Av. España - Jr. Pizarro (OR)	12:18	0.83
34	Av. España - Jr. Pizarro (OR)	14:03	0.34
35	Av. España - Jr. Pizarro (OR)	18:55	0.80
36	Av. España - Jr. Pizarro (OR)	11:20	0.00
37	Av. España - Independencia (Janos)	08:02	0.23
38	Av. España - Independencia (Janos)	09:10	0.80
39	Av. España - Independencia (Janos)	12:41	0.20
40	Av. España - Independencia (Janos)	15:46	0.06
41	Av. España - Independencia (Janos)	17:34	1.09
42	Av. España - Independencia (Janos)	08:41	0.89
43	Av. España - Jr. Bolognesi - Av. Pedro Muñiz	08:08	0.91
44	Av. España - Jr. Bolognesi - Av. Pedro Muñiz	10:52	0.54
45	Av. España - Jr. Bolognesi - Av. Pedro Muñiz	12:29	0.00
46	Av. España - Jr. Bolognesi - Av. Pedro Muñiz	15:06	0.57
47	Av. España - Jr. Bolognesi - Av. Pedro Muñiz	17:50	0.60
48	Av. España - Jr. Bolognesi - Av. Pedro Muñiz	17:10	0.20
49	Av. España - Ca. Salaverry	07:35	0.43
50	Av. España - Ca. Salaverry	10:02	0.83
51	Av. España - Ca. Salaverry	11:53	0.03
52	Av. España - Ca. Salaverry	14:51	0.66
53	Av. España - Ca. Salaverry	19:57	0.29
54	Av. España - Ca. Salaverry	10:51	0.97
55	Av. España - Av. Mansiche	07:47	0.60
56	Av. España - Av. Mansiche	09:40	0.29
57	Av. España - Av. Mansiche	11:42	0.23
58	Av. España - Av. Mansiche	15:04	0.11
59	Av. España - Av. Mansiche	16:36	0.49
60	Av. España - Av. Mansiche	12:27	0.34
61	Av. España - Av. M. Vera Enriquez	08:22	0.23
62	Av. España - Av. M. Vera Enriquez	10:19	0.29
63	Av. España - Av. M. Vera Enriquez	11:35	0.94
64	Av. España - Av. M. Vera Enriquez	14:15	0.06
65	Av. España - Av. M. Vera Enriquez	20:20	0.51
66	Av. España - Av. M. Vera Enriquez	19:32	0.31
67	Av. España - Av. Miraflores	07:20	1.00
68	Av. España - Av. Miraflores	09:46	0.43
69	Av. España - Av. Miraflores	11:40	0.11
70	Av. España - Av. Miraflores	15:05	0.20
71	Av. España - Av. Miraflores	18:14	0.80
72	Av. España - Av. Miraflores	11:21	0.23
73	Av. España - Av. Perú	07:11	0.94
74	Av. España - Av. Perú	11:09	0.97
75	Av. España - Av. Perú	13:14	0.09
76	Av. España - Av. Perú	15:10	1.00
77	Av. España - Av. Perú	20:56	0.23
78	Av. España - Av. Perú	20:06	0.71
79	Jr. San Martin - Jr. Almagro	08:25	0.44
80	Jr. San Martin - Jr. Almagro	10:29	0.08

	Etiqueta	Hora	Ind. Congestion
81	Jr. San Martin - Jr. Almagro	12:46	0.16
82	Jr. San Martin - Jr. Almagro	15:25	0.96
83	Jr. San Martin - Jr. Almagro	17:41	0.56
84	Jr. San Martin - Jr. Almagro	20:52	0.76
85	Jr. Pizarro Cuadra 1 2 3	07:21	0.88
86	Jr. Pizarro Cuadra 1 2 3	10:35	0.32
87	Jr. Pizarro Cuadra 1 2 3	12:47	0.32
88	Jr. Pizarro Cuadra 1 2 3	15:27	0.56
89	Jr. Pizarro Cuadra 1 2 3	18:04	0.92
90	Jr. Pizarro Cuadra 1 2 3	08:31	0.60
91	Jr. Independencia - Jr. Bolognesi	07:05	0.04
92	Jr. Independencia - Jr. Bolognesi	11:05	0.24
93	Jr. Independencia - Jr. Bolognesi	13:27	0.28
94	Jr. Independencia - Jr. Bolognesi	15:55	0.44
95	Jr. Independencia - Jr. Bolognesi	19:03	1.20
96	Jr. Independencia - Jr. Bolognesi	18:18	0.56
97	Jr. Gamarra - Jr. Grau	08:01	0.20
98	Jr. Gamarra - Jr. Grau	08:47	0.00
99	Jr. Gamarra - Jr. Grau	11:45	0.80
100	Jr. Gamarra - Jr. Grau	15:24	0.28
101	Jr. Gamarra - Jr. Grau	18:52	0.52
102	Jr. Gamarra - Jr. Grau	19:59	1.08
103	Jr. Bolivar Cuadra 3 4 5	07:23	0.52
104	Jr. Bolivar Cuadra 3 4 5	10:06	0.36
105	Jr. Bolivar Cuadra 3 4 5	11:58	0.00
106	Jr. Bolivar Cuadra 3 4 5	15:44	0.60
107	Jr. Bolivar Cuadra 3 4 5	21:15	0.64
108	Jr. Bolivar Cuadra 3 4 5	09:51	0.04
109	Jr. Gamarra - Jr. Bolivar	07:55	0.20
110	Jr. Gamarra - Jr. Bolivar	09:09	0.84
111	Jr. Gamarra - Jr. Bolivar	13:05	0.08
112	Jr. Gamarra - Jr. Bolivar	15:22	0.60
113	Jr. Gamarra - Jr. Bolivar	17:45	0.64
114	Jr. Gamarra - Jr. Bolivar	18:37	0.20
115	Jr. Gamarra - Jr. Ayacucho	08:10	0.36
116	Jr. Gamarra - Jr. Ayacucho	08:36	0.48
117	Jr. Gamarra - Jr. Ayacucho	11:43	0.96
118	Jr. Gamarra - Jr. Ayacucho	16:29	0.16
119	Jr. Gamarra - Jr. Ayacucho	18:49	0.36
120	Jr. Gamarra - Jr. Ayacucho	13:48	0.64
121	Jr. Pizarro - Jr. Junin	07:49	0.92
122	Jr. Pizarro - Jr. Junin	09:58	0.84
123	Jr. Pizarro - Jr. Junin	13:16	0.40
124	Jr. Pizarro - Jr. Junin	15:35	0.16
125	Jr. Pizarro - Jr. Junin	19:01	0.36
126	Jr. Pizarro - Jr. Junin	16:10	0.96
127	Av. Nicolás de Pierola - Av. Valcárcel	08:12	0.29
128	Av. Nicolás de Pierola - Av. Valcárcel	10:21	0.34
129	Av. Nicolás de Pierola - Av. Valcárcel	13:26	0.54
130	Av. Nicolás de Pierola - Av. Valcárcel	14:34	0.37
131	Av. Nicolás de Pierola - Av. Valcárcel	20:24	0.54
132	Av. Nicolás de Pierola - Av. Valcárcel	19:38	0.86
133	Av. América Norte - Av. M. Vera Enriquez	08:14	0.00
134	Av. América Norte - Av. M. Vera Enriquez	09:08	0.63
135	Av. América Norte - Av. M. Vera Enriquez	12:17	0.51
136	Av. América Norte - Av. M. Vera Enriquez	13:50	0.60
137	Av. América Norte - Av. M. Vera Enriquez	21:14	0.77
138	Av. América Norte - Av. M. Vera Enriquez	17:25	0.83
139	Av. M. Vera Enriquez - (Banco de la Nación)	07:01	0.89
140	Av. M. Vera Enriquez - (Banco de la Nación)	08:50	0.83
141	Av. M. Vera Enriquez - (Banco de la Nación)	13:00	0.46
142	Av. M. Vera Enriquez - (Banco de la Nación)	14:57	0.66
143	Av. M. Vera Enriquez - (Banco de la Nación)	19:24	0.97
144	Av. M. Vera Enriquez - (Banco de la Nación)	17:59	0.94
145	Av. M. Vera Enriquez - Av. 9 de Octubre	07:33	0.74
146	Av. M. Vera Enriquez - Av. 9 de Octubre	11:23	0.49
147	Av. M. Vera Enriquez - Av. 9 de Octubre	12:39	0.83
148	Av. M. Vera Enriquez - Av. 9 de Octubre	15:50	0.80
149	Av. M. Vera Enriquez - Av. 9 de Octubre	18:50	0.71
150	Av. M. Vera Enriquez - Av. 9 de Octubre	19:14	1.20
151	Av. America Norte - 8 de Octubre (las Hermelindas)	07:26	0.91
152	Av. America Norte - 8 de Octubre (las Hermelindas)	10:40	0.57
153	Av. America Norte - 8 de Octubre (las Hermelindas)	12:07	0.49
154	Av. America Norte - 8 de Octubre (las Hermelindas)	14:05	0.09
155	Av. America Norte - 8 de Octubre (las Hermelindas)	19:36	0.91
156	Av. America Norte - 8 de Octubre (las Hermelindas)	19:48	0.71
157	Av. 8 de Octubre - Ca. Pucara (Las hermelindas)	07:19	0.03
158	Av. 8 de Octubre - Ca. Pucara (Las hermelindas)	13:30	0.86
159	Av. 8 de Octubre - Ca. Pucara (Las hermelindas)	10:14	0.97
160	Av. America Norte - Av. Miraflores	07:14	0.29

	Etiqueta	Hora	Ind. Congestion
161	Av. America Norte - Av. Miraflores	11:28	0.37
162	Av. America Norte - Av. Miraflores	12:53	0.00
163	Av. America Norte - Av. Miraflores	14:06	0.69
164	Av. America Norte - Av. Miraflores	20:01	0.20
165	Av. America Norte - Av. Miraflores	17:31	1.17
166	Av. América Norte - Av. Perú	07:22	0.46
167	Av. América Norte - Av. Perú	09:53	0.34
168	Av. América Norte - Av. Perú	11:50	0.89
169	Av. América Norte - Av. Perú	14:19	0.03
170	Av. América Norte - Av. Perú	16:59	0.80
171	Av. América Norte - Av. Perú	12:43	0.31
172	Av. América Sur - Av. Union	07:15	0.97
173	Av. América Sur - Av. Union	09:44	0.34
174	Av. América Sur - Av. Union	12:06	0.23
175	Av. América Sur - Av. Union	16:06	0.31
176	Av. América Sur - Av. Union	20:57	0.94
177	Av. América Sur - Av. Union	09:18	0.43
178	Av. America Sur - Av. Vallejo	07:43	0.74
179	Av. America Sur - Av. Vallejo	08:38	0.09
180	Av. America Sur - Av. Vallejo	13:29	0.91
181	Av. America Sur - Av. Vallejo	15:57	0.69
182	Av. America Sur - Av. Vallejo	20:46	0.31
183	Av. America Sur - Av. Vallejo	14:00	0.77
184	Av. América Sur - Av. Ricardo Palma	08:24	0.89
185	Av. América Sur - Av. Ricardo Palma	11:26	0.83
186	Av. América Sur - Av. Ricardo Palma	12:10	0.83
187	Av. América Sur - Av. Ricardo Palma	15:33	0.74
188	Av. América Sur - Av. Ricardo Palma	21:00	0.40
189	Av. América Sur - Av. Ricardo Palma	14:30	0.00
190	Av. América Sur - Ca. Santa Cruz	08:17	0.74
191	Av. América Sur - Ca. Santa Cruz	09:31	0.57
192	Av. América Sur - Ca. Santa Cruz	12:00	0.03
193	Av. América Sur - Ca. Santa Cruz	15:32	0.54
194	Av. América Sur - Ca. Santa Cruz	17:51	0.26
195	Av. América Sur - Ca. Santa Cruz	12:02	0.83
196	Av. América Sur - Ca Túpac Yupanqui	07:00	0.74
197	Av. América Sur - Ca Túpac Yupanqui	08:45	0.17
198	Av. América Sur - Ca Túpac Yupanqui	11:57	0.89
199	Av. América Sur - Ca Túpac Yupanqui	15:58	0.57
200	Av. América Sur - Ca Túpac Yupanqui	20:49	0.20
201	Av. América Sur - Ca Túpac Yupanqui	09:15	0.46
202	Av. América Sur - Av. Costa Rica	07:59	0.00
203	Av. América Sur - Av. Costa Rica	09:23	0.20
204	Av. América Sur - Av. Costa Rica	12:20	0.06
205	Av. América Sur - Av. Costa Rica	15:29	0.86
206	Av. América Sur - Av. Costa Rica	21:20	0.54
207	Av. América Sur - Av. Costa Rica	17:48	0.77
208	Av. América Sur (UPAO)	07:58	0.00
209	Av. América Sur (UPAO)	09:16	0.83
210	Av. América Sur (UPAO)	11:34	0.17
211	Av. América Sur (UPAO)	14:17	0.17
212	Av. América Sur (UPAO)	20:18	0.91
213	Av. América Sur (UPAO)	09:52	0.20
214	Av. América Sur (Ovalo Larco)	07:10	0.49
215	Av. América Sur (Ovalo Larco)	11:01	0.17
216	Av. América Sur (Ovalo Larco)	12:36	0.43
217	Av. América Sur (Ovalo Larco)	15:17	0.00
218	Av. América Sur (Ovalo Larco)	20:58	1.17
219	Av. América Sur (Ovalo Larco)	13:28	1.00
220	Av. América Oeste - Av. Juan Pablo II	07:37	0.51
241	Av. América Oeste (Frente al Mall Saga Falabella)	14:22	0.97
242	Av. América Oeste (Frente al Mall Saga Falabella)	20:21	0.20
243	Av. América Oeste (Frente al Mall Saga Falabella)	07:30	1.00
244	Av. América Oeste - Av Mansiche	07:48	0.40
245	Av. América Oeste - Av Mansiche	10:05	0.86
246	Av. América Oeste - Av Mansiche	13:06	0.43
247	Av. América Oeste - Av Mansiche	15:12	0.66
248	Av. América Oeste - Av Mansiche	20:12	0.29
249	Av. América Oeste - Av Mansiche	21:17	0.74
250	Av. Túpac Amaru - Av. Indoamérica	08:09	0.14
251	Av. Túpac Amaru - Av. Indoamérica	10:24	0.91
252	Av. Túpac Amaru - Av. Indoamérica	11:51	0.74
253	Av. Túpac Amaru - Av. Indoamérica	16:24	0.37
254	Av. Túpac Amaru - Av. Indoamérica	16:54	0.20
255	Av. Túpac Amaru - Av. Indoamérica	21:29	1.14
256	Av. Nicolás de Pierola - Av. Indoamérica	07:51	0.00
257	Av. Nicolás de Pierola - Av. Indoamérica	08:35	0.97
258	Av. Nicolás de Pierola - Av. Indoamérica	12:26	0.20
259	Av. Nicolás de Pierola - Av. Indoamérica	13:53	0.74
260	Av. Nicolás de Pierola - Av. Indoamérica	20:14	0.86

