## CASOS DE PRUEBA

## Configuración:

Nombre	Clase	Escenario
setupScenary1	ButterflyTest	Colombia
setupScenary2	ButterflyTest	Dubai \$3000 Portugal \$2750 \$1200 S590 Madagascar
setupScenary3	ButterflyTest	España  \$3000 \$5000  Japón  Rusia \$1000  \$1000  \$1000  \$1930  Dubai  Portugal  Madagascar

## Diseño de casos de prueba para el grafo representado con matrices de adyacencia:

Objetivo de la	Objetivo de la prueba: Verificar que se agrega un vértice correctamente al grafo.					
Clase	Método	Escenario	Valores de entrada	Resultado		
MatrixGraph	addVertice	setupScenary1	Vértice = "Japón"	Colombia		
MatrixGraph	addVertice	setupScenary1	Vértice = "Portugal"	Colombia		

**Objetivo de la prueba:** Verificar que se agrega una arista correctamente al grafo de peso no negativo.

Clase	Método	Escenario	Valores de entrada	Resultado
MatrixGraph	addEdge	setupScenary2	Vértice inicial = "Portugal" Vértice final = "Colombia" Costo = 1600	\$3000 Portugal \$1600 \$2750 \$1200 Colombia Madagascar
MatrixGraph	addEdge	setupseemary2	Vértice inicial = "Madagascar" Vértice final = "Colombia" Costo = -200	No se pudo añadir la arista al grafo, ya que su costo es un número negativo.

Objetivo de la prueba: Verificar que se elimina un vértice correctamente del grafo.						
Clase	Método	Escenario	Valores de entrada	Resultado		
MatrixGraph	deleteVertice	setupScenary2	Vértice = "Dubai"			

			S2750 Colombia Madagascar
MatrixGraph	deleteVertice	setupScenary1	Arraylist de vértices vacía.

Objetivo de la prueba: Verificar que el algoritmo bfs funciona correctamente						
Clase	Método	Escenario	Valores de entrada	Resultado		
MatrixGraph	bfs	setupScenary2	Vértice inicial =" Colombia"	Colombia Dubai Portugal Madagascar		
MatrixGraph	bfs	setupScenary3	Vértice inicial =" Colombia"	Colombia Espira EEUU Japon Asintala Nigeria Russa Dubisi Portugal Melegacori		

Objetivo de la prueba: Verificar que el algoritmo dfs funciona correctamente					
Clase	Método	Escenario	Valores de entrada	Resultado	
MatrixGraph	dfs	setupScenary3		España EEUU Coombia Japon Australia Noperia Rusia Dubai Perlugial Medaglascar  Size=10	
Objetivo de l	a prueba: `	Verificar que se	elimina una	arista correctamente del grafo.	
Clase	Método	Escenario	Valores de entrada	Resultado	
MatrixGraph	deleteEdge		Vértice inicial = "Colombia" Vértice final = "Dubai" Costo = 1200	Dubal \$3000 Portugal \$2750	

MatrixGraph deleteEdge		Vértice inicial = "Dubai" Vértice final = "Portugal" Costo = 3000	Dubai Portugal \$2750 \$590  Colombia  Madagascar
------------------------	--	---	---

Objetivo de la	Objetivo de la prueba: Validar que se modificó una adyacencia dentro del grafo.						
Clase	Método	Escenario	Valores de entrada	Resultado			
MatrixGraph	_	setupScenary2	Vértice inicial = "Dubai" Vértice final = "Portugal" Costo = 2130	Dubai \$2130 Portugai \$2750 \$1200 S590 Madagascar			
MatrixGraph	setEdge	setupScenary2	Vértice inicial = "Colombia" Vértice final = "Dubai" Costo = 3000	Dubai \$3000 Portugal \$2750 \$3000 Colombia Madagascar			

Objetivo de la prueba: Verificar que el algoritmo Dijkstra funciona correctamente.					
Clase	Método	Escenario	Valores de entrada	Resultado	
MatrixGraph	makeDijsktra	setupScenary3	Vértice = " Portugal" Vértice = "Japón"	\$6100 – El costo mínimo que hay en el vuelo de Colombia a Japón.	
MatrixGraph	makeDijsktra		vertice – hun	[∞, ∞, \$6100, \$9100, \$2100, \$5100, \$4600, \$0, \$3350, \$8350]	

	Donde los precios corresponden a los
	siguientes países
	respectivamente:
	Colombia, España,
	Japón, EEUU, Rusia,
	Australia, Nigeria,
	Portugal, Dubai y
	Madagascar.

