**6Universidad Peruana de Ciencias Aplicadas**

**Facultad de Ingeniería**



**Ciclo: Cuarto**

**Curso: Complejidad Algorítmica, CC43**

**Docente: Luis Martín Canaval Sanchez**

**Informe de Trabajo Parcial**

Problema: Conexión de Centros de Generación Eléctrica

| **Integrantes** | **Código** | **Carrera** |
| --- | --- | --- |
| André Elliot Escalante Bueno | U20191E817 | Ciencias de la Computación |
| Ernesto David de Loayza Zamudio | U202121772 | Ciencias de la Computación |

**Lima - Perú**

**2024**

**Descripción del problema**

El Ministerio de Energía y Minas tiene un proyecto de reestructuración de la distribución eléctrica en el país. Para esto, se ha abierto una licitación para contratar a una concesionaria que se pueda encargar del trabajo. Cada una de las concesionarias entregará al ministerio un conjunto de datos en el que se describirán las conexiones entre generadores y subestaciones eléctricas. Las especificaciones de los cables están estandarizadas, por lo que cada empresa solo tendrá que enfocarse en usar la menor cantidad de cable posible y asegurarse de habilitar la energía necesaria a cada subestación. El ministerio precisa de software que ayude a automatizar verificar que el flujo de la energía pueda ser soportado por las conexiones para alimentar a las ciudades alrededor del país.

**Descripción del conjunto de datos (Dataset)**

El dataset a utilizar consiste se obtuvo en internet a través del siguiente link:[GEO GPS PERÚ: Mapa de ELECTRICIDAD Líneas de Transmisión LT y mas - Shapefile Gratis (geogpsperu.com)](https://www.geogpsperu.com/2020/09/mapa-de-electricidad-lineas-de.html)

Este contiene varias carpetas que contienen distintos aspectos de la red eléctrica del Perú.

* Subestaciones de transformación
* Centros Hidroeléctricos
* Recursos Renovables
* Centrales Termoeléctricas
* Centros de Generación Aislada (municipales y por concesión)
* Líneas de transmisión

Estos se encontraban en formato .shp (Shapefile), el cual es usado por distintas aplicaciones de análisis geográfico.

Todos estos fueron exportados a formato CSV el cual nos permite leer y manipular de manera más sencilla los datos.

La información contenida en estos nodos es la siguiente:

* Nombre
* Nro de tensión
* Potencia
* Nro de transmisión
* Coordenadas

**Propuesta**

Para resolver este problema, se utilizará el algoritmo Ford-Fulkerson que logrará determinar el flujo máximo de energía que puede pasar a través de las conexiones propuestas. Se utilizarán a los generadores y a las subestaciones como los nodos del grafo y a las conexiones como las aristas. Se determinará si es que se está abasteciendo la energía suficiente a todas las subestaciones y si no es el caso, se mostrará al usuario una lista de subestaciones a las que no se les está proporcionando las conexiones pertinentes.

**Diseño del aplicativo**

En este proyecto el software se tendrá que procesar los datos de las subestaciones y las conexiones. Se tendrá en una base de datos la información de las estaciones de energía y el usuario proporcionará la información de las conexiones. Como se ha mencionado antes, se utilizará el algoritmo de Ford-Fulkerson para hallar el flujo máximo y asegurar que la capacidad es la suficiente para alimentar a las subestaciones.

Para analizar el algoritmo en este caso se declararán dos cosas:

* La cantidad de conexiones se representará con la letra *E*.
* La capacidad máxima de flujo se representará con la letra *F*.
* El número de aristas en el camino más largo con la letra *n*.

Este algoritmo itera en tres pasos:

1. Encontrar un camino de aumento (Un camino posible de flujo).
2. Encontrar el cuello de botella.
3. Aumentar el flujo total de cada una de las aristas del camino.

En el peor de los casos, estos pasos se ejecutarán una cantidad *F* de veces. Además, cada uno de los pasos tiene su propia complejidad:

1. Encontrar un camino de aumento:
   * Para esto, se utilizará el algoritmo de búsqueda DFS el cual tiene un O(*E*).
2. Encontrar el cuello de botella
   * El peor costo posible es el camino más largo desde el nodo de origen hasta el nodo de llegada. Por lo que tienen un O(n).
3. Aumentar el flujo total de cada una de las aristas del camino:
   * El peor costo posible es el camino más largo desde. Tiene un O(*n*).

Con esto se tiene en suma un O(*F* \* *E* + *n*).