	Etiqueta	Hora	Ind. Congestion
261	Av. Nicolás de Pierola - Av. Indoamérica	10:00	0.09
262	Av. Villareal - Av. 8 de Octubre (Mercado las Hermelindas)	07:36	0.57
263	Av. Villareal - Av. 8 de Octubre (Mercado las Hermelindas)	09:24	0.34
264	Av. Villareal - Av. 8 de Octubre (Mercado las Hermelindas)	12:34	1.00
265	Av. Villareal - Av. 8 de Octubre (Mercado las Hermelindas)	15:11	0.26
266	Av. Villareal - Av. 8 de Octubre (Mercado las Hermelindas)	17:49	1.03
267	Av. Villareal - Av. 8 de Octubre (Mercado las Hermelindas)	18:20	1.00
268	Av. Villareal - Av. Peru	08:00	0.66
269	Av. Villareal - Av. Peru	11:16	0.97
270	Av. Villareal - Av. Peru	12:12	0.49
271	Av. Villareal - Av. Peru	14:13	0.71
272	Av. Villareal - Av. Peru	18:41	1.14
273	Av. Villareal - Av. Peru	13:21	0.09
274	Av. Villareal - Av. Vallejo	08:13	0.71
275	Av. Villareal - Av. Vallejo	11:24	0.77
276	Av. Villareal - Av. Vallejo	12:15	0.06
277	Av. Villareal - Av. Vallejo	14:04	0.34
278	Av. Villareal - Av. Vallejo	16:37	0.57
279	Av. Villareal - Av. Vallejo	13:37	0.74
280	Av. Larco - Av. Húsares de Junín	07:25	0.63
281	Av. Larco - Av. Húsares de Junín	09:49	0.29
282	Av. Larco - Av. Húsares de Junín	12:09	0.11
283	Av. Larco - Av. Húsares de Junín	13:34	0.11
284	Av. Larco - Av. Húsares de Junín	20:48	1.06
285	Av. Larco - Av. Húsares de Junín	14:54	0.17
286	Av. Fátima - Av. Húsares de Junín	07:41	0.31
287	Av. Fátima - Av. Húsares de Junín	10:31	0.63
288	Av. Fátima - Av. Húsares de Junín	13:15	0.80
289	Av. Fátima - Av. Húsares de Junín	13:56	0.97
290	Av. Fátima - Av. Húsares de Junín	17:32	0.49
291	Av. Fátima - Av. Húsares de Junín	10:47	0.66
292	Av. Fátima - Av. Larco	08:04	0.14
293	Av. Fátima - Av. Larco	09:21	0.74
294	Av. Fátima - Av. Larco	11:30	0.26
295	Av. Fátima - Av. Larco	15:52	1.00
296	Av. Fátima - Av. Larco	19:26	1.11
297	Av. Fátima - Av. Larco	17:22	0.46
298	Ca. Agua Marina (Frente a la ULADECH)	07:09	0.68
299	Ca. Agua Marina (Frente a la ULADECH)	11:04	0.12
300	Ca. Agua Marina (Frente a la ULADECH)	12:56	0.68
301	Ca. Agua Marina (Frente a la ULADECH)	14:31	0.12
302	Ca. Agua Marina (Frente a la ULADECH)	19:22	0.36
303	Ca. Agua Marina (Frente a la ULADECH)	15:18	0.76
304	Av. Perú Cdra. 1 2 3 y 4	07:42	0.17
305	Av. Perú Cdra. 1 2 3 y 4	08:33	0.49
306	Av. Perú Cdra. 1 2 3 y 4	12:57	0.23
307	Av. Perú Cdra. 1 2 3 y 4	15:38	0.63
308	Av. Perú Cdra. 1 2 3 y 4	17:17	0.40
309	Av. Perú Cdra. 1 2 3 y 4	11:25	0.29
310	Ca. Rimac (Mercado Unión)	07:29	0.52
311	Ca. Rimac (Mercado Unión)	11:13	0.08
312	Ca. Rimac (Mercado Unión)	12:51	1.00
313	Ca. Rimac (Mercado Unión)	14:52	0.04
314	Ca. Rimac (Mercado Unión)	19:55	0.84
315	Ca. Rimac (Mercado Unión)	13:08	0.12
316	Ca. Amazonas - Ca. Rímac	08:18	0.92
317	Ca. Amazonas - Ca. Rímac	11:11	0.04
318	Ca. Amazonas - Ca. Rímac	11:39	0.28
319	Ca. Amazonas - Ca. Rímac	13:40	0.96
320	Ca. Amazonas - Ca. Rímac	19:00	1.00
321	Ca. Amazonas - Ca. Rímac	09:14	0.48
322	Av. Ejército (Frente a la UPN)	07:07	0.69
323	Av. Ejército (Frente a la UPN)	09:28	0.06
324	Av. Ejército (Frente a la UPN)	12:44	0.46
325	Av. Ejército (Frente a la UPN)	14:25	0.74
326	Av. Ejército (Frente a la UPN)	18:56	0.29
327	Av. Ejército (Frente a la UPN)	18:22	0.63
328	Av. José María Eugeren - Ca. Balboa	07:13	0.57
329	Av. José María Eugeren - Ca. Balboa	09:35	0.49
330	Av. José María Eugeren - Ca. Balboa	13:09	0.46
331	Av. José María Eugeren - Ca. Balboa	16:13	0.71
332	Av. José María Eugeren - Ca. Balboa	16:50	0.77
333	Av. José María Eugeren - Ca. Balboa	15:39	0.97
334	Ca. Balboa - Psje. Albarracin	07:40	0.60
335	Ca. Balboa - Psje. Albarracin	09:32	0.76
336	Ca. Balboa - Psje. Albarracin	12:25	0.60
337	Ca. Balboa - Psje. Albarracin	15:20	0.32
338	Ca. Balboa - Psje. Albarracin	21:54	0.64
339	Ca. Balboa - Psje. Albarracin	07:57	0.60
340	Ca. Sinchi Roca (Mercado Mayorista)	07:32	0.52

	Etiqueta	Hora	Ind. Congestion
341	Ca. Sinchi Roca (Mercado Mayorista)	09:30	0.40
342	Ca. Sinchi Roca (Mercado Mayorista)	13:23	0.96
343	Ca. Sinchi Roca (Mercado Mayorista)	15:07	0.20
344	Ca. Sinchi Roca (Mercado Mayorista)	16:57	0.24
345	Ca. Sinchi Roca (Mercado Mayorista)	14:43	1.00
346	Ca. Sinchi Roca - Ca. Zela	08:20	0.08
347	Ca. Sinchi Roca - Ca. Zela	10:09	0.96
348	Ca. Sinchi Roca - Ca. Zela	13:10	0.40
349	Ca. Sinchi Roca - Ca. Zela	15:45	0.88
350	Ca. Sinchi Roca - Ca. Zela	19:15	0.84
351	Ca. Sinchi Roca - Ca. Zela	15:00	0.56
352	Av. La Marina - (Terminal América Expresa)	07:02	0.63
353	Av. La Marina - (Terminal América Expresa)	11:07	0.40
354	Av. La Marina - (Terminal América Expresa)	13:11	0.20
355	Av. La Marina - (Terminal América Expresa)	14:40	0.80
356	Av. La Marina - (Terminal América Expresa)	16:33	1.14
357	Av. La Marina - (Terminal América Expresa)	16:32	0.94
358	Av. Prolg. Unión (Plaza Vea)	08:03	0.37
359	Av. Prolg. Unión (Plaza Vea)	10:13	0.80
360	Av. Prolg. Unión (Plaza Vea)	12:23	0.77
361	Av. Prolg. Unión (Plaza Vea)	13:43	0.46
362	Av. Prolg. Unión (Plaza Vea)	21:01	0.34
363	Av. Prolg. Unión (Plaza Vea)	09:50	0.49
364	Av. Mansiche - Mall Aventura Plaza	08:15	0.37
365	Av. Mansiche - Mall Aventura Plaza	09:36	0.86
366	Av. Mansiche - Mall Aventura Plaza	11:48	0.34
367	Av. Mansiche - Mall Aventura Plaza	14:56	0.63
368	Av. Mansiche - Mall Aventura Plaza	17:36	0.34
369	Av. Mansiche - Mall Aventura Plaza	21:07	0.86
370	Av. Mansiche - Ca. Los Zafiros (Hospital Regional)	07:44	1.00
371	Av. Mansiche - Ca. Los Zafiros (Hospital Regional)	08:51	0.23
372	Av. Mansiche - Ca. Los Zafiros (Hospital Regional)	12:49	0.60
373	Av. Mansiche - Ca. Los Zafiros (Hospital Regional)	15:49	0.74
374	Av. Mansiche - Ca. Los Zafiros (Hospital Regional)	17:58	0.74
375	Av. Mansiche - Ca. Los Zafiros (Hospital Regional)	20:54	0.51
376	Av. Nápoles - Ca. Salaverry	08:19	0.89
377	Av. Nápoles - Ca. Salaverry	09:34	0.34
378	Av. Nápoles - Ca. Salaverry	11:36	0.69
379	Av. Nápoles - Ca. Salaverry	15:56	0.77
380	Av. Nápoles - Ca. Salaverry	19:31	0.83
380	Av. Nápoles - Ca. Salaverry	19:31	0.83
381	Av. Nápoles - Ca. Salaverry	13:12	0.83
382	Av. Juan Pablo II (Puerta Principal UNT)	08:26	0.49
383	Av. Juan Pablo II (Puerta Principal UNT)	10:55	0.23
384	Av. Juan Pablo II (Puerta Principal UNT)	12:13	0.34
385	Av. Juan Pablo II (Puerta Principal UNT)	15:51	0.94
386	Av. Juan Pablo II (Puerta Principal UNT)	17:21	0.20
387	Av. Juan Pablo II (Puerta Principal UNT)	20:19	1.03
388	Av. Fatima - Prol. Cesar Vallejo	08:06	0.03
389	Av. Fatima - Prol. Cesar Vallejo	09:25	0.11
390	Av. Fatima - Prol. Cesar Vallejo	12:52	0.77
391	Av. Fatima - Prol. Cesar Vallejo	16:05	0.03
392	Av. Fatima - Prol. Cesar Vallejo	20:44	0.80
393	Av. Fatima - Prol. Cesar Vallejo	10:59	0.20
394	Prol Cesar Vallejo - Paisajistica	07:54	0.89
395	Prol Cesar Vallejo - Paisajistica	10:46	1.00
396	Prol Cesar Vallejo - Paisajistica	12:01	0.49
397	Prol Cesar Vallejo - Paisajistica	14:37	0.03
398	Prol Cesar Vallejo - Paisajistica	20:59	0.97
399	Prol Cesar Vallejo - Paisajistica	20:28	0.94

Figura 35. Data preprocesada de cada nodo estableciendo un posible horario por cada nivel de congestión.

ANEXO C. MATRIZ DE CONSISTENCIA

Titulo	Formulación del Problema	Hipótesis	Variables	Objetivos	Material de Estudio
Aplicación de la minería de datos espaciales basada en técnicas de agrupamiento al congestionamiento del tráfico vehicular en la ciudad de Trujillo, Perú	Determinar la situación actual del tráfico vehicular y determinar mediante técnicas de agrupamiento de minerías de datos espaciales los niveles de congestión vehicular en la ciudad de Trujillo.	La utilización de minería de datos espaciales basada en técnicas de agrupamiento, permite gestionar y controlar el tráfico vehicular en la red vial de la ciudad de Trujillo.	**Variable Independiente**: Minería de Datos Espaciales basada en Técnicas de Agrupamiento **Variable Dependiente** Congestionamiento del Tráfico vehicular	**General:** Determinar la situación actual del tráfico vehicular y determinar mediante técnicas de agrupamiento de usando algoritmos de minerías de datos espaciales los niveles de congestión vehicular en la ciudad de Trujillo. **Específicos:** •Desarrollar un sistema de software simulador para determinar el nivel del congestionamiento, aplicando técnicas de agrupamiento kmeans y dbscan. •Evaluar los algoritmos k-means y dbscan para clustering y clasificación del modelo. •Determinar el grado de satisfacción de los usuarios que influye la implantación del sistema de software simulador para identificar el nivel de congestionamiento.	**Población:** Para este estudio se consideró como población todas las vías de la ciudad de Trujillo. **Muestra:** La muestra está conformada por los 66 Puntos de congestionamiento especificados por la dirección de Transporte Metropolitano de la ciudad de Trujillo.

