UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

Procesamiento en paralelo de algoritmo de agrupamiento dinámico de trayectorias GPS

PROYECTO DE TITULACIÓN

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

AUTORES: GABRIEL RAUL LIGUA ARISTEGA
LUIS EDUARDO VIVAS MERA

TUTOR: ING. GARY XAVIER REYES ZAMBRANO, Mgs.

GUAYAQUIL – ECUADOR
2022

<table>
<tr><td colspan="3">    </td></tr>
<tr><td colspan="3">REPOSITORIO NACIONAL EN CIENCIAS Y TECNOLOGÍAS</td></tr>
<tr><td colspan="3">FICHA DE REGISTRO DE TRABAJO DE TITULACIÓN</td></tr>
<tr><td colspan="3">TÍTULO: “Procesamiento en paralelo de algoritmo de agrupamiento dinámico de trayectorias GPS”</td></tr>
<tr><td>AUTORES:
Gabriel Raul Ligua Aristega
Luis Eduardo Vivas Mera</td><td colspan="2">REVISOR:
Ing. Oscar León Granizo, M.Sc</td></tr>
<tr><td>INSTITUCIÓN: Universidad de Guayaquil</td><td colspan="2">FACULTAD: Ciencias Matemáticas y Físicas</td></tr>
<tr><td colspan="3">CARRERA: Ingeniería en Sistemas Computacionales</td></tr>
<tr><td>FECHA DE PUBLICACIÓN:</td><td colspan="2">N° DE PAGS: 170</td></tr>
<tr><td colspan="3">AREA TEMÁTICA: Investigación</td></tr>
<tr><td colspan="3">PALABRAS CLAVES: GPS, procesamiento en paralelo, Dyclee, Python, procesamiento secuencial, distancia, densidad.</td></tr>
<tr><td colspan="3">RESUMEN: El procesamiento de datos de coordenadas GPS en la actualidad es un desafio para los investigadores, cada vez más información se genera a cada momento y sin cesar, y con ello aumenta también la complejidad de esta tarea, es por esto que siempre se está buscando nuevas y mejores maneras de procesar estos ambientes en continua evolución. Un método propuesto se basa en procesar esta información con un algortimo de dos fases, una basada en distancia y la otra basada en densidad, con buenos resultados, pero al buscar una mayor optimización para este algoritmo, el siguiente trabajo de investigación propone un ambiente de procesamiento en paralelo de los datos, que mediante el uso de técnicas propias del lenguaje Python, ha logrado obtener una versión en paralelo de algoritmo Dyclee, el cual ha sido puesto a prueba en diversos escenarios, tanto de manera local como en la nube, para demostrar la reducción de los tiempos de procesamiento con respecto a la versión secuencial del algoritmo.</td></tr>
<tr><td>N° DE REGISTRO:</td><td colspan="2">N° DE CLASIFICACIÓN:</td></tr>
<tr><td colspan="3">DIRECCIÓN URL: (PROYECTO DE TITULACION EN LA WEB)</td></tr>
<tr><td>ADJUNTO PDF</td><td>SI [x]</td><td>NO [ ]</td></tr>
<tr><td>CONTACTO CON AUTORES:
Ligua Aristega Gabriel Raul
Vivas Mera Luis Eduardo</td><td>Teléfono:
0985593550
0991209362</td><td>Email:
gabrieligua1999@gmail.com
luis.vivasm@ug.edu.ec</td></tr>
<tr><td rowspan="3">CONTACTO DE LA INSTITUCIÓN</td><td colspan="2">Nombre: Ab. Juan Chávez Atocha</td></tr>
<tr><td colspan="2">Teléfono: 2307729</td></tr>
<tr><td colspan="2">Email: juan.chaveza@ug.edu.ec</td></tr>
</table>

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Titulación, "PROCESAMIENTO EN PARALELO DE ALGORITMO DE AGRUPAMIENTO DINÁMICO DE TRAYECTORIAS GPS" elaborado por el Sr. GABRIEL RAUL LIGUA ARISTEGA y el Sr. LUIS EDUARDO VIVAS MERA, **estudiantes no titulados** de la Carrera de Ingeniería en Sistemas Computacionales, Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil, previo a la obtención del Título de Ingeniero en Sistemas Computacionales, me permito declarar que luego de haber orientado, estudiado y revisado, la **apruebo** en todas sus partes.

Atentamente,

Ing. Gary Reyes Zambrano, Mgs.

TUTOR

DEDICATORIA

Dedico este trabajo a mi familia, especialmente a aquellos que estuvieron apoyándome durante toda mi formación académica. A mis padres Arturo Ligua y Margarita Aristega que con amor y dedicación supieron guiarme por el sendero hacia la superación. A mi hermano Carlos, que fue mi ejemplo a seguir y que por sus palabras de motivación me ayudaron a no desmayar y continuar con mi proceso de titulación. A todos ellos dedico este trabajo.

Gabriel Raul Ligua Aristega

El presente trabajo está dedicado a las personas más especiales de mi vida, mi esposa, mi madre y mis hermanos, quienes con su amor incondicional me ayudaron a completar esta etapa de mi vida.

Luis Eduardo Vivas Mera

AGRADECIMIENTO

Agradezco a Dios, por todas las bendiciones que me da día a día. A mis padres y hermano, que gracias a su esfuerzo y motivación me ayudaron a alcanzar mi meta, a mi enamorada Vanessa y mi amiga Angie que siempre tuvieron disponibilidad para escucharme y aconsejarme, a mi grupo de amigos por el apoyo mutuo en todos estos años de carrera, a mis docentes que con sus conocimientos y experiencias ayudaron a formarme como profesional. Agradezco también a mi tutor de tesis Ing. Gary Reyes Zambrano por su paciencia y enseñanzas brindadas en este proceso de titulación.

Gabriel Raul Ligua Aristega

Quisiera agradecer a mi esposa por su apoyo y paciencia, a mi madre por todo su amor y enseñanzas, a mis profesores y mis compañeros que formaron parte de todo este proceso de superación personal.

Luis Eduardo Vivas Mera

TRIBUNAL PROYECTO DE TITULACIÓN

Ing. Douglas Iturburu Salvador, M.Sc.
DECANO DE LA FACULTAD
CIENCIAS MATEMÁTICAS Y FÍSICAS

Ing. Lorenzo Cevallos Torres, Mgs.
DIRECTOR DE LA CARRERA DE
INGENIERÍA EN SISTEMAS
COMPUTACIONALES

Ing. Gary Reyes Zambrano, Mgs
PROFESOR TUTOR DEL PROYECTO
DE TITULACIÓN

Ing. Oscar León Granizo, M.sc
PROFESOR REVISOR DEL
PROYECTO DE TITULACIÓN

Ab. Juan Chávez Atocha, Esp.
SECRETARIO

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Proyecto de Titulación, me corresponden exclusivamente; y el patrimonio intelectual de la misma a la UNIVERSIDAD DE GUAYAQUIL".

GABRIEL RAUL LIGUA ARISTEGA

LUIS EDUARDO VIVAS MERA

CESIÓN DE DERECHOS DE AUTOR

Ingeniero

Douglas Iturburu Salvador, M.Sc.

DECANO DE LA FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

Presente.

A través de este medio indico a usted que procedo a realizar la entrega de la cesión de derechos de autor en forma libre y voluntaria del trabajo de titulación "**Procesamiento en paralelo de algoritmo de agrupamiento dinámico de trayectorias GPS**", realizado como requisito previo para la obtención del Título de Ingeniero en Sistemas Computacionales de la Universidad de Guayaquil.

Guayaquil, septiembre de 2022.

Gabriel Raul Ligua Aristega
C.I. N° 0952492742

Luis Eduardo Vivas Mera
C.I. N° 0910969765

UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

PROCESAMIENTO EN PARALELO DE ALGORITMO DE AGRUPAMIENTO DINÁMICO DE TRAYECTORIAS GPS

Proyecto de Titulación que se presenta como requisito para optar por el título de

INGENIERO EN SISTEMAS COMPUTACIONALES

Autores: Gabriel Raul Ligua Aristega

C.I. N° 0952492742

Luis Eduardo Vivas Mera

C.I. N° 0910969765

Tutor: Ing. Gary Reyes Zambrano, Mgs.

Guayaquil, septiembre de 2022

Mes Año

CERTIFICADO DE ACEPTACIÓN DEL TUTOR

En mi calidad de Tutor del Proyecto de Titulación, nombrado por el Consejo Directivo de la Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil.

CERTIFICO:

Que he analizado el Proyecto de Titulación presentado por los estudiantes **GABRIEL RAUL LIGUA ARISTEGA, LUIS EDUARDO VIVAS MERA**, como requisito previo para optar por el Título de Ingeniero en Sistemas Computacionales cuyo proyecto es:

PROCESAMIENTO EN PARALELO DE ALGORITMO DE AGRUPAMIENTO DINÁMICO DE TRAYECTORIAS GPS

Considero aprobado el trabajo en su totalidad.

Presentado por:

Ligua Aristega Gabriel Raul **C.I.:** 0952492742

Vivas Mera Luis Eduardo **C.I.:** 0910969765

Tutor: _____

Firma

Guayaquil, septiembre de 2022

Mes Año

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

AUTORIZACIÓN PARA PUBLICACIÓN DE PROYECTO DE TITULACIÓN EN FORMATO DIGITAL

1. Identificación del Proyecto de Titulación

Nombre del Estudiante: Gabriel Raul Ligua Aristega	
Dirección: Pascuales, Coop. En Pie de Lucha Manzana 87 solar 11	
Teléfono: 0985593550	**Email:** gabrieligua1999@gmail.com

Nombre del Estudiante: Luis Eduardo Vivas Mera	
Dirección: Urdesa Norte 220 Ave. Primera y calle quinta	
Teléfono: 0991209362	**Email:** luis.vivasm@ug.edu.ec

Facultad: Facultad de Ciencias Matemáticas y Físicas
Carrera: Carrera de Ingeniería en Sistemas Computacionales
Proyecto de Titulación al que opta: Ingeniero en Sistemas Computacionales
Profesor Tutor: Ing. Gary Reyes Zambrano, Mgs.

Título del Proyecto de Titulación: Procesamiento en paralelo de algoritmo de agrupamiento dinámico de trayectorias GPS.

Palabras Claves: : GPS, procesamiento en paralelo, Dyclee, Python, procesamiento secuencial, distancia, densidad.

2. Autorización de Publicación de Versión Electrónica del Proyecto de Titulación
A través de este medio autorizo a la Biblioteca de la Universidad de Guayaquil y a la Facultad de Ciencias Matemáticas y Físicas a publicar la versión electrónica de este Proyecto de Titulación.

Publicación Electrónica:

Inmediata	**X**	Después de 1 año	

Firma Estudiante:

Ligua Aristega Gabriel Raul — **C.I.:** 0952492742

Vivas Mera Luis Eduardo — **C.I.:** 0910969765

3. Forma de envío:

El texto del Proyecto de Titulación debe ser enviado en formato Word, como archivo .docx, .RTF o .Puf para PC. Las imágenes que la acompañen pueden ser: .gif, .jpg o .TIFF.

DVDROM ☐ CDROM ☐

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ABREVIATURAS

AWS	Amazon Web Services
CSV	Valores Separados por Comas
EMR	Elastic Map Reduce
EC2	Elastic Compute Cloud
S3	Simple Storage Services
ML	Machine Learning
GPS	Global Positioning System
Ing.	Ingeniero
GIL	Global Interpreter Lock
HDFS	Hadoop File System
VSC	Visual Studio Code
M.Sc.	Máster
Mtra.	Maestra
UG	Universidad de Guayaquil
URL	Localizador de Fuente Uniforme
WWW	World Wide Web (Red Mundial)
API	Application Programming Interface

SIMBOLOGÍA

s	Desviación estándar
N	Población
a_x	Longitud (eje horizontal)
a_y	Latitud (eje vertical)
s	Segundo
c	Celda

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

PROCESAMIENTO EN PARALELO DE ALGORITMO DE AGRUPAMIENTO DINÁMICO DE TRAYECTORIAS GPS

Autores: Gabriel Raul Ligua Aristega
C.I. N° 0952492742
Luis Eduardo Vivas Mera
C.I. N° 0991209362

Tutor: Ing. Gary Reyes Zambrano, Mgs.

RESUMEN

El procesamiento de datos de coordenadas GPS en la actualidad es un desafio para los investigadores, cada vez más información se genera a cada momento y sin cesar, y con ello aumenta también la complejidad de esta tarea, es por esto que siempre se está buscando nuevas y mejores maneras de procesar estos ambientes en continua evolución. Un método propuesto se basa en procesar esta información con un algortimo de dos fases, una basada en distancia y la otra basada en densidad, con buenos resultados, pero al buscar una mayor optimización para este algoritmo, el siguiente trabajo de investigación propone un ambiente de procesamiento en paralelo de los datos, que mediante el uso de técnicas propias del lenguaje Python, ha logrado obtener una versión en paralelo de algoritmo Dyclee, el cual ha sido puesto a prueba en diversos escenarios, tanto de manera local como en la nube, para demostrar la reducción de los tiempos de procesamiento con respecto a la versión secuencial del algoritmo.

Palabras clave: GPS, procesamiento en paralelo, Dyclee, Python, procesamiento secuencial, distancia, densidad.

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

PARALLEL PROCESSING OF A GPS TRAJECTORY DYNAMIC CLUSTERING ALGORITHM

Authors: Gabriel Raul Ligua Aristega
C.I. N° 0952492742
Luis Eduardo Vivas Mera
C.I. N° 0991209362

Tutor: Ing. Gary Reyes Zambrano, Mgs.

ABSTRACT

GPS data processing nowdays is a challenge for researchers, more and more information is generated every moment and without ceasing, and with it also increases the complexity of this task, this is why we are always looking for new and better ways to process these constantly evolving environments. A proposed method is based on processing this information with a two-phase algorithm, one based on distance and the another based on density, with good results, but when looking for a greater optimization for this algorithm, the following research work proposes a parallel data processing environment, which through the use of techniques of the Python language, has managed to obtain a parallel version of the Dyclee algorithm, which has been tested in various scenarios, both on premises and in the cloud, to demonstrate the reduction of processing times in comparison to the sequential version of the algorithm.

Key words: GPS, parallel processing, Dyclee, Python, sequential processing, distance, density.

INTRODUCCIÓN

La evolución de la tecnología ha provocado avances significativos en muchas áreas de estudio, que lleva consigo la generación de cantidades inimaginable de datos de diversas procedencias, a cada instante; a esto se lo conoce como Big Data. Todos estos datos se producen de diferentes formas y se encuentran almacenados en distintos repositorios según el contexto en el que se requieran. La importancia de los datos crece constantemente puesto que son utilizados para realizar diversos estudios de temáticas distintas, como proyecciones financieras que necesitan las grandes compañías para la toma de decisiones, trabajos de investigación cuyo fin consiste en observar fenómenos y obtener conclusiones acertadas apoyándose de datos procesados o generando nuevos datos a través de la experimentación; es decir, los datos son la base de cualquier estudio.

Todo estudio que se requiera realizar necesita procesar grandes cantidades de volúmenes de datos para obtener resultados óptimos, de acuerdo al tipo de temática que se esté investigando. En el caso de investigaciones referentes a trayectorias vehiculares, congestionamiento vehicular, flujo de tránsito vehicular o cualquier tema que involucre estudios inmersos en el transito; necesitan datos de vehículos esencialmente, que por lo general son las trayectorias que siguen los vehículos para cumplir con ciertos recorridos. Estos datos son recopilados a cada instante por sensores, aplicativos incorporados en los vehículos o celulares inteligentes que son capaces de mostrar la ubicación en tiempo real de los vehículos. Al tratarse de millones de vehículos circulando por las calles de todos los países del mundo y estimando que cada uno genera muchos datos de forma constante, se puede estimar que las cifras de datos son infinitas. En estos casos resulta muy complicado procesar todos estos datos, haciendo uso de algoritmos de algoritmos de inteligencia artificial, de una forma correcta y en el menor tiempo posible. Si se pudiese implementar una arquitectura en estos algoritmos que permita el procesamiento en paralelo de grandes volúmenes de datos, atraería ventajas en

cuanto al tiempo de procesamiento de datos se refiere, esto podría contribuir en gran medida con el proceso investigativo de estudios referentes al flujo de tránsito vehicular, o incluso, cualquier tipo de estudio que requiera procesamiento de datos para el análisis; es por ello que se ha considerado desarrollar el siguiente proyecto.

Este proyecto se centra especialmente en la implementación de una arquitectura en paralelo de un algoritmo de clustering para el procesamiento de datos de trayectorias GPS. La investigación consistirá principalmente en una abundante revisión de información en buscadores académico y bases de datos científicas con la finalidad de tener un soporte teórico para realizar la implementación del algoritmo de agrupamiento en paralelo; para ello, es muy importante considerar diversas características, como el diseño de la arquitectura, el lenguaje de programación, librerías y sintaxis adecuadas para una implementación en paralelo, entre otros. Además, de que se espera procesar datos de trayectorias GPS por lo que será necesario realizar ciertas adaptaciones en el algoritmo de agrupamiento y de esta manera realizar las experimentaciones que se requieran haciendo uso de datasets obtenidos de repositorios públicos.

Este proyecto de investigación se encuentra conformado por cuatro capítulos, que se detallan a continuación de forma general:

Capítulo I: En este primer capítulo se detalla todo el proceso del planteamiento del problema, los objetivos considerados en el estudio, el alcance del proyecto, la importancia y las limitaciones que se presentan.

Capítulo II: En este segundo capítulo se encuentran los antecedentes de estudio, la fundamentación teórica, la revisión sistemática y su meta análisis, la pregunta científica a contestarse, las variables de estudio y las definiciones conceptuales referentes al tema de estudio.

Capítulo III: En este tercer capítulo se redacta la metodología que sigue este proyecto, la modalidad y el tipo de investigación, el diseño metodológico de la investigación, la selección de la población, se detallan los entregables y beneficiarios del estudio, la propuesta en conjunto con su criterio de validación y el detalle de los resultados obtenidos.

Capítulo IV: En este cuarto capítulo, se exponen las conclusiones y recomendaciones del presente estudio, además de ciertas sugerencias para posibles trabajos futuros.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

Descripción de la situación problemática

Ubicación del problema en un contexto

En la actualidad, el manejo del Big Data puede llegar a ser fundamental para el análisis de diferentes escenarios que requieren un estudio minucioso de datos y contribuir a una mejor toma de decisiones. Son diversas las áreas que pueden hacer uso del Big Data para realizar los estudios pertinentes y tomar decisiones de acuerdo con los resultados obtenidos, una de ellas es el área empresarial, que siempre se encuentra en constante cambio y mejora continua con respecto a los productos o servicios que ofrecen; puesto que es un área muy competitiva. El análisis del Big Data también es imprescindible en el área de la investigación, donde se estudian fenómenos de diversa índole con el fin de formular nuevas teorías o generar nuevo conocimiento.

Por ello, Big Data es un concepto que ha ido adquiriendo mucha acogida en los últimos años y consiste en la fabricación de grandes volúmenes de datos mediante el empleo de diferentes dispositivos y sensores. Con la tecnología actual, la mayoría de estos datos se crean de forma instantánea, por ejemplo, los datos geolocalizados que deja una persona al trasladarse de un punto (A) a un punto (B) se registran mediante el GPS que posee su teléfono inteligente, o incluso, mediante el sistema de navegación que tienen incorporado algunos autos modernos (Gutiérrez et al., 2016).

Los sistemas de transportación inteligente procesan grandes cantidades de datos de trayectorias GPS que generan los vehículos en la calle en tiempo real. Los datos obtenidos deben ser analizados para ser convertidos en conocimiento para luego poder ser utilizada de respaldo en la toma de decisiones (Reyes et al., 2020).

La generación automática, en abundancia, de datos geolocalizados ha permitido realizar estudios enfocados a la gestión del tráfico en diversas partes del mundo, lo que ayuda a identificar las causas de la congestión vehicular. La congestión vehicular es un problema que ha ganado fuerza en los últimos años y afecta de diferentes formas a la población en general, y esto motiva a realizar estudios que contribuyan a buscar soluciones efectivas, por esta razón, el procesamiento y análisis de trayectorias GPS forma parte fundamental en el proceso.

El acceso a grandes volúmenes de datos GPS de los vehículos terrestres, entre las que destacan las coordenadas angulares (longitud y latitud), tiempo y velocidad; facilita en gran medida el procesamiento y análisis de trayectorias GPS. Existen variedad de algoritmos diseñados para realizar el procesamiento de datos en base a características específicas, entre ellos se encuentran los algoritmos de clustering o de agrupamiento.

El proceso de clustering es un instrumento muy beneficioso para la ciencia de datos, que consiste en agrupar conjunto de datos con características similares e irlos clasificando de acuerdo con esas cualidades (Sinaga & Yang, 2020). Entre los algoritmos de clustering se encuentran los que son basados en densidad, estos identifican grupos de objetos de densidad elevada en regiones dispersas, además, encuentran grupos de forma arbitraria y filtran con facilidad datos con ruido (Loh & Park, 2014). A su vez, existen algoritmos de agrupamiento dinámico, estos no asumen la estructura de datos con anticipación, sino que la encuentran gradualmente modificando la estructura de agrupamiento (Reyes, Lanzarini, Estrebou, et al., 2021).

Independientemente de la forma en que agrupen datos los diferentes algoritmos de clustering, estos deben procesar cantidades inimaginables de volúmenes de datos de trayectorias GPS en cierto tiempo, lo que podría afectar significativamente el tiempo de espera de los resultados finales. Otro aspecto a tener en consideración es la codificación del algoritmo, puesto que, al estar regida bajo una arquitectura secuencial podría restar eficiencia en los procesos que realiza y aumentar el tiempo de espera. Una potencial solución sería la implementación de una arquitectura en paralelo que permita realizar varios procesos de forma simultánea, en este caso, permitiría el procesamiento de grandes volúmenes de datos en tiempos más cortos lo que aumentaría significativamente la eficiencia del algoritmo empleado para el análisis de trayectorias GPS.

Situación conflicto nudos críticos

El procesamiento de grandes volúmenes de datos de trayectorias GPS podría mejorar exponencialmente mediante el paralelismo o programación en paralelo, sin embargo, existen lenguajes de programación como R, que se utilizan especialmente para el análisis de datos; que no pueden implementar multihilos, multisesión y programación en paralelo al 100 por ciento, puesto que cuenta con ciertas restricciones o limitaciones propias del lenguaje. Por esta razón, es importante considerar otras alternativas que puedan contribuir a la optimización de los algoritmos de procesamiento dinámico.

Existen lenguajes de programación como Java y Python que permiten la programación en paralelo mediante la implementación de librerías, frameworks y métodos propios del lenguaje. Migrar a otro lenguaje de programación puede ser una tarea muy laboriosa, sin embargo, si lo que se busca es más eficiencia y optimización de procesos para realizar estudios referentes a las trayectorias GPS.

Al momento de realizar paralelismo, es importante considerar ciertos aspectos, puesto que no solo se trata de utilizar un lenguaje de programación adecuado, sino también, de

disponer de recursos de hardware suficientes; puesto que las tareas que se vayan a ejecutar de forma simultánea requieren recursos físicos de memoria.

Delimitación del problema

El presente proyecto de investigación se encuentra inmerso en el campo de las Ciencias Básicas, Bioconocimiento y Desarrollo Industrial, además, corresponde al área de Tecnologías de la Información y Telecomunicaciones. Su aspecto comprende el procesamiento de grandes volúmenes de datos de trayectorias GPS y el tema es "Procesamiento en paralelo de algoritmo de agrupamiento dinámico de trayectorias GPS". La tabla 1 muestra de forma clara la delimitación del problema.

Tabla 1

Delimitación del problema

Delimitador	Descripción
Campo	Ciencias Básicas, Bioconocimiento y Desarrollo Industrial
Área	Tecnologías de la Información y Telecomunicaciones
Aspecto	Procesamiento de grandes volúmenes de datos de trayectoria GPS
Tema	Procesamiento en paralelo de algoritmo de agrupamiento dinámico de trayectorias GPS

Nota: En esta tabla se abordan los términos de estudio aplicados para la delimitación del problema con respecto al entorno en donde se despliega la problemática. La elaboración de esta tabla es propia y la fuente surge a partir del trabajo de investigación.

Evaluación del Problema

Los aspectos generales de evaluación del problema son:

- **Delimitado:** En el siguiente estudio para la implementación de un ambiente de procesamiento en paralelo para un algoritmo de agrupamiento dinámico de trayectorias GPS se utilizará como base un algoritmo de clustering basado en centroides y también basado en densidad que procesará los datos en paralelo con la ayuda de una arquitectura que se va a diseñar y administrar en Amazon Web Services.

- **Claro:** El algoritmo programado en python se ejecutará en paralelo para poder ser adaptado al servicio EMR de Amazon Web Services que permite procesar grandes volúmenes de datos.
- **Evidente:** La ejecución en paralelo permitirá que la gran cantidad de datos se procesen más rápida y eficientemente.
- **Relevante:** La programación en paralelo es un mecanismo que puede ayudar a satisfacer la necesidad de procesar la creciente cantidad de información que se generan los diferentes dispositivos de trayectorias GPS.
- **Original:** Actualmente no existe una versión del algoritmo Dyclee que se ejecute paralelo.
- **Factible:** Los recursos necesarios para este estudio como son los datasets y el servicio EMR de Amazon Web Services están disponibles, solo el algoritmo Dyclee que es en el que se va a basar el estudio requiere modificaciones para poder procesar información en paralelo.

Causas y consecuencias del problema

Tabla 2

Matriz de causas y consecuencias del problema

Causas	Consecuencias
C1. Procesamiento secuencial de algoritmo de clustering para grandes cantidades de datos.	E1.1 Aumenta el tiempo de procesamiento de grandes volúmenes de datos.
C2. Presencia de ciertas restricciones en lenguaje de programación R para usar multiprocesamiento.	E1.2 No se aprovecha en su totalidad las características que ofrece el paralelismo.
C3. Carencia de recursos para saber cómo modificar algoritmo en secuencia a paralelo.	E2. Solo puede procesar datos en forma secuencial.
C4. Implementación incompleta de algoritmo de clustering.	
C5. Implementación de procesamiento en paralelo en la nube.	E3. Aumenta el tiempo de investigación.
	E4. Dificulta la modificación para procesamiento en paralelo.
	E5. Puede resultar costoso.

Nota: En la presente tabla se enuncian las causas y consecuencias del problema. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

Formulación del problema

¿Cómo incide en el tiempo de procesamiento la implementación de un ambiente de procesamiento en paralelo para un algoritmo de agrupamiento dinámico de trayectorias GPS?

Objetivos del proyecto

Objetivo general

Implementar un ambiente de procesamiento en paralelo para un algoritmo de agrupamiento dinámico de trayectorias GPS con el uso de software de una plataforma de computación en la nube, para poder procesar grandes volúmenes de conjuntos de datos de trayectorias GPS.

Objetivos específicos

1. Realizar la migración de un algoritmo de agrupamiento dinámico de trayectorias GPS desde Lenguaje R a un Lenguaje que permita multihilos, multisesión y programación en paralelo.
2. Diseñar y programar un algoritmo de programación en paralelo de un conjunto de datos de trayectorias GPS y realizar la codificación utilizando las librerías, métodos y sintaxis adecuada.
3. Implementar un ambiente de procesamiento en paralelo que permita procesar un algoritmo de agrupamiento dinámico de trayectorias GPS, con el uso de software de una plataforma de computación en la nube.
4. Realizar experimentos con grandes volúmenes de conjuntos de datos de trayectorias GPS.
5. Validar los resultados mediante estadística descriptiva y juicio de expertos.
6. Elaborar un artículo científico del presente estudio.

Alcance del proyecto

El presente proyecto tiene la finalidad de implementar una arquitectura en paralelo en un algoritmo de agrupamiento dinámico para el procesamiento de trayectorias GPS a través de diversas herramientas que permitan montar esta arquitectura. Entre los alcances que se esperan alcanzar en la elaboración de este proyecto, se encuentran los siguientes:

1. Investigar acerca de los lenguajes de programación que permiten trabajar con multihilos, multisesión y programación en paralelo, considerando la sintaxis, librerías, funciones y métodos necesarios para efectuar la implementación en paralelo. Elegir un lenguaje de programación apropiado.
2. Investigar acerca del algoritmo de agrupamiento dinámico, para comprender su funcionamiento y demás características.

3. Realizar la migración del algoritmo de clustering a un lenguaje que trabaje con multihilos, multisesión y programación en paralelo.
4. Investigar y seleccionar dos conjuntos de datos provenientes de repositorios públicos que incluyan los datos requeridos para el desarrollo del presente estudio.
5. Adaptación del algoritmo de clustering para que pueda procesar datos de trayectorias GPS.
6. Codificar una sección de código que permita visualizar de forma clara los mapas y grupos formados durante el proceso de agrupación del algoritmo.
7. Diseñar una arquitectura en paralelo para el algoritmo de agrupamiento dinámico de trayectorias GPS.
8. Programar, mediante el uso de librerías y métodos, la implementación en paralelo del algoritmo de clustering para agrupar trayectorias vehiculares.
9. Implementar el algoritmo en paralelo haciendo uso de un software de una plataforma de computación en la nube.
10. Realizar los experimentos con el uso de grandes volúmenes de datos de trayectorias GPS para la medición de las variables de estudio.
11. Aplicar estadística descriptiva a las variables de estudio y realizar su posterior análisis de resultados.
12. Realizar un juicio de expertos como criterio de validación de la propuesta de investigación.
13. Elaboración de un artículo científico referente a la investigación realizada.

Es importante mencionar que la implementación del algoritmo, haciendo uso de un software en la plataforma Amazon Web Services, no hace uso de la abstracción Map-reduce puesto que presenta otras consideraciones no aplicadas en el código de programación del algoritmo de clustering y no forma parte del alcance del presente estudio.

Justificación e importancia

Actualmente, la recopilación de grandes cantidades de información es parte del día a día de todo tipo de organizaciones, negocios, instituciones y demás. Esta gran cantidad de información se ha vuelto tremendamente difícil de procesar, por su gran volumen, de forma eficiente.

Por lo anteriormente mencionado, y la relevancia del manejo eficiente de la información para la economía el crecimiento y desarrollo es que constantemente se buscan nuevas maneras y técnicas que den respuesta a esta imperiosa necesidad.

Este proyecto se enfoca en modificar un algoritmo que funciona procesando la información de coordenadas GPS de manera secuencial, y adaptarlo para que pueda procesar la misma información de manera paralela, es decir, que la información pueda ser procesada por diferentes partes al mismo tiempo y con esto lograr que las grandes cantidades de información que generan las coordenadas GPS puedan ser procesadas de una manera más rápida y eficiente.

Una trayectoria GPS puede definirse por un conjunto de ubicaciones geográficas, cada una representada por su latitud y longitud en un momento determinado (Reyes, Lanzarini, Estrebou, et al., 2021).

En el caso de las trayectorias GPS, nuestros dispositivos están siempre recopilando información relacionada a nuestra ubicación y debido a que la cantidad de dispositivos que recopilan este tipo de información en la actualidad aumenta de manera vertiginosa y con ello la información que se recopila aumenta de manera exponencial es necesario encontrar siempre descubrir nuevas maneras de procesar esta información de manera eficiente.

El constante incremento del volumen de tráfico en las grandes ciudades causa problemas en el flujo vehicular, por eso el análisis de los datos que los sistemas de monitoreo de los vehículos generan se vuelve relevante (Reyes, Lanzarini, Estrebou, et al., 2021).

Es por esto por lo que se propone que un logaritmo de procesamiento dinámico en paralelo sería trascendental para lograr reducir los tiempos de procesamiento y generar los resultados obtenibles de manera rápida y sin fallos.

Limitaciones del estudio

Los recursos de los equipos de cómputo, puede afectar el desempeño del algoritmo al momento de su ejecución, por lo tanto, los resultados de los tiempos de ejecución pueden variar según la computadora donde se realicen las experimentaciones.

La selección de conjuntos de datos de repositorios públicos puede ser restringida en ciertos casos y dificultar su accesibilidad; además, pueden presentar gran cantidad de ruido.

Existen datasets con datos incompletos que dificulta el cumplimiento de ciertos objetivos, como la agrupación en base a rangos de velocidad.

La implementación de un ambiente en paralelo en la nube no emplea la abstracción Map- reduce puesto que presenta otras consideraciones no contempladas en el código fuente del algoritmo. Solo se realizan las configuraciones necesarias para ejecutar el código en la nube.

CAPÍTULO II

MARCO TEÓRICO

Antecedentes del estudio

El avance constante de la tecnología ha abierto muchas puertas hacia la automatización e innovación de procesos mediante nuevas técnicas y métodos computacionales que permiten el procesamiento de grandes volúmenes de datos de forma más efectiva. Los siguientes estudios han implementado arquitecturas en paralelo en diferentes temáticas de investigación y han demostrado resultados óptimos:

Lapeira et al., (2017) realizaron un estudio con un algoritmo de minería de datos de aprendizaje no supervisado que ayuda a identificar predicados difusos de forma normal, este algoritmo recibe el nombre de FuzzyPred. En este método, cada predicado generado se valora en cada registro de la base de datos, por lo que el volumen de la base de datos juega un rol fundamental en el tiempo de respuesta del algoritmo. Ellos identificaron que al realizar todo el proceso de forma secuencial, se desaprovechan los recursos de hardware que brindan las nuevas tecnologías en la actualidad para el procesamiento de grandes volúmenes de datos, además, de llegar a obtener largos tiempos de ejecución del algoritmo. Es por ello, que optaron por implementar una versión con arquitectura en paralelo del algoritmo FuzzyPred, basada en la cantidad de datos que cada hilo de procesamiento puede procesar de forma simultánea e independiente. Los resultados obtenidos demostraron que el algoritmo

en paralelo puede ser 10 veces más rápido con respecto al secuencial y por esta razón se considera que puede ser efectivo ante bases de datos muy grandes.

Zhang et al., (2013) propusieron un algoritmo de K-means paralelo con un modelo de programación de paso de mensajes llamado MPI (Message Passing Interface). El algoritmo en paralelo recibió el nombre de MKmeans y permite aplicar de forma efectiva el algoritmo de agrupamiento en el ambiente paralelo. El estudio experimental realizado evidenció que MKmeans demuestra estabilidad y portabilidad, además, de emplear poco tiempo de procesamiento en grandes volúmenes de conjuntos de datos.

(Hu et al., 2018) introduce un nuevo algoritmo GA-DBSCAN basado en un marco de programación MapReduce, esto debido a que el algoritmo GA-DBSCAN original no era lo suficientemente eficiente para manejar una gran cantidad de información. Bajo la premisa de asegurar que la partición de los datos sea razonable se usó el algoritmo FPRBP para lograr una división exacta del Dataset y así asegurar que no haya áreas superpuestas. Con esto usaron las funciones de Map y Reduce para lograr una paralelización del algoritmo, y con esto resolvieron el problema de la baja eficiencia.

Dafir et al., (2021) realizó un estudio comparativo entre diferentes clases de algoritmos de agrupamiento en paralelo, entre estos se encuentra k-means, DBSCAN, aseguran que la naturaleza de estos algoritmos los hace propicios para ser paralelizados en diferentes diseños y en diferentes plataformas para manejar una gran cantidad de datos. Mediante pruebas con diferentes versiones de los algoritmos según la plataforma de Big Data que se iba a utilizar se pudo analizar el rendimiento de los algoritmos en su versión paralela, en el caso de k-means, el cual es muy popular y fácil de implementar, tiene desventajas como la determinación del número adecuado de clusters k, además de problemas de escalabilidad cuando existen valores esparcidos, pero si es apto para llevar a cabo las tareas de los clusters y las actualizaciones de los centroides en paralelo con un

modelo de MapReduce. En cuanto al algoritmo DBSCAN que lleva a cabo el agrupamiento en dos pasos, que también pueden ser paralelizados en un modelo MapReduce en los pedazos de un dataset, ofrece ventajas como ser capaz de descubrir clusters de tamaños arbitrarios y no necesita establecer un número de clusters para su funcionamiento, sin embargo, la estimación de los parámetros de entrada es una tarea compleja y crítica para su buen funcionamiento, así como el resto de los algoritmos tradicionales de agrupamiento sufre de problemas de escalabilidad.

Fundamentación teórica

GPS

"El Sistema de Posicionamiento Global (GPS) fue desarrollado por el Departamento de Defensa de los Estados Unidos para proporcionar un sistema de navegación por satélite para el Ejercito de los Estados Unidos" (Villegas, 2021, p.44).

El GPS consiste en "un sistema de radionavegación basado en satélites que puede proporcionar geolocalización en tiempo real e información horaria a un receptor GPS en cualquier parte de la Tierra" (S. Wang et al., 2020).

El análisis de datos espaciales generados por diversas tecnologías de localización usadas en la actualidad es de vital importancia para la elaboración de planes para el desarrollo urbano (Reyes, Lanzarini, Hasperue, et al., 2021).

