Universidad de Ciencias Aplicadas

INFORME DEL TRABAJO PARCIAL

CURSO DE COMPLEJIDAD ALGORÍTMICA

Carrera de Ingeniería de Software

Sección: WS6D

Proyecto: OptiNet

Código	Nombres y apellidos
u202213208	Christian Renato Espinoza Saenz
u202312230	Daniel José Huapaya Vargas
u202211846	Joaquin David Rivadeneyra Ramos

2024

Contenido

Problema

El crecimiento exponencial de la demanda de servicios de internet de alta velocidad ha impulsado a los proveedores de servicios de internet (ISP) a expandir y mejorar sus redes de fibra óptica para ofrecer conexiones más rápidas y confiables. Sin embargo, el diseño e implementación de estas redes representan un desafío significativo, ya que deben optimizarse para minimizar los costos de infraestructura, manteniendo al mismo tiempo un alto rendimiento y cobertura. El problema principal radica en encontrar la configuración óptima de estas conexiones que minimicen el costo total de la red, sin comprometer la calidad del servicio.

Para resolver este problema, se propone el uso del algoritmo de Kruskal, una técnica que forma parte de los algoritmos de árboles de expansión mínima (MST). El algoritmo de Kruskal selecciona las aristas de menor peso en orden ascendente y construye un árbol de expansión mínima evitando ciclos (Cormen, Leiserson, Rivest & Stein, 2009). Además, se implementó el algoritmo de Prim, ya que, es particularmente útil en grafos densos, donde hay muchas conexiones posibles minimizando el costo de cada paso al considerar solo las aristas más cercanas en cada iteración. Por último, se empleó la búsqueda en profundidad (DFS) para detectar el componente máximo del grafo, guardando los nodos en una lista y comparando si el tamaño de esa lista es mayor a otra lista de nodos que aún no han sido visitados.

El uso de los algoritmos de Kruskal y Prim en la planificación de redes de fibra óptica es crucial, ya que permite a los ISP reducir los costos de infraestructura y optimizar el uso de recursos. Según Kershenbaum (2003), en redes de telecomunicaciones, la minimización de costos es esencial para la sostenibilidad del proyecto y su expansión futura. Así, la aplicación de OptiNet asegura una red eficiente en términos de costo, sin comprometer la calidad de la conectividad.

Conjunto de Datos (Dataset)

El conjunto de datos utilizado para el análisis corresponde a una representación geográfica de una red de fibra óptica simulada (ver Figura 2). Cada dato en el conjunto incluye ubicaciones geoespaciales de puntos de acceso, representados como nodos, y las posibles conexiones entre ellos, representadas como aristas ponderadas con costos asociados. Los pesos se determinan aleatoriamente.

El dataset está compuesto por las siguientes características principales:

- **Nodos:** Cada nodo representa un punto de acceso a la red de fibra óptica, generalmente ubicaciones estratégicas como estaciones base o centros de distribución de internet.
- **Aristas:** Las posibles conexiones entre nodos, con un costo asociado.
- **Pesos de Conexión:** Cada arista tiene un valor numérico que representa el peso de establecer la conexión entre dos nodos. Los costos pueden variar en función de la distancia, el tipo de terreno, la infraestructura existente y las limitaciones legales o regulatorias.

El origen de los datos es un conjunto simulado de datos (ver Figura 1) donde se planificará un sistema de redes de fibra óptica. Para simulaciones, los datos se generan mediante modelos aleatorios. Estos datos son fundamentales para aplicar el algoritmo de Kruskal y Prim, ya que permiten identificar las conexiones más económicas para construir una red de fibra óptica eficiente.

Figura 1

Muestra de los dataset de conexiones en servicio de internet fijo por velocidad de bajada, empresa operadora y distrito (111922 conexiones)

