UNIVERSIDADE FEDERAL DO TOCANTINS (UFT)
CÂMPUS UNIVERSITÁRIO DE PALMAS
PROGRAMA DE MESTRADO PROFISSIONAL EM MATEMÁTICA
EM REDE NACIONAL – PROFMAT

LETÍCIA MENESES MARQUES

TEORIA DOS GRAFOS COMO ESTRATÉGIA PARA A APRENDIZAGEM SIGNIFICATIVA DE MATEMÁTICA

PALMAS (TO)
2024

LETÍCIA MENESES MARQUES

TEORIA DOS GRAFOS COMO ESTRATÉGIA PARA A APRENDIZAGEM SIGNIFICATIVA DE MATEMÁTICA

Dissertação apresentada ao Programa de Mestrado Profissional em Matemática em Rede Nacional - PROFMAT da Universidade Federal do Tocantins como requisito parcial para a obtenção do título de Mestra - Área de Concentração: Matemática.
Orientador: Prof. Dr. Warley Gramacho da Silva.

PALMAS (TO)

2024

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal do Tocantins

M543t Meneses Marques, Letícia.

TEORIA DOS GRAFOS COMO ESTRATÉGIA PARA A APRENDIZAGEM SIGNIFICATIVA DE MATEMÁTICA. / Letícia Meneses Marques. – Palmas, TO, 2024.

139 f.

Dissertação (Mestrado Profissional) - Universidade Federal do Tocantins – Câmpus Universitário de Palmas - Curso de Pós-Graduação (Mestrado) Profissional em Matemática, 2024.

Orientador: Warley Gramacho da Silva

1. Ensino de Grafos. 2. Modelagem Matemática. 3. Algoritmo PageRank. 4. Python. I. Título

CDD 510

Elaborado pelo sistema de geração automática de ficha catalográfica da UFT com os dados fornecidos pelo(a) autor(a).

LETÍCIA MENESES MARQUES

TEORIA DOS GRAFOS COMO ESTRATÉGIA PARA A APRENDIZAGEM SIGNIFICATIVA DE MATEMÁTICA

Dissertação apresentada ao Programa de Mestrado Profissional em Matemática em Rede Nacional - PROFMAT, foi avaliada para obtenção do título de Mestra e aprovada em sua forma final pelo Orientador e Banca Examinadora.

Data de Aprovação: 22 de agosto de 2024.

Banca examinadora:

Documento assinado digitalmente
gov.br
WARLEY GRAMACHO DA SILVA
Data: 20/09/2024 09:55:23-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Warley Gramacho da Silva
Orientador, UFT

Documento assinado digitalmente
gov.br
HELLENA CHRISTINA FERNANDES APOLINARIO
Data: 20/09/2024 09:58:59-0300
Verifique em https://validar.iti.gov.br

Prof.ª Dra. Hellena Christina Fernandes Apolinário
Examinadora, UFT

Documento assinado digitalmente
gov.br
DIEGO NUNES BRANDAO
Data: 20/09/2024 12:21:24-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Diego Nunes Brandão
Examinador, CEFET/RJ

Aos obstinados estudiosos,

cuja paixão pelo aprendizado inspira e transforma o mundo,

eu dedico este trabalho.

AGRADECIMENTOS

A mim, por não desistir do sonho de lecionar, apesar da desmotivação gerada pela sociedade na qual vivemos. À minha noiva Isabella, por me ajudar com as imagens, códigos e por todo apoio aos meus estudos. Agradeço ao meu amigo, prof. Me. Arthur Batista, que nunca desistiu da nossa amizade, por todo suporte durante o curso.

Aos professores do PROFMAT Palmas, que contribuiram de forma significativa para minha carreira enquanto professora e pesquisadora, em especial a prof.ª Dra. Hellena Apolinário que com todo seu rigor matemático me fez lembrar o motivo de eu gostar tanto desta ciência.

Deixo também meu agradecimento ao prof. Dr. Warley Gramacho, que com toda paciência me orientou e fez o possível para termos um bom trabalho, apesar das circunstâncias.

O estudo em geral, a busca da verdade e da beleza,

são domínios em que nos é consentido permanecer crianças por toda a vida.

(Albert Einstein)

RESUMO

Este trabalho aborda os conceitos da Teoria dos Grafos e sua possibilidade de ensino na Educação Básica, utilizando a modelagem matemática como uma metodologia que promove aprendizagem significativa. É explanado acerca do Novo Ensino Médio e a organização dos itinerários formativos, que dão possibilidade de aprofundar o tema com os estudantes. A teoria dos Grafos é trabalhada de maneira teórica introdutória, com definições, conceitos, representação, história e principais teoremas. O Algoritmo Pagerank é explicado de maneira simplificada, a fim de promover a integração da teoria à utilização de grafos no cotidiano, que grande parte das pessoas usa, mas desconhece. Por fim, são propostas atividades que visam trabalhar a introdução da Teoria dos Grafos como método de resolução de problemas. Os objetivos centrais incluem a compreensão do papel da matemática como ciência que investiga padrões e a aplicação da modelagem matemática através de grafos para resolver problemas. Conclui-se que o desenvolvimento de atividades investigativas relacionadas ao tema pode melhorar a compreensão de conceitos matemáticos de outras áreas e promover o desenvolvimento de competências e habilidades analíticas e organizacionais relacionadas à Base Nacional Comum Curricular. A metodologia utilizada foi a bibliográfica e exploratória, com o propósito de explorar na literatura os fundamentos da Teoria dos Grafos, bem como suas possíveis abordagens em sala de aula, tendo como público-alvo os professores de matemática.

Palavras-chave: Ensino de Grafos. Modelagem Matemática. Algoritmo PageRank. Python.

ABSTRACT

This paper addresses the concepts of Graph Theory and its potential for teaching in Basic Education, using mathematical modeling as a methodology that promotes meaningful learning. It explores the New High School and the organization of formative itineraries, which provide opportunities to delve deeper into the subject with students. Graph Theory is introduced theoretically, with definitions, concepts, representation, history, and main theorems. The Pagerank Algorithm is explained in a simplified manner to promote the integration of theory with the use of graphs in everyday life, which many people use but are unaware of. Finally, activities are proposed that aim to introduce Graph Theory as a method for problem-solving. The central objectives include understanding the role of mathematics as a science that investigates patterns and applying mathematical modeling through graphs to solve problems. It is concluded that the development of investigative activities related to the topic can improve the understanding of mathematical concepts from other areas and promote the development of analytical and organizational skills and competencies related to the Brazilian National Common Curricular Base. The methodology used was bibliographic and exploratory, with the purpose of exploring in the literature the foundations of Graph Theory, as well as its possible approaches in the classroom, targeting mathematics teachers.

Keywords: Teaching Graphs. Mathematical Modeling. PageRank Algorithm. Python.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SÍMBOLOS

$\in$	Pertence a
G	Representação de um grafo
V	Conjunto dos vértices de um grafo
A	Conjunto das arestas de um grafo
$\varnothing$	Conjunto vazio
M	Matriz de adjacência
m_{ij}	Elemento da matriz de adjacência
N	Matriz de incidência
n_{ij}	Elemento da matriz de incidência
$N(x)$	Conjunto de vizinhos um vértice x
$N^+(x)$	Conjunto de sucessores de um vértice x
$N^-(x)$	Conjunto de antecessores de um vértice x
$d(x)$	Grau de um vértice x
$d^+(x)$	Número de sucessores de um vértice x
$d^-(x)$	Número de antecessores de um vértice x
$p_t(X)$	PageRank de X na iteração t
$\sum$	Somatório

SUMÁRIO

1 INTRODUÇÃO

Neste trabalho, se discorre acerca da Teoria dos Grafos bem como seus principais estudiosos, conceitos e definições, com especial atenção aos Trajetos Eulerianos e ao Algoritmo PageRank.

A pesquisa se apoia no seguinte problema de pesquisa: **De que maneira o uso da Teoria dos Grafos pode auxiliar na compreensão de conceitos matemáticos e promover desenvolvimento de habilidades analíticas e de resolução de problemas em estudantes da Educação Básica?**

A proposta concentra-se na introdução e exploração de conceitos da Teoria dos Grafos, através de atividades que estimulem o raciocínio lógico e propiciem sua modelagem através de um grafo. Estes conceitos apresentam frequentemente desafios aos jovens estudantes devido à sua natureza abstrata. Este estudo pretende preencher esta lacuna educacional, explorando como o ensino destes tópicos pode beneficiar o desenvolvimento cognitivo dos alunos e melhorar as suas habilidades e competências matemáticas, sobretudo na resolução de problemas e desenvolvimento de raciocínio lógico e algorítmico.

O objetivo geral deste trabalho é propor atividades introdutórias aos conceitos de Teoria dos Grafos para alunos da Educação Básica, visando promover o desenvolvimento de habilidades em matemática e suas tecnologias.

A pesquisa se justifica pela necessidade de uma educação matemática mais prática e envolvente. Destaca-se, também, a carência de diretrizes pedagógicas específicas para a incorporação desses conceitos na Educação Básica e o potencial impacto positivo na motivação dos alunos. Os resultados buscam contribuir de forma significativa para o ensino da matemática, oferecendo aos educadores novas possibilidades pedagógicas que possam colaborar para o desenvolvimento das habilidades cognitivas dos estudantes.

Considera-se os professores de matemática público-alvo deste trabalho, tanto os que estão em atividade, quanto os que estão em formação. Procura-se amparar seus conhecimentos e prover noções do emprego do objeto de conhecimento considerado, com a finalidade de proporcionar maior propriedade ao ministrar aulas e colocar em movimento os saberes adquiridos, ao proporcionar a socialização das possíveis resoluções dos problemas propostos.

O tema de pesquisa surgiu de uma apresentação de trabalho feita por estudantes da

pesquisadora, em uma turma da 2ª série do Ensino Médio, sobre aplicações de matrizes e sistemas lineares. Na apresentação, os estudantes discorreram sobre o uso de matrizes no Algoritmo PageRank (algoritmo de busca do Google), e utilizaram um modelo de grafo para ilustrar um exemplo. Mesmo que eles não soubessem sobre a Teoria dos Grafos, a imagem foi útil para explanar acerca do assunto. Assim, surgiu a curiosidade de como explorar a Teoria dos Grafos em turmas da Educação Básica, como forma de suporte à modelagem de problemas.

A Teoria dos Grafos é um campo que estuda as relações entre objetos ao utilizar esquemas de diagrama como modelo. Pode-se utilizar grafos para representar objetos que se conectam de alguma maneira, como cidades em um mapa, computadores em uma rede, pontos em um sistema de transporte, páginas da *web*, circuitos elétricos entre outros. Esta é uma área fundamental da Matemática Discreta e tem uma ampla gama de aplicações em ciência da computação, logística, biologia, química, neurociência, engenharias, ciências sociais e etc. (Balakrishnan, 1997).

O presente trabalho se caracteriza como uma pesquisa bibliográfica e exploratória, com o propósito de explorar na literatura os fundamentos da Teoria dos Grafos, bem como busca oferecer alternativas pedagógicas para educadores interessados em adotar essa abordagem, ainda pouco explorada na Educação Básica.

Este estudo, fundamentado em Netto, Oswaldo e Jurkiewicz (2009), adota a abordagem bibliográfica, sustentando-se em materiais já construídos, incluindo artigos científicos publicados em periódicos acadêmicos, dissertações e teses. A coleta de informações foi conduzida por meio de uma busca de artigos relacionados ao tema em questão. O levantamento bibliográfico foi realizado através dos mecanismos de pesquisa Periódicos Capes e *Google Acadêmico*, por meio dos quais é possível encontrar livros, artigos, monografias e citações relativas ao assunto. Neste trabalho, os conhecimentos adquiridos são organizados de forma a oportunizar uma reflexão construtiva sobre o tema estudado (Gil, 2008).

Foram considerados artigos acadêmicos, notas de aula, dissertações e teses publicados em Língua Portuguesa, Língua Inglesa ou Espanhola, disponíveis gratuitamente online nos bancos de dados e revistas acadêmicas. Os descritores utilizados foram: Grafos, Teoria dos Grafos, Ensino de Grafos, Algoritmo PageRank, Circuitos Eulerianos, Educação Matemática, Modelagem Matemática e Novo Ensino Médio.

Após a fase inicial da pesquisa, procedeu-se à síntese dos principais achados presentes na literatura, através de uma leitura minuciosa e considerando os aspectos relevantes conforme os objetivos estabelecidos para a pesquisa, os quais consistem em promover a introdução da

Teoria dos Grafos na Educação Básica, analisando seu impacto nas habilidades matemáticas e na motivação dos alunos, e fornecer orientações pedagógicas para educadores interessados nesta abordagem. Os principais capítulos deste trabalho estão organizados da seguinte maneira:

No Capítulo (2) é apresentado o referencial teórico do trabalho como justificativa para introduzir o tema de grafos no Ensino Médio, ao destacar o processo de ensino-aprendizagem de matemática, o conceito de modelagem matemática, a importância de uma aprendizagem significativa e os aspectos pedagógicos e normativos do Novo Ensino Médio.

Fazendo uma abordagem teórica, o Capítulo (3) introduz a Teoria dos Grafos e definições iniciais, apresenta vocabulários específicos da área, estabelece a ideia de vizinhança e grau, demonstra o Teorema do Aperto de Mãos e suas implicações, mostra como armazenar grafos em matrizes, apresenta os passeios e tipos de passeios em grafos, explana acerca da origem histórica da teoria dos grafos e demonstra o Teorema de Euler, além de apresentar os Trajetos Eulerianos, Caminhos Hamiltonianos e o Problema das Quatro Cores.

O Algoritmo PageRank, importante aplicação de grafos no cotidiano, é explicado no Capítulo (4), de forma que não seja necessário conhecimento prévio superior de Matemática Discreta, Cadeias de Markov, Álgebra Linear ou programação; também são mostrados códigos em linguagem *python* para gerar as iterações do algoritmo.

Finalmente, no Capítulo (5), são propostas atividades para introduzir o ensino de grafos, constituindo o produto final deste trabalho. Esse capítulo fornece uma abordagem prática para a teoria apresentada anteriormente, permitindo que os estudantes apliquem conceitos teóricos em situações reais ou simuladas. As atividades foram cuidadosamente elaboradas para facilitar a compreensão dos princípios fundamentais dos grafos, a fim de incentivar o pensamento lógico-crítico e a resolução de problemas.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda temas fundamentais para a compreensão do processo de ensino-aprendizagem da matemática, com ênfase na modelagem matemática e sua possibilidade de integração no contexto do Novo Ensino Médio no Brasil.

Inicialmente, discute-se o papel da Matemática como ciência que investiga regularidades e padrões, destacando sua importância como sistema formal de pensamento. Em seguida, explora-se a modelagem matemática como uma metodologia que utiliza conceitos matemáticos para descrever e resolver problemas do mundo real, promovendo uma aprendizagem significativa e conectando teoria e prática.

Por fim, examina-se o impacto da Lei nº 13.415/2017, que altera a Lei de Diretrizes e Bases da Educação Nacional (LDB) no que tange à estrutura curricular do Ensino Médio, com ênfase nos itinerários formativos e na inclusão da Matemática e suas Tecnologias como área de conhecimento estruturante, na qual a modelagem matemática e a teoria dos grafos podem ser amplamente exploradas.

2.1 O processo de ensino-aprendizagem de Matemática

Compreender a natureza da Matemática é essencial para promover reflexões sobre seu processo de ensino-aprendizagem. Conforme Devlin (2004), a Matemática é a ciência que estuda regularidades e padrões. De acordo com essa definição, o papel do matemático é analisar padrões abstratos, que podem ser reais ou imaginários, visuais ou conceituais. A Matemática também pode ser definida como um sistema formal de pensamento para reconhecer, classificar e explorar padrões.

> A matemática como uma expressão da mente humana, ativará os reflexos, o contemplamento da razão e o desejo pela perfeição estética. É também chamada por muitos de linguagem universal (é uma linguagem porque é formada por signos linguísticos que passam ideias e significados). Ela pode ser dividida em matemática pura e aplicada e seus elementos básicos são a lógica e a intuição, análise e construção, generalização e individualização (Onodera, 2004).

É importante compreender que a natureza da Matemática não se reduz à sua instrumentalidade. A ciência moderna tem sua fundação com Galileu Galilei (1564 – 1642), cujo método se baseia em matematizar a experiência (Mariconda, 2006). Segundo a Competência Geral 2 da BNCC, Brasil (2017), a escola é um espaço de desenvolvimento do pensamento científico,

sendo assim, a Matemática transcende os objetos de conhecimento abordados no componente curricular, consistindo em abstração e organização lógica de ideias.

A técnica matemática constitui um conjunto de atitudes, ideias e métodos necessários para descobrir, formular e resolver problemas matemáticos. As atitudes referem-se à postura diante dos problemas matemáticos, cuja resolução requer engajamento e racionalidade. As ideias matemáticas são noções intuitivas fundamentais, como conjunto, transformação e algoritmo (Fassarella, 2015).

> Os métodos matemáticos constituem as seguintes modalidades de pensamento: indução, analogia, dedução, integração, abstração, simplificação, generalização, especialização, representação simbólica, pensamento integrativo, pensamento desenvolvimentista. Todas as atitudes, ideias e modalidades de pensamento matemático são usadas e também desenvolvidas pela aplicação na resolução dos problemas matemáticos (FASSARELLA, 2015 apud ISODA; KATAGIRI, 2012).

A Teoria dos Grafos, enquanto ramo da Matemática que se dedica ao estudo de conexões e relações, oferece um campo vasto para o desenvolvimento do pensamento crítico e analítico em estudantes do ensino básico. A introdução do tema por meio de problemas investigativos permite que os alunos visualizem e manipulem conjuntos, ao estabelecer padrões e promover uma compreensão mais concreta e intuitiva dos conceitos matemáticos, trabalhando as modalidades de pensamento descritos por Isoda e Katagiri (2012). Dessa forma, também podem ser desenvolvidas habilidades essenciais para criatividade, resolução de problemas complexos e tomada de decisões acertivas.

A reconfiguração recente do Ensino Médio, instituída pela Lei 13.415/2017, introduziu uma flexibilização no currículo, permitindo aos alunos a escolha de disciplinas alinhadas aos seus interesses, conferindo-lhes protagonismo em seu desenvolvimento escolar. Essas disciplinas, incluindo a possibilidade de inserção da Teoria dos Grafos, são denominadas Itinerários Formativos (Silva, Krawczyk e Calçada, 2023).

Os itinerários visam aprofundar conhecimentos estruturantes nas diversas áreas do saber, como Matemática e suas tecnologias. Sua oferta deve considerar a realidade local e os recursos disponíveis em termos físicos, materiais e humanos de cada instituição de ensino, proporcionando uma formação contemporânea que esteja em sintonia com a vida presente e futura dos estudantes (Aleixo, 2020).

2.2 Modelagem matemática e a aprendizagem significativa

A Modelagem Matemática é um processo no qual os conceitos e as ferramentas da matemática são utilizados para descrever, analisar e compreender situações do mundo real (Bassanezi, 2015).

Ao longo do tempo, a validação de uma teoria científica passou a depender de sua capacidade de ser formulada em termos matemáticos. A matemática, por sua vez, tem acompanhado e respondido de forma significativa às necessidades de diversas áreas de pesquisa, gerando novas teorias matemáticas.

Segundo Melo e Bisognin (2021):

> A busca do conhecimento científico, em qualquer campo, deve consistir, essencialmente, em:
>
> 1. Aceitar somente aquilo que seja tão claro em nossa mente que exclua qualquer dúvida;
> 2. Dividir os grandes problemas em problemas menores;
> 3. Argumentar, partindo do simples para o complexo;
> 4. Verificar o resultado final.

Atualmente, é notável a presença generalizada da matemática nos aspectos teóricos das ciências naturais, como Física, Astrofísica, e Química. Nas Ciências Biológicas, a matemática desempenha um papel fundamental na modelagem de mecanismos que regem a dinâmica de populações, a epidemiologia, a ecologia, a neurologia, a genética e os processos fisiológicos.

A aprendizagem significativa, de acordo com Ausubel (1963), ocorre quando um novo conhecimento é relacionado de forma substantiva e não arbitrária com o conhecimento prévio do discente. Esse tipo de aprendizagem difere da simples memorização mecânica isolada, pois permite que o aluno integre o novo conteúdo à sua estrutura cognitiva existente, tornando o aprendizado mais relevante e duradouro.

Deste modo, a modelagem matemática oferece uma articulação essencial entre teoria e prática. Enquanto metodologia de ensino, a modelagem visa promover a aprendizagem significativa do estudante, proporcionando-lhe a oportunidade de explorar conceitos matemáticos de forma contextualizada e relevante para a vida cotidiana (Costa e Igliori, 2016).

Podemos definir formalmente:

Definição 1. *Um **modelo matemático** é um conjunto de símbolos e relações matemáticas que representam, de alguma forma, um objeto de estudo.*

Nesse contexto, os modelos matemáticos são fundamentais para o processo de modelagem, pois servem como ferramentas que permitem representar situações reais por meio da matemática.

Conforme Brandt, Burak e Kluber (2016), "a Modelagem Matemática é uma metodologia diferenciada para o ensino da Matemática, uma vez que não se configura como uma metodologia tradicional, isto é, não enfatiza a memorização e procedimentos mecânicos".

Ao conectar conceitos matemáticos abstratos a situações cotidianas, a modelagem pode tornar a matemática mais significativa para os alunos, despertando o interesse e a curiosidade. Essa abordagem permite que os alunos construam seu próprio conhecimento, desenvolvam habilidades de resolução de problemas e compreendam a aplicabilidade da matemática em diversos contextos. Além disso, a modelagem incentiva a comunicação, a colaboração e o desenvolvimento do pensamento crítico, preparando os alunos para enfrentar os desafios do mundo real (Ribeiro, 2010).

Neste trabalho, a teoria dos grafos é utilizada como um recurso importante para modelar problemas, promover a interdisciplinaridade e o desenvolvimento do conhecimento profissional dos docentes.

2.3 Novo Ensino Médio

Em consonância com MEC (2022), o Novo Ensino Médio é uma reforma educacional instituída pela Lei nº 13.415/2017, que visa flexibilizar e modernizar o currículo do Ensino Médio brasileiro, ao reformular profundamente sua estrutura. Esta lei trouxe mudanças importantes para a Lei de Diretrizes e Bases da Educação Nacional (LDB); uma das principais alterações foi a ampliação da carga horária mínima anual de 800 para 1000 horas, que terminou de ser implementada em 2022.

A Lei nº 13.415/2017 introduziu uma nova organização curricular que se destaca pela flexibilidade. Com a Base Nacional Comum Curricular (BNCC), o ensino médio passou a ofertar diversos itinerários formativos, permitindo aos estudantes escolher áreas específicas de conhecimento ou formação técnica e profissional. Além disso, o Novo Ensino Médio propôs a inclusão de disciplinas eletivas e projetos integradores, ampliando as opções curriculares e possibilitando uma maior personalização da educação. A BNCC estabelece as competências gerais que devem ser desenvolvidas pelos estudantes, enquanto os itinerários formativos permitem um aprofundamento em áreas de interesse específico.

A implementação dessas mudanças foi gradual, dando tempo para que as escolas adaptem suas infraestruturas e metodologias. No entanto, esse processo tem gerado debates e críticas. Muitos questionam a capacidade das escolas de se adaptarem a essas novas exigências, especialmente em regiões com menos recursos. Há também preocupações sobre o impacto dessas mudanças na qualidade da educação oferecida e na igualdade de acesso às novas oportunidades educacionais (Oliveira, 2023).

Outros pontos de discussão incluem a preparação dos professores para lidar com a nova estrutura curricular e a efetividade dos itinerários formativos em realmente proporcionar uma formação adequada para os estudantes. A autonomia dada às redes de ensino para definir os itinerários a serem oferecidos pode levar a uma grande variação na implementação das mudanças, resultando em desigualdades entre diferentes regiões e escolas.

As mudanças propostas pela Lei nº 13.415/2017 visam modernizar o ensino médio e torná-lo mais relevante para as necessidades contemporâneas. No entanto, a efetividade dessas reformas depende de uma série de fatores, incluindo investimentos adequados em infraestrutura, capacitação de professores e um acompanhamento rigoroso para garantir que todas as escolas possam oferecer uma educação de qualidade e equitativa. Desta maneira, ao professor cabe se adaptar à nova realidade juntamente com os estudantes, ao pensar em diferentes estratégias para atender às demandas do novo formato educacional.

2.3.1 Itinerários Formativos

Os itinerários formativos são um componente central do Novo Ensino Médio. Eles representam um conjunto de disciplinas, projetos, oficinas, núcleos de estudo e outras atividades, que os estudantes podem escolher de acordo com suas aspirações acadêmicas e profissionais. O objetivo é proporcionar uma formação mais personalizada e alinhada às necessidades individuais dos estudantes.

A estrutura dos itinerários formativos é dividida em cinco grandes áreas de conhecimento: Linguagens e suas Tecnologias, Matemática e suas Tecnologias, Ciências da Natureza e suas Tecnologias, Ciências Humanas e Sociais Aplicadas, e Formação Técnica e Profissional. Cada uma dessas áreas oferece opções específicas que permitem aos estudantes se aprofundarem em temas que consideram mais relevantes para sua futura carreira acadêmica ou profissional.

É importante destacar que os itinerários formativos são diferentes das disciplinas eletivas. Enquanto os itinerários formativos são blocos temáticos de estudo que podem abranger vários

anos do ensino médio, as disciplinas eletivas são cursos ou atividades de menor duração que complementam a formação básica. As disciplinas eletivas podem incluir, por exemplo, artes, línguas estrangeiras, esportes, entre outros, e têm como objetivo enriquecer o currículo e proporcionar aos alunos experiências educacionais diversificadas.

A implementação dos itinerários requer uma reorganização significativa do currículo das escolas. As instituições de ensino devem garantir que os estudantes tenham acesso a pelo menos dois desses itinerários, proporcionando uma variedade de opções que atendam às diferentes demandas e interesses. As escolas têm autonomia para definir quais itinerários oferecer, mas devem considerar as necessidades e expectativas da comunidade escolar, bem como as diretrizes estabelecidas pela BNCC (Brasil, 2017).

Além disso, a flexibilização curricular proposta exige uma maior capacitação dos professores, que precisam estar preparados para lidar com uma diversidade de conteúdos e metodologias de ensino. As escolas também precisam investir em infraestrutura e recursos didáticos que suportem a oferta desses itinerários, garantindo que todos os estudantes tenham acesso às mesmas oportunidades educacionais.

Dentro desse contexto, a Modelagem Matemática pode ser integrada de forma aprofundada na área de Matemática e suas Tecnologias, conectando os eixos estruturantes do Novo Ensino Médio, como investigação científica, processos criativos, mediação e intervenção sociocultural, e empreendedorismo, conforme proposto por Bertone, Bassanezi e Jafelice (2014).

A proposta deste trabalho é fornecer maior suporte aos professores que desejam inserir grafos como forma de modelagem matemática e desenvolvimento de novas habilidades no Educação Básica, incluindo o Ensino Médio, com maior versatilidade.

3 TEORIA DOS GRAFOS

Neste capítulo são explorados os conceitos elementares da teoria dos grafos. Inicia-se com definições e vocabulário específico, passando pelos tipos de grafos, como simples, direcionados e ponderados.

São discutidos os conceitos de vizinhança, grau e o Teorema do Aperto de Mãos. Em seguida, são apresentadas as matrizes que representam grafos em estruturas de dados; também são introduzidos os conceitos de passeios e a origem histórica da teoria dos grafos. Finalmente, são abordados os trajetos eulerianos, caminhos hamiltonianos e o teorema das quatro cores, o que oferece uma introdução fundamental para a compreensão e aplicação da teoria.

A maior parte das imagens deste trabalho foi produzida em linguagem *Python*. Alguns de seus códigos podem ser vistos no Apêncide (A).

3.1 Conceito

A teoria dos grafos é uma área da matemática que estuda as relações entre objetos. É importante destacar que as imagens (diagramas) de grafos são apenas representações visuais dessas relações, e não os grafos em si, que são conceitos abstratos. De acordo com Netto, Oswaldo e Jurkiewicz (2009):

Definição 2. *Um grafo é uma estrutura matemática formada por* ***dois conjuntos****.*

O primeiro conjunto, que ***denominaremos de*** V*, é o conjunto não-vazio de vértices (ou nós). O outro conjunto é o das relações entre vértices, chamado conjunto de arestas,* ***representado por*** A*.*

Se dois vértices x e y de V *estão relacionados, diremos que entre eles existe uma* ***aresta*** *pertencente a A, que chamaremos de* (x,y)*.*

Deste modo, o grafo será denominado $G = (V,A)$*.*

Considerações:

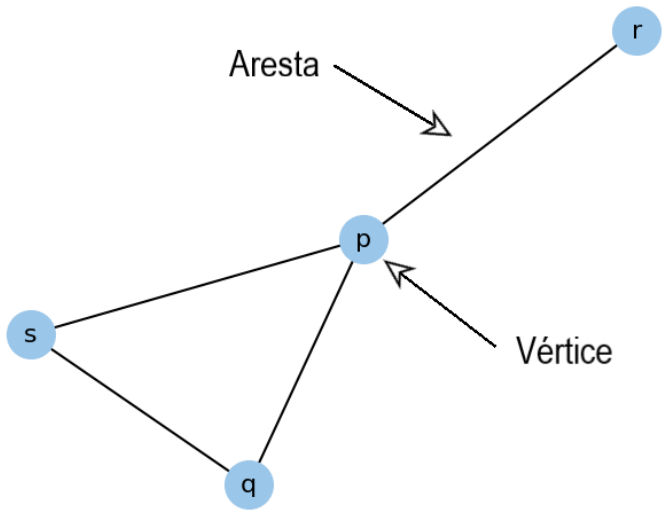
(i) A **ordem** de G, denotada por $|V|$, é a quantidade de vértices que ele possui, ou seja, a quantidade de elementos em V;

(ii) O **tamanho** de G, denotado por $|A|$, é a quantidade de ligações que ele possui, ou seja, a quantidade de elementos em A;

Exemplo 1. $G = (V,A)$ tal que $V = \{p,q,r,s\}$ e $A = \{(p,q),(p,r),(p,s),(q,s)\}$. Assim, $|V| = 4$ e $|A| = 4$.

