

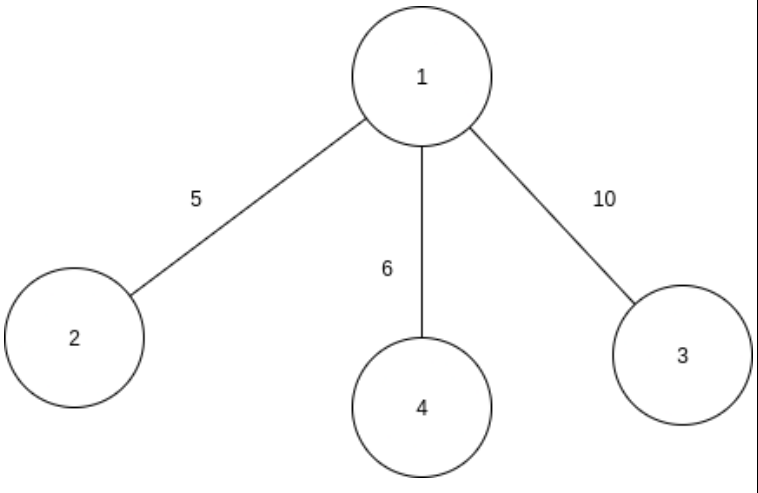
Camilo Campaz Jimenez  
 Daniel Esteban Jaraba  
 Gabiria Johan Stiven  
 Ricardo Sibaja

## DISEÑOS DE PRUEBAS UNITARIAS

### Clase de gráfico:

Nombre	Clase	Etapas
configuración1	Gráfico	Un grafo no dirigido sin vértices

Nombre	Clase	Etapas
configuración2	Gráfico	Grafo grafo vértices del grafo $=\{1,2,3,4\}$ Grafo dirigido = falso

Nombre	Clase	Etapas
configuración3	Gráfico	

Nombre	Clase	Etapas
--------	-------	--------

configuración4	Gráfico	Gráfico gráfico Vértices del grafico = {1,2,3} gráfico. Dirigido=verdadero
----------------	---------	---

Nombre	Clase	Etapas
configuración 5	Gráfico	

**Objetivo de la prueba:** Verificar si el método `addVertice()` es capaz de establecer nuevos vértices en el grafo en un contexto específico

Clase	Método	Etapas	Valores de entrada	Resultado
Gráfico	<code>addVertice()</code>	configuración 1	clave = 1	Verdadero, lo que significa que el vértice se agregó al gráfico. Esto se verifica buscando el valor del nuevo vértice en el gráfico

**Objetivo de la prueba:** Verificar si el método `existVertice()` es capaz de establecer nuevos vértices en el grafo en un contexto específico

Clase	Método	Etapas	Valores de entrada	Resultado
Gráfico	<code>existVertice()</code>	configuración 1	clave = 8	Cierto, lo que significa que el valor no pertenece al conjunto de vértices del gráfico

**Objetivo de la prueba :** Comprobar Si el método `addEdge()` es capaz de establecer nuevas aristas en el gráfico en un contexto específico

Clase	Método	Etapas	Valores de entrada	Resultado
Gráfico	addEdge()	configuración 2	sourceKey = 1 endKey = 2 peso = 5	Cierto, lo que significa que si consultamos los nuevos valores de la arista en la gráfica podremos encontrarlos

**Objetivo de la prueba:** Verificar si el método addEdge() es capaz de establecer nuevas aristas en el gráfico en un contexto específico

Clase	Método	Etapas	Valores de entrada	Resultado
Gráfico	addEdge()	configuración 4	sourceKey = 1 endKey = 2 peso = 5	Cierto, lo que significa que si consultamos los nuevos valores de la arista en la gráfica podremos encontrarlos

**Objetivo de la prueba:** Verificar si el método edit() es capaz de establecer nuevos valores para cualquier vértice o peso en el gráfico en un contexto específico

Clase	Método	Etapas	Valores de entrada	Resultado
Gráfico	editar()	configuración 3	clave = 1 newValue = 0	Falso, lo que significa que si buscamos el valor antiguo para el vértice no existirá ya que ese valor se actualizó

**Objetivo de la prueba:** Verificar si el método edit() es capaz de establecer nuevos valores para cualquier vértice o peso en el gráfico en un contexto específico

Clase	Método	Etapas	Valores de entrada	Resultado
Gráfico	editar()	configuración 3	clave = 1 Nuevo valor = 20 valor antiguo = 10	Verdadero, lo que significa que se ha actualizado el valor seleccionado

**Objetivo de la prueba:** Verificar si el método prim() es capaz de encontrar la ruta más corta

Clase	Método	Etapas	Valores de entrada	Resultado
-------	--------	--------	--------------------	-----------

Gráfico	prime()	configuración n5	clave = 1	Verdadero, lo que significa que la comparación de la ruta esperada es la misma que la ruta dada por el código
---------	---------	---------------------	-----------	---

<b>Objetivo de la prueba:</b> <b>Verificar si el método floyWarshall() es capaz de encontrar el entre cada par de vértices</b>				
Clase	Método	Etapas	Valores de entrada	Resultado
Gráfico	<b>floyWarshall()</b>	configuración n3	V = 4 s = 1 v = 3	Verdadero, lo que significa que la comparación de la ruta esperada es la misma que la ruta dada por el código

## Administrador de clase:

Nombre	Clase	Etapas
configuración1	Director	Esta clase tiene un grafo conectado con 56 vértices que representan todas las estaciones MIO con su respectivo peso. Toda esta información es dada por un archivo csv

**Objetivo de la prueba:** Verificar si el método `bestRoute()` es capaz de encontrar la mejor ruta de una estación a otra

Clase	Método	Etapas	Valores de entrada	Resultado
Gestión	<code>bestRoute()</code>	configuración1	valor1 = 26 valor2 = 43	Verdadero, lo que significa que la comparación de la ruta esperada es la misma que la ruta dada por el código