NUM	FECHA_CORTE	PERIODO	EMPRESA	DEPARTAMENTO	PROVINCIA	DISTRITO	UBIGEO_DISTRITO	TECNOLOGÍA	SEGMENTO	RANGO_VELOC_BAJADA
1	20240617	202306	AGUSTINA SERVICIOS GENERALES S.A.C.	LA LIBERTAD	CHEPEN	CHEPEN	130401	Fibra Óptica	Residencial	50 Mbps <= BW < 100 Mb
2	20240617	202306	AGUSTINA SERVICIOS GENERALES S.A.C.	LA LIBERTAD	CHEPEN	CHEPEN	130401	Fibra Óptica	Residencial	100 Mbps <= BW < 200 Mb
3	20240617	202306	AGUSTINA SERVICIOS GENERALES S.A.C.	LA LIBERTAD	CHEPEN	CHEPEN	130401	Fibra Óptica	Residencial	BW >= 200 Mbps
4	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	BAGUA	BAGUA	10201	Otras Tecnologías	Residencial	2048 <= BW < 4096 kbps
5	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	BAGUA	BAGUA	10201	Otras Tecnologías	Residencial	16 Mbps <= BW < 30 Mbps
6	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	BAGUA	IMAZA	10205	Otras Tecnologías	Residencial	16 Mbps <= BW < 30 Mbps
7	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	BAGUA	LA PECA	10206	Otras Tecnologías	Residencial	16 Mbps <= BW < 30 Mbps
8	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	BONGARA	JAZAN	10307	Otras Tecnologías	Residencial	16 Mbps <= BW < 30 Mbps
9	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	BONGARA	VALERA	10311	Otras Tecnologías	Residencial	16 Mbps <= BW < 30 Mbps
10	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	BONGARA	YAMBRASBAMBA	10312	Otras Tecnologías	Residencial	16 Mbps <= BW < 30 Mbps
11	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	CHACHAPOYAS	CHACHAPOYAS	10101	Fibra Óptica	Comercial	30 Mbps <= BW < 50 Mbps
12	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	CHACHAPOYAS	CHACHAPOYAS	10101	Fibra Óptica	Comercial	50 Mbps <= BW < 100 Mb
13	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	CHACHAPOYAS	CHACHAPOYAS	10101	Fibra Óptica	Comercial	100 Mbps <= BW < 200 Mb
14	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	CHACHAPOYAS	CHACHAPOYAS	10101	Fibra Óptica	Residencial	8 Mbps <= BW < 16 Mbps
15	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	CHACHAPOYAS	CHACHAPOYAS	10101	Fibra Óptica	Residencial	30 Mbps <= BW < 50 Mbps
16	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	CHACHAPOYAS	CHACHAPOYAS	10101	Fibra Óptica	Residencial	50 Mbps <= BW < 100 Mb
17	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	CHACHAPOYAS	CHACHAPOYAS	10101	Fibra Óptica	Residencial	100 Mbps <= BW < 200 Mb
18	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	CHACHAPOYAS	CHACHAPOYAS	10101	Fibra Óptica	Residencial	BW >= 200 Mbps
19	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	CHACHAPOYAS	CHACHAPOYAS	10101	Otras Tecnologías	Comercial	16 Mbps <= BW < 30 Mbps
20	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	CHACHAPOYAS	CHACHAPOYAS	10101	Otras Tecnologías	Residencial	8 Mbps <= BW < 16 Mbps
21	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	CHACHAPOYAS	CHACHAPOYAS	10101	Otras Tecnologías	Residencial	16 Mbps <= BW < 30 Mbps
22	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	CHACHAPOYAS	LEIMEBAMBA	10110	Otras Tecnologías	Residencial	16 Mbps <= BW < 30 Mbps
23	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	LUYA	LUYA	10509	Otras Tecnologías	Residencial	16 Mbps <= BW < 30 Mbps
24	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	LUYA	TINGO	10522	Otras Tecnologías	Residencial	8 Mbps <= BW < 16 Mbps
25	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	LUYA	TINGO	10522	Otras Tecnologías	Residencial	16 Mbps <= BW < 30 Mbps
26	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	UTCUBAMBA	BAGUA GRANDE	10701	Otras Tecnologías	Residencial	2048 <= BW < 4096 kbps
27	20240617	202306	AMERICA MOVIL PERU S.A.C.	AMAZONAS	UTCUBAMBA	BAGUA GRANDE	10701	Otras Tecnologías	Residencial	8 Mbps <= BW < 16 Mbps

Nota. *Plataforma Nacional de Datos Abiertos. (2024)*

Primero, utilizamos dos bases de datos. Una sirve para obtener la altitud y longitud a partir del código de ubigeo, el cual obtenemos en otro archivo del Excel de conexiones en servicio de internet fijo por velocidad de bajada, empresa operadora y distrito. Luego, generamos un dataset más refinado, que contiene las distancias euclidianas, asegurándonos de seleccionar las tres más pequeñas y diferentes. En otras palabras, debe haber al menos tres conexiones. Después, procedemos a refinar los datos utilizando Google Maps. Debido a que el proceso de obtención de los datos es lento, utilizamos el archivo 'dataset.csv' como referencia.

Figura 2

Muestra del dataset.csv (4733 conexiones)

distric 1:	lon1	lat1	distric 2:	lon2	lat2	distance	cost
CHEPEN	-79.4294	-7.2275	GUADALUPE	-79.4703	-7.2436	5.639	6.03373
CHEPEN	-79.4294	-7.2275	PACANGA	-79.4856	-7.1714	12.547	13.42529
CHEPEN	-79.4294	-7.2275	SAN JOSE	-79.4553	-7.35	19.757	21.13999
CHEPEN	-79.4294	-7.2275	SAN PEDRO DE LLOC	-79.5147	-7.4183	29.078	31.11346
CHEPEN	-79.4294	-7.2275	JEQUETEPEQUE	-79.5631	-7.3375	25.421	27.20047
BAGUA	-78.5311	-5.6389	EL MILAGRO	-78.5583	-5.6378	4.688	5.01616
BAGUA	-78.5311	-5.6389	LA PECA	-78.4369	-5.6119	15.336	16.40952
BAGUA	-78.5311	-5.6389	COPALLIN	-78.4231	-5.675	15.939	17.05473
BAGUA	-78.5311	-5.6389	CAJARURO	-78.4267	-5.7364	18.92	20.2444
BAGUA	-78.5311	-5.6389	BELLAVISTA	-78.6772	-5.6678	40.884	43.74588
BAGUA	-78.5311	-5.6389	ARAMANGO	-78.4378	-5.4164	40.883	43.74481
IMAZA	-78.2889	-5.1636	ARAMANGO	-78.4378	-5.4164	61.047	65.32029
IMAZA	-78.2889	-5.1636	LA PECA	-78.4369	-5.6119	109.607	123.85591
IMAZA	-78.2889	-5.1636	HUARANGO	-78.7758	-5.2722	192.51	217.5363
LA PECA	-78.4369	-5.6119	COPALLIN	-78.4231	-5.675	13.469	14.41183
LA PECA	-78.4369	-5.6119	EL MILAGRO	-78.5583	-5.6378	19.662	21.03834
LA PECA	-78.4369	-5.6119	CAJARURO	-78.4267	-5.7364	32.835	35.13345
LA PECA	-78.4369	-5.6119	BAGUA GRANDE	-78.4428	-5.7547	37.011	39.60177
LA PECA	-78.4369	-5.6119	ARAMANGO	-78.4378	-5.4164	55.858	59.76806
JAZAN	-77.9772	-5.9414	SHIPASBAMBA	-77.9806	-5.9106	9.417	10.07619
JAZAN	-77.9772	-5.9414	SAN CARLOS	-77.9453	-5.9661	8.044	8.60708
JAZAN	-77.9772	-5.9414	CHURUJA	-77.9519	-6.0194	10.402	11.13014
JAZAN	-77.9772	-5.9414	FLORIDA	-77.9694	-5.8261	31.605	33.81735
VALERA	-77.9192	-6.0428	CHURUJA	-77.9519	-6.0194	12.345	13.20915

Nota. Elaboración propia

Propuesta

Objetivo

El objetivo de esta propuesta es diseñar una red de fibra óptica optimizada para proveedores de servicios de internet (ISP), que conecte múltiples puntos de acceso geográficamente distribuidos, minimizando los costos de infraestructura. Para alcanzar este objetivo, se implementaron los algoritmos de Kruskal y Prim para construir árboles de expansión mínima (MST), reduciendo el costo total de las conexiones. Kruskal es ideal para grafos dispersos, mientras que Prim se aplica eficazmente en grafos densos. Adicionalmente, se utilizó la búsqueda en profundidad (DFS) para identificar el componente máximo del grafo, asegurando la conectividad total de los puntos relevantes de la red. (GeeksforGeeks, 2023).