Os grafos podem ser representados por meio de um diagrama cujos vértices são representados por pontos, e as arestas são mostradas como linhas que conectam os pares de vértices correspondentes. O diagrama do grafo G mencionado no Exemplo (1) está ilustrado na Figura (1).

Figura 1 – Exemplo (1) - vértices e arestas



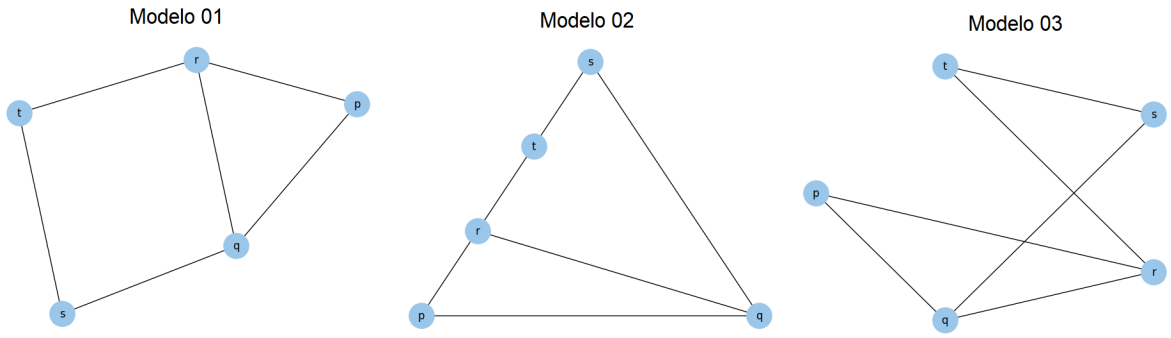
Fonte: Autora (2024).

Um mesmo grafo pode ter diferentes representações gráficas devido à flexibilidade de como os vértices e arestas podem ser dispostos visualmente. A curvatura das arestas, ordem dos vértices, simetria, dimensão do desenho e estilo visual influenciam na maneira que a representação pode ser feita. Desde que a estrutura de conexões (arestas) entre os vértices seja mantida, a aparência gráfica pode variar. A escolha da representação depende do contexto e do objetivo da visualização, destacando as necessidades estéticas ou de clareza. Observe:

Exemplo 2. $G' = (V,A)$ tal que $V = \{p,q,r,s,t\}$ e $A = \{(p,q),(p,r),(q,r),(q,s),(s,t),(r,t)\}$. Assim, $|V| = 5$ e $|A| = 6$.

Na Figura (2), são apresentadas três disposições possíveis, mas não únicas, de representação geométrica do grafo G' do Exemplo (2):

Figura 2 – Exemplo (2) - diferentes representações



Fonte: Autora (2024).

Conforme Oliveira, Rangel e Araujo (2018), dado um grafo $G = (V,A)$ e uma aresta $a = (x,y) \in A$, dizemos que:

(i) x e y são os **extremos** da aresta a;

(ii) a aresta a é dita ser **incidente** nos vértices x e y;

(iii) x e y são chamados de vértices **adjacente**s ou **vizinhos**;

(iv) se $x = y$, a aresta a é chamada de ***loop*** ou **laço**;

(v) se existir uma aresta $b = (w,z)$ tal que $w = x$ e $z = y$, então as arestas a e b são chamadas de arestas **paralelas**.

(vi) Um vértice que não possui nenhuma aresta incidente é chamado de **isolado**.

(vii) Um vértice é **universal** quando está conectado por arestas a todos os demais vértices.

Os grafos são classificados em diversas categorias, além disso, eles podem ter pesos atribuídos às arestas, indicando a magnitude ou custo associado a cada conexão. Entenda as nomenclaturas em cada caso:

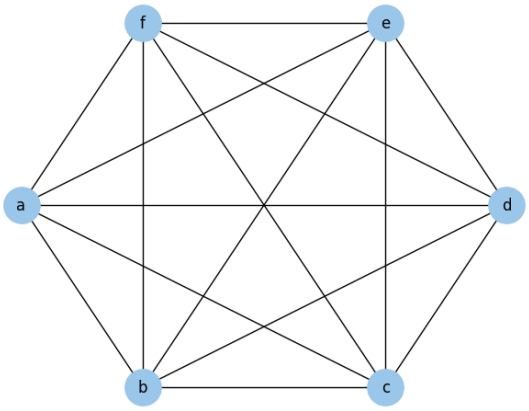
- Um grafo é dito **completo** se existe uma aresta entre cada par de vértices. É representado por K_n, onde n é o número de vértices do grafo. O número de arestas de um grafo completo é dado por $|A| = \dfrac{|V| \cdot (|V|-1)}{2}$. A justificativa se encontra na Subseção (3.3.1).

  Exemplo 3. $K_6 = (V,A)$ tal que $V = \{a,b,c,d,e,f\}$ e $A = \{(a,b),(a,c),(a,d),(a,e),(a,f),(b,c),(b,d),(b,e),(b,f),(c,d),(c,e),(c,f),(d,e),(d,f),(e,f)\}$. No caso de K_6, o grafo

tem 6 vértices e cada vértice está conectado a todos os outros 5 vértices, portanto, ele é completo.

A visualização do grafo K_6 pode ser um hexágono onde todos os vértices são conectados entre si, formando um total de 15 arestas $\left(\text{pois } \frac{6 \cdot 5}{2} = 15\right)$. Observe a Figura (3):

Figura 3 – Exemplo (3) - grafo completo

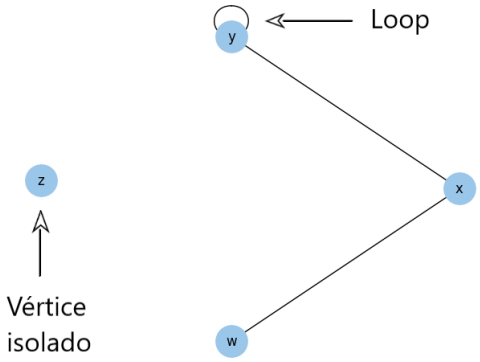


Fonte: Autora (2024).

- Um grafo é **simples** se não possui loops e/ou arestas paralelas. Os Exemplos (1) e (2) são grafos simples.
- Um grafo que contém *loops* é chamado de **pseudografo**.

Exemplo 4. O grafo $G = (V, A)$ tal que $V = \{x, y, z, w\}$ e $A = \{(x, y), (x, w), (y, y)\}$ é um pseudografo, visto que possui *loop*. Veja a Figura (4):

Figura 4 – Exemplo (4) - pseudografo

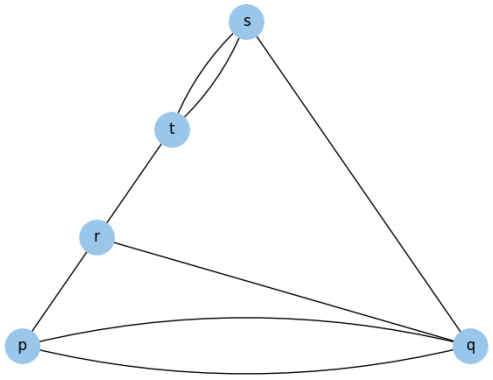


Fonte: Autora (2024).

- Um grafo que contém arestas paralelas é chamado de **multigrafo**.

  Exemplo 5. Seja o multigrafo $G = (V,A)$ tal que $V = \{p,q,r,s,t\}$ e $A = \{(p,q),(q,p)(p,r),(q,r),(q,s),(s,t),(s,t)(r,t)\}$. Para melhor vizualização de suas arestas paralelas, podemos fazê-las curvas, conforme a Figura (5).

Figura 5 – Exemplo (5) - multigrafo

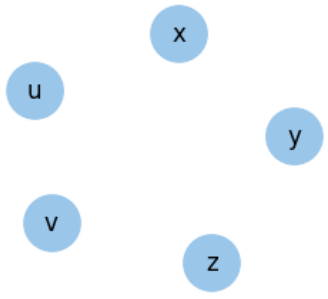


Fonte: Autora (2024).

- Um grafo é dito **nulo** se o seu conjunto de arestas é vazio.

  Exemplo 6. O grafo $G = (V,A)$ tal que $V = \{u,v,x,y,z\}$ e $A = \varnothing$ é nulo. Veja a Figura (6):

Figura 6 – Exemplo (6) - grafo nulo

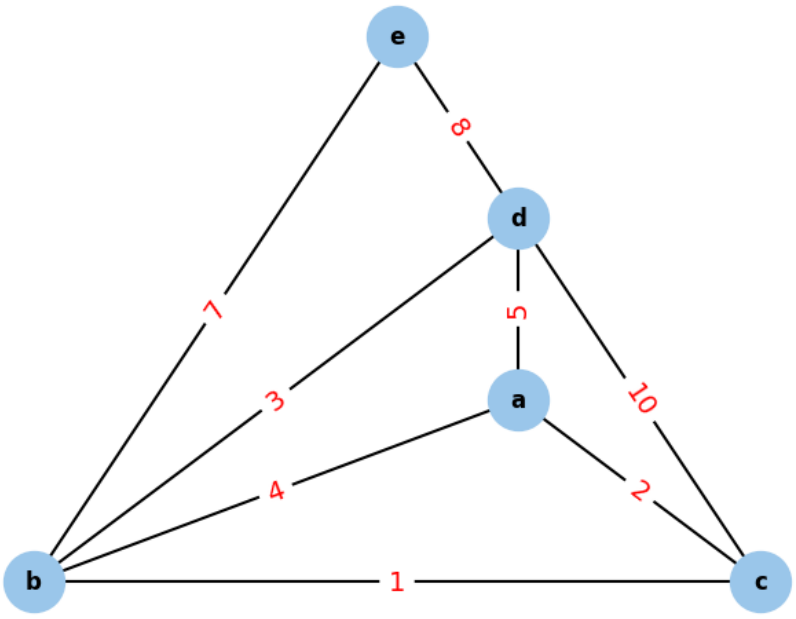


Fonte: Autora (2024).

- Um grafo é dito **ponderado** (ou rede) se são atribuídos valores (pesos) para suas arestas ou vértices.

  Exemplo 7. O grafo $G = (V,A)$ é ponderado, tal que $V = \{a,b,c,d,e\}$ e $\{(a,b,4),(a,c,2),(a,d,5),(b,c,1),(b,d,3),(b,e,7),(c,d,10),(d,e,8)\}$. Cada número indica ao peso da aresta correspondente. Veja sua representação na Figura (7):

Figura 7 – Exemplo (7) - grafo ponderado



Fonte: Autora (2024).

- Um grafo G é dito **planar** se puder ser representado graficamente no plano de tal forma que não haja cruzamento de suas arestas. Os grafos dos Exemplos (2), (5) e (7) são grafos planares.

Segundo Balakrishnan (1997), a estrutura básica de um grafo pode ser direcionada ou não direcionada. Em um **grafo não direcionado**, as arestas não têm uma direção específica, ou seja, o par (x, y) não é ordenado; enquanto em um **grafo direcionado** ou **digrafo**, as arestas possuem uma orientação definida, sendo (x, y) um par ordenado.

Um **grafo direcionado** consiste em um conjunto finito V de vértices e um conjunto E de pares ordenados de vértices distintos chamados **arcos**. Se o par ordenado (x, y) for um arco a, dizemos que o arco a é direcionado de x para y. Neste contexto, o arco a é adjacente ao vértice x e é adjacente ao vértice y. Em um **grafo misto**, haverá pelo menos uma aresta e pelo menos um arco.

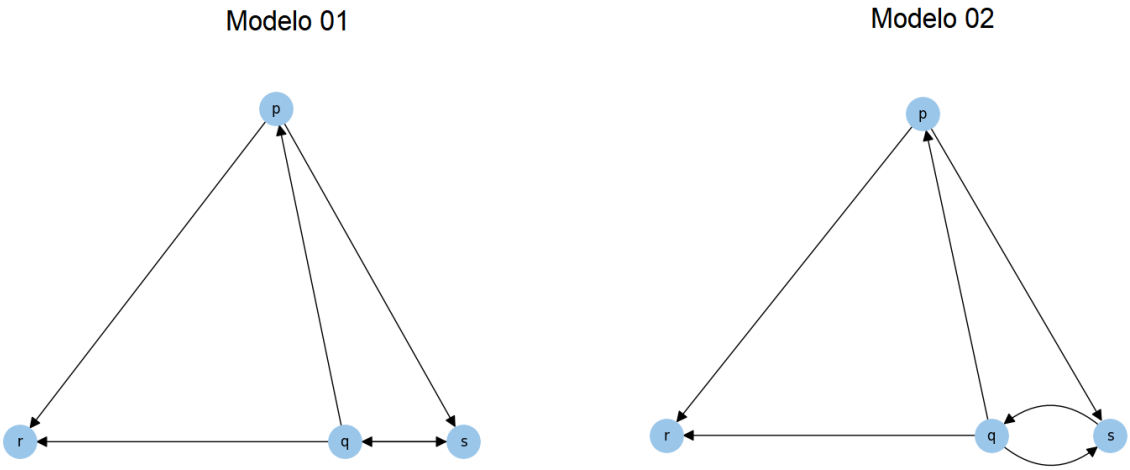
Um digrafo simétrico é um grafo direcionado onde, para cada arco de um vértice x para um vértice y, existe um arco correspondente de y para x. Em outras palavras, se o digrafo contém uma aresta direcionada de (u, v), ele também deve conter a aresta direcionada de (v, u).

Os digrafos também podem ser ilustrados usando um diagrama no qual os vértices são representados por pontos, contudo, cada arco (x, y) é mostrado como uma linha que conecta x a

y, com uma seta apontando para y.

Exemplo 8. Veja na Figura (8) o digrafo $G = (V,A)$, tal que $G = \{p,q,r,s\}$ e $A = \{((p,r),(p,s),(q,p),(q,r),(q,s),(s,q)\}$.

Figura 8 – Exemplo (8) - digrafo



Fonte: Autora (2024).

3.2 Vizinhança

Antes de avançar nas propriedades dos grafos, é importante introduzir alguns conceitos fundamentais que guiarão as análises. A compreensão da relação entre vértices, tanto em grafos quanto em digrafos, é essencial para a correta modelagem de diversos problemas. Para isso, serão utilizadas as definições propostas pelos autores Netto, Oswaldo e Jurkiewicz (2009), que esclarecem a noção de vizinhança, conforme será abordado a seguir.

Definição 3. *Denota-se $N(x)$ o **conjunto dos vértices vizinhos** (adjacentes) de x. Isto é, se existir em um grafo $G = (V,A)$ uma aresta (x,y), então y é vizinho de x e portanto $y \in N(x)$.*

Em um grafo direcionado (digrafo), os vizinhos de um vértice x se subdividem em sucessores e antecessores. Se designa a saída por um sobrescrito $(+)$ e a entrada por um $(-)$.

*Os **sucessores** de um vértice são aqueles vértices que podem ser alcançados a partir dele seguindo as direções das setas. Em outras palavras, se houver um arco (a,b), então b é sucessor de a, ou seja, $b \in N^+(a)$.*

*Os **antecessores** de um vértice são aqueles vértices que possuem uma aresta saindo deles e chegando ao vértice analisado. Portanto, se há um arco (c,d), então c é um antecessor de d,*

ou seja, $c \in N^-(d)$.

3.3 O Grau de um Vértice

A partir da noção de vizinhança, é possível apresentar o conceito de grau de um vértice, que descreve sua conectividade no grafo, sendo essencial para caracterizar a estrutura da rede. Segundo Chartrand, Lesniak e Zhang (2010), temos a seguinte definição:

Definição 4. ***O grau de um vértice*** *x, representado por* $d(x)$*, em um grafo G sem loops, é definido como o número de vértices em G que são adjacentes a x. Caso haja loops, estas arestas contribuem com grau 2.*

Em outras palavras, o grau de x é a quantidade de vértices em sua vizinhança $N(x)$. Alternativamente, podemos dizer que o grau de x é igual ao número de arestas que incidem em x.

O maior grau entre os vértices de G é denominado **grau máximo** de G e é simbolizado por $\Delta(G)$. O menor grau entre os vértices de G é denominado **grau mínimo** de G e é simbolizado por $\delta(G)$. Assim, se x é um vértice de um grafo G com $|V|$ vértices, então:

$$0 \leq \delta(G) \leq d(x) \leq \Delta(G) \leq |V| - 1. \tag{1}$$

De acordo com Oliveira, Rangel e Araujo (2018), podemos classificar os vértices de um grafo da seguinte maneira:

(i) Um vértice x é **isolado** se $d(x) = 0$.

(ii) Um vértice x é **pendente** se $d(x) = 1$.

(iii) Se x é um vértice **universal**, então $d(x) = |V| - 1$.

(iv) Uma aresta que incide em um vértice pendente é chamada de **aresta pendente**.

(v) Um grafo é dito **regular** se todos os seus vértices possuem o mesmo grau.

Em um grafo orientado, a definição (4) também é aplicável, porém podemos distinguir entre duas vizinhanças: os sucessores e os antecessores de um vértice. O número de sucessores de um vértice x é chamado de **semigrau exterior**, denotado por $d^+(x)$ enquanto o número de antecessores é denominado **semigrau interior** e denotado por $d^-(x)$. O grau total de um vértice é então a soma desses dois semigraus.

3.3.1 Teorema do Aperto de Mãos

O teorema apresentado apresentado a seguir é um resultado clássico em teoria dos grafos, utilizado para descrever a relação entre os vértices e arestas de um grafo simples não direcionado. Esse teorema é aplicado em diversos contextos, como análise de redes sociais, interações em sistemas biológicos e outras áreas que envolvem conexões entre diferentes entidades. Ele fornece uma forma simples de contar o número total de arestas de um grafo ao observar os graus dos vértices.

Esse teorema é informalmente chamado de "Teorema do Aperto de Mãos"por analogia a uma situação social em que pessoas se cumprimentam, simbolizando a conexão entre os vértices. Cada aperto de mão entre duas pessoas corresponde a uma aresta, e como o aperto envolve duas pessoas, cada aresta é contada duas vezes.

Agora, o teorema é enunciado formalmente de acordo com Souza (2014):

Teorema 1. *Para qualquer grafo simples não direcionado $G = (V,A)$, a soma dos graus de todos os vértices é igual ao dobro do número de arestas. Matematicamente, isso pode ser escrito como:*

$$\sum_{x \in V} d(x) = 2|A| \tag{2}$$

Demonstração. A contagem de todas as incidências de arestas é $\sum_{x \in V} d(x)$. Cada aresta do grafo contribui exatamente uma vez para o grau de dois vértices (um para cada extremidade da aresta), desta maneira, cada aresta é contada duas vezes na soma $\sum_{x \in V} d(x)$. Portanto, $\sum_{x \in V} d(x) = 2|A|$. □

Como consequência direta do Teorema do Aperto de Mãos, como discutido por Oliveira, Rangel e Araujo (2018), temos o seguinte corolário:

Corolário 1. *Todo grafo não direcionado tem uma quantidade par de vértices de grau ímpar.*

Demonstração. Vamos dividir a soma (2) em duas parcelas. Os vértices com grau par e os vértices com grau ímpar:

$$\sum_{x \in V} d(x) = \sum_{\text{grau par}} d(x) + \sum_{\text{grau ímpar}} d(x) \tag{3}$$

O lado esquerdo da equação (3) é par (pelo Teorema (1)). A primeira parcela do lado direito também é par, pois é a soma de números pares. Para que a igualdade seja válida, a segunda parcela também deve ser par.

$$\sum_{\text{grau ímpar}} d(x) \text{ é par.} \tag{4}$$

Como cada parcela $d(x)$ de (4) é ímpar, temos que ter um número par de elementos para que a soma seja um número par. Portanto, uma consequência direta do Teorema (1) é que todo grafo não direcionado tem uma quantidade par de vértices de grau ímpar. □

Corolário 2. *O número máximo de arestas em um grafo é dado por*

$$|A| = \frac{|V| \cdot (|V| - 1)}{2} \tag{5}$$

Demonstração. O número máximo de arestas de um grafo ocorre quando todos os seus vértices estão conectados uns aos outros, ou seja, quando o grafo é completo. Se um grafo completo possui $|V|$ vértices, então cada vértice está conectado a $|V - 1|$ outros vértices.

Aplicando o Teorema (1), temos:

$$\sum_{x \in V} d(x) = 2|A| = |V| \cdot (|V| - 1) \tag{6}$$

O que implica,

$$|A| = \frac{|V| \cdot (|V| - 1)}{2} \tag{7}$$

□

Este resultado nos diz que qualquer que seja a quantidade de arestas de um grafo, ela será sempre menor ou igual à quantidade de arestas de um grafo completo com o mesmo número de vértices.

3.4 Lista de adjacência

Para representar grafos de maneira mais prática, em vez de escrevermos todos os conjuntos podemos fazer uma lista de vértices e seus vizinhos.

Definição 5. *Uma lista de adjacência é uma estrutura de dados usada para representar grafos, onde cada vértice possui uma lista que contém todos os vértices adjacentes a ele.*

A lista de adjacência do Exemplo (1) é:

Vértice	Vizinhança
p	q,r,s
q	s
r	
s	

A lista de adjacência do Exemplo (2) é:

Vértice	Vizinhança
p	q,r
q	p,r,s
r	p,q,t
s	q,t
t	s,r

3.5 Matriz de adjacência

Nas resoluções de modelos de grafos usa-se computadores, visto que muitos cálculos são trabalhosos e complexos para se fazer manualmente. Os computadores não manipulam através de desenhos, então convém utilizar de matrizes para para armazenar, representar de forma única e manipular um grafo no computador. Conforme Oliveira, Rangel e Araujo (2018):

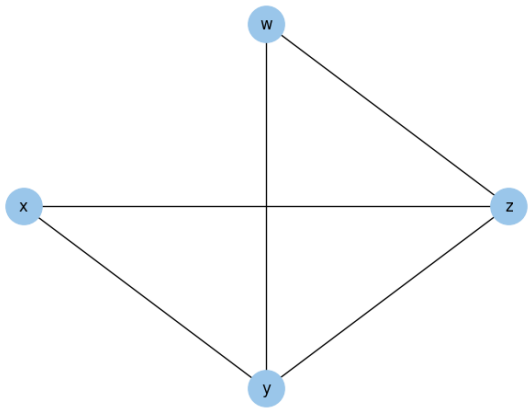
Definição 6. *Considere um grafo $G = (V,A)$ de ordem $|V|$. A **matriz de adjacência** é uma matriz quadrada, de dimensão $|V| \times |V|$, representada por $M = (m_{ij})$ é definida da seguinte forma:*

$$m_{ij} = \begin{cases} 1, \text{ se existe uma aresta entre os vértices } v_i \text{ e } v_j, \\ 0, \text{ caso contrário}. \end{cases}$$

Exemplo 9. Faremos a matriz M de adjacência do grafo $G = (V,A)$ representado na Figura (9), tal que $V = \{x,y,z,w\}$ e $A = \{(x,y),(x,z),(y,z),(y,w),(z,w)\}$. Assim:

$$M = \begin{array}{c} \\ x \\ y \\ z \\ w \end{array} \begin{array}{c} \begin{array}{cccc} x & y & z & w \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Figura 9 – Exemplo (9) - representação da matriz de adjacência



Fonte: Autora (2024).

Considerações:

(i) A matriz de adjacência pode representar apenas grafos que não possuem arestas paralelas.

(ii) Se G é ponderado, então m_{ij} contém o peso da aresta (v_i, v_j) em vez de 1.

(iii) As entradas ao longo da diagonal principal de M são todas zeros se, e somente se, o grafo não possui *loops*. Quando há um *loop* em um vértice v_i, então $m_{ii} = 1$.

(iv) Se o grafo é simples, o grau $d(x)$ de um vértice x é dado pela soma dos elementos de sua linha (ou coluna) correspondente.

(v) Se G é não direcionado, então a matriz M correspondente é simétrica ao longo da diagonal principal; isto é, a entrada $m_{ij} = m_{ji}$.

(vi) Se G é direcionado, a matriz M só será simétrica se o digrafo for simétrico.

(vii) Num digrafo sem pesos, a entrada m_{ij} da matriz M é 1 se há um arco de v_i para v_j e 0 caso contrário.

(viii) Num digrafo com pesos, a entrada m_{ij} da matriz M representa o peso do arco de v_i para v_j.

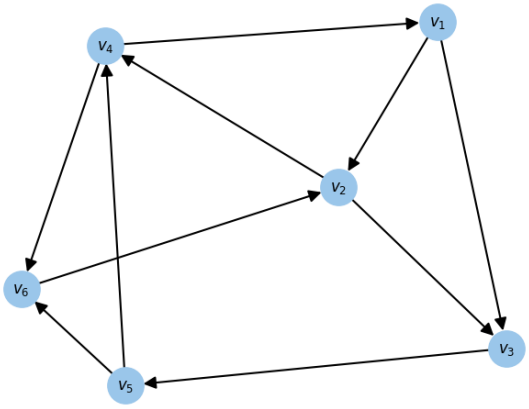
(ix) Se o grafo é direcionado, o grau de saída do vértice v_i é dado pela soma dos elementos da linha i. O grau de entrada do vértice v_i é dado pela soma dos elementos da coluna i.

(x) Se M é a matriz de adjacência de um digrafo G, então a sua transposta M^T é a matriz de adjacência do digrafo obtido pela inversão da orientação das arestas de G.

Exemplo 10. Faremos a matriz M de adjacência do digrafo $G = (V,A)$, representado na Figura (10), tal que $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ e $A = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_2, v_4), (v_3, v_5), (v_4, v_1), (v_4, v_6), (v_5, v_4), (v_5, v_6), (v_6, v_2)\}$.

$$M = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Figura 10 – Exemplo (10) - representação da matriz de adjacência de um digrafo

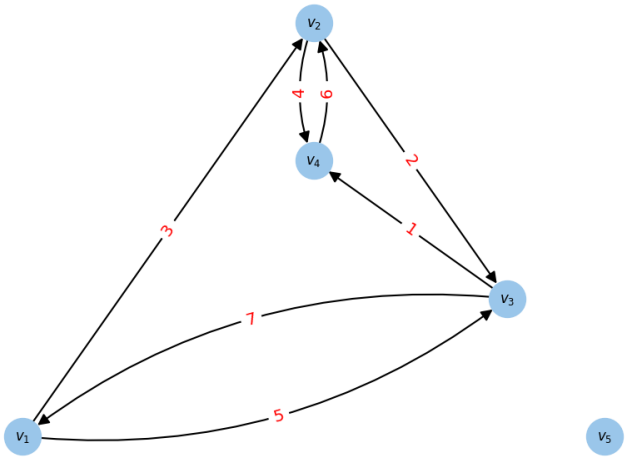


Fonte: Autora (2024).

Exemplo 11. Faremos a matriz M de adjacência do digrafo ponderado $G = (V,A)$, representado na Figura (11), tal que $V = \{v_1, v_2, v_3, v_4, v_5\}$ e $A = \{(v_1, v_2, 3), (v_1, v_3, 5), (v_2, v_3, 2), (v_3, v_1, 7), (v_3, v_4, 1), (v_2, v_4, 4), (v_4, v_2, 6)\}$.

$$M = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{pmatrix} 0 & 3 & 5 & 0 & 0 \\ 0 & 0 & 2 & 4 & 0 \\ 7 & 0 & 0 & 1 & 0 \\ 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Figura 11 – Exemplo (11) - representação da matriz de adjacência de um digrafo ponderado



Fonte: Autora (2024).

3.6 Matriz de incidência

Enquanto a matriz de adjacência armazena as relações vértice-vértice, a matriz de incidência representa vértice-aresta.

De acordo com Oliveira, Rangel e Araujo (2018):

Definição 7. *Considere um grafo $G = (V,A)$ de ordem $|V|$ e tamanho $|A|$. A* ***matriz de incidência*** *é uma matriz de dimensão $|V| \times |A|$, representada por $N = (n_{ij})$ e definida da seguinte forma:*

$$n_{ij} = \begin{cases} 1, \text{ se aresta } a_j \text{ é incidente no vértice } v_i, \\ 0, \text{ caso contrário.} \end{cases}$$

Considerações:

(i) Apenas grafos que não possuem *loops* podem ser armazenados em uma matriz de incidência.

(ii) a matriz de incidência é adequada para representar grafos que possuem arestas paralelas.

(iii) Como cada aresta é incidente em exatamente dois vértices, cada coluna de N contém exatamente dois valores iguais a 1.

(iv) O grau $d(x)$ de um vértice x é dado pela soma dos elementos de sua linha correspondente.

(v) Uma linha composta apenas por zeros indica um vértice isolado, ou seja, um vértice que não está conectado a nenhuma aresta.

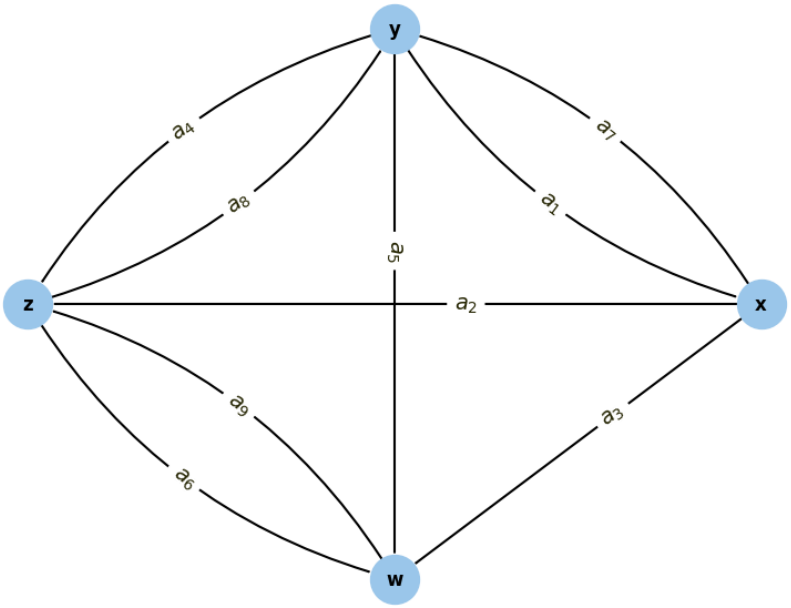
(vi) Arestas paralelas são representadas por colunas idênticas na matriz de incidência.

