Informe 3 TP 2:

Se nos solicitó idear una forma de que los mensajes enviados por las palomas mensajeras desde la aldea peligros lleguen a repartir la noticia al resto de las 21 aldeas de la forma más eficiente posible.Para ello la cátedra nos proporcionó parte del código de las clases vértices y grafos las cuales tuvimos que usar de base y realizarles cierta cantidad de cambios para llevar a cabo nuestra tarea.También tenemos a disposición un archivo txt en el cual detalla las conexiones entre las aldeas y la distancia.Usamos esta información para armar nuestro grafo donde cada aldea era un vértice.

Para mostrar las aldeas en orden alfabético simplemente creamos los métodos cargar_aldeas(nombre_archivo) el cual recibe como parámetro la ruta del archivo aldeas.txt y nos crea un grafo, y para ordenar las aldeas usamos obtenerVericesOrdenados() el cual hace un sorted al resultado del método obtenerVertices().Todos estos métodos se encontraban en nuestra clase Grafo.

En cuanto a la difusión del mensaje de la forma más eficiente tuvimos que pensar en qué algoritmo podríamos aplicar en nuestro grafo(Dijkstra,Prim,Warshall), en primer lugar implementamos Dijkstra porque se basa en busqueda en anchura y nos devolvió el camino más corto(en términos de distancia) a cada aldea partiendo desde “Peligros”.Pero este algoritmo no nos permite trazar la ruta más eficiente en sí, por lo que implementamos el algoritmo Prim el cual se diferenciaba de nuestra implementación anterior en que dijkstra trabajaba con la distancia más corta mediante una suma y en prim usaba una variable:nuevoCosto el cual se basaba en las ponderaciones de los vértices.Por lo que de esta forma implementamos el algoritmo de Prim y tambien un metodo para obtener las distancias del “camino” que nos trazaba el cual era distanciatotal(grafo).

De esta forma pudimos mostrar de cada aldea de que vecina recibe la noticia y si era necesario a que aldeas debía de mandarles el mensaje.

Cabe mencionar que para la correcta implementación de nuestro grafo y el algoritmo prim usamos la estructura de datos de actividades pasadas: una cola de prioridad cuyo funcionamiento interno se basa en un montículo de mínima cuyo criterio de ordenamiento son las ponderaciones del vértice raíz(al que se le asigna distancia cero).A su vez tuvimos que incorporar a nuestra cola de prioridad el metodo decrementarClave(verticeSiguiente, nuevaDistancia) el cual decrementa la distancia de los demás vértices que ya están en la cola