Técnica Utilizada

La técnica utilizada será el algoritmo de Kruskal, un método eficiente para la resolución de problemas de árboles de expansión mínima (MST). Kruskal se caracteriza por conectar los nodos del grafo de forma que el costo total de las conexiones sea el mínimo posible, evitando ciclos y redundancias. Este algoritmo es ideal para diseñar redes donde se requiere minimizar costos en infraestructuras de telecomunicaciones (Ravikiran, 2024).

Además, se implementó el algoritmo de Prim, que también construye un árbol de expansión mínima, pero sigue un enfoque diferente. Prim comienza desde un nodo inicial y expande el árbol agregando iterativamente la arista de menor peso conectada al conjunto de nodos ya incluidos. Este algoritmo es particularmente útil para grafos densos, donde hay muchas conexiones posibles, ya que minimiza el costo en cada paso al evaluar solo las aristas adyacentes al árbol. (Weiss, 2012).

Por último, se empleó la búsqueda en profundidad (DFS) para identificar el componente máximo del grafo. La DFS permite explorar exhaustivamente cada rama del grafo, almacenando los nodos visitados en una lista y comparando su tamaño con otra lista

de nodos no visitados, lo que facilita detectar el subconjunto más grande de nodos conectados. (Sedgewick, 2011).

Metodología

1. **Recopilación de Datos:** Se utilizarán conjuntos de datos simulados que especifique los puntos de acceso y los costos asociados a las posibles conexiones entre ellos.
2. **Modelado del Grafo:** Los puntos de acceso se representarán como nodos en un grafo, y las posibles conexiones entre ellos como aristas ponderadas por su costo.
3. **Aplicación de Algoritmos:**

- Se aplicó el algoritmo de Kruskal para seleccionar las conexiones con menor costo total, construyendo un árbol de expansión mínima sin ciclos y conectando todos los puntos de acceso.
- Se implementó el algoritmo de Prim, comenzando desde un nodo inicial y expandiendo el MST agregando iterativamente las aristas de menor peso. Este enfoque es particularmente efectivo para grafos densos, donde hay muchas conexiones posibles.
- Se utilizó la búsqueda en profundidad (DFS) para detectar el componente máximo del grafo, guardando los nodos visitados en una lista y comparando su tamaño con otra lista de nodos no visitados, para identificar el subconjunto más grande de nodos conectados.

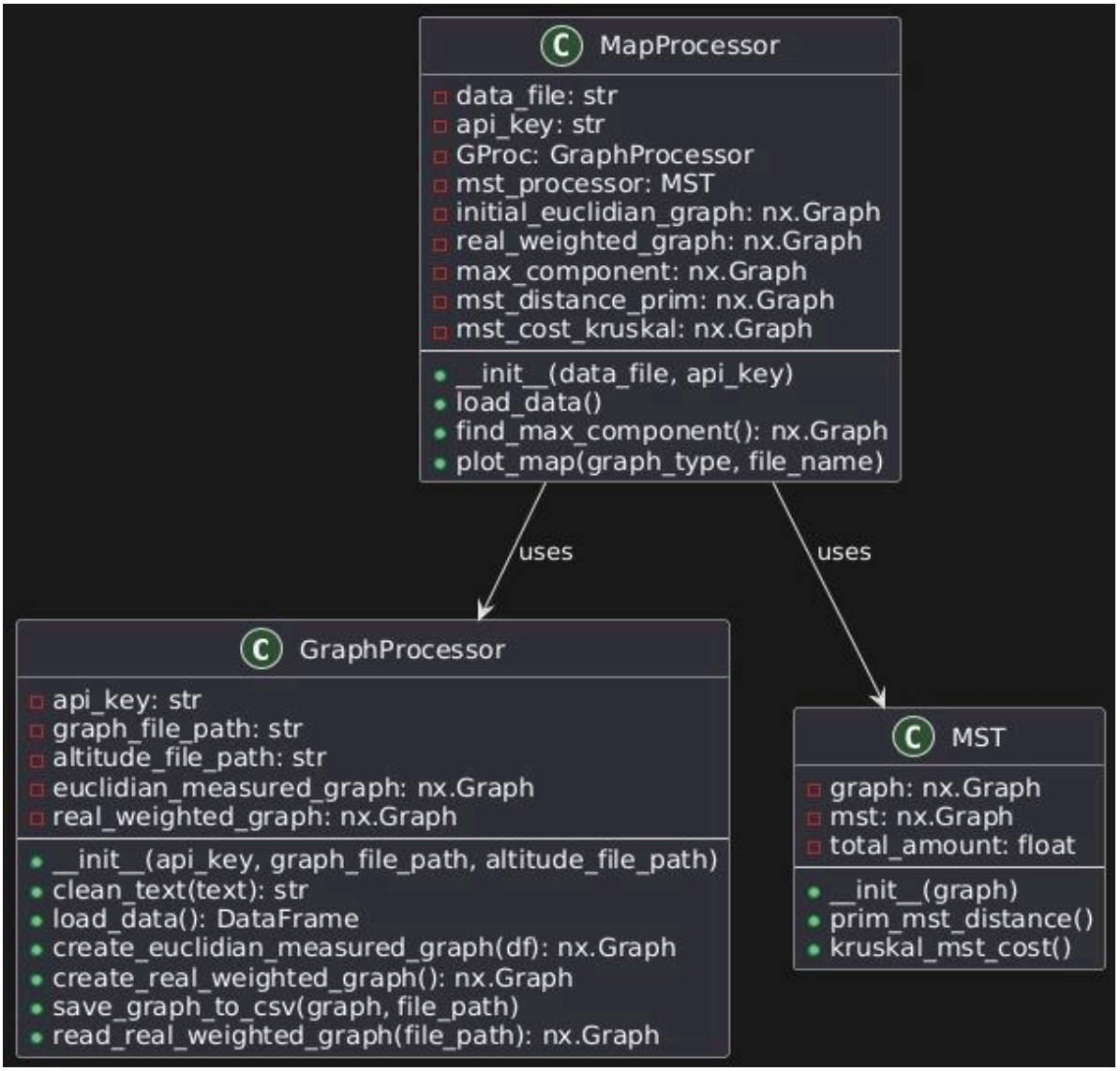
4. **Evaluación del Diseño:** Se validará la red propuesta para asegurar que minimiza los costos y cumple con los requisitos de conectividad, escalabilidad y eficiencia de la red.