(vii) Caso as arestas possuam pesos, basta escrever o valor do peso no lugar de 1.

Exemplo 12. Faremos a matriz N de incidência do grafo $G = (V,A)$ representado na Figura (12), tal que $V = \{x,y,z,w\}$ e $A = \{a_1 = (x,y), a_2 = (x,z), a_3 = (x,w), a_4 = (y,z), a_5 = (y,w), a_6 = (z,w), a_7 = (x,y), a_8 = (y,z), a_9 = (z,w)\}$. Assim:

$$N = \begin{array}{c} \\ x \\ y \\ z \\ w \end{array} \begin{array}{c} \begin{array}{ccccccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Figura 12 – Exemplo (12) - representação da matriz de incidência



Fonte: Autora (2024).

Para representar um grafo direcionado, a matriz de incidência requer mais atenção. Observe:

Definição 8. *Seja $G = (V,A)$ um digrafo com $|V|$ vértices e $|A|$ arestas e sem loop. Sua matriz*

de incidência $N = (a_{ij})$ *de dimensão* $|V| \times |A|$ *é definida como:*

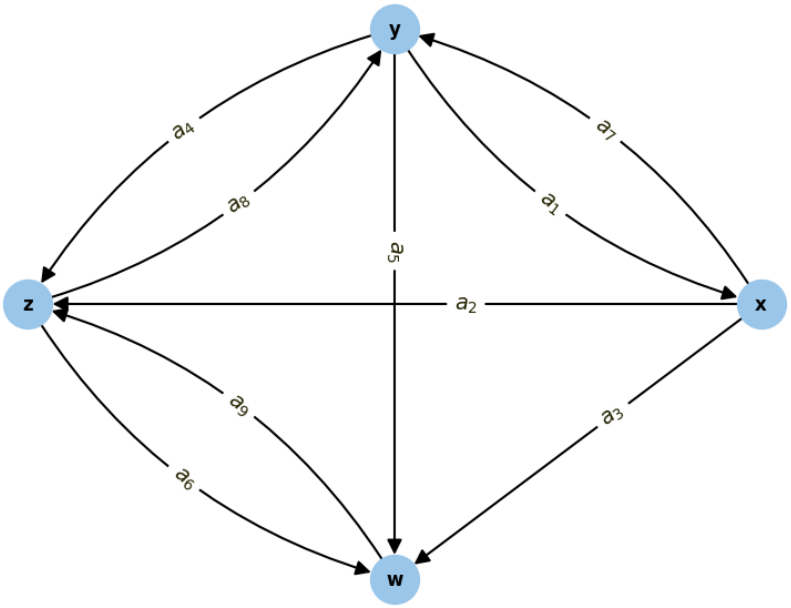
$$n_{ij} = \begin{cases} +1, \text{ se o vértice } v_i \text{ é saída do arco } a_j, \\ -1, \text{ se o vértice } v_i \text{ é entrada do arco } a_j, \\ 0, \text{ se o vértice } v_i \text{ não é extremo do arco } a_j. \end{cases}$$

Caso as arestas possuam pesos, basta escrever o valor do peso no lugar de 1, preservando os sinais $+$ ou $-$ de direção da incidência nos vértices extremos da aresta em questão.

Exemplo 13. Faremos a matriz N de incidência do grafo G do exemplo (12), caso ele fosse direcionado, como representado na Figura (13). Assim, $A = \{a_1 = (x,y), a_2 = (x,z), a_3 = (x,w), a_4 = (y,z), a_5 = (y,w), a_6 = (z,w), a_7 = (y,x), a_8 = (z,y), a_9 = (w,z)\}$. Logo:

$$N = \begin{array}{c} \\ x \\ y \\ z \\ w \end{array} \begin{array}{c} \begin{array}{ccccccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 \end{array} \\ \begin{pmatrix} +1 & +1 & +1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & +1 & +1 & 0 & +1 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 & 0 & +1 & -1 \\ 0 & 0 & -1 & 0 & -1 & -1 & 0 & 0 & +1 \end{pmatrix} \end{array}$$

Figura 13 – Exemplo (13) - representação da mtriz de incidência de um digrafo



Fonte: Autora (2024).

3.7 Passeios

Nesta seção, são explorados os conceitos fundamentais relacionados a passeios, que representam sequências de vértices e arestas em grafos, ajudando a compreender a conectividade e a estrutura das redes. Conforme Scheinerman (2011), são definidos e diferenciados passeios, trajetos e caminhos, cada um com suas características e aplicações específicas. Além disso, aborda-se o conceito de conexidade em grafos e ciclos, elementos essenciais para entender as propriedades estruturais dos grafos.

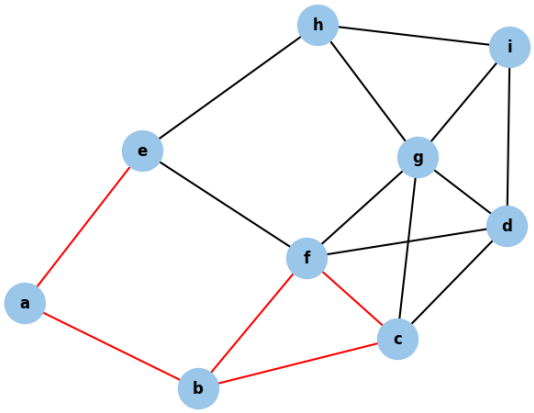
Dado um grafo $G = (V, A)$, podemos definir:

Definição 9. *Um **passeio** em G é uma sequência finita que alterna entre vértices e arestas, começando e terminando em vértices, onde cada aresta está conectada aos vértices imediatamente anteriores e posteriores na sequência. Um passeio pode visitar vértices e arestas mais de uma vez.*

Definição 10. *O **comprimento** de um passeio é dado pelo seu número de arestas. Já em um grafo ponderado, o comprimento do percurso é dado pela soma dos pesos de suas arestas. Se uma aresta aparece mais de uma vez, então ela também é contada mais de uma vez.*

Exemplo 14. Seja $G = (V, A)$, tal que $V = \{a, b, c, d, e, f, g, h, i\}$ e $A = \{(a,b), (a,e), (b,c), (b,f), (c,f), (c,g), (c,d), (d,f), (d,g), (d,i), (e,f), (e,h), (f,g), (g,h), (g,i), (h,i)\}$. Temos que $a \to b \to c \to f \to b \to a \to e$ é um passeio de G, com comprimento 6, conforme a Figura (14).

Figura 14 – Exemplo (14) - passeio



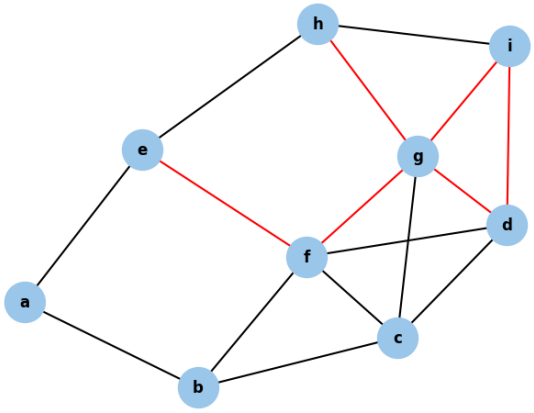
Fonte: Autora (2024).

Definição 11. *Um **trajeto** em G é um passeio onde nenhuma aresta é repetida. Um trajeto pode visitar vértices mais de uma vez, mas não a mesma aresta. Um trajeto no qual o vértice inicial e*

o final são iguais é chamado de ***trajeto fechado****.*

Exemplo 15. Seja $G =$ do Exemplo (14). Temos que $e \to f \to g \to d \to i \to g \to h$ é um trajeto de G, conforme a Figura 15.

Figura 15 – Exemplo (15) - trajeto

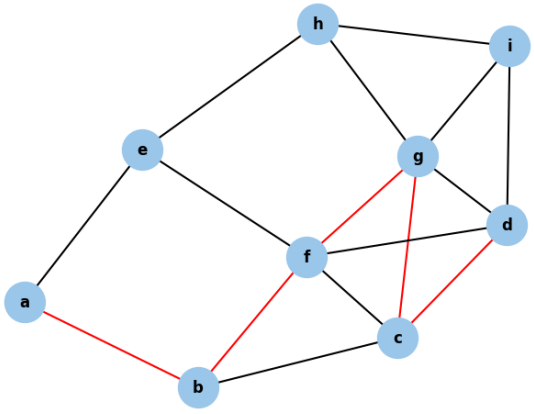


Fonte: Autora (2024).

Definição 12. *Um* ***caminho*** *é um passeio sem repetição de vértices. Consequentemente, não há repetição de arestas.*

Exemplo 16. Seja $G =$ do Exemplo (14). Temos que $a \to b \to f \to g \to c \to d$ é um caminho de G, conforme a Figura (16).

Figura 16 – Exemplo (16) - caminho



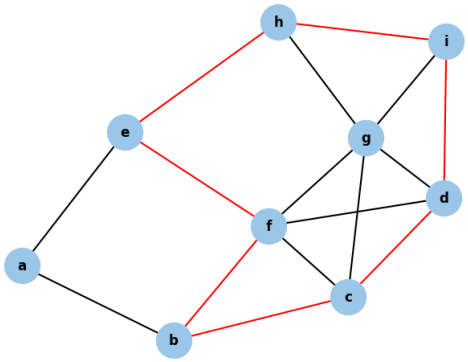
Fonte: Autora (2024).

Definição 13. *Um grafo $G = (V,A)$ é* ***conexo*** *se todos os seus pares de vértices estão ligados por um caminho. Caso contrário, é* ***desconexo****, se há pelo menos um par de vértices que não está ligado a nenhum caminho.*

Definição 14. *Um trajeto fechado no qual nenhum vértice (com exceção do inicial e do final) aparece mais de uma vez é chamado de* ***ciclo****. Consequentemente, não há repetição de arestas.*

Exemplo 17. Seja G do Exemplo (14). Temos que $c \to b \to f \to e \to h \to i \to d \to c$ é um ciclo de G, conforme a Figura (17).

Figura 17 – Exemplo (17) - ciclo



Fonte: Autora (2024).

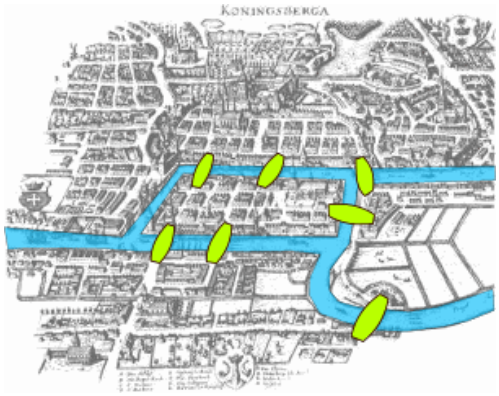
Os conceitos relacionados a passeios são aplicáveis tanto a grafos quanto a digrafos. Vale ressaltar que as terminologias podem variar conforme o autor ou o idioma consultado.

3.8 As pontes de Königsberg

O Problema de Euler é um dos problemas clássicos na teoria dos grafos, formulado pelo matemático suíço Leonhard Euler em 1735 (Souza, 2014). Durante sua estadia na cidade de Königsberg (atual Kaliningrado, Rússia), Euler deparou-se com um enigma que, embora aparentemente simples, ainda permanecia sem solução.

O problema envolvia as águas de um rio que atravessava a cidade, onde duas ilhas eram conectadas por uma ponte. Além disso, as duas ilhas estavam ligadas às margens por mais seis pontes, conforme ilustrado na Figura (18).

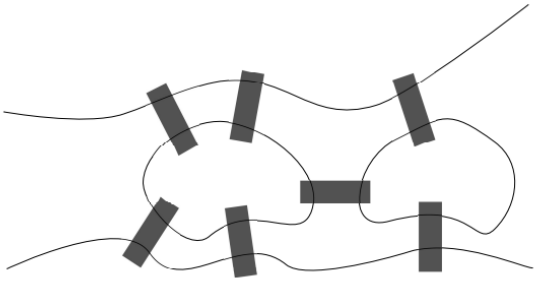
Figura 18 – As pontes de Königsberg



Fonte: Giuşcă (2005).

O desafio era descobrir uma rota para um passeio que iniciasse em uma margem, cruzasse cada ponte exatamente uma vez e retornasse à margem de origem.

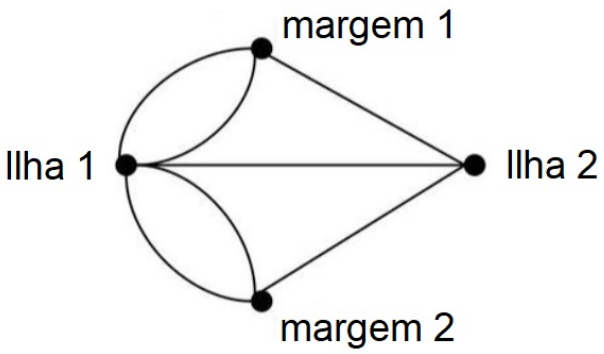
Figura 19 – As 7 pontes



Fonte: Oliveira, Rangel e Araujo (2018).

Euler (1741) resolveu essa questão em seu artigo “Solutio problematis ad geometriam situs pertinens” e sua tradução para o Português foi feita por Lopes e Táboas (2015) . Em seu texto, Euler representou o problema através de um desenho com 4 pontos (vértices), correspondentes as áreas de terra, e 7 linhas (arestas) que conectam esses vértices, representando as pontes; como podemos ver na Figura (20). Isso permitiu que ele se concentrasse nas propriedades estruturais do grafo em vez dos detalhes físicos das pontes e terrenos.

Figura 20 – Representação de modelo de grafo do problema de Euler



Fonte: Valci (2022).

Euler estabeleceu que para ser possível traçar um caminho que atravessasse cada ponte exatamente uma vez, cada margem precisaria estar conectada à outra por um número par de pontes. Isso porque, a entrada e saída de cada vértice devem se "equilibrar" (par), exceto no ponto inicial e no ponto final, que poderiam ser ímpares.

Desse modo, conta-se o número de pontes (arestas) que entravam e saíam de cada área de terra (vértice). Assim, a análise do problema envolveu examinar o número de arestas incidentes a cada vértice, ou seja, os graus dos vértices. Cada vértice em Königsberg possuía grau ímpar:

- Duas áreas tinham três pontes conectando-as (grau 3);
- Duas áreas tinham cinco pontes conectando-as (grau 5).

Em suma, para ser possível tal percurso (um trajeto euleriano), o grafo deve ter exatamente zero ou dois vértices de grau ímpar. No caso de Königsberg, todos os quatro vértices tinham grau ímpar, tornando impossível o percurso desejado.

Este problema nos permitiu nomear grafos cujo percurso sugerido anteriormente seja possível. Veremos sobre esse tipo de percurso na Seção (3.9), que trata de caminhos eulerianos.

Euler resolveu o problema específico das pontes de Königsberg como um desafio intelectual, sem muita importância, e não procurou expandir ou aplicá-lo. Apenas 111 anos depois, em 1847, o físico Gustav Kirchhoff , nascido em Königsberg, utilizou modelos de grafos para analisar circuitos elétricos. Ao longo do século XX, outros intelectuais estudaram problemas que também puderam ser resolvidos através de modelos de grafos, e assim contribuíram para o desenvolvimento do corpo téorico. Durante a década de 1930, Dénes Kőnig e outros matemáticos formalizaram muitos conceitos fundamentais da teoria dos grafos, que hoje utilizamos para abordar problemas novos.

Principalmente devido o surgimento e aprimoramento do computador, a teoria dos grafos é aplicada nas mais variadas áreas, em busca de melhores soluções para problemas de projeto, organização e distribuição:

- Ciência da Computação: Algoritmos para redes, otimização e inteligência artificial.
- Biologia: Estudo de redes de interação genética e ecológica.
- Sociologia: Análise de redes sociais.
- Engenharia: Planejamento e análise de redes de transporte e comunicação.

Exemplos destas aplicações e outras citadas ao longo do trabalho podem ser encontradas nas obras de Gross e Yellen (2006), Bondy e Murty (1976) e Barabási (2016).

O estudo de grafos continua a evoluir, impulsionado por novos desafios e aplicações em áreas multidisciplinares, consolidando-se como uma ferramenta poderosa para resolver problemas complexos. O trabalho de Euler não apenas resolveu o problema local, mas também estabeleceu as bases para a topologia e a combinatória, mostrando que a configuração espacial específica das pontes era irrelevante; o importante era a estrutura abstrata e, com isso, deu início à teoria dos grafos.

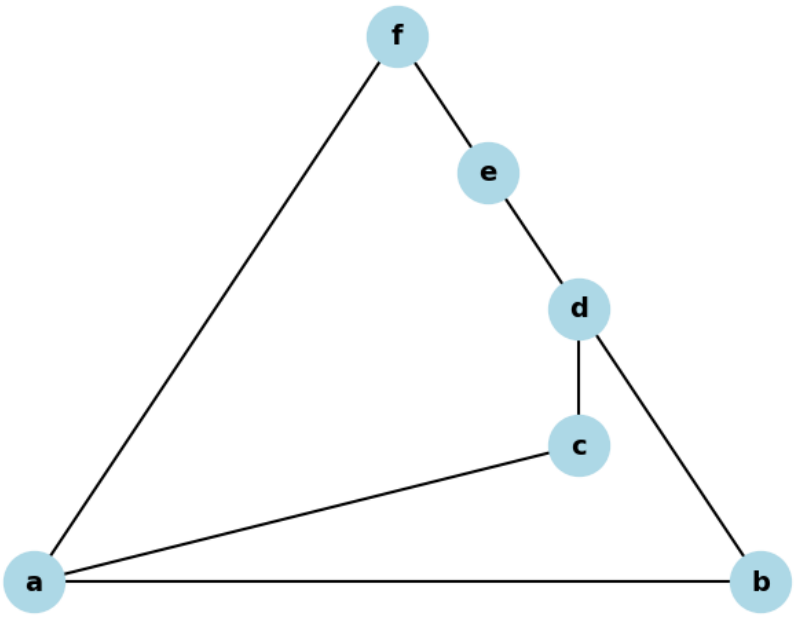
3.9 Trajetos Eulerianos

Os trajetos eulerianos são assim chamados em homenagem à Leonhard Euler. O problema das Pontes de Königsberg, é o exemplo histórico mais conhecido desse tipo de passeio. Atualmente, além de aplicados em problemas de rotas, os trajetos eulerianos também são usados em passatempos como desenhar figuras sem levantar o lápis do papel. Segundo Malta (2008):

Definição 15. *Dado um grafo $G = (V,A)$, um **trajeto euleriano** é um trajeto em G que contém todas as arestas do grafo e que visita toda aresta exatamente uma vez. Ele pode ser fechado (começa e termina no mesmo vértice) ou aberto (começa e termina em vértices diferentes); geralmente trajetos eulerianos abertos são chamados de trajetos semieulerianos.*

Exemplo 18. Seja o grafo $G = (V,A)$ tal que $V = \{a,b,c,d,e,f\}$ e $A = \{(a,b),(a,c),(b,d),(c,d),(d,e),(e,f),(f,a)\}$. Temos que $d \to b \to a \to f \to e \to d \to c \to a$ é um trajeto euleriano. Observe sua representação na Figura (21).

Figura 21 – Exemplo (18) - trajeto euleriano



Fonte: Autora (2024).

Nos trajetos eulerianos, precisamos passar por todas as arestas do grafo e não podemos repeti-las. Não há problema em repetir vértices. Como consequência dessa exigência decorre uma condição para que um grafo tenha trajeto euleriano ou não:

Teorema 2. ***Teorema de Euler (1741):*** *Um grafo conexo $G = (V,A)$ admite trajeto euleriano se, e somente se, todos os vértices tiverem grau par ou, apenas dois tiverem grau ímpar.*

Demonstração. $(\Rightarrow)$

Se os vértices inicial e final do trajeto são distintos, eles são os únicos que podem ter grau ímpar. Orientando as arestas seguindo o sentido do trajeto, cada vértice intermediário x terá exatamente metade de suas arestas, $\frac{d(x)}{2}$, como entradas e a outra metade como saídas. Isso garante que o trajeto possa prosseguir sem repetir arestas em qualquer momento. Portanto, o grau $d(x)$ de qualquer vértice intermediário deve ser par.

Se o trajeto é fechado, ou seja, os vértices inicial e final do trajeto são iguais, então todos os vértices são considerados intermediários e não poderão existir vértices de grau ímpar.

$(\Leftarrow)$

Sejam u e v os dois únicos vértices de grau ímpar em G. Ao construirmos um trajeto começando por u, observamos que, ao chegar a qualquer vértice intermediário x, utilizamos um número ímpar de suas arestas incidentes. Como todo vértice diferente de u e v tem grau par,

sempre haverá ao menos uma aresta disponível para o trajeto prosseguir. Caso isso não ocorra em algum vértice, esse vértice deve ser v, devido ao seu grau ímpar.

Construímos então um trajeto de u até v, sem a necessidade de utilizar todas as arestas disponíveis. Após isso, sobra um grafo parcial G' contendo as arestas não utilizadas, que pode ser não conexo.

A partir de u, podemos desviar o trajeto sempre que atingirmos uma componente conexa de G', percorrendo um trajeto euleriano sobre essa componente antes de prosseguir. Isso será sempre possível, uma vez que todos os vértices terão grau par. Ao atingir v, teremos percorrido um trajeto euleriano em G.

Se todos os vértices de G tiverem grau par, retiramos uma aresta para formar um novo grafo G'' com os vértices terminais u e v. Executamos então a sequência descrita acima, reintegrando a aresta ao alcançar v, o que permitirá o fechamento do trajeto euleriano. □

Observação: O Teorema (2) pode ser dividido em dois casos:

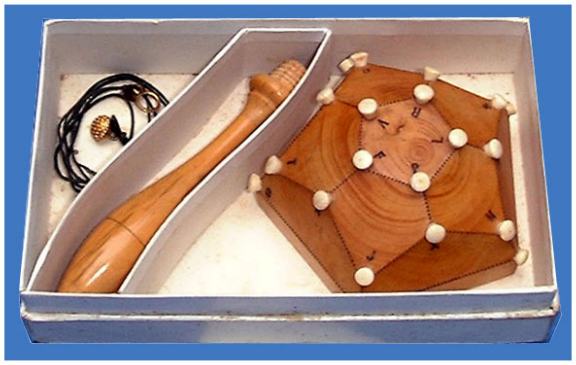
(i) Um grafo conexo $G = (V,A)$ admite trajeto euleriano (fechado) se, e somente se, todos os vértices tiverem grau par.

(ii) Um grafo conexo $G = (V,A)$ admite trajeto semieuleriano (aberto) se, e somente se, apenas dois vértices tiverem grau ímpar. Neste caso, a trilha começa em um vértice de grau ímpar e termina no outro vértice de grau ímpar.

3.10 Caminhos Hamiltonianos

Em 1857, um matemático irlandês chamado William Rowan Hamilton inventou um quebra-cabeça chamado "Jogo Icosiano". Este jogo consistia em um dodecaedro regular (um poliedro com 20 vértices 12 faces pentagonais) cujos vértices eram rotulados com nomes de cidades importantes da época, conforme a Figura (22). O objetivo do jogo era encontrar um ciclo que passasse por cada cidade exatamente uma vez, voltando à cidade inicial. Hamilton patenteou o jogo, mas, apesar de sua elegância matemática, o jogo não foi um sucesso comercial na época.

Figura 22 – Jogo Icosiano



Fonte: Museum (2017).

Embora a proposta do problema de Euler e de Hamilton pareçam ser semelhantes, a análise e complexidade da resolução estão em diferentes níveis de dificuldade. Enquanto o problema de Euler tratava de atravessar todas as arestas (pontes) de um grafo, o problema de Hamilton focava em visitar todos os vértices (cidades), diferenciando-se claramente em termos de objetivos e complexidade. Devido ao fato de que o dodecaedro regular de madeira era pesado e difícil de manusear, uma nova versão do jogo foi desenvolvida utilizando uma figura planificada (Souza, 2014). Uma solução do jogo pode ser vista na Figura (23).

Figura 23 – Solução do Jogo Icosiano

Fonte: Library (2019).

Definição 16. *Seja G um grafo. Um* ***caminho hamiltoniano*** *em G é um caminho que passa uma única vez por cada vértice.*

Definição 17. *Um* ***ciclo hamiltoniano*** *é um caminho hamiltoniano que retorna ao vértice inicial.*

A busca por caminhos e ciclos Hamiltonianos deu origem a vários problemas relacionados na teoria dos grafos e em outros campos, como o Problema do Caixeiro Viajante - veja mais em

(Wilhelm, 2005), que é amplamente estudado em otimização e ciência da computação.

Os conceitos de caminhos e ciclos Hamiltonianos têm diversas aplicações práticas, segundo Gross e Yellen (2006), Bondy e Murty (1976) e Barabási (2016), incluindo:

- Rotas de Veículos: Encontrar rotas eficientes para entrega de mercadorias.
- Planejamento de Circuitos: Otimização de circuitos em engenharia elétrica.
- Bioinformática: Sequenciamento de DNA onde é necessário encontrar caminhos que representem a sequência de nucleotídeos.

Não se conhece uma condição necessária e suficiente simples para a existência de um ciclo hamiltoniano em um grafo. "Na verdade, todos os teoremas se encontram muito longe de oferecer uma previsão razoável de solução"(Jurkiewicz, 2009).

O estudo dos teoremas e demonstrações sobre Caminhos Hamiltonianos não é o foco deste estudo. Para saber mais, leia (Oliveira, Rangel e Araujo, 2018).

3.11 O Problema das Quatro cores

O Problema das Quatro Cores é um dos problemas mais famosos e históricos na teoria dos grafos e na matemática em geral. Ele afirma que qualquer mapa plano (ou, de forma equivalente, qualquer grafo planar) pode ser colorido com não mais que quatro cores de forma que regiões vizinhas (ou vértices adjacentes) não compartilhem a mesma cor.

O problema foi proposto pela primeira vez em 1852 por Francis Guthrie, um estudante britânico, que notou que quatro cores eram suficientes para colorir os mapas dos condados da Inglaterra sem que duas regiões adjacentes tivessem a mesma cor. Ele compartilhou a conjectura ao seu irmão Frederick Guthrie, que levou a questão a seu professor Augustus De Morgan. De Morgan, por sua vez, comunicou o problema ao matemático William Rowan Hamilton. Ele, por sua vez, não achou o problema interessante e respondeu quatro dias mais tarde dizendo que não pretendia debruçar-se sobre a questão tão cedo.

Ao longo dos anos, muitos matemáticos tentaram provar a conjectura das quatro cores. Em 1879, Alfred Kempe anunciou uma prova que foi amplamente aceita pela comunidade matemática até que, em 1890, Percy Heawood encontrou uma falha na prova de Kempe. Apesar disso, as ideias de Kempe foram úteis e usadas em trabalhos subsequentes.

A primeira prova correta do teorema das quatro cores foi apresentada em 1976 por Kenneth Appel e Wolfgang Haken, usando um método que envolvia extensivamente a verificação

por computador. Eles reduziram o problema a um grande número de casos específicos, que foram então verificados por um programa de computador. Esta prova foi significativa porque foi a primeira vez que um computador foi usado de forma tão crucial na verificação de um teorema matemático e marcou o início de uma era onde a colaboração entre matemática e ciência da computação se tornou cada vez mais profunda e significativa.

Teorema 3. *Quatro cores são suficientes para colorir as regiões de qualquer mapa sem que duas regiões adjacentes recebam a mesma cor.*

Exemplos de aplicações expostas em Gross e Yellen (2006), Bondy e Murty (1976) e Barabási (2016):

- Cartografia: Em mapas, regiões como estados ou países são coloridas de modo que regiões vizinhas não tenham a mesma cor, facilitando a distinção visual.
- Programação de horários: Em problemas onde diferentes tarefas (representadas por vértices) não podem ocorrer ao mesmo tempo (arestas), a coloração dos vértices pode ajudar na criação de um cronograma eficiente.
- Design de redes: Em redes de comunicação, diferentes frequências (cores) são atribuídas a transmissores próximos para evitar interferência.

O teorema das quatro cores tem muitas implicações e aplicações em teoria dos grafos, cartografia, ciência da computação, e otimização. Ele inspirou a pesquisa em algoritmos de coloração de grafos e em métodos de prova assistida por computador. Explicações mais detalhadas sobre as demonstrações existentes podem ser encontradas nos trabalhos de (Duenha, 2002) e (Souza, 2001).

4 PAGERANK

De acordo com Rogers (2002) e Machado, Rocha, Magalhães e Silva (2016), o PageRank é uma família de algoritmos de análise de rede, um superconjunto de fórmulas matemáticas a fim de determinar a relevância ou importância de páginas da Internet, para que elas possam ser posicionadas (rankeadas) no motor de busca, sendo uma das métricas que o Google utiliza.

O PageRank é uma aplicação de grafos; foi desenvolvido em 1998 por Larry Page e Sergey Brin, fundadores do Google, e patenteado pela Universidade de Stanford, onde seus criadores cursavam doutorado na época.

Para mensurar a popularidade de uma página, o algoritmo leva em conta a **quantidade** e a **qualidade** dos *hyperlinks* que redirecionam a ela. Em outras palavras, o PageRank elege por meio de votos de outras páginas os conteúdos a serem posicionados no buscador. Cada voto pode ter diferentes pesos. Pode-se elencar três princípios básicos de funcionamento do PageRank:

- **Quantidade:** Cada *link* que uma determinada página X recebe de outra página é um **voto de suporte**. Receber *links* (votos) aumenta a autoridade de X, pois significa que outras páginas recomendam X para seus usuários.

- **Qualidade:** Os votos de suporte de *links* vindos de páginas com PageRank alto é mais valioso, dado que elas possuem mais autoridade, desta forma seus *links* possuem maior qualidade.

