

# Documentacin del Namespace Graph

Tu Nombre

July 31, 2023

## Contents

### 1 Introduccin

Este documento proporciona documentacin para el espacio de nombres **Graph** y la clase **MainWindow**. El cdigo presentado a continuacin define una estructura de datos bsica de grafo y utiliza el algoritmo de Dijkstra para encontrar el camino ms corto entre dos vrtices.

### 2 Clase MainWindow

#### 2.1 Descripcin de la Clase

La clase **MainWindow** es responsable de mostrar un grafo y encontrar el camino ms corto entre dos vrtices utilizando el algoritmo de Dijkstra.

#### 2.2 Cdigo Fuente

```
[language=[Sharp]C] namespace Graph /// ¡summary! /// Lgica de interaccin  
para MainWindow.xaml /// ¡/summary! public partial class MainWindow :  
Window // Miembros de clase y mtodos...
```

#### 2.3 Miembros de Clase y Mtodos

- Campos

- `private Graph graph`: Representa la estructura de datos del grafo.

- Constructor

- `public MainWindow()`: Inicializa **MainWindow**, crea el grafo, agrega vrtices y aristas, dibuja el grafo y encuentra el camino ms corto usando el algoritmo de Dijkstra.

- Clases Anidadas

- `public class Vertex`: Representa un vrtice en el grafo con un `Id`, coordenada `X` y coordenada `Y`.
- `public class Graph`: Representa la estructura de datos del grafo con un `AdjacencyList` para almacenar vrtices y sus vecinos con los pesos asociados.

- **Mtodos**

- `private void DrawGraph(Graph graph)`: Dibuja el grafo en el lienzo, incluyendo vrtices como elipses y aristas como lineas con pesos.
- `public List<int> DijkstraShortestPath(Graph graph, int startVertexId, int endVertexId)`: Encuentra el camino ms corto entre dos vrtices usando el algoritmo de Dijkstra y devuelve una lista de IDs de vrtices en el camino.
- `private void DisplayShortestPath(List<int> shortestPath)`: Muestra el camino ms corto en el `shortestPathTextBlock` `TextBlock`.

## 3 Clase Vertex

### 3.1 Descripcin de la Clase

La clase `Vertex` representa un vrtice en el grafo con un `Id`, coordenada `X` y coordenada `Y`.

### 3.2 Cdigo Fuente

```
[language=[Sharp]C] public class Vertex { public int Id { get; } public double X { get; } public double Y { get; }
    public Vertex(int id, double x, double y) { Id = id; X = x; Y = y; }
```

## 4 Clase Graph

### 4.1 Descripcin de la Clase

La clase `Graph` representa una estructura de datos de grafo con un `AdjacencyList` para almacenar vrtices y sus vecinos con los pesos asociados.

### 4.2 Cdigo Fuente

```
[language=[Sharp]C] public class Graph { public Dictionary<int, List<Vertex neighbor, int weight>>> AdjacencyList { get; }
    public Graph() { AdjacencyList = new Dictionary<int, List<Vertex, int>>(); }
    // Agregar vrtices y aristas segn sea necesario public void AddVertex(int vertexId, double x, double y) { // Implementacin... }
    public void AddEdge(int sourceId, int destinationId, int weight) { // Implementacin... }
```

## 5 Conclusin

El cdigo proporcionado define una estructura de datos bsica de grafo y demuestra cmo encontrar el camino ms corto entre dos vrtices utilizando el algoritmo de Dijkstra. La clase MainWindow maneja la visualizacin del grafo y la interaccin con el usuario.