Actualmente, este sistema está presente en diversos ámbitos y al alcance de cualquier usuario que por medio de los diferentes dispositivos tecnológicos que existen, le permite determinar su ubicación en tiempo real y desplazarse con mayor seguridad hacia un punto específico. Los diferentes medios de transportes (terrestre, marítimo y aéreo) también incorporan este sistema como apoyo para la navegación, lo que permite recolectar datos indispensables para el estudio de temáticas relacionadas a la congestión vehicular.

Trayectorias GPS

Una trayectoria GPS se presenta como una secuencia discreta de puntos de coordenadas geográficas (Reyes Zambrano, 2019).

Cada punto de trayectoria de cualquier objeto en movimiento representa una ubicación con marca de tiempo y está modelado por cuatro elementos importantes (x, y, s, c) donde a_x es la longitud, a_y es la latitud, a_s es la marca de tiempo y a_c es la celda a la que está asignada el punto a. Tanto latitud como longitud se expresan mediante números reales, la marca de tiempo muestra su exactitud en segundos y la celda se reconoce mediante un identificador (ID) en número entero (J. Wang et al., 2021).

Minería de datos

Jothi et al., (2015) indican que "la minería de datos es el proceso de descubrimiento y extracción de patrones en el que se involucra una gran cantidad de datos" (p.306). Han et al., (2012) mencionan que "las fuentes de datos pueden incluir bases de datos, almacenes de datos, la web, otros repositorios de información, o datos transmitidos al sistema dinámicamente" (p.8).

Existen variedades de técnicas de minería de datos que han sido desarrolladas para ser implementadas en proyectos de minería de datos. Entre las técnicas de minería de datos más reconocidas se encuentran: asociación, clasificación, agrupamiento, árbol de decisión, predicción, redes neuronales, entre otros. Cada técnica tiene sus propias normas y metodologías de aplicación para cada tipo de problema (Osman, 2019, p.1).

Machine learning

Según Carleo et al., (2019) "el Machine learning (ML) abarca una amplia gama de algoritmos y herramientas de modelado que se utilizan para una gran variedad de tareas de

procesamiento de datos, que ha entrado en la mayoría de las disciplinas científicas en los últimos años".

El Machine learning es un proceso donde se emplean modelos matemáticos de datos para que los equipos puedan aprender sin necesidad de alguna instrucción directa. Se considera un subconjunto de la inteligencia artificial (IA). El aprendizaje automático utiliza algoritmos para localizar patrones en los datos y usarlos para construir modelos de datos que pueden realizar predicciones. La cantidad de datos y la adquisición de experiencia es fundamental para obtener resultados de aprendizaje automático más precisos, esto se asemeja a la manera en que los humanos van mejorando con la práctica (Microsoft Azure, 2018).

Mahesh, (2020) indica que "el propósito del aprendizaje automático es aprender de los datos. Se han realizado muchos estudios sobre cómo hacer que las máquinas aprendan por sí mismas sin ser programadas explícitamente" (p.381).

Reconocimiento de patrones

El reconocimiento de patrones consiste en una disciplina científica cuyo propósito es la clasificación de objetos o datos en una colección de categorías o clases. Podría tratarse de imágenes, señales de onda o cualquier tipo de medida que necesita clasificarse. El reconocimiento de patrones forma parte integral de la pluralidad de los sistemas de inteligencia artificial diseñados para la toma de decisiones (Theodoridis y Koutroumbas, 2009, p.1).

Aprendizaje supervisado

Recibe el nombre de "supervisado" debido a que su entrenamiento requiere de un conjunto de datos clasificados y etiquetados con antelación. El algoritmo efectuará predicciones sobre este conjunto de datos de entrenamiento y las comparará con las etiquetas. En base a los errores obtenidos y la serie de iteraciones que pueda realizar el algoritmo se irá

adecuando el modelo contribuyendo a un aprendizaje progresivo (Barrionuevo et al., 2020, p.494).

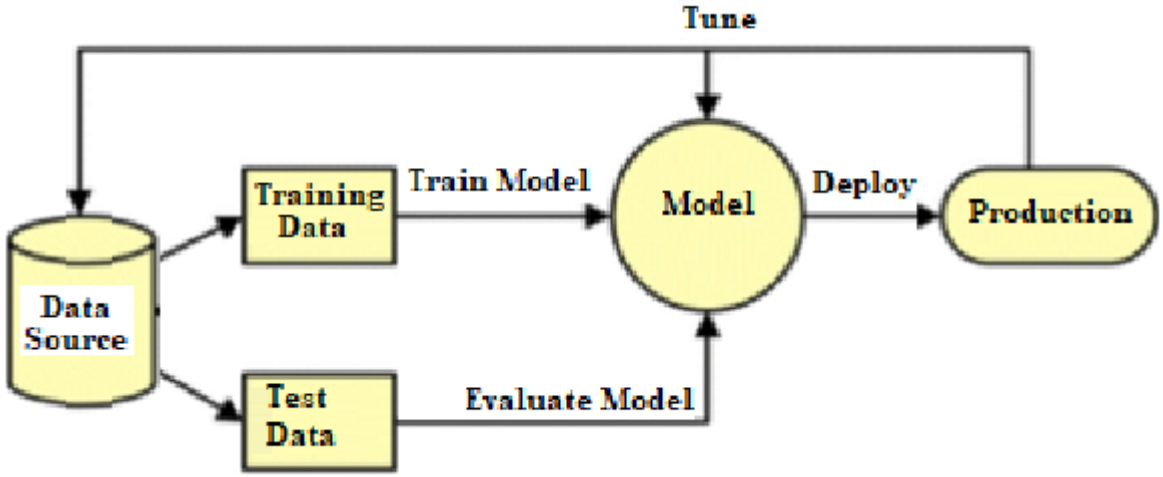
Para emplear métodos de aprendizaje supervisado se comienza por medio de datos de entrada, estos a su vez generan resultados o salidas que forman parte de un conjunto de sucesos establecidos. En base a estos datos, se puede construir un modelo predictivo, que permitirá predecir los resultados que lograrán los nuevos eventos. El proceso de aprendizaje supervisado se efectúa a través de un entrenamiento controlado por un supervisor que indica las respuestas que debe obtener a partir de datos de entrada específicos. El supervisor revisa y comprueba las salidas obtenidas por el algoritmo y en caso de que no coincida con lo que se espera, se procederá a corregir la arquitectura del modelo con la finalidad de mejorar sus predicciones (Cardenas et al., 2015).

Según González, (2015) "un ejemplo de aprendizaje supervisado es el problema de clasificación, en el cual la variable dependiente corresponde a un atributo que indica a qué clase (por ejemplo, el caso de un problema de diagnóstico médico) pertenece una muestra particular" (p.77).

A continuación, Mahesh, (2020) muestra en la figura 1 el flujo de trabajo de los algoritmos de aprendizaje supervisado.

Figura 1

Flujo de trabajo de aprendizaje supervisado



Nota: Se observa el flujo de trabajo que sigue el aprendizaje supervisado donde se destinan ciertos datos de la fuente de datos para el entrenamiento del modelo y otros datos de prueba para la evaluación del modelo. Se evaluan las predicciones del modelo y se ajusta hasta conseguir mejores resultados predictivos. Tomado de (Mahesh, 2020).

Aprendizaje no supervisado

Las técnicas de aprendizaje no supervisado carecen de intervención humana para generar conjuntos de datos preclasificados y ser presentados a algoritmos de aprendizaje. El objetivo del aprendizaje no supervisado es encontrar patrones observando la división y composición de los datos que se le presentan. Las técnicas de clustering son un ejemplo de aprendizaje no supervisado muy conocidas (Godoy, 2017, p.106).

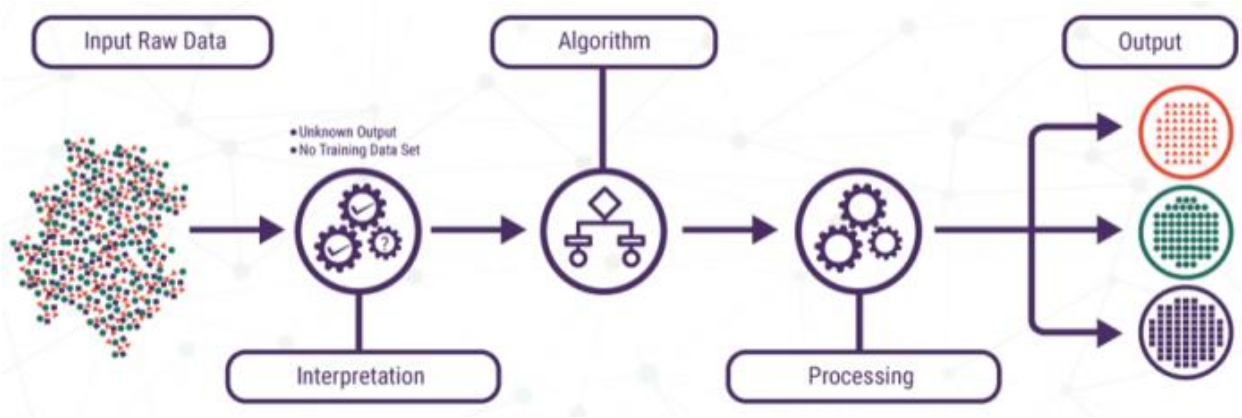
González, (2015) indica que “en los modelos de aprendizaje no supervisado no hay una distinción entre variables dependientes y no dependientes, en este caso se pretende encontrar la estructura subyacente que explique la estructura de los datos” (p.77).

Mahesh, (2020) indica que “los algoritmos de aprendizaje no supervisados aprenden pocas características de los datos. Cuando se introducen nuevos datos, se utilizan las características aprendidas previamente para reconocer la clase de los datos” (p.383).

A continuación, Farnos, (2018) muestra de forma simplificada en la figura 2 el proceso que siguen las técnicas de aprendizaje no supervisado para generar los resultados con la implementación de algún algoritmo de clustering.

Figura 2

Flujo de trabajo de aprendizaje no supervisado



Nota: Se observa el flujo de trabajo que sigue el aprendizaje no supervisado donde se ingresan datos sin procesar, sin datos de entrenamiento y mediante la implementación de un algoritmo de clustering se procesan esos datos y los agrupa de acuerdo a ciertas características objeto de estudio. Tomado de (Farnos, 2018).

Clustering

Según Reyes Zambrano et al., (2022) "las técnicas de clustering han sido utilizadas en el análisis de trayectorias desde hace varios años. Por lo general, se trata de adaptaciones de los algoritmos convencionales utilizando métricas de similitud especialmente diseñadas para trayectorias"(p.2).

Clustering es el proceso de agrupación de objetos similares, de tal manera que cada elemento dentro de un clúster tenga características similares entre sí y que difieran de los elementos de otras agrupaciones (Han et al., 2012, p.308).

En casos comunes la similitud de los objetos se define en base a su cercanía en el espacio, es decir, la distancia entre objetos cumple un rol fundamental para el agrupamiento. La calidad de un grupo puede representarse por el diámetro, la distancia máxima entre dos objetos en el grupo (Han et al., 2012, p.108).

Todos los objetos son agrupados de acuerdo a todas las variables, es por esta razón, que al existir una variable irrelevante puede provocar ruido dentro de los resultados obtenidos (Hernández, 2016, p.3).

Algoritmo de Clustering

Pagola et al., (2015) indican que "Los algoritmos de Clustering son una herramienta eficaz para extraer información de datos en bruto. Son métodos de aprendizaje no supervisado. El objetivo de los algoritmos de clustering es dividir los datos en clústeres o grupos" (p.1).

Algoritmo de Clustering basado en densidad

Los algoritmos basados en densidad establecen grupos como zonas densas de puntos, separadas por otras zonas densas. La agrupación espacial estima el parecido de acuerdo con las características espaciales de los datos, por lo que no significa la semejanza entre dos objetos, sino la proximidad en el espacio de los dos objetos (Dib Ashur et al., 2016, p.18).

Estos algoritmos identifican los grupos que son densos y aquellos que son poco densos. Hay variedad de algoritmos basados en densidad, sin embargo, uno de los más destacados es DBSCAN. En la presente investigación se hace uso del algoritmo Dyclee que se basa en dos etapas y una de ellas es basada en densidad, la cual se detallará a mayor profundidad más adelante.

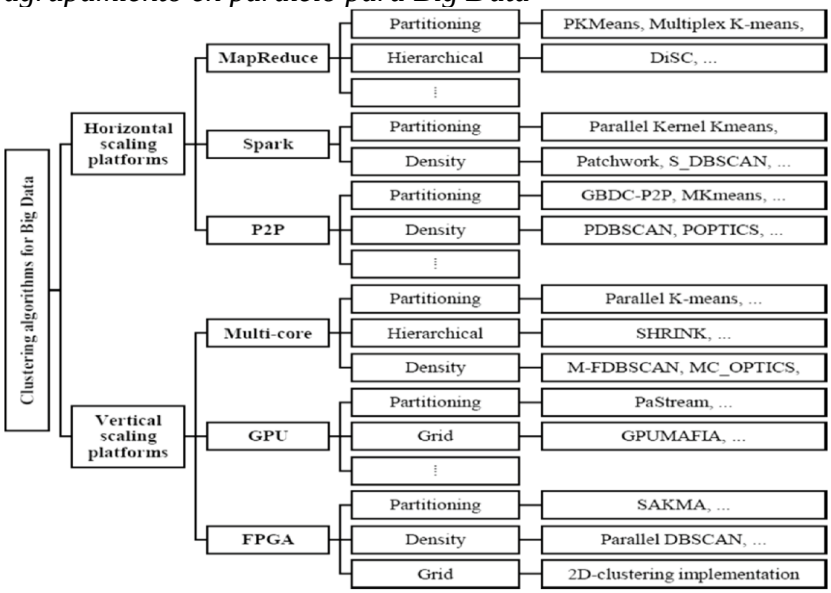
Algoritmos de Clustering en paralelo

Existen una variedad de tipo de tipos de agrupamiento, como los basados en particiones, en densidad, métodos bioinspirados, entre otros. Los algoritmos de agrupamiento paralelo basados en plataformas de escalado horizontal pueden ser MapReduce, Spark y redes Peer-to-Peer. Entre los algoritmos que utilizan MapReduce podemos nombrar a un k-means optimizado, un Multiplex k-means, MR-DBSCAN, MR-ABC, MBSC, un algoritmo de MapReduce utilizando spectral clustering, IB theory-based hierarchy clustering, BoW. Entre

los algoritmos que utilizan Spark podemos nombrar el Trimmed Kernel k-means, parallel k-means, Parallel Overlapping k-means, SRSIO-FCM, SLFCM, Patchwork, GraphX, S_DBSCAN, RDD-DBSCAN. Entre los algoritmos que utilizan redes peer-to-peer podemos nombrar GBDC-P2P, CYCLON, DIDIC, WaveCluster, p-PIC, MPI-DBSCAN. Los algoritmos de agrupamiento paralelo basados en plataformas de escalado vertical pueden ser GPU, CPU multi-nodo y plataformas FPGA. Entre los algoritmos que utilizan CPU multi-nodo podemos nombrar a una McKmeans y M-FDBSCAN. Entre los algoritmos que utilizan GPU podemos nombrar PaStream, Async-EM, POPTICS, G-DBSCAN, CUDA-MCL, COD-CAST-GPU. Entre los algoritmos que usan plataformas FPGA podemos nombrar FPGA-DBSCAN y FPGA-kmeans (Dafir et al., 2020).

Figura 3

Algoritmos de agrupamiento en paralelo para Big Data



Nota: Cuadro sinóptico de los diferentes tipos de algoritmos de agrupamiento para Big Data y en que marco de trabajo son utilizados. Tomado de (Dafir et al., 2020).

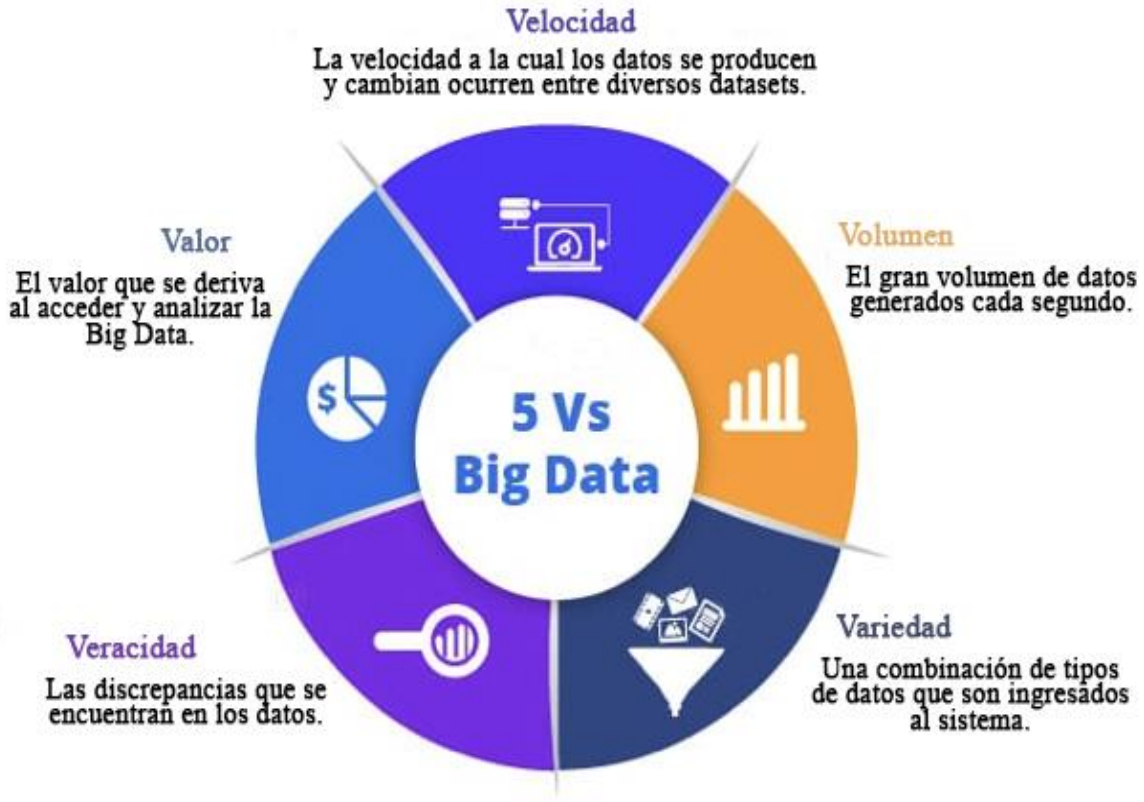
Big Data

Sowmya y Suneetha, (2017) definen Big Data como la "información disponible de miles de millones, incluso billones de registros generados por millones de personas y almacenados en innumerables fuentes en todo el universo cibernético" (p.246).

Este creciente conjunto de datos comprende formatos variados, es decir; datos estructurados, no estructurados y semiestructurados. Big Data es de naturaleza compleja, por lo tanto, exige el uso de poderosas tecnologías y algoritmos avanzados para su tratamiento (Oussous et al., 2018, p.433).

Figura 4

Las 5 V´s del Big Data



Nota: Gráfico que representa de manera resumida la características fundamentales de Big Data. Tomado de (Farnos, 2018).

Dyclee

Según Barbosa Roa et al., (2019)"Dyclee es un algoritmo basado en distancia y densidad que presenta varias propiedades, como el manejo de agrupaciones no convexas y de densidad múltiple con rechazo de valores atípicos, y logra ser completamente dinámico" (p.163).

Dyclee logra incorporar todas las características antes mencionadas en un mismo algoritmo y esto contribuye en gran medida al estado del arte (Barbosa Roa et al., 2019, p.163).

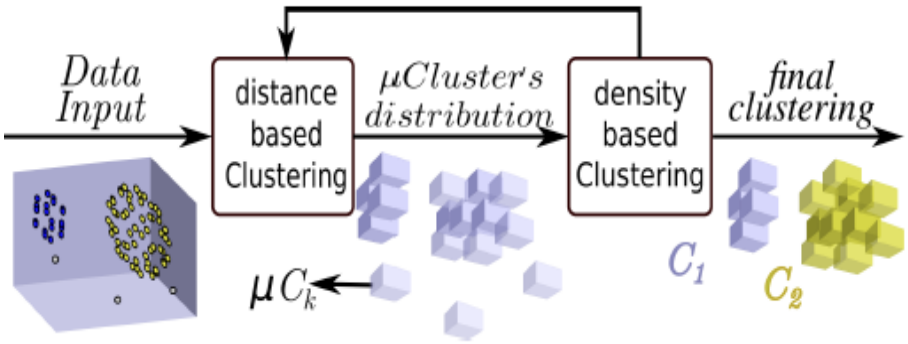
Dyclee fue desarrollado en base al paradigma de aprendizaje no supervisado de manera incremental, es decir, simulando la forma en que aprenden los seres humanos (sobre la marcha). Dyclee es un algoritmo de agrupación dinámico que estudia entornos en evolución, es decir, cuando trabaja sobre un grupo de datos, este no asume una estructura de datos predefinida, sino que la busca de forma progresiva a medida que se ingresan dichos datos cambiando la estructura del agrupamiento (Barbosa Roa et al., 2019, p.164).

La primera etapa de Dyclee opera la tasa del flujo de datos y crea micro clústeres juntando muestras de datos que se encuentran cercanos según la norma L1, también llamada distancia Manhattan. La segunda etapa de Dyclee funciona en base a repeticiones más bajas y analiza la división de los micro clústeres. La densidad de un micro clúster puede ser baja, media o alta y se utiliza para crear los clústeres finales con un enfoque basado en densidad (Barbosa Roa et al., 2019, p.164).

Un cluster es una agrupación de micro clústeres conectados donde cada micro clúster interno presenta una alta densidad y un micro clúster externo muestra una densidad media o alta. Las dos etapas trabajan en paralelo e intercambian información con la finalidad de mejorar su rendimiento mutuamente. Esto permite al algoritmo Dyclee localizar los clústeres finales en entornos en evolución con alto espacio dimensional y buena capacidad en el manejo de datos atípicos o que generan ruido (Barbosa Roa et al., 2019, p.164).

Figura 5

Principio de Dyclee



Nota: Se observa el flujo de trabajo que sigue el principio de Dyclee original donde en primera instancia se ingresa un conjunto de datos que posteriormente pasan a una etapa basada en distancia y como resultado de esta primera etapa se crean micro grupos; estos micro grupos pasan a la etapa basada en densidad y esta determina que grupo se convierten en densos o poco densos. Tomado de (Barbosa Roa et al., 2019).

Python

Python es un lenguaje de programación muy utilizado en el desarrollo de aplicaciones web, desarrollo de software, machine learning (ML) y la ciencia de datos. Los desarrolladores utilizan Python porque es competente y fácil de aprender, además, se puede ejecutar en distintas plataformas. Python se puede descargar sin costo alguno y se incorpora de forma efectiva en cualquier tipo de sistema (Amazon Web Service, 2022).

Marzal Varó et al., (2009) presentan a continuación, una serie de ventajas acerca de Python que generan mayor interés tanto a nivel profesional como para el aprendizaje de la programación:

- Los programas desarrollados en Python son muy compactos, es decir, suelen ser más cortos si se lo compara con lenguaje de programación como C. Por esta razón, Python es considerado un lenguaje de muy alto nivel.
- La sintaxis de Python permite entender los programas de una forma más facil debido a su escritura simple, en comparación con otros lenguajes de programación.

- Python facilita un entorno interactivo donde se pueden efectuar pruebas y ayuda a aclarar dudas con respecto a las características del lenguaje.
- El entorno de ejecución de Python permite la detección de muchos errores de programación y ofrece información abundante para detectarlos y corregirlos.
- Python puede usarse bajo el paradigma imperativo procedimental o orientado a objetos.
- Tiene a disposición variedad de estructuras de datos que se pueden manipular de forma sencilla. (p.16-17)

Paralelismo en Python

El procesamiento en paralelo es una forma de implementar tareas que se ejecutan al mismo tiempo en diferentes procesadores del mismo ordenador. El objetivo principal de esta forma de trabajo es disminuir el tiempo de espera al momento de procesar varias tareas (programmerclick, 2022). Python pone a disposición diferentes librerías o herramientas para lograr paralelismo, cada una con sus características propias, ventajas y limitaciones.

Hay dos maneras principales de emplear programas paralelos, una de ellas es mediante memoria compartida que es donde las subunidades pueden comunicarse entre sí haciendo uso del mismo espacio de memoria. Otra forma de usar programas en paralelo es a través de memoria distribuida, en este caso cada proceso está completamente separado y hace uso de su propio espacio de memoria (Acervo Lima, 2022).

Python utiliza hilos para lograr paralelismo mediante el manejo de memoria compartida. Consiste en subtareas que se crean en un proceso y a su vez comparten memoria. Sin embargo, Python en su diseño posee un mecanismo denominado Global Interpreter Lock (GIL) que solo permite que se ejecute un proceso a la vez independientemente del número de núcleos que tenga el computador y por esta razón no aumenta el rendimiento de los programas que usen hilos (Acervo Lima, 2022).

Las restricciones de GIL pueden superarse mediante el uso de procesos, sin embargo, hay que tener en consideración que la comunicación entre procesos es menos eficiente en este caso y es necesario hacer uso de técnicas que permitan una comunicación más efectiva (Acervo Lima, 2022).

Paralelismo basado en procesos

Según indica Python Software Foundation, (2021) en su documentación oficial, "multiprocessing es un paquete que permite crear procesos utilizando una API similar al módulo threading. El paquete multiprocessing ofrece concurrencia tanto local como remota, esquivando el Global Interpreter Lock mediante el uso de procesos en lugar de hilos (threads)".

El módulo multiprocessing permite aprovechar al máximo los procesadores con los que cuenta una determinada computadora y es compatible en sistema operativo Windows y Unix (Python Software Foundation, 2021).

Para crear un proceso en Python, haciendo uso del módulo multiprocessing, es necesario invocar a la clase Process, crear un objeto tipo Process y llamar a su método start(), además, es muy importante que los procesos sean declarado después de la siguiente cláusula: if __name__ == '__main__' (Python Software Foundation, 2021).

Además, hay que tener en consideración la comunicación entre procesos, el módulo multiprocessing admite dos tipos de canales para la comunicación entre procesos; uno de ellos es mediante el uso de colas (queues) para pasar mensajes de ida y vuelta, y el otro es implementando tuberías (Pipes) para la comunicación entre dos procesos (Python Software Foundation, 2021).

Amazon Web Services Elastic Map Reduce Hadoop Distribution

Amazon Elastic MapReduce es parte de Amazon Web Services (AWS), y existe desde que se inició Hadoop. AWS puede analizar datos con una interfaz fácil de utilizar y bien

distribuida que está construida en base al diseño de la estructura HDFS. Es uno de los vendedores mejor posicionados en el mundo. EMR es una herramienta para el procesamiento y análisis de Big Data, este servicio es de poca configuración y reemplazable como alternativa a correr los clusters localmente (Tanveer, 2020).

Amazon EMR está basado en Apache Hadoop, y procesa Big Data a través de cluster de servidores virtuales creados en Amazon Elastic Compute Cloud (EC2) y en Amazon Simple Storage Service (S3). El término elástico se refiere a la habilidad de cambiar su tamaño dinámicamente lo cual permite aumentar o disminuir el uso de recursos dependiendo de la demanda en cualquier momento (Tanveer, 2020).

Procesamiento de Big Data con Amazon EMR

Amazon EMR se utiliza para el análisis de datos en análisis de registros, almacenamiento de datos, análisis financiero, aprendizaje automático, indexación web, simulaciones científicas, bioinformática y más. EMR además soporta las cargas de trabajo basadas en Apache Spark, Presto y Apache Hbase, esta última se integra con Hive y Pig para obtener funciones adicionales (Tanveer, 2020).

AWS S3

Amazon S3 es un servicio de almacenamiento de objetos que ofrece adaptabilidad, accesibilidad a los datos, seguridad y productividad. Clientes de todos los sectores pueden proteger y almacenar sus datos para cualquier finalidad, como repositorios de datos, las aplicaciones en la nube y aplicaciones móviles. Con la utilidad de los modos de almacenamiento y las funciones de administración simples de utilizar, permite optimizar los precios, estructurar los datos y conformar controles de entrada detallados para cumplir con las condiciones empresariales, organizacionales y de aprobación específicos(Amazon Web Services, 2017).

AWS EMR

Amazon EMR es una plataforma de Big Data que actualmente lidera el mercado de los proveedores de servicios en la nube, con características como el procesamiento de grandes cantidades de datos de manera rápida y a una escala que sea eficiente en cuanto al costo usando herramientas como Apache Spark, Apache Hive, Apache Hbase, Apache Flink, Apache Hudi y Presto, con la capacidad de escalado automático de Amazon EC2 y la escalabilidad de almacenamiento de Amazon S3, EMR tiene la flexibilidad para correr clusters de poca duración y se puede escalar automáticamente para cumplir con la demanda, o para clusters de larga duración y de alta disponibilidad (Tanveer, 2020).

Uno de los mayores problemas con una aplicación en Big Data es afinar el programa, frecuentemente se puede dificultar el afinamiento de un programa de tal manera que todos los recursos asignados se usen de una manera eficiente. Debido a este factor, el tiempo que toma completar la tarea se incrementa gradualmente (Tanveer, 2020).

Elastic Map Reduce es un framework que administra todas las características necesarias en el procesamiento de Big Data de la manera más eficiente, rápida, segura y económica posible. Desde la creación del cluster hasta la distribución de los datos en varias instancias, todos estos aspectos son manejados fácilmente en Amazon EMR. Los servicios que ofrece son bajo demanda, es decir, que solo se va a utilizar basado en los datos (Tanveer, 2020).

Según Tanveer, (2020) los clusters son el componente central de la arquitectura EMR, son una colección de instancias EC2 llamadas nodos. Cada nodo tiene un rol específico que se denomina tipo de nodo y son de 3 diferentes tipos:

- **Master Node:** Es el responsable de administrar el cluster, ejecuta los componentes y distribuye los datos a través de los nodos para ser procesados, también supervisa que todo esté apropiadamente administrado y ejecutándose de manera correcta, y actúa en caso de fallos.

- **Core Node:** Tiene la responsabilidad de ejecutar los trabajos y guardar la información en HDFS en el clúster. Todas las partes de procesamiento son manejadas por el Core Node y los datos después de este procesamiento se guarda en la ubicación HDFS deseada.
- **Task Node:** Es opcional y tiene el único trabajo de ejecutar los trabajos que guarda los datos en HDFS.

Revisiones sistemáticas

Los investigadores hacen uso de la revisión sistemática para verificar y encontrar libros, artículos o cualquier documento literario destacado que contribuya al estudio de una determinada temática; empleando criterios de inclusión y exclusión establecidos con antelación. Esta metodología reduce el sesgo en la identificación, selección y resumen de los estudios, esto permite obtener mayor confiabilidad de la literatura consultada para el estudio del investigador (Quispe et al., 2021).

El presente trabajo de investigación realiza una descripción y análisis de la información científica en base al procesamiento en paralelo de un algoritmo de agrupamiento dinámico de trayectorias GPS.

Para realizar la búsqueda de la información se planteó una pregunta para centrarse específicamente en el área de interés del presente estudio. La pregunta se describe a continuación:

Pregunta: ¿Qué herramientas o metodologías se deben seguir para migrar un algoritmo de clustering a un ambiente que permita procesamiento en paralelo de grandes volúmenes de datos de trayectorias GPS?

La búsqueda a realizar se apoya en gran medida de la plataforma búsqueda Google Académico debido a que su motor de búsqueda proporciona resultados instantáneos referentes al tema de interés del investigador y de forma eficiente; en esta plataforma pueden encontrarse

libros, revistas o artículos científicos de gran relevancia. Además, hace uso de otros medios de búsqueda propios de terceros para realizar búsquedas a profundidad de un tema específico.

También se hace uso de la base de datos Scopus para realizar la búsqueda de literatura pertinente que contribuya a la realización del presente estudio. Para ello se creó el siguiente query necesario para la búsqueda dentro de Scopus que incluye las palabras claves TITLE-ABS-

KEY (clustering AND algorithm AND trajectories AND vehicular OR parallel AND pr ocessing) AND (LIMIT-TO (PUBYEAR , 2022) OR LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018)) AND (EXCLUDE (LANGUAGE , "Chinese")) AND (LIM IT-TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA , "MATH")).

Los criterios de búsqueda de inclusión utilizados para realizar la revisión sistemática se detallan a continuación:

- La búsqueda de la información debe incluir publicaciones de los últimos 5 años. Se pueden incluir publicaciones de mayor antigüedad siempre y cuando aporten conocimiento relevante sobre el presente estudio.
- La búsqueda de la información consultada debe estar orientada a áreas temáticas que se encuentren relacionadas con la ingeniería y ciencias de la computación.
- La búsqueda debe estar relacionada a temas de procesamiento de grandes volúmenes de datos y técnicas de procesamiento en paralelo.

Los criterios de búsqueda de exclusión que fueron considerados para el análisis y selección de los diferentes artículos científicos, libros u otro tipo de publicación científica son los siguientes:

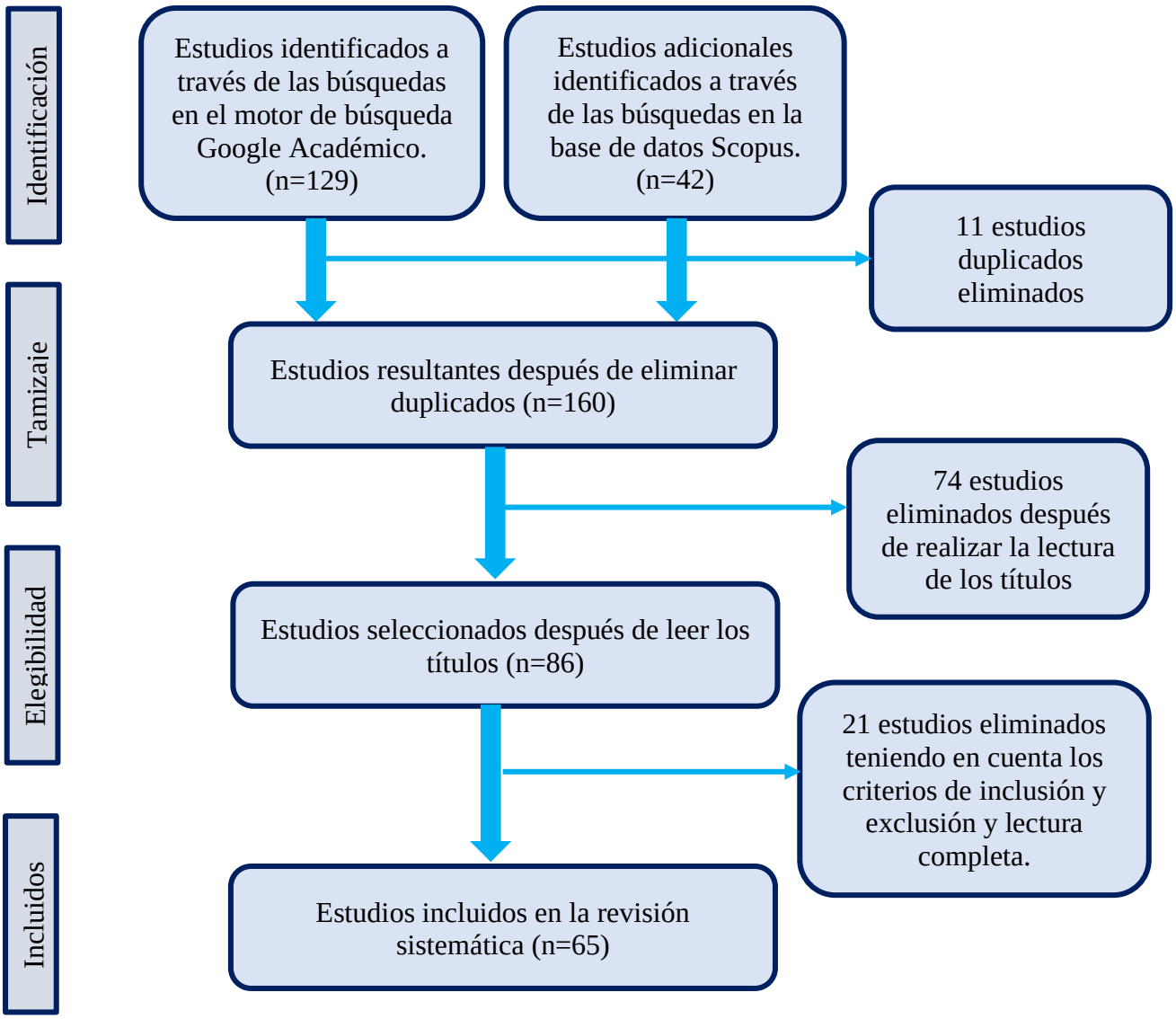
- Estudios que estén redactados en un idioma diferente al español o el inglés.