- **Peso:** Uma página que recomenda vários *links* deve ser levada menos em conta do que páginas que recomendam menos, pois o peso de seu voto de suporte é distribuído entre as páginas que ela recomenda.

Quanto maior o PageRank de uma página, mais ela sobe no *ranking* do algoritmo.

4.1 Modelo simplificado do PageRank

Considerando um conjunto de páginas na internet, cada uma conectada a outras por meio de *links*, as equações definem o PageRank como uma medida de importância de uma página. Inicialmente, cada página recebe um PageRank igualmente distribuído entre todas as páginas. Se houver um total de $|V|$ páginas, então cada página tem um PageRank inicial de $\frac{1}{|V|}$.

A cada iteração, o PageRank de cada página é atualizado com base no PageRank das páginas que têm *links* para ela. Para isso, todas as páginas que têm *links* para a página em questão são consideradas (representadas por $\zeta(X)$). Então, o PageRank de uma página X na iteração t é atualizado somando-se os PageRanks dessas páginas, cada um dividido pelo número de *links* que saem dela. Essa atualização reflete o fato de que uma página com muitos *links* de saída está "diluindo" sua importância ao espalhar seu PageRank por várias outras páginas.

A fórmula utilizada para atualizar o PageRank de uma página X na iteração t é a seguinte:

$$p_t(X) = \sum_{Y \in N^-(X)} \frac{p_{t-1}(Y)}{d^+(Y)} \tag{8}$$

Onde:

- $p_t(X)$ é o PageRank da página X na iteração t.
- $N^-(X)$ é o conjunto de páginas que possuem links para X;
- $\sum_{Y \in N^-(X)}$ representa a soma sobre todas as páginas Y que têm um link para a página X.
- $p_{t-1}(Y)$ é o PageRank da página Y na iteração $t-1$.
- $d^+(Y)$ é o número de links que saem da página Y.

Observações:

(i) Os links que apontam para a própria página são desconsiderados;

(ii) Vários links de uma página para outra são tratados como um único link;

(iii) O PageRank transferido de uma página para outra, em cada iteração, é igualmente dividido entre todos os *links* de saída;

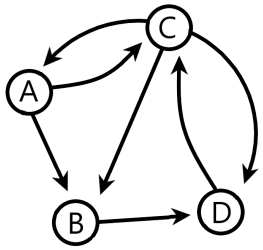
(iv) Pressupõe-se que todas as páginas têm pelo menos um *link*.

(v) Os PageRanks são expressos como uma distribuição de probabilidade, onde a soma total é igual a 1;

(vi) Os valores de PageRank indicam a probabilidade de onde alguém chegaria ao clicar em *links* aleatoriamente.

Exemplo 19. Modelo Simplificado: Vamos apresentar o cálculo do PageRank para os vértices A, B, C e D do grafo representado na Figura (24) de forma simplificada. Vale ressaltar que todos os resultados estão arredondados para três casas decimais, para melhor vizualização.

Figura 24 – Exemplo (19) - Grafo de 4 vértices



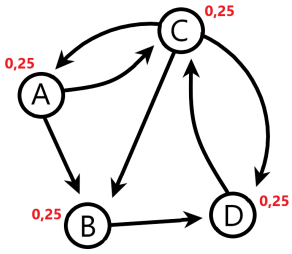
Fonte: Autora (2024).

Inicialização:

O PageRank inicial (p) para cada vértice é calculado como $\frac{1}{|V|}$, onde $|V|$ é o número de vértices. Como neste exemplo $|V| = 4$, então:

$$p(A) = p(B) = p(C) = p(D) = \frac{1}{4} = 0,25$$

Figura 25 – Exemplo (19) - Inicialização



Fonte: Autora (2024).

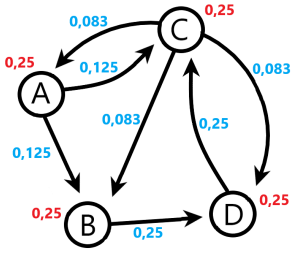
Precisamos redistribuir os PageRanks de acordo com os arcos de saída e entrada de cada vértice. Sendo assim, nos baseamos na matriz de adjacência do grafo:

	A	B	C	D	Total de saídas $d^+(Y)$
A	0	1	1	0	$d^+(A) = 2$
B	0	0	0	1	$d^+(B) = 1$
C	1	1	0	1	$d^+(C) = 3$
D	0	0	1	0	$d^+(D) = 1$

Logo, o peso inicial de cada arco (c) será calculado por:

$$c(A) = \frac{p(A)}{2} = \frac{0,25}{2} = 0,125$$
$$c(B) = \frac{p(B)}{1} = \frac{0,25}{1} = 0,250$$
$$c(C) = \frac{p(C)}{3} = \frac{0,25}{3} = 0,083$$
$$c(D) = \frac{p(D)}{1} = \frac{0,25}{1} = 0,250$$

Figura 26 – Exemplo (19) - Peso inicial dos arcos



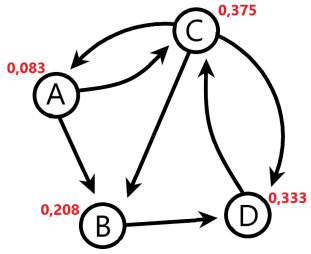
Fonte: Autora (2024).

Assim, o PageRank em $t = 0$ terá o valor do somatório dos valores dos arcos de entrada de cada vértice. O vértice A recebe de C. O vértice B recebe de A e C. O vértice C recebe de A e D. O vértice D recebe de B e C, logo:

Iteração $t = 0$

$$p_0(A) = c(C) = 0,083$$
$$p_0(B) = c(A) + c(C) = 0,125 + 0,083 = 0,208$$
$$p_0(C) = c(A) + c(D) = 0,125 + 0,250 = 0,375$$
$$p_0(D) = c(B) + c(C) = 0,250 + 0,083 = 0,333$$

Figura 27 – Exemplo (19) - Iteração $t = 0$

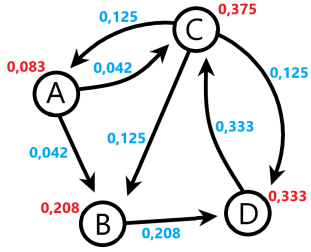


Fonte: Autora (2024).

O algoritmo continua, de modo que

$$c_0(A) = \frac{p_0(A)}{2} = \frac{0,083}{2} = 0,042$$
$$c_0(B) = \frac{p_0(B)}{1} = \frac{0,208}{1} = 0,208$$
$$c_0(C) = \frac{p_0(C)}{3} = \frac{0,375}{3} = 0,125$$
$$c_0(D) = \frac{p_0(D)}{1} = \frac{0,333}{1} = 0,333$$

Figura 28 – Exemplo (19) - peso c_0 dos arcos

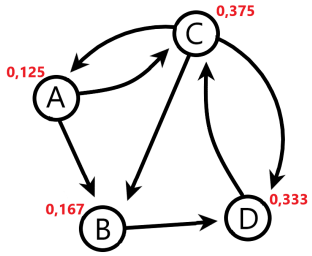


Fonte: Autora (2024).

Iteração $t = 1$:

$$p_1(A) = c_0(C) = 0,125$$
$$p_1(B) = c_0(A) + c_0(C) = 0,042 + 0,125 = 0,167$$
$$p_1(C) = c_0(A) + c_0(D) = 0,042 + 0,333 = 0,375$$
$$p_1(D) = c_0(B) + c_0(C) = 0,208 + 0,125 = 0,333$$

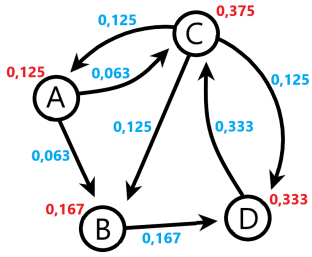
Figura 29 – Exemplo (19) - Iteração $t = 1$



Fonte: Autora (2024).

$$c_1(A) = \frac{p_1(A)}{2} = \frac{0,125}{2} = 0,063$$
$$c_1(B) = \frac{p_1(B)}{1} = \frac{0,167}{1} = 0,167$$
$$c_1(C) = \frac{p_1(C)}{3} = \frac{0,375}{3} = 0,125$$
$$c_1(D) = \frac{p_1(D)}{1} = \frac{0,333}{1} = 0,333$$

Figura 30 – Exemplo (19) - peso c_1 dos arcos

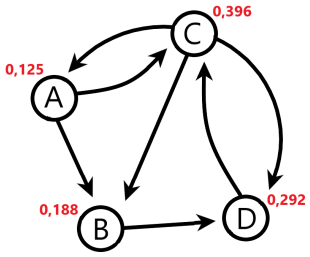


Fonte: Autora (2024).

Iteração $t = 2$:

$$p_2(A) = c_1(C) = 0,125$$
$$p_2(B) = c_1(A) + c_1(C) = 0,063 + 0,125 = 0,188$$
$$p_2(C) = c_1(A) + c_1(D) = 0,063 + 0,333 = 0,396$$
$$p_2(D) = c_1(B) + c_1(C) = 0,167 + 0,125 = 0,292$$

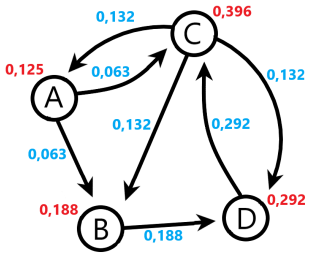
Figura 31 – Exemplo (19) - Iteração $t = 2$



Fonte: Autora (2024).

$$c_2(A) = \frac{p_2(A)}{2} = \frac{0,125}{2} = 0,063$$
$$c_2(B) = \frac{p_2(B)}{1} = \frac{0,188}{1} = 0,167$$
$$c_2(C) = \frac{p_2(C)}{3} = \frac{0,396}{3} = 0,125$$
$$c_2(D) = \frac{p_2(D)}{1} = \frac{0,292}{1} = 0,333$$

Figura 32 – Exemplo (19) - peso c_2 dos arcos

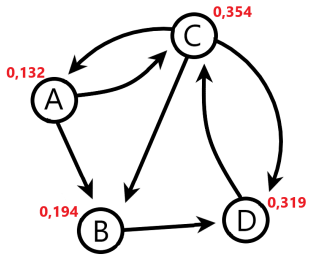


Fonte: Autora (2024).

Iteração $t = 3$:

$$p_3(A) = c_2(C) = 0,132$$
$$p_3(B) = c_2(A) + c_2(C) = 0,063 + 0,132 = 0,194$$
$$p_3(C) = c_2(A) + c_2(D) = 0,063 + 0,292 = 0,354$$
$$p_3(D) = c_2(B) + c_2(C) = 0,188 + 0,132 = 0,319$$

Figura 33 – Exemplo (19) - Iteração $t = 3$



Fonte: Autora (2024).

Observação: O algoritmo continua iterando até que os valores do PageRank convirjam para um valor estável.

Neste caso, os valores de p no algoritmo se estabilizam em $t = 80$, obtendo o PageRank final:

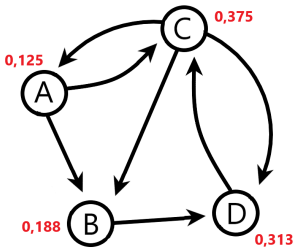
$$p_{80}(A) = 0,125$$

$$p_{80}(B) = 0,188$$

$$p_{80}(C) = 0,375$$

$$p_{80}(D) = 0,313$$

Figura 34 – Exemplo (19) - PageRank final



Fonte: Autora (2024).

Portanto, concluímos que a página C do Exemplo (19) possui o maior PageRank e consequentemente aparecerá primeiro nos resultados de pesquisa. Deste modo:

- 1º lugar: página C
- 2º lugar: página D
- 3º lugar: página B

- 4º lugar: página *A*

4.1.1 Obtendo as iterações através de linguagem *Python*

Importação e Definição do Grafo

```
import networkx as nx

G = nx.DiGraph()
G.add_nodes_from(["A", "B", "C", "D"])
G.add_edges_from([("A", "B"), ("A", "C"), ("B", "D"), ("C", "A"), ("C", "B"), ("C", "D"), ("D", "C")])
```

Importação da biblioteca: O código começa importando a biblioteca *networkx*, que é usada para a criação, manipulação e estudo da estrutura, dinâmica e funções de grafos complexos. Para abreviar, colocamos o apelido desta biblioteca como *nx*.

Definição do grafo dirigido (*DiGraph*): É criado um grafo dirigido *G* e adicionados os nós e as arestas. Cada aresta direcionada representa um *link* de uma página (nó) para outra página.

Contagem de Vértices

```
n = len(G.nodes())
print("Número de vértices (n) =", n)
```

A variável *n* armazena o número total de nós no grafo, que são 4 no exemplo fornecido: *A*, *B*, *C* e *D*.

Algoritmo PageRank Simplificado

```
def pagerank_simples(G, iteracoes=100):
    pagerank = {node: 1/n for node in G}

    for t in range(iteracoes):
        prox_pagerank = {}
        for node in G:
            prox_pagerank[node] = 0  # Inicializa o PageRank do nó
            for vizinho in G.predecessors(node):
                prox_pagerank[node] += pagerank[vizinho] / G.out_degree(vizinho)
        pagerank = prox_pagerank
```

```
18
19          # Imprime o PageRank para t em um intervalo específico (0 a 9 neste
        caso)
20          if t in range(9):  # Ajustado para incluir 0 a 9
21              print(f"t={t}: {pagerank}")
22
23      return pagerank
```

Inicialização do PageRank: Cria-se um dicionário pagerank onde cada nó começa com um valor inicial de $1/n$, ou seja, todos os nós começam com o mesmo valor de PageRank.

Iterações do PageRank: Para um número definido de iterações (iteracoes=100), o algoritmo atualiza os valores de PageRank.

Cálculo do próximo PageRank: Cria-se um dicionário temporário *prox_pagerank* para armazenar os novos valores de PageRank calculados nesta iteração.

Para cada nó no grafo, inicializa-se o valor de PageRank em 0.

Para cada nó, o algoritmo soma a contribuição dos nós predecessores (nós que têm arestas direcionadas para o nó atual). A contribuição de um predecessor é o valor de seu PageRank atual dividido pelo seu grau de saída (número de *links* que ele aponta para outros nós).

Atualiza-se o dicionário pagerank com os valores de *prox_pagerank* após cada iteração.

Impressão do PageRank: O código imprime os valores de PageRank a cada iteração de $t = 0$ a $t = 9$.

Exibição dos resultados

```
24 resultados = pagerank_simples(G)
25 print("PageRank final:", resultados)
```

Execução da Função: A função *pagerank_simples* é chamada com o grafo G e o resultado final (os valores de PageRank após 100 iterações) é armazenado na variável *resultados*.

Exibição do Resultado Final: O PageRank final é impresso.

Figura 35 – Resultados até $t = 9$ impressos em *python*

```
Número de vértices (n) = 4
t=0: {'A': 0.08333333333333333, 'B': 0.20833333333333331, 'C': 0.375, 'D': 0.3333333333333333}
t=1: {'A': 0.125, 'B': 0.16666666666666666, 'C': 0.375, 'D': 0.3333333333333333}
t=2: {'A': 0.125, 'B': 0.1875, 'C': 0.3958333333333333, 'D': 0.2916666666666663}
t=3: {'A': 0.13194444444444445, 'B': 0.19444444444444445, 'C': 0.3541666666666663, 'D': 0.3194444444444444}
t=4: {'A': 0.11805555555555554, 'B': 0.18402777777777776, 'C': 0.38541666666666663, 'D': 0.3125}
t=5: {'A': 0.1284722222222222, 'B': 0.18749999999999997, 'C': 0.3715277777777778, 'D': 0.3125}
t=6: {'A': 0.1238425925925926, 'B': 0.18807870370370372, 'C': 0.3767361111111111, 'D': 0.31134259259259256}
t=7: {'A': 0.1255787037037037, 'B': 0.1875, 'C': 0.37326388888888884, 'D': 0.31365740740740744}
t=8: {'A': 0.12442129629629628, 'B': 0.18721064814814814, 'C': 0.3764467592592593, 'D': 0.3119212962962963}
t=9: {'A': 0.12548225308641978, 'B': 0.18769290123456792, 'C': 0.3741319444444444, 'D': 0.31269290123456794}
PageRank final: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
```

Fonte: Autora (2024).

Caso prefira, pode-se ajustar a quantidade de iterações e quais serão exibidas apenas alterando os números na parte destinada do código. Por exemplo, se ajustarmos a exibição até a iteração 99 podemos observar que o algoritmo converge em $t = 80$, conforme a Figura (36).

Figura 36 – Resultados até $t = 99$ impressos em *python*

```
t=77: {'A': 0.12500000000000008, 'B': 0.1875000000000001, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=78: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.37500000000000017, 'D': 0.3125000000000001}
t=79: {'A': 0.12500000000000006, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=80: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=81: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=82: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=83: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=84: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=85: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=86: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=87: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=88: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=89: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=90: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=91: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=92: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=93: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=94: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=95: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=96: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=97: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=98: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
t=99: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
PageRank final: {'A': 0.12500000000000003, 'B': 0.18750000000000006, 'C': 0.3750000000000001, 'D': 0.3125000000000001}
```

Fonte: Autora (2024).

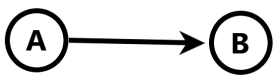
4.2 Desafios do Modelo Simplificado

No contexto do algoritmo PageRank, o modelo simplificado pode encontrar algumas dificuldades que afetam a precisão dos resultados:

Páginas sem *links* de saída: Quando uma página não possui nenhum *link* de saída, o PageRank atribuído a ela pode se tornar zero. Isso ocorre porque o algoritmo distribui o PageRank de uma página entre os *links* de saída para outras páginas. Assim, se não houver nenhum *link* para transferir o PageRank, a página é deixada com um valor zero.

Exemplo 20.

Figura 37 – Exemplo (20) - Página sem *links* de saída



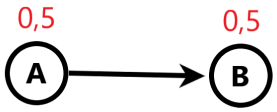
Fonte: Autora (2024).

Inicialização:

O PageRank inicial (p) para cada vértice é calculado como $\frac{1}{|V|}$, onde $|V|$ é o número de vértices. Como neste exemplo $|V| = 2$, então:

$$p(A) = p(B) = \frac{1}{2} = 0,5$$

Figura 38 – Exemplo (20) - Inicialização



Fonte: Autora (2024).

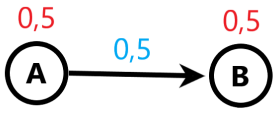
Precisamos redistribuir os PageRanks de acordo com os arcos de saída e entrada de cada vértice. Sendo assim, nos baseamos na matriz de adjacência do grafo:

	A	B	Total de saídas $d^+(Y)$
A	0	1	$d^+(A) = 1$
B	0	0	$d^+(B) = 0$

Logo, o peso inicial do arco A será calculado por:

$$c(A) = \frac{p(A)}{1} = \frac{0,5}{1} = 0,5$$

Figura 39 – Exemplo (20) - Peso inicial do arco A



Fonte: Autora (2024).

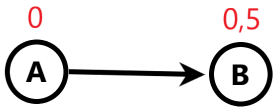
Assim, o PageRank em $t = 0$ terá o valor do somatório dos valores dos arcos de entrada de cada vértice. O vértice A não possui arco de entrada. O vértice B recebe de A, logo:

Iteração $t = 0$

$$p_0(A) = 0$$

$$p_0(B) = c(A) = 0,5$$

Figura 40 – Exemplo (20) - Iteração $t = 0$

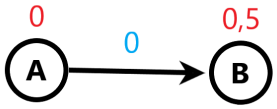


Fonte: Autora (2024).

O algoritmo continua, de modo que

$$c_0(A) = \frac{p_0(A)}{1} = \frac{0}{1} = 0$$

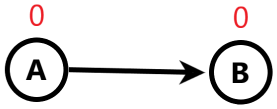
Figura 41 – Exemplo (20) - peso c_0 do arco



Fonte: Autora (2024).

A partir desta iteração, o pagerank de A e de B param de variar, pois como A não recebe valores de nenhum arco, seu PageRank sempre será 0; e como B recebe PageRank de A, então B sempre receberá PageRank 0. Assim, vemos que no modelo simplificado não é possível fazer ranking entre duas páginas sem que as duas possuam links de saída.

Figura 42 – Exemplo (20) - Pagerank final



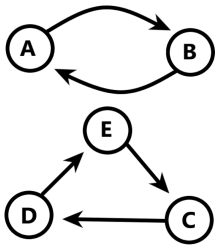
Fonte: Autora (2024).

	t inicial	$t = 0$	$t = 1$
A	$0,5$	0	0
B	$0,5$	$0,5$	0

Criação de Ciclos: Páginas que estão interligadas formando um ciclo podem criar um *loop* no cálculo do PageRank. Isso pode resultar em uma distribuição de PageRank que nunca converge para um estado estável, pois o fluxo de PageRank pode circular continuamente entre as páginas do ciclo, sem chegar a um equilíbrio final.

Exemplo 21.

Figura 43 – Páginas formando ciclo



Fonte: Autora (2024).

Podemos conferir o PageRank através de *python*:

```
import networkx as nx

# Define o grafo
G = nx.DiGraph()
G.add_nodes_from(["A", "B", "C", "D", "E"])
G.add_edges_from([("A", "B"), ("B", "A"), ("D", "E"), ("E", "C"), ("C", "D")])

n = len(G.nodes())
print("Número de vértices (n) =", n)

def pagerank_simples(G, iteracoes=100):
    pagerank = {node: 1/n for node in G}

    for t in range(iteracoes):
        prox_pagerank = {}
        for node in G:
            prox_pagerank[node] = 0  # Inicializa o PageRank do nó
            for vizinho in G.predecessors(node):
                prox_pagerank[node] += pagerank[vizinho] / G.out_degree(vizinho)
        pagerank = prox_pagerank
```

```
21
22          # Imprime o PageRank para t em um intervalo específico (0 a 10
        neste caso)
23          if t in range(10):  # Ajustado para incluir 0 a 10
24              print(f"t={t}: {pagerank}")
25
26      return pagerank
27
28  # Calcula e exibe o PageRank
29  resultados = pagerank_simples(G)
30  print("PageRank final:", resultados)
```

Que imprime os seguintes resultados:

Figura 44 – Exemplo (21) - Pagerank dos ciclos de páginas

```
Número de vértices (n) = 5
t=0: {'A': 0.2, 'B': 0.2, 'C': 0.2, 'D': 0.2, 'E': 0.2}
t=1: {'A': 0.2, 'B': 0.2, 'C': 0.2, 'D': 0.2, 'E': 0.2}
t=2: {'A': 0.2, 'B': 0.2, 'C': 0.2, 'D': 0.2, 'E': 0.2}
t=3: {'A': 0.2, 'B': 0.2, 'C': 0.2, 'D': 0.2, 'E': 0.2}
t=4: {'A': 0.2, 'B': 0.2, 'C': 0.2, 'D': 0.2, 'E': 0.2}
t=5: {'A': 0.2, 'B': 0.2, 'C': 0.2, 'D': 0.2, 'E': 0.2}
t=6: {'A': 0.2, 'B': 0.2, 'C': 0.2, 'D': 0.2, 'E': 0.2}
t=7: {'A': 0.2, 'B': 0.2, 'C': 0.2, 'D': 0.2, 'E': 0.2}
t=8: {'A': 0.2, 'B': 0.2, 'C': 0.2, 'D': 0.2, 'E': 0.2}
t=9: {'A': 0.2, 'B': 0.2, 'C': 0.2, 'D': 0.2, 'E': 0.2}
PageRank final: {'A': 0.2, 'B': 0.2, 'C': 0.2, 'D': 0.2, 'E': 0.2}
```

Fonte: Autora (2024).

Assim, vemos que páginas que criam ciclos possuem o mesmo PageRank, pois recebem *links* tanto quanto enviam, de maneira igualitária.

4.3 Modelo do PageRank com Fator de Amortecimento

- **Fator de Amortecimento (Dumping Factor)**

O fator de amortecimento, também conhecido como *dumping factor*, é uma adição crucial ao algoritmo PageRank para lidar com as limitações do modelo simplificado. Ele controla a probabilidade de um usuário continuar seguindo os *links* de uma página, em vez de pular para uma página aleatória.

A fórmula do PageRank com o fator de amortecimento é dada por:

$$p_t(X) = \frac{1-d}{|V|} + d \cdot \left(\sum_Y \frac{p_{t-1}(Y)}{d^+(Y)} \right) \tag{9}$$

Onde:

- $p_t(X)$ é o PageRank da página X na iteração t.
- d representa o fator de amortecimento, que varia entre 0 e 1. Ele indica a probabilidade de um usuário continuar navegando pelos links em vez de "saltar" para uma página aleatória.
- $|V|$ é o número total de páginas no grafo.
- $\sum_Y$ representa a soma sobre todas as páginas Y que têm um link para a página X.
- $p_{t-1}(Y)$ é o PageRank da página Y na iteração $t-1$.
- $d^+(Y)$ é o número de links que saem da página Y.

Em geral, o fator de amortecimento é configurado para um valor $d = 0,85$. Isso significa que há uma probabilidade de 85% de um usuário continuar seguindo os links e uma probabilidade de 15% de pular para uma página aleatória. Com essa configuração, todas as páginas recebem uma chance igual de serem visitadas ao acaso.

As iterações acabam quando o grafo converge (os PageRanks param de variar). De acordo com Lay (2012), são necessárias entre 50 e 100 iterações do algoritmo para obter o PageRank com a precisão que o Google usa para suas classificações.

O Google ainda utiliza o PageRank como um componente em seu algoritmo de classificação, mas ele não é mais o único fator determinante. Ao longo dos anos, a empresa aprimorou seu sistema de busca, incorporando diversos outros elementos além do PageRank, como aprendizado de máquina, o conteúdo da página, a qualidade dos *backlinks*, e até o comportamento dos usuários (Inqnest, 2024).

Exemplo 22. Modelo com Fator de Amortecimento. Faremos o exemplo de PageRank da Figura (37), desta vez com o fator de amortecimento.

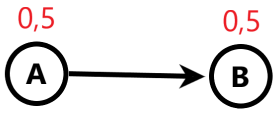
Inicialização:

A inicialização se dá da mesma forma que os exemplos anteriores.

O PageRank inicial (p) para cada vértice é calculado como $\frac{1}{|V|}$, onde $|V|$ é o número de vértices. Como neste exemplo $|V| = 2$, então:

$$p(A) = p(B) = \frac{1}{2} = 0,5$$

Figura 45 – Exemplo (22) amortizado - Inicialização



Fonte: Autora (2024).

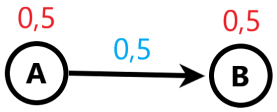
Precisamos redistribuir os PageRanks de acordo com os arcos de saída e entrada de cada vértice. Sendo assim, nos baseamos na matriz de adjacência do grafo:

	A	B	Total de saídas $d^+(Y)$
A	0	1	$d^+(A) = 1$
B	0	0	$d^+(B) = 0$

Logo, o peso inicial do arco A será calculado por:

$$c(A) = \frac{p(A)}{1} = \frac{0,5}{1} = 0,5$$

Figura 46 – Exemplo (22) amortizado - Peso inicial do arco A



Fonte: Autora (2024).

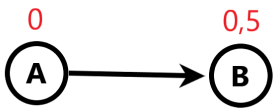
Assim, o PageRank em $t = 0$ terá o valor do somatório dos valores dos arcos de entrada de cada vértice. O vértice A não possui arco de entrada. O vértice B recebe de A, logo:

Iteração $t = 0$

$$p_0(A) = 0$$

$$p_0(B) = c(A) = 0,5$$

Figura 47 – Exemplo (22) amortizado - Iteração $t = 0$



Fonte: Autora (2024).

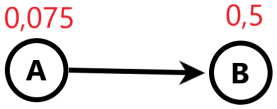
A diferença para o cálculo amortizado é que após obtermos os valores de PageRank simplificado dos nós, precisamos amortizá-los, ou seja, multiplicar o valor de cada PageRank por $0,85$ e adicionar a $\frac{0,15}{|V|}$. Neste caso, $\frac{0,15}{|V|} = \frac{0,15}{2} = 0,075$. Logo:

Iteração $t = 0$ amortizada

$$p_0(A) = 0 \cdot 0,85 + 0,075 = 0,075$$

$$p_0(B) = c(A) = 0,5 \cdot 0,85 + 0,075 = 0,5$$

Figura 48 – Exemplo (22) amortizado - Iteração $t = 0$ amortizada



Fonte: Autora (2024).

O algoritmo continua, de modo que

$$c_0(A) = \frac{p_0(A)}{1} = \frac{0,075}{1} = 0,075$$

Figura 49 – Exemplo (22) amortizado - peso c_0 do arco

Fonte: Autora (2024).

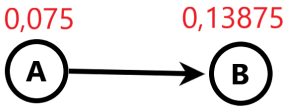
Iteração $t = 1$ amortizada

$$p_1(A) = p_0(A) = 0,075$$

$$p_1(B) = c_0(A) = 0,075$$

Amortização: $p_1(B) = 0,075 \cdot 0,85 + 0,075 = 0,13875$

Figura 50 – Exemplo (22) amortizado - Iteração $t = 1$ amortizada



Fonte: Autora (2024).

A partir desta iteração, os pageranks de A e de B param de variar, pois como A não recebe valores de nenhum arco, seu PageRank sempre será $0,075$; e como B recebe PageRank de A, então B sempre receberá PageRank $0,13875$. Assim vemos que no modelo completo, com o fator de amortecimento, o problema de zerar PageRanks é resolvido.

Desta maneira, **B aparece em 1º lugar e A aparece em 2º lugar.**

Podemos conferir o resultado com *python*:

```
   import networkx as nx

# Define o grafo
G = nx.DiGraph()
G.add_nodes_from(["A", "B"])
G.add_edges_from([("A", "B")])

n = len(G.nodes())
print("Número de vértices (n) =", n)
d=0.85

def pagerank_(G, iteracoes=100):
    pagerank = {node: 1/n for node in G}  # Inicializa o PageRank

    for t in range(iteracoes):
        prox_pagerank = {node: (1 - d)/n for node in G}  # Parte do fator
    de amortecimento

        for node in G:
            for vizinho in G.predecessors(node):
                prox_pagerank[node] += d*(pagerank[vizinho] / G.out_degree(
    vizinho))

        pagerank = prox_pagerank
```

```
23
24          # Imprime o PageRank para t em um intervalo específico (0 a 10
        neste caso)
25          if t in range(10):  # Ajustado para incluir 0 a 10
26              print(f"t={t}: {pagerank}")
27
28      return pagerank
29
30 # Calcula e exibe o PageRank
31 resultados = pagerank_(G)  # Define explicitamente o valor de d
32 print("PageRank final:", resultados)
```

Explicação das mudanças:

Inicialização do prox_pagerank: Inicializa prox_pagerank com $(1-d)/n$ para cada nó, que é a contribuição do fator de amortecimento.