ANEXO D. INFORMACION DE CONGESTIONAMIENTO (OFICINA DE PROYECTOS DE TRANSPORTES METROPOLITANOS DE TRUJILLO)

La velocidad de recorrido del sistema de transporte público en las vías Av. España-Av. Eguren y Av. César Vallejo es menor a 8 kph, considerándose vías altamente congestionada, asimismo en la Av. República de Panamá-Av. España- Av. Pedro Muñiz y La Av. Roma-Nazaret-España – 28 de Julio se observa que las velocidades están entre 9 kph y 12 kph respectivamente, en tanto en las Av. La Marina y Av. Nicolás de Piérola las velocidades de recorrido están sobre los 22 kph.

Tabla 12. Velocidades de circulación de Transporte Público (KPH)

					TRANSPORTE PÚBLICO				PROMEDIO Velocidad	
					Velocidad Recorrido		Velocidad Marcha		Recorrido	Marcha
RUTAS	VIA METROPOLITANA	TRAMO	IDA	VUELTA	Ida (Km/h)	Vuelta(Km/h)	Ida (Km/h)	Vuelta(Km/h)	Vr (Km/h)	Vm (Km/h)
Tramo 7	Av. Nicolás de Piérola	By Pass Mansiche - Av. Metropolitar	S-N	N-S	25.25	23.49	30.39	28.08	24.37	29.23
Tramo 10	Av. La Marina	Óvalo La Marina - Óvalo Grau	S-N	N-S	21.29	23.76	23.77	26.20	22.53	24.98
Tramo 6	Av. Mansiche	Hosp.Docente-Mall Aventura Plaza	E-O	O-E	17.37	16.45	23.91	22.85	16.91	23.38
Tramo 9	Av. Perú	Av. España - Av. America Norte	E-O	O-E	15.06	16.35	27.40	20.61	15.70	24.01
Tramo 11	Av. Victor Larco	Av. Los Paujiles - Ca. San Vicente	E-O	O-E	13.73	17.33	18.60	23.66	15.53	21.13
Tramo 2	Av. América Norte	AV. Túpac Amaru-Av. Cesar Vallejo	S-N	N-S	13.85	14.10	20.49	20.88	13.98	20.69
Tramo 1	Av. América Sur	Av. César Vallejo -Ovalo Papal	E-O	O-E	14.59	12.79	19.63	15.36	13.69	17.50
Tramo 5B	Av. Roma-Nasaret-España-28Julio	Hosp.Docente-Av.Costa Rica		N-S		12.03		15.28	12.03	15.28
Tramo 5A	Av. Panamá-España-Muñiz	Av. Los Incas-Ovalo By Pass Mansich	S-N		9.19		14.85		9.19	14.85
Tramo 3	Av. Vallejo	Av. América Sur-Ca. Panamá	E-O	O-E	7.60	7.97	14.94	17.00	7.79	15.97
Tramo 4	Av. Vera Enriquez-España-Eguren	By Pass Mansiche - Av. America Sur	S-N	N-S	7.30	8.11	13.56	14.18	7.71	13.87
				Promedio	14.52	15.24	20.75	20.41	14.49	20.08

El promedio de la velocidad de recorrido del transporte público en hora punta, en los tramos de vías estudiados, es 14 KPH; calificándose como vías congestionadas.

Figura 36. Velocidad recorrido transporte público (KPH)

El promedio de la velocidad de recorrido del transporte privado en hora punta en los tramos en estudiados, es 15 kph.

Tabla 13. Velocidad de recorrido promedio (KPH) del Transporte Privado

				TRANSPORTE PRIVADO				PROMEDIO Velocidad	
				Velocidad Recorrido		Velocidad Marcha		Recorrido	Marcha
RUTAS	VIA METROPOLITANA	IDA	VUELTA	Ida (Km/h)	Vuelta(Km/h)	Ida (Km/h)	Vuelta(Km/h)	Vr (Km/h)	Vm (Km/h)
Tramo 10	Av. La Marina	S-N	N-S	17.10	18.34	19.18	20.21	17.72	19.70
Tramo 6	Av. Mansiche	E-O	O-E	22.83	17.28	28.35	23.23	20.05	25.79
Tramo 11	Av. Victor Larco	E-O	O-E	19.43	14.14	24.54	19.86	16.78	22.20
Tramo 1	Av. América Sur	E-O	O-E	21.11	14.68	27.64	25.14	17.90	26.39
Tramo 3	Av. Vallejo	E-O	O-E	10.12	7.45	16.49	13.81	8.78	15.15
Tramo 4	Av. Vera Enriquez-España-Eguren	S-N	N-S	8.58	7.79	16.35	21.42	8.18	18.88
			Promedio	16.53	13.28	22.09	20.61	14.90	21.35

Similar situación al transporte público las vías Av. Eguren y Av. César Vallejo son las más congestionadas, con un promedio de 8 kph.

Figura 37. Velocidad recorrida del Transporte Privado (KPH)

Como resultado de la evaluación de las vías en estudio, se ha determinado que las vías arteriales y colectoras están dentro del rango de los límites de velocidad catalogada como congestionada, los Tramos 4, Tramo 3, Tramo 5A y Tramo 5B se encuentran muy congestionada donde la velocidad de recorrido del transporte público

está por debajo de 12 kph. Se concluye que todas las vías se encuentran congestionadas con velocidades inferiores a 22 kph.

Tabla 14. Resumen de calificación de nivel de congestión

RUTAS	VIA METROPOLITANA	TRAMO	TIPO DE VÍA	LIMITE DE VELOCIDAD (KPH)	VELOCIDAD CATALOGADA COMO CONGESTIONADA	**VELOCIDAD RECORRIDO ACTUAL (KPH)**	**CALIFICACIÓN**
Tramo 10	Av. La Marina	Óvalo La Marina - Óvalo Grau	ARTERIAL	60	<36 KPH	22.53	POCO CONGESTIONADA
Tramo 6	Av. Mansiche	Hosp.Docente-Mall Aventura Plaza	ARTERIAL	60	<36 KPH	16.91	CONGESTIONADA
Tramo 9	Av. Perú	Av. España - Av. America Norte	COLECTORA	60	<36 KPH	15.70	CONGESTIONADA
Tramo 11	Av. Victor Larco	Av. Los Paujiles - Ca. San Vicente	ARTERIAL	60	<36 KPH	15.53	CONGESTIONADA
Tramo 2	Av. América Norte	AV. Túpac Amaru-Av. Cesar Vallejo	ARTERIAL	60	<36 KPH	13.98	CONGESTIONADA
Tramo 1	Av. América Sur	Av. Prol.César Vallejo -Ovalo Papal	ARTERIAL	60	<36 KPH	13.69	CONGESTIONADA
Tramo 5B	Av. Roma-Nasaret-España-28Julio	Hosp.Docente-Av.Costa Rica	ARTERIAL	60	<36 KPH	12.03	MUY CONGESTIONADA
Tramo 5A	Av. Panamá-España-Muñiz	Av. Los Incas-Ovalo By Pass Mansiche	COLECTORA	60	<36 KPH	9.19	MUY CONGESTIONADA
Tramo 3	Av. Vallejo	Av. América Sur-Ca. Panamá	ARTERIAL	60	<36 KPH	7.79	MUY CONGESTIONADA
Tramo 4	Av. Vera Enriquez-España-Eguren	By Pass Mansiche - Av. America Sur	COLECTORA	60	<36 KPH	7.71	MUY CONGESTIONADA
					Promedio	**14.49**	CONGESTIONADA

ANEXO E: INFORMACION DE LA OFICINA DE PROYECTOS DE TRANSPORTES METROPOLITANOS DE TRUJILLO

Si bien la definición planteada tiene un enfoque técnico, este resulta difícil de verificar en la práctica, por lo que estableceremos una definición más taxativa partiendo de la Premisa de reducción de los tiempos de viaje, de la manera siguiente: "un tramo de vía tendrá la condición de congestionada si el tiempo de viaje de sus usuarios se Incrementa en un 50% respecto a su tiempo de circulación ideal (en base a una Velocidad de flujo libre)".

El incremento en los tiempos de viaje establece de manera inversa una reducción en la velocidad media de circulación, siendo que para un incremento en el tiempo de viaje del 50% (T*1.5) implicará una reducción de velocidad de alrededor el 40%, esto de acuerdo a la formulación siguiente.

$$\boldsymbol{E = V * (T * 1.5)}$$

$$\boldsymbol{V = (E / T) * 0.66}$$

Dónde:

E: espacio; V: velocidad y T: tiempo

El parámetro que se establece, desde la perspectiva de velocidad, sería: "un tramo de Vía tendrá la condición de congestionada si su velocidad media espacial es inferior al 60% del valor de su velocidad a flujo libre".

- Para los efectos de la evaluación, tomaremos el valor de la velocidad media espacial que posee la red vial de la ciudad (usada por el servicio de transporte público regular urbano), la misma que es obtenida del Modelo de Transporte antes indicado.
- En ese orden, dado que la normatividad nacional de tránsito ha fijado los límites de velocidad, tomaremos como velocidad a "flujo libre" la velocidad normativa que establece el artículo 162 del Texto Único Ordenado del Reglamento Nacional de Tránsito - Código de Tránsito, aprobado por Decreto Supremo Nº 016-2009-MTC:

• Asimismo, según Ordenanza Municipal N°036-2014-MPT; Articulo N° 7, los límites de velocidad en el Sistema Vial Urbano Metropolitano de Trujillo, son los siguientes:

a. Vías Metropolitanas o Arteriales: 60Km
b. Vías Principales o Colectoras: 50Km
c. Vías locales: 40Km
d. Para Zonas de Hospitales: 30Km
e. Para Zonas escolares: 25 Km

Asimismo, se establece los límites de Máximos Especiales:

f. Para las Intersecciones urbanas ni semaforizadas, la velocidad precautoria, no debe superar los 30 Km.
g. Para la proximidad de establecimientos escolares, deportivos y de gran afluencia de personas, durante el ingreso, su funcionamiento y evacuación, la velocidad precautoria no debe superar los 20 Km/hr.

Que mediante Ordenanza Municipal 038-2013-MPT se aprueba el Plan de Desarrollo Urbano Metropolitano de Trujillo al 2022; en el Item 2.4.3 Sistema Vial Metropolitano, considerándose la clasificación normativa de las vías y la Estructuración del Sistema Vial Metropolitano; considerando que las vías urbanas destinadas a canalizar los flujos de Transporte Urbano se clasifican en: vías Metropolitanas o “Arteriales”; Vías Urbanas Principales o “Colectoras”, Vías Urbanas Secundarias y las vías Locales.

Tabla 15. Clasificación de las Vías Metropolitanas en Trujillo

Tipo Según Codigo de Transito	Tipo de via Ordenanza N°038-MPT
Avenidas	vias Arteriales
	vias Colectoras
calles y Jirones	vias Locales

En consideración a lo anterior las vías locales tendrán una velocidad a flujo libre de 40 km/h, las vías Arteriales y Colectoras una velocidad a flujo libre de 60km/h y para el caso de Vías Expresas una velocidad de 80 km/h.

En ese orden de ideas, si las velocidades de un tramo de vía obtenida en el Modelo de Transporte, resultan ser menores en un 60% a la velocidad a "flujo libre" queda catalogada como un tramo de vía congestionada.

La tabla siguiente resume lo indicado:

Tabla 16. Parámetro de velocidad que califica congestión de la vía

TIPO DE VIA	VELOCIDAD A FLUJO (KPH)	PARAMETROS DE VELOCIDAD EN KPH QUE CATALOGA UANA VIA COMO CONGESTIONADA(LIMITE DE VELOCIDAD)
VIA ARTERIAL	60	<36
VIA COLECTORA	60	<36
VIA LOCAL	40	<24

Sobre la Unidad de Medida y Parámetros de la Variable "Peores Niveles de Servicio"

La capacidad teórica de la vía urbana está definida por el número y ancho de los carriles. Sin embargo, su capacidad real se ve mermada por condiciones físicas y de operación, condiciones ambientales, características del tráfico y si existe las medidas de control de tráfico; estos factores son tales como el estacionamiento, composición del tráfico, giro, poblacional, paraderos, y sincronización semafórica.

Los niveles de servicio se obtienen de los mismos cálculos de capacidades viales.

El nivel de servicio resulta de dividir el volumen vehicular existente respecto de la capacidad real que tiene la vía.