- Publicaciones escasamente relacionadas al tema principal del presente estudio.

A continuación, se presenta el diagrama “prisma” que indica de forma simplificada el proceso de selección de las diferentes publicaciones buscadas dentro de las bases de datos científicas o motores de búsqueda especificados anteriormente.

Figura 6

Diagrama Prisma del proceso de selección del presente estudio



Nota: Prisma: Proceso de selección, exclusión e inclusión y cantidad de documentos considerados para el presente estudio. Elaborado por Gabriel Ligua y Luis Vivas.

Después de realizar un análisis exhaustivo de la base de datos consultada y hacer uso de la plataforma Google académico, se puede evidenciar que para el procesamiento de grandes volúmenes de datos se utilizan diferentes metodologías que van relacionadas con la inteligencia artificial, una de ellas es el aprendizaje automático o machine learning. Esta temática puede ser empleada a través de diferentes tipos; aprendizaje supervisado que requiere datos de entrenamiento y se va perfeccionando según las iteraciones que realice un determinado algoritmo, y el aprendizaje no supervisado que consiste en identificar ciertos patrones dentro del conjunto de datos para clasificarlos, un ejemplo muy popular que utiliza aprendizaje no supervisado son los algoritmos de clustering. A su vez, estos algoritmos se pueden categorizar en algoritmos de clustering basados en distancia y algoritmos de clustering basados en densidad. El presente estudio hace uso del algoritmo Dyclee que contiene ambas características, es decir, es un algoritmo que hace uso de una etapa basada en distancia y otra etapa basada en densidad, esta última presenta micro clústeres agrupados según las características que se estén estudiando.

Además, en temas relacionados al paralelismo, se puede evidenciar técnicas o herramientas utilizadas para el procesamiento de grandes volúmenes de datos de una forma más rápida, una de las más adecuadas es el procesamiento en paralelo, que consiste en el uso de diferentes procesos que trabajan de forma simultánea con la finalidad de disminuir el tiempo de procesamiento de la información. Para ello, es necesario el uso de lenguajes de programación que soporten el multiprocesamiento, infraestructuras en la nube que permitan crear un ambiente en paralelo, entre otros.

Investigaciones como la que se evidencia en el artículo: “Algoritmo paralelo para la obtención de predicados difusos” de la revista: “Revista Cubana de Ciencias Informáticas” del año 2017 menciona que se puede considerar la migración a una arquitectura en paralelo de

ciertos algoritmos que inicialmente funcionan en secuencia para mejorar la velocidad de procesamiento de base de datos muy grandes.

Meta-análisis

El meta-análisis consiste en un análisis estadístico de un conjunto de resultados de trabajos o estudios individuales con la finalidad de integrar los hallazgos obtenidos. El meta-análisis provee una síntesis de forma numérica de los resultados (Sánchez Meca, 2010).

La búsqueda y selección de los artículos, libros y demás tipos de documentos científicos, haciendo uso del motor de búsqueda de Google Académico, proporcionó un total de 129 estudios relacionados a la temática del presente trabajo de investigación. Después de emplear los criterios de inclusión y exclusión se eligieron 63 estudios cuyo contenido fue analizado en un 45%. En la tabla 3 se muestra el porcentaje de estudios científicos preseleccionado y seleccionados para dar soporte al presente trabajo de investigación.

Tabla 3

Artículos obtenidos y seleccionados mediante el motor de búsqueda de Google Académico

Motor de Búsqueda	Estudios preseleccionados	Porcentaje	Estudios seleccionados	Porcentaje
Búsqueda automática				
Google Académico	129	100%	63	49%
Total	129	100%	63	49%

Nota: En esta tabla se especifica los porcentajes correspondientes a los artículos preseleccionados y seleccionados de Google Académico para realizar la revisión sistemática. La elaboración de esta tabla es propia y la fuente surge a partir del trabajo de investigación.

La búsqueda de artículos con el uso de la base de datos Scopus muestra como resultado, en primera instancia, 42 artículos científicos con cierta relación al tema principal del presente estudio. Finalmente se seleccionaron 2 artículos después de aplicar los criterios de inclusión y exclusión, además, de descartar estudios con poca relación al tema principal del presente estudio. En la tabla 4 se muestra el porcentaje de estudios científicos preseleccionado y seleccionados con la finalidad utilizarlos en este trabajo de investigación.

Tabla 4

Artículos obtenidos y seleccionados mediante la base de datos Scopus

Base de datos	Estudios preseleccionados	Porcentaje	Estudios seleccionados	Porcentaje
Búsqueda automática				
Scopus	42	100%	2	5%
Total	42	100%	2	5%

Nota: En esta tabla se especifica los porcentajes correspondientes a los artículos preseleccionados y seleccionados de la base de datos Scopus para realizar la revisión sistemática. La elaboración de esta tabla es propia y la fuente surge a partir del trabajo de investigación.

Hipótesis / Preguntas científicas a contestarse

¿Cómo contribuye la codificación en paralelo de un algoritmo de agrupamiento dinámico de trayectorias GPS y su posterior implementación en un ambiente de procesamiento en paralelo en una plataforma en la nube en el procesamiento de grandes volúmenes de datos de trayectoria GPS?

Variables de la investigación

Variable Dependiente: El tiempo de procesamiento de datos de trayectorias GPS.

Variable Independiente: Esquema de procesamiento del algoritmo de agrupamiento dinámico Dyclee.

Definiciones conceptuales

Algoritmo

López, (2009) define a los algoritmos como "una herramienta que permite describir claramente un conjunto finito de instrucciones, ordenadas secuencialmente y libres de ambigüedad, que debe llevar a cabo un computador para lograr un resultado previsible".

Agrupamiento

Campos, (2009) dice que "El agrupamiento se puede considerar como la aproximación más utilizada en aprendizaje no supervisado. Su objetivo general es encontrar algún tipo de

estructura en una colección de datos sin etiquetar o sin clasificar, ya que en la mayoría de los casos no se dispone de este tipo de información".

Procesamiento en paralelo

K. J. Wang et al., (2007) establece que el procesamiento en paralelo "Es una forma eficaz de procesamiento de información que favorece la explotación de los sucesos concurrentes en el proceso de la computación".

Trayectorias GPS

"Las trayectorias GPS de un objeto en movimiento es un conjunto de posiciones muestreadas con una marca de tiempo y otra información de movimiento relacionada (por ejemplo, velocidad y dirección de movimiento)" (Yang et al., 2018).

Dataset

Un Dataset es una agrupación de datos que se encuentran tabulados en algún sistema de almacenamiento de datos estructurados. El término se refiere a una base de datos, que se puede enlazar con otras. Las variables del Dataset están representadas en columnas y en las filas se ubican los datos a procesar (Redacción KeepCoding, 2022).

Velocidad

El término velocidad hace referencia a la comparación entre el desplazamiento seguido por una partícula con respecto al intervalo de tiempo utilizado para cumplir con dicho desplazamiento. Tambien se asocia con la la razón de cambio de los diferentes sitios que ocupa una partícula hasta culminar su recorrido (Díaz y González, 2010).

Distancia

La distancia es la longitud que se origina después del desplazamiento de un cuerpo desde un punto (A) a un punto (B). Se mide por medio de unidades de longitud como el metro o el kilómetro.

Densidad

En este estudio, la densidad hace referencia a la concentración de puntos o elementos en un área específica. Esta característica determina si un área es densa o poco densa.

Procesamiento en la nube

Aguilar, (2013) menciona que "El NIST ha definido la computación en nube (cloud computing) como: Un modelo que permite el acceso ubicuo, adaptado y bajo demanda en red a un conjunto compartido de recursos de computación configurables compartidos (por ejemplo: redes, servidores, equipos de almacenamiento, aplicaciones y servicios) que pueden ser aprovisionados y liberados rápidamente con el mínimo esfuerzo de gestión o interacción con el proveedor del servicio".

Big data

Aguilar, (2013) además nos indica que el Big Data es el "crecimiento exponencial de la creación de grandes volúmenes de datos, origen o fuentes de datos y la necesidad de su captura, almacenamiento y análisis para conseguir el mayor beneficio para organizaciones y empresas junto con las oportunidades que ofrecen y los riesgos de su no adopción".

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

En este capítulo se detalla la metodología empleada para el desarrollo del presente estudio. Se procura cumplir en su totalidad con los objetivos planteados mediante la migración de un algoritmo de agrupamiento, que inicialmente funciona de forma secuencial, a una arquitectura en paralelo y a su vez pueda ser ejecutado en una plataforma de computación en la nube. Para la ejecución de este algoritmo será necesario utilizar un conjunto de datos de trayectorias GPS que se originan o han sido recolectados previamente en otro tipo de estudios.

Modalidad de la investigación

El presente trabajo tiene la finalidad de disminuir el tiempo de ejecución en el procesamiento de grandes volúmenes de datos de trayectorias GPS, empleando un algoritmo de clustering basado en dos etapas; etapa basada en distancia y etapa basada en densidad. Se usaron diferentes herramientas, entre ellas, los diferentes datasets en forma de archivo con extensión .csv (Comma Separated Values) o valores separados por comas. Además, se usó el lenguaje de programación Python en conjunto con el editor de código fuente Visual Studio Code para realizar la migración y posterior paralelización del algoritmo Dyclee inicialmente codificado en R. También, se empleó una plataforma conocida como Amazon Web Services (AWS) para realizar la ejecución del algoritmo de clustering de trayectorias GPS con el uso de un software de la plataforma de computación en la nube.

Inicialmente se definió el algoritmo de clustering con el que se trabajó para realizar el presente estudio, el nombre del algoritmo es Dyclee, además, se investigó acerca de su funcionamiento y demás características esenciales para comprender su comportamiento al momento de realizar las agrupaciones. A su vez, se investigó acerca de los lenguajes de programación que permiten usar de forma más adecuada multihilos, multisesión y programación en paralelo, además, de las librerías y funciones necesarias para realizar una programación en paralelo. Finalmente, se eligió el lenguaje de programación adecuado para realizar la migración del algoritmo Dyclee a una arquitectura en paralelo.

A continuación, Martínez Ruiz, (2012) menciona en que consiste la investigación documental o investigación bibliográfica:

> La investigación bibliográfica se realiza mediante el análisis de fuentes de información escritas, documentos de distinto carácter, libros (bibliográficas), revistas y periódicos (hemerográficas), electrónicas o fuentes primarias (cartas, oficios y expedientes) que se encuentran en ficheros públicos y privados e Internet; esto supone hacer uso de métodos de análisis documental para la indagación de datos. (p.87)

Tomando de referencia el concepto anterior, se puede inferir que la modalidad de la investigación es, en su mayoría, la bibliográfica debido a la abundante consulta de información en fuentes secundarias. Además, este estudio también presenta ciertas características de una investigación de campo, debido a que se hizo uso de un algoritmo para el procesamiento de datos de trayectorias GPS. Los datos de trayectorias GPS corresponden a dos ciudades, Roma y Guayaquil.

Tipo de investigación

En este estudio se utiliza un tipo de investigación no experimental, puesto que es adecuado para cumplir con los alcances que se plantearon inicialmente, además, no se realiza manipulación directa de las variables de estudio, simplemente se adapta el algoritmo para

trabajar bajo un contexto específico que es objeto de estudio en este proyecto. El diseño de esta investigación es transversal debido a que las experimentaciones se efectuarán una única vez en un periodo de tiempo específico.

A continuación, Agudelo et al., (2008) señala en que consiste la investigación no experimental:

> Los estudios no experimentales son aquellos que se llevan a cabo sin manipulación deliberada de las variables. Es decir, es un estudio donde no se cambia intencionalmente las variables independientes. Lo que se realiza en la investigación no experimental es visualizar los fenómenos en su contexto natural y luego analizarlos. En la investigación no experimental no se construyen situaciones, sino que se observan situaciones que ya existen, no creadas deliberadamente por el investigador. (p.39)

Diseño metodológico de la investigación

El presente trabajo de investigación comienza realizando una revisión exhaustiva de información acerca de los diferentes lenguajes de programación que soportan multihilos, multisesión y programación en paralelo, además, de comprender el funcionamiento del algoritmo de agrupamiento dinámico Dyclee. Este novedoso algoritmo de clustering dinámico presenta dos etapas de agrupamiento, una de ellas en base a la distancia y otra en base a densidad. A su vez, este algoritmo tendrá ciertas modificaciones para cumplir con los objetivos planteados inicialmente, entre ellos, migrarlo a un lenguaje que soporte programación en paralelo, codificarlo bajo una arquitectura en paralelo haciendo uso de varios procesos que realizan una tarea en específico, además, que permita realizar las agrupaciones en basa a rangos de velocidades, en lugar de latitud y longitud.

Python es el lenguaje de programación seleccionado para realizar la migración y posterior paralelización del algoritmo Dyclee. Se hizo uso de la librería “Multiprocessing” para la creación de procesos, además, de implementar colas (queues) para la comunicación entre

procesos, que es necesaria para obtener resultados favorables en el desarrollo del presente estudio.

También se investigó acerca de las diferentes plataformas de computación en la nube y se escogió Amazon Web Services (AWS) para la ejecución del algoritmo. Además, se hizo uso de los datasets de Roma y Guayaquil, datos que son originarios de otros estudios y que se encuentran disponibles en repositorios públicos, para efectuar los experimentos.

Metodología de investigación

El presente estudio trabajará bajo una metodología de investigación no experimental y se ha seleccionado un diseño de experimentación transversal. A continuación, se detalla la metodología empleada:

1. Revisión de la literatura

Se realizará una consulta en fuentes bibliográficas acerca los diferentes paradigmas de programación en paralelo, además de los lenguajes de programación que soportan esta característica. A su vez, se realiza una revisión a nivel teórico del algoritmo de agrupación dinámico para comprender su metodología de agrupamiento. Además, se consulta acerca de temas relacionados a trayectorias GPS para identificar qué elementos serán utilizados para procesar rangos de velocidades. Finalmente, se consulta acerca de las diferentes plataformas de computación en la nube que existen, las ventajas y características que ofrecen y decidir de forma acertada la más apropiada para el desarrollo del presente estudio.

2. Migración del algoritmo de agrupamiento dinámico Dyclee

Para efectuar la migración del algoritmo Dyclee, se utilizará el lenguaje de programación Python, además, del uso de Visual Studio Code como herramienta editora de código fuente. El código, inicialmente se encuentra codificado en lenguaje R, sin embargo, esta adaptación habría sido migrada anteriormente, por otros investigadores, desde Python. Por lo

cual, se optó por hacer uso de la versión del algoritmo Dyclee original y realizar las respectivas modificaciones para que pueda agrupar por rangos de velocidades y no por latitud y longitud.

3. Selección de datos de trayectorias GPS

El uso de datasets que contengan datos referentes a trayectorias GPS, es decir, latitud, longitud, marca de tiempo, velocidad del vehículo, este último siendo muy necesario puesto que se requiere para realizar la agrupación mediante rangos de velocidades. Los datasets seleccionados contienen datos recopilados de ciudades como Guayaquil y Roma, todos provenientes de repositorios públicos con toda la información pertinente para efectuar las experimentaciones.

4. Diseño e implementación de una arquitectura en paralelo del algoritmo Dyclee

Se debe identificar en que partes del algoritmo Dyclee es posible implementar una arquitectura en paralelo. Inicialmente, se identifica la posible implementación de tres procesos; el primero será destinado a la recepción de datos, el segundo realizará la agrupación de datos mediante las funciones propias del algoritmo Dyclee, finalmente, el último proceso mostrará los grupos de datos con sus respectivas velocidades promedios como etiquetas de los micro clusters.

Figura 7

Procesos de Dyclee en paralelo

```
def proceso_imagenes(q):
    h3_tiempos=[]
    tiempo_puro=[]
    ciclo_imagenes=1
    while ciclo_imagenes <cant_ciclos :
        if q.empty():
            time.sleep(.25)
            #print("\t\t\tesperando..")
            continue
```

```
def proceso_agrupacion(q_1,q_2):
    h2_tiempos=[]
    tiempo_puro=[]
    # global ESTADO_1
    # global ESTADO_2
    ciclo_agrupacion=1
        #print(ESTADO_1)
```

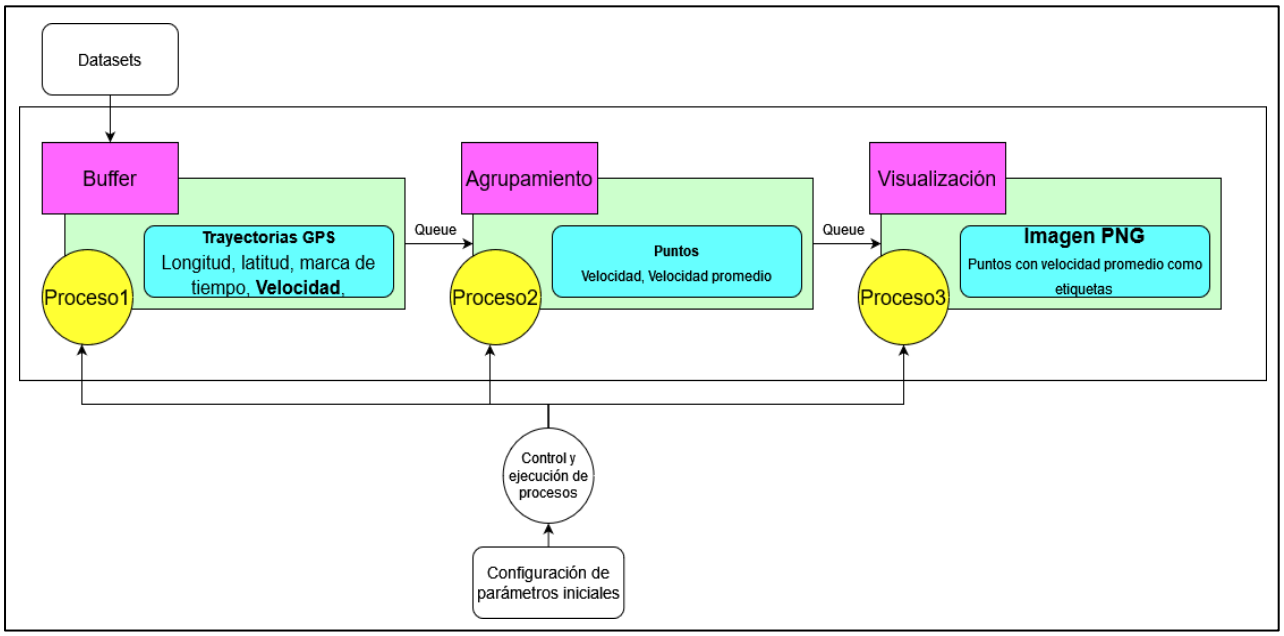
```
def proceso_buffer(q):
    h1_tiempos=[]

    global ESTADO_1
    #ciclo_buffer=1
    #while ciclo_buffer<cant_ciclos:
    for ciclo in range(1,cant_ciclos):
        h1_start=time.process_time()
        print("Ciclo "+ str(ciclo))
```

Elaborado por Gabriel Ligua y Luis Vivas.

Los grupos densos serán identificados con diferentes colores y los datos atípicos serán identificados con el color negro y con un signo menos (-) al inicio de la etiqueta del grupo. A continuación, en la figura 11 se muestra de forma gráfica el diseño propuesto para realizar la implementación en paralelo del algoritmo Dyclee.

Figura 8

Diseño para implementación de arquitectura en paralelo



Nota: En esta figura se muestra el diseño a seguir para realizar la implementación en paralelo del algoritmo Dyclee, se muestran tres procesos y cada uno con una tarea específica (buffer, agrupamiento de datos y visualización de datos mediante imágenes con extensión .png). Elaborado por Gabriel Ligua y Luis Vivas.

En cuanto a la implementación de una arquitectura en paralelo del algoritmo Dyclee, con Python, se hará uso del paquete "multiprocessing", específicamente la clase "Process".

Figura 9

Paquete multiprocessing

```
19    from multiprocessing import Process, Queue
```

Elaborado por Gabriel Ligua y Luis Vivas.

Además, es importante codificar la siguiente cláusula : if __name__ == '__main__' antes de declarar los procesos para evitar errores.

Figura 10

Uso de clausula if __name__ == '__main__'

```
if __name__=='__main__':
    main()

    ejecucion_final=time.perf_counter()
    print(f'Ejecución finalizada en {round(ejecucion_final,2)} segundos')
    archivo=open('archivoPG.txt','a')
    archivo.write("\n"+ strftime('%Y-%m-%d')+","+ strftime('%H:%M')+ ","+ str(round(ejecucion_final,2)))
    archivo.close()

    dta=pd.read_csv('archivoPG.txt', sep=",")
    dta.to_csv('TiemposPG.csv', index=None)
```

Elaborado por Gabriel Ligua y Luis Vivas.

Se llama a su método start( ), para crear los tres procesos antes mencionados. Hay que tener en consideración que los procesos necesitan comunicarse entre sí, debido a que requieren compartir información para que puedan cumplir con su tarea específica; para ello se hará uso de uno de los canales de comunicación que admite el módulo "multiprocessing", específicamente, mediante el uso de colas o "queues". Para implementar las colas o queues se necesita usar la clase "Queue" propia del paquete "multiprocessing".

Figura 11

Uso del método start() en el código

```
def main():
    processes = []
    q_1=Queue()
    q_2=Queue()

    hilo_1=Process(target=proceso_buffer, args=(q_1,))
    hilo_2=Process(target=proceso_agrupacion, args=(q_1,q_2,))
    hilo_3=Process(target=proceso_imagenes, args=(q_2,))


    hilo_1.start()
    hilo_2.start()
    hilo_3.start()

    processes.append(hilo_1)
    processes.append(hilo_2)
    processes.append(hilo_3)
```

Elaborado por Gabriel Ligua y Luis Vivas.

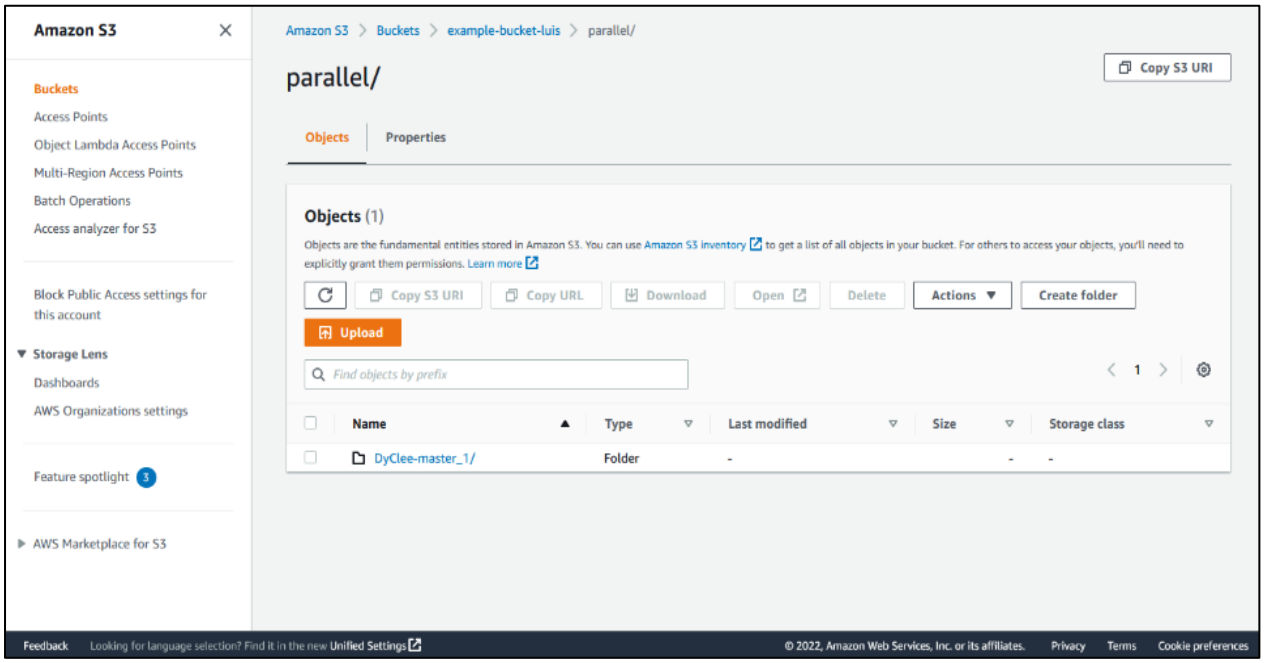
5. *Implementación de una plataforma de computación en la nube*

La implementación se realizará mediante la plataforma de computación en la nube Amazon Web Services, que mediante su servicio de EMR (Elastic Map Reduce), permite utilizar computadoras o nodos virtuales para el procesamiento de datos.

Para poder utilizar el servicio EMR es necesario utilizar el servicio llamado S3, que permite poder usar un disco duro virtual para guardar información en la nube de AWS, esto es necesario porque el servicio de EMR se comunica con S3 para poder procesar los datos, en este caso se ha creado un bucket que es donde va a guardar todos los archivos del algoritmo Dyclee.

Figura 12

Ubicación del algoritmo Dyclee en S3



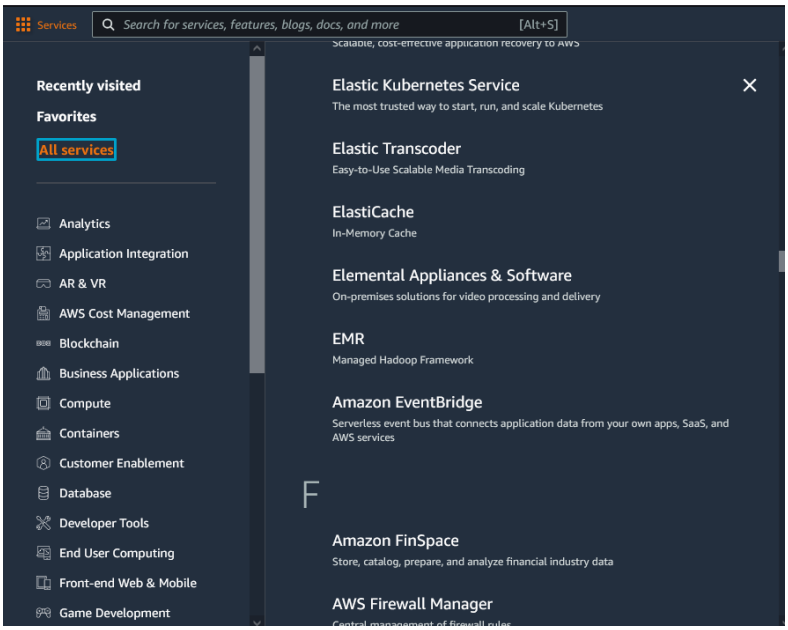
Fuente: Amazon Web Services.

Una vez creado el bucket en S3 se puede empezar a utilizar el servicio EMR y para poder realizar este proceso se siguieron los siguientes pasos:

- Solicitar a la universidad la creación un usuario en la plataforma AWS por medio del correo institucional y todos los permisos necesarios.
- En la lista de servicios escoger EMR.

Figura 13

Menú de la consola Amazon Web Services

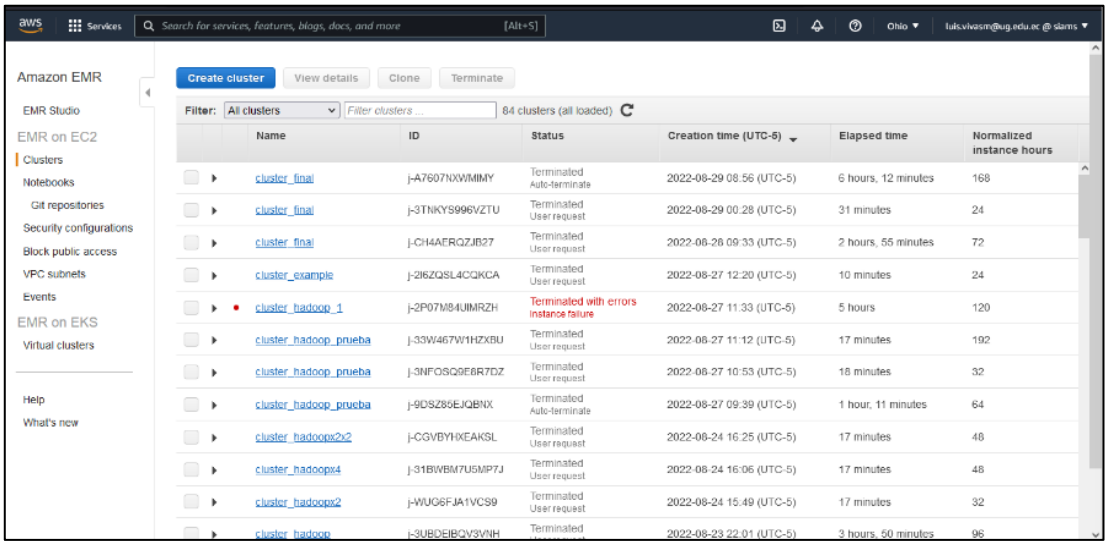


Fuente: Amazon Web Services.

- En la sección de clusters escoger la opción create cluster.

Figura 14

Menú de cluster en EMR

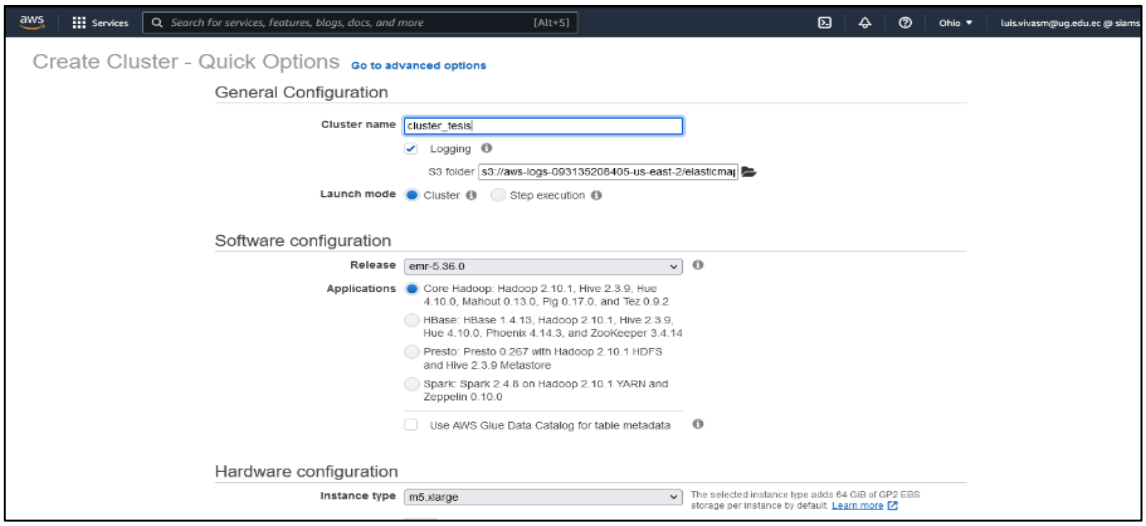


Fuente: Amazon Web Services.

- Agregar un nombre para el cluster, y dejar todas las demás opciones por defecto.

Figura 15

Opciones de creación de cluster en EMR

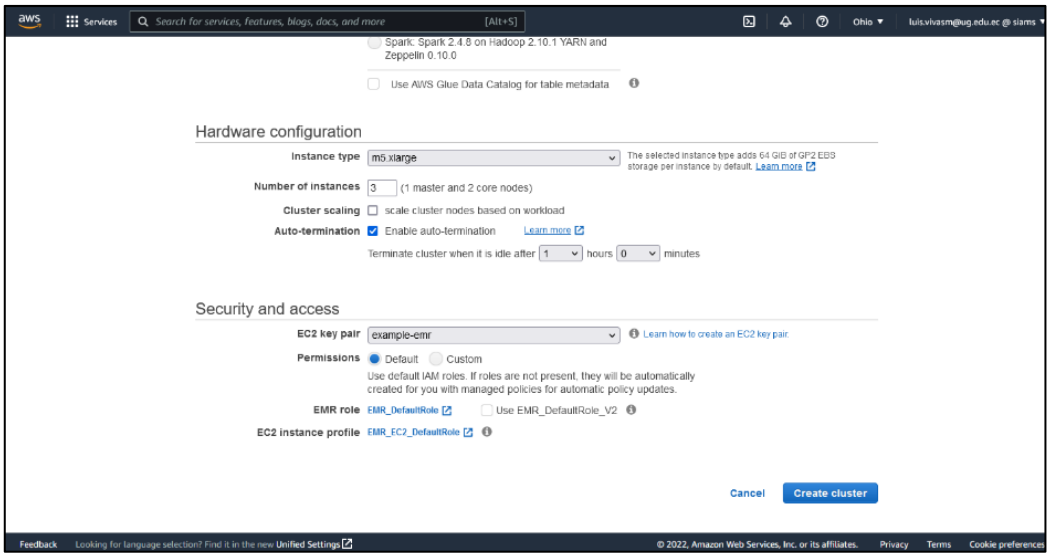


Fuente: Amazon Web Services.

- Elegir el EC2 Key pair creado previamente para poder usar el servicio EC2.

Figura 16

Opciones de creación de cluster en EMR (Elegir el EC2 Key pair)

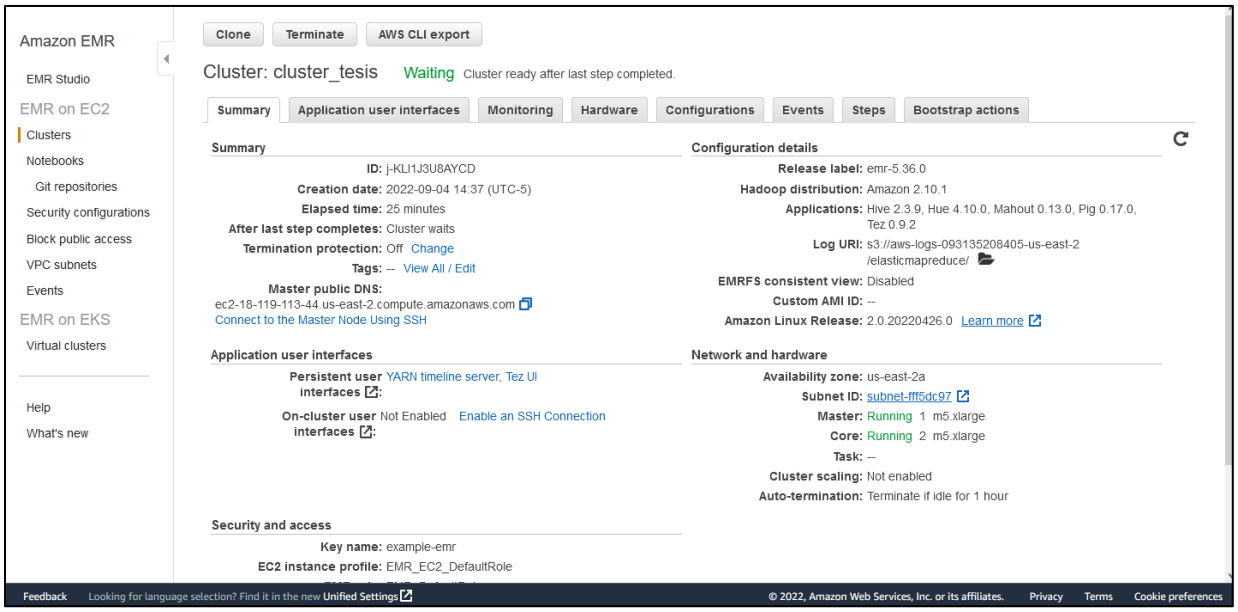


Fuente: Amazon Web Services.

- Elegir la opción create cluster
- Esperar a que se cree el cluster, el estado cambiará de starting a waiting.

Figura 17

Opciones del cluster



Fuente: Amazon Web Services.

