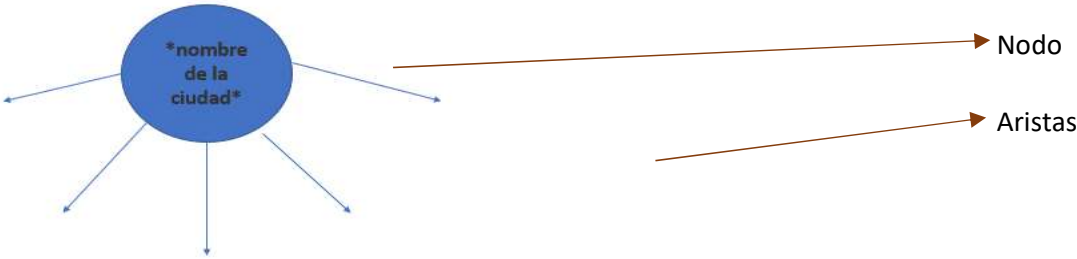


TAD NODO



{inv:
Aristas>=51
Peso arista!=null

Operaciones primitivas:

Node:	String	
get_adjacent:	String(city)	Node
get_adjacentDistance:	String(city)	int
get_lowestDistance:		Node
set_id:	String(city)	
set_distance:	String(city) x int(distance)	

Node(city)
Crea un nodo de al menos 51 aristas
{pre: TRUE}
{post: nodo={aristas:" "}}
get_adjacent(city)
Retorna el nodo adyacente, al cual le corresponda la ciudad ingresada
{pre: city ∈ String}
{post: Nodo[city]}
set_id(city):
Cambia el identificador del nodo
{pre: city ∈ String}
{pos: node.id = city}

get_adjacentDistance(city)
Retorna la distancia que hay entre este nodo y un vecino
{pre: city ∈ aristas}
{post: distancia(entero)}
get_lowestDistance()
Retorna el valor de la distancia más corta de entre todas las aristas del nodo
{pre: none}
{post: arista[].distancia}
set_distance(city,distance):
Establece la distancia hasta el nodo(city)
{pre: city ∈ String, distance ∈ int }
{post: [(if <nodo(ciudad) ∈ aristas[>):TRUE) (else: FALSE)]}

[illegible]