

# Representação Gráfica (*Graph Drawing*)

Teoria dos Grafos,  
Prof<sup>a</sup> Patrícia D. L. Machado, UASC/UFCG

# Representação Gráfica

- Um grafo pode ser representado de diferentes maneiras, desde que estas representações retratem fielmente sua definição formal.
- No entanto, ***o arranjo de vértices e arestas em uma representação afeta a compreensão, usabilidade, construção e estética.***
- Representação gráfica de grafos (***Graph drawing***) combina métodos de teoria dos grafos e visualização de informações para derivar descrições em duas dimensões de grafos que representam aplicações tais como análise de redes sociais, cartografia, linguística, bioinformática, etc.

# Conteúdo desta Aula

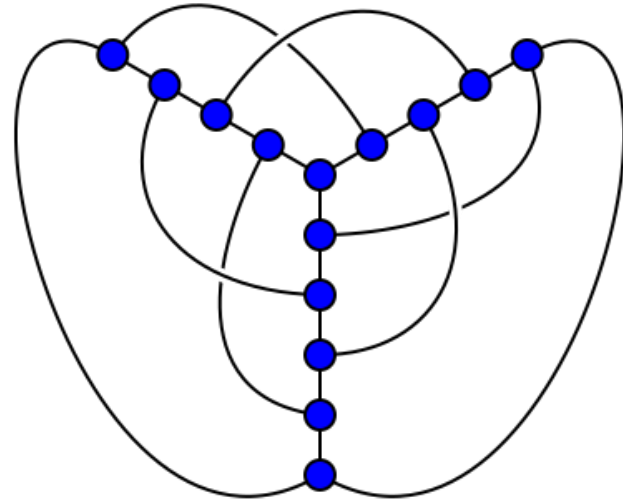
- Métricas
- Layouts

# Métricas de Qualidade

- Medidas objetivas usadas para avaliar a estética e a usabilidade de uma representação.
- Métodos de layout tentam otimizar medidas específicas.

# Crossing Number

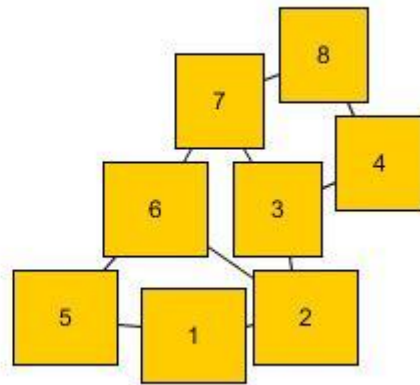
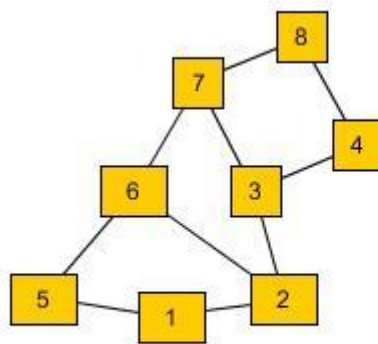
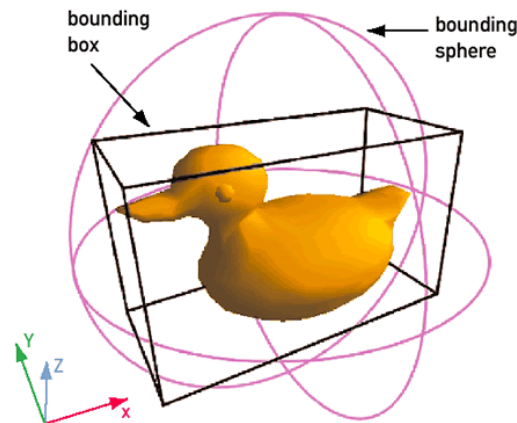
- O ***crossing number***  $cr(G)$  de um grafo  $G$  é o menor número de cruzamento de arestas possível em uma representação de  $G$  no plano.
- Estudos mostram que representações com poucos cruzamentos são melhor compreendidas.



