

Monografía Inteligencia Artificial: Sistemas peer to peer

SISTEMAS Y COMPUTACIÓN
MAYO DE 2020

1 CONTENIDO

CONTENIDO	1
PRESENTACIÓN	3
MARCO TEÓRICO	3
3.1 ¿Qué es un sistema peer to peer?	3
3.2 ¿En qué circunstancias se implementa un sistema Peer to Peer?	4
3.3 Ventajas y desventajas de los sistemas peer to peer	4
3.4 El anonimato de las redes peer-to-peer	5
3.5 Los problemas y huecos legales de los sistemas peer-to-peer	5
3.6 El origen y nacimiento de los sistemas peer to peer	6
3.7 Tipos de redes peer to peer	7
3.8 Aplicaciones peer to peer vigentes	8
3.9 Peer to peer y el procesamiento de la información	8
3.10 Sistemas peer to peer y la inteligencia artificial	9
OBJETIVOS	10
General	10
Específicos	10
DISEÑO APLICACIÓN	11
5.1 El modelo k-medias	11
5.2 Propuesta del aplicativo	12
5.3 Ejercicio 1: Ordenamiento de Clusters	12
5.4 Ejercicio 2: Ordenamiento y reducción de paquetes de información	15
CONCLUSIONES	18
BIBLIOGRAFÍA	19
FIRMA DEL DOCUMENTO	20

2 PRESENTACIÓN

Las redes peer-to-peer, en su forma más pura, se crean cuando dos o más PC's se conectan y comparten recursos sin necesidad de que los datos pasen por un servidor de computadora centralizado.

Pero las redes peer-to-peer también son utilizadas por las criptomonedas e incluso tienen el potencial de resolver algunos de los problemas más desafiantes al recopilar y analizar grandes cantidades de datos, dice Ben Hartman CEO de Bet Capital LLC, a Entrepreneur.

Nano Vision, una startup que premia a los usuarios con criptomonedas a cambio de sus datos moleculares, tiene como objetivo cambiar la forma en que abordamos las amenazas hacia la salud humana, como las superbacterias, las enfermedades infecciosas, el cáncer, entre otras.

Otro jugador que utiliza redes peer to peer e IA es Presearch, un motor de búsqueda descentralizado impulsado por una comunidad y recompensas a sus miembros con tokens para crear un sistema de búsqueda más transparente [1].

AUTOR

Sebastián Franco Gómez

1.225.092.429

3 MARCO TEÓRICO

3.1 ¿QUÉ ES UN SISTEMA PEER TO PEER?

Un sistema peer to peer se considera un sistema en el cual dos o más computadores se comunican y comparten información. Se diferencia principalmente de la arquitectura cliente servidor ya que todos los equipos actúan como cliente y como servidor, entendiendo un cliente como un usuario que consume un servicio y un servidor como quien lo provee [2, 3]

3.2 ¿EN QUÉ CIRCUNSTANCIAS SE IMPLEMENTA UN SISTEMA PEER TO PEER?

Los sistemas peer to peer son muy útiles a la hora de compartir o procesar información entre un conjunto de equipos sin necesidad de un servidor principal que guarde y sincronice a los equipos. Esto es especialmente útil cuando se necesita dividir la los procesos de un algoritmo gigante o cuando se necesitan dividir paquetes de información en secciones diferentes [4]

3.3 VENTAJAS Y DESVENTAJAS DE LOS SISTEMAS PEER TO PEER

Los sistemas peer to peer presentan las siguientes ventajas en comparación a la arquitectura cliente servidor clásico:

- No requiere de un sistema operativo de red propio.
- No requiere de servidores caros que centralicen y recuerden las conexiones dentro de una red ya que los equipos se encargan de configurar sus conexiones.
- Es mucho más sencilla de implementar y desplegar que una red cliente servidor.
- Los usuarios de la red pueden configurar sus entradas y salidas.
- Si un equipo de la red cae la red no se ve en peligro.

De esta manera se abstrae que los sistemas peer to peer son ideales cuando los equipos conocen bien su tarea y la divulgación de sus partes es óptimas, sin embargo, los sistemas peer to peer también presentan los siguientes inconvenientes:

- El desempeño de la red puede variar según los actores del sistema.