Diseño del aplicativo

Diagrama de clase y diagrama de base de datos

Figura 3

Diagrama de clases UML para el sistema completo.



Nota. El diagrama muestra un sistema de procesamiento de gráficos y mapas, compuesto por tres clases principales: GraphProcessor, MST y MapProcessor. GraphProcessor gestiona la carga y transformación de datos en gráficos, incluyendo un gráfico euclidiano y uno ponderado. Además, posee métodos para limpiar datos, cargar información desde archivos, y guardar gráficos en CSV. MST se especializa en calcular el Árbol de Expansión Mínima (MST) usando Prim y Kruskal para encontrar caminos mínimos en un gráfico ponderado. MapProcessor utiliza instancias de GraphProcessor y MST para cargar datos, identificar el componente máximo de un gráfico, y generar visualizaciones de mapas, representando así un sistema integrado que procesa y analiza estructuras de gráficos complejos y sus rutas óptimas.

Módulo para la adecuación de datos fuente a estructura de datos tipo grafo

Figura 4

Código de adecuación de datos fuente

```
class GraphProcessor:
    def __init__(self, api_key, graph_file_path, altitude_file_path):
        self.api_key = api_key
        self.graph_file_path = graph_file_path
        self.altitude_file_path = altitude_file_path
        self.euclidian_measured_graph = None
        self.real_weighted_graph = None
...
    def find_max_component(self):
        visited = set()
        max_component = set()

        def dfs(node, component):
            stack = [node]
            while stack:
                current = stack.pop()
                if current not in visited:
...
class MST:
    def __init__(self, graph):
        self.graph = graph
        self.mst = nx.Graph()
        self.total_amount = 0
...
```

Nota. GraphProcessor manipula los datos del dataset original y convierte estos en un grafo; este se encuentra en initial_dataset.py. La función find_max_component encuentra la componente máxima de dicha manipulación de datos y se encuentra en Map_Processor.py. Por último, MST, procesa la componente máxima y los transforma en MSTs.

Módulo para la representación de datos en el grafo

Figura 5

Código parcial de MapProcessor

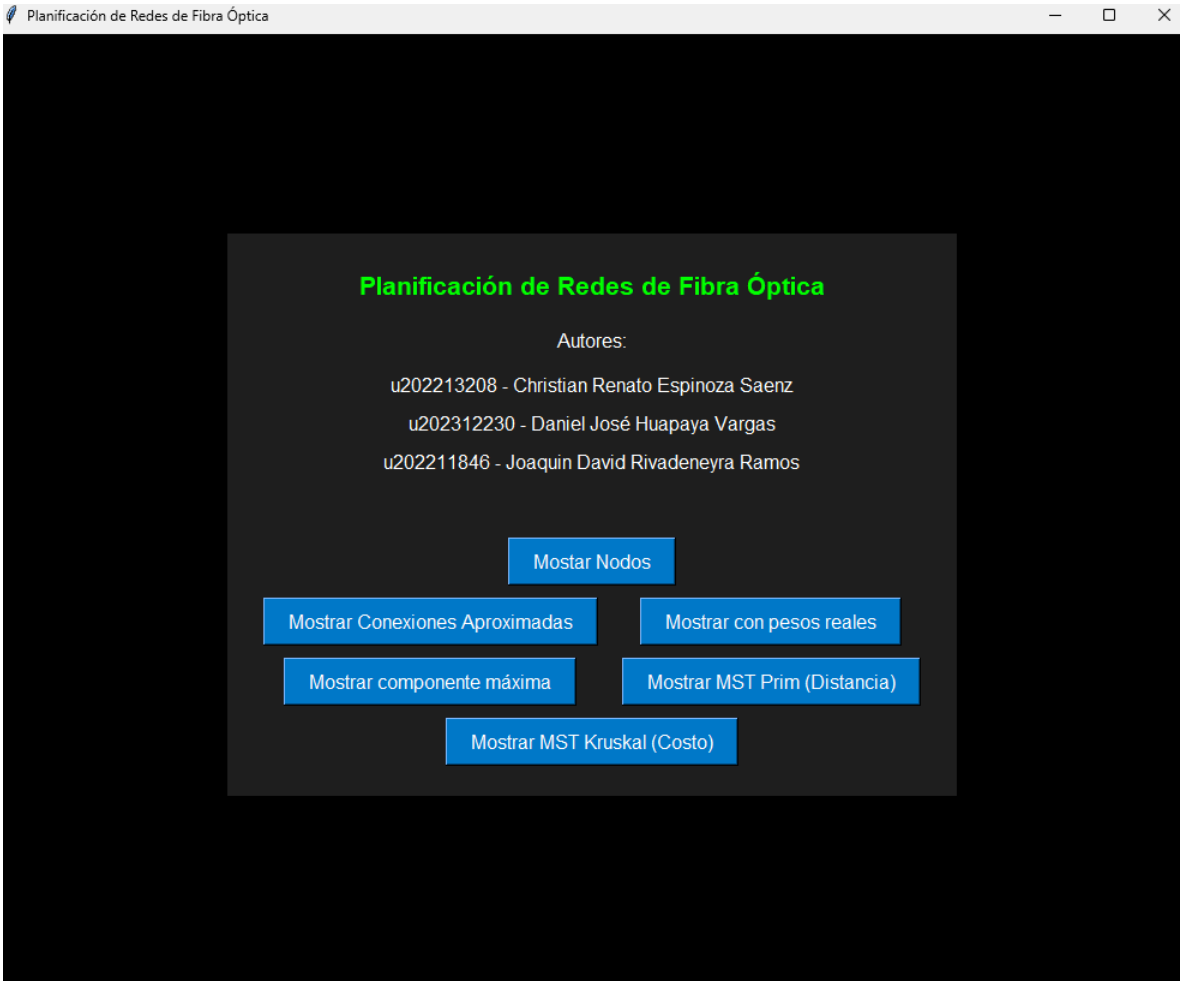
```
class MapProcessor:
    def __init__(self, data_file, api_key):
        self.data_file = data_file
        self.api_key = api_key
        self.load_data()

    def load_data(self):
        self.GProc = GraphProcessor(self.api_key, 'CAN.csv',
'TB_UBIGEOS.csv')
        df = self.GProc.load_data()
        self.initial_euclidian_graph =
self.GProc.create_euclidian_measured_graph(df)
        self.real_weighted_graph =
self.GProc.read_real_weighted_graph('dataset.csv')
        self.max_component = self.find_max_component()
        self.mst_processor = MST(self.max_component)
        self.mst_processor.prim_mst_distance()
        self.mst_distance_prim = self.mst_processor.mst
        self.mst_distance_total = self.mst_processor.total_amount
        self.mst_processor.kruskal_mst_cost()
        self.mst_cost_kruskal = self.mst_processor.mst
        self.mst_cost_total = self.mst_processor.total_amount
...
```