Atualização do PageRank: Para cada nó, adiciona o valor do PageRank vindo dos nós predecessores ajustado pelo fator de amortecimento d. Este ajuste garante que parte do valor do PageRank (15% no caso de $d = 0.85$) é distribuído igualmente entre todos os nós, enquanto os 85% restantes são distribuídos de acordo com os *links* de entrada, simulando a navegação de um usuário da web.

Este código produz o seguinte resultado:

Figura 51 – Pagerank do Exemplo (22) amortizado

```
Número de vértices (n) = 2
t=0: {'A': 0.07500000000000001, 'B': 0.5}
t=1: {'A': 0.07500000000000001, 'B': 0.13875}
t=2: {'A': 0.07500000000000001, 'B': 0.13875}
t=3: {'A': 0.07500000000000001, 'B': 0.13875}
t=4: {'A': 0.07500000000000001, 'B': 0.13875}
t=5: {'A': 0.07500000000000001, 'B': 0.13875}
t=6: {'A': 0.07500000000000001, 'B': 0.13875}
t=7: {'A': 0.07500000000000001, 'B': 0.13875}
t=8: {'A': 0.07500000000000001, 'B': 0.13875}
t=9: {'A': 0.07500000000000001, 'B': 0.13875}
PageRank final: {'A': 0.07500000000000001, 'B': 0.13875}
```

Fonte: Autora (2024).

Deste modo, percebemos que a página B possui o maior PageRank, o que inicialmente seria trivial, visto que ela é a única página da rede que está recebendo algum *link*.

4.4 Código em *python* para Cálculo do PageRank e Visualização com Imagens

Nesta seção, faremos as imagens das iterações do Pagerank do Exemplo (19), desta vez amortizado, utilizando linguagem pyhton.

Importações

```
import networkx as nx
import matplotlib.pyplot as plt
```

Essas linhas importam as bibliotecas *networkx* para manipulação de grafos e *matplotlib.pyplot* para visualização.

Definição do Grafo

```
G = nx.DiGraph()
G.add_nodes_from(["A", "B", "C", "D"])
G.add_edges_from([("A", "B"), ("A", "C"), ("B", "D"), ("C", "A"), ("C", "B"
    ), ("C", "D"), ("D", "C")])
```

- Cria um grafo dirigido (DiGraph).
- Adiciona os nós A, B, C, e D ao grafo.
- Adiciona as arestas dirigidas ao grafo.

Contagem de Vértices e Fator de Amortecimento

```
n = len(G.nodes())
print("Número de vértices (n) =", n)

d = 0.85  # Fator de amortecimento
```

- Calcula o número de nós no grafo.
- d = 0.85: Define o fator de amortecimento para o algoritmo PageRank.

Função para Desenhar o Grafo

Esta função desenha o grafo, mostrando os rótulos dos nós, os valores do PageRank e os pesos das arestas.

```
10 def desenhar_grafo(G, pagerank, edge_weights, pos, iteracao):
11         labels = {node: node for node in G}  # Rótulos com os nomes dos nós
12 
13     # Identificar arestas bidirecionadas
14     bidirecionadas = [(u, v) for u, v in G.edges() if (v, u) in G.edges()]
15 
16     # Separar arestas curvadas e arcos retos
17     arc_edges = bidirecionadas
18     reto_edges = [e for e in G.edges() if e not in bidirecionadas]
```

- Cria um dicionário de rótulos para os nós.
- Identifica as arestas bidirecionadas (arestas que existem em ambas as direções).
- Separa as arestas curvadas e retas.

Desenho do Grafo

```
19     plt.figure(figsize=(6, 6))  # Tamanho da figura
20 
21     # Desenha o grafo com arcos retos
22     nx.draw(G, pos, with_labels=True, labels=labels, node_size=1000,
    node_color="pink", edgelist=reto_edges, font_weight="bold", edge_color="
    black", width=2, arrowsize=20, font_size=15)
```

Configura a figura e desenha o grafo com arestas retas. Os ajustes de tamanho do nó, cor do nó, tamanho da fonte, cor da fonte, espessura da aresta, tamanho da seta e etc podem ser realizados de acordo com preferências pessoais.

Adicionar Rótulos de PageRank

```
23         pagerank_labels = {node: f"{round(pagerank[node], 3)}" for node in
    G}
24     for node, (x, y) in pos.items():
25         plt.text(x + 0.08, y + 0.06, s=pagerank_labels[node], bbox=dict(
    facecolor='white', alpha=0.7),
26                  horizontalalignment='center', fontsize=15, color="black",
    fontweight="bold")
```

Adiciona os rótulos dos valores de PageRank fora dos nós.

Adicionar Arestas Curvadas e Seus Rótulos

```
27          # Adiciona os arcos curvos para arestas bidirecionadas com uma seta
28      for (u, v) in bidirecionadas:
29          # Desenha apenas uma direção para bidirecionadas
30          nx.draw_networkx_edges(G, pos, edgelist=[(u, v)], connectionstyle='
        arc3, rad=0.15', width=2, arrowsize=20, node_size=1000)
31
32      # Adiciona os rótulos dos arcos curvos
33      edge_labels_curvos = {(u, v): f"{round(w, 3)}" for (u, v), w in
        edge_weights.items() if (u, v) in bidirecionadas}
34      nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels_curvos,
        font_color='red', font_size=15, connectionstyle='arc3, rad=0.15')
```

Adicionar Rótulos das Arestas Retas

```
35          # Adiciona os rótulos dos arcos retos
36      edge_labels_retos = {(u, v): f"{round(w, 3)}" for (u, v), w in
        edge_weights.items() if (u, v) not in bidirecionadas}
37      nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels_retos,
        font_color='red', font_size=15, label_pos=0.5)  # Ajuste de posição para
         melhor visualização
```

Título da Figura e Exibição

```
38          if iteracao == 100:
39          titulo = "Iteração Final"
40      else:
41          titulo = f"Iteração {iteracao}"
42      plt.title(titulo, y=1.05)  # Título da figura
43
44      if isinstance(iteracao, int) and (iteracao <= 11 or iteracao == 100):
45          plt.show()  # Exibe a figura
46      else:
47          plt.close()  # Fecha a figura sem exibir
```

Define o título da figura e decide se a figura deve ser exibida ou não com base na iteração.

Algoritmo PageRank Amortizado com Imagens

```
48      def pagerank_amortizado(G, iteracoes=11):
49      pagerank = {node: 1/n for node in G}
50      pos = nx.planar_layout(G)  # Posição dos nós
```

```
51
52      # Desenha o PageRank inicial
53      edge_weights = {(u, v): 0 for u, v in G.edges()}
54      desenhar_grafo(G, pagerank, edge_weights, pos, "Inicial")
55
56      for t in range(iteracoes):
57          prox_pagerank = {node: (1 - d) / n for node in G}  # Inicializa o
    PageRank com o fator de amortecimento
58          edge_weights = {}
59          for node in G:
60              for vizinho in G.predecessors(node):
61                  prox_pagerank[node] += d * (pagerank[vizinho] / G.
    out_degree(vizinho))
62          for node in G:
63              if G.out_degree(node) > 0:
64                  for vizinho in G.successors(node):
65                      edge_weights[(node, vizinho)] = pagerank[node] / G.
    out_degree(node)
66          pagerank = prox_pagerank
67
68          # Desenha o grafo para a iteração atual
69          desenhar_grafo(G, pagerank, edge_weights, pos, t)
70
71      # Desenha o PageRank final
72      for _ in range(100 - iteracoes):  # Atualiza para iterar até 100
73          prox_pagerank = {node: (1 - d) / n for node in G}  # Inicializa o
    PageRank com o fator de amortecimento
74          edge_weights = {}
75          for node in G:
76              for vizinho in G.predecessors(node):
77                  prox_pagerank[node] += d * (pagerank[vizinho] / G.
    out_degree(vizinho))
78          for node in G:
79              if G.out_degree(node) > 0:
80                  for vizinho in G.successors(node):
81                      edge_weights[(node, vizinho)] = pagerank[node] / G.
    out_degree(node)
82          pagerank = prox_pagerank
83
```

```
84      # Desenha o PageRank final
85      desenhar_grafo(G, pagerank, edge_weights, pos, 100)
86
87      return pagerank
```

Esta função implementa o algoritmo PageRank com amortecimento e desenha o grafo a cada iteração importante. Foi configurado para exibir as iterações de $t = 0$ a $t = 10$, além da iteração final $t = 100$.

- Inicializa o PageRank de cada nó com valor $1/n$.
- Define a posição dos nós.
- Desenha o estado inicial do grafo.
- Executa as iterações do PageRank.
- Calcula o PageRank para cada nó na iteração atual.
- Desenha o grafo para cada iteração.

Após as iterações configuradas, itera até 100 para calcular o PageRank final e desenha o grafo na iteração final. A posição dos nós foi definida como ***pos = nx.planar_layout(G)*** , mas pode ser alterada de acordo com a posição que gere mais conforto visual em cada grafo inserido.

Opções de posição no lugar de "planar":

- **planar:** Tenta posicionar os nós de forma que as arestas não se cruzem. É adequado apenas para grafos planares.
- **circular:** Posiciona os nós em um círculo, de forma que todos os nós fiquem igualmente espaçados.
- **random:** Posiciona os nós aleatoriamente em um espaço bidimensional. Esse layout pode ser útil para ver uma distribuição geral do grafo, mas não é adequado para a visualização detalhada.
- **spectral**: Posiciona os nós usando os autovalores e autovetores da matriz Laplaciana do grafo. Esse layout pode ajudar a revelar a estrutura interna do grafo, como *clusters* ou grupos de nós.

- **spring:** posiciona os nós como se as arestas fossem molas que se repulsam e atraem. Ele tenta minimizar a energia total do sistema, resultando em uma distribuição geralmente agradável e equilibrada dos nós.
- **shell:** Posiciona os nós em camadas concêntricas, ou "conchas". Pode ser útil para grafos hierárquicos ou para destacar nós centrais e periféricos.

Função para Exibir o Ranking

```
88     def exibir_ranking(pagerank):
89     ranking = sorted(pagerank.items(), key=lambda item: item[1], reverse=
       True)
90     print("Ranking das páginas com base no PageRank:")
91     for i, (node, rank) in enumerate(ranking, start=1):
92         print(f"{i}º lugar: Página {node} com PageRank {rank:.3f}")
```

Esta função classifica e exibe o ranking dos nós com base no valor do PageRank.

- Classifica os nós pelo valor do PageRank em ordem decrescente.
- Retorna uma lista de tuplas (nó, valor do PageRank).
- item[1]: Especifica que a classificação deve ser feita com base no valor do PageRank.
- Ordena a lista em ordem decrescente.
- Itera sobre a lista ordenada de nós e valores de PageRank.
- Fornece o índice i começando em 1.
- Exibe a posição do nó no ranking, o identificador do nó e o valor do PageRank arredondado para três casas decimais.

Parte Final: Execução do Algoritmo PageRank e Exibição do Ranking

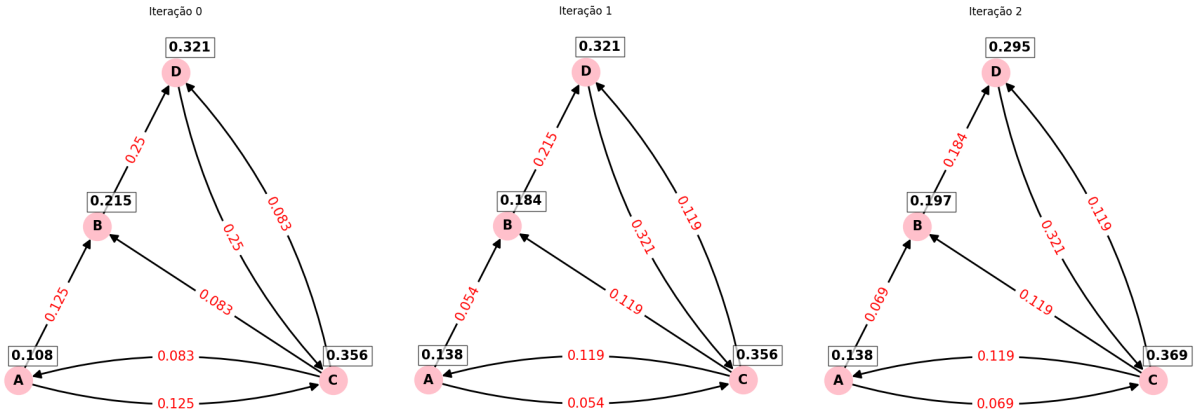
```
93 # Calcula e exibe o PageRank
94 resultados = pagerank_amortizado(G)
95 print("PageRank final:", resultados)
96
97 # Exibe o ranking dos nós
98 exibir_ranking(resultados)
```

Esta parte do código executa o algoritmo PageRank e exibe o ranking final dos nós.

Resultados

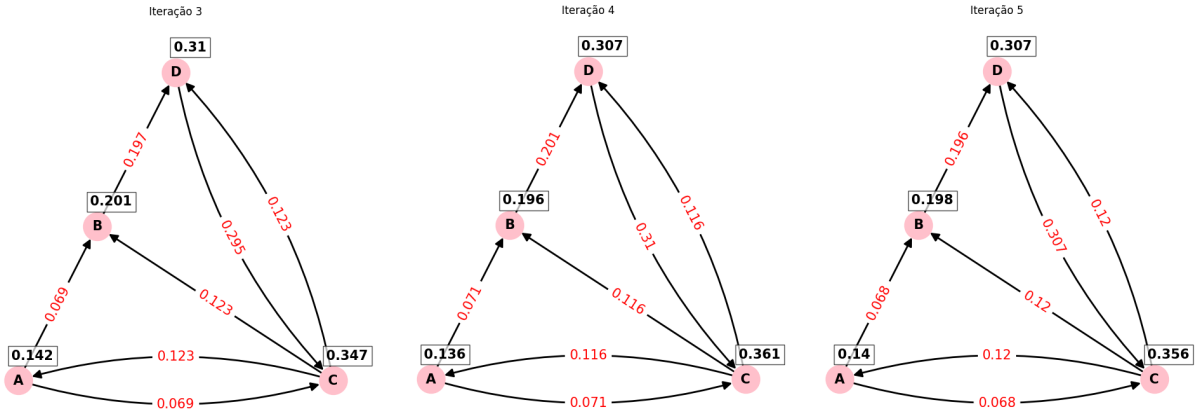
Seguem abaixo as imagens geradas a partir do código e também o *ranking* após as iterações:

Figura 52 – Imagens geradas em *python*: iterações 0 a 2



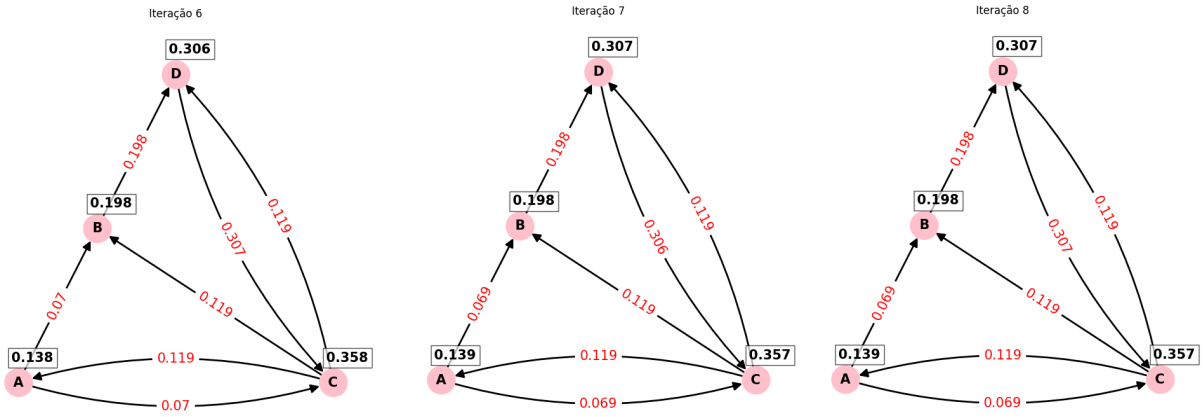
Fonte: Autora (2024).

Figura 53 – Imagens geradas em *python*: iterações 3 a 5



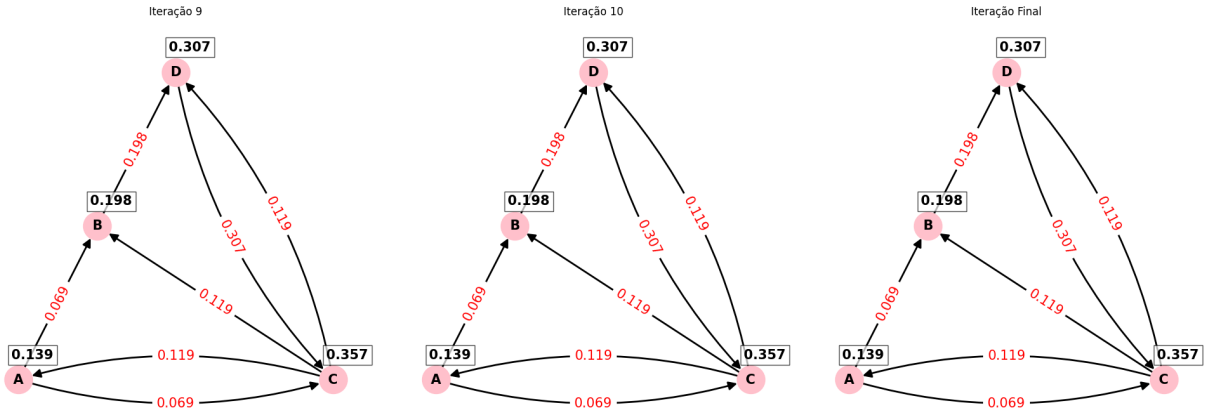
Fonte: Autora (2024).

Figura 54 – Imagens geradas em *python*: iterações 6 a 8



Fonte: Autora (2024).

Figura 55 – Imagens geradas em *python*: iterações 9 , 10 e final



Fonte: Autora (2024).

Figura 56 – Ranking gerado em *python*

```
PageRank final: {'A': 0.13867252573095729, 'B': 0.19760834916661413, 'C': 0.3570795025798492, 'D': 0.30663962252257926}
Ranking das páginas com base no PageRank:
1º lugar: Página C com PageRank 0.357
2º lugar: Página D com PageRank 0.307
3º lugar: Página B com PageRank 0.198
4º lugar: Página A com PageRank 0.139
```

Fonte: Autora (2024).

Verificamos que este ranking é o mesmo do mostrado no Exemplo (19), com o modelo simplificado. Não houve alterações no resultado pois este grafo não possui ciclos ou páginas sem *links* de saída. Apenas os valores de cálculo mudaram, pois foi adicionado o fator de amortecimento.

Mais exemplos de códigos para gerar imagens de grafos em *python* podem ser visualizados no Apêndice (A).

5 ATIVIDADES PARA INTRODUÇÃO DO ENSINO DE GRAFOS

As atividades propostas neste capítulo foram desenvolvidas com o intuito de introduzirem os estudantes à teoria dos grafos. Desta forma, recomendamos que o professor deixe os alunos tentarem resolver por si próprios, sem conhecimento prévio da teoria. Acredita-se que aparecerão resoluções corretas de diferentes maneiras e isso é importante para o desenvolvimento do raciocínio lógico dos alunos, bem como socialização das resoluções encontradas.

A cada explicação posterior da atividade desenvolvida, o professor deve introduzir novos conceitos relacionados à teoria, como definição, vocabulário e representação específicos, que irão auxiliar a resolver os problemas de maneira mais eficaz. Cada atividade possui um nível de dificuldade diferente e cabe ao professor analisar a viabilidade de aplicabilidade das questões em cada turma, de acordo com o interesse, pré-requisitos lógicos, capacidade do grupo e flexibilidade do currículo na escola. Os níveis de dificuldade propostos variam entre habilidades do Ensino Fundamental, Médio e Superior.

Veja na Tabela (1) as habilidades e competências da BNCC que podem ser desenvolvidas nas atividades propostas:

Tabela 1 – Habilidades e competências da BNCC relacionadas às atividades propostas

Competências Gerais	
CG1	Valorizar e utilizar os conhecimentos historicamente construídos sobre o mundo físico, social, cultural e digital para entender e explicar a realidade, continuar aprendendo e colaborar para a construção de uma sociedade justa, democrática e inclusiva.
CG2	Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas.

CG4	Utilizar diferentes linguagens verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo.
CG5	Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.
Competências Específicas do Ensino Fundamental	
CEF2	Desenvolver o raciocínio lógico, o espírito de investigação e a capacidade de produzir argumentos convincentes, recorrendo aos conhecimentos matemáticos para compreender e atuar no mundo.
CEF3	Compreender as relações entre conceitos e procedimentos dos diferentes campos da Matemática (Aritmética, Álgebra, Geometria, Estatística e Probabilidade) e de outras áreas do conhecimento, sentindo segurança quanto à própria capacidade de construir e aplicar conhecimentos matemáticos, desenvolvendo a autoestima e a perseverança na busca de soluções.
CEF5	Utilizar processos e ferramentas matemáticas, inclusive tecnologias digitais disponíveis, para modelar e resolver problemas cotidianos, sociais e de outras áreas de conhecimento, validando estratégias e resultados.
CEF6	Enfrentar situações-problema em múltiplos contextos, incluindo situações imaginadas, não diretamente relacionadas com o aspecto prático-utilitário, expressar suas respostas e sintetizar conclusões, utilizando diferentes registros e linguagens (gráficos, tabelas, esquemas, além de texto escrito na língua materna e outras linguagens para descrever algoritmos, como fluxogramas, e dados).

CEF8	Interagir com seus pares de forma cooperativa, trabalhando coletivamente no planejamento e desenvolvimento de pesquisas para responder a questionamentos e na busca de soluções para problemas, de modo a identificar aspectos consensuais ou não na discussão de uma determinada questão, respeitando o modo de pensar dos colegas e aprendendo com eles.
Competências Específicas do Ensino Médio	
CEM1	Utilizar estratégias, conceitos e procedimentos matemáticos para interpretar situações em diversos contextos, sejam atividades cotidianas, sejam fatos das Ciências da Natureza e Humanas, das questões socioeconômicas ou tecnológicas, divulgados por diferentes meios, de modo a contribuir para uma formação geral.
CEM3	Utilizar estratégias, conceitos, definições e procedimentos matemáticos para interpretar, construir modelos e resolver problemas em diversos contextos, analisando a plausibilidade dos resultados e a adequação das soluções propostas, de modo a construir argumentação consistente.
CEM4	Compreender e utilizar, com flexibilidade e precisão, diferentes registros de representação matemáticos (algébrico, geométrico, estatístico, computacional etc.), na busca de solução e comunicação de resultados de problemas.
CEM5	Investigar e estabelecer conjecturas a respeito de diferentes conceitos e propriedades matemáticas, empregando estratégias e recursos, como observação de padrões, experimentações e diferentes tecnologias, identificando a necessidade, ou não, de uma demonstração cada vez mais formal na validação das referidas conjecturas.
Habilidades do Ensino Fundamental	
EF06MA34	Interpretar e desenvolver fluxogramas simples, identificando as relações entre os objetos representados.
EF07MA05	Resolver um mesmo problema utilizando diferentes algoritmos.
EF07MA06	Reconhecer que as resoluções de um grupo de problemas que têm a mesma estrutura podem ser obtidas utilizando os mesmos procedimentos.

EF07MA07	Representar por meio de um fluxograma os passos utilizados para resolver um grupo de problemas.
EF08MA03	Resolver e elaborar problemas de contagem cuja resolução envolva a aplicação do princípio multiplicativo.
Habilidades do Ensino Médio	
EF05MA09	Resolver e elaborar problemas simples de contagem envolvendo o princípio multiplicativo, como a determinação do número de agrupamentos possíveis ao se combinar cada elemento de uma coleção com todos os elementos de outra coleção, por meio de diagramas de árvore ou por tabelas.
EM13MAT106	Identificar situações da vida cotidiana nas quais seja necessário fazer escolhas levando-se em conta os riscos probabilísticos.
EM13MAT315	Identificar situações da vida cotidiana nas quais seja necessário fazer escolhas levando-se em conta os riscos probabilísticos.
EM13MAT405	Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática.
EM13MAT407	Interpretar e comparar conjuntos de dados estatísticos por meio de diferentes diagramas e gráficos (histograma, de caixa (box-plot), de ramos e folhas, entre outros), reconhecendo os mais eficientes para sua análise.

Fonte: Brasil (2017).

5.1 ATIVIDADE 1 - DESENHO DA CAIXA

Objetivo geral: Introduzir os alunos à teoria dos grafos, com foco nos trajetos eulerianos, por meio de desafios que envolvem a representação e a análise de problemas cotidianos e abstratos. A atividade busca desenvolver o raciocínio lógico e a capacidade de resolução de problemas, incentivando os alunos a explorar diferentes estratégias e a compreender as propriedades estruturais dos grafos.

Tabela 2 – Competências e Habilidades relacionadas à Atividade 1

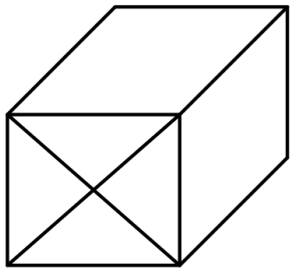
Competências Gerais	CG1, CG2, CG4
Competências Específicas	CEF2, CEF3, CEF5, CEF6, CEF8 CEM1, CEM3, CEM4
Habilidades	EF06MA34, EF07MA05, EF07MA06, EF07MA07, EM13MAT106, EM13MAT315

Fonte: Autora (2024).

ATIVIDADE

1. Você tem um desafio! Desenhe a caixa sem tirar o lápis do papel e sem riscar mais de uma vez a mesma linha.

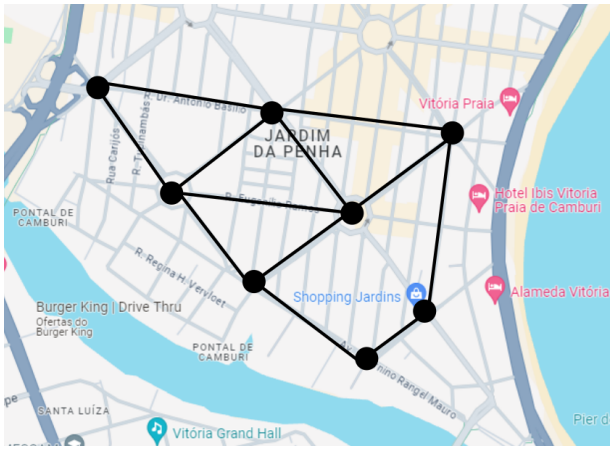
Figura 57 – Desenho da caixa



Fonte: Autora (2024).

2. O mapa abaixo mostra o bairro Jardim da Penha, em Vitória. Nos dias de quarta-feira, o caminhão de coleta de lixo precisa percorrer todas as ruas que estão destacadas. De forma a ser mais eficiente, o caminhão completa sua rota passando apenas uma vez por cada rua. Observe:

Figura 58 – Jardim da Penha



Fonte: Autora (2024) - Adaptado do Google maps.

Encontre uma das possíveis rotas que o caminhão faz. Por onde ele começa? por onde ele termina?

DESENVOLVIMENTO ESPERADO

Primeiro contato com a teoria dos grafos.

Objetivo específico - Compreender os comandos das questões e encontrar de forma intuitiva um trajeto euleriano em cada figura.

QUESTÃO 1

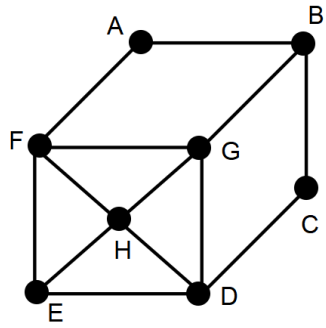
Você tem um desafio! Desenhe a caixa sem tirar o lápis do papel e sem riscar mais de uma vez a mesma linha.

Objetivo específico - Representar cada encontro de linhas como um vértice. Compreender a noção de um grafo.

Desenvolvimento - O desafio consiste em encontrar um **trajeto euleriano** no grafo, mas os estudantes ainda não sabem disso. Apresente o problema e deixe que eles resolvam; a atividade pode ser realizada em dupla. Posteriormente, deixe que os alunos se sintam à vontade para compartilhar seus resultados com o restante da turma.

Para facilitar a compreensão do trajeto que o lápis faz, podemos representar cada encontro de linhas como um vértice e nomeá-los. Como apresentamos na Figura (59).

Figura 59 – Grafo da caixa



Fonte: Autora (2024).

Algumas das resoluções do problema são:

$B \to A \to F \to G \to D \to C \to B \to G \to H \to F \to E \to H \to D \to E$
$B \to A \to F \to G \to B \to C \to D \to H \to G \to D \to E \to F \to H \to E$
$B \to G \to H \to D \to C \to B \to A \to F \to G \to D \to E \to F \to H \to E$
$B \to C \to D \to G \to B \to A \to F \to E \to H \to F \to G \to H \to D \to E$
$B \to C \to D \to E \to F \to A \to B \to G \to D \to H \to F \to G \to H \to E$

Deve-se considerar também seus respectivos caminhos inversos. Não vamos expor todos os possíveis trajetos aqui, até porque são muitos. O professor e os estudantes devem conferir se a resolução de cada dupla está correta. Após a análise de suas soluções, faça as seguintes perguntas:

a) Em quais pontos os trajetos começam?

b) Em quais pontos os trajetos terminam?