Donde el nivel de servicio está dado por:

$$\text{Nivel de Servicio} = \frac{\text{Volumen demanda de la aproximación en UCP}}{\text{Capacidad de la Aproximación}}$$

Tabla 17. Calificación de los Niveles de Servicio de las Vías Urbanas

Nivel de Servicio	Definición circulación / demora	Índice de Congestión
A	Fluida	<0.60
B	Estable / ligera	0.61-0.70
C	Estable/Aceptable	0.71-0.80
D	Pre-Inestable/Tolerable	0.81-0.90
E	Inestable, Congestionada/Intolerable	0.91-1.00
F	Forzada, congestión total	>1.00

Según De Rus, Campos y Nombela consideran que una vía congestionada vehicularmente es cuando en un determinado momento, el número de vehículos que usa una vía se halla cercano al límite de su capacidad, lo que genera un peor nivel de servicio, que se traduce en velocidades medias más bajas y en tiempos empleados en los trayectos más elevados de lo normal.

ANEXO F: DESARROLLO DE UN SISTEMA DE SOFTWARE PARA SIMULACIÓN DEL TRAFICO VEHICULAR

MODELO DE SOLUCIÓN PROPUESTO

Para el desarrollo del sistema se consideró las fases del ciclo de vida del software:

- Requerimientos
- Análisis
- Diseño
- Implementación
- Pruebas

➢ Requerimientos

En base a la observación del fenómeno de la congestión y la información obtenida por la oficina metropolitana de la municipalidad de Trujillo, se consideraron las siguientes variables a analizar:

- ✓ Exceso de vehículos
- ✓ Mala gestión de uso de las avenidas
- ✓ Imprevistos de obras publicas
- ✓ Mal funcionamiento de señalizaciones
- ✓ Paraderos informales
- ✓ Comercio ambulatorio
- ✓ Caso omiso de la autoridad
- ✓ Falta de policías de transito
- ✓ Falta de toma de conciencias de transeúntes
- ✓ Tránsito de vehículos pesados
- ✓ Accidentes de transito
- ✓ Mala calidad de empresas de transporte público

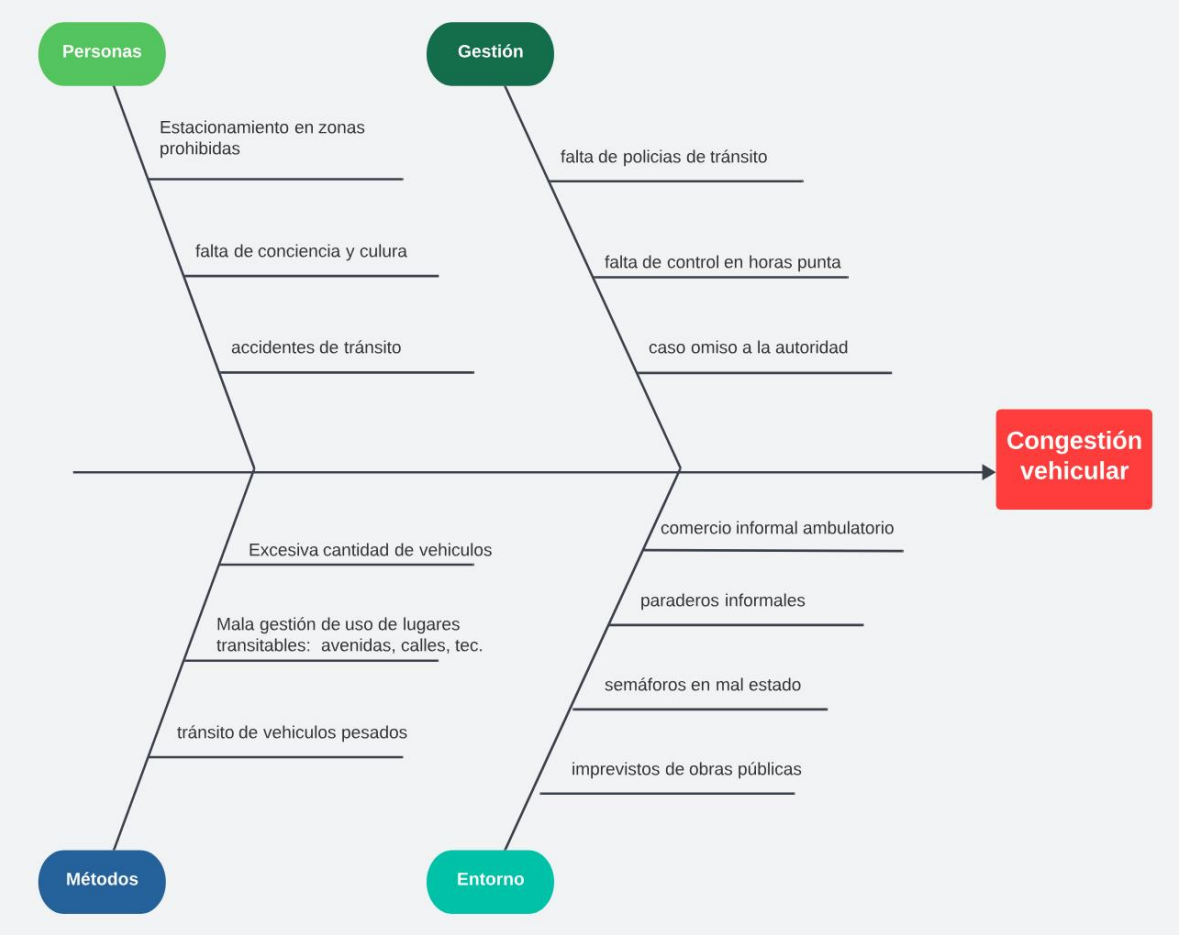

Figura 38. Causas de la congestión del tráfico vehicular.

- **Análisis de los datos**

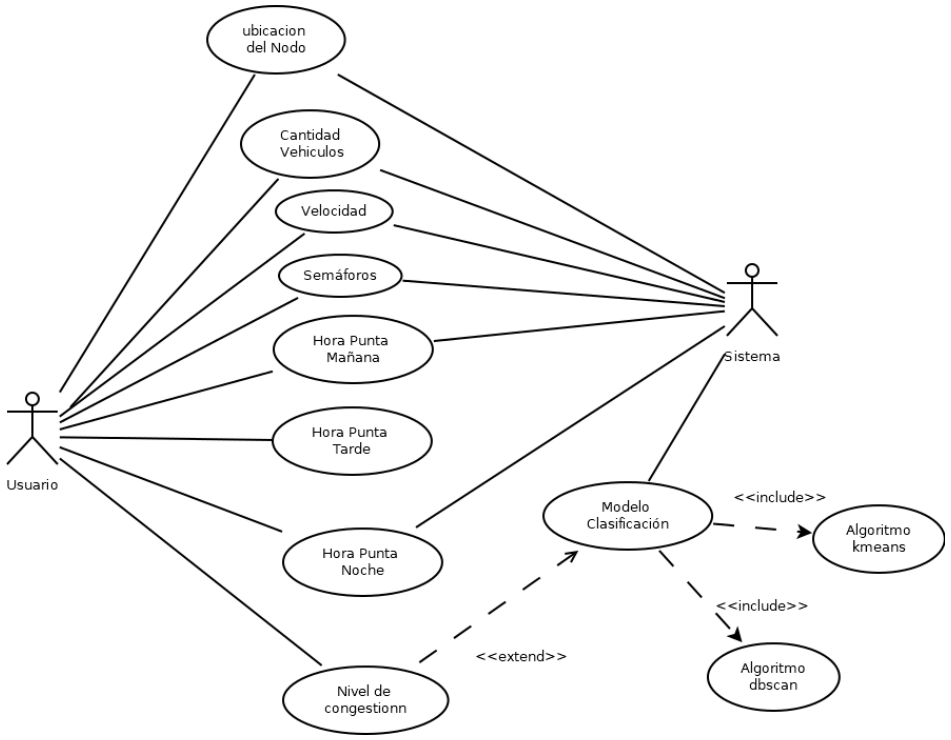

Figura 39. Análisis de los datos del sistema.

- **Diseño**

En esta etapa se procedió a diseñar el modelo de la base de datos, para lo cual se elaboró un diagrama de clases para el diseño de los datos y luego se procedió a diseñar la base de datos con la herramienta MySQL Workbench

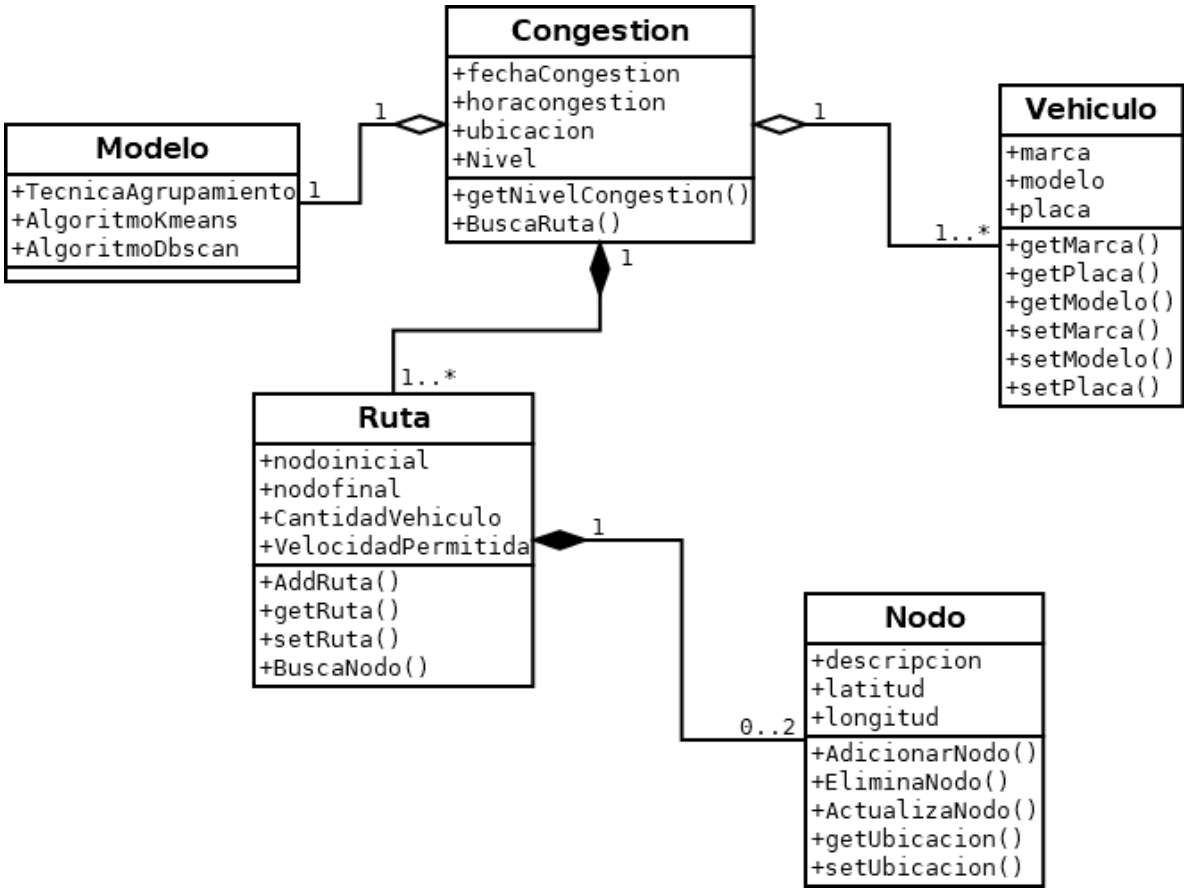

Figura 40. Diagrama de clases del análisis de los datos.

Diseño de la base de datos MySQL

Modelo relacional de la base de datos espacial, para almacenar y determinar la congestión y sus niveles en la ciudad de Trujillo.

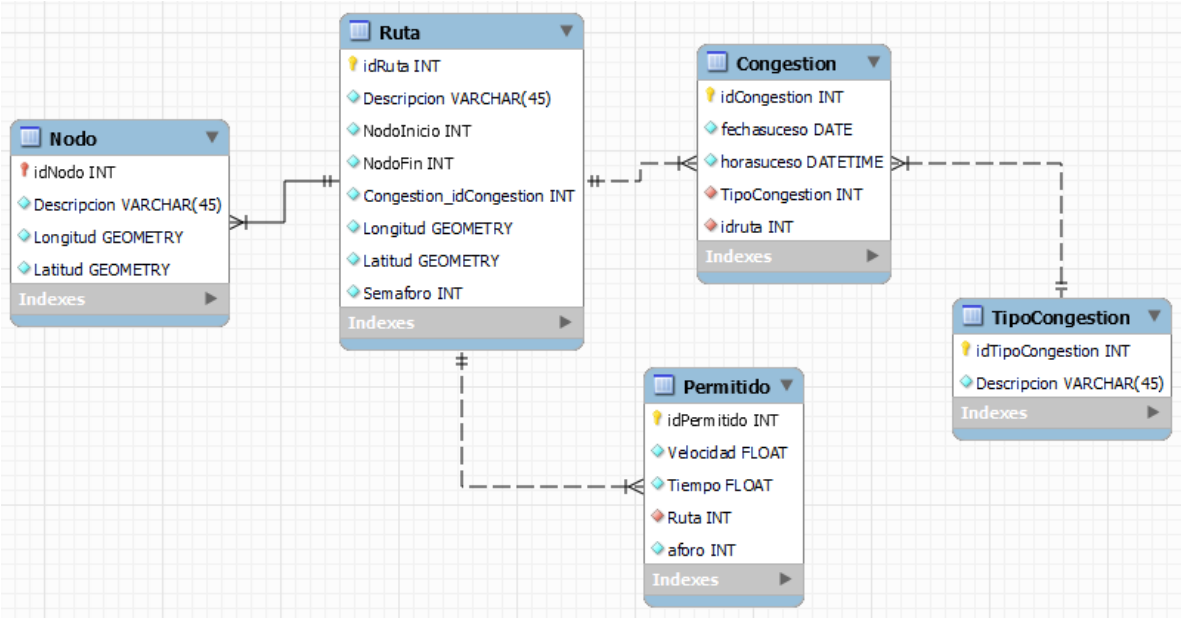

Figura 41. Modelo relacional de la base de datos espacial.