Nota. MapProcessor se encarga de dibujar los mapas por cada tipo de grafo.

Interfaz tentativa (visual) para usuarios

Figura 6

Interfaz gráfica inicial



Validación de resultados y pruebas

Figura 7

Mapa con todos los nodos del dataset inicial

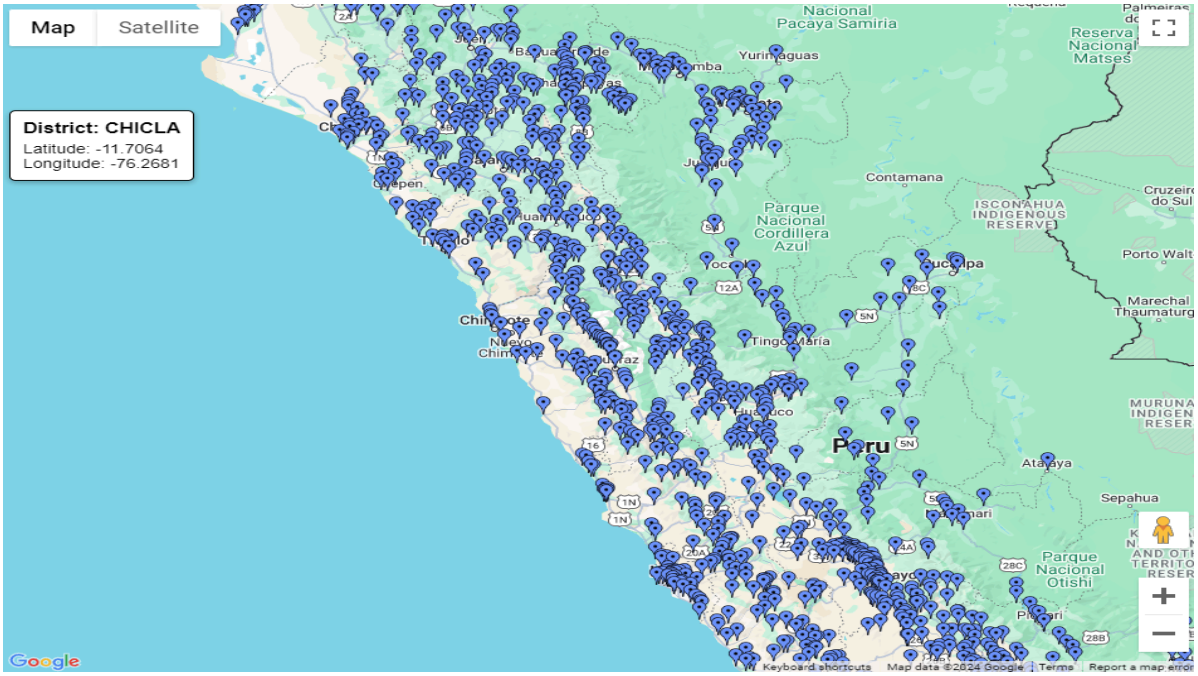


