



Análise de Grafos Complexos

Este notebook demonstra a criação, visualização e análise de diferentes modelos de grafos.

🚩 Objetivos:

- Construir diferentes tipos de grafos (manuais e modelos)
- Calcular métricas locais, globais e estruturais
- Visualizar a estrutura dos grafos e suas propriedades



Instalação de Bibliotecas

```
In [ ]: import sys
!{sys.executable} -m pip install networkx matplotlib pandas random
```



Importação de Módulos e Funções

```
In [28]: from graph import grafo_simples, grafo_direcionado, grafo_barabasi_albert, grafo
from metrics import calcular_metricas, metricas_globais, propriedades_estruturais
from view import mostrar_grafo, plotar_distribuicao_de_grau, mostrar_centro
```



Função de Análise Completa

Esta função encapsula todas as etapas para análise e visualização de um grafo.

```
In [29]: def executar_analise(grafo, nome, plotar_grau=False):
    print(f"\n=== {nome} ===")

    mostrar_grafo(grafo, nome)

    print("\nMétricas locais:")
    display(calcular_metricas(grafo))

    print("\nMétricas globais:")
    print(metricas_globais(grafo))

    print("\nPropriedades estruturais:")
    print(propriedades_estruturais(grafo))

    mostrar_centro(grafo, f"Centro - {nome}")

    if plotar_grau:
        plotar_distribuicao_de_grau(grafo, f"Distribuição de Grau - {nome}")
```



Geração dos grafos para análise

```
In [ ]: G1 = grafo_simples()
        G2 = grafo_direcionado()
        G3 = grafo_barabasi_albert()
        G4 = grafo_random()
        G5 = grafo_small_world()
```



Análise de cada grafo

◆ Grafo Simples

Grafo manual e não direcionado com 5 nós. Usado para validação de métricas básicas.

```
In [ ]: executar_analise(G1, "Grafo Simples")
```



Grafo Direcionado

Grafo com arestas direcionadas. Muito útil para representar fluxos de informação ou redes com assimetria de conexão (ex.: web e tráfego urbano).

```
In [ ]: executar_analise(G2, "Grafo Direcionado")
```

◆ Grafo Barabási-Albert

Modelo de rede scale-free com hubs centrais. Espera-se distribuição de grau em power law.

```
In [ ]: executar_analise(G3, "Grafo Barabási-Albert", plotar_grau=True)
```



Grafo Erdős-Rényi (Random)

Modelo de rede aleatória com probabilidade fixa de conexão entre pares de nós. Espera-se uma distribuição de grau próxima à Poisson.

```
In [ ]: executar_analise(G4, "Grafo Erdős-Rényi (Random)", plotar_grau=True)
```



Grafo Watts-Strogatz (Small-world)

Modelo que combina propriedades de redes regulares e aleatórias. Apresenta alto clustering e baixo comprimento médio de caminho, típico de redes sociais e sistemas naturais.

```
In [ ]: executar_analise(G5, "Grafo Watts-Strogatz (Small-world)")
```



Conclusões

- Os diferentes modelos de grafos geram topologias distintas (graus, centro, caminho médio).

- O modelo Barabási-Albert apresenta hubs, típicos de redes sociais.
- O modelo Erdős-Rényi mostra uma distribuição mais aleatória (distribuição de grau próxima à Poisson).
- As visualizações ajudaram a entender e verificar as métricas estruturais observadas.

📌 Todas as análises foram feitas utilizando `networkx`, `matplotlib` e `pandas`.