

AT — Grafos na folha de cálculo: matriz de adjacência e o teorema decidido por fórmula

11.º Ano MACS 3 páginas Atividade de tecnologia

Objetivo: Representar um grafo por **matriz de adjacência** numa folha de cálculo, calcular automaticamente **ordem**, **número de arestas** e **graus**, conferir $\sum \text{graus} = 2a$ e **decidir por fórmula** a existência de circuito/caminho de Euler — além de usar **Python** para obter os graus a partir de uma lista de arestas.

Material

Material	Nota
Folha de cálculo (LibreOffice Calc, Excel ou Google Sheets)	1 por aluno/grupo
Grafo G da Tabela A (página 2)	6 vértices, 7 arestas
Python 3 (IDLE) — opcional	extensão com lista de arestas
Calculadora	conferência das somas

Normas de segurança e de trabalho

- Trabalhar sobre uma **cópia** e guardar por versões (grafos_v1.xlsx).
- Vértices/arestas** com cor de fundo diferente das **fórmulas** — nunca digitar número onde deve haver fórmula.
- Matriz: zeros e uns com **referência à lista de arestas** (fórmula, não mão) — ao alterar uma aresta, tudo se atualiza.
- Conferências obrigatórias:** matriz **simétrica** (grafo não orientado) e $\sum \text{graus} = 2 \times (\text{nº de arestas})$ — falha uma, a conclusão do teorema mente.

Procedimento

- Transcrever a lista de vértices (A–F) e de arestas da Tabela A.
- Montar a **matriz 6 × 6**: célula = 1 se a aresta existe, 0 se não (a diagonal fica 0 — sem laços).
- Conferir a simetria** (comparar triangular superior e inferior: fórmula =M; senão... por célula ou inspeção).
- Ordem** = contar vértices · **nº de arestas** = SOMAR o triângulo superior (÷2 para evitar dupla contagem).
- Grau de cada vértice** = **SOMAR(linha)** da matriz (laços: cada laço acrescenta +1 extra, pois na diagonal conta 1 mas no grau vale 2).
- Verificação:** soma total dos graus = 2 × arestas (fórmula e esperado 14).
- Decisão de Euler por fórmula:** contar ímpares com =CONT.SE(graus;"ímpar") e escrever a conclusão (0 → circuito · 2 → caminho · resto → impossível).
- Sensibilidade:** remover a aresta CD e observar a conclusão mudar (repara na Tabela D).
- Extensão Python (opcional):** lista de arestas → dicionário de graus (5 linhas) e comparar com a folha.

Fórmulas essenciais:

```
grau linha      = SOMAR(matriz[l])
arst (E tri)    = SOMAR(triângulo superior)      # arestas sem dupla contagem
conferência     = SOMAR(graus) - 2*arst → deve dar 0
ímpares         = CONT.SE(graus;"ímpar")          # decide: 0 / 2 / resto
```

Registo de dados

Tabela A — Dados do grafo G (vértices A, B, C, D, E, F) — arestas: AB, AC, AD, BE, CD, CF, DF

Tabela B — Matriz de adjacência (preencher com fórmulas)

	A	B	C	D	E	F	grau (linha)
A	0				0	0	
B		0		0		0	
C			0		0		
D				0	0		
E					0	0	
F						0	

Tabela C — Verificações

Verificação	Esperado	Obtido (fórmula)
Matriz simétrica? (não orientado)	sim	
Ordem (nº de vértices)	6	
Nº de arestas (triângulo superior)	7	
Σ graus = 2 × arestas	14	
Vértices de grau ímpar	—	
Conclusão do teorema	—	

Tabela D — Sensibilidade (remover a aresta CD) — novos graus · ímpares · nova conclusão

Tratamento de resultados

(a) Matriz e graus esperados (conferência):

$graus = g(A) = 3 \cdot g(B) = 2 \cdot g(C) = 3 \cdot g(D) = 3 \cdot g(E) = 1 \cdot g(F) = 2$ (soma = 14 = 2 × 7 ✓)

(b) Decisão de Euler: ímpares = A, C, D, E → 4 vértices ímpares → sem caminho nem circuito (só 0 ou 2 servem).

(c) Sensibilidade (sem CD): graus 3, 2, 2, 2, 1, 2 → ímpares = A e E (2) → passa a existir caminho euleriano (começa em A e termina em E, ou vice-versa) — uma só aresta mudou a conclusão do teorema.

O «olho» lê a matriz; a fórmula decide: ao alterar a lista de arestas, graus, ímpares e conclusão atualizam-se sozinhos — é o ponto desta AT (e a ponte para a AT 14, onde a sensibilidade será de modelo, não de aresta).

Análise e conclusão

1. Simetria: porquê é que a matriz de um grafo **não orientado** tem de ser simétrica? O que significaria um 1 espelho em falta (e como é que a conferência a detecta)?
2. A regra dos laços: se um vértice tivesse laço, a soma da linha daria o grau errado de quanto? (Laço aparece 1x na diagonal e vale 2 no grau $\rightarrow$ acrescenta +1.) Aplica a um caso: laço em B com aresta AB $\rightarrow$ grau da linha vs grau correto.
3. Diagnóstico de erro: a Σ graus deu 12 em vez de 14. Indica os dois erros clássicos que explicam (aresta registada numa só direção da lista; fórmula de arestas sem $\div 2$) e porque as conferências os detetam.
4. Interpretação do teorema: com 4 ímpares, a folha diz «impossível». Escreve a frase de **comunicação** a um responsável da rede — e a proposta de **eulerização** (que arestas duplicar, no mínimo) que a seguirias.
5. Sensibilidade (Tabela D): remover CD mudou a conclusão de «impossível» para «caminho». (a) Confirma com os graus 3,2,2,2,1,2. (b) Porque é que um dado único (uma aresta) muda tudo — e o que isso diz sobre projetar redes (uma ligação nova reorganiza os percursos)?
6. Conclusão: em que é que a folha de cálculo é mais rigorosa que o desenho à mão para decidir percursos? E o que continua a depender do humano (a lista de arestas, a leitura da conclusão, a escolha da eulerização)?

Conclusão-chave: a matriz de adjacência traduz o grafo em **números** e as fórmulas (**SOMAR linha $\rightarrow$ CONT.SE ímpares $\rightarrow$ teorema**) decidem sem inspeção visual — as duas conferências (simetria, Σ graus = 2a) garantem que os dados estão certos antes da decisão. O ex. 5 mostra o poder da **sensibilidade**: uma aresta trocou «impossível» por «caminho» — e é assim que se planeiam redes reais (uma ligação nova muda o sistema).

Nota: esta AT concretiza o Pensamento Computacional das AE de «Introdução aos grafos / Grafos de Euler» (representação, verificação e decisão por algoritmo) e prepara as fichas 2–3 e o AT 17 (projeto de grafos). Extensão Python: arestas = [("A", "B"), ("A", "C"), ("A", "D"), ("B", "E"), ("C", "D"), ("C", "F"), ("D", "F")]

```
graus = {v:0 for v in "ABCDEF"}
```

```
for x,y in arestas: graus[x]+=1; graus[y]+=1
```

```
print(graus, sum(graus.values())) # 14 – confere com a folha
```