Figura 8

Mapa del grafo inicial con las conexiones por la distancia euclidiana

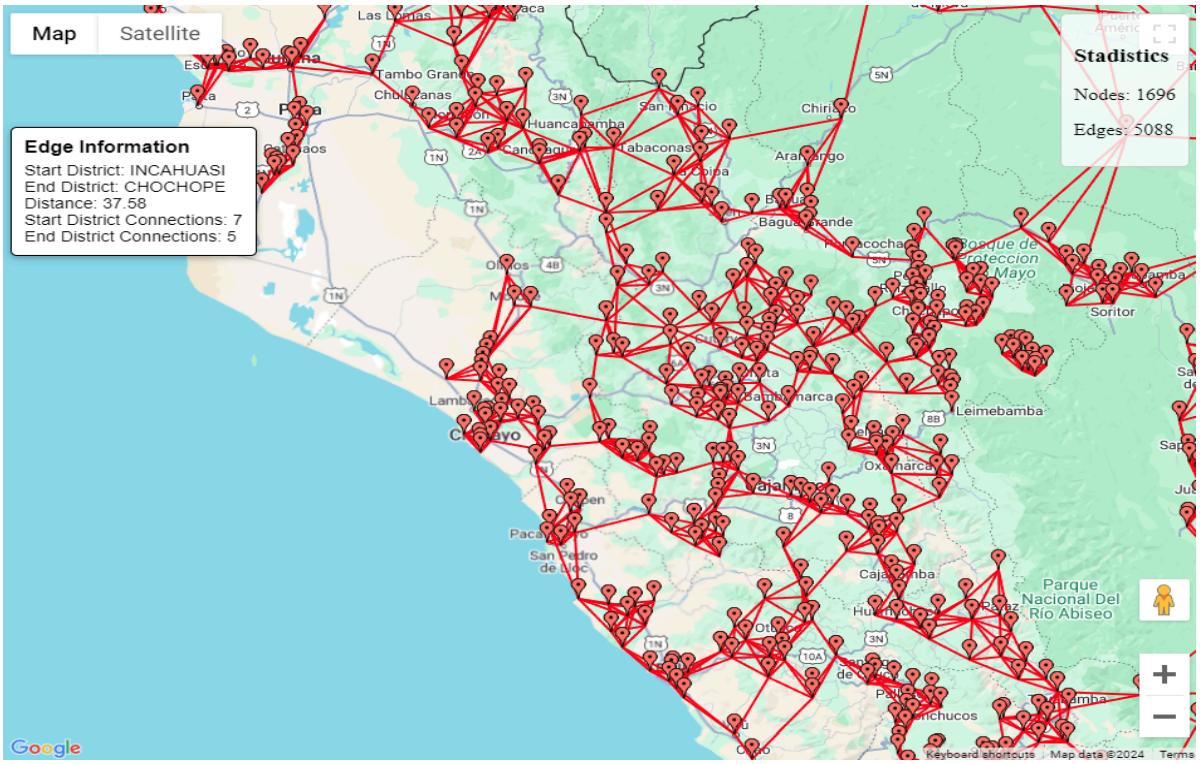


Figura 9

Mapa del grafo inicial con conexiones a través de google maps

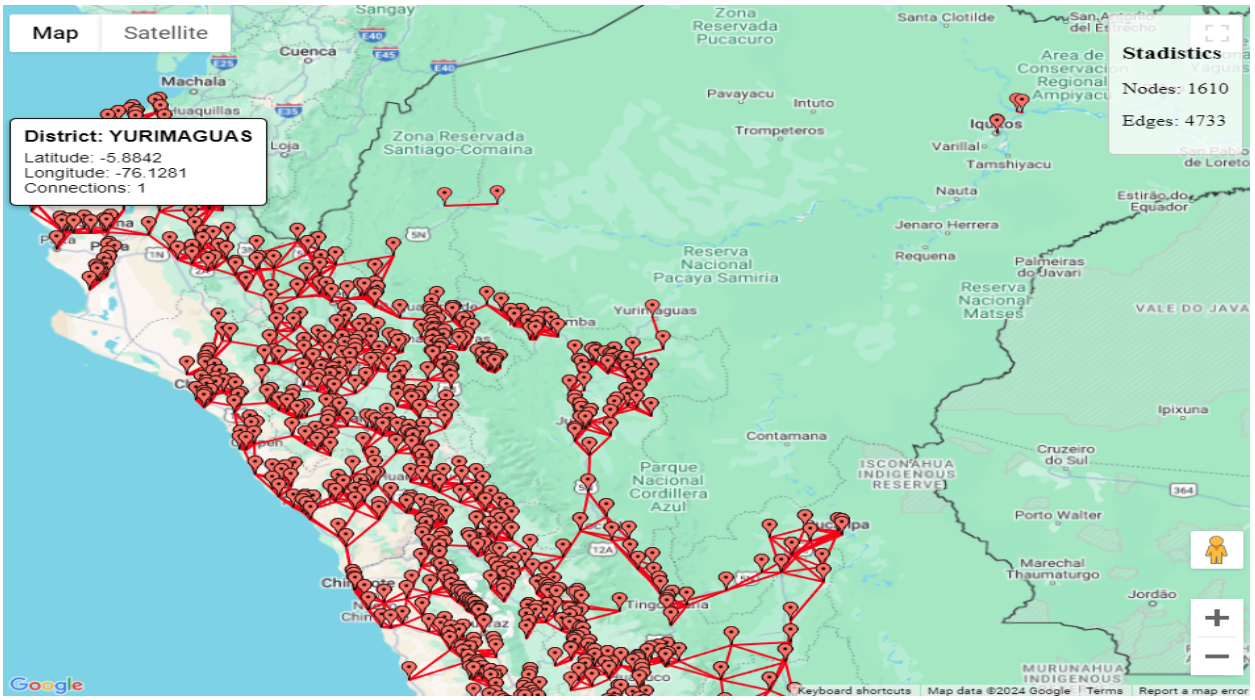
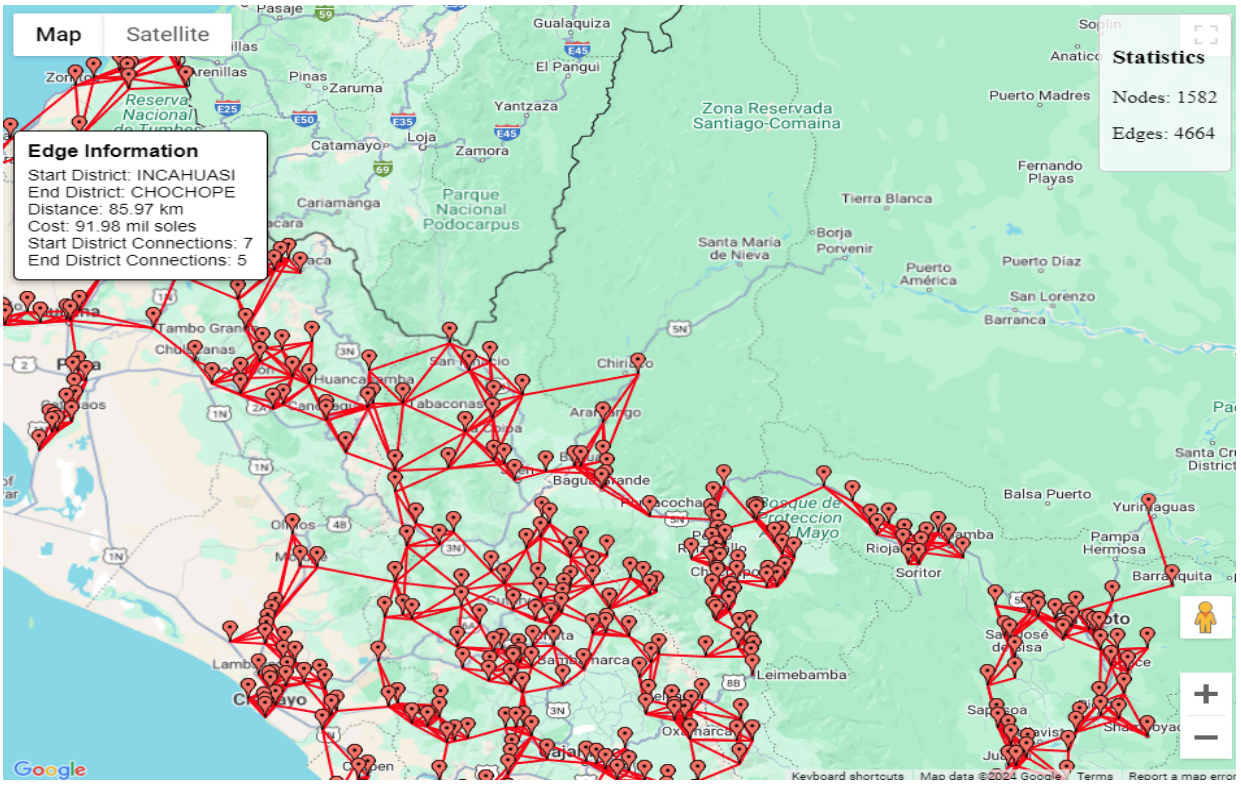