Diseño de interfaces

Figura 42. Interface general del sistema de software para mostrar la clasificación de agrupamientos de nodos, según el nivel de congestión vehicular.

Implementación

A continuación, se muestra el código fuente del sistema de software creado en lenguaje de programación Python.

```
from datetime import datetime, timedelta
import sys # para leer argumentos de la linea de comandos
import io
import random

from PyQt5 import QtCore
from PyQt5 import QtGui
from PyQt5.QtCore import QRegExp, Qt
from PyQt5.QtGui import QBrush, QColor, QRegExpValidator, QStandardItemModel # librería
para leer archivos
import folium
from matplotlib.backends.backend_qt import NavigationToolbar2QT # pip install folium
import numpy as np # pip install numpy
import csv # librería para leer archivos csv
import pyperclip # pip install pyperclip para copiar texto
from PyQt5.QtWidgets import QLineEdit, QMenuBar, QApplication, QComboBox, QHeaderView,
QPushButton, QSpinBox, QStyledItemDelegate, QTabWidget, QTableWidget, QTableWidgetItem,
QWidget, QGridLayout, QHBoxLayout, QVBoxLayout  #pip install PyQt5
import PyQt5.QtWidgets as QtWidgets # para usar QFileDialog
from PyQt5.QtWebEngineWidgets import QWebEngineView # pip install PyQtWebEngine
from folium.plugins import HeatMap # para mostrar mapa de calor
from sklearn.datasets import make_blobs # pip install sklearn
from sklearn.cluster import DBSCAN # para usar DBSCAN
from reloj import DigitalClock

import matplotlib # pip install matplotlib
matplotlib.use('Qt5Agg') # para usar matplotlib en Qt
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg # libreria para mostrar
graficos
from matplotlib.figure import Figure # librería para crear gráficos
import matplotlib.pyplot as plt

import seaborn as sb # pip install seaborn
from sklearn.cluster import KMeans # para usar KMeans
from sklearn.metrics import pairwise_distances_argmin_min # para usar pares de distancias
mínimas

class MplCanvas(FigureCanvasQTAgg): # Clase para graficos MatplotLib en PyQt5

    def __init__(self, parent=None, width=5, height=4, dpi=100): # Constructor de la clase
        fig = Figure(figsize=(width, height), dpi=dpi , ) # Crea el gráfico
        self.axes = fig.add_subplot(111) # Crea el subplot
        super(MplCanvas, self).__init__(fig) # Inicializa la clase padre
```

```
class NumericDelegate(QStyledItemDelegate):
    def createEditor(self, parent, option, index):
        editor = super(NumericDelegate, self).createEditor(parent, option, index)
        if isinstance(editor, QLineEdit):
            reg_ex = QRegExp("[0-9]+.?[0-9]{,3}")
            validator = QRegExpValidator(reg_ex, editor)
            editor.setValidator(validator)
        return editor

class Program(QWidget): # Clase principal del programa
    def __init__(self): # Constructor de la clase
        super().__init__() # Inicializa la clase padre
        self.setWindowTitle('Program') # titulo de la ventana
        self.showMaximized() # maximizar ventana

        # Layout principal
        grid_layout = QGridLayout() # inicializamos el layout principal tipo rejilla
        self.setLayout(grid_layout) # le asignamos el layout principal

        #Boton Seleccionar archivo
        buttonSelect = QPushButton("Seleccionar .csv") # boton para seleccionar archivo csv
        grid_layout.addWidget(buttonSelect, 0, 0, 1, 2) # agregamos el boton al layout
        buttonSelect.clicked.connect(self.selectFile) # conectamos el boton con la funcion
selectFile

        #Etiqueta Label Algoritmo
        labelAlgoritmo = QtWidgets.QLabel() # inicializamos el label "ALGORITMO"
        labelAlgoritmo.setText('Algoritmo') # le asignamos el texto al label
        grid_layout.addWidget(labelAlgoritmo, 0, 3 , 1 , 1) # agregamos el label al layout

        #Selector metodo
        self.cb = QComboBox() # inicializamos el combobox
        self.cb.addItems(["DBSCAN", "K-Means"]) # agregamos los items al combobox
        self.cb.currentIndexChanged.connect(self.changeAlgoritmo) # conectamos el combobox
con la funcion changeAlgoritmo
        grid_layout.addWidget(self.cb, 0, 4, 1, 2) # agregamos el combobox al layout

        #Boton Ejecutar
        buttonRun = QPushButton("Run") # boton para ejecutar el algoritmo
        grid_layout.addWidget(buttonRun, 0, 7, 1, 1) # agregamos el boton al layout
        buttonRun.clicked.connect(self.Ejecutar) # conectamos el boton con la funcion
Ejecutar

        #MAPA DE LA IZQUIERDA

        self.m = folium.Map(  # inicializamos el mapa
            location=[-8.110441, -79.027203], # especificamos la ubicación inicial
            zoom_start=14.45, # especificamos el zoom inicial
```

```
        zoom_control=True, # habilitamos el control de zoom
        scrollWheelZoom=True, # habilitamos el scroll del mouse
        dragging=True # habilitamos el arrastre del mapa
    )

    self.fileName = "" # inicializamos el nombre del archivo
    data = io.BytesIO() # inicializamos el buffer de memoria
    self.m.save(data, close_file=False) # guardamos el mapa en el buffer de memoria
    self.webView = QWebEngineView() # inicializamos el webview
    self.webView.setHtml(data.getvalue().decode()) # cargamos el mapa en el webview
    grid_layout.addWidget(self.webView, 1, 0, 6, 8) # agregamos el webview al layout


    #PANEL DE LA DERECHA
    #TABS
    self.tabs = QTabWidget()
    self.tab1 = QWidget()
    self.tab2 = QWidget()
    self.tabs.resize(270,120)

    # Tabs
    self.tabs.addTab(self.tab1,"Hora")
    self.tabs.addTab(self.tab2,"Dataset")

    # Primer tab
    self.tab1.layout = QGridLayout(self)
    self.clock = DigitalClock()
    self.tab1.layout.addWidget(self.clock,0,1,3,2)

    buttonHUp = QPushButton('+', self)
    buttonHUp.clicked.connect(self.UpHour)
    self.tab1.layout.addWidget(buttonHUp,0,0,2,1)
    buttonHDown = QPushButton('-', self)
    buttonHDown.clicked.connect(self.DownHour)
    self.tab1.layout.addWidget(buttonHDown,1,0,2,1)

    buttonMUp = QPushButton('+', self)
    buttonMUp.clicked.connect(self.UpMin)
    self.tab1.layout.addWidget(buttonMUp,0,4,2,1)
    buttonMDown = QPushButton('-', self)
    buttonMDown.clicked.connect(self.DownMin)
    self.tab1.layout.addWidget(buttonMDown,1,4,2,1)
    self.tab1.setLayout(self.tab1.layout)
    grid_layout.addWidget(self.tabs, 0, 8 , 3 , 1 )
    # Segundo Tab
    self.tab2.layout = QVBoxLayout(self)
    # tabla de datos
    self.tableWidget = QTableWidget() # inicializamos la tabla
    self.tableWidget.setRowCount(0)  # le asignamos el numero de filas
```

```
        self.tableWidget.setColumnCount(15)  # le asignamos el numero de columnas
        self.tableWidget.setHorizontalHeaderLabels(['Etiqueta', 'Latitud',
'Longitud','Semaforo','Hora Punta M', 'Hora Punta T','Hora Punta N', 'Tope', 'Velocidad
Permitida','Tiempo Permitido', 'Espacio Permitido','Cant Vehiculos','Velocidad
Aprox','Tiempo Aprox','Ind. Congestion' ])
        #self.tableWidget.horizontalHeader().setSectionResizeMode(QHeaderView.Stretch) # le
asignamos el tamaño de las columnas
        #self.tableWidget.verticalHeader().setVisible(False) # ocultamos la cabecera
vertical => 1 | 2 | 3 | 4 | 5
        #self.tableWidget.clicked.connect(self.tableClicked) # conectamos la tabla con la
funcion tableClicked
        self.tableWidget.verticalHeader().sectionClicked.connect(self.colocarMarcador)
        delegate = NumericDelegate(self.tableWidget)
        self.tableWidget.setItemDelegate(delegate)
        self.tab2.layout.addWidget(self.tableWidget)
        self.tab2.setLayout(self.tab2.layout)

        # grid_layout.addWidget(self.tableWidget, 2, 8 , 2 , 1 ) # agregamos la tabla al
layout
        # grafico Matplot del DBSCAN
        self.sc = MplCanvas(self, width=5, height=6, dpi=100) # inicializamos el MplCanvas
(declarado arriba línea 18)

        self.sc.axes.xaxis.set_major_locator(plt.NullLocator())
        self.sc.axes.yaxis.set_major_locator(plt.NullLocator())
        grid_layout.addWidget(self.sc, 3, 8 , 4 , 1) # agregamos el MplCanvas al layout

    def colocarMarcador(self): # funcion para cuando se hace click en la tabla
        row = self.tableWidget.currentRow() # obtenemos la fila seleccionada
        name = self.tableWidget.item(row, 0).text() # obtenemos el nombre de la fila
seleccionada
        lon = self.tableWidget.item(row, 1).text() # obtenemos el valor de la celda
seleccionada
        lat = self.tableWidget.item(row, 2).text() # obtenemos el valor de la celda
seleccionada
        data = [[lon,lat]]
        self.m = folium.Map(  # inicializamos el mapa
            location=[-8.110441, -79.027203], # ubicacion del mapa
            zoom_start=14.45, # zoom inicial
            zoom_control=True, # activamos el control de zoom
            scrollWheelZoom=True, # activamos el scroll del mouse
            dragging=True # activamos el arrastre del mouse
        )
        folium.Marker(
            location=[lon,lat],
            popup = name,
        ).add_to(self.m)
        data = io.BytesIO() # inicializamos el buffer
```

```
        self.m.save(data, close_file=False) # guardamos el mapa en el buffer
        self.webView.setHtml(data.getvalue().decode()) # mostramos el mapa en el webview

    def time_in_range(self,start, end, x):
        if start <= end:
            return start <= x <= end
        else:
            return start <= x or x <= end

    def Ejecutar(self): # funcion para ejecutar el algoritmo
        if self.fileName == "": # si no se ha seleccionado un archivo
            print("No hay archivo cargado") # imprimimos que no hay archivo cargado
            return # salimos de la función
        lat = [] # inicializamos la lista de latitudes
        lon = [] # inicializamos la lista de longitudes
        semaforo = []
        etiquetas = []
        horapuntaM = [] # inicializamos la lista de horas punta M
        horapuntaT = [] # inicializamos la lista de horas punta T
        horapuntaN = [] # inicializamos la lista de horas punta N
        cantidad = [] # inicializamos la lista de cantidades minima
        velLimite = [] # inicializamos la lista de velocidades limite
        velocidades = []
        tiempLimite = [] # inicializamos la lista de tiempos limite
        tiempos = []
        cantMaxima = [] # inicializamos la lista de cantidades maxima
        valorMaximo = 0 # inicializamos el valor maximo

        for i in range(0,self.tableWidget.rowCount()): # recorremos todas las filas
            if int(self.tableWidget.item(i,8).text()) > valorMaximo:
                valorMaximo = int(self.tableWidget.item(i,8).text()) # obtenemos el valor
maximo
        # Obtenemos los datos de la tabla
        for i in range(0,self.tableWidget.rowCount()):
            for k in range(0,self.tableWidget.columnCount()):
                if k == 0: # almacenamos las latitudes
                    etiquetas.append(str(i+1))
                if k == 1: # almacenamos las latitudes
                    lat.append(float(self.tableWidget.item(i,k).text()))
                elif k==2: # almacenamos las longitudes
                    lon.append(float(self.tableWidget.item(i,k).text()))
                elif k==3:
                    semaforo.append(str(self.tableWidget.item(i,k).text()))
                elif k==4: # almacenamos las hora punta m
                    horapuntaM.append(str(self.tableWidget.item(i,k).text()))
                elif k==5: # almacenamos las hora punta t
                    horapuntaT.append(str(self.tableWidget.item(i,k).text()))
                elif k==6: # almacenamos las hora punta n
```

```
                    horapuntaN.append(str(self.tableWidget.item(i,k).text()))
                elif k==7: # almacenamos topes
                    if int(self.tableWidget.item(i,k).text()) == 0: # validacion division
por 0
                        cantMaxima.append(0.001) #asignamos un valor minimo
                    else:
                        cantMaxima.append(int(self.tableWidget.item(i,k).text())) #
almacenamos las cantidades minima
                elif k==8: # almacenamos velocidad permitido
                    velLimite.append(int(self.tableWidget.item(i,k).text()))
                elif k==9: # almacenamos tiempo permitido
                    tiempLimite.append(int(self.tableWidget.item(i,k).text()))

        # determinamos cantidad
        for i in range(0,self.tableWidget.rowCount()):
            qty = 0
            encontrado = False
            if str(self.tableWidget.item(i,4).text()) !="" and not encontrado:
                intervalo = str(self.tableWidget.item(i,4).text()).split("-")
                if self.time_in_range(datetime.strptime(intervalo[0], '%H:%M'),
datetime.strptime(intervalo[1], '%H:%M'), self.clock.time):
                    if str(self.tableWidget.item(i,3).text()) == "Si":
                        qty = random.randint(int(cantMaxima[i-1]/5),int(cantMaxima[i-
1]*6/5))
                    else:
                        qty = random.randint(int(cantMaxima[i-1]/5),int(cantMaxima[i-1]))
                    encontrado = True
            if str(self.tableWidget.item(i,5).text()) !="" and not encontrado:
                intervalo = str(self.tableWidget.item(i,5).text()).split("-")
                if self.time_in_range(datetime.strptime(intervalo[0], '%H:%M'),
datetime.strptime(intervalo[1], '%H:%M'), self.clock.time):
                    if str(self.tableWidget.item(i,3).text()) == "Si":
                        qty = random.randint(int(cantMaxima[i-1]/5),int(cantMaxima[i-
1]*6/5))
                    else:
                        qty = random.randint(int(cantMaxima[i-1]/2),int(cantMaxima[i-1]))
                    encontrado = True
            if str(self.tableWidget.item(i,6).text()) !="" and not encontrado:
                intervalo = str(self.tableWidget.item(i,6).text()).split("-")
                if self.time_in_range(datetime.strptime(intervalo[0], '%H:%M'),
datetime.strptime(intervalo[1], '%H:%M'), self.clock.time):
                    if str(self.tableWidget.item(i,3).text()) == "Si":
                        qty = random.randint(int(cantMaxima[i-1]/5),int(cantMaxima[i-
1]*6/5))
                    else:
                        qty = random.randint(int(cantMaxima[i-1]/5),int(cantMaxima[i-1]))
                    encontrado = True
            if not encontrado:
```

```
                if self.time_in_range(datetime.strptime('02:00', '%H:%M'),
datetime.strptime('04:00', '%H:%M'), self.clock.time):
                    qty= qty = random.randint(0,int(cantMaxima[i-1]/3))
                else:
                    qty = random.randint(0,int(cantMaxima[i-1]/2))
            item = QTableWidgetItem(str(qty),QtCore.Qt.ItemIsEnabled)
            item.setFlags(QtCore.Qt.ItemIsEnabled)
            self.tableWidget.setItem(i,11, QTableWidgetItem(item))
            cantidad.append(qty)

        # determinamos tiempo aproximado
        for i in range(0,self.tableWidget.rowCount()):
            valor = random.randint(10,20)
            item = QTableWidgetItem(str(valor),QtCore.Qt.ItemIsEnabled)
            item.setFlags(QtCore.Qt.ItemIsEnabled)
            self.tableWidget.setItem(i,13, QTableWidgetItem(item))
            tiempos.append(valor)

        # determinamos velocidades
        for i in range(0,self.tableWidget.rowCount()):
            item =
QTableWidgetItem(str(round(velLimite[i]*tiempLimite[i]/tiempos[i],2)),QtCore.Qt.ItemIsEnabl
ed)
            item.setFlags(QtCore.Qt.ItemIsEnabled)
            self.tableWidget.setItem(i,12, QTableWidgetItem(item))
            velocidades.append(velLimite[i]*tiempLimite[i]/tiempos[i])

        # Establecemos espacios
        for i in range(0,self.tableWidget.rowCount()):
            item =
QTableWidgetItem(str(velLimite[i]*tiempLimite[i]),QtCore.Qt.ItemIsEnabled)
            item.setFlags(QtCore.Qt.ItemIsEnabled)
            self.tableWidget.setItem(i,10, QTableWidgetItem(item))

        # Establecemos los indices
        for i in range(0,self.tableWidget.rowCount()):
            valor = round(cantidad[i]/cantMaxima[i],2)
            indice = ""
            if valor <= 0.6:
                indice = "Fluida"
            elif valor <= 0.7:
                indice = "Estable/Ligera"
            elif valor <= 0.8:
                indice = "Estable/Aceptable"
            elif valor <= 0.9:
                indice = "Pre-inestable/Tolerable"
            elif valor <= 1:
                indice = "Inestable,Congestionada/Intolerable"
```

```
            elif valor >1:
                indice = "Forzada/Congestión Total"
            item = QTableWidgetItem(str(indice), QtCore.Qt.ItemIsEnabled)
            item.setFlags(QtCore.Qt.ItemIsEnabled)
            item.setForeground(QBrush(QColor(280, 0,0)))
            self.tableWidget.setItem(i,14, QTableWidgetItem(item))

        x1 = []
        x2 = []
        for i in range(self.tableWidget.rowCount()):
            x1.append(self.clock.time)
            x2.append(cantidad[i]/cantMaxima[i] if cantMaxima[i]>=cantidad[i] else 1)
        coordenadas = [ [lat[int(i/3)],lon[int(i/3)],x2[i]] for i in range(len(x2))] #
lista de coordenadas para el mapa de calor
        self.m = folium.Map(  # inicializamos el mapa
            location=[-8.110441, -79.027203], # ubicacion del mapa
            zoom_start=14.45, # zoom inicial
            zoom_control=True, # activamos el control de zoom
            scrollWheelZoom=True, # activamos el scroll del mouse
            dragging=True # activamos el arrastre del mouse
        )
        HeatMap(coordenadas).add_to(folium.FeatureGroup(name='Heat Map').add_to(self.m)) #
agregamos el mapa de calor al mapa
        data = io.BytesIO() # inicializamos el buffer
        self.m.save(data, close_file=False) # guardamos el mapa en el buffer
        self.webView.setHtml(data.getvalue().decode()) # mostramos el mapa en el
webview
        #DATA
        #horas =  [x for x in range(24)]
        X = np.array([ [x1[i].hour*60+x1[i].second/60,x2[i]] for i in
range(self.tableWidget.rowCount())]) # inicializamos la matriz de datos
        self.ejecutarAlgoritmo(X,etiquetas)

    def ejecutarAlgoritmo(self,X,etiquetas):
        if self.fileName == "": # si no se ha seleccionado un archivo
            print("No hay archivo cargado") # imprimimos que no hay archivo cargado
            return # salimos de la función
        if self.cb.currentText() == "DBSCAN":
            #DBSCAN
            epsilon = 0.04 # le asignamos el valor de epsilon
            min_samples = 10 # le asignamos el valor de min_samples
            db = DBSCAN(eps=epsilon, min_samples=min_samples).fit(X) # inicializamos el
algoritmo DBSCAN
            labels = db.labels_
            aux = list(dict.fromkeys(labels))
            col = [plt.cm.Spectral(each) for each in np.linspace(0, 1,
len(aux))]
            colors = []
```

```
            for i in range(len(labels)):
                for k in range(len(aux)):
                    if labels[i] == aux[k]:
                        colors.append(col[k])
            #colors = list(map(lambda x: '#3b4cc0' if x == 1 else '#b40426', labels)) #
asignamos los colores a las etiquetas
            #print(aux) # imprimimos los colores
            self.sc.axes.clear() # limpiamos el grafico
            valores = {}
            horas = {}
            for i in range(len(labels)):
                if "'"+str(labels[i])+"'" in valores:
                    valores["'"+str(labels[i])+"'"].append(X[i,1])
                else:
                    valores["'"+str(labels[i])+"'"] = [X[i,1]]
                if "'"+str(labels[i])+"'" in horas:
                    val = datetime.strptime(str(int(X[i,0]/60))+":"+str(int(X[i,0]%60)),
'%H:%M').strftime("%H:%M")
                    horas["'"+str(labels[i])+"'"].append(val)
                else:
                    val = datetime.strptime(str(int(X[i,0]/60))+":"+str(int(X[i,0]%60)),
'%H:%M').strftime("%H:%M")
                    horas["'"+str(labels[i])+"'"]= [val]
            for i in range(len(aux)):
                label = str(aux[i])
                if label == '-1':
                    label = "RUIDO"
                else:
                    label = "CLUSTER "+str(aux[i]+1)
                self.sc.axes.scatter(horas["'"+str(aux[i])+"'"],valores["'"+str(aux[i])+"'"
],label=label)
            #self.sc.axes.scatter( x3[:] , X[:,1], c=colors, marker="o", picker=True) #
dibujamos los puntos en el mapa
            valx=[]
            valy=[]
            etiq=[]
            texts = []
            for i,txt in enumerate(etiquetas):
                if str(labels[i]) != "-1":
                    valx.append(float(X[i,0]))
                    valy.append(float(X[i,1]))
                    etiq.append(str(labels[i]))
                    texts.append(txt)
            arrx = []
            arry = []
            arre = {}
            for i in range(len(etiq)):
                if not valy[i] in arry:
```

```
                    arrx.append(valx[i])
                    arry.append(valy[i])
                    arre[str(valy[i])] = str(texts[i])
                else:
                    if str(texts[i]) not in arre[str(valy[i])]:
                        arre[str(valy[i])] += ", "+ str(texts[i])
            #for i,txt in enumerate(arry):
            #    self.sc.axes.annotate(" "*random.randint(1,3)+str(arre[str(txt)]),
(datetime.strptime(str(int(int(arrx[i])/60))+":"+str(int(int(arrx[i])%60)),
'%H:%M').strftime("%H:%M"),arry[i]),size = 7)
            anadidos = []
            for i,txt in enumerate(arry):
                cercano=1
                val =
datetime.strptime(str(int(int(arrx[i])/60))+":"+str(int(int(arrx[i])%60)),
'%H:%M').strftime("%H:%M")
                for valor in anadidos:
                    if round(valor,1) == round(arry[i],1):
                        cercano+=1
                if i%2 == 0:
                    self.sc.axes.annotate(str(arre[str(txt)])+"-"*(cercano*5), xy = (val
,arry[i]),size = 6, ha='right')
                else:
                    self.sc.axes.annotate("-"*(cercano*5)+str(arre[str(txt)]), xy = (val
,arry[i]),size = 6, ha='left')
                anadidos.append(arry[i])
            '''
            for i, txt in enumerate(etiquetas):
                if "-1" not in str(labels[i]):
                    self.sc.axes.annotate(txt, (aux[i] , X[i,1]))            '''
            self.sc.axes.legend(loc='upper left', frameon=False, fancybox=True,
shadow=True)
        elif self.cb.currentText() == "K-Means":
            #K-Means
            k = 6 # le asignamos el valor de k
            kmeans = KMeans(
                init="random",
                n_clusters=6,
                n_init=10,
                max_iter=300,
                random_state=42
            )
            label = kmeans.fit_predict(X)
            #Getting unique labels
            u_labels = np.unique(label)
            #plotting the results:
            self.sc.axes.clear() # limpiamos el grafico
            x3 = []
```

```
            for i in u_labels:
                temp=[]
                for j in range(len(X[label == i , 0])):
                    val = float(X[label == i , 0][j])
                    temp = datetime.strptime(str(int(val/60))+":"+str(int(val%60)),
'%H:%M').strftime("%H:%M")
                    temp = np.array(temp)
                x3.append(temp)
            x3 = np.array(x3)


            for i in u_labels:
                temp = []
                for j in range(len(X[label == i , 0])):
                    val = float(X[label == i , 0][j])
                    temp.append(datetime.strptime(str(int(val/60))+":"+str(int(val%60)),
'%H:%M').strftime("%H:%M"))
                repl = np.array(temp)
                self.sc.axes.scatter(repl, X[label == i , 1] , label = "CLUSTER "+str(i+1))
            valx=[]
            valy=[]
            etiq=[]
            texts = []
            for i, txt in enumerate(etiquetas):
                valx.append(float(X[i,0]))
                valy.append(float(X[i,1]))
                texts.append(txt)
            arrx = []
            arry = []
            arre = {}
            for i in range(len(texts)):
                if not valy[i] in arry:
                    arrx.append(valx[i])
                    arry.append(valy[i])
                    arre[str(valy[i])] = str(texts[i])
                else:
                    if str(texts[i]) not in arre[str(valy[i])]:
                        arre[str(valy[i])] += ", "+ str(texts[i])
            #for i,txt in enumerate(arry):
            #    val = float(arrx[i])
            #    val = datetime.strptime(str(int(val/60))+":"+str(int(val%60)),
'%H:%M').strftime("%H:%M")
            #    self.sc.axes.annotate(" "*random.randint(1,3)+str(arre[str(arry[i])]),
(val, arry[i]))
            anadidos=[]
            for i,txt in enumerate(arry):
                val = float(arrx[i])
```

```
                val = datetime.strptime(str(int(val/60))+":"+str(int(val%60)),
'%H:%M').strftime("%H:%M")
                cercano=1
                for valor in anadidos:
                    if round(valor,1) == round(arry[i],1):
                        cercano+=1
                if i%2 ==0:
                    self.sc.axes.annotate("-"*(cercano*5)+str(arre[str(arry[i])]), (val,
arry[i]) ,size = 6, ha='left')
                else:
                    self.sc.axes.annotate(str(arre[str(arry[i])])+"-"*(cercano*5), (val,
arry[i]) ,size = 6, ha='right')
                anadidos.append(arry[i])

            #self.sc2.axes.legend(loc='upper center', bbox_to_anchor=(0.5, -
0.05),prop={'size': 6}, fancybox=True, shadow=True, ncol=5)
            self.sc.axes.legend(loc='upper left',fancybox=True, shadow=True, frameon=False)

        self.sc.draw() # dibujamos el mapa
        self.sc.flush_events() # actualizamos el mapa

    def changeAlgoritmo(self):
        if self.fileName == "": # si no se ha seleccionado un archivo
            print("No hay archivo cargado") # imprimimos que no hay archivo cargado
            return # salimos de la función
        lat = [] # inicializamos la lista de latitudes
        lon = [] # inicializamos la lista de longitudes
        semaforo = []
        etiquetas = []
        horapuntaM = [] # inicializamos la lista de horas punta M
        horapuntaT = [] # inicializamos la lista de horas punta T
        horapuntaN = [] # inicializamos la lista de horas punta N
        cantidad = [] # inicializamos la lista de cantidades minima
        cantMaxima = [] # inicializamos la lista de cantidades maxima
        valorMaximo = 0 # inicializamos el valor maximo

        for i in range(0,self.tableWidget.rowCount()): # recorremos todas las filas
            if int(self.tableWidget.item(i,9).text()) > valorMaximo:
                valorMaximo = int(self.tableWidget.item(i,9).text()) # obtenemos el valor
maximo
        # Obtenemos los datos de la tabla
        for i in range(0,self.tableWidget.rowCount()):
            for k in range(0,self.tableWidget.columnCount()):
                if k == 0: # almacenamos las latitudes
                    etiquetas.append(str(i+1))
                if k == 1: # almacenamos las latitudes
                    lat.append(float(self.tableWidget.item(i,k).text()))
                elif k==2: # almacenamos las longitudes
```

```
                    lon.append(float(self.tableWidget.item(i,k).text()))
                elif k==3:
                    semaforo.append(str(self.tableWidget.item(i,k).text()))
                elif k==4: # almacenamos las hora punta m
                    horapuntaM.append(str(self.tableWidget.item(i,k).text()))
                elif k==5: # almacenamos las hora punta t
                    horapuntaT.append(str(self.tableWidget.item(i,k).text()))
                elif k==6: # almacenamos las hora punta n
                    horapuntaN.append(str(self.tableWidget.item(i,k).text()))
                elif k==7:
                    if int(self.tableWidget.item(i,k).text()) == 0: # validacion division
por 0
                        cantMaxima.append(0.001) #asignamos un valor minimo
                    else:
                        cantMaxima.append(int(self.tableWidget.item(i,k).text())) #
almacenamos las cantidades minima
        # determinamos cantidad
        if self.tableWidget.rowCount()>0:
            for i in range(0,self.tableWidget.rowCount()):
                cantidad.append(int(self.tableWidget.item(i,11).text()))

        if 0 in cantidad:
            return # salimos de la función
        x1 = []
        x2 = []
        for i in range(self.tableWidget.rowCount()):
            x1.append(self.clock.time)
            x2.append(cantidad[i]/cantMaxima[i] if cantMaxima[i]>=cantidad[i] else 1)
        #DATA
        X = np.array([ [x1[i].hour*60+x1[i].second/60,x2[i]] for i in
range(self.tableWidget.rowCount())]) # inicializamos la matriz de datos
        self.ejecutarAlgoritmo(X,etiquetas)

    def selectFile(self):
        self.fileName, _ = QtWidgets.QFileDialog.getOpenFileName(self, "Abrir archivo", "",
"CSV (*.csv)")
        if self.fileName == "":
            return
        self.loadCsv(self.fileName)

    def keyPressEvent(self, event):
        super().keyPressEvent(event)
        '''
        if event.key() == Qt.Key_V and (event.modifiers() & Qt.ControlModifier):
            if self.tableWidget.currentRow() == -1:
                return
            row = self.tableWidget.currentRow()
            filas = pyperclip.paste().split("\n")
```

```
            for fila in filas:
                if fila == "":
                    continue
                cols = fila.split("\t")
                val = 9
                for col in cols:
                    if row>=self.tableWidget.rowCount():
                        return
                    self.tableWidget.setItem(row, val, QTableWidgetItem(col))
                    val+=1
                row+=1
        '''
    def UpHour(self):
        hora = self.clock.time
        hours_added = timedelta(hours = 1)
        future_date_and_time = hora + hours_added
        self.clock.setTime(future_date_and_time)
        #self.clock.setTime(self.clock.time().addSecs(3600))

    def DownHour(self):
        hora = self.clock.time
        hours_added = timedelta(hours = -1)
        future_date_and_time = hora + hours_added
        self.clock.setTime(future_date_and_time)

    def UpMin(self):
        hora = self.clock.time
        hours_added = timedelta(minutes= 1)
        future_date_and_time = hora + hours_added
        self.clock.setTime(future_date_and_time)

    def DownMin(self):
        hora = self.clock.time
        hours_added = timedelta(minutes= -1)
        future_date_and_time = hora + hours_added
        self.clock.setTime(future_date_and_time)

    def loadCsv(self, fileName):
        fileX = open(fileName)
        reader = csv.reader(fileX)
        lines= len(list(reader))
        self.tableWidget.setRowCount(lines-1)
        k=0
        with open(fileName,encoding='utf-8',errors='ignore') as csv_file:
            csv_reader = csv.reader(csv_file, delimiter=',')
            headings = next(csv_reader)
            for row in csv_reader:
                h=0
```

```
                for field in row:
                    if h>0:
                        if h==4:
                            if int(field)==1:
                                item = QTableWidgetItem("Si",QtCore.Qt.ItemIsEnabled)
                                item.setFlags(QtCore.Qt.ItemIsEnabled)
                                self.tableWidget.setItem(k,h-1, item )
                            else:
                                item = QTableWidgetItem("No",QtCore.Qt.ItemIsEnabled)
                                item.setFlags(QtCore.Qt.ItemIsEnabled)
                                self.tableWidget.setItem(k,h-1, item )
                        else:
                            item = QTableWidgetItem(field,QtCore.Qt.ItemIsEnabled)
                            item.setFlags(QtCore.Qt.ItemIsEnabled)
                            self.tableWidget.setItem(k,h-1, item )
                    h+=1
                item = QTableWidgetItem("",QtCore.Qt.ItemIsEnabled)
                item.setFlags(QtCore.Qt.ItemIsEnabled)
                self.tableWidget.setItem(k,h-1, QTableWidgetItem(item))
                h+=1
                self.tableWidget.setItem(k,h-1, QTableWidgetItem(item))
                h+=1
                self.tableWidget.setItem(k,h-1, QTableWidgetItem(item))
                k+=1

if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = Program()
    ex.show()
    try:
        sys.exit(app.exec ())
    except SystemExit:
        print("Closing program")
```