Objetivo de la prueba: Verificar que el algoritmo Dijkstra NO encuentra valores mínimos cuando no hay vuelos entre dos países.

Clase	Método	Escenario	Valores de entrada	Resultado
				No se encontró un
MatrixGraph	trixGraph makeDijsktra		Vértice = "Madagascar"	costo mínimo en el
			Vértice = "Dubai"	vuelo de Madagascar
				a Dubái.
				No se encontró un
MatrixGraph	MatrixGraph makeDijsktra	setupScenary3	Vértice = "España"	costo mínimo en el
		Vértice = "Andorra"	vuelo de España a	
				Andorra.

Objetivo de la prueba: Verificar que el algoritmo Prim funciona correctamente.								
Clase	Método	Escenario	Valores de entrada	Resultado				
MatrixGraph	prim	setupScenary3	Vértice = " Colombia"	\$28370 – El costo mínimo que hay en el vuelo de Colombia a todos los demás países.				
MatrixGraph	prim	setupScenary2	Vértice = "Colombia"	\$6950 – El costo mínimo de ir desde Colombia a todos los demás países.				

Objetivo de la prueba: Verificar que el algoritmo Prim NO encuentra el costo mínimo de

viajar por todos los países si hay países a los que no se puede viajar.

rajar por todos ros parses si hay parses a ros que no se paede viajar.								
Clase	Método	Escenario	Valores de entrada	Resultado				
				\$1930 – Solo se				
MatrixGraph	prim	setupScenary3	Vértice = "Madagascar"	encontró el valor de ir				
				de Madagascar a				
				Japón, ya que no se				
				podía viajar hacia				
				más países.				
				No se encontró un				
MatrixGraph	prim	setupScenary3	Vértice = "Rusia"	costo mínimo				

		teniendo como país
		de origen a Rusia.

**Objetivo de la prueba:** Verificar que el algoritmo de floyd-warshall acomoda de forma correcta los costos de cada par de vértices.

Clase	Método	Escenario	Valores de entrada	Resultado										
		Se organizó correctamente la matriz con los costos d							de					
				cada par de vértices obteniendo el siguiente resultado.										
						España	Japón	EEUU	Rusia	Australia		Dubai		Madagascar
				Colombia	0	3000	8000	700	10200	7700	10700	8950	11620	13950
ListGraph	floyd	setupScenary3		España	0	0	5000	0	0	0	0	0	0	0
1	,	1 3		Japón EEUU	0	0	11000	0	9500	7000	10000	0 8250	10920	13250
				Rusia	0	0	0	0	0	0	0	0	0	0
			Australia	0	0	4000	7000	2500	0	3000	1250	3920	6250	
			Nigeria	0	0	1000	4000	5520	3020	0	4270	920	9270	
				Dubai	0	0	6930	0	0	0	0	0	0	5000
				Portugal Madagascar	0	0	6100 1930	9100 0	4600 0	2100	5100 0	3350 0	0	8350
ListGraph	floyd	setupScenary1		La mat iniciali					_					
				Se org	anizó	cor	rectar	nente	e la n	natriz	con	los c	ostos	de
				cada p	ar de	vert	ices (	obten	ienac	er si	iguiei	ne re	esuma	ao.
							Dubai	Pc	rtuga	al Co	lomb	ia N	/ladag	ascar
ListGraph	floyd	setupScenary2		Du	bai		0		3000		0		575	50
				Port	ugal		3340		0		0		275	50
				Colo	mbia		1200		4200		0		695	50
				Mada	gasca	r	590	,	3590		0		0	

Objetivo de la prueba: Validar que el algoritmo de kruskal calcula el menor recorrido entre todos los vértices