A resposta deverá ser B ou E, para ambos os casos. Indague se alguém da turma sabe responder o motivo disso acontecer. Provavelmente, a turma ficará curiosa em saber o motivo dos trajetos sempre começarem e terminarem em B ou E. O professor deve usar dessa curiosidade para discutir a segunda questão.

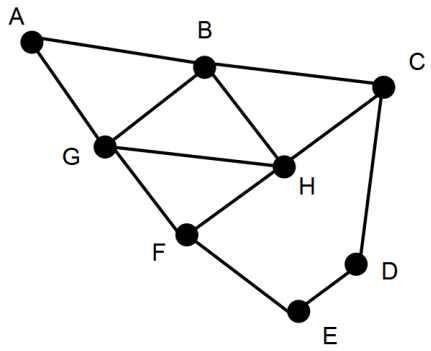
QUESTÃO 2

Encontre uma das possíveis rotas que o caminhão faz. Por onde ele começa? Por onde ele termina?

Objetivo específico - Representar cada cruzamento como um vértice e cada rua como uma aresta, compreender a noção de um grafo e o conceito de trajeto euleriano.

Desenvolvimento - O princípio de resolução deste problema se assemelha ao anterior, então iremos nos ater a nomear os cruzamentos e elencar possíveis trajetos:

Figura 60 – Grafo de Jardim da Penha



Fonte: Autora (2024).

$F \to G \to A \to B \to H \to G \to B \to C \to D \to E \to F \to H \to C$
$F \to E \to D \to C \to B \to A \to G \to B \to H \to G \to F \to H \to C$
$F \to E \to D \to C \to B \to A \to G \to B \to H \to G \to F \to H \to C$
$F \to H \to G \to B \to A \to G \to F \to E \to D \to C \to B \to H \to C$
$F \to H \to C \to D \to E \to F \to G \to H \to B \to G \to A \to B \to C$

Novamente, não iremos elencar todos os trajetos eulerianos deste grafo. Para os estudantes, basta encontrar um. Novamente, temos a pergunta: por onde começar e terminar os trajetos?

Para todos os trajetos corretos, a resposta é F ou C. Outra vez, fica a indagação:

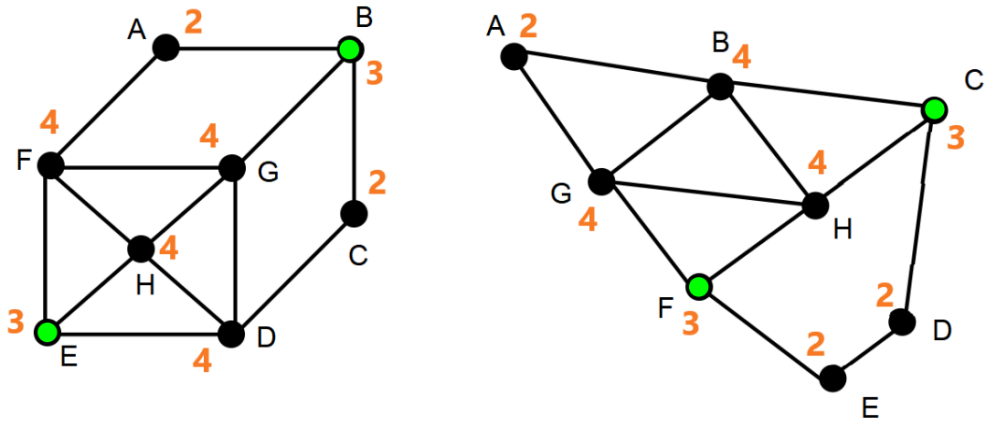
Por qual motivo estes pontos começam e terminam todos os trajetos? Que semelhanças podemos encontrar entre os pontos B e E da Questão 1 e os pontos F e C da Questão 2?

Após as discussões, espera-se chegar no seguinte raciocício: para que o trajeto possa ser feito de maneira que não repita nenhuma linha e passe por todas as linhas, todos os pontos devem ter uma linha de chegada e outra de saída. Desta maneira, a quantidade de linhas conectadas aos pontos deve ser par, pois chegada e saída representam sempre pares de linhas.

Todavia, percebemos: nem todos os pontos das figuras tem quantidade par de linhas conectadas, o que é o caso dos pontos B e E da Questão 1 e os pontos F e C da Questão 2. Estes

pontos possuem quantidade ímpar de linhas conectadas. No caso, 3 linhas. Observe a Figura (61):

Figura 61 – Graus dos vértices



Fonte: Autora (2024).

Deste modo, como é possível realizar o trajeto? Bem, em todos os pontos com quantidade par de linhas haverá entrada e saída, ou seja, por eles não há repetição de linhas. Porém, se existir um ponto com quantidade ímpar de linhas, então deve haver outro ponto com quantidade ímpar de linhas, pois assim um ponto será a partida do trajeto e o outro o fim do trajeto.

É aí que o professor introduz a teoria dos grafos para os estudantes. Explique que esta conclusão surgiu de um problema parecido, das pontes de Königsberg. Um matemático chamado Euler, em 1741 publicou o resultado de seu estudo sobre um roteiro que passasse sobre todas pontes da cidade, de maneira que nenhuma ponte fosse visitada mais de uma vez. Euler concluiu que tal roteiro era impossível, observando a quantidade de pontes conectadas a cada porção de terra. Veja mais detalhes sobre o problema na Seção (3.8).

Assim, o professor explica o problema de Euler e introduz o conceito de grafos para a turma:

Um grafo é uma estrutura que mostra a relação entre objetos. Estes objetos são representados por pontos chamados vértices e a relação entre eles é representada por linhas, chamadas de arestas.

A noção mais importante para resolução deste tipo de problema é a do **grau de um vértice**. O grau de um vértice é a quantidade de arestas a ele conectadas. Um trajeto que passa por todos as arestas do grafo, sem repetir nenhuma é chamado **trajeto euleriano**.

Basicamente, há três coisas a se destacar para o estudantes:

(i) É possível fazer um trajeto euleriano se os graus de todos os vértices for par. Neste caso, o começo e o fim do trajeto estarão no mesmo vértice.

(ii) É possível fazer um trajeto euleriano se apenas dois vértices possuirem grau ímpar, enquanto todos os outros têm grau par. Neste caso, o começo e o fim do trajeto estarão em vértices diferentes, que possuem grau ímpar.

(iii) Se o grafo possui mais de dois vértices com grau ímpar, então é impossível fazer um trajeto euleriano.

5.2 ATIVIDADE 2 - ANÁLISE DE AMIZADES

Objetivo geral: Compreender e analisar relações sociais através da teoria dos grafos, utilizando conceitos de graus de vértices, caminhos, e representações gráficas para interpretar, resolver problemas e encontrar soluções eficientes.

Tabela 3 – Competências e Habilidades relacionadas à Atividade 2

Competências Gerais	CG2, CG4
Competências Específicas	CEF2, CEF5, CEF6, CEM1, CEM3, CEM4
Habilidades	EF06MA34, EF07MA07, EM13MAT315

Fonte: Autora (2024).

ATIVIDADE

Vamos analisar relações de amizade entre cinco pessoas:

- Alice é amiga de Bernardo e Clara.
- Bernardo é amigo de Alice, Clara e Érica.
- Clara é amiga de Alice e Bernardo.
- Daniel é amigo de Érica.
- Érica é amiga de Bernardo e Daniel.

Responda:

1. Quem é o mais popular?
2. Se uma fofoca partir de Alice, quem ficaria sabendo por último?
3. Quais amigos Clara e Daniel têm em comum?
4. Desenhe uma maneira de todos se sentarem à uma mesa circular, de modo que tenham suas amizades o mais perto possível.

DESENVOLVIMENTO ESPERADO

Modelagem do problema

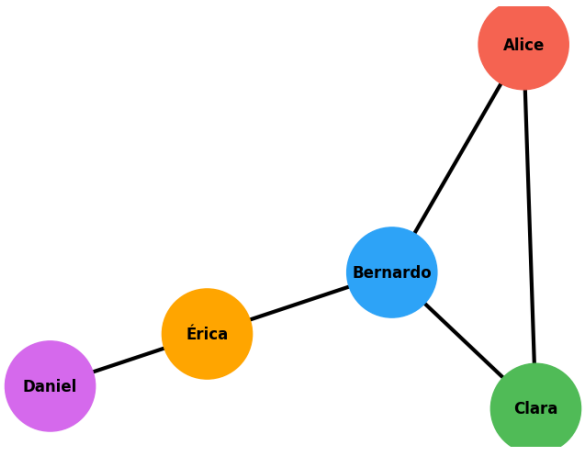
Objetivo específico - Representar cada pessoa como um vértice e cada relação de amizade como uma aresta.

Resolução - A lista de amizades fornecida é uma lista de adjacência.

Pessoa	**Amigos(as)**
Alice	Bernardo, Clara
Bernardo	Alice, Clara, Érica
Clara	Alice, Bernardo
Daniel	Érica
Érica	Bernardo, Daniel

Desta maneira, podemos construir a representação do grafo que mostra estas relações. Sendo cada pessoa um vértice e cada relação de amizade uma aresta.

Figura 62 – Relação de amizades



Fonte: Autora (2024).

QUESTÃO 1

Quem é o mais popular?

Objetivo específico - Calcular o grau de cada vértice para determinar popularidade.

Resolução - No grafo, analisamos os graus de cada vértice:

Pessoa	**Amizades**	**Grau do vértice**
Alice	Bernardo, Clara	2
Bernardo	Alice, Clara, Érica	3
Clara	Alice, Bernardo	2
Daniel	Érica	1
Érica	Bernardo, Daniel	2

Resposta - Como Bernardo tem grau igual a 3, sendo este o maior, significa que ele possui mais conexões (amizades) e portanto é o mais popular.

QUESTÃO 2

Se uma fofoca partir de Alice, quem ficaria sabendo por último?

Objetivo específico - Resolver problemas de alcance e determinar a propagação de informações no grafo, identificando o comprimento dos caminhos entre vértices.

Resolução - Podemos analisar essa situação como um problema de caminho para a fofoca chegar até as pessoas, assim identificamos os caminhos possíveis e seus comprimentos :

Caminhos	**Comprimento**
Alice $\rightarrow$ Bernardo	1
Alice $\rightarrow$ Clara	1
Alice $\rightarrow$ Bernardo $\rightarrow$ Clara	2
Alice $\rightarrow$ Bernardo $\rightarrow$ Érica	2
Alice $\rightarrow$ Bernardo $\rightarrow$ Érica $\rightarrow$ Daniel	3

Resposta - Verificando o comprimento de cada um dos passeios que a fofoca pode fazer, concluímos que Daniel será o último a saber, pois ele não tem conexão direta com Alice, e ele é o mais distante dela, em relação às amizades.

QUESTÃO 3

Quais amigos Clara e Daniel têm em comum?

Objetivo específico - Verificar interseções de amizades entre pares de vértices, identificando amigos em comum.

Resposta - Nenhum. Verificamos que nas arestas que incidem em Daniel e Clara, não há nenhuma cujo outro vértice extremo seja comum.

QUESTÃO 4

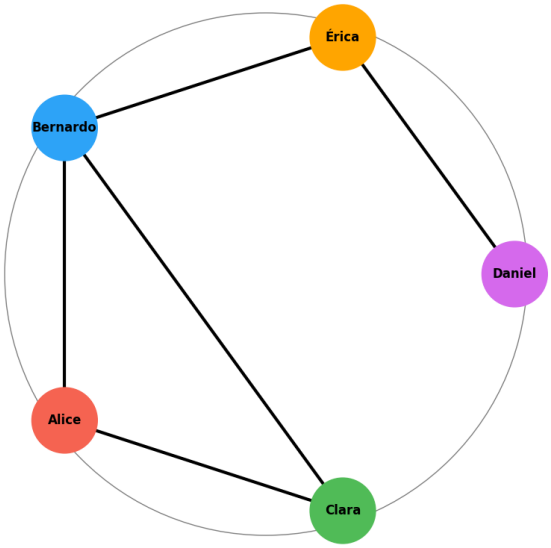
Desenhe uma maneira de todos se sentarem à uma mesa circular, de modo que tenham suas amizades o mais perto possível.

Objetivo específico - Desenhar representações gráficas que otimizem a proximidade de vértices conectados em uma disposição circular, minimizando a sobreposição de arestas.

Resolução - Existem três configurações possíveis para que todos se sentem à uma mesa circular, de modo que tenham suas amizades o mais perto possível. A melhor maneira de pensar na solução é organizar os vértices de modo que haja o mínimo possível de arestas que passem "sobre" a mesa.

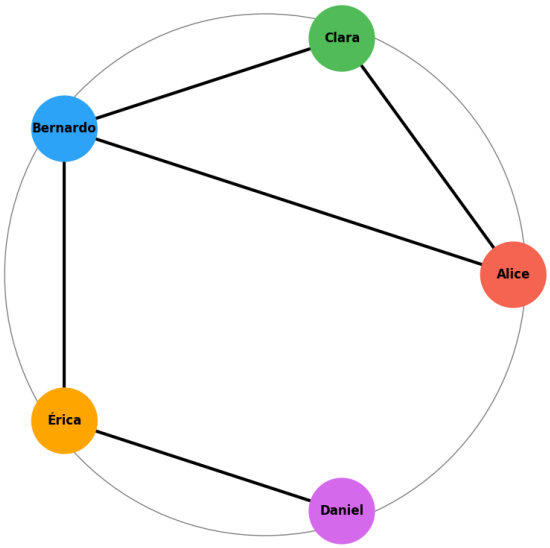
Configuração	Sequência	Representação
1	C, A, B, E, D	Figura (63)
2	A, C, B, E, D	Figura (64)
3	D, E, B, A, C	Figura (65)

Figura 63 – Configuração circular 1



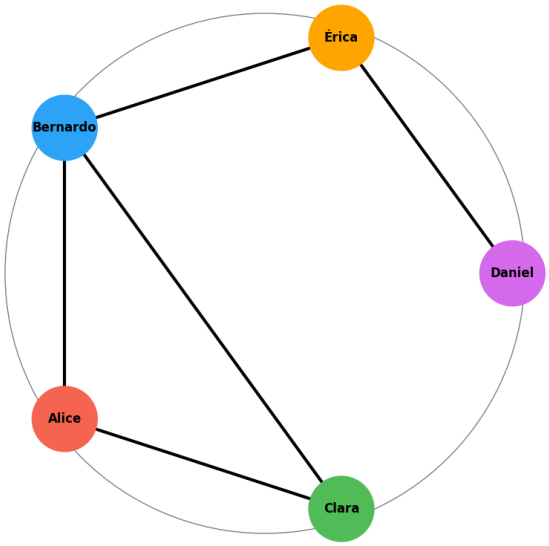
Fonte: Autora (2024).

Figura 64 – Configuração circular 2



Fonte: Autora (2024).

Figura 65 – Configuração circular 3



Fonte: Autora (2024).

5.3 ATIVIDADE 3 - MAPA DO JALAPÃO E RECIFE

Objetivo geral: Aplicar a teoria dos grafos para resolver problemas práticos de otimização de rotas, localização estratégica e viabilidade de percursos, utilizando conceitos como grau de vértice, caminho mínimo, caminho hamiltoniano e trajeto euleriano, a fim de desenvolver habilidades de modelagem, análise e tomada de decisão eficiente em situações reais.

Tabela 4 – Competências e Habilidades relacionadas à Atividade 3

Competências Gerais	CG2, CG4
Competências Específicas	CEF2, CEF3, CEF5, CEF6, CEM1, CEM3, CEM4
Habilidades	EF06MA34, EF07MA05, EF07MA06, EF07MA07, EM13MAT106

Fonte: Autora (2024).

Responda às questões conforme o mapa da Figura (66).

Figura 66 – Mapa do Jalapão

Fonte: Autora (2024).

ATIVIDADE

1. Um empresário deseja construir um posto de gasolina em uma cidade que atenda o maior número de estradas asfaltadas conectadas diretamente a outras cidades. Em qual cidade seria adequado construir este posto?

2. Uma ambulância precisa levar um paciente de Ponte Alta para o hospital HGP, em Palmas. Para isso, o motorista precisa fazer o percurso mais curto entre Ponte Alta e

Palmas, para diminuir os custos do transporte. Qual caminho ele deve escolher?

3. Um grupo de turistas deseja sair de Palmas e visitar todas as cidades/distritos, passando apenas uma vez em cada (sem se importar com o destino final). Este trajeto é possível?

4. Trabalhadores responsáveis pela verificação das estradas precisam sair de Palmas e passar por todas as estradas. É possível passar vistoriando apenas uma vez cada estrada e voltar para Palmas?

DESENVOLVIMENTO ESPERADO

Modelagem do problema

Objetivo específico - Representar cada cidade/distrito/cruzamento como um vértice e cada estrada como uma aresta, e apelidar os locais de maneira mais simples.

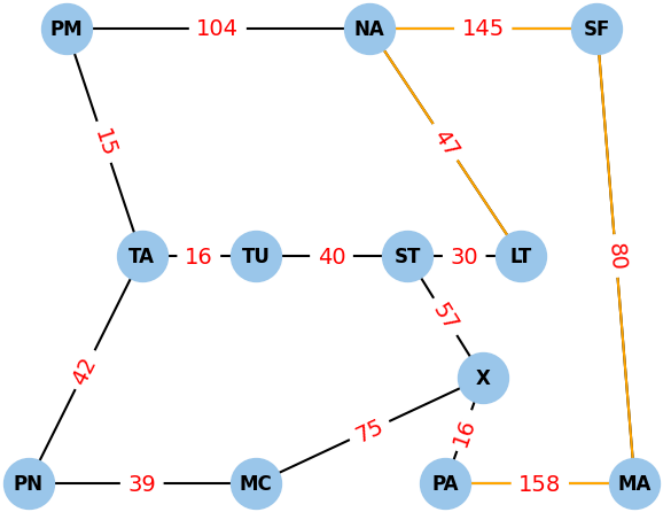
Resolução - Para simplificar a visualização, podemos abreviar os nomes de cada local:

Abreviação	**Local**
PM	Palmas
TA	Taquaralto
TU	Taquaruçu
ST	Santa Tereza
LT	Lagoa do Tocantins
NA	Novo Acordo
SF	São Félix do Tocantins
MA	Mateiros
PA	Ponte Alta
MC	Monte do Carmo
PN	Porto Nacional

Assim, fazemos o grafo que representa o mapa, colocando cada distância entre as cidades como peso da aresta, sendo as arestas laranjas representando estrada de terra, e as pretas as

estradas asfaltadas, conforme a Figura (67).

Figura 67 – Grafo - Jalapão



Fonte: Autora (2024).

QUESTÃO 1

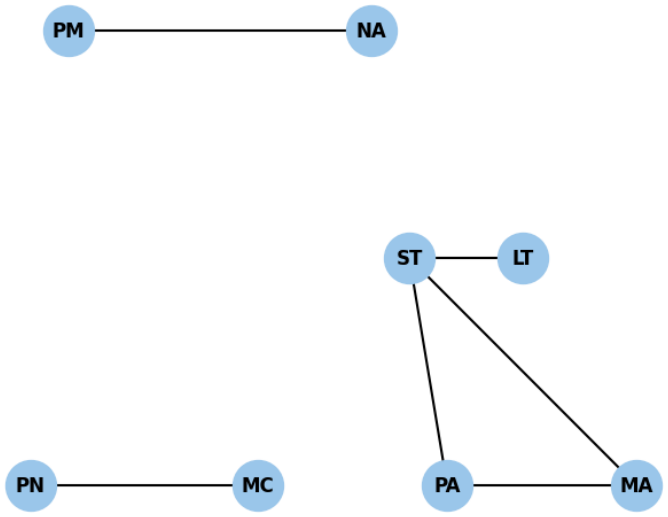
Um empresário deseja construir um posto de gasolina em uma cidade que atenda o maior número de estradas asfaltadas conectadas diretamente a outras cidades. Em qual cidade seria adequado construir este posto?

Objetivo específico - Extrair e filtrar as informações do problema, a fim de resolvê-lo segundo os critérios estabelecidos, utilizando do conceito de grau de um vértice no grafo.

Resolução - No grafo, analisamos os graus de cada vértice, mas precisamos nos atentar para o seguinte: a questão se limita a cidades vizinhas conectadas diretamente por estradas asfaltadas. Logo, excluiremos da análise todas as estradas de terra e distritos.

Na região do cruzamento, podemos considerar um trajeto direto e excluir o vértice X, já que as cidades se conectam diretamente, sem passar por nenhuma outra intermediária. Como a distância entre cidades é irrelevante, também tiramos do grafo os pesos das arestas. Desta maneira, o modelo de grafo analisado é como o da Figura (68):

Figura 68 – Grafo - Estradas asfaltadas entre cidades



Fonte: Autora (2024).

Fazendo a análise dos graus dos vértices através da lista de adjacências:

Cidade	**Conexões**	**Grau do vértice**
PM	NA	1
NA	PM	1
PN	MC	1
MC	PN	1
ST	LT, PA, MA	3
LT	ST	1
PA	ST, MA	2
MA	PA, ST	2

Pela Figura (68) e sua lista de adjacências, fica fácil perceber que o vértice com mais conexões é o ST, que representa a cidade de Santa Tereza do Tocantins.

Resposta - Seria adequado construir o posto na cidade de Santa Tereza do Tocantins, pois segundo análise ela é a que mais se conecta com outras cidades segundo os critérios exigidos.

QUESTÃO 2

Uma ambulância precisa levar um paciente de Ponte Alta para o hospital HGP, em Palmas. Para isso, o motorista precisa escolher o percurso mais curto entre Ponte Alta e Palmas,

para diminuir os custos do transporte. Qual caminho ele deve escolher?

Objetivo específico - Compreender a aplicação prática da teoria dos grafos em situações reais de otimização e desenvolver habilidades de resolução de problemas, utilizando métodos matemáticos e/ou lógicos para tomar decisões eficientes.

Resolução -A análise de caminhos mínimos geralmente é uma questão mais complexa a depender do grafo e dos vértices analisados, que envolve o chamado **Algoritmo de Dijkstra** - veja mais em (Carvalho, 2008), mas neste caso faremos a análise de maneira mais intuitiva e visual, considerando que o mapa possui escala e portanto não faz sentido, por exemplo, passar por São Félix ou Mateiros para ir à Palmas, visto que estas cidades estão direção contrária (leste) ao nosso destino (oeste).

Também não faz sentido visitar Novo Acordo no trajeto, visto que esta cidade está mais distante (mais ao norte) do que a cidade de Palmas. Encontraremos a resposta analisando o mapa da Figura (66) e encontrando possíveis caminhos de maneira lógica.

Caminho	**Arestas**	**Distâncias**	**Comprimento**
1	$PA \to X \to ST \to TU \to TA \to PM$	$16+57+40+16+15$	128 km
2	$PA \to X \to MC \to PN \to TA \to PM$	$16+75+39+42+15$	187 km

Uma outra maneira de encontrar a resposta, fazendo menos contas, é encontrando as arestas comuns aos dois caminhos e eliminando-as da análise, assim temos:

Caminho	**Arestas**	**Distâncias**	**Comprimento**
1	$X \to ST \to TU \to TA$	$57+40+16$	113 km
2	$X \to MC \to PN \to TA$	$75+39+42$	156 km

Resposta - De qualquer modo, pelo comprimento dos caminhos concluímos que o mais curto é o Caminho 1, ou seja, Palmas $\to$ Taquaralto $\to$ Taquaruçu $\to$ Santa Tereza $\to$ X $\to$ Ponte Alta.

Observação - Este problema é mais intuitivo, pois se trata de um mapa, ou seja, as distâncias obedecem certa escala no plano. A depender do grafo, o problema não pode ser tratado desta forma, pois nem todo vértice que está mais distante visualmente é o mais distante no grafo. Por isso, todo problema deve ser analisado cautelosamente de maneira a assegurar que todas as

possibilidades sejam consideradas.

QUESTÃO 3

Um grupo de turistas deseja sair de Palmas e visitar todas as cidades/distritos, passando apenas uma vez em cada (sem se importar com o destino final). Este trajeto é possível?

Objetivo específico - Desenvolver habilidades de resolução de problemas para determinar a viabilidade do percurso proposto.

Resolução - A resolução deste problema é um encontrar um **caminho hamiltoniano** - veja mais na Seção (3.10). Não se conhece uma condição necessária e suficiente trivial para a existência de um ciclo hamiltoniano em um grafo. Deste modo, fazemos uma busca exaustiva, que é possível para o grafo em questão, mas muito difícil para outros grafos mais complexos.

Caminhos hamiltonianos existentes:

Caminho	**Arestas**
1	PM $\rightarrow$ NA $\rightarrow$ SF $\rightarrow$ MA $\rightarrow$PA $\rightarrow$ X $\rightarrow$ MC $\rightarrow$PN $\rightarrow$ TA $\rightarrow$ TU $\rightarrow$ ST $\rightarrow$ LT
2	PM $\rightarrow$ TA $\rightarrow$ TU $\rightarrow$ ST $\rightarrow$ LT $\rightarrow$ NA $\rightarrow$ SF $\rightarrow$ MA $\rightarrow$ PA $\rightarrow$ X $\rightarrow$ MC $\rightarrow$ PN
3	PM $\rightarrow$ TA $\rightarrow$ PN $\rightarrow$ MC $\rightarrow$ X $\rightarrow$ PA $\rightarrow$ MA $\rightarrow$ NA $\rightarrow$ LT $\rightarrow$ ST $\rightarrow$ TU

QUESTÃO 4

Trabalhadores responsáveis pela verificação das estradas precisam sair de Palmas e passar por todas as estradas. É possível passar vistoriando apenas uma vez cada estrada e voltar para Palmas?

Objetivo específico - Compreender as condições necessárias para a existência de um trajeto euleriano em um grafo.

Resolução - Para passarmos por todas as estradas apenas uma vez, cada cidade deve ter pelo menos uma estrada de entrada e outra de saída, mantendo sempre a quantidade par de estradas. A resolução deste problema é encontrar um **trajeto euleriano** fechado. Pelo Teorema de Euler (1741), sabemos que este tipo de trajeto só é possível em um grafo que contenha todos os vértices

com grau par. Vamos verificar o grau de cada vértice através da lista de adjacências do grafo representado na Figura (67):

Vértice	Adjacências	Grau
PM	TA, NA	2
TA	PM, TU, PN	3
TU	TA, ST	2
PN	TA, MC	2
MC	PN, X	2
X	MC, ST, PA	3
ST	X, LT, TU	3
PA	MA, X	2
MA	PA, SF	2
SF	MA, NA	2
NA	PM, SF, LT	3
LT	NA, ST	2

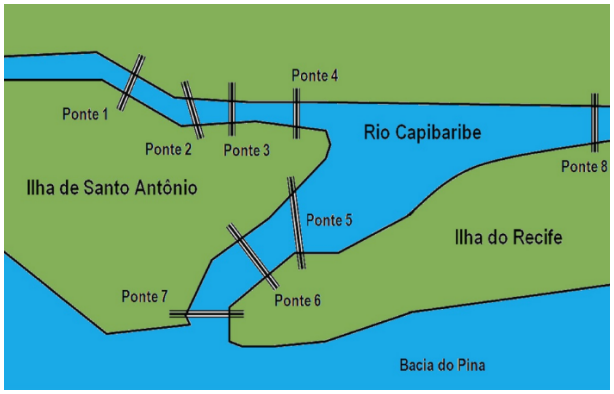
O grafo do mapa em questão, contém 4 vértices de grau ímpar, portanto não contém um trajeto euleriano.

Resposta - É impossível sair de Palmas e passar por todas as estradas apenas uma vez.

QUESTÃO EXTRA

É possível fazer um passeio por todas as pontes do centro de Recife - PE exatamente uma vez e retornar ao ponto de partida?

Figura 69 – Centro de Recife com suas 8 pontes

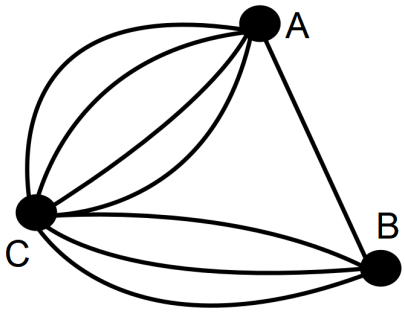


Fonte: Carvalho (2019).

DESENVOLVIMENTO ESPERADO

Podemos modelar o problema construindo a imagem do seguinte grafo:

Figura 70 – Grafo do centro de Recife



Fonte: Autora (2024).

Assim, verificamos os graus de cada vértice:

- Vértice A: possui grau 5;
- Vértice B: possui grau 4;
- Vértice C: possui grau 7.

Resposta - Como o grafo possui exatamente dois vértices de grau ímpar, é possível traçar um trajeto semieuleriano, ou seja, passar por todas as pontes exatamente uma vez; mas é impossível fazer um trajeto euleriano fechado, visto que o grafo não possui todos os vértices com grau par, logo não é possível voltar ao ponto de partida com estas condições.

5.4 ATIVIDADE 4 - PINTAR UM MAPA

Objetivo geral- Introduzir os alunos à coloração de grafos, utilizando o Teorema das Quatro Cores, e ao conceito de vizinhança em grafos, promovendo a compreensão e aplicação da teoria dos grafos na modelagem e resolução de problemas práticos, como a coloração de mapas, garantindo que regiões adjacentes não compartilhem a mesma cor.

Tabela 5 – Competências e Habilidades relacionadas à Atividade 4

Competências Gerais	CG1, CG2, CG4
Competências Específicas	CEF2, CEF3, CEF5, CEF6, CEM1, CEM3, CEM4
Habilidades	EF06MA34, EF07MA05

Fonte: Autora (2024).

ATIVIDADE

1. Pinte o mapa abaixo, de maneira que nenhum país vizinho tenha a mesma cor. Use o mínimo de cores que conseguir.

Figura 71 – Mapa da América do Sul

Fonte: Autora (2024).