Figura 10

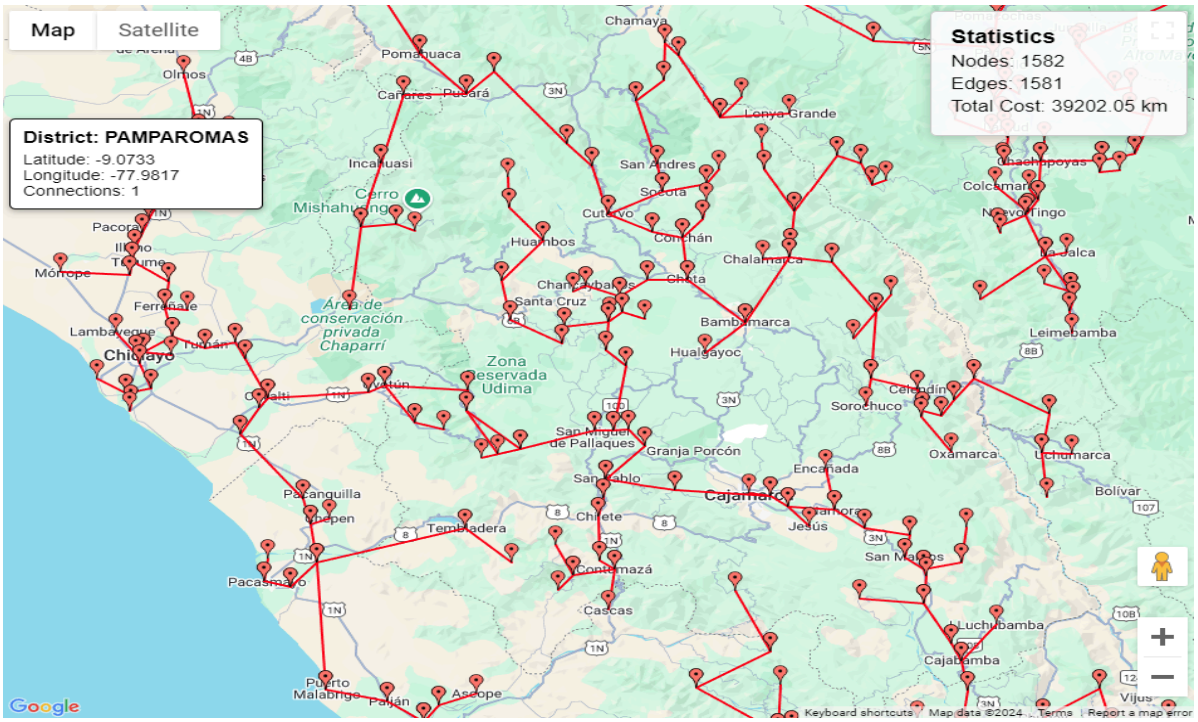
Mapa del grafo de la máxima componente



Nota. Este grafo se obtuvo a partir del grafo de conexiones a través de Google Maps y se utilizará para obtener los MSTs.