Pruebas

A continuación, se prueban las diferentes interfaces desarrolladas

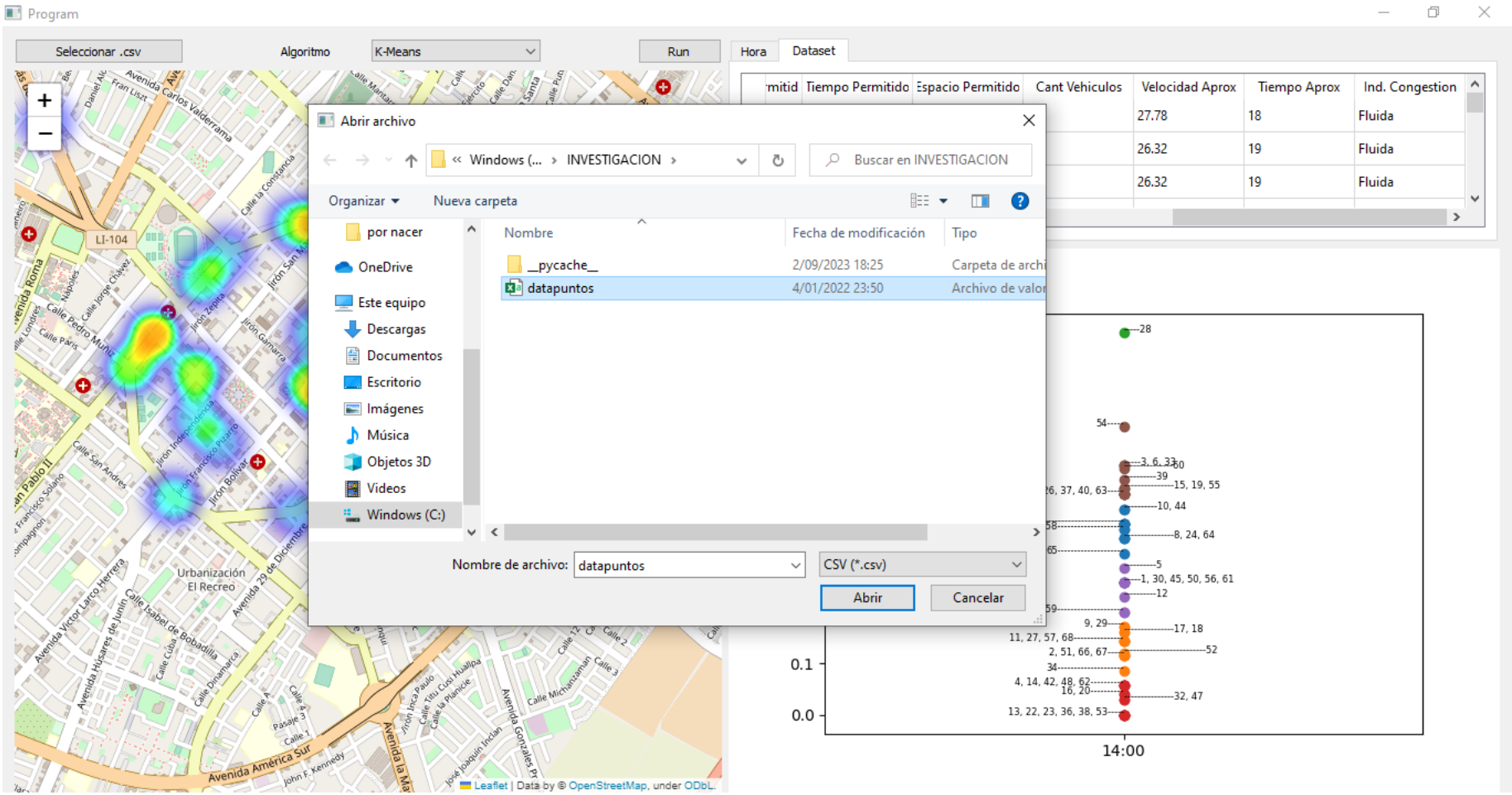

Figura 43. Cargado de datos al sistema de software.

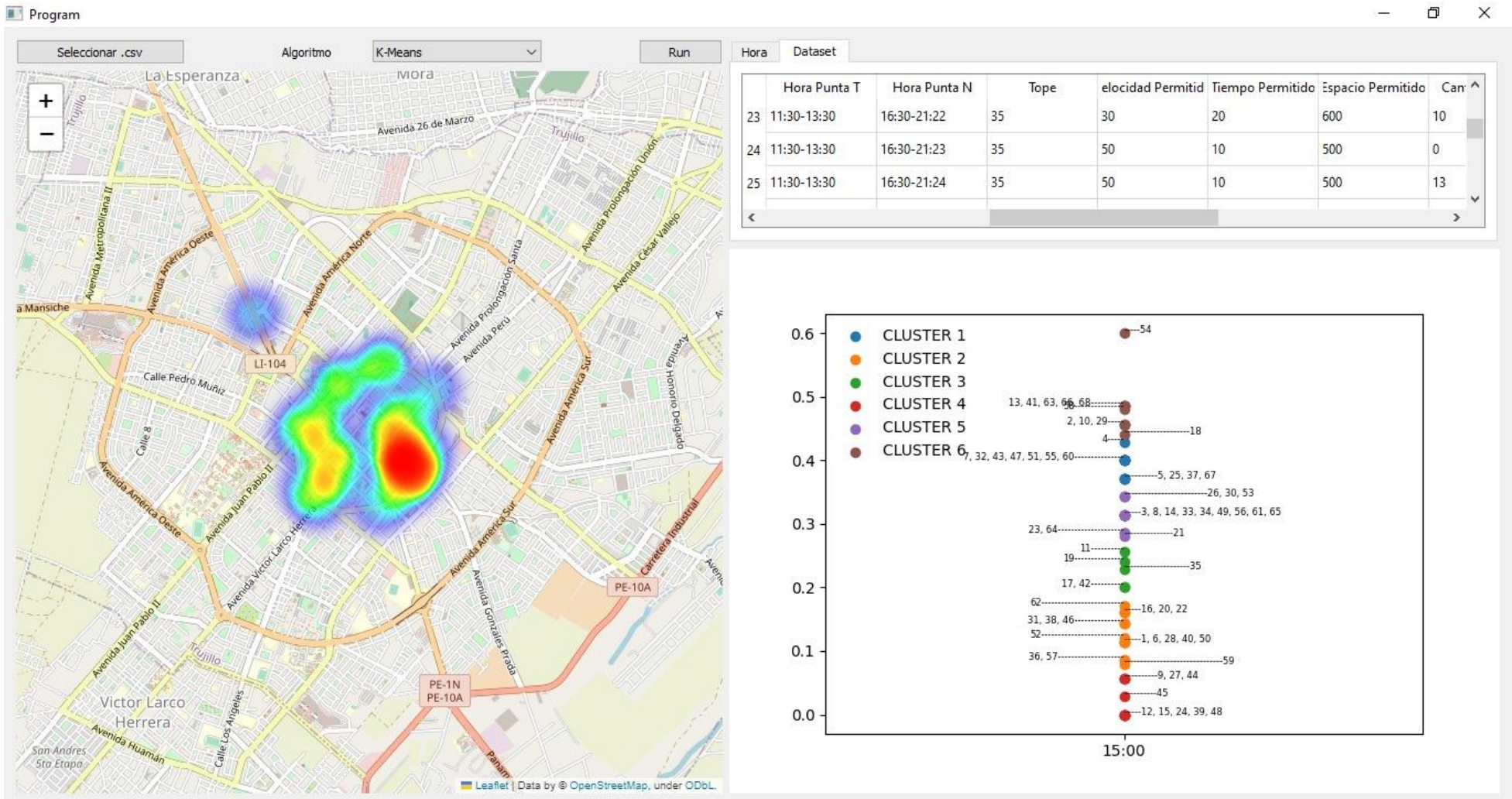

Figura 44. Clasificacion de agrupamientos mediante el algoritmo k-means.