6. Experimentación

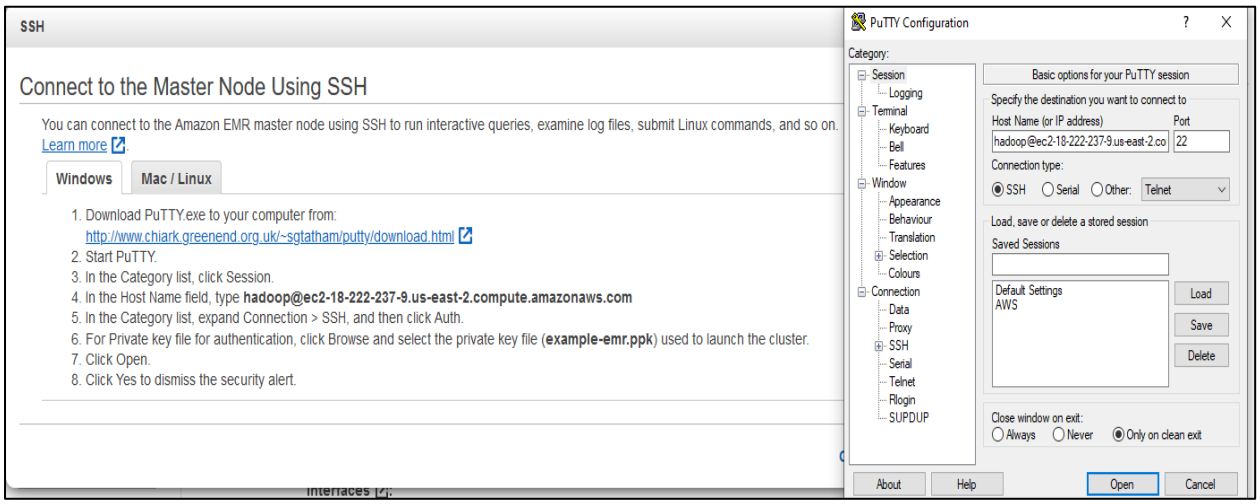
Se pondrá a prueba el algoritmo Dyclee, codificado de forma secuencial y bajo una arquitectura en paralelo, en dos contextos; de forma local en un computador y en una plataforma en la nube, específicamente, Amazon Web Services (AWS). Para ello, es necesario el uso de los datasets previamente seleccionados, que incluyen datos de trayectorias GPS (longitud, latitud y marca de tiempo) y velocidad. La ejecución del algoritmo permitirá obtener una serie de resultados relacionados con el tiempo que tarda el algoritmo en procesar los datos de los dos datasets, tanto de forma secuencial y en paralelo, además, permitirá obtener los grupos formados después del procesamiento.

Una vez creado el cluster en EMR se tuvo que realizar el siguiente proceso para realizar la experimentación de algoritmo Dyclee en la plataforma en la nube, siguiendo los pasos descritos a continuación:

- Conectarse con el cluster creado, para esto utilizamos una herramienta llamada putty, que crea un terminal virtual de Linux, en ella se ingresa la información para realizar la conexión segura SSH.

Figura 18

Opciones de conexión de putty

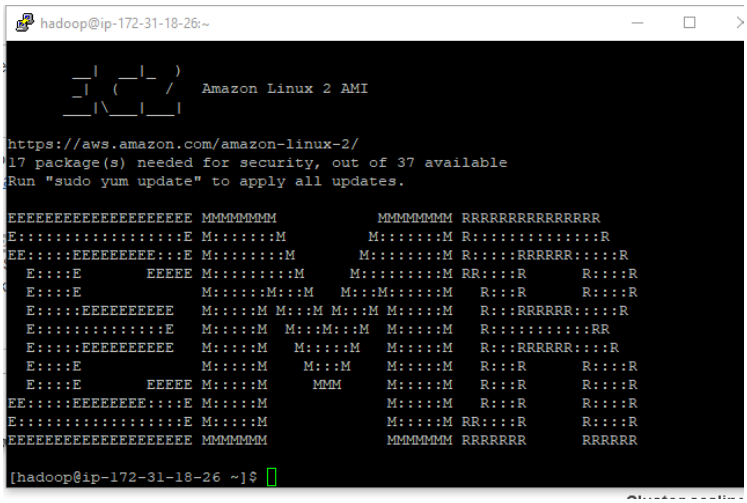


Fuente: Amazon Web Services.

- Cuando la conexión con el cluster se haya realizado correctamente aparecerá la siguiente pantalla.

Figura 19

Pantalla principal de EMR



Fuente: Amazon Web Services.

- Lo siguiente es instalar las respectivas librerías.

pip install termcolor==1.1.0

pip install matplotlib==3.1.1

pip install PyQt5==5.13.1

pip install numpy==1.17.3

pip install scikit_learn==0.21.3

pip install basemap-data

pip install basemap-data-hires

pip install basemap

pip install pandas

- A continuación, bajar el código dyclee alojado en S3 al cluster con el siguiente código: *aws s3 cp s3://example-bucket-luis/parallel/DyClee-master_1 . –recursive*
- Por último, se corre el algoritmo con el siguiente comando: *python main-Dyclee_GUAYAQUIL.py*

7. Análisis y validación de los resultados

Se utilizará la estadística descriptiva para el análisis de los resultados, además, se validará mediante juicio de expertos, la implementación en paralelo del algoritmo Dyclee teniendo en cuenta los tiempos de ejecución y los grupos que se forman en cada experimento.

Se ha seleccionado este tipo de metodología porque se ajusta a todas las fases que comprende este proyecto de investigación, esto con la finalidad de obtener los resultados que se esperan. Además, se ha seleccionado el diseño transversal puesto que las experimentaciones se realizarán en un periodo corto de tiempo y los datos tratados no necesitan una observación constante en tiempos distintos.

Población

Según Garcia Dihigo, (2016) “la población es el conjunto de elementos que tienen una característica común que es observable y acerca del cual queremos realizar determinados estudios” (p.130).

La población del presente estudio consistirá en un conjunto de datos de trayectorias GPS que serán utilizadas para realizar las diferentes experimentaciones. Estos datos se encuentran almacenados en dos datasets correspondientes a las ciudades de Guayaquil y Roma. En la tabla 5 se puede visualizar a detalle, la cantidad de registros que contiene cada dataset.

Tabla 5

Cantidad de registros de los datasets seleccionados

Población	Cantidad de registros
Guayaquil	30557
Roma	34118

Nota: En esta tabla se muestran a detalle la cantidad de registros que contienen los datasets seleccionados. La elaboración de esta tabla es propia y la fuente surge a partir del trabajo de investigación.

Análisis de los datasets seleccionados

Dataset de Guayaquil. El dataset de Guayaquil se origina de la recopilación de datos obtenidos con la ayuda de estudiantes universitarios que se movilizan en algún medio de transporte como moto, taxi, metro vía, buses urbanos, entre otros, utilizados en un proyecto de movilidad. Las ubicaciones de este dataset fueron recolectadas haciendo uso de artefactos móviles (smartphones) con un intervalo de tiempo promedio, entre dos ubicaciones o puntos consecutivos, de 5 segundos. Se trata de un conjunto de datos de trayectorias reducido puesto que se realizó un análisis donde se identificó que el horario con mayor cantidad de datos registrados fue entre las 16:30 hasta las 18:30. Después de realizar el filtro se consiguieron 30557 registros que constituyen 206 trayectorias de todo el dataset; el dataset original comprende 218 trayectorias. Este conjunto de datos fue recolectado el 28 de octubre de 2017

en la ciudad de Guayaquil, Ecuador. En la tabla 6 se detallan los campos del dataset con su descripción correspondiente.

Tabla 6

Campos del dataset de la ciudad de Guayaquil

Campo	Descripción
Longitud	Registra la longitud de la coordenada en formato double
Latitud	Registra la latitud de la coordenada en formato double
Velocidad	Registra la velocidad de la coordenada en formato double
Fecha	Registra la marca de tiempo de la coordenada en unixtime en formato double
Id	Registra el identificador único del vehículo

Nota: En esta tabla se muestra a detalle los campos que contiene el dataset de la ciudad de Guayaquil. La elaboración de esta tabla es propia y la fuente surge a partir del trabajo de investigación.

Dataset de Roma. El dataset de Roma tiene origen en un proyecto de movilidad Roma Taxi y está compuesto por 30 archivos diferentes que corresponden a 30 días (un archivo diario) en el que se han recolectado datos de trayectorias de taxis en la ciudad de Roma, Italia. El periodo de recolección de datos está comprendido entre las fechas del 2 de febrero del 2014 hasta el 2 de marzo del 2014. En este caso, se utilizan los registros correspondientes al día 12 de febrero del 2014 que comprende 708097 en total, sin embargo, se hace uso de una versión reducida de 34118 registros que corresponde a 2 horas del día antes mencionado. Estos datos fueron recolectados entre las 12:00 horas y 14:00 horas del 12 de febrero del 2014. A continuación, en la tabla 7 se detallan los campos del dataset con su descripción correspondiente.

Tabla 7

Campos del dataset de la ciudad de Roma

Campo	Descripción
Latitud	Registra la latitud de la coordenada en formato double
Longitud	Registra la longitud de la coordenada en formato double
Tiempo	Registra las horas, minutos y segundos en que fue tomada la coordenada GPS
Velocidad	Registra la velocidad en km/h a la próxima coordenada en formato double
Id	Registra el identificador único del taxi al que pertenece la coordenada

Nota: En esta tabla se muestra a detalle los campos que contiene el dataset de la ciudad de Roma. La elaboración de esta tabla es propia y la fuente surge a partir del trabajo de investigación.

Beneficiarios directos e indirectos del proyecto

Los beneficiarios del presente estudio se dividen como se muestra a continuación:

Beneficiarios directos

Los beneficiarios directos del proyecto son profesores y estudiantes que requieran un estudio como base o antecedente que contribuya a futuros trabajos de investigación relacionados con la implementación de una arquitectura en paralelo en algoritmos de clustering y su aplicación en plataformas de computación en la nube.

Además, investigadores y profesionales que se encuentran desarrollando estudios de mayor magnitud acerca de trayectorias GPS y requieren procesar grandes volúmenes de datos en un periodo de tiempo más corto.

Beneficiarios indirectos

Los beneficiarios indirectos del proyecto son investigadores que se dedican al análisis del flujo de tránsito vehicular con el fin de mejorar la toma de decisiones con respecto a las principales avenidas de una determinada ciudad. También, investigadores que se encuentran desarrollando proyectos con otro enfoque de estudio y requieran tomar una fuente de referencia para adaptarla según su conveniencia.

Entregables del proyecto

Artículo científico:

Se elaborará un artículo científico referente a la investigación realizada donde se detalle la problemática y los objetivos, la metodología empleada, los resultados que se obtuvieron de las experimentaciones y las conclusiones y recomendaciones como aporte final del estudio.

Propuesta de Investigación

Se realizarán tres tipos de experimentos. El primer experimento consistirá en la ejecución del algoritmo Dyclee de forma secuencial y en paralelo sobre los datasets de Roma y Guayaquil; esta experimentación se ejecutará de forma local en una computadora con las siguientes características: sistema operativo Windows 11 de 64 bits, procesador Intel® Core™ i7-6700k, RAM de 16.0 GB. La experimentación tendrá varias ejecuciones donde se estarán procesando datos en diferentes intervalos de tiempo, tal y como se detalla en la tabla 8. Finalmente se visualizarán los micro clusters formados en imágenes con extensión .png; esto después de realizar las agrupaciones en base a la velocidad de los vehículos. Además, se registran los tiempos de procesamiento totales para realizar la comparativa entre el algoritmo secuencial y en paralelo, para determinar si existe una mejoría en cuanto a tiempo de procesamiento del conjunto de datos.

El segundo experimento evaluará los resultados de los micro clusters que se formaron después de realizar la agrupación de los datos en base a rangos de velocidades con el algoritmo en secuencia y en paralelo, tanto para el dataset de Guayaquil como para el dataset de Roma. Se determinará, si existen variaciones considerables de los resultados analizando las velocidades promedios por intervalo de tiempo y velocidades promedios totales de todas las ejecuciones en secuencia y en paralelo, además, de verificar si la cantidad de grupos y ciclos varían cuando se modifica el intervalo de tiempo en cada ejecución.

El tercer experimento evaluará un indicador de tiempos promedios por ciclo del algoritmo Dyclee en secuencia y en paralelo para el dataset de Guayaquil y Roma. Se registrarán los tiempos por ciclo y se realizará un cálculo para obtener el tiempo promedio por ciclo; esto, para efectuar la comparativa entre el algoritmo secuencial y en paralelo; y determinar si existe una mejoría en cuanto a tiempos de procesamiento por ciclo del conjunto de datos.

Se realizará una experimentación adicional que consistirá en la ejecución del mismo algoritmo Dyclee de forma secuencial y en paralelo sobre los datasets de Roma y Guayaquil, sin embargo, en esta ocasión se realizará mediante el uso de Amazon Web Services que presenta las siguientes características: almacenamiento de 32GB, memoria de 16GB, 4 vCore. Esto con el fin de realizar una comparativa final y determinar si existen cambios favorables en los tiempos de la implementación en paralelo.

Tabla 8

Detalle de los tiempos de procesamiento de datos de las ejecuciones de los experimentos

Dataset	**Número de ejecución**	**Tipo de arquitectura**	
		Secuencial Intervalo de tiempo de procesamiento	**Paralelo Intervalo de tiempo de procesamiento**
Roma	1	5 minutos	5 minutos
	2	3 minutos	3 minutos
	3	2 minutos	2 minutos
	4	1 minuto	1 minuto
Guayaquil	1	5 minutos	5 minutos
	2	3 minutos	3 minutos
	3	2 minutos	2 minutos
	4	1 minuto	1 minuto

Nota: En esta tabla se detallan los intervalos de tiempo de procesamiento de datos de las diferentes ejecuciones que se realizaron tanto a nivel local como en la plataforma AWS con los datasets de la ciudad de Roma y Guayaquil . La elaboración de esta tabla es propia y la fuente surge a partir del trabajo de investigación.

Criterios de validación de la propuesta

Para la validación de la propuesta de investigación se realizó un juicio de expertos, que consistió en una encuesta con un formulario en línea. Esta encuesta se la realizó a tres

ingenieros expertos en el área de la minería de datos, trayectorias vehiculares y algoritmos de clustering. La encuesta está compuesta por ocho preguntas de escala Likert, con la finalidad de medir el nivel de veracidad de los resultados obtenidos en este proyecto de investigación.

El proceso de selección de expertos consistió, en primer lugar, en elaborar un listado inicial de personas posibles que pueden cumplir con las condiciones adecuadas para ser los expertos que evalúen el presente estudio. A estas personas, se les otorgó una valoración inicial en una escala creciente del 1 al 10, sobre su nivel de experiencia que poseen sobre temas relacionados al presente estudio. A continuación, se calcula el coeficiente de conocimiento o información (Kc), que consiste en la división entre número seleccionado anteriormente en la escala y el número mayor de la escala. Luego, se calcula el coeficiente de argumentación (Ka), para ello es necesario basarse en una tabla donde se encuentren descritas las fuentes de argumentación o fundamentación y seleccionar (alto, medio o bajo) según la experiencia y trabajos realizados por el posible experto; y se determinan los aspectos de mayor influencia comparando con una tabla patrón que contiene los pesos de cada fuente. Una vez obtenido los pesos de cada aspecto, se suman y se obtendrá el valor del coeficiente de argumentación. Finalmente, se calcula el valor del coeficiente de competencia (K), que es el que determina que experto se toma en consideración para realizar el proceso de juicio de expertos del presente estudio, para ello se aplica la siguiente fórmula:

Fórmula del coeficiente de competencia

$$K = 0{,}5(Kc + Ka)$$

Nota: Se muestra la fórmula para el cálculo del coeficiente de competencia. K es el coeficiente de competencia, Kc es el coeficiente de conocimiento y Ka es el coeficiente de argumentación. Tomado de (Mendoza, 2012).

Resultados

En esta sección se detallan los resultados de los tres experimentos realizados de forma local sobre los dos datasets escogidos. El primer experimento abarca los resultados de los

tiempos totales de ejecución (secuencial y paralelo) del algoritmo Dyclee para los intervalos de tiempo de procesamiento de cinco minutos, tres minutos, dos minutos y un minuto. El segundo experimento abarca los resultados de las velocidades promedio de cada uno de los grupos que se generan tanto mediante el algoritmo Dyclee (secuencial y paralelo) para los intervalos de tiempo de procesamiento de cinco minutos, tres minutos, dos minutos y un minuto. El tercer experimento proporciona los resultados de los tiempos promedio por ciclo del algoritmo Dyclee en secuencia y en paralelo, para los intervalos de tiempo de procesamiento de cinco minutos, tres minutos, dos minutos y un minuto. Se realizó un experimento adicional que abarca el tiempo total de ejecución del algoritmo en la plataforma Amazon Web Services. Además, se detallan los resultados obtenidos del juicio de expertos.

Resultados de la primera experimentación

Tiempos totales de ejecución

Ejecución secuencial. Se realizó el experimento para obtener el tiempo total de ejecución del algoritmo Dyclee secuencial para cuatro diferentes intervalos de tiempo de procesamiento: cinco minutos, tres minutos, dos minutos y un minuto.

Ejecución en paralelo. Se realizó el experimento para obtener el tiempo total de ejecución del algoritmo Dyclee en paralelo para cuatro diferentes intervalos de tiempo de procesamiento: cinco minutos, tres minutos, dos minutos y un minuto.

Resultados de los tiempos totales de ejecución con dataset de Guayaquil

Los resultados del experimento para el dataset de Guayaquil se pueden apreciar a continuación en la tabla 9. En esta, se muestran diferentes columnas. La primera columna consiste en los intervalos de tiempo de procesamiento que utiliza el algoritmo para procesar los datos en minutos (cinco minutos, tres minutos, dos minutos, un minuto). Luego, en la siguiente columna, se muestra la cantidad de ciclos que genera el algoritmo en cada ejecución, es decir,

23 ciclos en cinco minutos, 39 ciclos en tres minutos, 58 ciclos en dos minutos y 115 ciclos en un minuto. A continuación, se muestran las columnas con los tiempos totales que tarda en ejecutarse el código de forma secuencial y en paralelo y cuyos resultados muestran que en el algoritmo Dyclee en paralelo, el tiempo total de ejecución en todos los casos son menores que los tiempos totales de la ejecución del algoritmo Dyclee secuencial, además, se muestran los tiempos promedios totales en cada caso; en secuencia muestra un tiempo promedio total de 105,08 segundos y en paralelo muestra un tiempo promedio total de 86,72 segundos. Por último, se muestran las desviaciones estándar de los tiempos totales de cada ejecución, en el caso secuencial la desviación estándar es 69,38 y en la implementación en paralelo es 59,03. No hay gran variación en el resultado de ambas, lo cual indica que los tiempos totales de ejecución se encuentran dentro de un rango adecuado de variación.

Tabla 9

Comparación de los resultados del tiempo total de ejecuciones del algoritmo Dyclee secuencial y paralelo para el dataset de Guayaquil para 4 diferentes intervalos de tiempo de procesamiento

Intervalo	Cantidad de ciclos	Tiempo total secuencial	Tiempo total paralelo
5	23	44,23	35,95
3	39	70,15	56,99
2	58	103,33	83,64
1	115	202,62	170,29
	Tiempo total promedio	**105,08**	**86,72**
	Desviación estándar	**69,38**	**59,03**

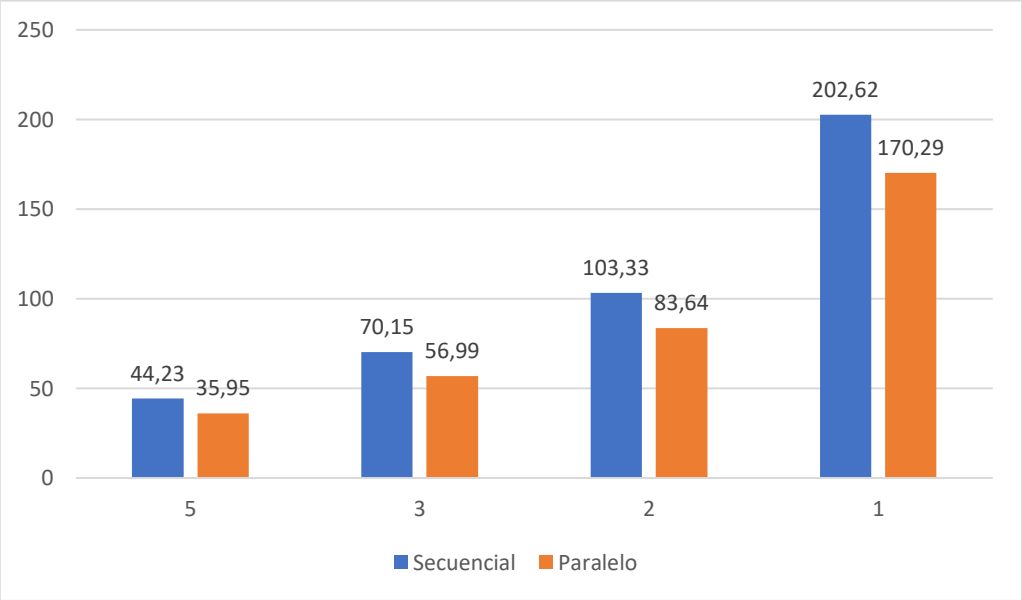
Nota: En la presente tabla se detalla la información más importante resumida de los resultados del tiempo total de ejecución del algoritmo Dyclee secuencial y paralelo para el dataset de Guayaquil para 4 intervalos de tiempo de procesamiento. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

En la figura 20 se puede observar un gráfico de barras que representa los resultados de los tiempos totales de las ejecuciones del algoritmo Dyclee tanto en secuencial como en paralelo para el dataset de Guayaquil, el eje vertical representa el tiempo total de ejecución en segundos mientras que en el eje horizontal se puede apreciar los diferentes intervalos de tiempo de procesamiento en minutos. Se puede visualizar como el resultado del tiempo total de cada

ejecución del algoritmo Dyclee secuencial, representadas en azul, demoran más segundos en comparación con las ejecuciones en paralelo representadas en naranja, lo cual indica que el algoritmo en paralelo es capaz de procesar los datos de formas rápida.

Figura 20

Comparación de tiempos totales de ejecución del algoritmo Dyclee secuencial y paralelo para dataset de Guayaquil



Nota: En esta figura se muestra los resultados obtenidos en cada ejecución con intervalo de tiempo específico, se visualiza que en todos los casos los tiempos de ejecución del algoritmo Dyclee en paralelo, para el dataset de Guayaquil, es menor que en secuencia . La elaboración es propia y la fuente corresponde a datos de la experimentación.

Resultados de los tiempos totales de ejecución con dataset de Roma

Los resultados del experimento para el dataset de Roma se pueden apreciar a continuación en la tabla 10. En esta, se muestran diferentes columnas. La primera columna consiste en los intervalos de tiempo de procesamiento que utiliza el algoritmo para procesar los datos en minutos (cinco minutos, tres minutos, dos minutos, un minuto). Luego, en la siguiente columna, se muestra la cantidad de ciclos que genera el algoritmo en cada ejecución, es decir, 24 ciclos en cinco minutos, 40 ciclos en tres minutos, 60 ciclos en dos minutos y 120 ciclos en un minuto. A continuación, se muestran las columnas con los tiempos totales que tarda en ejecutarse el código de forma secuencial y en paralelo y cuyos resultados muestran que en el

algoritmo Dyclee en paralelo, el tiempo total de ejecución en todos los casos son menores que los tiempos totales de la ejecución del algoritmo Dyclee secuencial, además, se muestran los tiempos promedios totales en cada caso; en secuencia muestra un tiempo promedio total de 87,71 segundos y en paralelo muestra un tiempo promedio total de 74,22 segundos. Por último, se muestran las desviaciones estándar de los tiempos totales de cada ejecución, en el caso secuencial la desviación estándar es 56,37 y en la implementación en paralelo es 48,07. No hay gran variación en el resultado de ambas, lo cual indica que los tiempos totales de ejecución se encuentran dentro de un rango adecuado de variación.

Tabla 10

Comparación de los resultados del tiempo total de ejecución del algoritmo Dyclee secuencial y paralelo para el dataset de Roma para 4 diferentes intervalos de tiempo de procesamiento

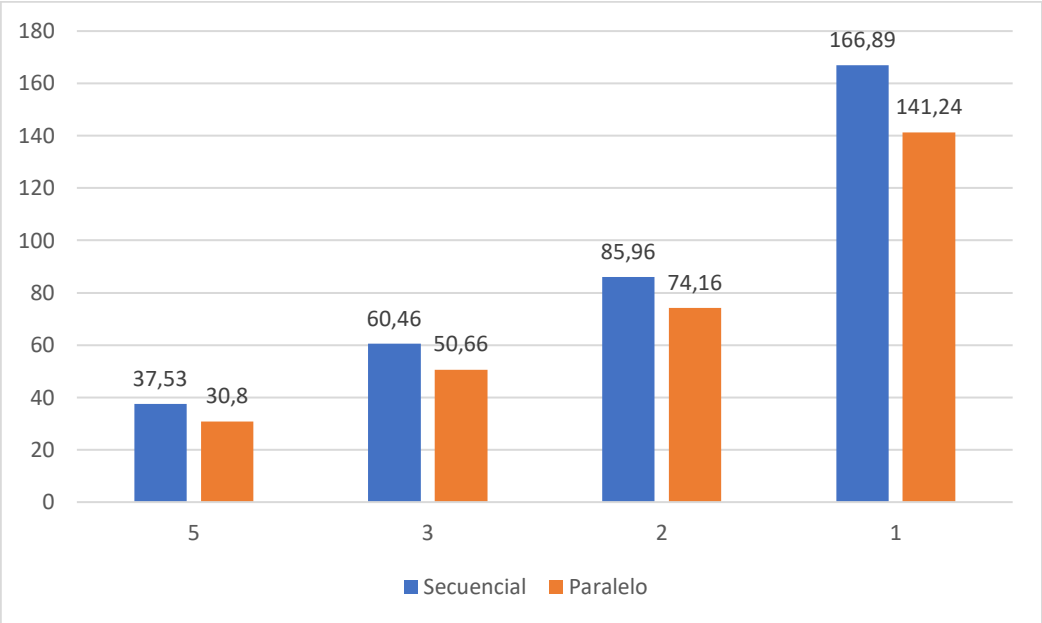
Intervalo	Cantidad de ciclos	Tiempo total secuencial	Tiempo total paralelo
5	24	37,53	30,8
3	40	60,46	50,66
2	60	85,96	74,16
1	120	166,89	141,24
	Tiempo total promedio	**87,71**	**74,22**
	Desviación estándar	**56,37**	**48,07**

Nota: En la presente tabla se detalla la información más importante resumida de los resultados del tiempo total de ejecución del algoritmo Dyclee secuencial y paralelo para el dataset de Roma para 4 intervalos de tiempo de procesamiento. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

En la figura 21 se puede observar un gráfico de barras que representa los resultados de los tiempos totales de las ejecuciones del algoritmo Dyclee tanto en secuencial como en paralelo para el dataset de Roma, el eje vertical representa el tiempo total de ejecución en segundos mientras que en el eje horizontal se puede apreciar los diferentes intervalos de tiempo de procesamiento en minutos. Se puede visualizar como el resultado del tiempo total de cada ejecución del algoritmo Dyclee secuencial, representadas en azul, demoran más segundos en comparación con las ejecuciones en paralelo representadas en naranja, lo cual indica que el algoritmo en paralelo es capaz de procesar los datos de formas rápida.

Figura 21

Comparación de tiempos totales promedios ejecución de Dyclee secuencial y paralelo para dataset de Roma



Nota: En esta figura se muestra los resultados obtenidos en cada ejecución con intervalo de tiempo específico, se visualiza que en todos los casos los tiempos de ejecución del algoritmo Dyclee en paralelo, para el dataset de Roma, es menor que en secuencia . La elaboración es propia y la fuente corresponde a datos de la experimentación.

Resultados de la segunda experimentación

Resultados de las velocidades de los grupos con dataset de Guayaquil

Los resultados del experimento para el dataset de Guayaquil se pueden apreciar a continuación en la tabla 11. Se muestran diferentes columnas. La primera columna consiste en los intervalos de tiempo de procesamiento que utiliza el algoritmo para procesar los datos en minutos (cinco minutos, tres minutos, dos minutos, un minuto). Las siguientes columnas corresponden a cada uno de los diferentes grupos y sus velocidades en Km/h, los grupos con números negativos representan grupos con valores atípicos y los grupos con valores positivos concentran la mayor cantidad de valores por lo que se denominan densos. Se puede apreciar que las velocidades de los grupos densos son menores a los de los grupos atípicos lo que denota una mayor concentración de valores cuando las velocidades disminuyen. La siguiente columna

muestra la velocidad promedio de todos los grupos en conjunto por intervalo de tiempo de procesamiento; en cinco minutos se muestra una velocidad promedio de 26,92 Km/h, en tres minutos se muestra una velocidad promedio de 27,195 Km/h, en dos minutos se muestra una velocidad promedio de 27,205 Km/h y finalmente, en un minuto se muestra una velocidad promedio de 27,218 Km/h. La última columna muestra el total de grupos generados por cada intervalo de tiempo de procesamiento, en todos los casos forma 4 grupos (2 atípicos y 2 densos). Además, se incluye la velocidad promedio por grupo; el grupo (-2) muestra una velocidad promedio de 36,277 Km/h, el grupo (-1) muestra una velocidad promedio de 48,587 Km/h, el grupo (1) muestra una velocidad promedio de 1,821 Km/h y el grupo (2) muestra una velocidad promedio de 21,852 Km/h. La desviación estándar analizada por grupo es de 0,277, lo cual indica que existe cohesión en las velocidades dentro de un mismo grupo, el valor de 20,106 para la desviación estándar por intervalo demuestra que las velocidades de cada grupo dentro de un mismo intervalo si sufre variaciones significativas.

Tabla 11

Velocidades promedio por intervalo de tiempo y por grupo generado del dataset de Guayaquil

Intervalo	Grupo -2	Grupo -1	Grupo 1	Grupo 2	Velocidad promedio por intervalo	Número de grupos
5	35,305	48,405	1,823	22,145	26,920	4
3	36,512	48,565	1,823	21,880	27,195	4
2	36,693	48,831	1,815	21,483	27,205	4
1	36,599	48,547	1,823	21,901	27,218	4
Velocidad promedio por grupo	**36,277**	**48,587**	**1,821**	**21,852**		
Promedio de desviación estándar total por grupo			**0,277**			
Promedio de desviación estándar total por intervalo			**20,106**			

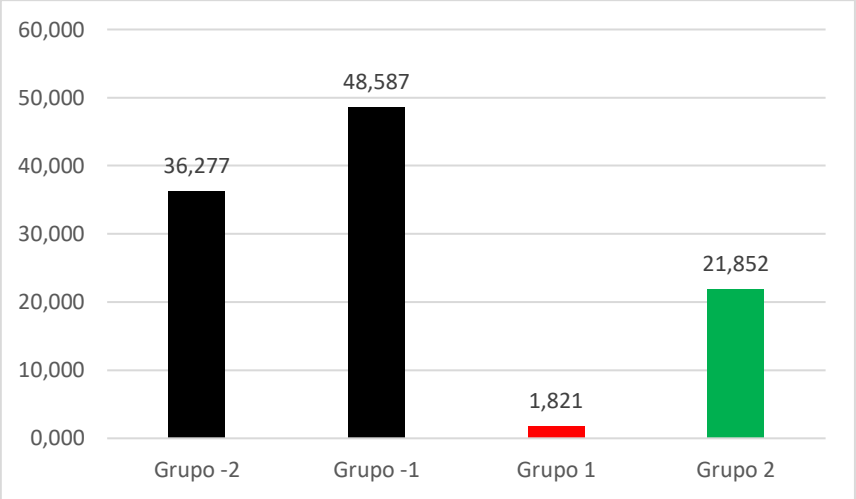
Nota: En la presente tabla se detalla la información más importante resumida de los resultados del algoritmo Dyclee por grupos y velocidades para el dataset de Guayaquil para 4 intervalos de tiempo de procesamiento. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

En la figura 22 se puede observar un gráfico de barras que representa la velocidad promedio en cada uno de los grupos generados por el algoritmo Dyclee con el dataset de

Guayaquil, se puede comprobar que los grupos 1 y 2 que son densos tienen velocidades menores a los grupos atípicos -1 y -2.

Figura 22

Velocidades promedios por grupo para el dataset de Guayaquil

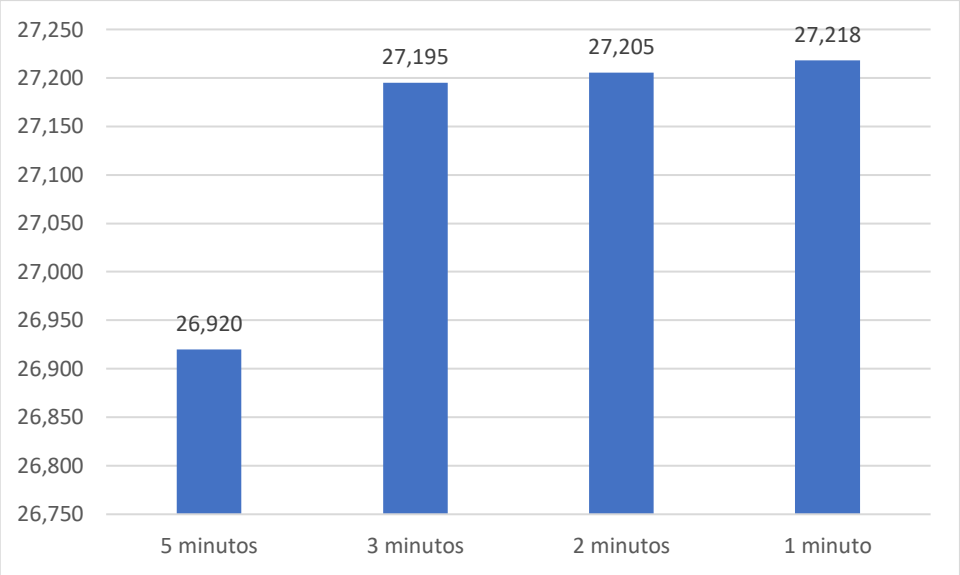


Nota: En esta figura se muestra los resultados de las velocidades promedios por grupo (4 grupos) del dataset de Guayaquil. La elaboración es propia y la fuente corresponde a datos de la experimentación.

En la figura 23 se puede observar un gráfico de barras que representa la velocidad promedio en cada uno de los intervalos de tiempo de procesamiento del algoritmo Dyclee con el dataset de Guayaquil, se puede comprobar que las velocidades, aunque son muy similares no son iguales para ningún caso siendo el tiempo promedio de velocidad para el intervalo de 1 minuto la mayor, lo que denota que mientras menor sea el intervalo de tiempo de procesamiento del algoritmo mayor será la velocidad promedio.

Figura 23

Velocidad promedio por intervalo de tiempo de procesamiento para el dataset de Guayaquil

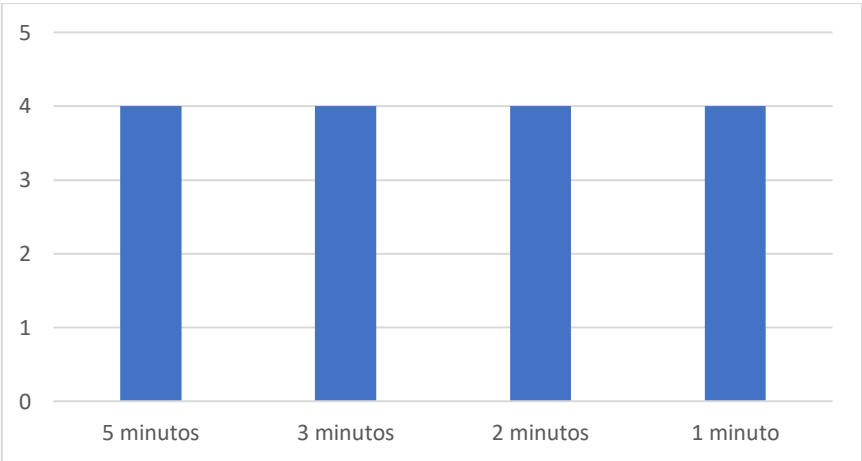


Nota: En esta figura se muestra los resultados de las velocidades promedios por intervalo de tiempo (5 minutos, 3 minutos, 2 minutos y 1 minuto) del dataset de Guayaquil. La elaboración es propia y la fuente corresponde a datos de la experimentación.

En la figura 24 se puede observar un gráfico de barras que muestra gráficamente la cantidad de grupos generados por el algoritmo Dyclee en cada intervalo de tiempo de procesamiento, en todos los casos la cantidad de grupos fue la misma.

Figura 24

Velocidad promedio por intervalo de tiempo de procesamiento para el dataset de Guayaquil



Nota: En esta figura se muestra la cantidad de agrupaciones generadas por intervalo de tiempo del dataset de Guayaquil. La elaboración es propia y la fuente corresponde a datos de la experimentación.

Ejecución secuencial. Se realizó el experimento con el dataset de Guayaquil para obtener las velocidades promedios por el total de grupos en todas las ejecuciones del algoritmo Dyclee secuencial, para 4 diferentes intervalos de tiempo de procesamiento: 5 minutos, 3 minutos, 2 minutos y 1 minuto.

Ejecución en paralelo. Se realizó el experimento con el dataset de Guayaquil para obtener las velocidades promedios por el total de grupos en todas las ejecuciones del algoritmo Dyclee paralelo, para 4 diferentes intervalos de tiempo de procesamiento: 5 minutos, 3 minutos, 2 minutos y 1 minuto.