Figura 11

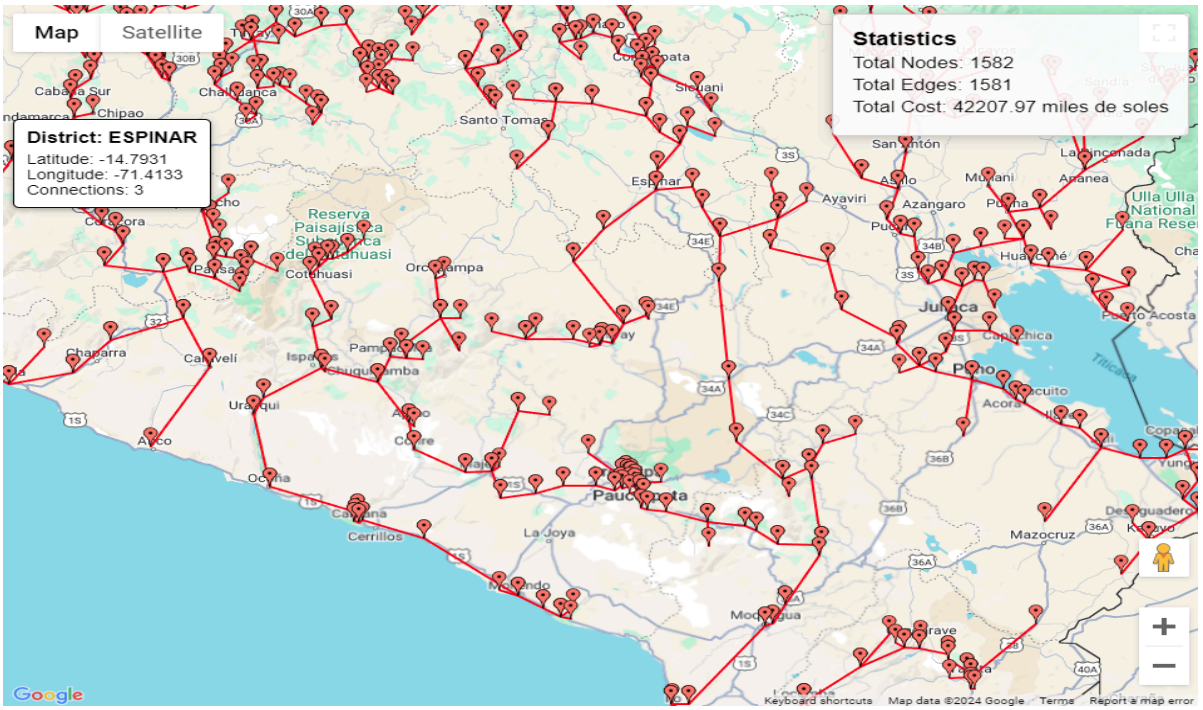
Mapa del MST con respecto a la distancia entre los nodos



Nota. Se aplicó el algoritmo de Prim.

Figura 12

Mapa del MST con respecto al costo entre los nodos



Nota. Se aplicó el algoritmo de Kruskal.

Interpretación de resultados:

Nuestro programa logra mostrar los datos de manera efectiva a través del procesamiento de archivos de excel, el despliegue de los grafos en un mapa y el uso de algoritmos correspondientes al curso. Esto asegura la conectividad entre múltiples puntos de acceso y permite una red escalable, lo que convierte al programa en una herramienta valiosa para optimizar las infraestructuras de los proveedores de servicios de internet.

Conclusiones

El diseño de OptiNet sobre una red de fibra óptica mediante el algoritmo de Kruskal y Prim resultó ser una estrategia eficiente para optimizar las conexiones entre múltiples puntos de acceso, reduciendo de manera significativa los costos de infraestructura. Este enfoque no solo garantiza una red de bajo costo, sino que también asegura que no existan ciclos redundantes, lo que mejora la eficiencia general del sistema. La implementación de una base de datos estructurada, que gestiona la información de los nodos, las conexiones y los costos, fue clave para facilitar el uso de los algoritmos y simplificar la toma de decisiones sobre las mejores rutas a seguir. Este diseño proporciona una base sólida para futuras expansiones de la red. De cara al futuro, sería valioso considerar factores adicionales como la latencia de las conexiones, la redundancia en caso de fallos, y la posibilidad de incorporar algoritmos más avanzados o simulaciones para mejorar la resiliencia y escalabilidad, especialmente en redes de telecomunicaciones más grandes y complejas. Esto permitirá enfrentar desafíos más amplios y mantener la red adaptable a las crecientes demandas de conectividad hoy en día.

Referencias

GeeksforGeeks. (2023, 05 de octubre). *Kruskal s Minimum Spanning Tree MST Algorithm*. Recuperado de https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/

Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2009). *Introduction to Algorithms*. MIT Press.

Ravikiran, A. (2024, 31 de agosto). *Your One-Stop solution to learn Kruskal algorithm from scratch*. Simplilearn. Recuperado de https://www.simplilearn.com/tutorials/data-structure-tutorial/kruskal-algorithm

Weiss: Weiss, M. A. (2012). Data Structures and Algorithm Analysis in C++ (4ta ed.). Pearson Education.

Sedgewick: Sedgewick, R. (2011). Algorithms (4ta ed.). Addison-Wesley Professional.

Kershenbaum, A. (2003). *Telecommunications Network Design Algorithms*. McGraw-Hill.

DataScientest. (2024, 20 de marzo.). *Kruskal algorithm: Definition and purpose*. DataScientest. Recuperado de https://datascientest.com/en/kruskal-algorithm-definition-and-purpose

Plataforma Nacional de Datos Abiertos. (2024, June 28). Cantidad de conexiones en servicio de internet fijo por velocidad de bajada, empresa operadora y distrito.Recuperado de https://www.datosabiertos.gob.pe/dataset/cantidad-de-conexiones-en-servicio-de-internet-fijo-por-velocidad-de-bajada-empresa

Plataforma Nacional de Datos Abiertos. (s.f.). Códigos equivalentes de ubigeo del Perú.Recuperado de https://www.datosabiertos.gob.pe/dataset/codigos-equivalentes-de-ubigeo-del-peru