# Área

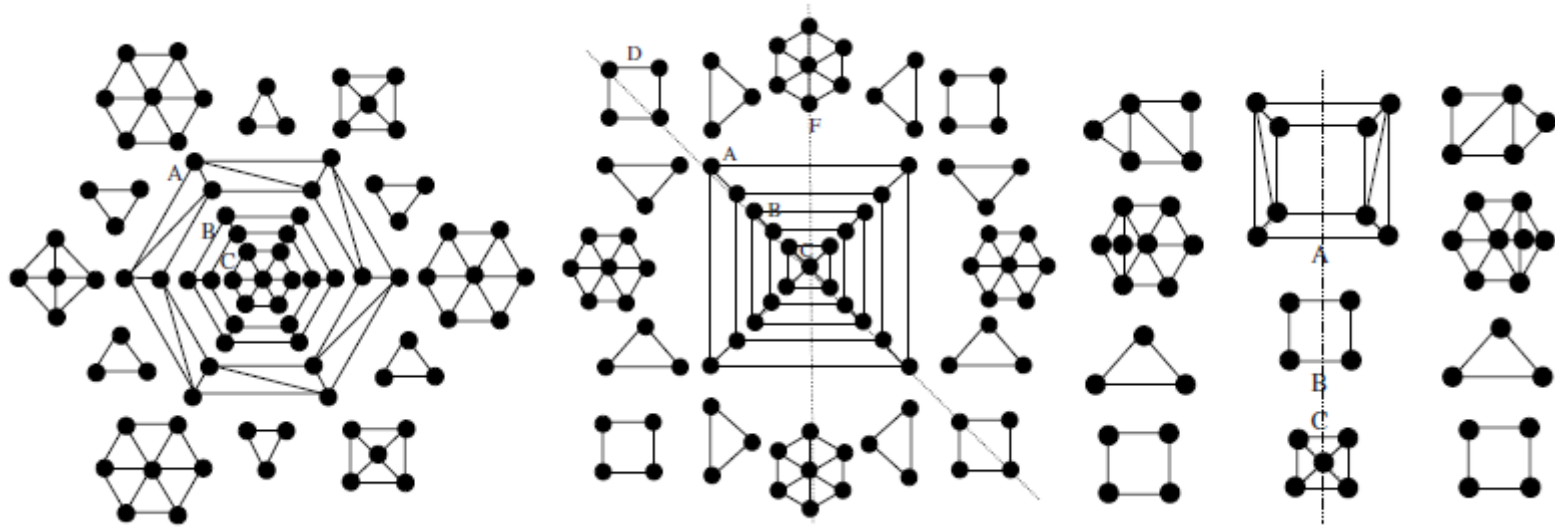
A área de uma representação é o tamanho de sua menor caixa delimitadora (*bounding box*), em relação à menor distância entre quaisquer 2 vértices.

Representações com área menor são geralmente preferíveis àquelas com área maior, porque permitem que as características do desenho sejam mostradas de forma mais legível.



# Grupos de Simetria (*Symmetry Groups*)

Simetria entre grupos de vértices de um grafo deve ser representada tanto quanto possível.

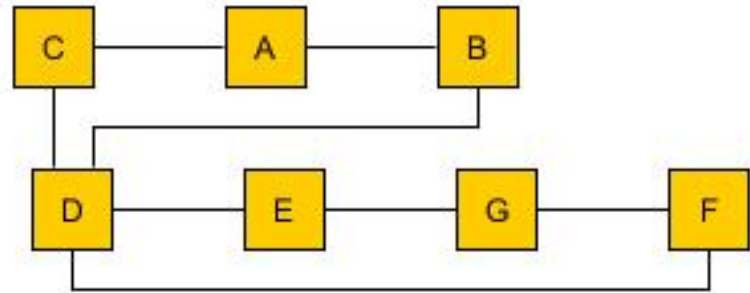


# Número de dobras (*bends*) e Tamanho de Aresta

A complexidade da representação de uma aresta pode ser medida pelo seu número de dobras (bends).

Minimizar o número de dobras totais na representação tem como objetivo deixar uma aresta tão simples quanto possível de ser entendida.