A continuación, se muestra la comparativa de resultados del experimento con el Dataset de Guayaquil que se puede apreciar en la tabla 12. En la tabla, se muestran diferentes columnas. La primera columna muestra la cantidad de ciclos generados por el algoritmo Dyclee, tanto en secuencia como en paralelo en cada intervalo de tiempo; es decir, 23 ciclos en cinco minutos, 39 ciclos en tres minutos, 58 ciclos en dos minutos y 115 ciclos en un minuto. En la tercera columna se pueden apreciar la cantidad de grupos generados en cada intervalo de tiempo, se visualiza que la cantidad de grupos no varía en ningún caso. En la siguiente columna se puede ver la velocidad promedio para cada intervalo de tiempo en secuencia y un promedio total de todas las ejecuciones de Dyclee secuencial cuyo valor es 27,134 Km/h. En la última columna se puede apreciar el promedio de velocidad por intervalo de tiempo en paralelo y el promedio total de todas las ejecuciones de Dyclee en paralelo cuyo valor es 27,134 Km/h. Se puede notar que, en ambos casos, los promedios totales de velocidad es la misma, lo mismo ocurre con las velocidades promedio por intervalo de tiempo; esto indica que la arquitectura en secuencia o paralelo no afecta en los más mínimo en los resultados finales de las agrupaciones para este caso. Se muestra, que tanto en las ejecuciones en secuencia como en las ejecuciones en paralelo se muestran exactamente los mismo resultados de cantidad de ciclos, cantidad de grupos y velocidades promedio.

Tabla 12

Comparación de velocidades promedios por intervalo de tiempo y velocidades promedios totales de todas las ejecuciones en secuencia y en paralelo con el dataset de Guayaquil

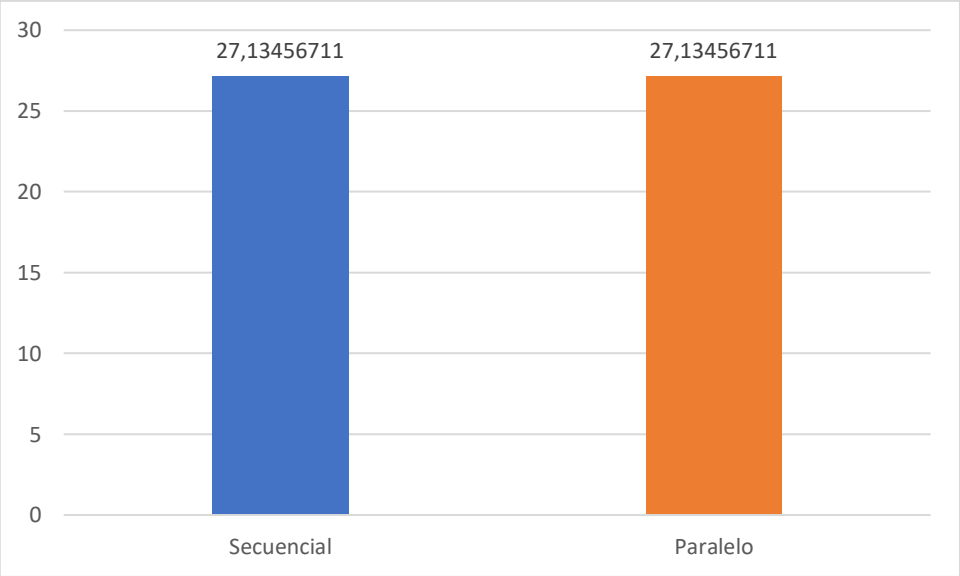
Ciclos	Intervalo	Cantidad de grupos	Velocidad promedio Secuencial	Velocidad promedio Paralelo
23	5	4	26,91973346	26,91973346
39	3	4	27,19514043	27,19514043
58	2	4	27,20545872	27,20545872
115	1	4	27,21793583	27,21793583
Total velocidad promedio			**27,13456711**	**27,13456711**

Nota: En la presente tabla se detalla la comparación de los resultados de las velocidades promedios de los grupos del dataset de Guayaquil haciendo uso del algoritmo Dyclee en secuencia y en paralelo. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

En la figura 25 se puede observar un gráfico de barras que representa la velocidad promedio total de ambas ejecuciones de Dyclee, tanto en secuencial como en paralelo, se puede notar como en ambos casos el resultado es el mismo, lo que quiere decir que las velocidades de los grupos generados por el algoritmo son independientes del tiempo total de ejecución.

Figura 25

Resultados de las velocidades promedios totales del algoritmo Dyclee secuencial y Dyclee en paralelo para dataset de Guayaquil

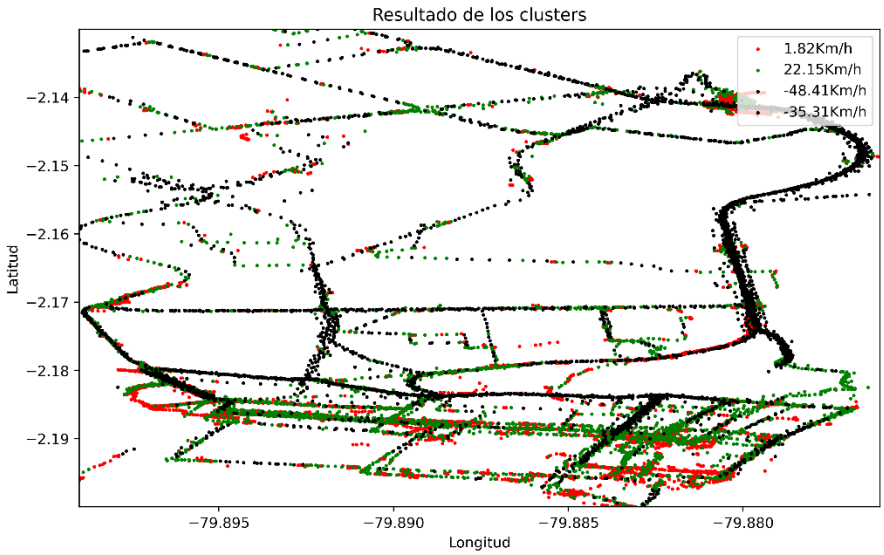


Nota: En esta figura se muestra la velocidad promedio total de todos los grupos y todas las ejecuciones para el algoritmo en secuencia y paralelo con el dataset de Guayaquil. La elaboración es propia y la fuente corresponde a datos de la experimentación.

En la figura 26 se puede apreciar el mapa de los grupos generados por el algoritmo Dyclee secuencial con un tiempo de 5 minutos y un total de 23 ciclos con el dataset de Guayaquil. La figura corresponde al último ciclo de la ejecución. En la figura 27 se puede apreciar el mapa de los grupos generados por el algoritmo Dyclee en paralelo con un tiempo de 5 minutos y un total de 23 ciclos con el dataset de Guayaquil. Esto con la finalidad de comparar los resultados de ambos mapas y comprobar que lo expuesto anteriormente es afirmativo, es decir, tanto en el algoritmo en secuencia como con el algoritmo en paralelo, lo grupos y velocidades promedios son las mismas.

Figura 26

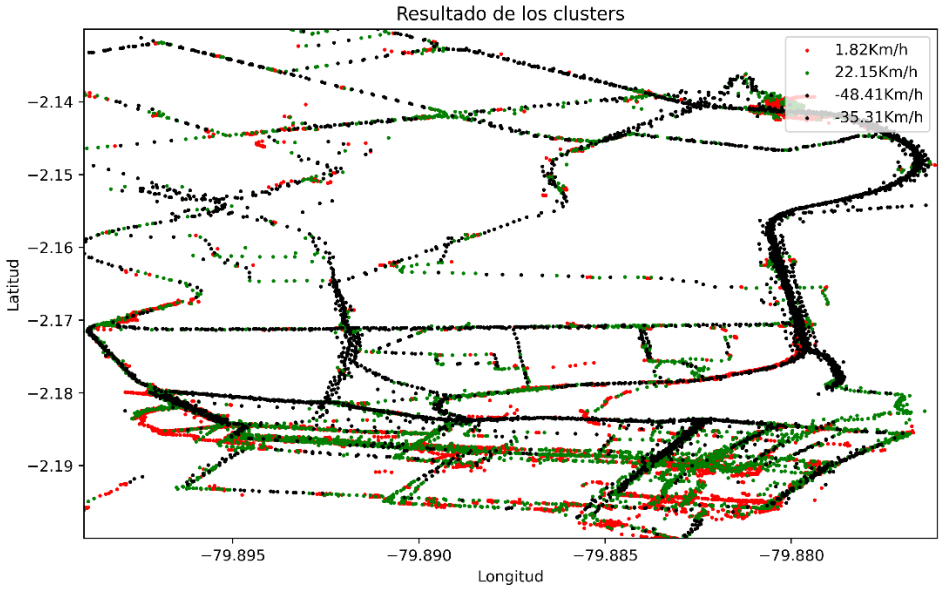
Mapa del ciclo 23 generado por el algoritmo Dyclee secuencial para dataset de Guayaquil en el intervalo de tiempo de procesamiento de 5 minutos



Nota: En esta figura se muestra el mapa generado por el algoritmo en secuencia. El mapa corresponde al ciclo 23 con intervalo de 5 minutos haciendo uso del dataset de Guayaquil, Se muestran las 4 agrupaciones por punto generadas. Los grupos densos corresponden a los que se encuentran identificados con colores rojo y verde, los grupos atípicos se identifican con el color negro y con un signo negativo al comienzo de cada etiqueta. Las etiquetas son las velocidades promedios de cada grupo formado puesto que el algoritmo agrupa en base a rangos de velocidades. La velocidad promedio del primer grupo denso (color rojo) corresponde a 1,82 Km/h. La velocidad promedio del segundo grupo denso (color verde) corresponde a 22,15 Km/h. La velocidad promedio del primer grupo atípico (color negro) corresponde a 48,41 Km/h. La velocidad promedio del segundo grupo atípico (color negro) corresponde a 35,31 Km/h. La elaboración es propia y la fuente corresponde a datos de la experimentación.

Figura 27

Mapa del ciclo 23 generado por el algoritmo Dyclee en paralelo para dataset de Guayaquil en el intervalo de tiempo de procesamiento de 5 minutos



Nota: En esta figura se muestra el mapa generado por el algoritmo en paralelo. El mapa corresponde al ciclo 23 con intervalo de 5 minutos haciendo uso del dataset de Guayaquil, Se muestran las 4 agrupaciones por punto generadas. Los grupos densos corresponden a los que se encuentran identificados con colores rojo y verde, los grupos atípicos se identifican con el color negro y con un signo negativo al comienzo de cada etiqueta. Las etiquetas son las velocidades promedios de cada grupo formado puesto que el algoritmo agrupa en base a rangos de velocidades. La velocidad promedio del primer grupo denso (color rojo) corresponde a 1,82 Km/h. La velocidad promedio del segundo grupo denso (color verde) corresponde a 22,15 Km/h. La velocidad promedio del primer grupo atípico (color negro) corresponde a 48,41 Km/h. La velocidad promedio del segundo grupo atípico (color negro) corresponde a 35,31 Km/h. La elaboración es propia y la fuente corresponde a datos de la experimentación.

Resultados de las velocidades de los grupos con dataset de Roma

Los resultados del experimento para el dataset de Roma se pueden apreciar a continuación en la tabla 13. Se muestran diferentes columnas. La primera columna consiste en los intervalos de tiempo de procesamiento que utiliza el algoritmo para procesar los datos en minutos (cinco minutos, tres minutos, dos minutos, un minuto). Las siguientes columnas corresponden a cada uno de los diferentes grupos y sus velocidades en Km/h, los grupos con números negativos representan grupos con valores atípicos y los grupos con valores positivos concentran la mayor cantidad de valores por lo que se denominan densos. Se puede apreciar

que las velocidades de los grupos densos son menores a los de los grupos atípicos lo que denota una mayor concentración de valores cuando las velocidades disminuyen. La siguiente columna muestra la velocidad promedio de todos los grupos en conjunto por intervalo de tiempo de procesamiento; en cinco minutos se muestra una velocidad promedio de 43,324 Km/h, en tres minutos se muestra una velocidad promedio de 43,318 Km/h, en dos minutos se muestra una velocidad promedio de 44,824 Km/h y finalmente, en un minuto se muestra una velocidad promedio de 40,755 Km/h. La última columna muestra el total de grupos generados por cada intervalo de tiempo de procesamiento. En los intervalos de tiempo de cinco, tres y dos minutos se formaron 7 grupos (3 grupos atípicos y 4 grupos densos). En el intervalo de un minuto se formaron 6 grupos (3 grupos atípicos y 3 grupos densos), esto provoca una ligera variación en los resultados de la velocidad promedio por intervalo de tiempo; debido a que se forman menos grupos en la ejecución de un minuto, los datos se redistribuyen de forma diferente en los grupos con respecto a las demás ejecuciones. Aunque se puede notar que esta redistribución afecta en menor medida a los grupos densos 1, 2 y 3 de todas las ejecuciones. Además, se incluye la velocidad promedio por grupo; el grupo (-3) muestra una velocidad promedio de 74,583 Km/h, el grupo (-2) muestra una velocidad promedio de 79,192 Km/h, el grupo (-1) muestra una velocidad promedio de 55,659 Km/h, el grupo (1) muestra una velocidad promedio de 0,945 Km/h, el grupo (2) muestra una velocidad promedio de 16,096 Km/h, el grupo (3) muestra una velocidad promedio de 30,657 Km/h y el grupo (4) muestra una velocidad promedio de 45,422 Km/h. La desviación estándar analizada por grupo es de 3,490, lo cual indica existe cohesión de en las velocidades dentro de un mismo grupo, el valor de 30,197 para la desviación estándar por intervalo demuestra que las velocidades de cada grupo dentro de un mismo intervalo si sufre variaciones significativas.

Tabla 13

Velocidades promedio por intervalo de tiempo y por grupo generado del dataset de Roma

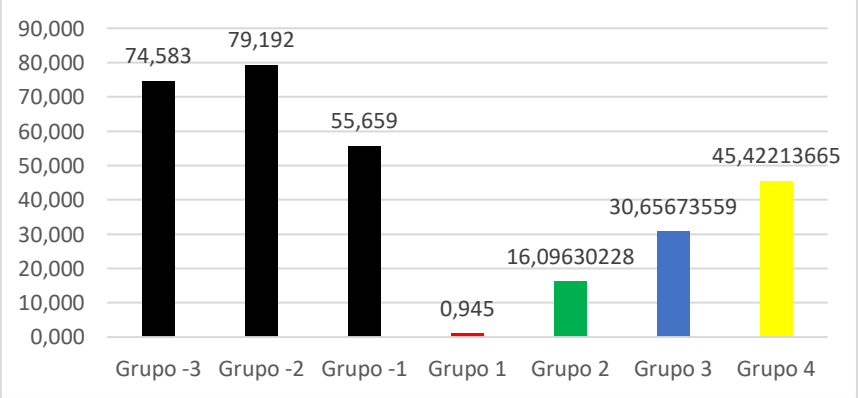
Intervalo	Grupo -3	Grupo -2	Grupo -1	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Velocidad promedio por intervalo	Total Grupos
5	70,667	84,000	57,383	0,937	15,909	29,987	44,385	43,324	7
3	70,667	84,000	55,947	0,937	15,944	30,343	45,387	43,318	7
2	73,000	84,000	62,424	0,938	16,062	30,852	46,494	44,824	7
1	84,000	64,767	46,882	0,968	16,470	31,444		40,755	6
Velocidad promedio por grupo	**74,583**	**79,192**	**55,659**	**0,945**	**16,096**	**30,657**	**45,422**		
Promedio de desviación estándar total por grupo			**3,490**						
Promedio de desviación estándar total por intervalo			**30,197**						

Nota: En la presente tabla se detalla la información más importante resumida de los resultados del algoritmo Dyclee por grupos y velocidades para el dataset de Roma para 4 intervalos de tiempo de procesamiento. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

En la figura 28 se puede observar un gráfico de barras que representa la velocidad promedio en cada uno de los grupos generados por el algoritmo Dyclee con el dataset de Roma, se puede comprobar que los grupos 1,2,3 y 4 que son densos tienen velocidades menores a los grupos atípicos -1,-2 y -3; aunque cabe destacar que el grupo atípico -2 tiene una mayor velocidad promedio que el grupo atípico -3, lo cual puede ser producto de la generación de un grupo menos en el intervalo de tiempo de agrupación de 1 minuto.

Figura 28

Velocidades promedios por grupo para el dataset de Roma

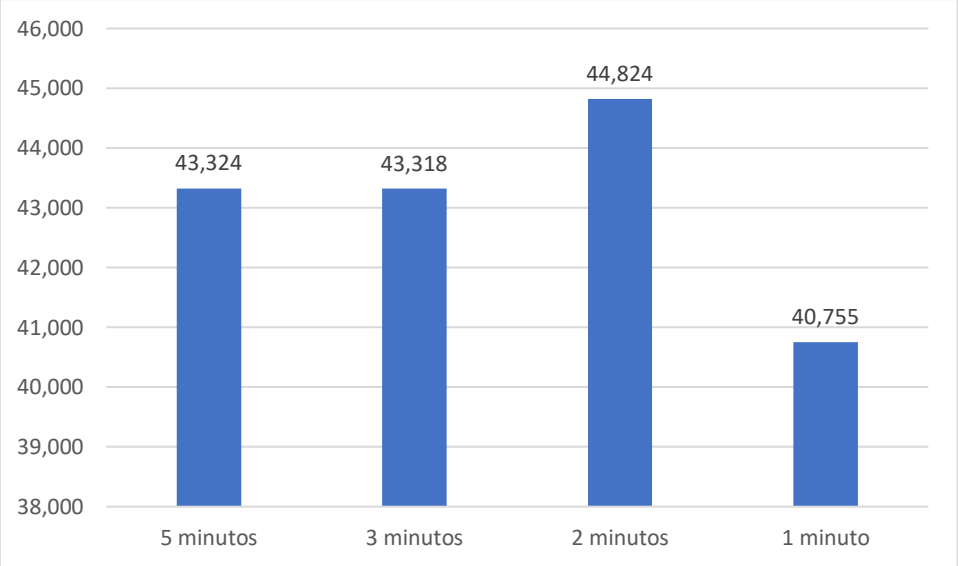


Nota: En esta figura se muestra los resultados de las velocidades promedios por grupo (7 grupos) del dataset de Roma. La elaboración es propia y la fuente corresponde a datos de la experimentación.

En la figura 29 se puede observar un gráfico de barras que representa la velocidad promedio en cada uno de los intervalos de tiempo de procesamiento del algoritmo Dyclee con el dataset de Roma, se puede comprobar que las velocidades, aunque son muy similares no son iguales para ningún caso. El tiempo promedio de velocidad para el intervalo de 2 minuto es la mayor, posiblemente por la diferencia de número de grupos en cada intervalo, por lo que se puede deducir en este caso que mientras menor sea el intervalo de tiempo de procesamiento del algoritmo, la velocidad promedio no será necesariamente menor.

Figura 29

Velocidad promedio por intervalo de tiempo de procesamiento para el dataset de Roma

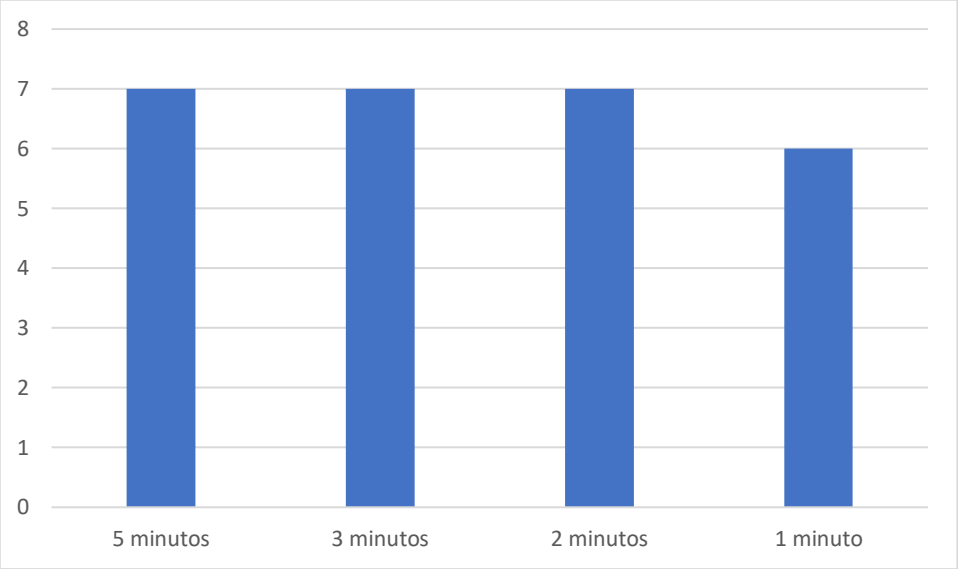


Nota: En esta figura se muestra los resultados de las velocidades promedios por intervalo de tiempo (5 minutos, 3 minutos, 2 minutos y 1 minuto) del dataset de Roma. La elaboración es propia y la fuente corresponde a datos de la experimentación.

En la figura 30 se puede observar un gráfico de barras que muestra la cantidad de grupos generados por el algoritmo Dyclee en cada intervalo de tiempo de procesamiento, en este caso se puede comprobar como el intervalo de tiempo de procesamiento de 1 minuto generó un grupo menos que todos los anteriores para el dataset de Roma.

Figura 30

Cantidad de grupos generados por intervalo de tiempo de procesamiento para dataset de Roma



Nota: En esta figura se muestra la cantidad de agrupaciones generadas por intervalo de tiempo del dataset de Roma. La elaboración es propia y la fuente corresponde a datos de la experimentación.

Ejecución secuencial. Se realizó el experimento con el dataset de Roma para obtener las velocidades promedios por el total de grupos en todas las ejecuciones del algoritmo Dyclee secuencial, para 4 diferentes intervalos de tiempo de procesamiento: 5 minutos, 3 minutos, 2 minutos y 1 minuto.

Ejecución en paralelo. Se realizó el experimento con el dataset de Roma para obtener las velocidades promedios por el total de grupos en todas las ejecuciones del algoritmo Dyclee paralelo, para 4 diferentes intervalos de tiempo de procesamiento: 5 minutos, 3 minutos, 2 minutos y 1 minuto.

A continuación, se muestra la comparativa de resultados del experimento con el Dataset de Roma que se puede apreciar en la tabla 14. En la tabla, se muestran diferentes columnas. La primera columna muestra la cantidad de ciclos generados por el algoritmo Dyclee, tanto en secuencia como en paralelo en cada intervalo de tiempo; es decir, 24 ciclos en cinco minutos, 40 ciclos en tres minutos, 60 ciclos en dos minutos y 120 ciclos en un minuto. En la tercera

columna se pueden apreciar la cantidad de grupos generados en cada intervalo de tiempo, se visualiza que la cantidad de grupos cambia unicamente en el intervalo de un minuto (6 grupos). En la siguiente columna se puede ver la velocidad promedio para cada intervalo de tiempo en secuencia y un promedio total de todas las ejecuciones de Dyclee secuencial cuyo valor es 43,055 Km/h. En la última columna se puede apreciar el promedio de velocidad por intervalo de tiempo en paralelo y el promedio total de todas las ejecuciones de Dyclee en paralelo cuyo valor es 43,055 Km/h. Se puede notar que, en ambos casos, los promedios totales de velocidad es la misma, lo mismo ocurre con las velocidades promedio por intervalo de tiempo; esto indica que la arquitectura en secuencia o paralelo no afecta en los más mínimo en los resultados finales de las agrupaciones para este caso. Se muestra, que tanto en las ejecuciones en secuencia como en las ejecuciones en paralelo se muestran exactamente los mismo resultados de cantidad de ciclos y velocidades promedio, a excepción de la cantidad de agrupaciones que varía unicamente en la ejecución de un minuto.

Tabla 14

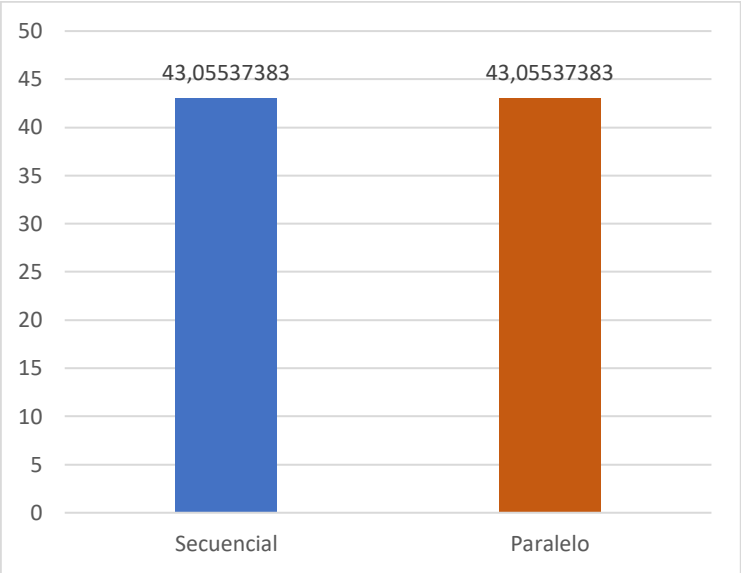
Comparación de velocidades promedios por intervalo de tiempo y velocidades promedios totales de todas las ejecuciones en secuencia y en paralelo

Ciclos	Intervalo	Cantidad de grupos	Velocidad promedio Secuencial	Velocidad promedio Paralelo
24	5	7	43,32407617	43,32407617
40	3	7	43,31791825	43,31791825
60	2	7	44,82439068	44,82439068
120	1	6	40,75511021	40,75511021
Total velocidad promedio			**43,05537383**	**43,05537383**

Nota: En la presente tabla se detalla la comparación de los resultados de las velocidades promedios de los grupos del dataset de Roma haciendo uso del algoritmo Dyclee en secuencia y en paralelo. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

Figura 31

Resultados de las velocidades promedios totales del algoritmo Dyclee secuencial y Dyclee en paralelo para dataset de Roma

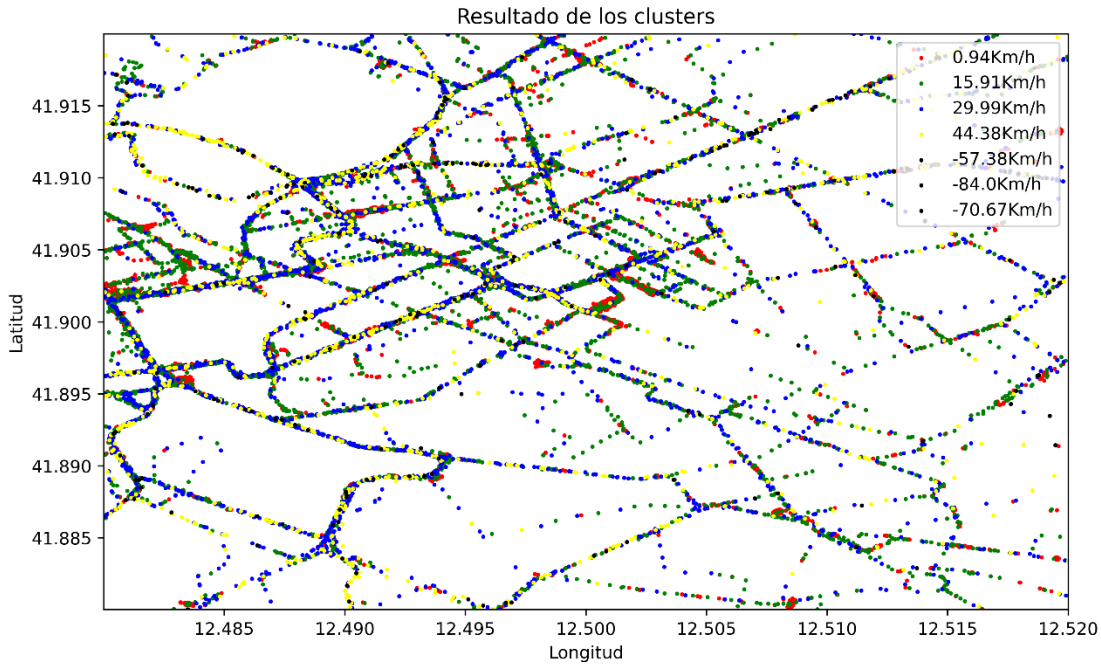


Nota: En esta figura se muestra la velocidad promedio total de todos los grupos y todas las ejecuciones para el algoritmo en secuencia y paralelo con el dataset de Roma. La elaboración es propia y la fuente corresponde a datos de la experimentación.

En la figura 32 se puede apreciar el mapa de los grupos generados por el algoritmo Dyclee secuencial con un tiempo de 5 minutos y un total de 24 ciclos con el dataset de Roma. La figura corresponde al último ciclo de la ejecución. En la figura 33 se puede apreciar el mapa de los grupos generados por el algoritmo Dyclee en paralelo con un tiempo de 5 minutos y un total de 24 ciclos con el dataset de Roma. Esto con la finalidad de comparar los resultados de ambos mapas y comprobar que lo expuesto anteriormente es afirmativo, es decir, tanto en el algoritmo en secuencia como con el algoritmo en paralelo, lo grupos y velocidades promedios son las mismas, exceptuando unicamente la ejecución de un minuto.

Figura 32

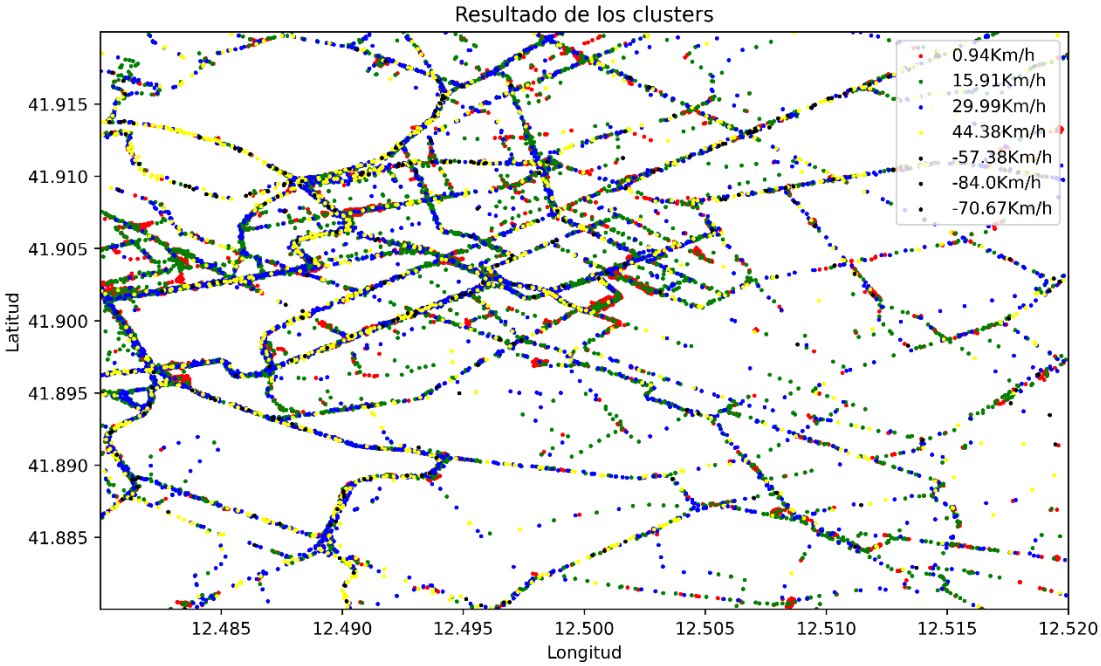
Mapa del ciclo 24 generado por el algoritmo Dyclee secuencial para dataset de Roma en el intervalo de tiempo de procesamiento de 5 minutos



Nota: En esta figura se muestra el mapa generado por el algoritmo en secuencia. El mapa corresponde al ciclo 24 con intervalo de 5 minutos haciendo uso del dataset de Roma, Se muestran las 7 agrupaciones por punto generadas. Los grupos densos corresponden a los que se encuentran identificados con colores rojo, verde, azul y amarillo; los grupos atípicos se identifican con el color negro y con un signo negativo al comienzo de cada etiqueta. Las etiquetas son las velocidades promedios de cada grupo formado puesto que el algoritmo agrupa en base a rangos de velocidades. La velocidad promedio del primer grupo denso (color rojo) corresponde a 0,94 Km/h. La velocidad promedio del segundo grupo denso (color verde) corresponde a 15,91 Km/h. La velocidad promedio del tercer grupo denso (color azul) corresponde a 29,99 Km/h. La velocidad promedio del cuarto grupo denso (color amarillo) corresponde a 44,38 Km/h. La velocidad promedio del primer grupo atípico (color negro) corresponde a 57,38 Km/h. La velocidad promedio del segundo grupo atípico (color negro) corresponde a 84,0 Km/h. La velocidad promedio del tercer grupo atípico (color negro) corresponde a 70,67 Km/h. La elaboración es propia y la fuente corresponde a datos de la experimentación.

Figura 33

Mapa del ciclo 24 generado por el algoritmo Dyclee en paralelo para dataset de Roma en el intervalo de tiempo de procesamiento de 5 minutos



Nota: En esta figura se muestra el mapa generado por el algoritmo en secuencia. El mapa corresponde al ciclo 24 con intervalo de 5 minutos haciendo uso del dataset de Roma, Se muestran las 7 agrupaciones por punto generadas. Los grupos densos corresponden a los que se encuentran identificados con colores rojo, verde, azul y amarillo; los grupos atípicos se identifican con el color negro y con un signo negativo al comienzo de cada etiqueta. Las etiquetas son las velocidades promedios de cada grupo formado puesto que el algoritmo agrupa en base a rangos de velocidades. La velocidad promedio del primer grupo denso (color rojo) corresponde a 0,94 Km/h. La velocidad promedio del segundo grupo denso (color verde) corresponde a 15,91 Km/h. La velocidad promedio del tercer grupo denso (color azul) corresponde a 29,99 Km/h. La velocidad promedio del cuarto grupo denso (color amarillo) corresponde a 44,38 Km/h. La velocidad promedio del primer grupo atípico (color negro) corresponde a 57,38 Km/h. La velocidad promedio del segundo grupo atípico (color negro) corresponde a 84,0 Km/h. La velocidad promedio del tercer grupo atípico (color negro) corresponde a 70,67 Km/h. La elaboración es propia y la fuente corresponde a datos de la experimentación.

Resultados de la tercera experimentación

Tiempos promedio por ciclo

Ejecución secuencial. Se realizó el experimento para obtener el tiempo promedio por ciclo del algoritmo Dyclee secuencial para 4 diferentes intervalos de tiempo de procesamiento: cinco, tres, dos y un minuto.

Ejecución en paralelo. Se realizó el experimento para obtener el tiempo promedio por ciclo del algoritmo Dyclee en paralelo para 4 diferentes intervalos de tiempo de procesamiento: 5 minutos, 3 minutos, 2 minutos y 1 minuto.

Resultados de los tiempos promedio por ciclo con dataset de Guayaquil

Los resultados del experimento para el dataset de Guayaquil se pueden apreciar a continuación en la tabla 15. En esta, se muestran diferentes columnas. La primera muestra los intervalos de tiempo de procesamiento del algoritmo, la segunda columna muestra los tiempos promedios por ciclo obtenidos por el algoritmo Dyclee secuencial en cada intervalo de tiempo. La tercera columna muestra los tiempos promedios por ciclo obtenidos por el algoritmo Dyclee paralelo en cada intervalo de tiempo. Además, se muestra el tiempo promedio por ciclo total de todas las ejecuciones, tanto en secuencia como en paralelo, siendo los valores de 1,58 segundos y 1,28 segundos respectivamente. De esta forma se comprueba que en el algoritmo en paralelo los tiempo promedios por ciclo son menores, con respecto a los tiempos por ciclo del algoritmo en secuencia; lo que demuestra por qué el tiempo total de la ejecución también es menor en paralelo. Adicionalmente, se muestran las desviaciones estándar de los tiempos promedio totales por ciclo de cada ejecución, en el caso secuencial la desviación estándar es 0,063 y en la implementación en paralelo es 0,045. Esto indica que mientras menor sea el intervalo de tiempo, el tiempo promedio por ciclo disminuirá ligeramente.

