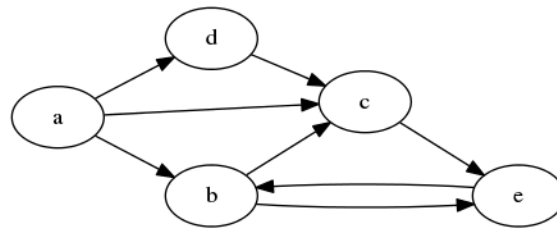


- *ad hoc* puro: Nesta, a comunicação se dá apenas de veículo para veículo (V2V), sua principal vantagem é o baixo custo, já que não é necessário gastar com infraestruturas externas. O ponto negativo desta arquitetura é que sua conectividade que é afetada pela mobilidade dos veículos.
- Infra estruturada: As arquiteturas infra estruturadas são caracterizadas por manter uma comunicação entre veículo e infraestruturas que se localizam as margens das vias. Neste tipo, ao contrário do *ad hoc* puro, há uma maior conectividade da rede, tendo em vista que os RSU (*Road Side Units*), são nós estacionados que trocam informações com os veículos. A desvantagem é o custo para implementação.
- Híbrida: Na arquitetura híbrida, há os dois tipos de comunicação (Comumente chamado de V2X). Neste a comunicação tenta mesclar as duas arquiteturas para obter o melhor. Assim, uma quantidade mínima de RSUs fica fixada para aumentar a conectividade da rede sem elevar o seu custo.

## 2.3 Grafos

Em ciência da computação, a teoria dos grafos é o estudo dos grafos, uma estrutura matemática usada para modelagem de relações entre pares de objetos de uma certa coleção. Sendo assim, um grafo refere-se a uma coleção de vértices conetados por arestas. (RIAZ; ALI, 2011)

Figura 1 – Exemplo de grafo.



Existem atualmente na literatura, várias maneiras para se representar um grafo. As duas mais usadas são usando linhas e círculo, como na Figura 1 (2.3), ou ainda usando uma matriz que contenha todos os vértices e os pesos, como na Figura 2 (2.3).

Figura 2 – Exemplo de grafo representado por matriz de pesos.

	a	b	c	d	e
a	0	5	9	5	0
b	0	0	4	0	10
c	0	0	0	3	4
d	0	0	0	0	4
e	0	10	0	0	0

Segundo, Riaz e Ali (2011), existem ainda dois tipos de representações em matrizes: matriz de incidência e matriz de adjacência. Na matriz de incidência, o grafo é representado por uma matriz onde é verificada a participação dos vértices em tais arestas, enquanto que na matriz de adjacência, são examinados a quando tais vértices formam grafos.