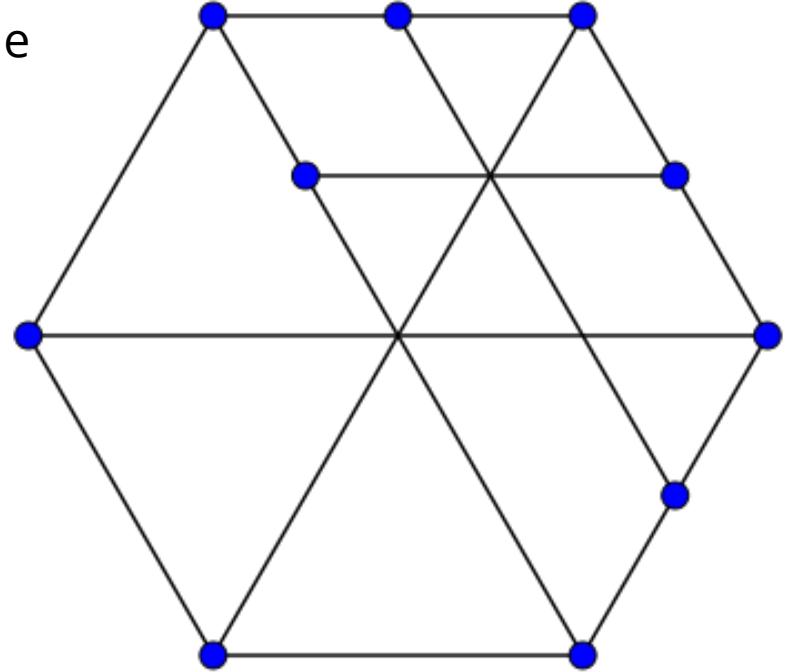
Também é importante ter tamanhos uniformes.



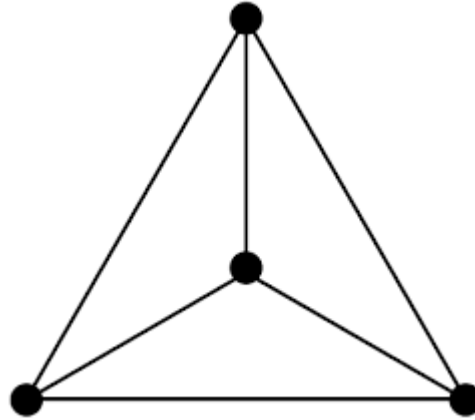
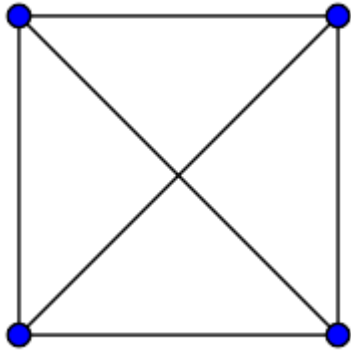


# Número de Inclinação (*Slope Number*)

Número mínimo de inclinações distintas de arestas que são necessárias em uma representação com linhas retas, permitindo cruzamentos.



# Métricas podem ser conflitantes



# Métodos de Layout

Visam produzir um arranjo de vértices e arestas em uma representação que facilite a ***compreensão, usabilidade, construção e estética do grafo.***

