**MODELO DE SOLUCIÓN DE TRANSPORTE**

Por Sebastián Caballero y Santiago Ortiz

Resumen Ejecutivo:

Actualmente las ciudades de todo el mundo enfrentan un problema en los sistemas de transporte que estas poseen, el embotellamiento producido por la congestión de usuarios en las vías de transporte se vuelve contraproducente en la sociedad. En el modelo que se desarrolló para una ciudad con *n* localidades se enfrente el problema de embotellamiento en los casos de ciclovía, transporte público y ruta vehicular entre las localidades. Las localidades se deben ver cómo nodos en un grafo y los pesos entre los nodos deben ser considerados como la cantidad de personas que usan la vía de transporte que representa la conexión entre los nodos. La propuesta hecha para la solución al embotellamiento en la ciudad está dada para calcular cuánto costaría invertir en las vías para poder tolerar la cantidad de transeúntes. El código recibe n localidades y pregunta al usuario por las características de estas para así poder un análisis exhaustivo entre todos los transeúntes que se mueven, aparte de ello pregunta al usuario el valor máximo de personas que pueden moverse en el tipo de vía para así poder ver si en efecto hay embotellamiento o no, para calcular el precio de las vías y cuántas necesita solo pide al usuario que ingrese cuanto cuesta la vía para k transeúntes.

Funcionalidad de la Herramienta:

La herramienta va a recibir *n* localidades con sus características, estas características son su nombre, sus habitantes, el tipo de rutas que posee, su presupuesto individual, luego recibirá cuantas conexiones de tipo ciclovía, público y vehicular posee (EJ. Si la localidad *w* tiene 3 conexiones de tipo ciclovía, debe estar conectada por ciclovía con otras tres localidades) para así analizar con quienes se establecen las relaciones, después preguntará individualmente cuántas personas se mueven de la localidad hacia una de las mencionadas para empezar a guardar los datos para finalmente cuándo tenga los datos de todas las localidades obtenga los pesos que transcurren por cada conexión entre dos localidades, al obtener los pesos y cómo último dato preguntará cuál es el número máximo de transeúntes que pueden transcurrir por el tipo de vía y cuánto cuesta la vía de este tipo para así finalmente arrojar los datos de qué necesitan algunas conexiones para superar el embotellamiento o si están controladas y no tienen embotellamiento. Alternamente le preguntará si quiere observar la información de todas las localidades que creó. Por lo tanto, el problema del embotellamiento queda solucionado en toda conexión para todo tipo de vía en la ciudad modelada.

Descripción:

Para leer la descripción es recomendable que observe en paralelo con el código. Primeramente observe el archivo *classLocalidad.py* , en este archivo se observa la clase *Localidad* que crea instancias de objetos con atributos de una localidad, esta clase es insuficiente debido a que solo tiene atributos internos, atributos que no le sirven para relacionarse con otros objetos de tipo Localidad con el fin de ver embotellamiento. Para ello en el archivo *relación.py* se crea otra clase llamada *Relacion* que es hija de Localidad, esta clase satisface todo lo que localidad no lograba, posee los atributos de conexiones que son listas de las otras localidades con las que la localidad conecta, los atributos de movimiento que son diccionarios de las localidades con las que conecta y las personas que se mueven a estos y los atributos de traslado que van a servir después para saber cuántas personas salen de la localidad.

El archivo *MetodosAnalisis.py* utiliza las funciones de *ConexionesEntreDosObjetos* para establecer el peso entre un nodo y otro, esto es útil ya que después será usado para poder calcular si hay embotellamiento o no.

El archivo *AlgoritmoEvaluacionGeneral.py* primeramente utilizando la función *PresupuestoGeneral* calcula el presupuesto que posee toda la ciudad para invertir en las vías y solucionar el embotellamiento. Luego utiliza las funciones de *Evaluacion* para analizar para cada tipo de ruta en todas las localidades si existe o no embotellamiento, esta función es el algoritmo usado para analizar TODOS los pesos y ver el embotellamiento donde análogamente si existe el embotellamiento se soluciona dando la cantidad de inversión.

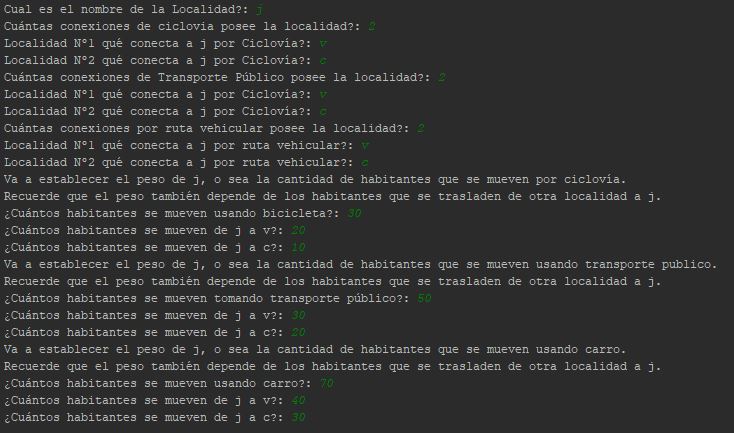
Finalmente, el archivo *ciudadPrincipal.py* es el main (por decirlo así). En el caso del código se dejaron comentados 4 ejemplos. El primero entre dos localidades que se conectan por todas las vías. El segundo es entre tres localidades que también se conectan a través de todas las vías, estos tres ejemplos no desafían tanto algoritmo. En el tercer ejemplo se conectan tres localidades, pero no todas utilizan los tres tipos de vías por lo que este reta un poco más al algoritmo. Finalmente, el cuarto ejemplo tiene cuatro localidades que no se conectan a través de todas las vías entre ellas, este es el ejemplo que demuestra la capacidad del algoritmo para casos en donde halla varias localidades. A continuación, se muestra cómo se idealizó el caso para uno de los ejemplos y que se contestó en la terminal para la resolución del embotellamiento.