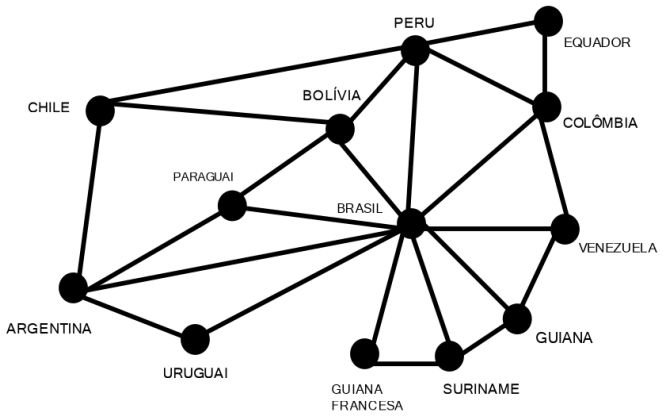
DESENVOLVIMENTO ESPERADO

Modelagem do problema

Objetivo específico - Representar o mapa da América do Sul por meio de um grafo.

Resolução - Qualquer mapa pode ser representado por meio de um grafo, no qual os vértices são as regiões e as arestas são as fronteiras entre as regiões. No caso da América do Sul, podemos representar seu grafo da seguinte maneira:

Figura 72 – Grafo que representa o mapa da América do Sul



Fonte: Autora (2024).

Uma coisa interessante sobre grafos de mapas, é que eles sempre são planares, ou seja, é possível desenhá-lo sem que suas arestas se cruzem. Deste modo, o problema de colorir um mapa é equivalente ao problema de colorir os vértices de um grafo. Neste caso, sem que vértices vizinhos tenham a mesma cor.

Na apresentação da resolução do problema, é interessante que o professor explique o conceito de vizinhança: dois vértices são vizinhos quando estão conectados por uma aresta.

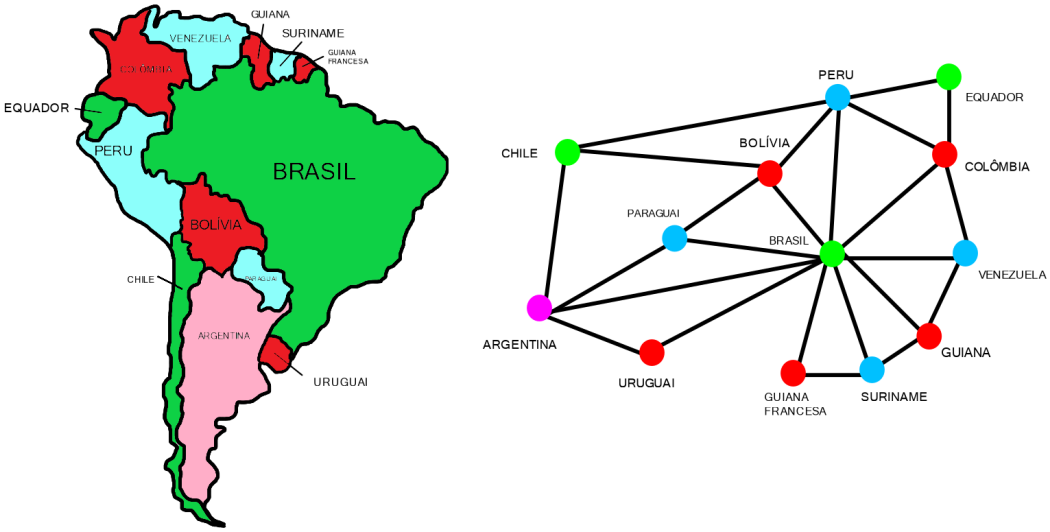
QUESTÃO 1

Pinte o mapa, de maneira que nenhum país vizinho tenha a mesma cor. Use o mínimo de cores que conseguir.

Objetivo específico - Compreender o conceito de vizinhança em um grafo.

Possível Resolução:

Figura 73 – Mapa e grafo coloridos - problemas equivalentes



Fonte: Autora (2024).

Esta atividade tem várias possíveis soluções de cores, mas apenas uma resposta para a quantidade mínima de cores. O **Teorema das Quatro Cores** afirma que não mais do que quatro cores são necessárias para colorir as regiões de qualquer mapa, de modo que duas regiões adjacentes não tenham a mesma cor - veja mais na Seção (3.11). Este teorema, que a princípio parece bem simples, demorou mais de um século para ser provado, e sua confirmação só foi possível com o auxílio de um computador, que precisou de 1200 horas de processamento e 10 bilhões de decisões lógicas até chegar à prova. Um computador não reconhce desenhos, nem mapas, por isso modelar o problema como um grafo é tão importante em resoluções mais complexas.

5.5 ATIVIDADE 5 - CONTAGEM DE ARESTAS

Objetivo geral- Desenvolver e aplicar princípios de contagem em grafos e verificar como podemos modelar problemas do cotidiano através de grafos.

Tabela 6 – Competências e Habilidades relacionadas à Atividade 5

Competências Gerais	CG1, CG2, CG4
Competências Específicas	CEF2, CEF3, CEF5, CEF6, CEM1, CEM3, CEM4, CEM5
Habilidades	EF07MA06, EF08MA03, EF05MA09, EM13MAT310

Fonte: Autora (2024).

ATIVIDADE

1. Em um estado há 80 municípios, sabemos que de cada município parte exatamente 3 estradas diretas para outras cidades. Quantas estradas existem no total?
2. Em um torneio de futebol, há 20 times, e cada time joga exatamente 5 partidas com outros times. Ao final do torneio, quantos jogos ocorreram no total?
3. Em uma festa há 13 pessoas, cada uma cumprimentou todas as outras com um único aperto de mão. Quantos cumprimentos houveram na festa?
4. É possível haver um grafo com 55 vértices, de maneira que cada vértice seja vizinho de exatamente 3 vértices?

DESENVOLVIMENTO ESPERADO

Modelagem do problema

Objetivo específico - Compreender como grafos podem ser "encontrados" em situações do cotidiano.

Desenvolvimento - Cada uma das questões 1, 2, e 3 pode ser interpretada como um grafo. Observe: 1. Cada município é um vértice, cada estrada é uma aresta. 2. Cada time é um vértice, cada jogo é uma aresta. 3. Cada pessoa é um vértice, cada aperto de mão é uma aresta.

QUESTÃO 1

Em um estado há 80 municípios, sabemos que de cada município parte exatamente 3 estradas diretas para outras cidades. Quantas estradas existem no total?

Objetivo específico - Responder a questão através de análise combinatória.

Resolução- Se de cada município parte 3 estradas, como há 80 municípios então contamos $80 \cdot 3 = 240$ estradas. O problema é que esta resolução é incompleta, visto que estaríamos contando cada estrada duas vezes (uma vez quando a estrada sai de uma cidade e outra vez quando ela chega na outra cidade), ou seja, precisamos tirar o problema de duplicidade da contagem.

Resposta- Desta maneira, há $\frac{240}{2} = 120$ estradas no total.

QUESTÃO 2

Em um torneio de futebol, há 20 times, e cada time joga exatamente 5 partidas com outros times. Ao final do torneio, quantos jogos ocorreram no total?

Objetivo específico - Responder a questão através de análise combinatória.

Resolução- Se de cada time joga 5 vezes, como há 20 times então contamos $20 \cdot 5 = 100$ jogos. O problema é que esta resolução é incompleta, visto que estaríamos contando cada jogo duas vezes (uma vez para um time e outra vez para o outro time), ou seja, precisamos tirar o problema de duplicidade da contagem.

Resposta- Desta maneira, há $\frac{100}{2} = 50$ jogos no total.

QUESTÃO 3

Em uma festa há 13 pessoas, cada uma cumprimentou todas as outras com um único aperto de mão. Quantos cumprimentos houveram na festa?

Objetivo específico - Responder a questão através de análise combinatória.

Resolução- Se cada pessoa cumprimentou todas as outras, significa que cada pessoa apertou a mão de 12 pessoas, já que ela não aperta a mão de si mesma. Assim, contamos $13 \cdot 12 = 156$ apertos de mão. O problema é que esta resolução é incompleta, visto que estaríamos contando cada aperto de mão duas vezes (uma vez para uma pessoa envolvida e outra vez para a outra pessoa), ou seja, precisamos tirar o problema de duplicidade da contagem.

Resposta- Desta maneira, há $\frac{156}{2} = 78$ cumprimentos no total.

QUESTÃO 4

É possível haver um grafo com 55 vértices, de maneira que cada vértice seja vizinho de exatamente 3 vértices?

Objetivo específico - Compreender o Teorema do Aperto de Mãos.

Resolução- Utilizando o mesmo raciocínio das questões anteriores, contamos a quantidade de arestas da seguinte maneira: $55 \cdot 3 = 165$ arestas, mas assim cada aresta estaria sendo contada duas vezes (uma vez quando sai de um vértice e uma vez quando ela chega em outro vértice), deste modo tiramos a duplicidade, fazendo $\frac{165}{2}$. O problema é que $\frac{165}{2}$ não é um número inteiro, visto que 165 não é par, e podemos contar arestas apenas com números naturais (logo, inteiros).

Resposta- Deste modo, é **impossível** haver um grafo com 55 vértices, de maneira que cada vértice seja vizinho de exatamente 3 vértices.

Desenvolvimento - Estes problemas são justificados pelo **Teorema do Aperto de mãos**, que diz o seguinte: a soma dos graus de todos os vértices é igual ao dobro do número de arestas. Veja mais na Subseção (3.3.1). Deste modo, percebemos que a soma dos graus de todos os vértices é sempre par.

Neste problema, sendo a quantidade de vértices, 55, um número ímpar; e o grau de cada vértice, 3, um numero ímpar, a soma dos graus de todos os vértices também seria ímpar, pois o produto de números ímpares é sempre ímpar. Portanto, pelo Teorema do Aperto de mãos, é impossível um grafo com estas configurações existir.

5.6 ATIVIDADE 6 - REDE SOCIAL

Objetivo geral- Promover a compreensão e aplicação de conceitos fundamentais da teoria dos grafos no contexto de redes sociais, identificar e representar grafos direcionados, interpretar vizinhanças de vértices e calcular graus de entrada e saída através da representação gráfica, de tabela e análise das relações estabelecidas; bem como traduzir um grafo em uma matriz e fazer operações com matrizes.

Tabela 7 – Competências e Habilidades relacionadas à Atividade 6

Competências Gerais	CG2, CG4, CG5
Competências Específicas	CEF2, CEF3, CEF5, CEF6, CEM1, CEM3, CEM4, CEM5
Habilidades	EF06MA34, EF07MA05, EF07MA06, EM13MAT315, EM13MAT405, EM13MAT407

Fonte: Autora (2024).

ATIVIDADE

Um novo time de voleibol está se formando, e para fortalecer o entrosamento, os jogadores decidiram participar de uma rede social exclusiva do time, onde podem seguir alguém ou serem seguidos. No primeiro dia, os jogadores adicionaram uns aos outros na rede de acordo com quem eles queriam fazer amizade. A tabela abaixo mostra quem cada jogador seguiu.

Jogador	**Quem o jogador seguiu**
Ana	Bruno, Carlos
Bruno	Ana, Carlos, Ester
Carlos	Ana, Bruno
Davi	Ester
Ester	Bruno, Davi
Fábio	Carlos, Davi, Helen
Gabriel	Davi, Isabela, João
Helen	Bruno, Ana
Isabela	Ester, Carlos, Fábio
João	Helen
Kátia	

1. Represente de forma gráfica a rede social.

2. Quais alunos formam um “círculo” de amizades, onde todos se seguem mutuamente?

3. Quem poderia aparecer na rede como recomendação de amizade para o João?
4. Quais são os jogadores mais influenciados na rede social?
5. Existe alguém isolado na rede social?
6. Quem é o mais popular da rede? (quem tem mais seguidores)
7. Quem poderia ser o *influencer* da rede? (quem tem mais seguidores e segue menos pessoas)

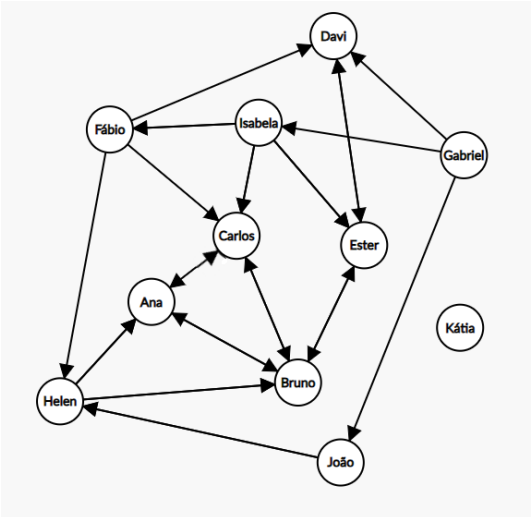
DESENVOLVIMENTO ESPERADO - QUESTÃO 1

Represente de forma gráfica a rede social.

Objetivo específico - Compreender como grafos estão presentes em redes sociais e representar um grafo a partir de sua lista de adjacência.

Resolução - Representamos a rede social através do desenho de um grafo. Cada jogador deve ser representado por um vértice. As arestas representam "quem segue quem", mas aqui há um conceito diferente dos outros vistos anteriormente: João segue Helen, mas Helen não segue João. Neste caso, coloca-se setas nas arestas para direcionar a relação, assim a aresta pode ser chamada de arco; é o que chamamos de **grafo direcionado**, conforme mostrado na Figura (74).

Figura 74 – Rede Social



Fonte: Autora (2024).

QUESTÃO 2

Quais alunos formam um “círculo” de amizades, onde todos se seguem mutuamente?

Objetivo específico - Analisar a vizinhança do grafo e tirar conclusões a partir das relações representadas.

Resposta - Pela Figura (74) fica fácil perceber que Ana, Carlos e Bruno formam um “círculo” de amizade, onde todos se seguem mutuamente (setas bidirecionadas)

QUESTÃO 3

Quem poderia aparecer na rede como recomendação de amizade para o João?

Objetivo específico - Analisar a vizinhança do grafo e tirar conclusões a partir das relações representadas.

Resposta - Pela Figura (74) percebemos que Gabriel segue João, mas João não segue Gabriel, então Gabriel poderia aparecer como recomendação de amizade. Também temos que João segue Helen, e Helen segue Ana e Bruno, logo Ana e Bruno também poderiam aparecer como recomendação de amizade para João.

QUESTÃO 4

Quais são os jogadores mais influenciados na rede social? (que seguem mais pessoas)

Objetivo específico - Calcular o grau de saída de cada vértice para determinar os mais influenciados.

Resolução - No grafo, analisamos os graus de saída de cada vértice:

Jogador	Quem o jogador seguiu	Grau de saída de cada vértice
Ana	Bruno, Carlos	2
Bruno	Ana, Carlos, Ester	3
Carlos	Ana, Bruno	2
Davi	Ester	1
Ester	Bruno, Davi	2
Fábio	Carlos, Davi, Helen	3
Gabriel	Davi, Isabela, João	3
Helen	Bruno, Ana	2
Isabela	Ester, Carlos, Fábio	3
João	Helen	1
Kátia		0

Resposta - Como Bruno, Fábio, Gabriel e Isabela seguem mais pessoas, então eles são os mais infleunciados da rede.

QUESTÃO 5

Existe alguém isolado na rede social?

Objetivo específico - Verificar a existência de um vértice isolado na rede social.

Resposta - Sim, a kátia é isolada na rede social, pois não segue ninguém e também não possui seguidores.

QUESTÃO 6

Quem é o mais popular da rede? (quem tem mais seguidores)

Objetivo específico - Calcular o grau de entrada de cada vértice para determinar popularidade.

Resolução - o grafo, analisamos os graus de entrada de cada vértice:

Jogador	Seus seguidores	Grau de entrada do vértice
Ana	Bruno, Carlos, Helen	3
Bruno	Ana, Carlos, Ester, Helen	4
Carlos	Isabela, Fábio, Ana, Bruno	4
Davi	Fábio, Ester, Gabriel	3
Ester	Bruno, Davi, Isabela	3
Fábio		0
Gabriel		0
Helen	Fábio, João	2
Isabela	Gabriel	1
João	Gabriel	1
Kátia		0

Resposta - Como Bruno e Carlos possuem mais seguidores que o restante, eles são os mais populares da rede social.

QUESTÃO 7

Quem poderia ser o *influencer* da rede? (quem tem mais seguidores e segue menos pessoas)

Objetivo específico - Calcular a diferença entre grau de entrada e saída de cada vértice, a fim de determinar a maior capacidade de influência, bem como resolver o problema através de uma matriz de adjacência do grafo.

Resolução - No grafo, calculamos a diferença entre o grau de entrada e saída de cada vértice, assim o *influencer* será o que tiver a maior diferença entre os valores.

Jogador	Grau de entrada do vértice	Grau de saída de cada vértice	Diferença
Ana	3	2	1
Bruno	4	3	1
Carlos	4	2	2
Davi	3	1	2
Ester	3	2	1
Fábio	0	3	-3
Gabriel	0	3	-3
Helen	2	2	0
Isabela	1	3	-2
João	1	1	0
Kátia	0	0	0

Resposta - Como Carlos e Davi são os que possuem maior diferença entre número de seguidores

e número de pessoas que eles seguem, então eles são os *influencers* da rede social.

Observação - Este problema também pode ser resolvido através de uma matriz de adjacência, onde cada relação linha-coluna representa uma aresta. Na existência da aresta, a célula é preenchida com 1, na ausência da aresta a célula fica vazia ou é preenchida com zero.

O grau de saída do vértice v_i é dado pela soma dos elementos da linha i. O grau de entrada do vértice v_i é dado pela soma dos elementos da coluna i. Para facilitar a escrita, representamos cada jogador pela primeira letra de seu nome. Assim, temos a seguinte matriz de adjacência:

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \\ J \\ K \\ \text{entrada} \end{array} \begin{array}{c} \begin{array}{cccccccccccc} A & B & C & D & E & F & G & H & I & J & K & \text{saída} \end{array} \\ \begin{pmatrix} & 1 & 1 & & & & & & & & & 2 \\ 1 & & 1 & & 1 & & & & & & & 3 \\ 1 & 1 & & & & & & & & & & 2 \\ & & & & 1 & & & & & & & 1 \\ & 1 & & 1 & & & & & & & & 2 \\ & & 1 & 1 & & & & 1 & & & & 3 \\ & & & 1 & & & & & 1 & 1 & & 3 \\ 1 & 1 & & & & & & & & & & 2 \\ & & 1 & & 1 & 1 & & & & & & 3 \\ & & & & & & & 1 & & & & 1 \\ & & & & & & & & & & & 0 \\ 3 & 4 & 4 & 3 & 3 & 0 & 0 & 2 & 1 & 1 & 0 & \end{pmatrix} \end{array}$$

Deste modo, fazemos:

$$
\begin{array}{c}
\\
A \\
B \\
C \\
D \\
E \\
F \\
G \\
H \\
I \\
J \\
K
\end{array}
\begin{array}{c}
\text{entrada} \\
\begin{pmatrix} 3 \\ 4 \\ 4 \\ 3 \\ 3 \\ 0 \\ 0 \\ 2 \\ 1 \\ 1 \\ 0 \end{pmatrix}
\end{array}
-
\begin{array}{c}
\text{saída} \\
\begin{pmatrix} 2 \\ 3 \\ 2 \\ 1 \\ 2 \\ 3 \\ 3 \\ 2 \\ 3 \\ 1 \\ 0 \end{pmatrix}
\end{array}
=
\begin{array}{c}
\text{diferença} \\
\begin{pmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 1 \\ -3 \\ -3 \\ 0 \\ -2 \\ 0 \\ 0 \end{pmatrix}
\end{array}
$$

Resposta - Como Carlos e Davi são os que possuem maior diferença entre número de seguidores e número de pessoas que eles seguem, então eles são os *influencers* da rede social.

As matrizes são muito importantes para armazenar e manipular grafos no computador, pois é assim que os computadores entendem os vértices e arestas existentes, bem como outras propriedades.

5.7 ATIVIDADE 7 - COMO O GOOGLE FUNCIONA?

Objetivo geral- Compreender a relação entre páginas da *web* e grafos, ser capaz de organizar informações e seguir passo a passo os comandos de um algoritmo, a fim de tirar conclusões com base nos resultados alcançados.

Tabela 8 – Competências e Habilidades relacionadas à Atividade 7

Competências Gerais	CG1, CG4, CG5
Competências Específicas	CEF2, CEF3, CEF5, CEF6, CEM1, CEM3, CEM4, CEM5
Habilidades EF06MA34	EF07MA05, EF07MA06, EF07MA07, EM13MAT106, EM13MAT315, EM13MAT405, EM13MAT407

Fonte: Autora (2024).

Prévia:

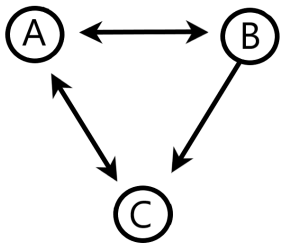
Você já parou para pensar em como o motor de busca do *Google* funciona? Dentre os *sites* que contém as palavras chaves que você pesquisar, o *Google* faz um *ranking* para te mostar as mais relevantes primeiro. Este *ranking* é calculado de acordo com a popularidade de cada página. Quanto mais *links* uma página recebe de outras, mais relevante ela é. Quanto mais links uma página envia para outras, menos relevante ela é.

Assim, o algoritmo chamado *PageRank* faz o cálculo destas relevâncias, levando em conta que se uma página recebe *links* de páginas relevantes, então ela também se torna mais relevante. Este algoritmo é uma das várias aplicações da teoria dos grafos, pois cada página é um vértice e cada *link* enviado para outra página é uma aresta direcionada.

Para calcular o ranking, os vértices (páginas) devem ser valorados e as arestas (*links*) devem conter pesos. Assim, definimos que o grafo no geral deve conter peso 1, inicialmente, este peso será distribuído igualmente entre todas as páginas. Veja mais no Capítulo (4).

Vamos apresentar como exemplo, o cálculo do PageRank para os vértices A, B e C da Figura (75) de forma simplificada. Todos os resultados serão aproximados para três casas decimais, para facilitar os cálculos e a visualização.

Figura 75 – Mini rede web



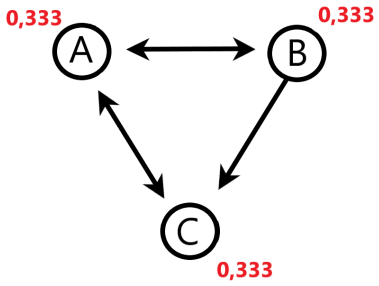
Fonte: Autora (2024).

Inicialização

O PageRank inicial (p) para cada vértice é calculado como $\frac{1}{|V|}$, onde $|V|$ é o número de vértices. Como neste exemplo $|V| = 3$, temos:

$$p(A) = p(B) = p(C) = \frac{1}{3} = 0,333$$

Figura 76 – Mini rede web - PageRank Inicial



Fonte: Autora (2024).

Matriz de Adjacência

Precisamos redistribuir os PageRanks de acordo com os arcos de saída e entrada de cada vértice. Vamos considerar a seguinte matriz de adjacência para o grafo:

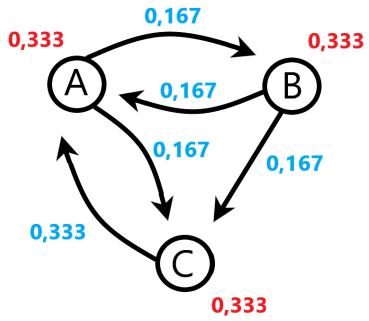
	A	B	C	Total de saídas $d^+(Y)$
A	0	1	1	$d^+(A) = 2$
B	1	0	1	$d^+(B) = 2$
C	1	0	0	$d^+(C) = 1$

Peso Inicial dos Arcos

Distribuímos o valor inicial dos vértices entre seus arcos de saída. O peso inicial de cada arco (c) será calculado por:

$$c(A) = \frac{p(A)}{2} = \frac{0,333}{2} = 0,167$$
$$c(B) = \frac{p(B)}{2} = \frac{0,333}{2} = 0,167$$
$$c(C) = \frac{p(C)}{1} = \frac{0,333}{1} = 0,333$$

Figura 77 – Mini rede web - distribuição inicial de arcos



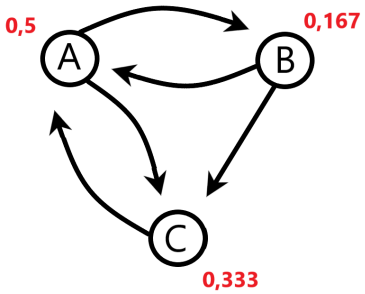
Fonte: Autora (2024).

Iteração $t = 0$

O PageRank em $t = 0$ terá o valor do somatório dos valores dos arcos de entrada de cada vértice. O vértice A recebe de B e C. O vértice B recebe de A. O vértice C recebe de A e B, logo:

$$p_0(A) = c(B) + c(C) = 0,167 + 0,333 = 0,500$$
$$p_0(B) = c(A) = 0,167$$
$$p_0(C) = c(A) + c(B) = 0,167 + 0,167 = 0,333$$

Figura 78 – Mini rede web - PageRank em $t-=0$



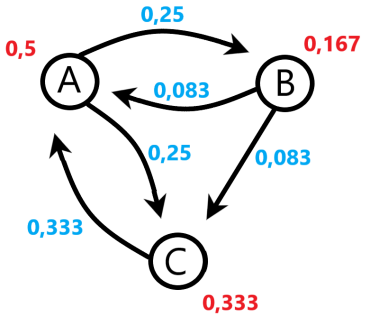
Fonte: Autora (2024).

Iteração $t = 1$

Recalculamos os pesos dos arcos:

$$c_0(A) = \frac{p_0(A)}{2} = \frac{0,500}{2} = 0,250$$
$$c_0(B) = \frac{p_0(B)}{2} = \frac{0,167}{2} = 0,083$$
$$c_0(C) = \frac{p_0(C)}{1} = \frac{0,333}{1} = 0,333$$

Figura 79 – Mini rede web - peso dos arcos em $t = 1$

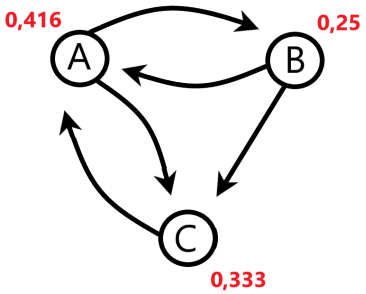


Fonte: Autora (2024).

E então os novos PageRanks:

$$p_1(A) = c_0(B) + c_0(C) = 0,083 + 0,333 = 0,416$$
$$p_1(B) = c_0(A) = 0,250$$
$$p_1(C) = c_0(A) + c_0(B) = 0,250 + 0,083 = 0,333$$

Figura 80 – Mini rede web - PageRank em $t = 1$



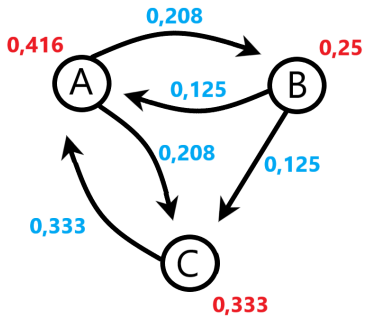
Fonte: Autora (2024).

Iteração $t = 2$

Recalculamos os pesos dos arcos:

$$c_1(A) = \frac{p_1(A)}{2} = \frac{0,416}{2} = 0,208$$
$$c_1(B) = \frac{p_1(B)}{2} = \frac{0,250}{2} = 0,125$$
$$c_1(C) = \frac{p_1(C)}{1} = \frac{0,333}{1} = 0,333$$

Figura 81 – Mini rede web - peso dos arcos em $t = 2$

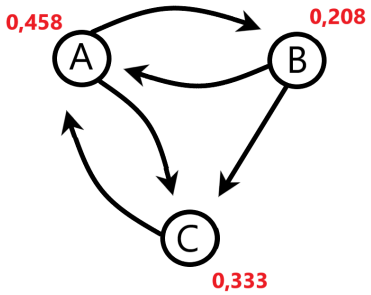


Fonte: Autora (2024).

E então os novos PageRanks:

$$p_2(A) = c_1(B) + c_1(C) = 0,125 + 0,333 = 0,458$$
$$p_2(B) = c_1(A) = 0,208$$
$$p_2(C) = c_1(A) + c_1(B) = 0,208 + 0,125 = 0,333$$

Figura 82 – Mini rede web - PageRank em $t = 2$



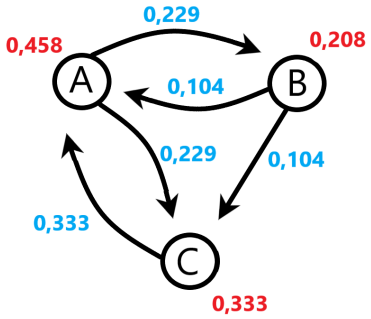
Fonte: Autora (2024).

Iteração $t = 3$

Recalculamos os pesos dos arcos:

$$c_2(A) = \frac{p_2(A)}{2} = \frac{0,458}{2} = 0,229$$
$$c_2(B) = \frac{p_2(B)}{2} = \frac{0,208}{2} = 0,104$$
$$c_2(C) = \frac{p_2(C)}{1} = \frac{0,333}{1} = 0,333$$

Figura 83 – Mini rede web - peso dos arcos em $t = 3$

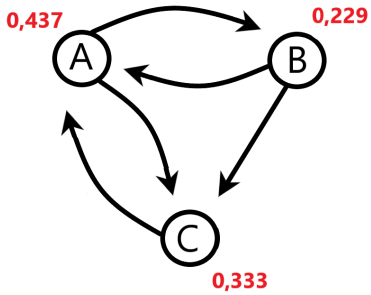


Fonte: Autora (2024).

E então os novos PageRanks:

$$p_3(A) = c_2(B) + c_2(C) = 0,104 + 0,333 = 0,437$$
$$p_3(B) = c_2(A) = 0,229$$
$$p_3(C) = c_2(A) + c_2(B) = 0,229 + 0,104 = 0,333$$

Figura 84 – Mini rede web - PageRank em $t = 3$



Fonte: Autora (2024).