---

# Métodos de Layout: Force-based

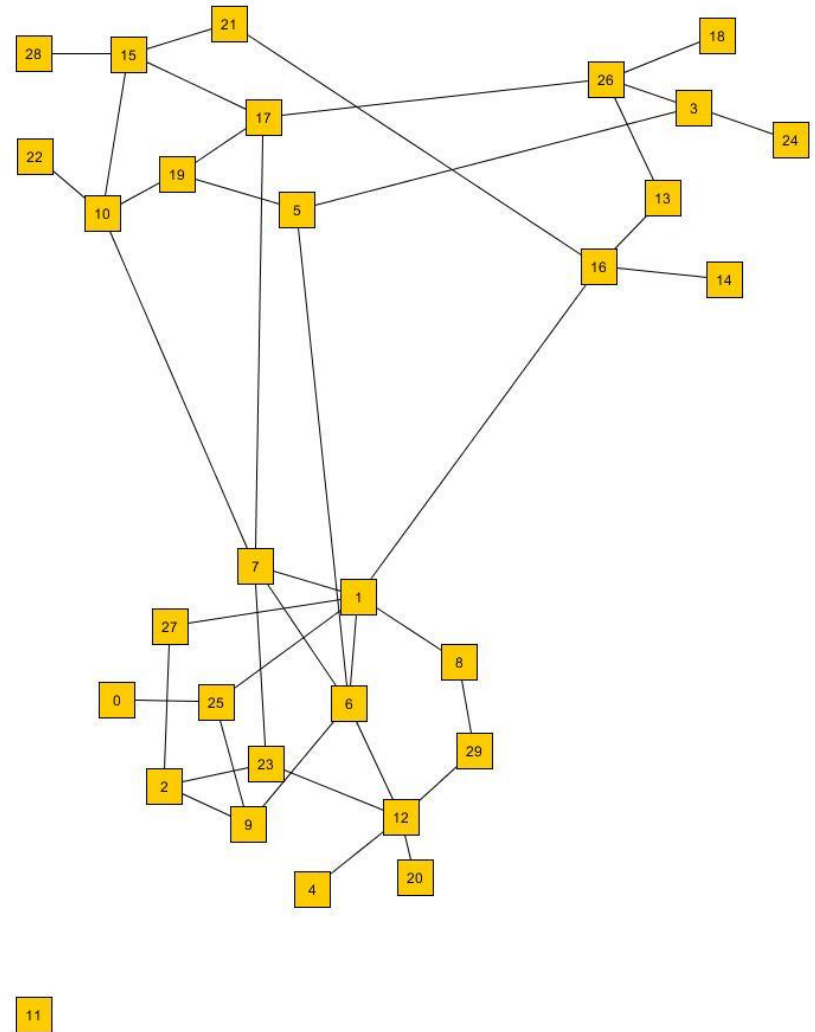
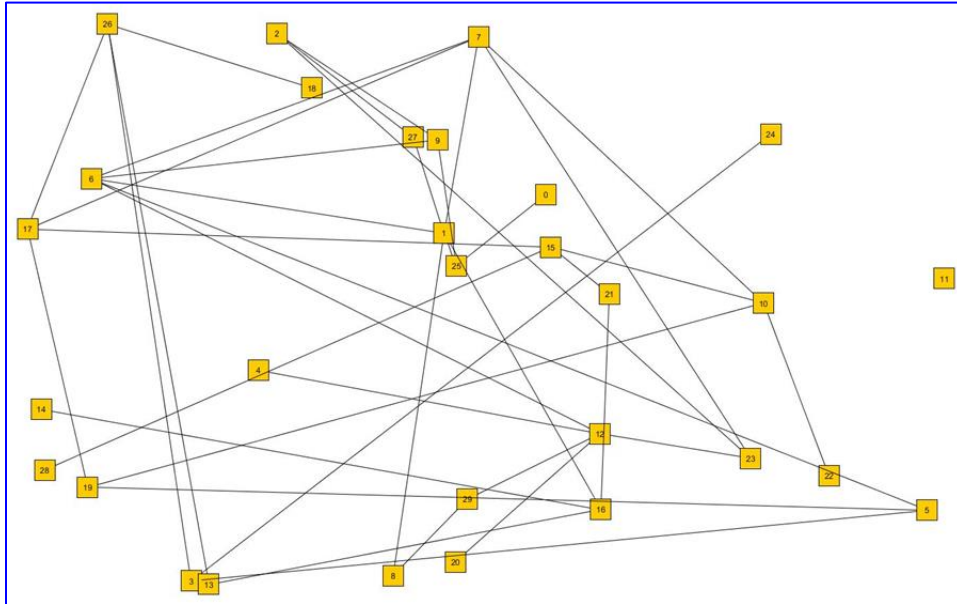
Modifica o arranjo original com base em um sistema de forças, tipicamente forças atrativas entre vértices adjacentes e repulsivas entre todos os pares de vértices.

Busca um arranjo onde os tamanhos das arestas são menores e vértices são adequadamente separados.



# Force-based (Organic)

Baseado em força. Grafos não direcionados.  
Bioinformática, Representação de  
Conhecimento, etc.

