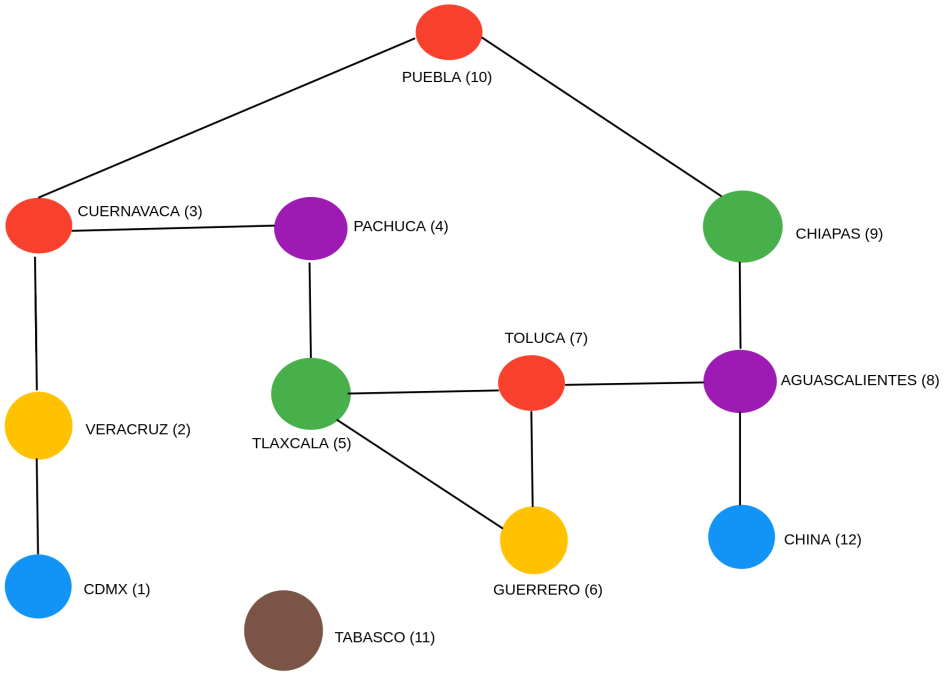
Estructuras de Datos
Proyecto Final: Comunicaciones telefónicas

Bonilla Ruiz Roberto Adrián
Num. Cta. 31721903-8

Fecha de entrega: Miércoles 10 de Febrero 2021

---

1. Se deberá enviar un archivo ***README.pdf*** en donde se justifiquen de forma detallada las decisiones para el diseño del programa.



***ProyectoFinal.xml* en una gráfica visual**

Justificación de decisiones para el diseño del programa: El programa inicia pidiendole al usuario escoger sobre que archivo con extensión **xml** desea trabajar, pues aquí disponemos de dos opciones:

1.– Usar el archivo de ejmplo brindado para el proyecto
2.– Usar el archivo creado por un servidor.

Después de ello se mostrará un segundo menú el cuál indica otras dos opciones:

1.– Realizar una llamada
2.– Salir del programa

Suponiendo que se escogió la opción dos; surgen diferentes tipos de validaciones tales como verificar que ambos números esten asociados a una gráfica, que no puedas llamar a un número que pertenezca a dos personas distintas e incluso que no puedas llamarte a ti mismo. En caso de ingresar de manera correcta todos los datos podrás realizar llamadas ya sea usando **video - fono** ó **llamada tradicional** así el comportamiento del programa varia dependiendo de las desiciones o datos que sean proporcionados.

Descripción: De primera instancia decidí realizar una gráfica no dirigida porque al requerir que ambos vértices estuvieran conectados, era importante mantener las aristas en ambos sentidos, en un inicio decidí irme por la implementación de gráficas usando una lista de adyacencias, sin embargo tras darme cuenta de lo complicado que me llegó a resultar fue que opté por cambiarme a la impementación con una matriz de adyacencias.

Para ello reusé estructuras de datos que se fueron implementadas a lo largo del curso tales como colas y listas ligadas, honestamente en un inicio no tenía una idea muy clara de qué debía hacer como tal, sin embargo volver a ver el video de la sesión donde se explicaba a detalle la resolución del proyecto, me dió un impulso para realizar lo que serían mis primeros pasos con miras a la implementación completa de este proyecto.

Implementación: Sinceramente lo más dificil que surgió fue saber como añadir una arista entre dos vértices ya que al iniciar usando listas, no presté mucha atención a las gráficas usando una matriz, pero nuevamente en ayudantias o sesiones de clase me fue rápido encontrar una solución. Lo segundo más complejo y lo que más me demoró en todo el proyecto fue implementar el método **ruta** ya que ahí fusioné el método **BFS** junto con **el problema del camino más corto y mostrar la ruta**, lo demás fue cuestión de realizar pruebas de escritorio a papel y lápiz.

```
/**
 * Metodo que ejecuta el camino mas corto de
 * una grafica y regresa su ruta
 * @param graph la grafica donde vamos a iterar
 * @param inicial el verticie inicial
 * @param llegada el vertice al cual queremos llegar
 * @return la ruta del verticie inicial al de llegada,
 * null en caso de que no exista.
 */
public static ListaLigada<Integer> ruta(GraficaMatriz<Estacion,Integer> graph,
Estacion inicial, Estacion llegada){

ColaLista<Estacion> cola  = new ColaLista<>();
ListaLigada<Integer> ruta = new ListaLigada<>();
boolean [] visitados      = new boolean [graph.numVertex()+1];
int [] padres             = new int [graph.numVertex()+1];
int [] distancia          = new int [graph.numVertex()+1];

cola.insert(inicial);

visitados[inicial.getId()] = true;
padres[inicial.getId()]    = -1;
while(!cola.isEmpty()){
    Estacion u = cola.delete();
    int id     = u.getId();
    ListaLigada <Estacion> vec = vecinos(graph,u);
    for(int i = 0;i< vec.size();i++){
        Estacion r = vec.get(i);
        int idR    = r.getId();
            if(visitados[idR]==false){
                    visitados [idR] = true;
                    cola.insert(r);
                    distancia [idR] = distancia[id]+1;
                    padres [idR]    = id;
                }
        }
}
int idLlegada = llegada.getId();
        if(!visitados[idLlegada]){
                System.out.println("No existe conexion a la estacion donde se encuentra
                                    No se puede realizar tu llamada)");
                return null;
        }

        for (int j=idLlegada;j!=-1;j=padres[j]){
                        ruta.add(0,j);
        }
return ruta;
}
```

GRACIAS POR TODO
Profesor: Carlos Zerón Martínez
Ayudante: José Antonio Vilchis Salazar
Ayud. Lab. Emmanuel Cruz Hernández
Ayud. Lab. Sara Doris Montes Incin