**Universidad Peruana de Ciencias Aplicadas**



Competencia de Pensamiento Innovador

COMPLEJIDAD ALGORITMICA

Carrera de Ciencias de la Computación

Sección: CC41

Alumno: Marlon Omar Llaguento de la Cruz

Código: U2020B055

Julio 2021

En la realización del presente proyecto se puso en práctica los conocimientos adquiridos a lo largo del ciclo, como los algoritmos de búsqueda en grafos, del camino más corto, etc. Gracias a estos se pueden crear mejores programas, combinando los distintos algoritmos, puesto que, todos los algoritmos no son perfectos. Entonces, uno de estos podrá suplir la debilidad del otro, para así dar a lugar un algoritmos más eficiente y efectivo. Por otro lado, el aprendizaje del lenguaje Python es crucial para la implementación de estos. Así pues, se usaron todos estos recursos para poder desarrollar una solución para el proyecto, que yo llamaré “MiniGoogleMaps”.

La problemática para solucionar es la implementación de algoritmos de búsqueda del camino mas corto para un grafo. En primer lugar, se necesita mapear una ciudad, donde cada intersección será un vértice y las calles que conectas estos, serán las aristas. También, se tomará en cuenta el sentido de las calles. En segundo lugar, saber las horas pico de tráfico y la distancia entre vértices para una correcta asignación de los pesos de las aristas. Finalmente, encontrar el algoritmo adecuado para brindar al usuario una guía del camino con menor tiempo de recorrido.

Con mi equipo, se realizaron reuniones y asignaciones de tareas para ir desarrollando la implementación del programa. En las llamadas virtuales, planeamos primero en elegir una ciudad y marcar cada intersección. Luego, asignar a cada nodo una intersección. Después, comenzaríamos con el desarrollo del programa teniendo en cuenta la información recolectada. Además, se agregaría una interfaz gráfica para hacer más amigable la visualización de los caminos y el dibujo del mapa.

Para implementar el programa, primero usamos la herramienta Colab que nos brinda Google. Ahí guardaremos la información de los vértices y su conexión en una lista de adyacencia, para que así pueda ser más fácil la lectura de esta lista. Segundo, utilizamos el algoritmo Perlin Noise para obtener un valor que se multiplicará con la distancia de la arista y la hora, dando así un peso que será asignado a cada calle. Tercero, hicimos uso de los algoritmos de Dijkstra y de Jhonson para obtener el camino más corto que será guardado en una lista en Python. Sin embargo, el algoritmo de Jhonson no se pudo implementar correctamente, por ello hicimos uso del algoritmo de Dijkstra para obtener dicho camino. Tercero, obtuvimos las coordenadas de cada vértice y desarrollamos una fórmula para transformar las coordenadas que son decimales y negativas a enteras positivas. Finalmente, se implementó código HTML, JavaScrip y CCS, en combinación con el programa Conda, para la visualización del mapa con sus conexiones en una página web usando nuestra IP pública. En consecuencia, se pudo implementar exitosamente la interfaz gráfica.