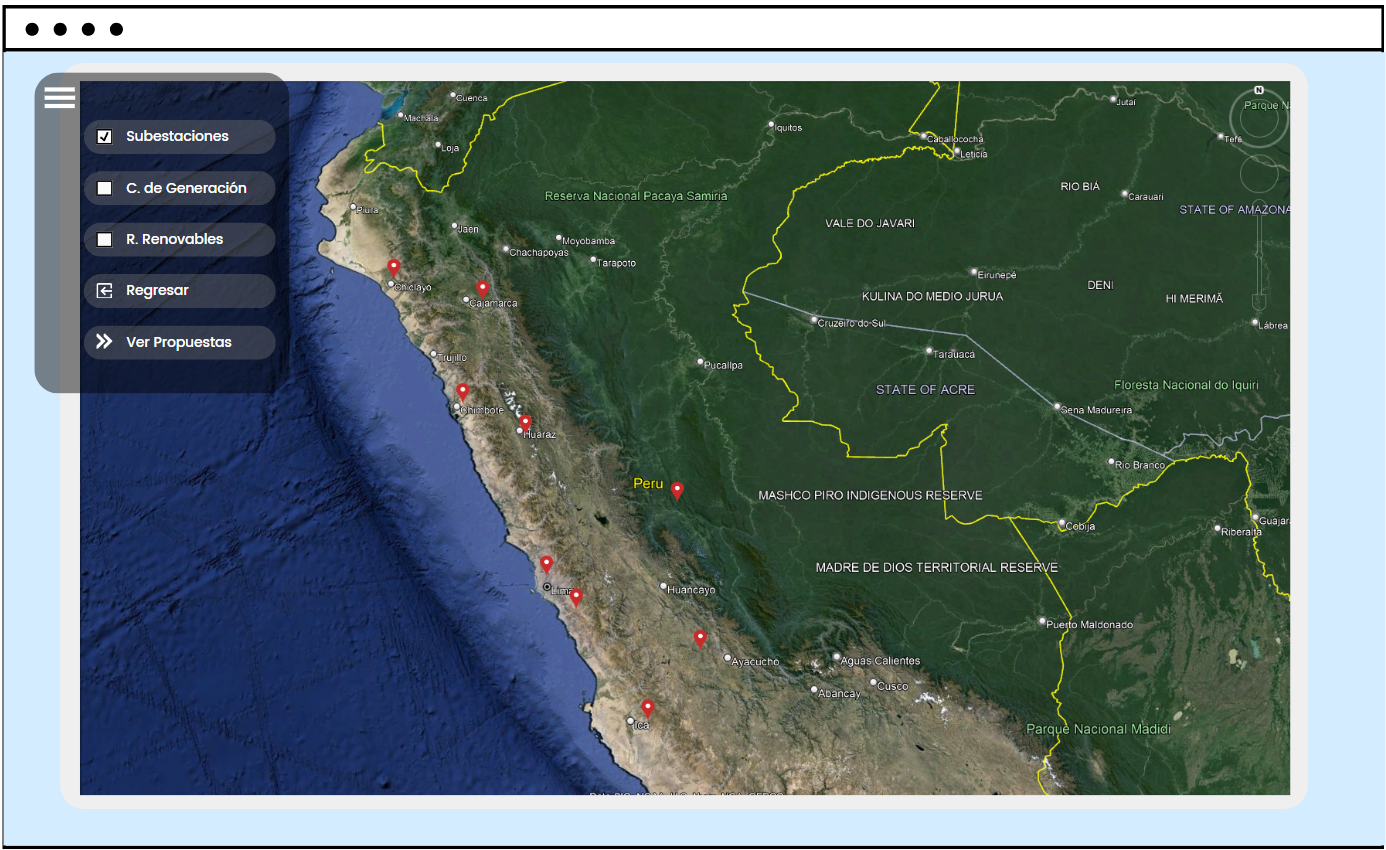
Siguiendo con el diseño del aplicativo. Se diseñará una arquitectura de cliente-servidor. Del lado del servidor se tendrá un rest API en el que se procesará la información y se aplicará los algoritmos antes descritos. Se utilizará python django para desarrollar el programa. Del lado del cliente, usaremos la biblioteca de Node.js: Astro. En esta parte del desarrollo se creará la interfaz de usuario con la que se enviará la información al servidor y se mostrará los resultados dados por el servidor en mapas interactivos que el usuario podrá visualizar.

**Pantallazos de la interfaz de usuario:**

**Pantalla inicial**

****

**Ver nodos/mapa**

****

**Propuestas de Red**

****

**Bibliografía**

(2023, 14 noviembre). *Cómo se distribuye la energía eléctrica: todo lo que necesitas saber*. Polaridad.es. https://polaridad.es/como-se-distribuye-la-energia-electrica/