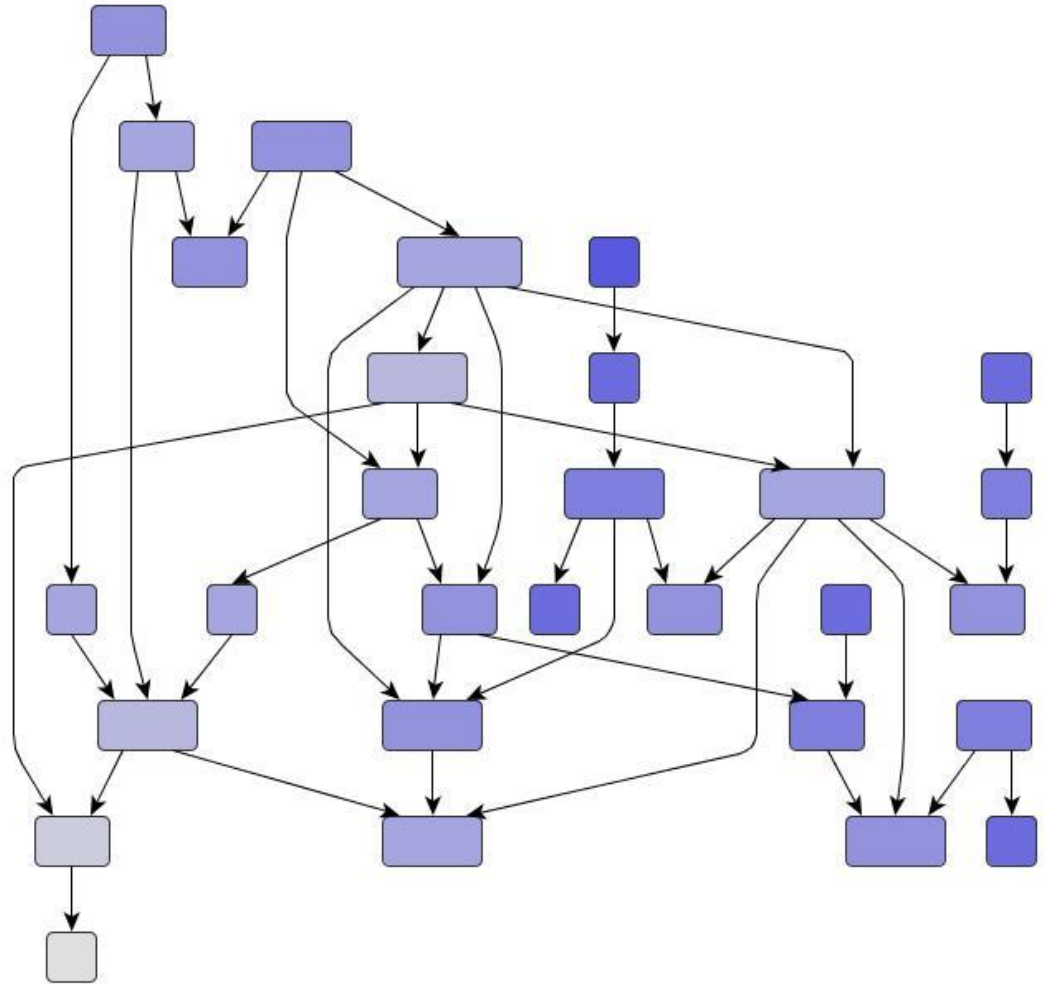


# Hierárquico

Identifica relação de precedência em grafos direcionados.

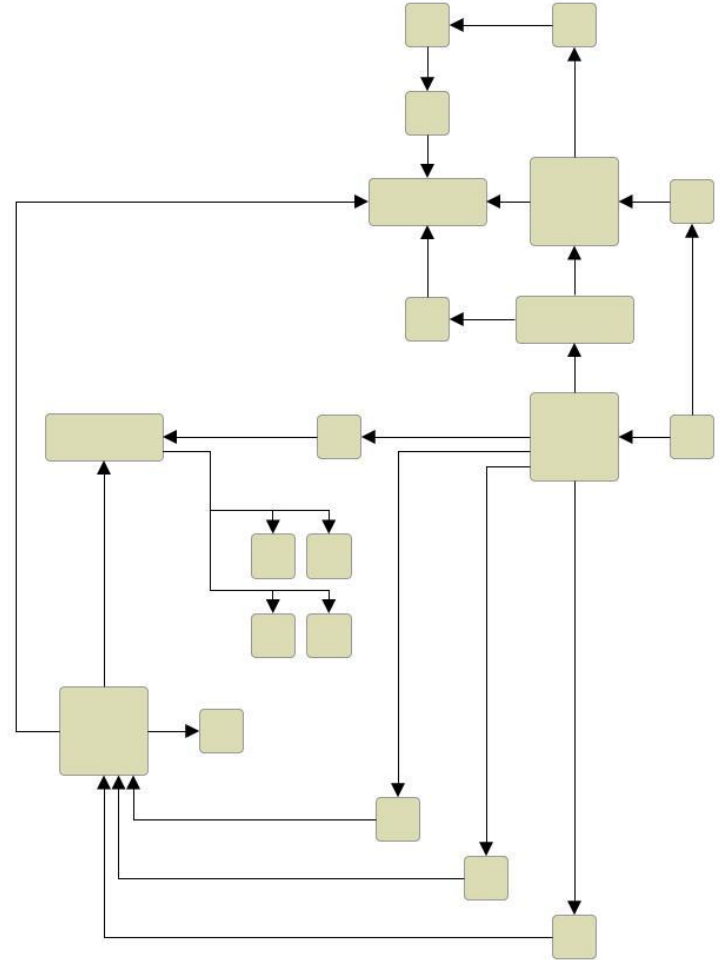
Aplicações:

- Workflows
- Engenharia de Software
- Modelo de dados



# Orthogonal

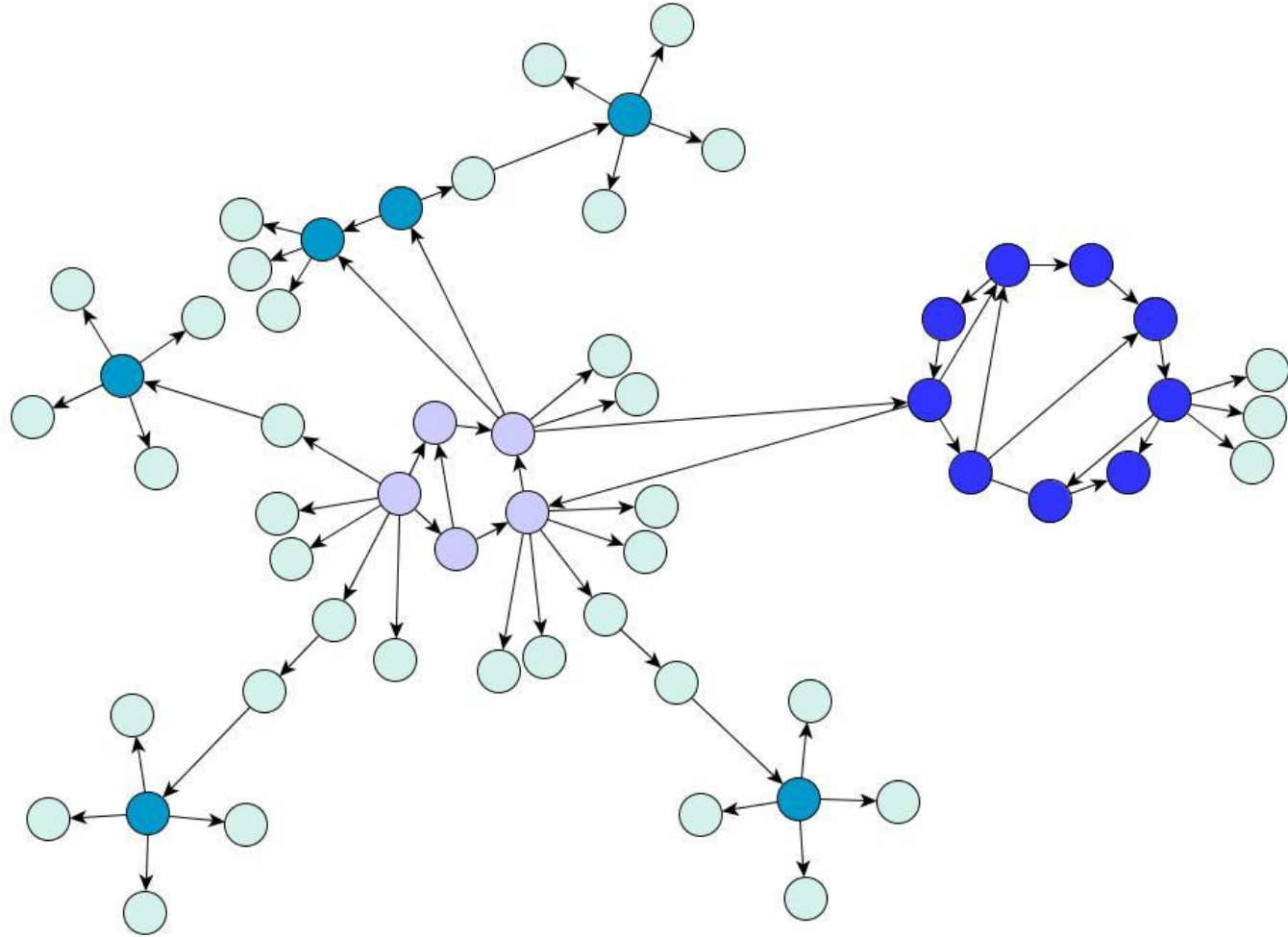
Grafos de tamanho médio. Evita sobreposição de vértices, e minimiza cruzamentos e curvaturas. Utiliza apenas linhas horizontais e verticais. Engenharia de Software.



# Circular

Enfatiza grupos (particionamento) e árvores (estrutura radial) dentro de uma rede.  
Arranja grupos em círculos separados, com topologia estrela.