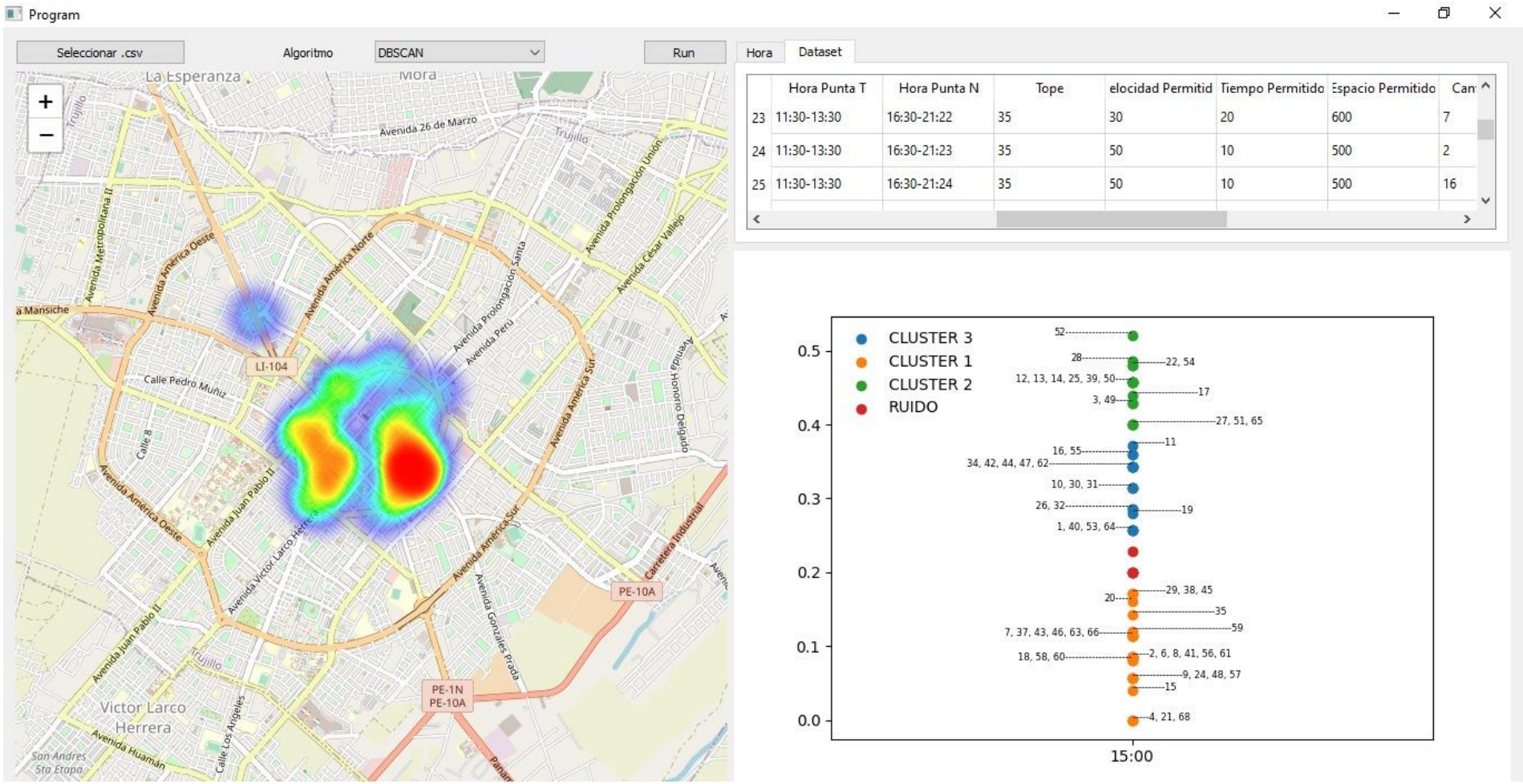

Figura 45. Clasificación de agrupamientos mediante el algoritmo dbscan.

Revision Tesis Doctoral

por Jose Arturo Diaz Pulido

Fecha de entrega: 25-oct-2023 04:37a.m. (UTC-0500)
Identificador de la entrega: 2206737510
Nombre del archivo: Tesis_UNS_jadp.docx (28.41M)
Total de palabras: 25562
Total de caracteres: 146395

10	ciudadmas.com Fuente de Internet	1 %
11	www.bibliotecadigitaldebogota.gov.co Fuente de Internet	1 %
12	acimed.sld.cu Fuente de Internet	1 %
13	repository.uaeh.edu.mx Fuente de Internet	<1 %
14	fdocuments.mx Fuente de Internet	<1 %
15	repositorio.ug.edu.ec Fuente de Internet	<1 %
16	vdocuments.site Fuente de Internet	<1 %
17	www0.unsl.edu.ar Fuente de Internet	<1 %
18	fdocuments.es Fuente de Internet	<1 %
19	wwwae.ciemat.es Fuente de Internet	<1 %
20	es.stackoverflow.com Fuente de Internet	<1 %
21	inteligenciaartificial.science Fuente de Internet	<1 %

22	www.tmt.gob.pe Fuente de Internet	<1 %
23	rcics.sld.cu Fuente de Internet	<1 %
24	eprints.uanl.mx Fuente de Internet	<1 %
25	elvex.ugr.es Fuente de Internet	<1 %
26	eprints.ucm.es Fuente de Internet	<1 %
27	lookformedical.com Fuente de Internet	<1 %
28	siar.minam.gob.pe Fuente de Internet	<1 %
29	repositorio.espe.edu.ec Fuente de Internet	<1 %
30	revistas.unab.edu.co Fuente de Internet	<1 %
31	idoc.pub Fuente de Internet	<1 %
32	repositorio.uach.mx Fuente de Internet	<1 %
33	cdn.www.gob.pe Fuente de Internet	<1 %

34	mafiadoc.com Fuente de Internet	<1 %
35	repositorio.ucv.edu.pe Fuente de Internet	<1 %
36	www.prism.uvsq.fr Fuente de Internet	<1 %
37	repositorio.upla.edu.pe Fuente de Internet	<1 %
38	revvialibre.com.mx Fuente de Internet	<1 %
39	www.redalyc.org Fuente de Internet	<1 %
40	rua.ua.es Fuente de Internet	<1 %
41	repositoriobibliotecas.uv.cl Fuente de Internet	<1 %
42	repositorio.uss.edu.pe Fuente de Internet	<1 %
43	pure.roehampton.ac.uk Fuente de Internet	<1 %
44	www.lsi.us.es Fuente de Internet	<1 %
45	www.ve-mas.com Fuente de Internet	<1 %

46	repositorio.puce.edu.ec Fuente de Internet	<1 %
47	repositorio.uta.edu.ec Fuente de Internet	<1 %
48	sistemagestorbasededatos19.blogspot.com Fuente de Internet	<1 %
49	www.dsic.upv.es Fuente de Internet	<1 %
50	mariaconcepciongomezlopez.blogspot.com Fuente de Internet	<1 %
51	aprendeia.com Fuente de Internet	<1 %
52	forum.qt.io Fuente de Internet	<1 %
53	issuu.com Fuente de Internet	<1 %
54	question-it.com Fuente de Internet	<1 %
55	repositorio.usanpedro.edu.pe Fuente de Internet	<1 %
56	funcionpublica.gov.co Fuente de Internet	<1 %
57	revistas.unilibre.edu.co Fuente de Internet	<1 %

58	www.knowstack.com Fuente de Internet	<1 %
59	pensamiento.unal.edu.co Fuente de Internet	<1 %
60	repository.unab.edu.co Fuente de Internet	<1 %
61	rstudio-pubs-static.s3.amazonaws.com Fuente de Internet	<1 %
62	designscad.com Fuente de Internet	<1 %
63	"Tesis Exploración de la relación entre rendimiento académico de alumnos de pregrado, consultas de los servicios de biblioteca y multidisciplinariedad, aplicando técnicas de minería de datos", Pontificia Universidad Catolica de Chile, 2016 Publicación	<1 %
64	guatemalatraditionaltravel.com Fuente de Internet	<1 %
65	dialnet.unirioja.es Fuente de Internet	<1 %
66	dominiodelasciencias.com Fuente de Internet	<1 %
67	up-rid.up.ac.pa Fuente de Internet	<1 %

68	Amable Tinizaray Olmedo, José Raúl Castro, Ruth Reátegui, Tuesman Castillo. "Aplicación de algoritmos de Machine Learning para la segmentación del consumo de agua potable. Caso de estudio en Catamayo, Ecuador", 2023 18th Iberian Conference on Information Systems and Technologies (CISTI), 2023 Publicación	<1 %
69	openaccess.uoc.edu Fuente de Internet	<1 %
70	www.cenidet.edu.mx Fuente de Internet	<1 %
71	github.com Fuente de Internet	<1 %
72	dspace.ups.edu.ec Fuente de Internet	<1 %
73	tanriverdiirem.medium.com Fuente de Internet	<1 %
74	tel.archives-ouvertes.fr Fuente de Internet	<1 %
75	www.cacic2016.unsl.edu.ar Fuente de Internet	<1 %
76	latorredehercules.blogia.com Fuente de Internet	<1 %
77	pythonspot.com Fuente de Internet	

		<1 %
78	www.macworld.es Fuente de Internet	<1 %
79	"Knowledge-Based Intelligent Information and Engineering Systems", Springer Science and Business Media LLC, 2004 Publicación	<1 %
80	DANIEL JIMENEZ GONZALEZ. "ALGORITMOS DE CLUSTERING PARALELOS EN SISTEMAS DE RECUPERACIÓN DE INFORMACIÓN DISTRIBUIDOS", Universitat Politecnica de Valencia, 2011 Publicación	<1 %
81	link.springer.com Fuente de Internet	<1 %
82	ore.exeter.ac.uk Fuente de Internet	<1 %
83	www.dspace.espol.edu.ec Fuente de Internet	<1 %
84	opac.pucv.cl Fuente de Internet	<1 %
85	repositorioinstitucional.uson.mx Fuente de Internet	<1 %
86	static.eoi.es Fuente de Internet	

		<1 %
87	www.tesis.uchile.cl Fuente de Internet	<1 %
88	Miguel Á. Abella-González, Pedro Carollo-Fernández, Louis-Noël Pouchet, Fabrice Rastello, Gabriel Rodríguez. "PolyBench/Python: benchmarking Python environments with polyhedral optimizations", Proceedings of the 30th ACM SIGPLAN International Conference on Compiler Construction, 2021 Publicación	<1 %
89	www.rimac.com.pe Fuente de Internet	<1 %
90	Jean Metz. "Interpretação de clusters gerados por algoritmos de clustering hierárquico", 'Universidade de Sao Paulo, Agencia USP de Gestao da Informacao Academica (AGUIA)', 2015 Fuente de Internet	<1 %
91	gsitic.wordpress.com Fuente de Internet	<1 %
92	repositorio.sangregorio.edu.ec Fuente de Internet	<1 %
93	www.hcnl.gob.mx Fuente de Internet	

		<1 %
94	www.programcreek.com Fuente de Internet	<1 %
95	dokumen.tips Fuente de Internet	<1 %
96	empiezoinformatica.wordpress.com Fuente de Internet	<1 %
97	matplotlib.org Fuente de Internet	<1 %
98	pt.scribd.com Fuente de Internet	<1 %
99	repositorio.uncp.edu.pe Fuente de Internet	<1 %
100	discovery.ucl.ac.uk Fuente de Internet	<1 %
101	papiro.unizar.es Fuente de Internet	<1 %
102	theibfr.com Fuente de Internet	<1 %
103	www.hebergementwebs.com Fuente de Internet	<1 %
104	wwwisis.ufg.edu.sv Fuente de Internet	<1 %

105	cmap.upb.edu.co Fuente de Internet	<1 %
106	docspike.com Fuente de Internet	<1 %
107	marquina88.wordpress.com Fuente de Internet	<1 %
108	python.hotexamples.com Fuente de Internet	<1 %
109	repositoriocyt.unlam.edu.ar Fuente de Internet	<1 %
110	science.donntu.edu.ua Fuente de Internet	<1 %
111	webdoc.sub.gwdg.de Fuente de Internet	<1 %
112	www.ecorfan.org Fuente de Internet	<1 %

Excluir citas	Activo	Excluir coincidencias	< 18 words
Excluir bibliografía	Activo		