Clase	Método	Escenario	Valores de entrada	Resultado
				Retorna el
				valor de
ListGraph	kruskal	setupScenary3		mínimo de
				recorrer todos
				los países
				int = 17.400
ListGraph	kruskal	setupScenary1		No hay un
				grafo en el
				cuál trabajar
				int = 0
				Retorna el
				valor mínimo
ListGraph	kruskal	setupScenary2		de recorrer
				todos los
				países.
				int = 4.540

## Diseño de casos de prueba para el grafo representado con listas de adyacencia:

Objetivo de la prueba: Verificar que se agrega un vértice correctamente al grafo.							
Clase	Método	Escenario	Valores de entrada	Resultado			
ListGraph	addVertice	setupScenary1	Vértice = "Nigeria"	Colombia			
ListGraph	addVertice	setupScenary1	Vértice = "Madagascar"	Colombia			

Objetivo de la prueba: Verificar que se agrega una arista correctamente al grafo.							
Clase	Método	Escenario	Valores de entrada	Resultado			
ListGraph	addEdge	setupScenary2	Vértice inicial = " Colombia" Vértice final = "Portugal" Costo = 1600	Dubal \$3000 Portugal \$2750 \$1200 S1600 Madagascar			
ListGraph	addEdge	setupScenary2	Vértice inicial = " Dubai" Vértice final = "Madagascar" Costo = 780	Dubai \$3000 Portugal \$2750 \$1200 \$590 Colombia \$780 Madagascar			

Objetivo de la	Objetivo de la prueba: Verificar que se elimina un vértice correctamente del grafo.							
Clase	Método	Escenario	Valores de entrada	Resultado				
ListGraph	deleteVertice	setupScenary2	Vértice = "Dubai"	Portugal \$2750  Colombia Madagascar				
ListGraph	deleteVertice	setupScenary1	Vértice = "Colombia"	El arraylist de vértices está vacio.				

Objetivo de	<b>Dbjetivo de la prueba:</b> Verificar que se elimina una arista correctamente del grafo.							
Clase	Método	Escenario	Valores de entrada	Resultado				
ListGraph	deleteEdge	setupScenary2	Vértice inicial = "Colombia" Vértice final = "Dubai" Costo = 1200	Dubal \$3000 Portugal \$2750 \$590  Madagascar				
ListGraph	deleteEdge	setupScenary2	Vértice inicial = "Dubai" Vértice final = "Portugal" Costo = 3000	Dubal Portugal S2750 \$1200 \$590 Madagascar				

Objetivo de la prueba: Validar que se modificó una adyacencia dentro del grafo.								
Clase	Método	Escenario	Valores de entrada	Resultado				
ListGraph	setEdge		Vértice inicial = "Portugal" Vértice final = "Madagascar" Costo = 1320					

			Dubal \$3000 Portugal \$1320 \$1200 Colombia Madagascar
ListGraph	setEdge	Vértice inicial = "Madagascar" Vértice final = "Dubai" Costo = 3000	Dubal \$3000 Portugal \$2750 \$1200 Colombia Madagascar

Objetivo de la	Objetivo de la prueba: Verificar que el algoritmo Dijkstra funciona correctamente.								
Clase	Método	Escenario	Valores de entrada	Resultado					
ListGraph	makeDijsktra	setupScenary3	Vértice = " Portugal" Vértice = "Japón"	\$6100 – El costo mínimo que hay en el vuelo de Colombia a Japón.					
ListGraph	dijsktra	setupScenary3	Vértice = "Portugal" Vértice = "null"	[∞, ∞, \$6100, \$9100, \$4600, \$2100, \$5100, \$3350, \$0, \$8350]					
				Donde los precios corresponden a los siguientes países respectivamente: Colombia, España, Japón, EEUU, Rusia, Australia, Nigeria, Portugal, Dubai y Madagascar.					

Objetivo de la prueba: Verificar que el algoritmo Dijkstra NO encuentra valores mínimos

cuando no hay vuelos entre dos países.

Resultado
No se encontró un costo mínimo en el vuelo de Madagascar a Dubái.
No se encontró un costo mínimo en el vuelo de España a Andorra.

Objetivo de la prueba: Verificar que el algoritmo Prim funciona correctamente.											
Clase	Método	Escenario	Valores de entrada	Resultado							
ListGraph	prim	setupScenary3	Vértice = " Colombia"	\$28370 – El costo mínimo que hay en el vuelo de Colombia a todos los demás países.							
ListGraph	prim	setupScenary2	Vértice = "Colombia"	\$6950 – El costo mínimo de ir desde Colombia a todos los demás países.							