Tabla 15

Comparación de los resultados del tiempo promedio por ciclo del algoritmo Dyclee secuencial y paralelo para el dataset de Guayaquil para 4 diferentes intervalos de tiempo de procesamiento

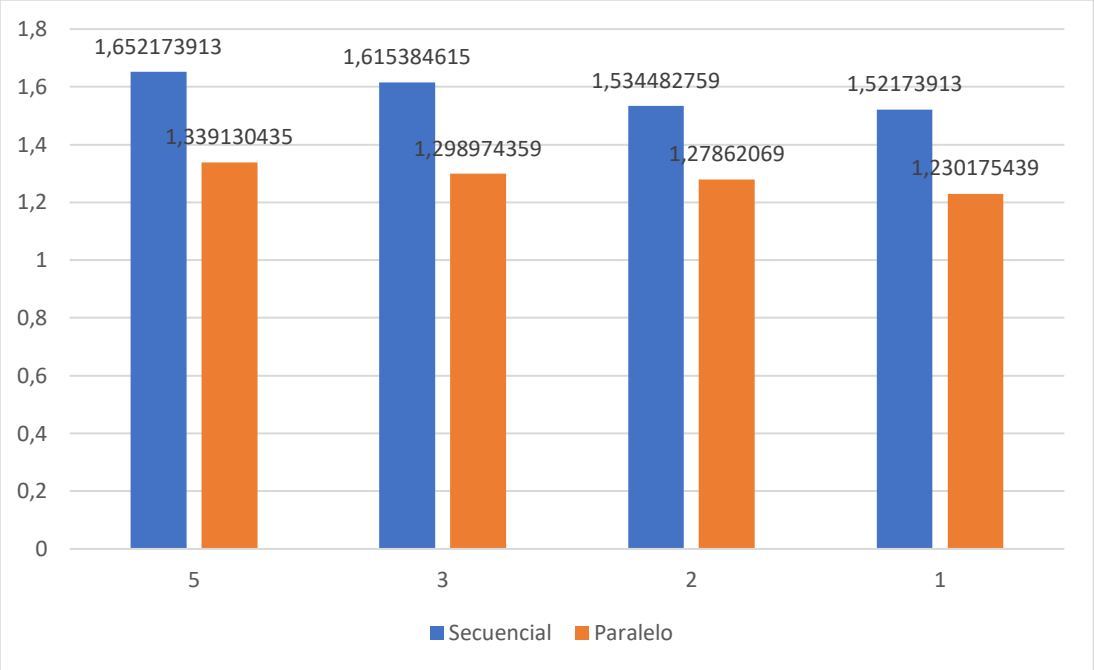
Minutos	Cantidad de ciclos	Tiempo promedio por ciclo secuencial	Tiempo promedio por ciclo paralelo
5	23	1,652173913	1,339130435
3	39	1,615384615	1,298974359
2	58	1,534482759	1,27862069
1	115	1,52173913	1,230175439
Tiempo promedio por ciclo total		**1,580945104**	**1,286725231**
Desviación estándar		**0,063044217**	**0,045313387**

Nota: En la presente tabla se detalla la comparativa de los resultados del tiempo promedio por ciclo del algoritmo Dyclee secuencial y paralelo para el dataset de Guayaquil para 4 intervalos de tiempo de procesamiento. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

En la figura 34 se puede observar un gráfico de barras que representa los resultados de los tiempos promedios por ciclo de las ejecuciones del algoritmo Dyclee tanto en secuencial como en paralelo para el dataset de Guayaquil, el eje vertical representa el tiempo promedio por ciclo en segundos mientras que en el eje horizontal se puede apreciar los diferentes intervalos de tiempo de procesamiento en minutos. Se puede visualizar como el resultado del tiempo promedio por ciclo total de las ejecuciones del algoritmo Dyclee secuencial, representadas en azul, son superiores en comparación con los resultados de las ejecuciones en paralelo, representadas en naranja, lo cual es indica que el algoritmo en paralelo es capaz de procesar los datos de forma más rápida.

Figura 34

Comparación de tiempos promedios por ciclo del algoritmo Dyclee secuencial y paralelo para dataset de Guayaquil



Nota: En esta figura se muestra los resultados obtenidos en cada ejecución con intervalo de tiempo específico, se visualiza que en todos los casos los tiempos promedios por ciclo del algoritmo Dyclee en paralelo, para el dataset de Guayaquil, es menor que en secuencia. La elaboración es propia y la fuente corresponde a datos de la experimentación.

Resultados de los tiempos promedio por ciclo con dataset de Roma

Los resultados del experimento para el dataset de Roma se pueden apreciar a continuación en la tabla 16. En esta, se muestran diferentes columnas. La primera muestra los intervalos de tiempo de procesamiento del algoritmo, la segunda columna muestra los tiempos promedios por ciclo obtenidos por el algoritmo Dyclee secuencial en cada intervalo de tiempo. La tercera columna muestra los tiempos promedios por ciclo obtenidos por el algoritmo Dyclee paralelo en cada intervalo de tiempo. Además, se muestra el tiempo promedio por ciclo total de todas las ejecuciones, tanto en secuencia como en paralelo, siendo los valores de 1,71 segundos y 1,43 segundos respectivamente. De esta forma se comprueba que en el algoritmo en paralelo los tiempo promedios por ciclo son menores, con respecto a los tiempos por ciclo del algoritmo en secuencia; lo que demuestra por qué el tiempo total de la ejecución también

es menor en paralelo. Adicionalmente, se muestran las desviaciones estándar de los tiempos promedio totales por ciclo de cada ejecución, en el caso secuencial la desviación estándar es 0,0308 y en la implementación en paralelo es 0,0445, lo cual denota que los tiempos promedio por ciclo secuencial tienen una l mayor cohesión que los resultados en paralelo, aquí se muestra también una diferencia con respecto al dataset de Guayaquil, ya que el tiempo promedio por ciclo no necesariamente se reduce al disminuir el intervalo de tiempo de procesamiento.

Tabla 16

Comparación de los resultados del tiempo promedio por ciclo del algoritmo Dyclee secuencial y paralelo para el dataset de Roma para 4 diferentes intervalos de tiempo de procesamiento

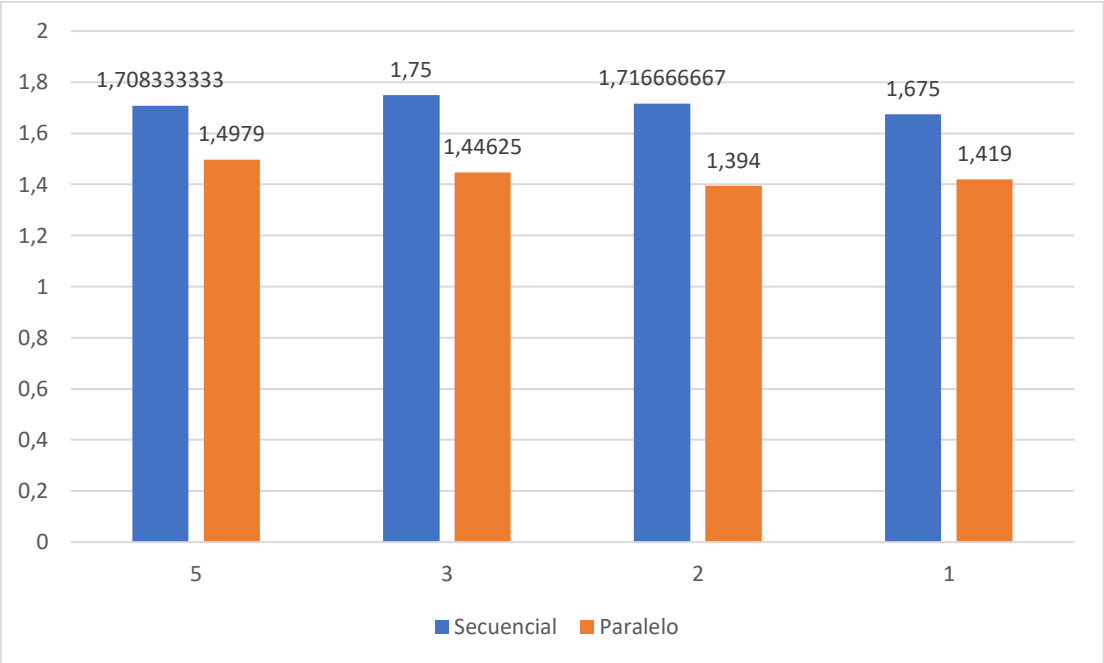
Intervalo	Cantidad de ciclos	Tiempo promedio por ciclo secuencial	Tiempo promedio por ciclo paralelo
5	24	1,708333333	1,497916667
3	40	1,75	1,44625
2	60	1,716666667	1,394
1	120	1,675	1,419083333
Tiempo promedio por ciclo total		**1,7125**	**1,4393125**
desviación estándar		**0,030807046**	**0,044515986**

Nota: En la presente tabla se detalla la comparativa de los resultados del tiempo promedio por ciclo del algoritmo Dyclee secuencial y paralelo para el dataset de Roma para 4 intervalos de tiempo de procesamiento. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

En la figura 35 se puede observar un gráfico de barras que representa los resultados de los tiempos promedios por ciclo de las ejecuciones del algoritmo Dyclee tanto en secuencial como en paralelo para el dataset de Roma, el eje vertical representa el tiempo promedio por ciclo en segundos mientras que en el eje horizontal se puede apreciar los diferentes intervalos de tiempo de procesamiento en minutos. Se puede visualizar como el resultado del tiempo promedio por ciclo total de las ejecuciones del algoritmo Dyclee secuencial, representadas en azul, son superiores en comparación con los resultados de las ejecuciones en paralelo, representadas en naranja, lo cual es indica que el algoritmo en paralelo es capaz de procesar los datos de forma más rápida.

Figura 35

Comparación de tiempos promedios por ciclo del algoritmo Dyclee secuencial y paralelo para dataset de Roma



Nota: En esta figura se muestra los resultados obtenidos en cada ejecución con intervalo de tiempo específico, se visualiza que en todos los casos los tiempos promedios por ciclo del algoritmo Dyclee en paralelo, para el dataset de Roma, es menor que en secuencia. La elaboración es propia y la fuente corresponde a datos de la experimentación.

Resultados de la experimentación en Amazon Web Services

Tiempos totales de ejecución

Para realizar este experimento se utilizó el servicio de Amazon Web Services llamado EMR (Elastic Map Reduce), para poder montar un cluster de computadoras en la nube que permitan ejecutar el algoritmo Dyclee de manera secuencial y en paralelo.

Ejecución secuencial. Se realizó el experimento en la nube para obtener el tiempo total de ejecución del algoritmo Dyclee secuencial para 4 diferentes intervalos de tiempo de procesamiento: cinco, tres, dos y un minuto.

Ejecución en paralelo. Se realizó el experimento en la nube para obtener el tiempo total de ejecución del algoritmo Dyclee en paralelo para 4 diferentes intervalos de tiempo de procesamiento: 5 minutos, 3 minutos, 2 minutos y 1 minuto.

Resultados de los tiempos totales de ejecución con dataset de Guayaquil en Amazon Web Services

Los resultados del experimento para el dataset de Guayaquil en Amazon Web Services se pueden apreciar a continuación en la tabla 17. En esta, se muestran diferentes columnas. La primera columna consiste en los intervalos de tiempo de procesamiento que utiliza el algoritmo para procesar los datos en minutos (cinco minutos, tres minutos, dos minutos, un minuto). Luego, en la siguiente columna, se muestra la cantidad de ciclos que genera el algoritmo en cada ejecución, es decir, 23 ciclos en cinco minutos, 39 ciclos en tres minutos, 58 ciclos en dos minutos y 115 ciclos en un minuto. A continuación, se muestran las columnas con los tiempos totales que tarda en ejecutarse el código de forma secuencial y en paralelo y cuyos resultados muestran que en el algoritmo Dyclee en paralelo, el tiempo total de ejecución en todos los casos son mayores que el tiempo total de la ejecución del algoritmo Dyclee secuencial; esto es debido a que aunque está disponible un clúster de computadoras para poder distribuir el procesamiento en paralelo, AWS solo está usando una sola computadora para realizar el trabajo. La única manera de poder utilizar todo el cluster de computadoras disponibles en EMR es adaptar el algoritmo de Dyclee para que utilice la abstracción denominada Map-reduce.

Tabla 17

Comparación de los resultados del algoritmo Dyclee secuencial y paralelo para el dataset de Guayaquil para 4 diferentes intervalos de tiempo de procesamiento en la plataforma Amazon Web Services

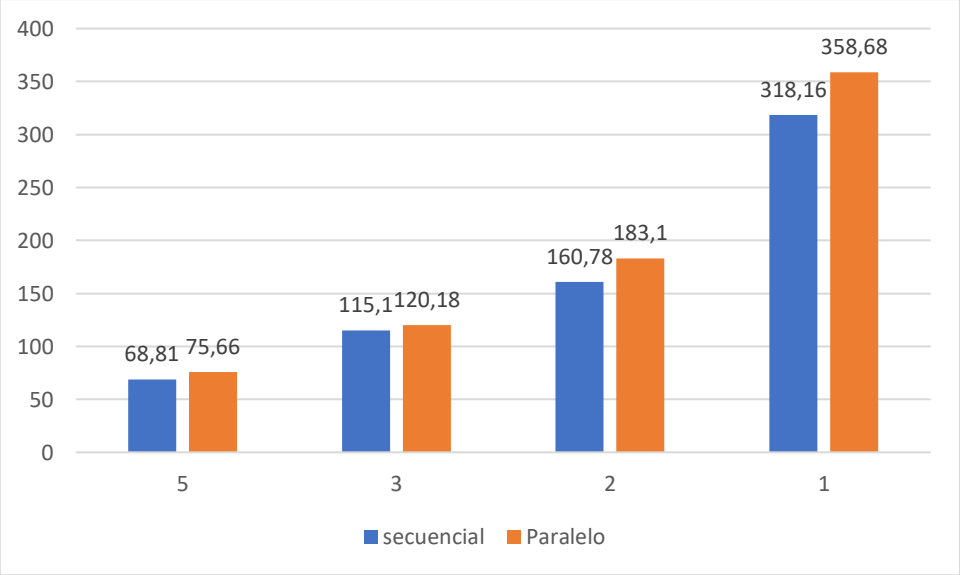
Intervalo	Cantidad de Ciclos	Tiempo total secuencial	Tiempo total paralelo
5	23	68,81	75,66
3	39	115,1	120,18
2	58	160,78	183,1
1	115	318,16	358,68
Tiempo promedio		**165,7125**	**184,405**

Nota: En la presente tabla se detalla la información más importante resumida de los resultados del algoritmo Dyclee secuencial y paralelo para el dataset de Guayaquil para 4 intervalos de tiempo de procesamiento en la plataforma Amazon Web Services. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

En la figura 36 se puede observar un gráfico de barras que representa los resultados de los tiempos totales de las ejecuciones del algoritmo Dyclee tanto en secuencial como en paralelo para el dataset de Guayaquil, el eje vertical representa el tiempo total de ejecución en segundos mientras que en el eje horizontal se puede apreciar los diferentes intervalos de tiempo de procesamiento en minutos. Se puede visualizar como el resultado del tiempo total de cada ejecución del algoritmo Dyclee secuencial, representadas en azul, es menor en comparación con las ejecuciones en paralelo representadas en naranja, lo cual indica que el algoritmo en paralelo Dyclee ejecutado en la plataforma de Amazon Web Services requiere nuevas implementaciones para funcionar correctamente en la nube.

Figura 36

Comparación de tiempos totales de ejecución del algoritmo Dyclee secuencial y paralelo para dataset de Guayaquil en AWS



Nota: En esta figura se muestra los resultados obtenidos en cada ejecución con intervalo de tiempo específico en la plataforma Amazon Web Services, se visualiza que en todos los casos los tiempos de ejecución del algoritmo Dyclee en paralelo, para el dataset de Guayaquil, es mayor que en secuencia . La elaboración es propia y la fuente corresponde a datos de la experimentación.

Análisis de los resultados de juicio de expertos

Las respuestas referente a la primera pregunta que se detalla a continuación: ¿Es adecuada la implementación en una arquitectura en paralelo del algoritmo de agrupamiento

Dyclee para procesar grandes volúmenes de datos de trayectorias GPS? se muestran en la tabla 18. En esta se observa que dos expertos, que representan el 66,7% del total de encuestados, están totalmente de acuerdo con lo que se plantea en la primera pregunta; y un experto, que representa el 33,3% del total de encuestados, está de acuerdo. Los resultados indican que los expertos consideran que la implementación de la arquitectura en paralelo de algoritmo Dyclee es adecuada para procesar grandes volúmenes de datos de trayectorias GPS.

Tabla 18

Resultados de la primera pregunta del cuestionario de juicio de expertos

Alternativa	Frecuencia Absoluta	Frecuencia Relativa
Totalmente de acuerdo	2	66,7%
De acuerdo	1	33,3%
Ni de acuerdo ni en desacuerdo	0	0%
En desacuerdo	0	0%
Totalmente en desacuerdo	0	0%
Total	**3**	**100%**

Nota: En la presente tabla se detallan los resultados de la primera pregunta del cuestionario de juicio de expertos. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

Las respuestas de la segunda pregunta que se detalla a continuación: ¿Es adecuada la modificación realizada en el algoritmo de agrupamiento Dyclee para que este pueda agrupar en base a rangos de velocidades? se muestran en la tabla 19. En esta se visualiza que dos expertos, que representan el 66,7% del total de encuestados, están totalmente de acuerdo con lo que se plantea en la segunda pregunta; y un experto, que representa el 33,3% está de acuerdo. Los resultados indican que los expertos consideran que la modificación realizada en el algoritmo de agrupamiento Dyclee es adecuada para que pueda agrupar en base a rangos de velocidades.

Tabla 19

Resultados de la segunda pregunta del cuestionario de juicio de expertos

Alternativa	Frecuencia Absoluta	Frecuencia Relativa
Totalmente de acuerdo	2	66,7%
De acuerdo	1	33,3%
Ni de acuerdo ni en desacuerdo	0	0%
En desacuerdo	0	0%
Totalmente en desacuerdo	0	0%
Total	**3**	**100%**

Nota: En la presente tabla se detallan los resultados de la segunda pregunta del cuestionario de juicio de expertos. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

Los resultados de la tercera pregunta que fue planteada de la siguiente forma: ¿Cree usted que la presente investigación cumple con los objetivos propuestos? se presentan en la tabla 20. En esta tabla se observa que dos expertos, representando el 66,7% del total de encuestados, respondieron que están de acuerdo con lo que se plantea en la segunda pregunta; y un experto, que representa el 33,3% del total de encuestados, respondió que está totalmente de acuerdo. Los resultados demuestran que los objetivos propuestos en el presente estudio fueron cumplidos en su totalidad.

Tabla 20

Resultados de tercera pregunta del cuestionario de juicio de expertos

Alternativa	Frecuencia Absoluta	Frecuencia Relativa
Totalmente de acuerdo	1	33,3%
De acuerdo	2	66,7%
Ni de acuerdo ni en desacuerdo	0	0
En desacuerdo	0	0
Totalmente en desacuerdo	0	0
Total	**3**	**100%**

Nota: En la presente tabla se detallan los resultados de la tercera pregunta del cuestionario de juicio de expertos. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

Las respuestas de la cuarta pregunta formulada de la siguiente manera: ¿Considera usted que la presente investigación es acorde con los métodos tecnológicos actuales y aportan soluciones a temáticas de la actualidad? se muestran en la tabla 21. En esta se visualiza que dos expertos, representando el 66,7% del total de encuestados, están de acuerdo con lo que se formula en la cuarta pregunta; y un experto, que representa el 33,3% del total de encuestados,

indica que está totalmente de acuerdo. Los resultados demuestran que los expertos consideran que la temática de la presente investigación es acorde con los métodos tecnológicos actuales y aportan soluciones a temas de actualidad.

Tabla 21

Resultados de la cuarta pregunta del cuestionario de juicio de expertos

Alternativa	Frecuencia Absoluta	Frecuencia Relativa
Totalmente de acuerdo	1	33,3%
De acuerdo	2	66,7%
Ni de acuerdo ni en desacuerdo	0	0
En desacuerdo	0	0
Totalmente en desacuerdo	0	0
Total	**3**	**100%**

Nota: En la presente tabla se detallan los resultados de la cuarta pregunta del cuestionario de juicio de expertos. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

Los resultados de la quinta pregunta que consiste: ¿Los métodos experimentales usados son acordes a los resultados esperados en la presente investigación? se presentan en la tabla 22. En esta tabla se visualiza que dos expertos, representados por el 66,7% del total de encuestados, están de acuerdo con lo que se formula en la quinta interrogante; y un experto, que representa el 33,3% del total de encuestados, está totalmente de acuerdo. Estos resultados dan a entender que los expertos consideran que los métodos experimentales usados si son acordes a los resultados esperados en la presente investigación.

Tabla 22

Resultados de la quinta pregunta del cuestionario de juicio de expertos

Alternativa	Frecuencia Absoluta	Frecuencia Relativa
Totalmente de acuerdo	1	33,3%
De acuerdo	2	66,7%
Ni de acuerdo ni en desacuerdo	0	0
En desacuerdo	0	0
Totalmente en desacuerdo	0	0
Total	**3**	**100%**

Nota: En la presente tabla se detallan los resultados de la quinta pregunta del cuestionario de juicio de expertos. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

Las respuestas de la sexta pregunta que trata de: ¿Los resultados obtenidos reflejan lo esperado acorde a los métodos usados en la presente investigación? se muestran en la tabla 23.

En esta se observa que dos expertos (66,7% del total de encuestados) están de acuerdo con lo que se plantea en la sexta pregunta y un experto (33,3% del total de encuestados) está totalmente de acuerdo con lo planteado. Los resultados demuestran que los expertos si consideran que lo resultados obtenidos del presente estudio van acordes a los métodos empleados.

Tabla 23

Resultados de la sexta pregunta del cuestionario de juicio de expertos

Alternativa	Frecuencia Absoluta	Frecuencia Relativa
Totalmente de acuerdo	1	33,3%
De acuerdo	2	66,7%
Ni de acuerdo ni en desacuerdo	0	0
En desacuerdo	0	0
Totalmente en desacuerdo	0	0
Total	**3**	**100%**

Nota: En la presente tabla se detallan los resultados de la sexta pregunta del cuestionario de juicio de expertos. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

Los resultados de la séptima pregunta que consiste en: ¿El desarrollo de la investigación está basado en aspectos teóricos y científicos? se presentan en la tabla 24. En esta tabla se observa que dos expertos (66,7%) están de acuerdo con lo planteado en la séptima pregunta y un experto (33,3%) está totalmente de acuerdo. Los resultados demuestran que los expertos si consideran que la presente investigación está basada en aspectos teóricos y científicos.

Tabla 24

Resultados de la séptima pregunta del cuestionario de juicio de expertos

Alternativa	Frecuencia Absoluta	Frecuencia Relativa
Totalmente de acuerdo	1	33,3%
De acuerdo	2	66,7%
Ni de acuerdo ni en desacuerdo	0	0
En desacuerdo	0	0
Totalmente en desacuerdo	0	0
Total	**3**	**100%**

Nota: En la presente tabla se detallan los resultados de la séptima pregunta del cuestionario de juicio de expertos. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

Los resultados de la octava pregunta que se formuló de la siguiente forma: ¿La metodología no experimental – transversal está relacionada con el desarrollo del proyecto? Se

muestran en la tabla 24. En esta tabla se visualiza que dos expertos (66,7%) están de acuerdo con lo que se plantea en la última pregunta del cuestionario y un experto (33,3%) está totalmente de acuerdo. Estos resultados demuestran que los expertos consideran adecuada la metodología empleada en el presente estudio.

Tabla 25

Resultados de la octava pregunta del cuestionario de juicio de expertos

Alternativa	Frecuencia Absoluta	Frecuencia Relativa
Totalmente de acuerdo	1	33,3%
De acuerdo	2	66,7%
Ni de acuerdo ni en desacuerdo	0	0%
En desacuerdo	0	0%
Totalmente en desacuerdo	0	0%
Total	**3**	**100%**

Nota: En la presente tabla se detallan los resultados de la octava pregunta del cuestionario de juicio de expertos. La elaboración de la tabla es propia y la fuente proviene de datos generados por el trabajo investigativo.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Tras la realización del desarrollo del presente trabajo de investigación, se han obtenido las siguientes conclusiones:

- Se realizaron las modificaciones necesarias para que el algoritmo Dyclee pueda ser ejecutado en un ambiente local; usando el lenguaje de programación Python, para el procesamiento de datos de trayectorias GPS.
- Se realizaron las adaptaciones necesarias para que el algoritmo Dyclee sea capaz de agrupar datos en base a rangos de velocidades. Además, se utilizó el paralelismo basado en procesos mediante la librería multiprocessing propia de Python, y se implementó colas como canal de comunicación entre procesos.
- Se implementó una arquitectura de procesamiento en la nube por medio de Amazon Web Services y su servicio EMR, para comprobar el comportamiento de la ejecución del algoritmo en este ambiente, lo que dio como resultado que lo tiempos de ejecución del algoritmo en paralelo eran mayores a los tiempos del algoritmo en secuencia.
- Se realizaron tres experimentos diferentes, mediante el uso de dos datasets diferentes (Guayaquil y Roma). El primero, para hacer una comparación del comportamiento de los tiempos totales de ejecución del algoritmo en secuencia y en paralelo. El segundo

con el fin de identificar las diferentes velocidades y grupos que se generan y el tercero para hacer una comparación del comportamiento de los tiempos promedios por ciclo del algoritmo en secuencia y en paralelo.

- Los resultados del primer experimento demostraron que la ejecución del algoritmo en paralelo generaba tiempo totales de ejecución menores a los de la ejecución del algoritmo en secuencia en ambos datasets.
- En el segundo experimento se pudo comprobar que los grupos generados por el algoritmo Dyclee en el dataset de Guayaquil eran constantes en todos los intervalos de tiempo de procesamiento, la desviación estándar en cada grupo indica que no existe una gran variación en las velocidades dentro de un mismo grupo, mientras que por intervalos las velocidades de varían mucho más. En el dataset de Roma se pudo notar diferencias en la generación de los grupos ya que no fue constante en todos los intervalos, puesto que el intervalo de tiempo de procesamiento de 1 minuto generó 1 grupo menos que todos los otros, alterando un poco el agrupamiento de las velocidades por grupo y por intervalo. Al comparar los resultados del algoritmo en secuencia con respecto a paralelo se puede comprobar que en ambos casos se generan los mismos grupos y velocidades, lo que demuestra que estos son independientes de la manera como se procesen los datos.
- El tercer experimento demuestra que la ejecución del algoritmo en paralelo genera promedios de tiempo por ciclo menores a los de la ejecución del algoritmo en secuencia en ambos datasets.
- Se presentaron los resultados de las agrupaciones formadas mediante imágenes .png haciendo uso de librerías propias de Python; donde los diferentes grupos se distinguieron por un color distinto y una etiqueta que representa la velocidad promedio de cada grupo, los grupos atípicos se representaron en color negro con un

signo negativo al inicio de la etiqueta. Cada ciclo que realizó el algoritmo muestra una imagen diferente del mapa formado con las agrupaciones hasta completar todos los ciclos de procesamiento.

- Se validaron los resultados obtenidos en todos los experimentos utilizando estadística descriptiva, utilizando medidas de tendencia central como el promedio para identificar las tendencias de los resultados generados en los experimentos y medidas de dispersión para identificar la cohesión de los resultados producidos por el algoritmo Dyclee.
- Se elaboró un artículo científico con la plantilla de la Revista Ibérica de Sistemas y Tecnologías de información (RISTI), el cual será enviado para la revisión y publicación en dicha revista.

Recomendaciones

- Identificar datasets que sean de fácil acceso y con poca cantidad de ruido en sus datos, para un mejor análisis.
- Implementar el uso de un gestor de base de datos para administrar los datos de forma más optima y estandarizar el nombre de los campos para evitar cambios significativos en el código cada que se realice una experimentación con distintos conjuntos de datos.
- Implementar el uso de otro lenguaje de programación distinto a Python que soporte paralelismo entre procesos y que no posea restricciones como el GIL de Python.

Trabajos futuros

- Desarrollar una versión en paralelo del algoritmo Dyclee implementando librerías más modernas proporcionadas por Python como concurrent.futures, que incluya otra metodología de comunicación entre procesos y permita optimizar aún más los tiempos de ejecución en una arquitectura en paralelo.

- Elaborar un estudio que incluya el uso de la abstracción de map reduce en el algoritmo Dyclee para utilizar la arquitectura en la nube de Amazon Web Services de una manera más eficiente.
- Elaborar un estudio que incluya el uso de Spark como herramienta para montar una arquitectura en paralelo del algoritmo Dyclee en Amazon Web Services.

REFERENCIAS BIBLIOGRÁFICAS

Acervo Lima. (2022). *Procesamiento paralelo en Python*. https://es.acervolima.com/procesamiento-paralelo-en-python/#google_vignette

Agudelo, G., Aigneren, M., & Ruiz, J. (2008). Diseños De Investigación Experimental Y No-Experimental. In *Centro de Estudios de Opinión* (pp. 1–46). http://bibliotecadigital.udea.edu.co/dspace/bitstream/10495/2622/1/AgudeloGabriel_disenosinvestigacionexperimental.pdf

Aguilar, L. J. (2013). *Big Data Analisis de Grandes Volumenes de Datos en Organizaciones*.

Amazon Web Service. (2022). *¿Qué es Python?* https://aws.amazon.com/es/what-is/python/

Amazon Web Services. (2017). *AWS | Almacenamiento de datos seguro en la nube (S3)*. Amazon.Com. https://aws.amazon.com/es/s3/?nc1=h_ls

Barbosa Roa, N., Travé-Massuyès, L., & Grisales-Palacio, V. H. (2019). DyClee: Dynamic clustering for tracking evolving environments. *Pattern Recognition*, *94*, 162–186. https://doi.org/10.1016/j.patcog.2019.05.024

Barrionuevo, C., Ierache, J., & Sattolo, I. (2020). Reconocimiento de emociones a través de expresiones faciales con el empleo de aprendizaje supervisado aplicando regresión logística. *XXVI Congreso Argentino de Ciencias de La Computación (CACIC)*, 491–500. http://sedici.unlp.edu.ar/handle/10915/114089

Campos, M. G. (2009). *APLICACION DE TECNICAS DE CLUSTERING PARA LA MEJORA DEL APRENDIZAJE*.

Cardenas, M., Medel, R., Castillo, J., Vázquez, J. C., & Casco, O. (2015). Modelos de Aprendizaje Supervisados : aplicaciones para la predicción de incendios forestales en la provincia de Córdoba. *XVII Workshop de Investigadores En Ciencias de La Computación*, 1–5. http://sedici.unlp.edu.ar/handle/10915/45467%0Ahttp://hdl.handle.net/10915/45467

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt Maranto, L., & Zdeborová, L. (2019). Machine learning and the physical sciences. *Reviews of Modern Physics*, *91*(4). https://doi.org/10.1103/RevModPhys.91.045002

Dafir, Z., Lamari, Y., & Slaoui, S. C. (2020). A survey on parallel clustering algorithms for Big Data. In *Artificial Intelligence Review* (Vol. 54, Issue 4). Springer Netherlands. https://doi.org/10.1007/s10462-020-09918-2

Díaz, S., & González, L. (2010). *Reflexiones sobre los conceptos velocidad y rapidez de una partícula en física*. *56*(2), 181–189. https://www.redalyc.org/articulo.oa?id=57048175005

Dib Ashur, J., Vallón, J., Martínez, C., & Said, C. (2016). SACO: Un algoritmo de clustering espacial con hormigas inteligentes. *In Simposio Argentino de Inteligencia Artificial (ASAI)*, 17–24.

Farnos, J. (2018). *Los algoritmos supervisados nos llevan al personalized learning y a sus interfaces (construcción)*. https://juandomingofarnos.wordpress.com/2018/11/03/los-algoritmos-supervisados-nos-llevan-al-personalized-learning-y-a-sus-interfaces-construccion/

Garcia Dihigo, J. (2016). *Metodologia de la investigacion para administradores*. Ediciones de la U. https://elibro.net/es/lc/uguayaquil/titulos/70269

Godoy, A. (2017). Técnicas de aprendizaje de máquina utilizadas para la minería de texto. *Investigacion Bibliotecológica*, *31*(71), 103–126. https://doi.org/10.22201/iibi.0187358xp.2017.71.57812

González, F. (2015). Modelos de aprendizaje computacional en reumatología. *Revista Colombiana de Reumatología*, *22*(2), 77–78. https://doi.org/10.1016/j.rcreu.2015.6.001

Gutiérrez, J., García, J., & Salas, M. (2016). Big (Geo)Data en Ciencias Sociales: Retos y Oportunidades. *Revista de Estudios Andaluces (RAE)*, *33*(1), 1–23.

https://doi.org/10.12795/rea.2016.i33

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques* (Elsevier). https://doi.org/10.1016/C2009-0-61819-5

Hernández, J. (2016). Clustering basado en el algoritmo K-means para la identificación de grupos de pacientes quirúrgicos. *Universidad Santo Tomás, Seccional Bucaramanga*, 1–8. www.udi.edu.co/congreso/historial/.../descargar_pdf1.php?f...%0A

Hu, X., Liu, L., Qiu, N., Yang, D., & Li, M. (2018). A MapReduce-based improvement algorithm for DBSCAN. *Journal of Algorithms and Computational Technology*, *12*(1), 53–61. https://doi.org/10.1177/1748301817735665

Jothi, N., Rashid, N. A., & Husain, W. (2015). Data Mining in Healthcare - A Review. *Procedia Computer Science*, *72*, 306–313. https://doi.org/10.1016/j.procs.2015.12.145

Lapeira, O., Ceruto, T., Rosete, A., & Díaz, H. (2017). Algoritmo paralelo para la obtención de predicados difusos. *Revista Cubana de Ciencias Informáticas*, *11*(2), 117–133. https://www.redalyc.org/articulo.oa?id=378350964009

Loh, W.-K., & Park, Y.-H. (2014). A survey on density-based clustering algorithms. *Springer*, *280*, 775–780. https://doi.org/10.1007/978-3-642-41671-2_98

López, J. (2009). *Algoritmos y programación guía para docentes*. *December 2008*, 96.

Mahesh, B. (2020). Machine Learning Algorithms-A Review. *International Journal of Science and Research (IJSR)*, *9*(1), 381–386. https://doi.org/10.21275/ART20203995

Martínez Ruiz, H. (2012). *Metodología de la Investigación*. Cengage Learning. https://elibro.net/es/lc/uguayaquil/titulos/39957

Marzal Varó, A., Gracia Luengo, I., & García Sevilla, P. (2009). *Introducción a la programación con Python* (Universita). Universitat Jaume I. Servei de Comunicació i Publicacions. https://doi.org/10.6035/sapientia93

Mendoza, S. (2012). Criterio de expertos. Su procesamiento a través del método delphy.

Universitat de Barcelona, 7–12. http://www.ub.edu/histodidactica/index.php%3Foption%3Dcom_content%26view%3Darticle%26id%3D21:criterio-de-expertos-su-procesamiento-a-traves-del-metodo-delphy%26catid%3D11:metodologia-y-epistemologia%26Itemid%3D103

Microsoft Azure. (2018). *¿Qué es el Aprendizaje Automático?* Microsoft. https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-machine-learning-platform/

Osman, A. S. (2019). Data Mining Techniques: Review. *International Journal of Data Science Research*, *2*(1), 1–4. https://www.educba.com/7-data-

Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*, *30*(4), 431–448. https://doi.org/10.1016/j.jksuci.2017.06.001

Pagola, M., De Miguel, L., Marco, C., & Bustince, H. (2015). Algoritmo de clustering intervalo-valorado difuso. *Actas de La XVI Conferencia CAEPIA*, 1–10. http://giara.unavarra.es/

programmerclick. (2022). *Procesamiento paralelo en Python: una guía de programación de ejemplo*. https://programmerclick.com/article/323485547/

Python Software Foundation. (2021). *multiprocessing — Paralelismo basado en procesos*. 3.10.0rc2 Documentation. https://docs.python.org/es/3/library/multiprocessing.html

Quispe, A. M., Hinojosa-Ticona, Y., Miranda, H. A., & Sedano, C. A. (2021). Serie de Redación Científica: Revisiones Sistemáticas. *Revista Del Cuerpo Medico Hospital Nacional Almanzor Aguinaga Asenjo*, *14*(1), 94–99. https://doi.org/10.35434/rcmhnaaa.2021.141.906

Redacción KeepCoding. (2022). *¿Qué son los Datasets? [4 sitios donde encontrarlos]*. https://keepcoding.io/blog/que-son-datasets/

Reyes, G., Lanzarini, L., Estrebou, C., & Maquilón, V. (2021). Vehicular Flow Analysis Using Clusters. *XXVII CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACION (CACIC)(Modalidad Virtual, 4 Al 8 de Octubre de 2021)*, 261–270. http://sedici.unlp.edu.ar/handle/10915/130341

Reyes, G., Lanzarini, L., Hasperue, W., & Bariviera, A. (2021). A proposal for a pivot-based vehicle trajectory clustering method. *Transportation Research Record*, *20*(10), 1–20. https://doi.org/DOI: 10.1177/ToBeAssigned

Reyes, G., Lanzarini, L., Hasperué, W., & Bariviera, A. F. (2020). GPS trajectory clustering method for decision making on intelligent transportation systems. *Journal of Intelligent and Fuzzy Systems*, *38*(5), 5529–5535. https://doi.org/10.3233/JIFS-179644

Reyes Zambrano, G. (2019). GPS trajectory compression algorithm. *Communications in Computer and Information Science*, *959*, 57–69. https://doi.org/10.1007/978-3-030-12018-4_5

Reyes Zambrano, G., Córdova Rizo, F., León Granizo, O., & Carabali Noriega, E. (2022). *GPS Trajectory segmentation and clustering method*. *1*, 1–8.