- Los archivos e información no puede tener respaldos y copias de seguridad con facilidad.
- Si un equipo no puede localizar a un miembro específico de la red, su información es inalcanzable.
- La seguridad de la información de la red es responsabilidad de cada usuario de la red
- Si la seguridad no se establece bien se puede saturar una o más partes del sistema.

De esta manera se considera que el manejo de esta red debe ser cuidadoso y bajo la responsabilidad de cada uno de los partícipes. Una red demasiado abierta se puede volver insegura, ineficaz e incluso infructuosa en comparación a una arquitectura clásica [5]

3.4 EL ANONIMATO DE LAS REDES PEER-TO-PEER

Los sistemas peer to peer permiten una característica que no se puede catalogar como ventaja o desventaja y es el anonimato dentro de las redes, ya que, más allá del conocimiento de la IP pública del actor de la red, no se ejerce un agente central regulador el cual verifique las identidades o los datos de los miembros del sistema. Por esta razón cualquier persona puede hacer parte de los sistemas peer-to-peer sin riesgo a ser identificados en la realidad, esto permite el libre intercambio de información y archivos como música, libros, películas y demás contenido que debe ser centralizado y de pago para ser legal, además de contenido ilícito como pornografía infantil. El anonimato de las redes peer-to-peer complica el rastreo y seguimiento de los aportadores de esta información que además puede ser replicada con total facilidad, ya que si un usuario da de baja un archivo otro puede subirlo y mantener la integridad de la información. Los sistemas peer to peer son incontrolables por esta razón y esto los hace un tipo de sistema muy persistente en el tiempo [6].

3.5 LOS PROBLEMAS Y HUECOS LEGALES DE LOS SISTEMAS PEER-TO-PEER

Los sistemas peer to peer se encuentran en constante amenaza por parte de los gobiernos y entes privados ya que se puede compartir información susceptible a derechos de autor sin necesidad de pagar por su divulgación mediante la divulgación abierta en estos sistemas que suelen ser anónimos, por lo que en múltiples ocasiones las plataformas peer to peer grandes suelen ser demandadas para dar de baja su estructura principal, pero incluso bajo estas condiciones, son incontrolables a niveles atómicos, ya que una red peer

to peer no es explícitamente formal, con que 2 o más equipos tengan conexiones directas, se puede considerar peer to peer.

El dilema con la legalidad de los archivos transportados en los sistemas peer to peer consta en la definición de piratería digital, donde se define como el uso ilegal del material protegido del derecho de autor, sin embargo, descargar los bits lógicos de este no es considerado delito, por lo que, hasta que no se reproduzca este archivo no es explícitamente ilegal. Este vacío jurídico ha sido explotado a través del mundo entero y hoy en día una gran cantidad de países y uniones han tenido que cambiar la jurisdicción para ser más duros con la piratería digital [7].

3.6 EL ORIGEN Y NACIMIENTO DE LOS SISTEMAS PEER TO PEER

Definir el origen de los sistemas peer to peer es muy complejo gracias a lo difuso que es separar este paradigma de la computación distribuida en general, además de considerarse peer to peer la conexión entre 2 o más partes, por lo que experimentos desde los años 60 podrían ser considerados la implementación primigenia. Por facilidad, se habla entonces del nacimiento “cultural” de los sistemas peer to peer mediante la primera aparición comercial importante de estos sistemas.