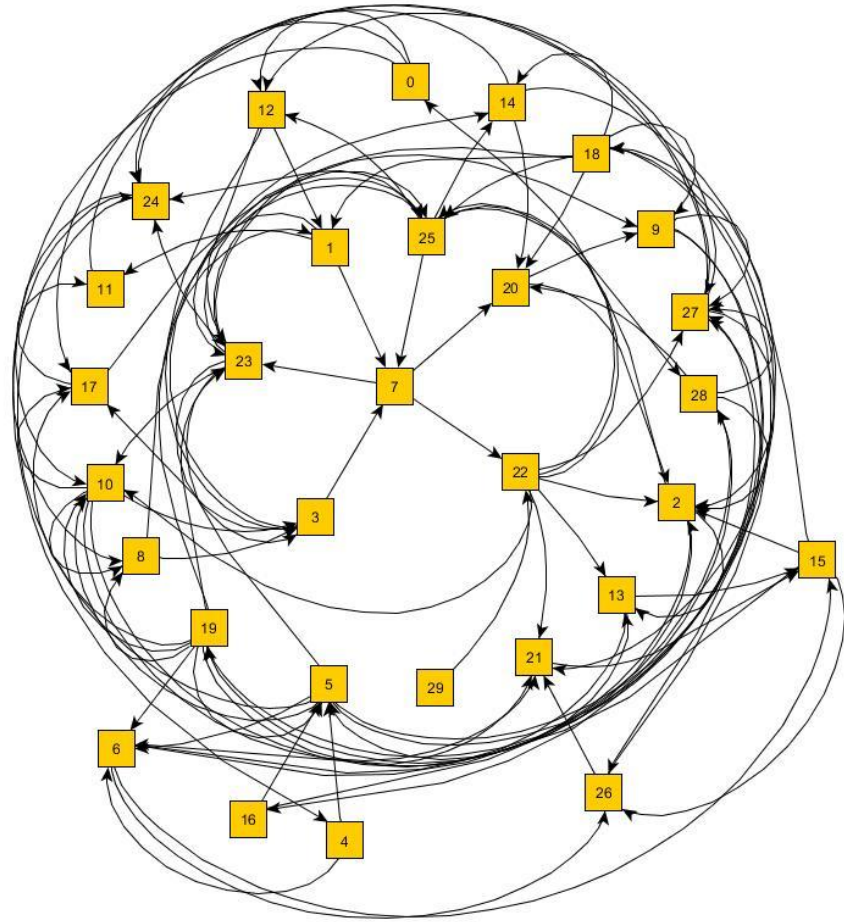
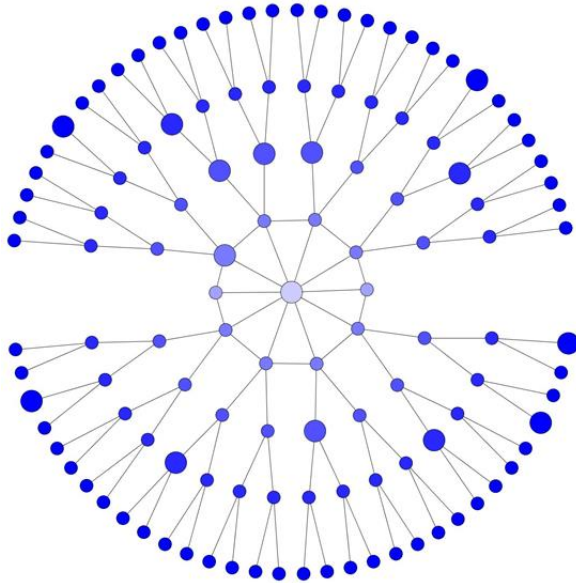
Redes Sociais e de Computadores.





# Métodos de Layout: Radial

Vértices são organizados em círculos virtuais em torno de um centro comum. Segue uma medida de Centralidade. Ex. Mínima quantidade de círculos.



# JGraphX

É uma biblioteca Java Swing que provê visualização e interação com grafos direcionados.

Dá suporte a diferentes algoritmos de layout.

# Class JGraphXAdapter

Adaptador para desenhar um grafo gerado usando a biblioteca JGraphT.