Sánchez Meca, J. (2010). Cómo realizar una revisión sistemática y un meta-análisis. *Universidad de Murcia*, *38*(2), 53–64.

Sinaga, K. P., & Yang, M.-S. (2020). Unsupervised K-means clustering algorithm. *IEEE Access*, *8*, 80716–80727. https://doi.org/10.1109/ACCESS.2020.2988796

Sowmya, R., & Suneetha, K. R. (2017). Data Mining with Big Data. *2017 11th International Conference on Intelligent Systems and Control, (ISCO)*, 246–250. https://doi.org/10.1109/ISCO.2017.7855990

Theodoridis, S., & Koutroumbas, K. (2009). *Pattern Recognition* (Elsevier). http://elsevier.com

Villegas, F. (2021). Relatividad y el Sistema de Posicionamiento Global (GPS). *Revista de*

Investigación de Física, *23*(1), 44–47. https://doi.org/10.15381/rif.v23i1.20289

Wang, J., Wu, N., Lu, X., Zhao, W. X., & Feng, K. (2021). Deep Trajectory Recovery with Fine-Grained Calibration using Kalman Filter. *IEEE Transactions on Knowledge and Data Engineering*, *33*(3), 1–14. https://doi.org/10.1109/TKDE.2019.2940950

Wang, K. J., Zhang, J. Y., Li, D., Zhang, X. N., & Guo, T. (2007). Adaptive affinity propagation clustering. *Zidonghua Xuebao/Acta Automatica Sinica*, *33*(12), 1242–1246. https://doi.org/10.1360/aas-007-1242

Wang, S., Ding, S., & Xiong, L. (2020). A new system for surveillance and digital contact tracing for COVID-19: Spatiotemporal reporting over network and GPS. *JMIR MHealth and UHealth*, *8*(6). https://doi.org/10.2196/19457

Yang, X., Stewart, K., Tang, L., Xie, Z., & Li, Q. (2018). A review of GPS trajectories classification based on transportation mode. *Sensors*, *18*(11), 1–20. https://doi.org/10.3390/s18113741

Zhang, J., Wu, G., Hu, X., Li, S., & Hao, S. (2013). A Parallel Clustering Algorithm with MPI – MKmeans. *Journal of Computers*, *8*(1), 10–17. https://doi.org/10.4304/jcp.8.1.10-17

ANEXOS

Anexo 1. Planificación de actividades del proyecto

HITO	FECHA INICIO	FECHA FIN
Fase 1: Realizar la migración del algoritmo.	**30/05/2022**	**16/06/2022**
Investigar acerca de los lenguajes de programación que permiten trabajar con multihilos, multisesión y programación en paralelo.	30/05/2022	30/05/2022
Elegir el lenguaje de programación apropiado.	31/05/2022	31/05/2022
Consultar acerca de la diferentes librerías y métodos que maneja el lenguaje de programación elegido, así como su sintaxis, para el manejo de multihilos, multisesión y programación en paralelo.	01/06/2022	01/06/2022
Comprender el funcionamiento del algoritmo escrito en lenguaje R.	03/06/2022	03/06/2022
Migrar el algoritmo escrito inicialmente en lenguaje R.	04/06/2022	13/06/2022
Probar algoritmo y verificar resultados.	14/06/2022	15/06/2022
Comprobar resultados.	16/06/2022	16/06/2022
Fase 2: Diseñar y programar algoritmo de programación en paralelo.	**17/06/2022**	**06/07/2022**
Diseño del algoritmo mediante Pseudo Código o diagramas de flujos.	17/06/2022	19/06/2022
Programar el algoritmo.	20/06/2022	29/06/2022
Realizar pruebas del algoritmo.	30/06/2022	02/07/2022
Realizar correctivos necesarios en caso de errores.	03/07/2022	04/07/2022
Mostrar resultados finales del código automatizado.	05/07/2022	06/07/2022
Fase 3: Implementar un ambiente de procesamiento en paralelo que permita procesar un algoritmo de agrupamiento dinámico de trayectorias GPS, con el uso de software de una plataforma de computación en la nube.	**09/07/2022**	**25/07/2022**
Investigar acerca de las diferentes plataformas de computación en la nube que permiten procesar algoritmos de agrupamiento dinámico de trayectorias GPS.	09/07/2022	09/07/2022
Elegir una plataforma de computación en la nube	10/07/2022	10/07/2022
Familiarizarse con la plataforma elegida.	11/07/2022	14/07/2022
Empezar a implementar las instancias, servidores y recursos necesarios en la plataforma de computación en la nube para permitir el procesamiento del algoritmo de agrupamiento dinámico.	15/07/2022	21/07/2022

Realizar las pruebas pertinentes teniendo en consideración el trabajo en conjunto de la plataforma en la nube con el algoritmo.	22/07/2022	24/07/2022
Fase 4: Realizar experimentos con grandes volúmenes de conjunto de datos de trayectorias GPS.	**25/07/2022**	**01/08/2022**
Realizar varias pruebas con la base de datos de trayectorias GPS implementando el algoritmo de programación en paralelo.	25/07/2022	28/07/2022
Mostrar los resultados obtenidos e identificar el tiempo empleado en procesar grandes volúmenes de datos.	29/07/2022	30/07/2022
Verificar que los resultados obtenidos demuestren confiabilidad y validez o si por el contrario existen errores.	31/07/2022	01/08/2022
Fase 5: Validar los resultados mediante estadística descriptiva y juicio de expertos.	**02/08/2022**	**07/08/2022**
Realizar el respectivo proceso de estadística descriptiva empleando tablas y gráficos.	02/08/2022	04/08/2022
Calcular los diferentes parámetros básicos de estadística descriptiva del conjunto de datos.	05/08/2022	06/08/2022
Emitir juicio de expertos con la ayuda de un colaborador.	07/08/2022	07/08/2022
Fase 6: Elaborar un artículo científico del presente estudio	**08/08/2022**	**26/08/2022**
Realizar la revisión de literatura orientado al tema propuesto en la siguiente investigación.	08/08/2022	10/08/2022
Elegir las fuentes necesarias y útiles para respaldar el tema de investigación.	11/08/2022	12/08/2022
Redactar de forma correcta toda la estructura del artículo científico.	13/08/2022	16/08/2022
Establecer el orden correcto de los diferentes gráficos e imágenes que se utilizarán para dar soporte a los resultados.	17/08/2022	18/08/2022
Insertar las diferentes referencias utilizadas de acuerdo con el estilo establecido.	19/08/2022	19/08/2022
Realizar una revisión profunda de todo el artículo.	20/08/2022	21/08/2022
Conseguir un asesor que pueda realizar las correcciones necesarias del trabajo en caso de ser necesario.	22/08/2022	22/08/2022
Elaborar la versión final del artículo científico teniendo en cuenta las observaciones del revisor o asesor.	23/08/2022	24/08/2022
Enviar el artículo a una revista indexada Q2-Q3.	25/08/2022	26/08/2022

Elaboración: Investigadores.
Fuente: Propia.

Anexo 4. Fundamentación Legal

El presente proyecto de titulación se fundamenta en la Constitución de la República del Ecuador, La Ley Orgánica de Educación superior y el Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Invención; lo que comprende el soporte legal del presente proyecto y se muestran a continuación:

ARTÍCULO DE LA LOES	**CONTEXTO**
¿Qué regula la LOES? **ART. 1 ÁMBITO**	Esta Ley regula el sistema de educación superior en el país, a los organismos e instituciones que lo integran; determina derechos, deberes y obligaciones de las personas naturales y jurídicas, y establece las respectivas sanciones por el incumplimiento de las disposiciones contenidas en la Constitución y la presente Ley ARTICULO 1
¿Cuál es el Objeto de esta Ley? **ART. 2 OBJETO**	Esta Ley tiene como objeto definir sus principios, garantizar el derecho a la educación superior de calidad que propenda a la excelencia, al acceso universal, permanencia, movilidad y egreso sin discriminación alguna.
<u>Entre las funciones</u> **ART. 4 DERECHO A LA EDUCACION SUPERIOR**	a) Garantizar el derecho a la educación superior mediante la docencia, la investigación y su vinculación con la sociedad, y asegurar crecientes niveles de calidad, excelencia académica y pertinencia; n) Garantizar la producción de pensamiento y conocimiento articulado con el pensamiento universal; y, ñ) Brindar niveles óptimos de calidad en la formación
Principio de Igualdad y Principio de Calidad	El **principio de igualdad** de oportunidades consiste en garantizar a todos los actores del Sistema de Educación Superior las mismas posibilidades en el acceso, permanencia, movilidad y egreso del sistema, sin discriminación de género, credo, orientación sexual, etnia, cultura, preferencia política, condición socioeconómica o discapacidad. El **principio de calidad** consiste en la búsqueda constante y sistemática de la excelencia, la pertinencia, producción óptima, transmisión del conocimiento y

	desarrollo del pensamiento mediante la autocrítica, la crítica externa y el mejoramiento permanente
ART. 87	Como requisito previo a la obtención del título, los y las estudiantes deberán acreditar servicios a la comunidad mediante prácticas o pasantías pre profesionales. debidamente monitoreadas. en los campos de su especialidad, de conformidad con los lineamientos generales definidos por el Consejo de Educación Superior.
ARTÍCULO 19.- DEL REGLAMENTO.- NÓMINA DE GRADUADOS Y NOTIFICACIÓN A LA SENESCYT	Las instituciones de educación superior notificarán obligatoriamente a la SENESCYT la nómina de los graduados y las especificaciones de los títulos que expida, en un plazo no mayor de treinta días contados a partir de la fecha de graduación. (…) este será el único medio oficial a través del cual se verificará el reconocimiento y validez del título en el Ecuador.
ARTÍCULO 144 PRINCIPIOS	Art. 144.- Tesis Digitalizadas.- Todas las instituciones de educación superior estarán obligadas a entregar las tesis que se elaboren para la obtención de títulos académicos de grado y posgrado en formato digital para ser integradas al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

Elaboración: Investigadores.
Fuente: Ley Orgánica de Educación Superior.

ARTÍCULO DE LA CONSTITUCIÓN	**CONTEXTO**
ARTÍCULO 22	Establece: las personas tienen derecho a desarrollar su capacidad creativa, al ejercicio digno y sostenido de las actividades culturales y artísticas, y a beneficiarse de la protección de los derechos morales y patrimoniales que les correspondan por las producciones científicas, literarias o artísticas de su autoría.

ARTÍCULO 26	La educación es un derecho de las personas a lo largo de su vida y un deber ineludible e inexcusable del Estado. Constituye un área prioritaria de la política pública y de la inversión estatal, garantía de la igualdad e inclusión social y condición indispensable para el buen vivir.
ARTÍCULO 28	La educación responderá al interés público y no estará al servicio de intereses individuales y corporativos. Se garantizará el acceso universal, permanencia, movilidad y egreso sin discriminación alguna
ARTÍCULO 350	El sistema de educación superior tiene como finalidad la formación académica y profesional con visión científica y humanista; la investigación científica y tecnológica; la innovación, promoción, desarrollo y difusión de los saberes y las culturas; la construcción de soluciones para los problemas del país, en relación con los objetivos del régimen de desarrollo
ARTÍCULO 355 primer y segundo inciso	El Estado reconocerá a las universidades y escuelas politécnicas autonomía académica, administrativa, financiera y orgánica, acorde con los objetivos del régimen de desarrollo y los principios establecidos en la Constitución

Elaboración: Investigadores.
Fuente: Ley Orgánica de Educación Superior.

FACTIBILIDAD LEGAL. - Comprende la viabilidad legal del proyecto, es decir, conocer los alcances y limitaciones relacionadas con el desarrollo del mismo.

- La viabilidad legal busca principalmente determinar la existencia de alguna restricción legal en la realización de un proyecto.
- Se busca determinar la existencia de normas o regulaciones legales que impidan la ejecución u operación del proyecto.
- Promover el desarrollo de proyectos sin problemas y dentro de las disposiciones legales.
- Pueden ser registrados y patentados.
- Este proyecto no transgrede ninguna norma, leyes o reglamentos establecidos en la Constitución del Ecuador ni en estamentos legales, por tanto, es factible su desarrollo y aplicación.

CODIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INVENCIÓN

Artículo 104.- Obras susceptibles de protección. - La protección reconocida por el presente Título recae sobre todas las obras literarias, artísticas y científicas, que sean originales y que puedan reproducirse o divulgarse por cualquier forma o medio conocido o por conocerse. 12.- SOFTWARE

Artículo 131.- Protección de software. - El software se protege como obra literaria. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea como código fuente; es decir, en forma legible por el ser humano; o como código objeto; es decir, en forma legible por máquina, ya sea sistemas operativos o sistemas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa. Se excluye de esta protección las formas estándar de desarrollo de software. En este sentido, los documentos y textos producidos en las Instituciones de Educación Superior desarrollados con el objeto de obtener sus grados académicos y/o trabajos de facultad, son autores intelectuales con el patrocinio de cada institución, por lo tanto, son acreedores a los derechos de protección intelectual dispuestos en la normativa vigente.

Elaboración: Investigadores.

Fuente: Constitución del Ecuador (2010).

Anexo 7. Validación de expertos.

No se pueden editar las respuestas

Cuestionario para validación de propuesta

Saludos cordiales, el presente cuestionario servirá para validar la propuesta de la tesis: **"Procesamiento en paralelo de algoritmo de agrupamiento dinámico de trayectorias GPS"**, tutorizada por el Ing. Gary Reyes, y realizada por los estudiantes Gabriel Raul Ligua Aristega y Luis Eduardo Vivas Mera. El cuestionario consta de 8 preguntas de opción múltiple que ayudarán a determinar la validez de la propuesta. Agradecemos su colaboración.

*Obligatorio

Correo *

christopher.crespol@ug.edu.ec

¿Es adecuada la implementación en una arquitectura en paralelo del algoritmo de agrupamiento Dyclee para procesar grandes volúmenes de datos de trayectorias GPS? *

- ○ Totalmente de acuerdo
- ◉ De acuerdo
- ○ Ni de acuerdo ni en desacuerdo
- ○ En desacuerdo
- ○ Totalmente en desacuerdo

¿Es adecuada la modificación realizada en el algoritmo de agrupamiento Dyclee para que este pueda agrupar en base a rangos de velocidades?

- ○ Totalmente de acuerdo
- ◉ De acuerdo
- ○ Ni de acuerdo ni en desacuerdo
- ○ En desacuerdo
- ○ Totalmente en desacuerdo

¿Cree usted que la presente investigación cumple con los objetivos propuestos?

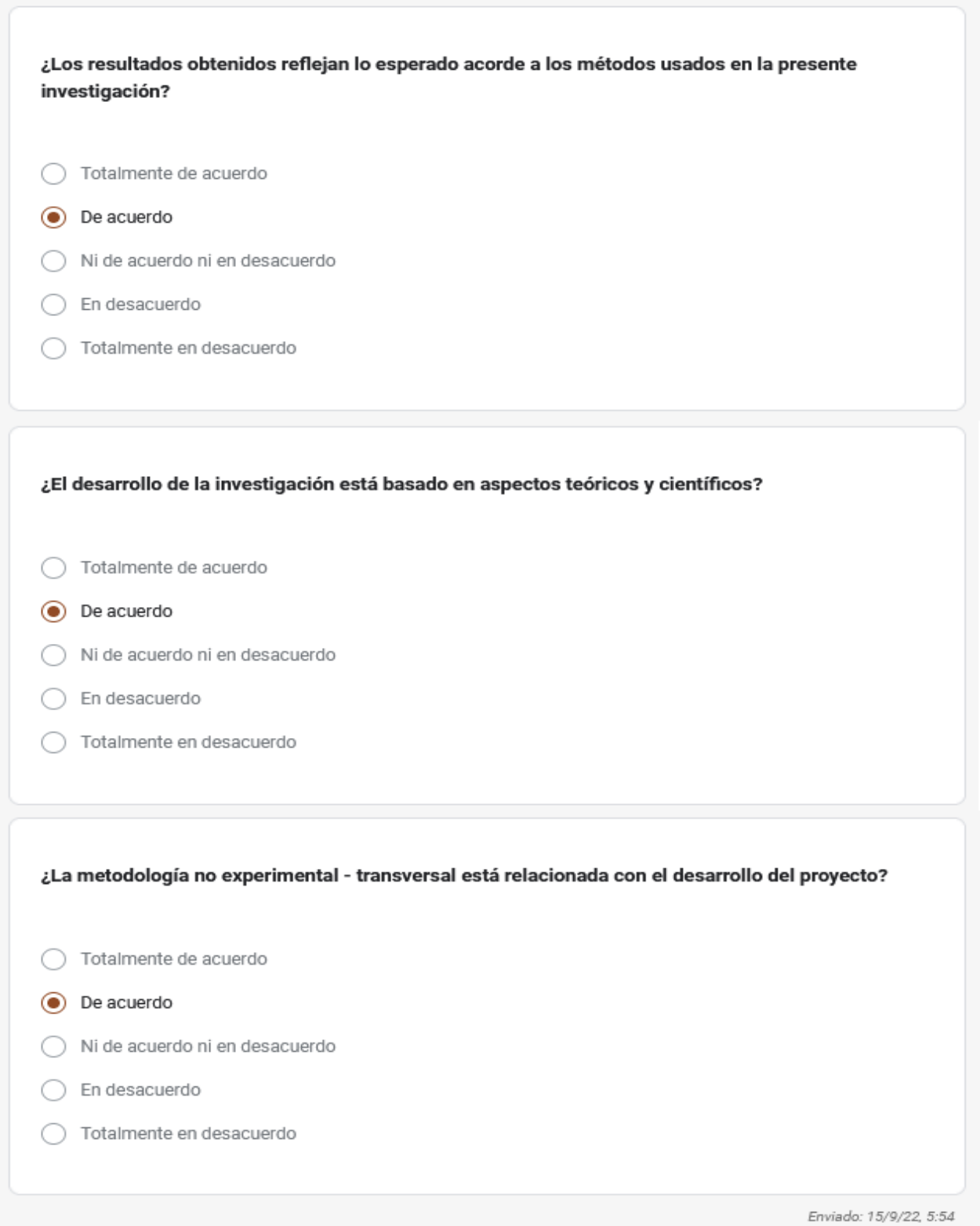
- ○ Totalmente de acuerdo
- ◉ De acuerdo
- ○ Ni de acuerdo ni en desacuerdo
- ○ En desacuerdo
- ○ Totalmente en desacuerdo

¿Considera usted que la presente investigación es acorde con los métodos tecnológicos actuales y aportan soluciones a temáticas de la actualidad?

- ○ Totalmente de acuerdo
- ◉ De acuerdo
- ○ Ni de acuerdo ni en desacuerdo
- ○ En desacuerdo
- ○ Totalmente en desacuerdo

¿Los métodos experimentales usados son acordes a los resultados esperados en la presente investigación?

- ○ Totalmente de acuerdo
- ◉ De acuerdo
- ○ Ni de acuerdo ni en desacuerdo
- ○ En desacuerdo
- ○ Totalmente en desacuerdo



¿Los resultados obtenidos reflejan lo esperado acorde a los métodos usados en la presente investigación?

- ○ Totalmente de acuerdo
- ◉ De acuerdo
- ○ Ni de acuerdo ni en desacuerdo
- ○ En desacuerdo
- ○ Totalmente en desacuerdo

¿El desarrollo de la investigación está basado en aspectos teóricos y científicos?

- ○ Totalmente de acuerdo
- ◉ De acuerdo
- ○ Ni de acuerdo ni en desacuerdo
- ○ En desacuerdo
- ○ Totalmente en desacuerdo

¿La metodología no experimental - transversal está relacionada con el desarrollo del proyecto?

- ○ Totalmente de acuerdo
- ◉ De acuerdo
- ○ Ni de acuerdo ni en desacuerdo
- ○ En desacuerdo
- ○ Totalmente en desacuerdo

Enviado: 15/9/22, 5:54

Elaboración: Investigadores.

Fuente: Propia.

CONSTANCIA DE JUICIO DE EXPERTO

Estimado Ingeniero

Gary Xavier Reyes Zambrano, Mgs.

DOCENTE TUTOR DEL TRABAJO DE TITULACIÓN

Ciudad. -

El presente instrumento certifica que se realizó la revisión del proyecto de titulación "PROCESAMIENTO EN PARALELO DE ALGORITMO DE AGRUPAMIENTO DINÁMICO DE TRAYECTORIAS GPS" cuyos criterios e indicadores empleados permitieron articular el trabajo según se muestra en el Anexo 7, por tanto, Gabriel Raul Ligua Aristega y Luis Eduardo Vivas Mera estudiantes no titulados de la Carrera de Ingeniería en Sistemas computacionales de la Universidad de Guayaquil, pueden continuar con el proceso de titulación en vista que no existen observaciones.

Por lo actuado en el Anexo 7, se procede a validar el trabajo de titulación.

Sin otro particular.

Firmado electrónicamente por:
CHRISTOPHER GABRIEL CRESPO LEON

Ing. Christopher Crespo León, Msg.
C.I. N° 0919211649

Elaboración: Investigadores.

Fuente: Propia.

No se pueden editar las respuestas

Cuestionario para validación de propuesta

Saludos cordiales, el presente cuestionario servirá para validar la propuesta de la tesis: **"Procesamiento en paralelo de algoritmo de agrupamiento dinámico de trayectorias GPS"**, tutorizada por el Ing. Gary Reyes, y realizada por los estudiantes Gabriel Raul Ligua Aristega y Luis Eduardo Vivas Mera. El cuestionario consta de 8 preguntas de opción múltiple que ayudarán a determinar la validez de la propuesta. Agradecemos su colaboración.

*Obligatorio

Correo *

jimmy_ism1@hotmail.com

¿Es adecuada la implementación en una arquitectura en paralelo del algoritmo de agrupamiento Dyclee para procesar grandes volúmenes de datos de trayectorias GPS? *

- (●) Totalmente de acuerdo
- ( ) De acuerdo
- ( ) Ni de acuerdo ni en desacuerdo
- ( ) En desacuerdo
- ( ) Totalmente en desacuerdo

¿Es adecuada la modificación realizada en el algoritmo de agrupamiento Dyclee para que este pueda agrupar en base a rangos de velocidades?

- (●) Totalmente de acuerdo
- ( ) De acuerdo
- ( ) Ni de acuerdo ni en desacuerdo
- ( ) En desacuerdo
- ( ) Totalmente en desacuerdo

¿Cree usted que la presente investigación cumple con los objetivos propuestos?

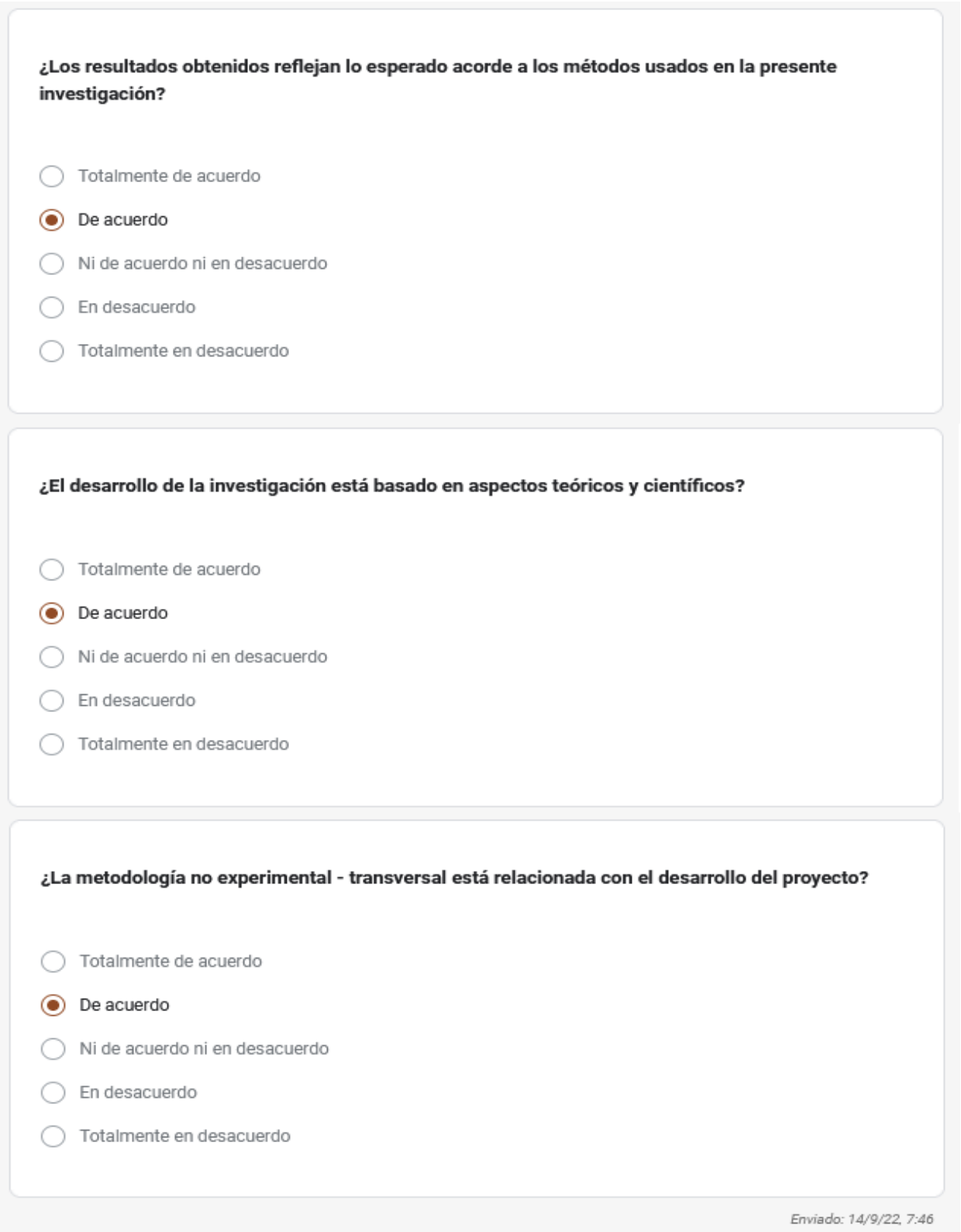
- ○ Totalmente de acuerdo
- ◉ De acuerdo
- ○ Ni de acuerdo ni en desacuerdo
- ○ En desacuerdo
- ○ Totalmente en desacuerdo

¿Considera usted que la presente investigación es acorde con los métodos tecnológicos actuales y aportan soluciones a temáticas de la actualidad?

- ○ Totalmente de acuerdo
- ◉ De acuerdo
- ○ Ni de acuerdo ni en desacuerdo
- ○ En desacuerdo
- ○ Totalmente en desacuerdo

¿Los métodos experimentales usados son acordes a los resultados esperados en la presente investigación?

- ○ Totalmente de acuerdo
- ◉ De acuerdo
- ○ Ni de acuerdo ni en desacuerdo
- ○ En desacuerdo
- ○ Totalmente en desacuerdo



Elaboración: Investigadores.

Fuente: Propia.

CONSTANCIA DE JUICIO DE EXPERTO

Estimado Ingeniero

Gary Xavier Reyes Zambrano, Mgs.

DOCENTE TUTOR DEL TRABAJO DE TITULACIÓN

Ciudad. -

El presente instrumento certifica que se realizó la revisión del proyecto de titulación "PROCESAMIENTO EN PARALELO DE ALGORITMO DE AGRUPAMIENTO DINÁMICO DE TRAYECTORIAS GPS" cuyos criterios e indicadores empleados permitieron articular el trabajo según se muestra en el Anexo 7, por tanto, Gabriel Raul Ligua Aristega y Luis Eduardo Vivas Mera estudiantes no titulados de la Carrera de Ingeniería en Sistemas computacionales de la Universidad de Guayaquil, pueden continuar con el proceso de titulación en vista que no existen observaciones.

Por lo actuado en el Anexo 7, se procede a validar el trabajo de titulación.

Sin otro particular.

Firmado electrónicamente por:
JIMMY IGNACIO
SORNOZA MOREIRA

Ing. Jimmy Sornoza Moreira
C.I. N° 0920433760

Elaboración: Investigadores.

Fuente: Propia.

No se pueden editar las respuestas

Cuestionario para validación de propuesta

Saludos cordiales, el presente cuestionario servirá para validar la propuesta de la tesis: **"Procesamiento en paralelo de algoritmo de agrupamiento dinámico de trayectorias GPS"**, tutorizada por el Ing. Gary Reyes, y realizada por los estudiantes Gabriel Raul Ligua Aristega y Luis Eduardo Vivas Mera. El cuestionario consta de 8 preguntas de opción múltiple que ayudarán a determinar la validez de la propuesta. Agradecemos su colaboración.

*Obligatorio

Correo *

apolinariooscar@gmail.com

¿Es adecuada la implementación en una arquitectura en paralelo del algoritmo de agrupamiento Dyclee para procesar grandes volúmenes de datos de trayectorias GPS? *

- (●) Totalmente de acuerdo
- ( ) De acuerdo
- ( ) Ni de acuerdo ni en desacuerdo
- ( ) En desacuerdo
- ( ) Totalmente en desacuerdo

¿Es adecuada la modificación realizada en el algoritmo de agrupamiento Dyclee para que este pueda agrupar en base a rangos de velocidades?

- (●) Totalmente de acuerdo
- ( ) De acuerdo
- ( ) Ni de acuerdo ni en desacuerdo
- ( ) En desacuerdo
- ( ) Totalmente en desacuerdo

¿Cree usted que la presente investigación cumple con los objetivos propuestos?

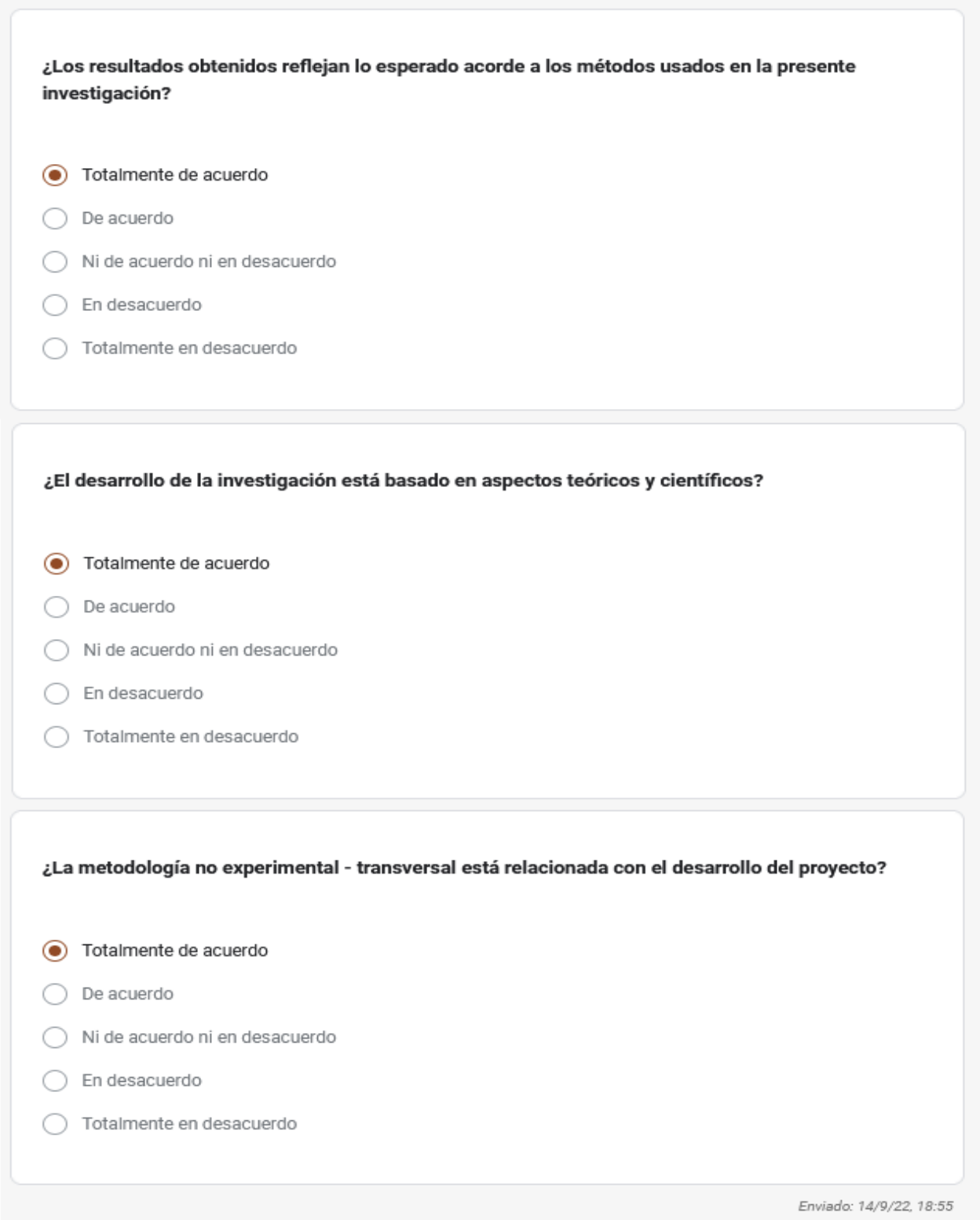
- (●) Totalmente de acuerdo
- ( ) De acuerdo
- ( ) Ni de acuerdo ni en desacuerdo
- ( ) En desacuerdo
- ( ) Totalmente en desacuerdo

¿Considera usted que la presente investigación es acorde con los métodos tecnológicos actuales y aportan soluciones a temáticas de la actualidad?

- (●) Totalmente de acuerdo
- ( ) De acuerdo
- ( ) Ni de acuerdo ni en desacuerdo
- ( ) En desacuerdo
- ( ) Totalmente en desacuerdo

¿Los métodos experimentales usados son acordes a los resultados esperados en la presente investigación?

- (●) Totalmente de acuerdo
- ( ) De acuerdo
- ( ) Ni de acuerdo ni en desacuerdo
- ( ) En desacuerdo
- ( ) Totalmente en desacuerdo



¿Los resultados obtenidos reflejan lo esperado acorde a los métodos usados en la presente investigación?

- (●) Totalmente de acuerdo
- ( ) De acuerdo
- ( ) Ni de acuerdo ni en desacuerdo
- ( ) En desacuerdo
- ( ) Totalmente en desacuerdo

¿El desarrollo de la investigación está basado en aspectos teóricos y científicos?

- (●) Totalmente de acuerdo
- ( ) De acuerdo
- ( ) Ni de acuerdo ni en desacuerdo
- ( ) En desacuerdo
- ( ) Totalmente en desacuerdo

¿La metodología no experimental - transversal está relacionada con el desarrollo del proyecto?

- (●) Totalmente de acuerdo
- ( ) De acuerdo
- ( ) Ni de acuerdo ni en desacuerdo
- ( ) En desacuerdo
- ( ) Totalmente en desacuerdo

Enviado: 14/9/22, 18:55

Elaboración: Investigadores.

Fuente: Propia.

CONSTANCIA DE JUICIO DE EXPERTO

Estimado Ingeniero
Gary Xavier Reyes Zambrano, Mgs.
DOCENTE TUTOR DEL TRABAJO DE TITULACIÓN
Ciudad. -

El presente instrumento certifica que se realizó la revisión del proyecto de titulación "PROCESAMIENTO EN PARALELO DE ALGORITMO DE AGRUPAMIENTO DINÁMICO DE TRAYECTORIAS GPS" cuyos criterios e indicadores empleados permitieron articular el trabajo según se muestra en el Anexo 7, por tanto, Gabriel Raul Ligua Aristega y Luis Eduardo Vivas Mera estudiantes no titulados de la Carrera de Ingeniería en Sistemas computacionales de la Universidad de Guayaquil, pueden continuar con el proceso de titulación en vista que no existen observaciones.

Por lo actuado en el Anexo 7, se procede a validar el trabajo de titulación.

Sin otro particular.

Firmado electrónicamente por:
OSCAR OMAR APOLINARIO ARZUBE

PHD. Oscar Omar Apolinario Arzube
C.I. N° 0911390441

Anexo 15. Artículo científico

Procesamiento en paralelo de algoritmo de agrupamiento dinámico de trayectorias GPS

Gabriel Raul Ligua Aristega [1], Luis Eduardo Vivas Mera [2]

gabriel.liguaa@ug.edu.ec, luis.vivasm@ug.edu.ec

[1] Universidad de Guayaquil, Pascuales Coop. En Pie de Lucha Mz.87 Sl.11, 090702, Guayaquil, Ecuador.