En la cultura superficial, se considera a Napster (la actual plataforma de música competencia directa de Spotify, Youtube Music, Amazon Prime Music, entre otras) como el padre de las plataformas peer to peer, sin embargo, antes de la aparición mundial de Amazon en 1999, se presenta otro partícipe de la historia. En 1996, Adam Hinkley, programador australiano, desarrolla para Mac OS una plataforma llamada Hotline Connect, una plataforma orientada a empresas, universidades y al sector empresarial en general que sirviera para la transmisión directa de archivos, sin embargo, los usuarios del común empezaron a encontrar utilidad en esta plataforma y dieron lugar a su uso para compartir todo tipo de archivos. El problema con Hotline Connect era que presentaba una arquitectura totalmente centralizada, por lo que las partes debían conectarse directamente entre ellas, esto causaba que al momento de descargar un archivo medianamente grande, la conexión entre ambas partes siempre tuviera que estar activa, en caso de perderse por un momento, el paquete de datos llegaba incompleto o corrupto y la descarga debía iniciarse de nuevo desde el inicio. Por esta razón el sistema se volvió obsoleto en cuestión de años y apareció Napster en 1999 por Shawn Fanning que planteó un nuevo tipo de alternativa. Napster se presenta originalmente como una plataforma de difusión gratuita de música libre para todos los usuarios, siendo estos los que subían la música y Napster quien les decía al resto de miembros donde encontrarla. Este modelo, aunque fuera un sistema peer to peer, presentaba una naturaleza centralizada, ya que

Napster era el directo responsable de conectar a los usuarios de la plataforma. Durante dos años Napster se convirtió en la plataforma virtual más recurrida del mundo con más de 80 millones de usuarios. En el año 2001, múltiples artistas de marca internacional como Dr. Dre, Madonna y Metallica demandaron a Fanning y su plataforma por copyright pero logró escapar de estas denuncias, sin embargo, las demandas por parte del estado hicieron que diera de baja su idea original y planteara el modelo de negocio actual de pago [8, 9].

3.7 TIPOS DE REDES PEER TO PEER

Las redes peer to peer se pueden clasificar según varias métricas, como su estructuración, su tipo de generación y su nivel de anonimato, sin embargo, para entender las redes peer to peer según su arquitectura, se definen según su grado de centralización.

Las redes peer to peer se definen entonces de la siguiente manera:

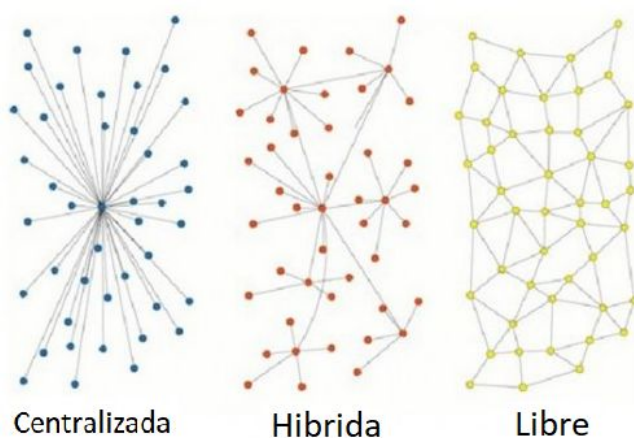


Imagen 1: Tipos de redes peer to peer por su centralización

La anterior imagen corresponde a los tipos de tipologías para sistemas peer to peer propuestos por Paul Baran [10]

Los tipos de redes se explican de la siguiente manera:

Redes centralizadas: Son redes peer to peer que son dependientes a un servidor principal que se encarga de direccionar las partes en el momento de hacer peticiones. Este servidor recuerda los usuarios y los archivos que poseen, de esta manera, si un nuevo usuario requiere un archivo, el servidor central se encarga de ser el puente entre ambas partes. La plataforma original de Napster usaba este modelo.

Redes híbridas: Son redes peer to peer que dividen el peso del sistema mediante el uso de múltiples servidores que hospedan la información de un grupo de usuarios. Los servidores pueden o no comunicarse entre ellos y su función principal es aliviar y volver el sistema más escalable y mantenible. Las redes Blockchain como Bitcoin usan este modelo.

Redes libres o puras: Son redes peer to peer totalmente independientes de servidores centralizados que hospeden o recuerden datos de usuarios. Estas redes suelen ser las aplicadas para transporte y difusión anónima de archivos. Su problema radica en la conexión entre dos partes que estén muy lejanas en la red o no se conozcan, siendo así necesaria una plataforma que indique donde se pueden obtener archivos específicos. La red Bittorrent usa este modelo.