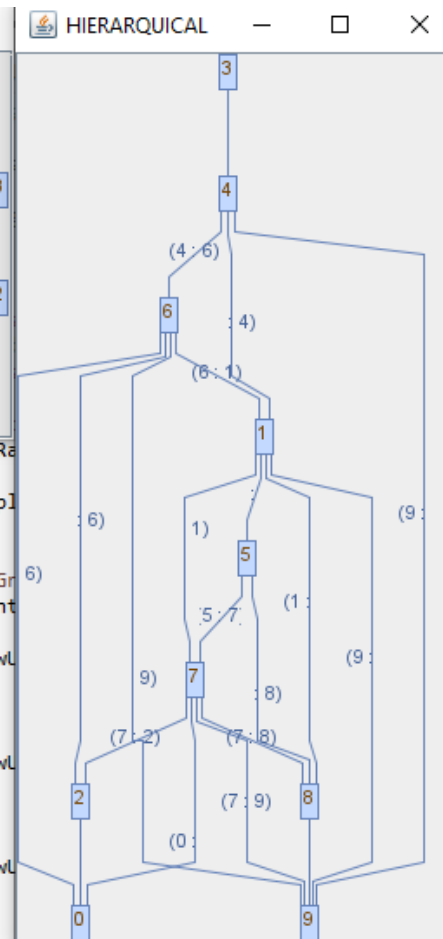
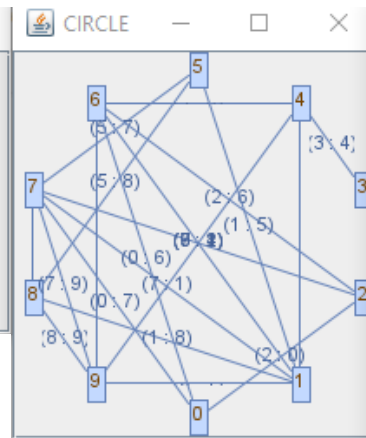
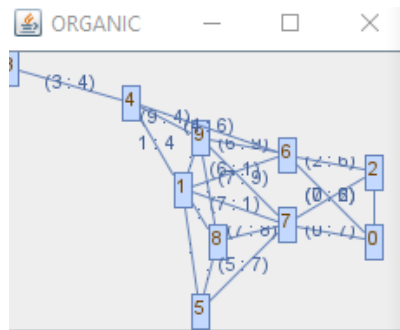
Construtor:

JGraphXAdapter(ListenableGraph <V,E> graph)

JGraphXAdapter é um adaptador para desenho de grafos da biblioteca JGraphT com a biblioteca JGraphX.

# Método para criar visualização em DrawUtil

```
22 public class DrawUtil {
23
24     public enum layout_type {CIRCLE,ORGANIC,HIERARCHICAL,ORTHOGONAL};
25
26     // Graphic view for directed graphs
27     public static <V,E> void createAndShowGui(Graph <V,E> graph, String frameLabel,
28                                             boolean directed,
29                                             boolean danglingEdges,
30                                             boolean labelsVisible,
31                                             boolean labelsClipped,
32                                             layout_type layoutType
33     ) {
34     }
```



```
public static void main(String[] args) {
    GnmRandomGraphGenerator <DefaultVertex,DefaultEdge> genGraph =
        new GnmRandomGraphGenerator <> (10,20);
    SimpleGraph <DefaultVertex,DefaultEdge> graph =
        new SimpleGraph <> (VertexEdgeUtil.createDefaultVertexSupplier(),
            SupplierUtil.createDefaultEdgeSupplier(),false);

    genGraph.generateGraph(graph);
    PrintUtil.printGraph(graph);

    DrawUtil.createAndShowGui(graph,"ORGANIC",
        false,false,true,true,
        DrawUtil.layout_type.ORGANIC);

    DrawUtil.createAndShowGui(graph,"HIERARQUICAL",
        false,false,true,true,
        DrawUtil.layout_type.HIERARCHICAL);

    DrawUtil.createAndShowGui(graph,"CIRCLE",
        false,false,true,true,
        DrawUtil.layout_type.CIRCLE);

    ExportUtil.exportDefaultGML(graph, graphpathname + "aula13.gml");
}
```

# Referências

yEd (<https://www.yworks.com/products/yed>)

Graph Drawing ([https://en.wikipedia.org/wiki/Graph\\_drawing](https://en.wikipedia.org/wiki/Graph_drawing))

JGraphT (JGraphXAdapter)

(<http://jgrapht.org/javadoc/>)

Referências Adicionais:

JGraphX (<https://github.com/jgraph/jgraphx>)

([https://jgraph.github.io/mxgraph/docs/manual\\_javavis.html](https://jgraph.github.io/mxgraph/docs/manual_javavis.html))

(<http://jgraph.github.io/mxgraph/java/docs/index.html>)