Objetivo de la prueba: Verificar que el algoritmo Prim NO encuentra el costo mínimo de

viajar por todos los países si hay países a los que no se puede viajar.

Clase	Método	Escenario	Valores de entrada	Resultado
				\$1930 – Solo se
ListGraph	prim	setupScenary3	Vértice = "Madagascar"	encontró el valor de ir
				de Madagascar a
				Japón, ya que no se
				podía viajar hacia
				más países.
				No se encontró un
ListGraph	prim	setupScenary3	Vértice = "Rusia"	costo mínimo
				teniendo como país
				de origen a Rusia.

Objetivo de la prueba: Verificar que el algoritmo de floyd-warshall acomoda de forma correcta los

costos de cada par de vértices.

Clase Método Escenario Valores de entrada	Resultado
---	-----------

				rganiz									
			cada	ı par de	e vert	ices (	obten	ienac	o er si	guier	ite re	esuita	ao.
				Colombia	España	Japón	EEUU	Rusia	Australia	Nigeria	Dubai	Portugal	Madagascar
			Colom	oia 0	3000	8000	700	10200	7700	10700	8950	11620	13950
ListGraph	floyd	setupScenary3	Espaí	a 0	0	5000	0	0	0	0	0	0	0
ListOraphi	noya	petapseemary 5	Japó	ո 0	0	0	0	0	0	0	0	0	0
			EEUI	0	0	11000	0	9500	7000	10000	8250	10920	13250
			Rusi	a 0	0	0	0	0	0	0	0	0	0
			Austra		0	4000	7000	2500	0	3000	1250	3920	6250
			Niger	_	0	1000	4000	5520	3020	0	4270	920	9270
			Duba	i 0	0	6930	0	0	0	0	0	0	5000
			Portug	al 0	0	6100	9100	4600	2100	5100	3350	0	8350
			Madaga	scar 0	0	1930	0	0	0	0	0	0	0
ListGraph	floyd	setupScenary1		natriz ı alizada		-		_					
			Se o	rganiz	ó cor	rectai	mente	e la n	natriz	con l	os c	ostos	de
			cada	ı par de	e vért	ices (	bten	iendo	o el si	guier	te re	esulta	do.
						Dubai	Po	rtuga	al Col	lombi	a N	⁄ladag	ascar
ListGraph	floyd	setupScenary2		Dubai		0		3000		0		5750	
			P	Portugal		3340		0		0		2750	
			Co	lombia		1200		4200		0		695	0
			Ma	dagasca	ar	590		3590		0		0	

**Objetivo de la prueba:** Validar que el algoritmo de kruskal calcula el menor recorrido entre todos los vértices

Clase	Método	Escenario	Valores de entrada	Resultado
				Retorna el
				valor de
ListGraph	kruskal	setupScenary3		mínimo de
				recorrer todos
				los países
				int = 17.400
ListGraph	kruskal	setupScenary1		No hay un
				grafo en el
				cuál trabajar
				int = 0
				Retorna el
				valor mínimo
ListGraph	kruskal	setupScenary2		de recorrer
				todos los
				países.
				int = 4.540

Objetivo de la prueba: Verificar que el algoritmo bfs funciona correctamente											
Objetivo de la prueba: Verificar que el algoritmo bfs funciona correctamente         Clase       Método       Escenario       Valores de entrada       Resultado				Resultado							

ListGraph	bfs	setupScenary2	Vértice inicial =" Colombia"	Colombia	Dubai	Portugal	Madaga	scar	
ListGraph	bfs	setupScenary3	Vértice inicial =" Colombia"	Colombia Espella	EEUU Japon	Australia Nigeria	Rusia Dubai	Portugal	Madagascar

Objetivo de la prueba: Verificar que el algoritmo dfs funciona correctamente													
Clase	Método	Escenario	Valores de	Resultado									
			entrada										
ListGraph	dfs	setupScenary3		Colombia España EEUU Japon Australia Nigera Rusia Dubai Pert Size=10					Portugal	Madagascar			