3.8 APLICACIONES PEER TO PEER VIGENTES

Hoy en día las aplicaciones peer to peer han perdido fuerza gracias a las plataformas de streaming como Netflix y la franquicia Prime, puesto que una de las razones por las que se usaban estas plataformas era por la inaccesibilidad de cierto contenido multimedia en la red, situación resuelta por las anteriores, sin embargo, existen varias aplicaciones merecedoras de mención que marcaron un lugar en la historia gracias a su divulgación de la información y la cultura las cuales son:

- Napster
- Grokster
- Morpheus
- Streamcast
- Kazaa
- Ares

Además de las anteriores, existen plataformas peer to peer vigentes en la actualidad como:

- Plataformas con protocolos Blockchain (Bitcoin, Ethereum, NEM, Hyperchain, Hyperledger, Multichain, entre otras)
- Utorrent y Bittorrent
- The pirate bay
- Skype
- Snocap

- Garena
- GameRanger

3.9 PEER TO PEER Y EL PROCESAMIENTO DE LA INFORMACIÓN

Durante los apartados anteriores, se habla de las redes peer to peer como redes para compartir archivos e información en general, donde los usuarios pueden actuar tanto de clientes como de servidores, sin embargo, existen otros tipos de propuestas para las redes peer to peer y es su capacidad de compartir poder de cómputo para procesar un algoritmo en concreto. De esta manera se puede empezar a hablar de sistemas peer to peer para el procesamiento de la información mediante algoritmos paralelos [11].

Los algoritmos paralelos se usan solucionar problemas que impliquen procesamiento de grandes cantidades de datos, la implicación de estos hace parte del paradigma general de la computación distribuida, sin embargo, en ocasiones lo que sucede con los algoritmos paralelos es que un servidor central envía a los miembros de la red un grupo de datos a procesar y cuando estos solucionan el paquete, piden otro para seguir aportando al problema [12]

En las redes Blockchain, puntualmente en sus algoritmos de consenso, se usa computación paralela para solucionar las pruebas de trabajo y de importancia, haciendo trabajar a clusters enteros de equipos para hallar la solución al problema planteado [13]

En la criptografía en general, se usa de algoritmos como la prueba de conocimiento 0 para entender si una parte tiene información que otra necesita sin revelarlo. Se trabaja sobre algoritmos paralelos que una serie de contactos de confianza procesa para verificar si son verdad [14]

3.10 SISTEMAS PEER TO PEER Y LA INTELIGENCIA ARTIFICIAL

La inteligencia artificial ha sido definida en la historia de muchísimas maneras y clasificada en variadas categorías que han evolucionado a través del tiempo [15] sin embargo, para fines prácticos, se define Inteligencia Artificial o IA como la capacidad de simular el comportamiento inteligente del humano; esto se puede dar en términos de percepción, resolución de problemas u optimización de procesos. Un algoritmo de ordenamiento puede ser considerado inteligencia artificial, un sistema de reconocimiento de voz puede ser considerado inteligencia artificial, un procesador de imágenes puede ser considerado inteligencia artificial e incluso un algoritmo de clasificación de datos.

Los sistemas peer to peer entonces pueden ser utilizados en el ámbito de la inteligencia artificial en el contexto de la computación paralela, donde son las partes las que se entregan el problema y los paquetes de datos a resolver o procesar hasta que el problema tenga solución. A partir de este punto se expone la combinatoria de los sistemas peer to peer y la inteligencia artificial en el caso de estudio del algoritmo de clasificación de datos k-means o k-medias.

4 OBJETIVOS

4.1 GENERAL

Reconocer la historia, trayectoria, utilidad y aplicación de los sistemas peer to peer en la informática y especialmente en el campo de la inteligencia artificial

4.2 ESPECÍFICOS

No	Objetivo Específico
1	Exponer los orígenes y la historia de los sistemas peer to peer
2	Indagar la arquitectura de los sistemas peer to peer
3	Comparar las ventajas y desventajas de los sistemas peer to peer
4	Comprender las implementaciones de los sistemas peer to peer
5	Explicar el uso de sistemas peer to peer en el campo de la inteligencia artificial

5 DISEÑO APLICACIÓN

A partir de la conclusión del apartado del marco teórico, se expone la implementación de un sistema peer to peer basado en el algoritmo k-means [16] el cual se encarga de ordenar cluster de datos y procesarlos siendo así este la combinatoria entre los sistemas peer to peer y la inteligencia artificial para lograr un fin común.