[2] Universidad de Guayaquil, Urdesa Norte Av. 1ra y calle 5ta, 090507, Guayaquil, Ecuador.

DOI: 10.17013/risti.n.pi-pf

Resumen: Este trabajo se enfoca en migrar un algoritmo de clustering dinámico a una arquitectura en paralelo. Se utilizan datasets de dos ciudades Guayaquil y Roma. Se realiza la implementación del algoritmo Dyclee con ciertas modificaciones para agrupar en base a rangos de velocidades, con lo cual se efectúan experimentos en los que se calculó el tiempo de procesamiento de datos del algoritmo y la cantidad de grupos que formó con sus respectivas velocidades promedio. La validación de la investigación consistió en tres experimentos, el primero determinó que el algoritmo Dyclee en paralelo optimiza el tiempo de procesamiento de los datos. El segundo experimento muestra la cantidad de grupos formados con sus velocidades promedio y el tercero determina la optimización del tiempo en cada ciclo de ejecución en paralelo con respecto al secuencial. Se concluyó que el algoritmo Dyclee bajo una arquitectura en paralelo mejora los tiempos de procesamiento de datos.

Palabras-clave: Paralelo; Clustering; Trayectorias; Densidad; Procesamiento.

Parallel processing of a GPS trajectory dynamic clustering algorithm

Abstract: The present work focuses on migrating a dynamic clustering algorithm to a parallel architecture. Datasets from two cities, Guayaquil and Rome, are used. The implementation of the Dyclee algorithm is carried out with certain modifications to group based on speed ranges, then experiments are carried out in which the data processing time was calculated and the number of groups formed with their respective average speeds. The research is largely bibliographical, however, it also incorporates field research qualities,

as the algorithm was run on GPS track data from two different datasets. The validation of the research consisted of three experiments, the first determined that the parallel Dyclee algorithm optimizes the data processing time. The second experiment shows the number of groups formed with their average speeds and the third determines the optimization of the time in each cycle of the parallel execution with respect to the sequential one. It was concluded that the Dyclee algorithm under a parallel architecture improves data processing times.

Keywords: Parallel; Clustering; Trajectories; Density; Processing.

1. Introducción

La evolución de la tecnología ha provocado avances significativos en muchas áreas de estudio, que lleva consigo la generación de cantidades inimaginables de datos de diversas procedencias, a cada instante; a esto se lo conoce como Big Data. Todos estos datos se producen de diferentes formas y se encuentran almacenados en distintos repositorios según el contexto en el que se requieran. La importancia de los datos crece constantemente puesto que son utilizados para realizar diversos estudios, como trabajos de investigación cuyo fin consiste en observar fenómenos y obtener conclusiones acertadas apoyándose de datos procesados o generando nuevos datos a través de la experimentación.

Todo estudio necesita procesar grandes cantidades de volúmenes de datos para obtener resultados óptimos que contribuyan a la investigación, según la temática objeto de estudio que se esté abordando. El análisis de datos espaciales generados por diversas tecnologías de localización usadas en la actualidad son de vital importancia para la elaboración de planes para el desarrollo urbano (Reyes, Lanzarini, Hasperue, et al., 2021). En estudios referentes a trayectorias vehiculares, congestionamiento vehicular, flujo de tránsito vehicular u otros estudios que involucren el tránsito; requieren datos de vehículos esencialmente, que por lo general son las trayectorias que siguen los vehículos para cumplir con ciertos recorridos. La mayoría de estos datos se registran de forma instantánea mediante el GPS de los teléfonos inteligentes, o mediante el sistema de navegación que poseen los autos modernos (Gutiérrez et al., 2016). Los sistemas de transporte inteligente también procesan grandes cantidades de datos de trayectorias GPS que generan los vehículos en la calle en tiempo real. Los datos obtenidos se analizan y se usan para la toma de decisiones (Reyes et al., 2020).Considerando la cantidad de datos vehiculares que se registran en estos dispositivos todos los días en todo el mundo, se puede estimar que las cifras son infinitas. Se utilizan diferentes técnicas para el procesamiento de estos datos en grupos comunes o de características similares, una de ellas es mediante la implementación de algoritmos de agrupación de inteligencia

artificial, sin embargo, pueden tardar mucho más tiempo en procesar los datos dependiendo de la cantidad y características de estos.

Si se pudiese implementar una arquitectura en estos algoritmos que permita el procesamiento en paralelo de grandes volúmenes de datos de trayectorias GPS, atraería ventajas en cuanto al tiempo de procesamiento de datos para un posterior análisis y beneficios para la comunidad científica que dedican tiempo a estudios referentes al flujo de tránsito vehicular; o incluso, para cualquier tipo de estudio donde se requieran procesar muchos datos, es por ello que se ha considerado desarrollar el siguiente proyecto.

Como se establece en (Reyes, Lanzarini, Estrebou, et al., 2021), los métodos de agrupamiento de trayectorias GPS ayudan a identificar patrones en grandes cantidades de datos. Sin embargo, el tiempo de procesamiento podría disminuir si se considera la implementación en paralelo de estos métodos o técnicas de agrupamiento de datos de trayectorias GPS. Por lo que el presente proyecto se centra especialmente en la implementación de una arquitectura en paralelo de un algoritmo de clustering, para optimizar el tiempo de procesamiento en la agrupación de los datos. La investigación consistirá de una revisión bibliográfica con la finalidad de tener un soporte teórico para efectuar la implementación del algoritmo de agrupamiento en una arquitectura en paralelo; para ello es, es importante considerar ciertas características, como el diseño de la arquitectura, el lenguaje de programación, librerías y sintaxis adecuadas para la implementación en paralelo, entre otros. Además, de realizar ciertas adaptaciones en el algoritmo para que pueda agrupar datos de trayectorias GPS y efectuar las experimentaciones con datasets públicos.

Como trabajos previos, se encuentra el estudio de (Lapeira et al., 2017), que consistió en la implementación de una versión con arquitectura en paralelo del algoritmo FuzzyPred, basada en la cantidad de datos que cada hilo de procesamiento puede procesar de forma simultánea e independiente. Los resultados obtenidos demostraron que el algoritmo en paralelo puede ser 10 veces más rápido con respecto al secuencial y por ello consideran que puede ser efectivo ante bases de datos muy grandes. El estudio presentado por (Zhang et al., 2013), que trata de una versión en paralelo del algoritmo K-means haciendo uso de un modelo de programación de paso de mensajes llamado MPI (Message Passing Interface), evidenció que este algoritmo demuestra estabilidad y portabilidad, además, de emplear poco tiempo de procesamiento en grandes volúmenes de conjuntos de datos.

Este artículo está organizado de la siguiente manera: la sección 2 muestra el marco teórico del presente estudio, la sección 3 describe el diseño de la investigación, la sección 4 presenta los resultados obtenidos, la sección 5 presenta la discusión de los resultados y limitaciones, finalmente la sección 6 contiene las conclusiones de este estudio.

2. Marco teórico

A continuación, se muestran definiciones y conceptos relacionados con el tema objeto de estudio del presente trabajo de investigación.

2.1. GPS

Según indica (S. Wang et al., 2020) el "GPS es un sistema de radionavegación basada en satélites que puede proporcionar geolocalización en tiempo real e información horaria a un receptor GPS en cualquier parte de la Tierra".

Una trayectoria GPS se presenta como una secuencia discreta de puntos de coordenadas geográficas (Reyes Zambrano, 2019).

2.2. Trayectorias GPS

De acuerdo con (Bian et al., 2020) "una trayectoria GPS es una secuencia de puntos GPS que registra la trayectoria espacial de un objeto en movimiento".

Cada punto de trayectoria de cualquier objeto en movimiento representa una ubicación con marca de tiempo y está modelado por cuatro elementos importantes (x, y, s, c) donde a_x es la longitud, a_y es la latitud, a_s es la marca de tiempo y a_c es la celda a la que está asignada el punto a. Tanto latitud como longitud se expresan mediante números reales, la marca de tiempo muestra su exactitud en segundos y la celda se reconoce mediante un identificador (ID) en número entero (J. Wang et al., 2021).

2.3. Algoritmo de clustering

Según (Reyes Zambrano et al., 2022) "las técnicas de clustering han sido utilizadas en el análisis de trayectorias desde hace varios años. Por lo general, se trata de adaptaciones de los algoritmos convencionales utilizando métricas de similitud especialmente diseñadas para trayectorias".

Según (Pagola et al., 2015) "los algoritmos de clustering son una herramienta eficaz para extraer información de datos en bruto. Son métodos de aprendizaje no supervisado. El objetivo de los algoritmos de clustering es dividir los datos en clústeres o grupos".

2.4. Algoritmos de clustering basado en densidad

Los algoritmos de clustering basados en densidad establecen grupos como zonas densas de puntos, separadas por otras zonas densas. La agrupación espacial estima el parecido de acuerdo a las características espaciales de los datos y, por lo tanto, en

vez de hablar de la semejanza entre dos objetos, se hace referencia a la proximidad en el espacio de dos objetos (Dib Ashur et al., 2016).

2.5. Dyclee

Según (Barbosa Roa et al., 2019) Dyclee es: "un algoritmo basado en distancia y densidad que presenta varias propiedades, como el manejo de agrupaciones no convexas y de densidad múltiple con rechazo de valores atípicos, y logra ser completamente dinámico".

Dyclee fue desarrollado en base al paradigma de aprendizaje no supervisado de manera incremental, es decir, simulando la forma en que aprenden los seres humanos (sobre la marcha).

Dyclee es un algoritmo de agrupación dinámico que estudia entornos en evolución, es decir, cuando trabaja sobre un grupo de datos, este no asume una estructura de datos predefinida, sino que la busca de forma progresiva a medida que se ingresan dichos datos cambiando la estructura del agrupamiento (Barbosa Roa et al., 2019).

La primera etapa de Dyclee opera la tasa del flujo de datos y crea micro clústeres juntando muestras de datos que se encuentran cercanos según la norma L1, también llamada distancia de Manhattan. La segunda etapa de Dyclee funciona en base a una frecuencia más baja y analiza la distribución de los micro clústeres. La densidad de un micro clúster puede ser baja, media o alta y se utiliza para crear los clústeres finales con un enfoque basado en densidad (Barbosa Roa et al., 2019).

2.6. Paralelismo en Python

Python utiliza hilos para lograr paralelismo mediante el manejo de memoria compartida. Consiste en subtareas que se crean en un proceso y a su vez comparten memoria. Sin embargo, Python en su diseño posee un mecanismo denominado Global Interpreter Lock (GIL) que solo permite que se ejecute un proceso a la vez independientemente del número de núcleos que tenga el computador (Acervo Lima, 2022).

Las restricciones de GIL pueden superarse mediante el uso de procesos, sin embargo, hay que tener en consideración que la comunicación entre procesos es menos eficiente en este caso y es necesario hacer uso de técnicas que permitan una comunicación más efectiva (Acervo Lima, 2022).

2.7. Paralelismo basado en procesos

Según indica (Python Software Foundation, 2021) en su documentación oficial, "multiprocessing es un paquete que permite crear procesos utilizando una API similar al módulo threading. El paquete multiprocessing ofrece concurrencia tanto

local como remota, esquivando el Global Interpreter Lock mediante el uso de procesos en lugar de hilos (threads)".

Para crear un proceso en Python, haciendo uso del módulo multiprocessing, es necesario invocar a la clase Process, crear un objeto tipo Process y llamar a su método start(), además, es muy importante que los procesos sean declarados después de la siguiente cláusula: if__name__=='__main__' (Python Software Foundation, 2021).

Además, hay que tener en consideración la comunicación entre procesos, el módulo multiprocessing admite dos tipos de canales para la comunicación entre procesos; uno de ellos es mediante el uso de colas (queues) para pasar mensajes de ida y vuelta, y el otro es implementando tuberías (Pipes) para la comunicación entre dos procesos (Python Software Foundation, 2021).

3. Materiales y métodos

De acuerdo con (Agudelo et al., 2008) la investigación no experimental se efectúa sin manipular las variables de estudio, solo se observa el fenómeno tal y como suceden en su contexto natural, para su posterior análisis.

En este estudio se utiliza un tipo de investigación no experimental, puesto que es adecuado para cumplir con los objetivos que se plantearon inicialmente, además, no se realiza manipulación directa de las variables de estudio, simplemente se adapta el algoritmo para trabajar bajo un contexto específico que es objeto de estudio en este proyecto. El diseño de esta investigación es transversal debido a que las experimentaciones se efectuarán una única vez en un periodo de tiempo específico.

3.1. Metodología

La investigación realizó tres experimentos. El primer experimento consistió en la ejecución del algoritmo de forma secuencial y en paralelo, en un ambiente local con las siguientes características: sistema operativo Windows 11 de 64 bits, procesador Intel ® Core™ i7-6700k, RAM de 16.0 GB. Se realizan varias ejecuciones del algoritmo Dyclee (secuencia y paralelo) con diferentes intervalos de tiempo (5 minutos, 3 minutos, 2 minutos y 1 minuto). Se registran los tiempos de procesamiento totales para realizar la comparativa entre el algoritmo secuencial y paralelo y determinar si existe una mejoría en cuanto al tiempo de procesamiento.

El segundo experimento evalúa los resultados de los micro clusters formados después de efectuar la agrupación de datos en base a las velocidades de los vehículos. Se determina la cantidad de grupos formados y cantidad de ciclos realizados en cada ejecución, tanto en secuencia como en paralelo. Además, se realiza el cálculo de velocidades promedios por intervalo de tiempo y velocidades

promedios totales de todas las ejecuciones para evaluar si existen diferencias significativas en los resultados.

El tercer experimento evalúa un indicador de tiempos promedios por ciclo del algoritmo Dyclee en secuencia y en paralelo. Se registran los tiempos por ciclo y se obtiene el tiempo promedio por ciclo; esto, con la finalidad de comparar los resultados en secuencia y en paralelo y evidenciar la optimización del tiempo de procesamiento por ciclo.

Se realiza una experimentación adicional que consiste en la ejecución del algoritmo en un ambiente en la nube que presenta las siguientes características: almacenamiento de 32 GB, memoria de 16 GB, 4 vCore. Esto con el fin de realizar una comparativa del algoritmo en secuencia y en paralelo para evaluar cambios favorables en los tiempos de procesamiento.

A continuación, se detalla la metodología empleada en el presente estudio:

Revisión de la literatura

Se consulta en fuentes bibliográficas los diferentes paradigmas de programación en paralelo y lenguajes de programación que soportan esta característica. Se consulta acerca de las características del algoritmo Dyclee, además, se consulta acerca de temas relacionados a trayectorias GPS para identificar qué elementos serán utilizados para agrupar rangos de velocidades. Finalmente, se consulta acerca de las diferentes plataformas de computación en la nube para elegir la más apropiada para el desarrollo del presente estudio.

Migración del algoritmo de agrupamiento dinámico Dyclee

Se hace uso del lenguaje de programación Python en conjunto con la herramienta editora de código fuente Visual Studio Code para efectuar la migración del algoritmo Dyclee y sus respectivas adaptaciones para que este pueda agrupar en base a rangos de velocidades. La figura 1 muestra la sección de código donde se realiza la adaptación del código para agrupar por rangos de velocidades.

```
for ind, p in buffer.iterrows():

    point = [p.latitud, p.longitud]
    dyclee.trainOnElement(point, p)

currMicroClusters = dyclee.getClusteringResult()
```

(a)

```
for ind, p in buffer.iterrows():

    point = [p.velocidad, p.velocidad]
    dyclee.trainOnElement(point, p)

dyclee.getClusteringResult()
```

(b)

Figura 1 – (a) Código Dyclee para agrupación en base a latitud y longitud. (b) Código Dyclee adaptado para agrupación en base a la velocidad.

Selección de datos de trayectorias GPS

Uso de datasets que contengan datos referentes a trayectorias GPS, es decir, latitud, longitud, marca de tiempo, velocidad del vehículo, este último es indispensable para realizar las agrupaciones en base a la velocidad. Los datasets seleccionados provienen de repositorios públicos contienen datos recopilados de ciudades como Guayaquil y Roma.

Diseño e implementación de una arquitectura en paralelo del algoritmo Dyclee

Inicialmente, se identifica la posible implementación de tres procesos; el primero comprender la recepción de datos, el segundo proceso realiza la agrupación de datos mediante las funciones propias del algoritmo Dyclee, finalmente, el tercer proceso muestra los grupos de datos con sus respectivas velocidades promedios como etiqueta de los micro clusters. Los grupos densos se identifican con diferentes colores y los grupos atípicos se identifican con el color negro y con un signo menos (-) al inicio de la etiqueta del grupo.

Además, se hace uso del paquete "multiprocessing" que proporciona Python; la clase "Process" y la invocación del método start() permite la creación de los tres procesos antes mencionados. Es importante codificar la siguiente cláusula: if__name__=='__main__' antes de declarar los procesos para evitar errores.

Para la comunicación entre estos procesos se hace uso de las colas o "queues", para esto es necesario usar la clase "Queue" propia del paquete "multiprocessing".

Implementación de una plataforma de computación en la nube

La plataforma de computación en la nube que se usa es Amazon Web Services (AWS), que mediante su servicio de EMR (Elastic Map Reduce), permite utilizar computadoras o nodos virtuales para el procesamiento de datos.

Para el uso de EMR es necesario adicionar otro servicio llamado S3, que permite usar un disco duro virtual para guardar información en la nube de AWS y de esta forma procesar los datos. En este caso se ha crea un bucket (contenedor para objetos almacenados en S3) para guardar los scripts y demás archivos del algoritmo Dyclee.

Experimentación

Se prueba el algoritmo Dyclee, codificado de forma secuencial y bajo una arquitectura en paralelo, en dos contextos; de forma local en un computador y en una plataforma en la nube. Se utilizan datasets previamente seleccionados que contienen datos de trayectorias vehiculares. La ejecución del algoritmo permite obtener una serie de resultados relacionados con el tiempo que tarda el algoritmo en procesar los datos, tanto de forma secuencial y en paralelo; además, de obtener los grupos formados después del procesamiento.

Análisis y validación de los resultados

Se utiliza la estadísticas descriptiva para obtener resultados pertinentes que contribuyan al análisis de este estudio.

3.2. Población

La población del estudio consistirá en un conjunto de datos de trayectorias GPS que serán utilizadas para realizar las experimentaciones. Estos datos se encuentran almacenados en dos datasets correspondientes a las ciudades de Guayaquil y Roma. En la tabla 1 se puede visualizar a detalle, la cantidad de registros que contiene cada dataset.

Tabla 1 – Cantidad de registros de datasets seleccionados

Población	Cantidad de registros
Guayaquil	30557
Roma	34118

4. Resultados

En esta sección se detallan los resultados de los experimentos realizados sobre los 2 datasets escogidos. El primer experimento comprende los resultados de los tiempos totales de ejecución (secuencial y paralelo) del algoritmo Dyclee para los intervalos de tiempo de cinco minutos, tres minutos, dos minutos y un minuto. El segundo experimento abarca los resultados de las velocidades promedio de cada grupo que

se genera mediante el algoritmo Dyclee (secuencial y paralelo) para los mismos intervalos de tiempo mencionados anteriormente. El tercer experimento proporciona resultados acerca del tiempo promedio por ciclo del algoritmo Dyclee en secuencia y en paralelo. El experimento adicional comprende resultados del tiempo total de ejecución del algoritmo en un ambiente de computación en la nube.

4.1. Primera experimentación

En la tabla 2 y tabla 3 se muestran los resultados referentes a tiempos totales de ejecución de los experimentos realizados sobre el dataset de Guayaquil y Roma respectivamente, con 4 intervalos de tiempo (en minutos) diferentes. Con respecto a la tabla 2, la primera columna muestra los intervalos de tiempo usados para las diferentes ejecuciones, la segunda columna presenta el tiempo total en segundos del algoritmo Dyclee en secuencia, siendo 202,62 el tiempo más alto de todas las ejecuciones en secuencia. La tercera columna muestra el tiempo total en segundos del algoritmo Dyclee en paralelo, siendo 86,72 el tiempo más alto de todas las ejecuciones en paralelo. Esto demuestra que a menor intervalo tiempo, el algoritmo tarda más tiempo en agrupar los datos. Sin embargo, el tiempo promedio total de las ejecuciones en secuencia que es 105,08 es superior al tiempo promedio total en paralelo, este valor es 86,72; comprobando que existe una optimización de tiempos totales de procesamiento de datos en paralelo. Por último, la desviación estándar del tiempo total en secuencia es 69,38 y posee un valor superior comparado con la calculada con los tiempos totales en paralelo, que es 59,03. La tabla 3 muestra los resultados obtenidos del dataset de Roma, se puede visualizar que los resultados se comportan de forma similar que lo expuesto anteriormente. A menor intervalo de tiempo, mayor es el tiempo de ejecución del algoritmo; el tiempo más alto en secuencia es 166,89 segundos y en paralelo es 141,24 segundos. Los tiempos promedios totales en secuencia y en paralelo siendo 87,71 segundos y 74,22 segundos respectivamente, indican que se mantiene la optimización del tiempo con el algoritmo adaptado a una arquitectura en paralelo. La desviación estándar del tiempo total en secuencia es 56,37 y posee un valor superior comparado con la calculada con los tiempos totales en paralelo, que es 48,07.

Tabla 2 – Comparación de tiempos de totales de ejecución del algoritmo en secuencia y en paralelo para dataset de Guayaquil

Intervalo	Tiempo total secuencial	Tiempo total paralelo
5	44,23	35,95
3	70,15	56,99
2	103,33	83,64
1	202,62	170,29

Tiempo total promedio	105,08	86,72
Desviación estándar	69,38	59,03

Tabla 3 – Comparación de tiempos de totales de ejecución del algoritmo en secuencia y en paralelo para dataset de Roma

Intervalo	Tiempo total secuencial	Tiempo total paralelo
5	37,53	30,8
3	60,46	50,66
2	85,96	74,16
1	166,89	141,24
Tiempo total promedio	87,71	74,22
Desviación estándar	56,37	48,07

4.2. Segunda Experimentación

En la tabla 4 y 5 se detallan los resultados referentes a las velocidades promedios y grupos formados después de realizar la agrupación de los datos con los dataset de Guayaquil y Roma. La tabla 4 muestra la cantidad de grupos formados con el dataset de Guayaquil, en total se formaron 4 grupos (2 grupos atípicos y 2 grupos densos). El rengo de velocidad promedio por grupo más alta se visualiza en el grupo (-1) (grupo atípico) cuyo valor es de 48,58 Km/h y la velocidad promedio más baja se muestra en el grupo denso (1) con un valor de 1,82 Km/h. Además, se muestra que en cada intervalo de tiempo se formaron la misma cantidad de grupos con valores promedios similares de velocidad. Por último se muestran las desviaciones estándar de velocidades por grupo, siendo 0,004 la más baja y 0,652 la más alta; el resultado de desviación estándar con respecto a la media muestra un resultado de 0,277 lo que indica que existe cohesión en las velocidades dentro de un mismo grupo. La tabla 5 muestra la cantidad de grupos formados con el dataset de Roma, se formaron 7 grupos (3 grupos atípicos y 4 grupos densos) para los intervalos de tiempo de 5 minutos, 3 minutos y 2 minutos; para 1 minutos se formaron 6 grupos (3 grupos atípicos y 3 grupos densos) lo que indica que hubo una reagrupación de datos en este último caso. El rango de velocidad promedio por grupo más alto se visualiza en el grupo (-2) cuyo valor promedio es de 79,19 Km/h y la velocidad promedio más baja se muestra en el grupo (1) con un valor de 0,94 Km/h. Las desviaciones estándar de velocidades por grupo, siendo 0,015 la más baja y 9,616 la más alta; el resultado de desviación estándar con respecto a la media

muestra un resultado de 3,490 lo que indica que existe cohesión en las velocidades dentro de un mismo grupo.

Tabla 4 – Velocidades promedio por grupo generado del dataset de Guayaquil

Intervalo	Grupo -2	Grupo -1	Grupo 1	Grupo 2	Cantidad de grupos
5	35,305	48,405	1,823	22,145	4
3	36,512	48,565	1,823	21,880	4
2	36,693	48,831	1,815	21,483	4
1	36,599	48,547	1,823	21,901	4
Velocidad promedio por grupo	36,277	48,587	1,821	21,852	
Desviación estándar	0,652	0,177	0,004	0,273	
Promedio Desviación estándar	0,277				

Tabla 5 – Velocidades promedio por grupo generado del dataset de Roma

Intervalo	Grupo -3	Grupo -2	Grupo -1	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Cantidad de grupos
5	70,667	84,000	57,383	0,937	15,909	29,987	44,385	7
3	70,667	84,000	55,947	0,937	15,944	30,343	45,387	7
2	73,000	84,000	62,424	0,938	16,062	30,852	46,494	7
1	84,000	64,767	46,882	0,968	16,470	31,444		6
Velocidad promedio por grupo	74,583	79,192	55,659	0,945	16,096	30,657	45,422	
Desviación estándar	6,373	9,616	6,477	0,015	0,257	0,633	1,054	
Promedio Desviación estándar	3,490							

4.3. Tercera Experimentación

En la tabla 6 y 7 se muestran los resultados de los tiempos promedios por ciclo en secuencia y en paralelo para el dataset de Guayaquil y Roma respectivamente. En la primera columna se muestra el intervalo de tiempo empleado en cada ejecución del algoritmo, la segunda columna muestra el tiempo promedio por ciclo, siendo 1,65 segundos el tiempo más alto en secuencia y el tiempo más bajo es 1,52 segundos. La tercera columna muestra el tiempo promedio por ciclo en segundos del algoritmo Dyclee en paralelo, siendo 1,33 segundos el tiempo más alto de todas las ejecuciones en paralelo. Al contrario de lo que sucedía con los tiempos totales, los tiempos promedio por ciclo tienden a aumentar con mayor intervalo de tiempo. Comparando tiempos promedios totales por ciclo, se comprueba que el tiempo en paralelo sigue siendo menor con respecto al tiempo en secuencia, siendo 1,28 segundos para el algoritmo en paralelo y 1,58 para el algoritmo en secuencia. Por último, la desviación estándar del tiempo promedio por ciclo en secuencia es 0,063 y posee un valor superior comparado con la calculada con los tiempos promedio por ciclo en paralelo, que es 0,045. La tabla 7 muestra que no necesariamente disminuye el tiempo promedio por ciclo por intervalo bajo de tiempo, estos datos son más irregulares; es decir, tienden a incrementar y disminuir independientemente del intervalo de tiempo. Finalmente, se demuestra que el promedio total por ciclo en paralelo es menor que en secuencia, siendo 1,43 y 1,71 segundos respectivamente.

Tabla 6 – Comparación de tiempos promedio por ciclo del algoritmo en secuencia y en paralelo para dataset de Guayaquil

Intervalo	Cantidad de ciclos	Tiempo promedio por ciclo (secuencia)	Tiempo promedio por ciclo (paralelo)
5	23	1,65	1,33
3	39	1,61	1,29
2	58	1,53	1,27
1	115	1,52	1,23
Tiempo total promedio		1,58	1,28
Desviación estándar		0,063	0,045

Tabla 7 – Comparación de tiempos promedio por ciclo del algoritmo en secuencia y en paralelo para dataset de Roma

Intervalo	Cantidad de ciclos	Tiempo promedio por ciclo (secuencia)	Tiempo promedio por ciclo (paralelo)
5	24	1,70	1,49
3	40	1,75	1,44
2	60	1,71	1,39

1	120	1,67	1,41
Tiempo total promedio		1,71	1,43
Desviación estándar		0,0308	0,0445

4.4. Experimentación en Amazon Web Services

En la tabla 8 se observan los resultados de los tiempos totales del experimento haciendo uso de Amazon Web Services, con el Dataset de Guayaquil. La primera columna muestra los intervalos de tiempo que utiliza el algoritmo para procesar los datos. La siguiente columna muestra los tiempos totales de ejecución, siendo 318,16 segundos el más alto con el algoritmo en secuencia para un intervalo de tiempo de 1 minuto y el más bajo de 68,81 segundos. La tercera columna muestra los tiempos totales de ejecución, siendo 358,68 segundos el más alto con el algoritmo en paralelo y el más bajo de 75,66 segundos. Los tiempos promedios totales muestran que el algoritmo en secuencia tarda menos en procesar los datos puesto que posee un tiempo de 165,71 segundos con respecto al tiempo en paralelo que es de 184,40 segundo; esto debido a que los datos en la nube no están distribuidos en diferentes nodos y como consecuencia el trabajo se recarga sobre un solo nodo y para ello el algoritmo debe adaptarse a una versión que emplee la abstracción Map-reduce.

Tabla 8 – Comparación de tiempos de totales de ejecución del algoritmo en secuencia y en paralelo para dataset de Guayaquil con Amazon Web Services

Intervalo	Tiempo total secuencial	Tiempo total paralelo
5	44,23	35,95
3	70,15	56,99
2	103,33	83,64
1	202,62	170,29
Tiempo total promedio	105,08	86,72
Desviación estándar	69,38	59,03

5. Discusión

Los resultados mostrados en la primera experimentación demuestran que el algoritmo Dyclee, migrado a una arquitectura en paralelo, procesa los datos en menos tiempo que el algoritmo en secuencia; esto evidencia que trabajar bajo una arquitectura en paralelo optimiza los tiempos de procesamiento de grandes volúmenes de datos de trayectorias GPS.

Los resultados de la segunda experimentación permiten identificar la velocidad promedio y cantidad de grupos formados por el algoritmo Dyclee. Las agrupaciones formadas en base a rango de velocidades poseen valores muy similares en cada intervalo de tiempo, lo que indica que no existen alteraciones considerables en los grupos; a excepción del experimento con el dataset de Roma que agrupa de forma diferente, únicamente, en la ejecución de un minuto.

Los resultados mostrados en la tercera experimentación demuestran que el algoritmo Dyclee en paralelo procesa los datos de forma más rápida con respecto al algoritmo en secuencia; esto se ha comprobado antes en la primera experimentación, sin embargo, también se lo analiza en base a tiempos promedios por ciclos por si existe alguna diferencia.

Los resultados obtenidos en Amazon Web Services, por el contrario, muestra resultados no esperados; puesto que pese a ejecutar el mismo algoritmo migrado a paralelo, no muestra resultados favorables en cuanto a optimización del tiempo.

Una de las limitaciones encontradas en el presente estudio es que a pesar de utilizar un ambiente en paralelo en la nube, no se emplea la abstracción Map-reduce puesto que presenta otras consideraciones, sin embargo, si se realizan las configuraciones necesarias para ejecutar el código en la nube.

Otra limitación es que la selección de conjuntos de datos de repositorios públicos puede ser restringida en ciertos casos y dificultar su accesibilidad; además, pueden presentar gran cantidad de ruido.

Por último, una limitación a considerar es que existen datasets con datos incompletos que dificulta el cumplimiento de ciertos objetivos, como la agrupación en base a rangos de velocidad.

6. Conclusiones

En este estudio se realizaron tres experimentos mediante el uso de dos datasets diferentes (Guayaquil y Roma). El primero, para hacer una comparación del comportamiento de los tiempos totales de ejecución del algoritmo Dyclee en secuencia y en paralelo. El segundo con el fin de identificar las diferentes velocidades y grupos que se generan después del procesamiento de datos en base a rangos de velocidades y el tercero para hacer una comparación del comportamiento de los tiempos promedios por ciclo del algoritmo en secuencia y en paralelo.

Los resultados del primer experimento demostraron que la ejecución del algoritmo en paralelo generaba tiempos totales de ejecución menores a los de la ejecución del algoritmo en secuencia en ambos datasets, es decir, que se evidenció la optimización del tiempo en una arquitectura en paralelo.

En el segundo experimento se comprobó que los grupos generados por el algoritmo Dyclee en el dataset de Guayaquil eran constantes en todos los intervalos de tiempo de procesamiento, la desviación estándar en cada grupo indica que no existe una gran variación en las velocidades dentro de un mismo grupo, mientras que por intervalos las velocidades de varían mucho más. En el dataset de Roma se pudo notar diferencias en la generación de los grupos ya que no fue constante en todos los intervalos, puesto que el intervalo de tiempo de procesamiento de 1 minuto generó 1 grupo menos que todos los otros, alterando un poco el agrupamiento de las velocidades por grupo y por intervalo. Al comparar los resultados del algoritmo en secuencia con respecto a paralelo se puede comprobar que en ambos casos se generan los mismos grupos y velocidades, lo que demuestra que estos son independientes de la manera como se procesen los datos.

La tercera experimentación demostró que la optimización del tiempo en la arquitectura en paralelo persiste, incluso, si solo se analiza el tiempo que tarde el algoritmo en cumplir un ciclo de ejecución; puesto que los resultados en tiempo promedio por ciclo en paralelo se mantienen bajos con respecto a los tiempos por ciclo en secuencia.

Referencias

Acervo Lima. (2022). Procesamiento paralelo en Python. https://es.acervolima.com/procesamiento-paralelo-en-python/#google_vignette

Agudelo, G., Aigneren, M., & Ruiz, J. (2008). Diseños De Investigación Experimental Y No-Experimental. In Centro de Estudios de Opinión (pp. 1–46).http://bibliotecadigital.udea.edu.co/dspace/bitstream/10495/2622/1/AgudeloGabriel_disenosinvestigacionexperimental.pdf

Barbosa Roa, N., Travé-Massuyès, L., & Grisales-Palacio, V. H. (2019). DyClee: Dynamic clustering for tracking evolving environments. Pattern Recognition, 94, 162–186. https://doi.org/10.1016/j.patcog.2019.05.024

Bian, W., Cui, G., & Wang, X. (2020). A trajectory collaboration based map matching approach for low-sampling-rate GPS trajectories. Sensors, 20(7), 1–22. https://doi.org/10.3390/s20072057

Dib Ashur, J., Vallón, J., Martínez, C., & Said, C. (2016). SACO: Un algoritmo de clustering espacial con hormigas inteligentes. In Simposio Argentino de Inteligencia Artificial (ASAI), 17–24.

Gutiérrez, J., García, J., & Salas, M. (2016). Big (Geo)Data en Ciencias Sociales: Retos y Oportunidades. Revista de Estudios Andaluces (RAE), 33(1), 1–23. https://doi.org/10.12795/rea.2016.i33

Lapeira, O., Ceruto, T., Rosete, A., & Díaz, H. (2017). Algoritmo paralelo para la obtención de predicados difusos. Revista Cubana de Ciencias Informáticas, 11(2), 117–133. https://www.redalyc.org/articulo.oa?id=378350964009

Pagola, M., De Miguel, L., Marco, C., & Bustince, H. (2015). Algoritmo de clustering intervalo-valorado difuso. Actas de La XVI Conferencia CAEPIA, 1–10. http://giara.unavarra.es/

Python Software Foundation. (2021). multiprocessing — Paralelismo basado en procesos. 3.10.0rc2 Documentation. https://docs.python.org/es/3/library/multiprocessing.html

Reyes, G., Lanzarini, L., Estrebou, C., & Maquilón, V. (2021). Vehicular Flow Analysis Using Clusters. XXVII CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACION (CACIC)(Modalidad Virtual, 4 Al 8 de Octubre de 2021), 261–270. http://sedici.unlp.edu.ar/handle/10915/130341

Reyes, G., Lanzarini, L., Hasperue, W., & Bariviera, A. (2021). A proposal for a pivot-based vehicle trajectory clustering method. Transportation Research Record, 20(10), 1–20. https://doi.org/DOI: 10.1177/ToBeAssigned

Reyes, G., Lanzarini, L., Hasperué, W., & Bariviera, A. F. (2020). GPS trajectory clustering method for decision making on intelligent transportation systems. Journal of Intelligent and Fuzzy Systems, 38(5), 5529–5535. https://doi.org/10.3233/JIFS-179644

Reyes Zambrano, G. (2019). GPS trajectory compression algorithm. Communications in Computer and Information Science, 959, 57–69. https://doi.org/10.1007/978-3-030-12018-4_5

Reyes Zambrano, G., Córdova Rizo, F., León Granizo, O., & Carabali Noriega, E. (2022). GPS Trajectory segmentation and clustering method. 1, 1–8.

Wang, J., Wu, N., Lu, X., Zhao, W. X., & Feng, K. (2021). Deep Trajectory Recovery with Fine-Grained Calibration using Kalman Filter. IEEE Transactions on Knowledge and Data Engineering, 33(3), 1–14. https://doi.org/10.1109/TKDE.2019.2940950

Wang, S., Ding, S., & Xiong, L. (2020). A new system for surveillance and digital contact tracing for COVID-19: Spatiotemporal reporting over network and GPS. JMIR MHealth and UHealth, 8(6). https://doi.org/10.2196/19457

Zhang, J., Wu, G., Hu, X., Li, S., & Hao, S. (2013). A Parallel Clustering Algorithm with MPI – MKmeans. Journal of Computers, 8(1), 10–17. https://doi.org/10.4304/jcp.8.1.10-17