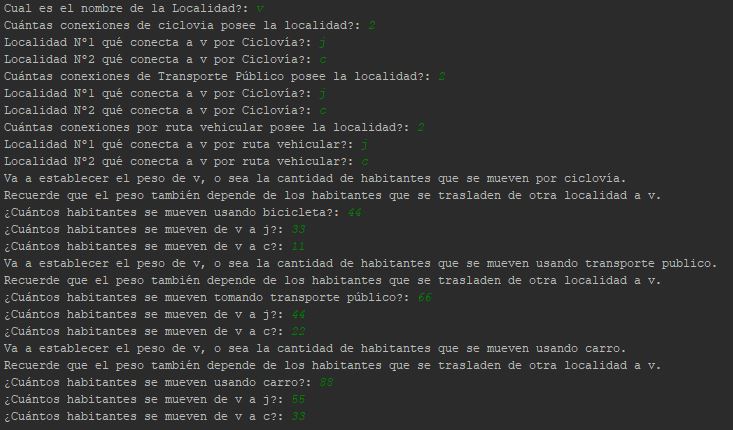
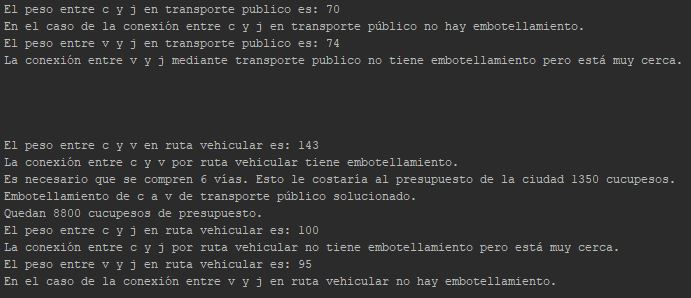
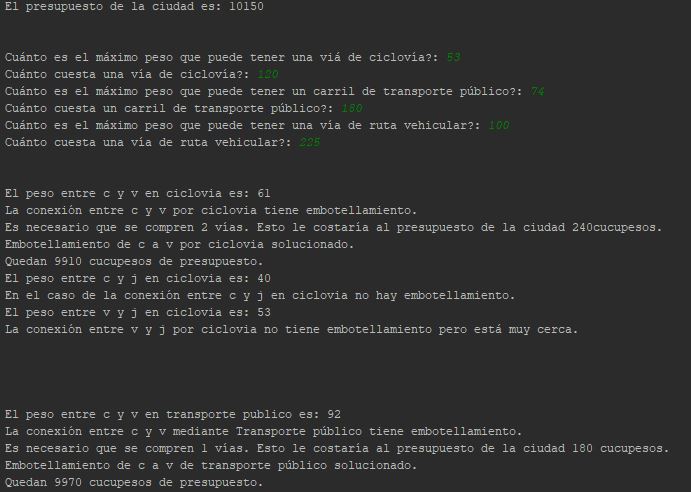
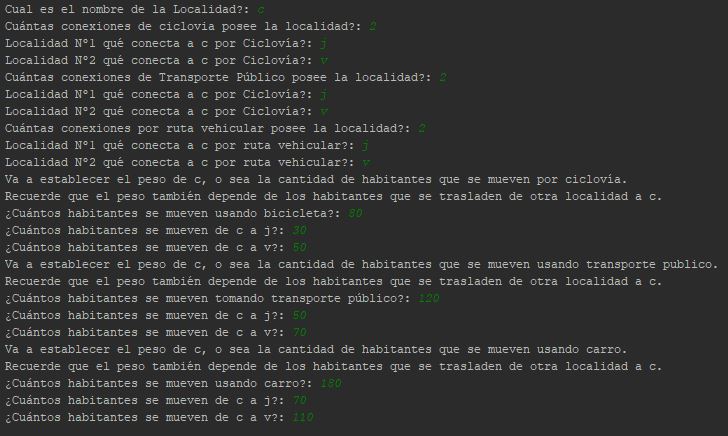
PARA EL EJEMPLO 2 (Quite los comentarios del ejemplo 2):

J

v

c

Las localidades j, v y c están conectadas por los tres tipos de vía. A continuación, se muestra lo que se introdujo en terminal para dar una situación hipotética.

Y así se ve el análisis que hizo el programa para poder analizar el caso de la ciudad. Los movimientos fueron:

|  |  |  |  |
| --- | --- | --- | --- |
| LOCALIDAD | TIPO DE VÍA | LOCALIDAD | TRÁNSITO |
| j | ciclovía, público, vehículo | v | 20, 30, 40 |
| j | ciclovía, público, vehículo | c | 10, 20, 30 |
| v | ciclovía, público, vehículo | j | 33, 44, 55 |
| v | ciclovía, público, vehículo | c | 11, 22, 33 |
| c | ciclovía, público, vehículo | j | 30, 50, 70 |
| c | ciclovía, público, vehículo | v | 50, 70, 110 |