5.1 EL MODELO K-MEDIAS

La clasificación automática de objetos o datos es uno de los objetivos del aprendizaje de máquina. Podemos considerar tres tipos de algoritmos:

Clasificación supervisada

Clasificación no supervisada

Clasificación semi-supervisada

K means es un algoritmo de clasificación no supervisado

K-means es un algoritmo que agrupa objetos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o cluster. Se suele usar la distancia cuadrática.

Los objetos se representan con vectores reales de d dimensiones $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ y el algoritmo *k-means* construye k grupos donde se minimiza la suma de distancias de los objetos, dentro de cada grupo $S = \{S_1, S_2, \dots, S_k\}$, a su centroide. El problema se puede formular de la siguiente forma:

$$\min_{\mathbf{S}} E(\boldsymbol{\mu}_i) = \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

donde \mathbf{S} es el conjunto de datos cuyos elementos son los objetos \mathbf{x}_j representados por

vectores, donde cada uno de sus elementos representa una característica o atributo. Tendremos k grupos o clusters con su correspondiente centroide μ_i .

En cada actualización de los centroides, desde el punto de vista matemático, imponemos la condición necesaria de extremo a la función $E(\mu_i)$ que, para la función cuadrática es

$$\frac{\partial E}{\partial \mu_i} = 0 \implies \mu_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j$$

y se toma el promedio de los elementos de cada grupo como nuevo centroide.

Las principales ventajas del método *k-means* son que es un método sencillo y rápido. Pero es necesario decidir el valor de k y el resultado final depende de la inicialización de los centroides. En principio no converge al mínimo global sino a un mínimo local.

5.2 PROPUESTA DEL APLICATIVO

Para la demostración del algoritmo K-means, se proponen 2 ejercicios, uno de ordenamiento de clusters y otro de ordenamiento y reducción de paquetes de información en una imagen, es decir, reducción de colores.

5.3 EJERCICIO 1: ORDENAMIENTO DE CLUSTERS

Para los siguientes apartados se van a usar los programas K1, K2, K3, K4 y K5 basados en la documentación de k-means obtenida de unioviado [17]. El código se encontrará disponible en el repositorio github.com/thefrancho

Como se explica anteriormente, el algoritmo K-means obtiene un grupo de datos incoloros y los clasifica según sus características, como datos en común o posición en un plano.

El programa k1.py genera estos grupos de datos en tres (3) espacios diferentes en un plano:

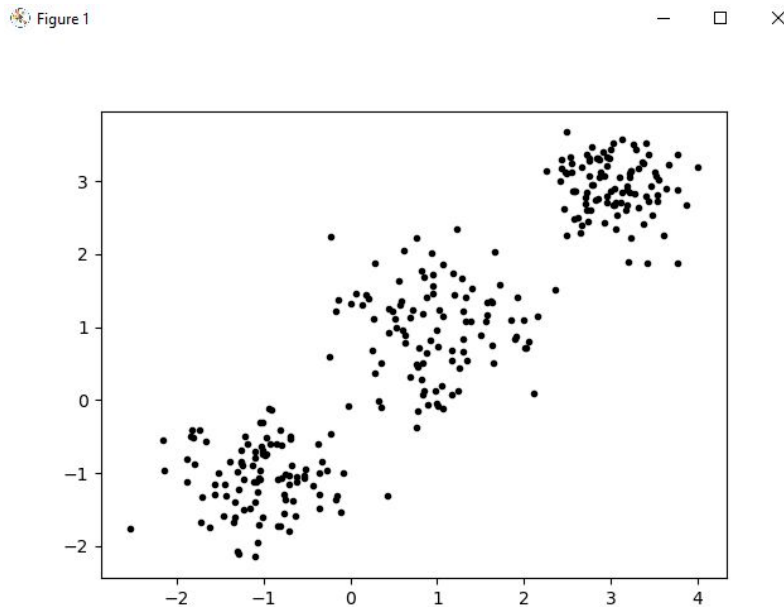


Imagen 2: Conjuntos de datos

Este gráfico se obtiene del siguiente código:

```

1
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from sklearn.cluster import KMeans
5
6 np.random.seed(7)
7
8 x1 = np.random.standard_normal((100,2))*0.6+np.ones((100,2))
9 x2 = np.random.standard_normal((100,2))*0.5-np.ones((100,2))
10 x3 = np.random.standard_normal((100,2))*0.4-2*np.ones((100,2))+5
11 X = np.concatenate((x1,x2,x3),axis=0)
12
13 plt.plot(X[:,0],X[:,1], 'k. ')
14 plt.show()

