Graph

“contiene aristas y vertices”

deleteEdge(name1,name2)--🡪void

addEdge(name1,name2)---🡪void

deleteVertex(String name)-🡪void

addVertex(String name)-----🡪String

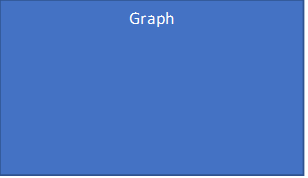
Dijkstra:source int-----🡪String

BFS: root\_node----🡪void

Kruskal: List-----------🡪void



Gráfico de rectángulos

Descripción generada automáticamente con confianza baja

addVertex(String name)-----🡪String adeleteVertex(String name)-🡪void

“creates an specific vertex and add it “delete vertex who exist in the arraylist”

into the vertex aarraylist” {pre:the vertex exist in the arraylist}

{pre: the vertex not rxist i the arraylist} {pos:vertex deleted}

{pos:vertex add}

Gráfico de rectángulos

Descripción generada automáticamente con confianza bajaGráfico de rectángulos

Descripción generada automáticamente con confianza baja

addEdge(name1,name2)---🡪void deleteEdge(name1,name2)--🡪void

“créate a new edge in the arraylist” “delete a Edge in the arraylist”

{ pre:the Edge dont exist} {pre:the Edge exist}

{post:Edge add} {post:Vertex added}

Dijkstra:source int-----🡪String

“found the minimun way from point a to b ”

{pre:have to exist a undirected weighted graph }

{pos:show the minimun way}

BFS: root\_node----🡪void

“traverse or search for elements in a graph”

{pre:have to exist a undirected weighted graph:}

{pos:show if the graph is related }

Kruskal: List-----------🡪void

“traverse or search for elements in a graph”

{pre:have to exist a undirected weighted graph:}

{posshow the minimun way passing for all the vertex:}