```

Imagen 3: Código de k1.py

Donde se importan las librerías numpy para procesamiento matemático, matplotlib para graficamiento de imágenes y sklearn para el procesamiento de clusters de información.

El código genera 3 rangos de información aleatorios en x1, x2 y x3 y los imprime en pantalla con la función plt.plot y plt.show.

Después de generar los cluster o conjuntos de información, se ejecuta el programa k2.py para que el algoritmo k-means ordene y agrupe los datos de la siguiente manera:

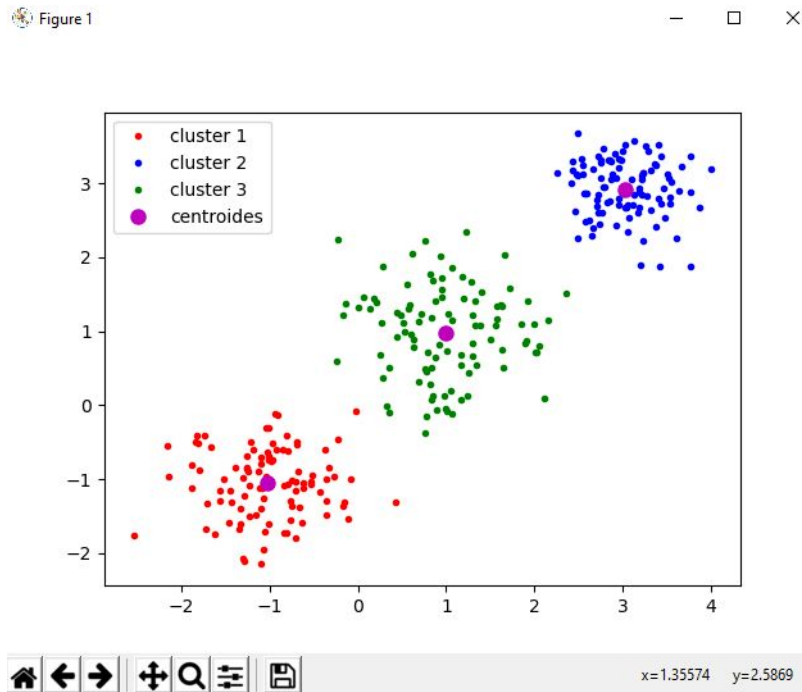


Imagen 4: Datos tratados con k-means

El algoritmo pinta y ordena los datos según 3 centroides en base a la explicación matemática anterior, de esta manera cualquier conjunto de datos puede ser ordenado con capacidad suficiente.


```

2  import numpy as np
3  import matplotlib.pyplot as plt
4  from sklearn.cluster import KMeans
5
6  np.random.seed(7)
7
8  x1 = np.random.standard_normal((100,2))*0.6+np.ones((100,2))
9  x2 = np.random.standard_normal((100,2))*0.5-np.ones((100,2))
10 x3 = np.random.standard_normal((100,2))*0.4-2*np.ones((100,2))+5
11 X = np.concatenate((x1,x2,x3),axis=0)
12
13 n = 3
14 k_means = KMeans(n_clusters=n)
15 k_means.fit(X)
16 centroides = k_means.cluster_centers_
17 etiquetas = k_means.labels_
18
19 plt.plot(X[etiquetas==0,0],X[etiquetas==0,1], 'r.', label='cluster 1')
20 plt.plot(X[etiquetas==1,0],X[etiquetas==1,1], 'b.', label='cluster 2')
21 plt.plot(X[etiquetas==2,0],X[etiquetas==2,1], 'g.', label='cluster 3')
22
23 plt.plot(centroides[:,0],centroides[:,1], 'mo', markersize=8, label='centroides')
24
25 plt.legend(loc='best')
26 plt.show()

```

Imagen 5: Código k2.py

Donde se usan las librerías explicadas anteriormente, se genera un valor aleatorio y 3 cluster de datos en posiciones aleatorias. Se define n como la cantidad de clusters que se van a procesar. Se definen la variable k_means como un objeto tipo KMeans con n clusters y se usa la función .fit para darle forma a esta variable. Definimos los centroides y las etiquetas e imprimimos en pantalla con los métodos de la librería matplotlib.

5.4 EJERCICIO 2: ORDENAMIENTO Y REDUCCIÓN DE PAQUETES DE INFORMACIÓN

Después de conocer la base gráfica y funcionamiento del algoritmo k-means, ahora se trabaja con procesamiento de imágenes para la reducción de colores de un archivo seleccionado.

Se presentan entonces el código de k4.py el cual selecciona una imagen desde una ubicación y la presenta:

Figure 1



Imagen 6: Imagen “Tienda” sin procesar

Donde matplotlib abre la imagen de PIL.

El código es el siguiente:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from PIL import Image
4
5
6 I = Image.open("tienda.jpg")
7 I1 = np.asarray(I,dtype=np.float32)/255
8 plt.figure(figsize=(12,12))
9 plt.imshow(I1)
10 plt.axis('off')
11 plt.show()
```

Imagen 7: Código de K4.py

Se presentan las librerías numpy y matplotlib explicadas anteriormente y se presenta la librería PIL que funciona para el procesamiento y manejo de archivos de imágenes (jpg y png)

Después de abrir la imagen base, se aplica sobre ella el algoritmo de k-medias para la reducción de colores en la imagen y reordenamiento de los nuevos colores seleccionados.

Se define entonces la cantidad de colores a reducir en la imagen y se aplica el algoritmo k-medias para el tratamiento de la misma, el resultado va a ser una nueva imagen con menos colores que la presentada anteriormente y con un peso menor.

Se usa la imagen de los platos llamada “tienda” mostrada anteriormente y se genera una nueva imagen llamada “tienda-comprimida” mostrada a continuación:

Figure 1



Imagen 8: Imagen “Tienda comprimida” procesada

A nivel visual, esta imagen no expone una gran diferencia en términos visuales, sin embargo al momento de comparar los colores que existían antes y ahora se presenta la siguiente diferencia:

```
C:\Users\villa\Desktop\utp\IA\k-medias>python k5.py
Número de pixels = 342240
Número de colores = 172389
```

Imagen 9: Pixeles y colores de “Tienda”

```
Número de pixels = 342240
Número de colores = 60
```

Imagen 10: Pixeles y colores de “Tienda_comprimida”

La razón por la que el cambio de colores se da de 172389 a 60 es porque generamos 60 clusters para los colores, 1 por cada color que queríamos agrupar, de esta manera obtenemos los 60 colores más iterativos en la imagen y rellenamos el resto de colores parecidos con estos.

No solo se reduce la cantidad de colores de la imagen sino el peso en kb de la misma



 tienda	31/05/2020 11:24 p. m.	Archivo JPG	752 KB
 tienda_comprimida	14/06/2020 7:09 p. m.	Archivo JPG	115 KB

Imagen 11: Pesos de tienda y tienda_comprimida

Se reduce el peso de la imagen en cuestión a menos de 1/5 del peso original.

De esta manera se puede usar el algoritmo K-means para el ordenamiento y reducción de paquetes de información.

5.5 K-Means y peer to peer

Los anteriores ejemplos nos muestran cómo con k clusters se puede ordenar y procesar conjuntos de información, de esta manera se podría generar un entorno en el cual, para grandes cantidades de datos, cada equipo dentro de una red pueda tomar uno o más clusters.

Los sistemas peer to peer son implementados para solucionar mediante computación paralela problemas de k-means y en general para cualquier problema de procesamiento de grandes paquetes de información

6 CONCLUSIONES

Los sistemas peer to peer tienen características y cualidades que los hacen útiles si están orientados correctamente, el concepto de conexión directa entre dos o más equipos lleva a la optimización de ciertos procesos frente a redes con servidores centrales.

A partir de la siguiente monografía se pueden obtener las siguientes conclusiones:

- Los sistemas peer to peer son una alternativa para procesos de computación distribuidas
- Los sistemas peer to peer presentan características específicas en términos de seguridad, escalabilidad y especialmente anonimato, por lo que son ideales para redes de divulgación de la información
- Los sistemas peer to peer se proponen como alternativas a la implementación de computación paralela
- Existen grandes sistemas peer to peer como plataformas Blockchain, Bittorrent y centros de procesamiento de datos públicos
- El algoritmo K-means puede ser implementado mediante protocolos peer to peer para optimizar el tiempo entre procesamiento de paquetes de datos


7 BIBLIOGRAFÍA

- [1] V. Gilberto. Alpha cero y diapositivas fin de curso. Grupo de investigación ADA. Diapositivas de presentación de elección de proyecto de fin de curso. Diapositivas 80 y 81. Mayo del 2020. Clase de inteligencia Artificial.
- [2] TechTerms. P2P Definition. Obtenido de <https://techterms.com/definition/p2p>
- [3] C. James. Computerworld. What's a Peer-to-Peer /P2P) Network? Obtenido de <https://www.computerworld.com/article/2588287/networking-peer-to-peer-network.html>
- [4] N. Codrut. Digital Citizen. What are P2P (peer-to-peer) networks and what are they used for? Obtenido de <https://www.digitalcitizen.life/what-is-p2p-peer-to-peer>
- [5] Teach-ICT. Peer to Peer Network: Advantages and Disadvantages. Obtenido de https://www.teach-ict.com/gcse_new/networks/peer_peer/miniweb/pg5.htm
- [6] R. Byung, C. Ki. Anonymous Communication Systems in P2P Network with Random Agent Nodes. Obtenido de https://link.springer.com/chapter/10.1007/11590354_97

- [7] L. Edgar Iván, S. Eduardo. Panorama jurisprudencial de las tecnologías peer-to-peer, obtenido de http://legal.legis.com.co/document/Index?obra=rmercantil&document=rmercantil_7680752a7fff404ce0430a010151404c
- [8] R. Wilson Rafael. Sistemas Peer to Peer (P2P) Análisis de las sentencias en los casos Napster, Grokster, Morpheus, Streamcast y Kazaa. Páginas de la 1 a la 7.
- [9] A. Sergio. ¿Vale la pena seguir descargando en redes P2P en la era del streaming? Obtenido de <https://www.genbeta.com/a-fondo/vale-la-pena-seguir-descargando-de-redes-p2p-en-la-era-del-streaming>
- [10] Ecured. Paul Baran. Obtenido de https://www.ecured.cu/Paul_Baran
- [11] B. Guy, M. Bruce. Parallel Algorithms. Obtenido de <https://www.cs.cmu.edu/~guyb/papers/BM04.pdf>
- [12] G. Stephen. 9 Parallel processing examples and applications. Obtenido de <https://builtin.com/hardware/parallel-processing-example>
- [13] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei and C. Qijun. A review on consensus algorithm of blockchain. Obtenido de <https://ieeexplore.ieee.org/abstract/document/8123011>
- [14] G. Oded, O. Yair. Definitions and properties of zero-knowledge proof systems.
- [15] H. Ben, M. Michael, W. Howard, J. Wolfgang. Panel I-A Definition and classification. Obtenido de <https://journals.sagepub.com/doi/abs/10.1177/00034894800893S203?journalCode=aora>
- [16] Data clustering: 50 years beyond k-means. Obtenido de <https://www.sciencedirect.com/science/article/abs/pii/S0167865509002323>
- [17] El algoritmo K-means aplicado a la clasificación y procesamiento de imágenes. Obtenido de https://www.uniovi.es/compnum/laboratorios_py/kmeans/kmeans.html

8 FIRMA DEL DOCUMENTO

Al actual día 12 de Junio del 2020:

Sebastián Franco Gómez 

Sebastián Franco Gómez

C.C 1.225.092.429