Convergência

O algoritmo continua iterando até que os valores do PageRank convergam para um valor estável. Neste caso, os valores de p no algoritmo se estabilizam em $t = 52$, obtendo o PageRank

final:

$$p(A) = 0,444$$

$$p(B) = 0,222$$

$$p(C) = 0,333$$

Portanto, concluímos que a página A possui o maior PageRank e, consequentemente, aparecerá primeiro nos resultados de pesquisa. Deste modo:

- 1º lugar: página A
- 2º lugar: página C
- 3º lugar: página B

ATIVIDADE

Agora você tem um desafio! Calcule o PageRank e faça o *ranking* das páginas até a iteração $t = 3$, de acordo com o grafo fornecido na matriz de adjacência:

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{pmatrix} & 1 & & \\ 1 & & & 1 \\ & 1 & & 1 \\ 1 & & 1 & \end{pmatrix} \end{array}$$

RESPOSTA

O procedimento para o cálculo do PageRank é o mesmo descrito na prévia da atividade, a diferença é que há mais um vértice, então os PageRank inicial de cada página é $\frac{1}{4} = 0,25$.

Ao fim de 3 iterações, com aproximação para três casas decimais, devemos ter:

$$p(A) = 0,297$$

$$p(B) = 0,343$$

$$p(C) = 0,109$$

$$p(D) = 0,25$$

Desta forma, o ranking será:

- 1º lugar: página B
- 2º lugar: página A
- 3º lugar: página D
- 4º lugar: página C

6 CONSIDERAÇÕES FINAIS

A modelagem matemática através da Teoria dos Grafos oferece oportunidade de integrar situações do cotidiano na matemática, a fim de torná-la mais interessante para os estudantes. Através de problemas como a análise de redes sociais, mapas, otimização de rotas ou organização de eventos, os alunos podem vivenciar a aplicabilidade da matemática em diferentes contextos, compreendendo sua utilidade e relevância para a vida real.

A pesquisa destaca a importância de atividades desafiadoras que promovam a curiosidade, socialização entre pares, registro de procedimentos, a documentação dos resultados e a formulação de conjecturas. O professor, ao apresentar as questões propostas deve incentivar a comunicação entre os alunos, a troca de ideias e a construção de argumentos matemáticos na resolução do problema. A documentação dos resultados, através de representações gráficas, tabelas e textos, permite que os alunos organizem seu pensamento e compreendam os processos de resolução.

A introdução da Teoria dos Grafos desde o Ensino Fundamental pode contribuir para a construção de uma rede conceitual consistente, que se liga a diferentes conteúdos e contextos, o que promove a interdisciplinaridade e a formação integral do discente. A familiarização com os conceitos básicos da teoria, como vértices, arestas e conjuntos, desde cedo, permite que os alunos desenvolvam habilidades de abstração, visualização e resolução de problemas, preparando-os para a compreensão de conceitos mais complexos em níveis de ensino posteriores.

O PageRank, um tópico específico mais avançado dentro desta proposta, oferece um exemplo concreto e motivador de aplicação prática da teoria, já que atualmente grande parte da população utiliza o motor de busca do *Google* diariamente. Dessa forma, os alunos são preparados para conceitos mais complexos e também serve como ferramenta de aprendizagem para outros conceitos da Teoria dos Grafos, como a representação em matrizes, cadeias de Markov e e análise de redes, bem como introdução ao uso de algoritmos e à ciência da computação.

Em suma, é importante que os professores se familizarizem com Teoria dos Grafos e explorem seu potencial para promover uma educação matemática mais dinâmica e significativa. A pesquisa apresentada neste trabalho serve como um guia para a implementação da teoria em sala de aula, oferecendo sugestões de atividades que podem ser adaptadas às necessidades e interesses dos alunos.

REFERÊNCIAS

ALEIXO, J. A. A teoria dos grafos e as novas diretrizes curriculares para a educação básica. **Ensino da Matemática em debate**, 2020. Disponível em: <https://sca.profmat-sbm.org.br/profmat_tcc.php?id1=5475&id2=171030847>. Acesso em: 25 out. 2023.

AUSUBEL, D. P. **The Psychology of Meaningful Verbal Learning**. [S.l.]: New York: Grune & Stratton, 1963.

BALAKRISHNAN, V. K. **Theory and Problems of Graph Theory**. McGraw Hill, 1997. ISBN 0-07-005489-4. Disponível em: <https://kashanu.ac.ir/Files/GraphTheoryBalakrishnan1.pdf>. Acesso em: 09 set. 2023.

BARABáSI, A.-L. **Network Science**. Cambridge University Press, 2016. Disponível em: <https://networksciencebook.com/>. Acesso em: 03 jul. 2024.

BASSANEZI, R. C. **Modelagem Matemática: Teoria e Prática**. São Paulo: Contexto, 2015.

BERTONE, A. M. A.; BASSANEZI, R. C.; JAFELICE, R. S. da M. **Modelagem Matemática**. [S.l.]: Universidade Federal de Uberlândia, 2014.

BONDY, J. A.; MURTY, U. S. R. **Graph theory with applications**. University of Waterloo,, 1976. ISBN 0-444-19451-7. Disponível em: <https://www.zib.de/userpage/groetschel/teaching/WS1314/BondyMurtyGTWA.pdf>. Acesso em: 06 mai. 2024.

BRANDT, C. F.; BURAK, D.; KLUBER, T. E. **Modelagem matemática: perspectivas, experiências, reflexões e teorizações**. 2. ed. Editora UEPG, 2016. ISBN 978-85-7798-232-5. Disponível em: <http://books.scielo.org/id/b4zpq/epub/brandt-9788577982325.epub>. Acesso em: 12 abr. 2024.

BRASIL. **A Base Nacional Comum Curricular**. Ministério da Educação, 2017. Disponível em: <http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf>. Acesso em: 14 jun. 2024.

CARVALHO, B. M. P. S. de. Algoritmo de dijkstra. **Universidade de Coimbra**, 2008. Disponível em: <https://student.dei.uc.pt/~brunomig/cp/Artigo.pdf>. Acesso em: 15 jun. 2024.

CARVALHO, M. A. M. Bcc204 - teoria dos grafos. **Universidade Federal de Ouro Preto**, 2019. Disponível em: <http://www.decom.ufop.br/marco/site_media/uploads/bcc204/17_aula_17.pdf>. Acesso em: 12 abr. 2024.

CHARTRAND, G.; LESNIAK, L.; ZHANG, P. **Graphs & Digraphs**. 5. ed. CRC Press, 2010. ISBN 9781439826270. Disponível em: <https://books.google.co.vi/books?id=K6-FvXRlKsQC&printsec=copyright&source=gbs_pub_info_r#v=onepage&q&f=false>. Acesso em: 15 jun. 2024.

COSTA, F. de A.; IGLIORI, S. B. C. Estudo da periodicidade a partir da modelagem matemática à luz da teoria da aprendizagem significativa. **Ensino da Matemática em debate**, v. 3, n. 1, p. 58–69, 2016. Disponível em: <https://revistas.pucsp.br/index.php/pdemat/article/view/37067/25294>. Acesso em: 18 jun. 2023.

DEVLIN, K. **O Gene da Matemática**. [S.l.]: Record, 2004. ISBN 978-8501064493.

DUENHA, L. D. O problema das quatro cores. **UFMS**, 2002. Disponível em: <https://repositorio.ufms.br/bitstream/123456789/469/1/Liana%20Dessandre%20Duenha.pdf>. Acesso em: 17 jun. 2024.

EULER, L. Solutio problematis ad geometriam situs pertinentis. **Commentarii Academiae Scientiarum Petropolitanae**, v. 8, 1741. Disponível em: <https://scholarlycommons.pacific.edu/cgi/viewcontent.cgi?article=1052&context=euler-works>. Acesso em: 20 mar. 2024.

FASSARELLA, L. S. **Matemática:natureza e ensino-aprendizagem**. UFES, 2015. Disponível em: <https://blog.ufes.br/lem/files/2015/04/CFC2015_matematica-ne_texto.pdf>. Acesso em: 01 set. 2022.

GIL, A. C. **Metodologia do Ensino Superior**. 4. ed. [S.l.]: Editora Atlas, 2008.

GIUşCă, B. **The problem of the Seven Bridges of Königsberg**. Wikimedia Commons, 2005. Disponível em: <https://pt.m.wikipedia.org/wiki/Ficheiro:Konigsberg_bridges.png>. Acesso em: 20 dez. 2023.

GROSS, J. L.; YELLEN, J. **Graph Theory and Its Applications**. 2. ed. CRC Press, 2006. ISBN 978-1-58488-505-4. Disponível em: <https://books.google.com.br/books?id=-7Q_POGh-2cC&printsec=frontcover&redir_esc=y#v=onepage&q&f=false>. Acesso em: 06 mai. 2024.

INQNEST. **The Evolution of Google's Algorithms: A Journey Through Time**. Inqnest, 2024. Disponível em: <https://inqnest.com/blog/the-evolution-google-algorithm-update/>. Acesso em: 20 feb. 2024.

ISODA, M.; KATAGIRI, S. Mathematical thinking: How to develop it in the classroom. **World Scientific: Singapura**, 2012.

JURKIEWICZ, S. **Grafos – Uma Introdução**. [S.l.]: OBMEP, 2009.

LAY, D. C. Chapter 10: Finite-state markov chains. In: ____. **Linear Algebra and Its Applications**. 4. ed. Pearson, 2012. ISBN 978-0-321-38517-8. Disponível em: <https://math.dartmouth.edu/~m22f18/Downloads/Docs/Lay_chapter10_MarkovChains.pdf>. Acesso em: 03 jun. 2024.

LIBRARY, R. I. A. **Icosian Game**. OpenMind BBVA, 2019. Disponível em: <https://www.bbvaopenmind.com/en/science/mathematics/william-hamilton-the-mathematical-vandal-who-invented-board-game/>. Acesso em: 22 jun. 2024.

LOPES, F. J. A.; TáBOAS, P. Z. Euler e as pontes de königsberg. **Revista Brasileira de História da Matemática**, v. 15, n. 30, p. 23 – 32, 2015. ISSN 1519-955X. Disponível em: <https://www.rbhm.org.br/index.php/RBHM/article/view/82/56>. Acesso em: 20 mar. 2024.

MACHADO, C. A.; ROCHA, C. M. M.; MAGALHãES, E. J.; SILVA, M. das Dores Pereira e. Utilização do algoritmo pagerank para obtenção de qualidade no ranqueamento de páginas web. **UEG**, 2016. Disponível em: <file:///C:/Users/user/Downloads/9172-Texto%20do%20artigo-26989-2-10-20170823.pdf>. Acesso em: 16 set. 2023.

MALTA, G. H. S. Grafos no ensino médio: uma inserção possível. **UFRS**, 2008. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/14829/000668628.pdf?sequence=1&isAllowed=y>. Acesso em: 15 fev. 2023.

MARICONDA, P. R. Galileu e a ciência moderna. **Especiaria: Cadernos de Ciências Humanas**, v. 9, n. 16, p. 267–292, 2006. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/7245081/mod_resource/content/1/MARICONDA%2C%20P.R.%20Galileu%20e%20a%20cie%CC%82ncia%20moderna.%20Cadernos%20de%20Cie%CC%82ncias%20Humanas%20-%20Especiaria.%20v.%209%2C%20n.16%2C%20jul.dez.%2C%202006%2C%20p.%20267-292.pdf>. Acesso em: 21 ago. 2022.

MEC. **Perguntas Frequentes**. MEC - Ministério da Educação, 2022. Disponível em: <https://www.gov.br/mec/pt-br/novo-ensino-medio/informacoes/perguntas-frequentes>. Acesso em: 10 abr. 2024.

MELO, C. B. da S.; BISOGNIN, E. Modelagem matemática como proposta de itinerário formativo no novo ensino médio: uma possibilidade para o desenvolvimento de habilidades e competências. **RIPEM**, v. 11, n. 1, p. 24–36, 2021. Disponível em: <https://www.sbembrasil.org.br/periodicos/index.php/ripem/article/download/2733/1871/8253>. Acesso em: 20 ago. 2023.

MUSEUM, T. puzzle. **The Icosian Game**. The puzzle museum, 2017. Disponível em: <https://www.puzzlemuseum.com/month/picm02/200207icosian.htm>. Acesso em: 22 jun. 2024.

NETTO, B.; OSWALDO, P.; JURKIEWICZ, S. **Grafos: introdução e prática**. [S.l.]: Editora Blusher, 2009.

OLIVEIRA, A. A. de. **Novo Ensino Médio: entenda as 3 principais críticas ao modelo**. Revista Galileu, 2023. Disponível em: <https://revistagalileu.globo.com/sociedade/educacao/noticia/2023/04/novo-ensino-medio-entenda-as-3-principais-criticas-ao-modelo.ghtml>. Acesso em: 03 jul. 2024.

OLIVEIRA, V. A. de; RANGEL, S.; ARAUJO, S. A. de. Elementos de teoria dos grafos. **UNESP**, 2018. Disponível em: <https://www.ibilce.unesp.br/Home/Departamentos/MatematicaAplicada/docentes/socorro/grafos---notas-de-aula_set2018.pdf>. Acesso em: 20 jun. 2024.

ONODERA, M. M. **Texto de Divulgação: O que é Matemática?** IME - Instituto de Matemática e Estatística, 2004. Disponível em: <https://www.ime.usp.br/~masaki/mat.html>. Acesso em: 01 set. 2022.

RIBEIRO, C. M. O impacto da modelação no processo de ensino aprendizagem: uma simbiose entre a resolução de problemas e a modelação do quotidiano. **Bolema**, v. 23, n. 37, p. 977–1004, 2010. Disponível em: <https://sapientia.ualg.pt/bitstream/10400.1/12085/1/Ribeiro2010Oimpactodamodelaonoprocessodeensino%20%281%29.pdf>. Acesso em: 20 ago. 2023.

ROGERS, I. The google pagerank algorithm and how it works. **IPR Computing Ltd.**, 2002. Disponível em: <https://cs.wmich.edu/gupta/teaching/cs3310/lectureNotes_cs3310/Pagerank%20Explained%20Correctly%20with%20Examples_www.cs.princeton.edu_~chazelle_courses_BIB_pagerank.pdf>. Acesso em: 16 set. 2023.

SCHEINERMAN, E. R. **Matemática Discreta: uma introdução**. 2. ed. Cengage Learning, 2011. ISBN :978-85-221-0796-4. Disponível em: <https://tsxvpsbr.dyndns.org/arquivos/UFFS/Matem%C3%A1ticaDiscreta-UmaIntrodu%C3%A7%C3%A3o-2aEd-EdwardRScheinerman.pdf>. Acesso em: 25 abr. 2024.

SILVA, M. R. da; KRAWCZYK, N. R.; CALçADA, G. E. C. Juventudes, novo ensino médio e itinerários formativos: o que propõem os currículos das redes estaduais. **Educação e Pesquisa**, v. 49, 2023. Disponível em: <https://www.revistas.usp.br/ep/article/view/218603>. Acesso em: 05 abr 2024.

SOUZA, L. O teorema das quatro cores. **Escola Superior de Tecnologia de Viseu**, 2001. ISSN 1647-662X. Disponível em: <https://repositorio.ipv.pt/handle/10400.19/647>. Acesso em: 17 jun. 2024.

SOUZA, R. F. de. Resolução de problemas via teoria de grafos. **USP - São Carlos**, 2014. Disponível em: <https://www.teses.usp.br/teses/disponiveis/55/55136/tde-06072015-103319/publico/Dissertacao_RenatoFdeSouza_Revisada.pdf>. Acesso em: 12 abr. 2024.

VALCI. **O problema de Euler – Grafos**. Professor Valci, 2022. Disponível em: <https://valci.com.br/home/2022/07/09/o-problema-de-euler-grafos/>. Acesso em: 20 dez. 2023.

WILHELM, E. Problema do caixeiro viajante. **UFPR**, 2005. Disponível em: <https://docs.ufpr.br/~volmir/PO_II_12_TSP.pdf>. Acesso em: 17 jun. 2024.

APÊNDICE A – CÓDIGOS EM PYTHON PARA GERAR IMAGENS DE GRAFOS

A.1 Figura (9) - Grafo simples

```
# Importações
import networkx as nx
import matplotlib.pyplot as plt

#Criar um grafo
G = nx.Graph()

#Adicionar os vértices
G.add_nodes_from(["x", "y", "z", "w"])

# Adicionar arestas
G.add_edge("x", "y")
G.add_edge("x", "z")
G.add_edge("y", "z")
G.add_edge("y", "w")
G.add_edge("z", "w")

#Desenhar o Grafo
nx.draw_shell(G, with_labels=True,node_size=800,node_color= (154/255,
    198/255, 234/255))
plt.show()
```

Para desenhar um grafo usando a cor $(154, 198, 234)$ no sistema RGB, é necessário dividir cada componente de cor por 255 porque o *matplotlib* espera que os valores de cor estejam no intervalo de 0 a 1, em vez do intervalo 0 a 255 usado em muitos sistemas de cores. Isso converte a cor de $(154, 198, 234)$ para aproximadamente $(0.60, 0.78, 0.92)$, que *matplotlib* interpreta corretamente.

Deve-se fazer o mesmo procedimento ao inserir outras cores.

A.2 Figura (7) - Grafo ponderado

```
# Importações
import networkx as nx
```

```
import matplotlib.pyplot as plt

# Criando um grafo
G = nx.Graph()

# Adicionando vértices
vertices = ['a', 'b', 'c', 'd', 'e']
G.add_nodes_from(vertices)

# Adicionando arestas com pesos
arestas = [('a', 'b', 4), ('a', 'c', 2), ('a', 'd', 5),
           ('b', 'c', 1), ('b', 'd', 3), ('b', 'e', 7),
           ('c', 'd', 10), ('d', 'e', 8)]
G.add_weighted_edges_from(arestas)

# Plotando o grafo
pos = nx.planar_layout(G)  # Posicionamento dos vértices
nx.draw(G, pos, with_labels=True, node_size=1000, node_color=(154/255,
    198/255, 234/255),  font_weight="bold", edge_color="black", width=1.5)

# Adicionando rótulos de peso nas arestas
labels = nx.get_edge_attributes(G, 'weight')
nx.draw_networkx_edge_labels(G, pos, edge_labels=labels, font_color='red',
    font_size=14)

plt.show()
```

A.3 Figura (8) - Digrafo

```
# Importações
import networkx as nx
import matplotlib.pyplot as plt

#Criando um digrafo e adicionando seus vértices
G = nx.DiGraph()
G.add_nodes_from(["p", "q", "r", "s"])

# Adicionar arestas
edges = [
```

```
    ("p", "r"),
    ("p", "s"),
    ("q", "p"),
    ("q", "r"),
    ("q", "s"),
    ("s", "q")
]
G.add_edges_from(edges)

# Identificar arestas bidirecionadas
bidirecionadas = [(u, v) for u, v in G.edges() if (v, u) in G.edges()]
unidirecionadas = [(u, v) for u, v in G.edges() if (v, u) not in G.edges()]

# Gerar a disposição dos nós usando spectral_layout
pos = nx.spectral_layout(G)

#Tamanho da figura
plt.figure(figsize=(8, 6))

# Desenhar arestas unidirecionadas (retas)
nx.draw(G, pos, with_labels=True, node_size=800, node_color=(154/255,
    198/255, 234/255), edgelist=unidirecionadas, arrowsize=20)

# Desenhar arestas bidirecionadas (curvas) com uma seta
for (u, v) in bidirecionadas:
    nx.draw_networkx_edges(G, pos, edgelist=[(u, v)], connectionstyle='arc3
    , rad=0.3', arrowsize=20, node_size=800)

plt.axis("off")
plt.show()
```

A.4 Figura (11) - Digrafo ponderado

```
import networkx as nx
import matplotlib.pyplot as plt

# Define o digrafo
G = nx.DiGraph()
G.add_nodes_from(["$v_1$", "$v_2$", "$v_3$", "$v_4$", "$v_5$"])
```

```
7
8  edges = [
9      ("$v_1$", "$v_2$", 3),
10     ("$v_1$", "$v_3$", 5),
11     ("$v_2$", "$v_3$", 2),
12     ("$v_3$", "$v_1$", 7),
13     ("$v_3$", "$v_4$", 1),
14     ("$v_2$", "$v_4$", 4),
15     ("$v_4$", "$v_2$", 6)
16 ]
17 G.add_weighted_edges_from(edges)
18
19 # Identificar arestas bidirecionadas
20 bidirecionadas = [(u, v) for u, v in G.edges() if (v, u) in G.edges()]
21 unidirecionadas = [(u, v) for u, v in G.edges() if (v, u) not in G.edges()]
22
23 # Posição dos nós
24 pos = nx.planar_layout(G)
25
26 # Tamanho da Imagem
27 plt.figure(figsize=(8, 6))
28
29 # Desenhar arestas unidirecionadas (retas)
30 nx.draw(G, pos, with_labels=True, node_size=1000, node_color=(154/255,
       198/255, 234/255), edgelist=unidirecionadas, font_weight="bold",
       edge_color="black", width=1.5, arrowsize=20)
31 nx.draw_networkx_edge_labels(G, pos, edge_labels={(u, v): f"{G[u][v]['
       weight']}" for u, v in unidirecionadas}, font_color='red', font_size=14)
32
33 # Desenhar arestas bidirecionadas (curvas)
34 for (u, v) in bidirecionadas:
35     # Desenha as duas direções com curvas diferentes
36     nx.draw_networkx_edges(G, pos, edgelist=[(u, v)], connectionstyle='arc3
       , rad=0.2', width=1.5, arrowsize=20, node_size=1000)
37     nx.draw_networkx_edges(G, pos, edgelist=[(v, u)], connectionstyle='arc3
       , rad=-0.2', width=1.5, arrowsize=20, node_size=1000)
38
39     # Adiciona os rótulos dos pesos das arestas bidirecionadas
40     nx.draw_networkx_edge_labels(G, pos, edge_labels={(u, v): f"{G[u][v]['
```

```
    weight']}"}, connectionstyle='arc3, rad=0.2', font_color='red',
    font_size=14)
41  nx.draw_networkx_edge_labels(G, pos, edge_labels={(v, u): f"{G[v][u]['
    weight']}"}, connectionstyle='arc3, rad=-0.2', font_color='red',
    font_size=14)
42
43 plt.show()
```

A.5 Figura (12) - Grafo com arestas paralelas nomeadas

```
1  # Importações
2  import networkx as nx
3  import matplotlib.pyplot as plt
4
5  #Definindo o grafo
6  F = nx.DiGraph()
7
8  # Adicionando vértices
9  vertices = ['x', 'y', 'z', 'w']
10 F.add_nodes_from(vertices)
11
12 # Adicionando arestas
13 edges = [
14     ('y', 'x', '$a_1$'),
15     ('x', 'z', '$a_2$'),
16     ('x', 'w', '$a_3$'),
17     ('y', 'z', '$a_4$'),
18     ('y', 'w', '$a_5$'),
19     ('z', 'w', '$a_6$'),
20     ('x', 'y', '$a_7$'),  # Aresta paralela
21     ('z', 'y', '$a_8$'),  # Aresta paralela
22     ('w', 'z', '$a_9$')   # Aresta paralela
23 ]
24
25 F.add_weighted_edges_from(edges)
26
27 #Definindo manualmente as arestas retas
28 arc_edge = [
29     ('x', 'z', '$a_2$'),
```

```
30      ('x', 'w', '$a_3$'),
31      ('y', 'w', '$a_5$'),
32      ]
33
34 #Definindo manualmente as arestas curvas
35 arc_edges = [('x', 'y', '$a_7$'), ('y', 'x', '$a_1$'), ('y', 'z', '$a_4$'),
       ('z', 'y', '$a_8$'), ('w', 'z', '$a_9$'), ('z', 'w', '$a_6$')]
36
37 # Desenha o grafo
38 pos = nx.circular_layout(F)  # Posição dos nós
39
40 #Definindo os nomes das arestas
41 weights = {(u, v): weight for u, v, weight in arc_edge}
42 weights_2 = {(u, v): weight for u, v, weight in arc_edges}
43
44 #Tamanho da Figura
45 plt.figure(figsize=(8, 6))
46
47 #Desenhando o grafo
48 nx.draw(F, pos, with_labels=True, node_size=1000, node_color=(154/255,
     198/255, 234/255), edgelist=arc_edge, font_weight="bold", edge_color="
     black", width=1.5,  arrowsize=0.1)
49 #Inserindo os rótulos das arestas retas
50 nx.draw_networkx_edge_labels(F, pos, edge_labels=weights, font_color
     =(36/255, 36/255, 0.628/255), font_size=14, label_pos=0.4)
51
52 # Desenhar as arestas curvas específicas
53 nx.draw_networkx_edges(F, pos, node_size=1000, edgelist=arc_edges,
     connectionstyle='arc3, rad=0.2', width=1.5,  arrowsize=0.1)
54 #Inserindo os rótulos das arestas curvas
55 nx.draw_networkx_edge_labels(F, pos, edge_labels=weights_2, connectionstyle
     ='arc3, rad=0.2', font_color=(36/255, 36/255, 0.628/255), font_size=14)
56
57 plt.show()
```

Neste código o tamanho das setas foi alterado para 0.1, deste modo elas ficam menores que a espessura da aresta, portanto não são visíveis.

Para exibir um digrafo nas mesmas condições, basta alterar o tamanho das setas.

A.6 Figura (17) - Arestas específicas coloridas

```
# Importações
import networkx as nx
import matplotlib.pyplot as plt

#Definir o grafo
D = nx.Graph()
D.add_nodes_from(["a", "b", "c", "d", "e", "f", "g", "h", "i"])

# Adicionar arestas
D.add_edge("a", "b")
D.add_edge("a", "e")
D.add_edge("b", "c")
D.add_edge("b", "f")
D.add_edge("c", "f")
D.add_edge("c", "g")
D.add_edge("c", "d")
D.add_edge("d", "f")
D.add_edge("d", "g")
D.add_edge("d", "i")
D.add_edge("e", "f")
D.add_edge("e", "h")
D.add_edge("f", "g")
D.add_edge("g", "h")
D.add_edge("g", "i")
D.add_edge("g", "i")
D.add_edge("h", "i")

# Definir arestas que precisam ser coloridas de vermelho
red_edges = [("b", "c"), ("b", "f"), ("e", "f"), ("e", "h"), ("h", "i"), ("
    d", "i"), ("c", "d")]

# Gerar a disposição dos nós usando spring_layout
pos = nx.spring_layout(D, seed=6)

# Criar uma lista de cores para as arestas
edge_colors = ["red" if edge in red_edges or tuple(reversed(edge)) in
    red_edges else "black" for edge in D.edges()]
```

```
36
37 #Desenhar o grafo
38 nx.draw(D, pos, with_labels=True,node_size=1000,node_color= (154/255,
       198/255, 234/255),font_weight="bold", width=1.5, edge_color=edge_colors)
39 plt.show()
```

A.7 Figura (65) - Personalizar as posições dos vértices de maneira circular

```
1  # Importações
2  import networkx as nx
3  import matplotlib.pyplot as plt
4  import numpy as np
5
6  # Definição da lista de adjacências
7  adj_list = {
8      'Alice': ['Bernardo','Bernardo'],
9      'Bernardo': ['Alice',  'Clara', 'Érica'],
10     'Clara': ['Alice', 'Bernardo'],
11     'Daniel': ['Érica'],
12     'Érica': ['Bernardo',  'Daniel']
13 }
14
15 # Criando um grafo
16 G = nx.Graph()
17
18 # Adicionando vértices e arestas ao grafo
19 for person, friends in adj_list.items():
20     G.add_node(person)
21     for friend in friends:
22         G.add_edge(person, friend)
23
24 # Definindo cores para os vértices
25 node_colors = {
26     'Alice': (245/255,99/255,81/255),
27     'Bernardo': (45/255,163/255,247/255),
28     'Clara': (80/255,187/255,87/255),
29     'Daniel': (213/255, 105/255, 236/255),
30     'Érica': 'orange'
31 }
```

```
32
33 # Definindo a ordem dos nós no layout circular
34 order = ['Daniel','Érica','Bernardo','Alice', 'Clara']
35
36 # Criando um layout circular com a ordem especificada
37 angle_step = 2 * np.pi / len(order)
38 pos = {node: (np.cos(i * angle_step), np.sin(i * angle_step)) for i, node
       in enumerate(order)}
39
40 # Desenhando o grafo com as cores definidas
41 plt.figure(figsize=(8, 8))
42 nx.draw(G, pos, with_labels=True, node_color=[node_colors[node] for node in
        G.nodes()],
43         node_size=4000, font_size=12, font_color='black', font_weight='bold
       ', edge_color='black', width=3)
44
45 # Adicionando o círculo para representar a mesa circular
46 circle = plt.Circle((0, 0), 1.05, color='gray', fill=False)
47 plt.gca().add_artist(circle)
48
49 # Ajustando limites e exibindo o gráfico
50 plt.xlim(-1.2, 1.2)
51 plt.ylim(-1.2, 1.2)
52 plt.axis('off')  # Removendo os eixos
53 plt.show()
```

A.8 Figura (67) - Personalizar as posições dos vértices de maneira específica

```
1 # Importações
2 import networkx as nx
3 import matplotlib.pyplot as plt
4
5 # Criando um grafo
6 G = nx.Graph()
7
8 # Adicionando vértices
9 vertices = ["PM", "TA", "TU", "PN", "MC", "NA", "ST", "LT", "PA", "MA", "SF
      ", "X"]
10 G.add_nodes_from(vertices)
```

```
11
12 # Adicionando arestas com pesos
13 arestas = [
14     ("NA", "LT", 47),
15     ("SF", "NA", 145),
16     ("MA", "SF", 80),
17     ("PA", "MA", 158),
18     ("PM", "TA", 15),
19     ("PM", "NA", 104),
20     ("TA", "TU", 16),
21     ("TA", "PN", 42),
22     ("PN", "MC", 39),
23     ("MC", "X", 75),
24     ("X", "ST", 57),
25     ("ST", "LT", 30),
26     ("TU", "ST", 40),
27     ("PA", "X", 16),
28     ("PA", "X", 16),
29 ]
30
31 # Separando as arestas em duas listas: laranja e preto
32 arestas_laranja = [("NA", "LT", 47), ("SF", "NA", 145), ("MA", "SF", 80), (
       "PA", "MA", 158)]
33 arestas_preto = [edge for edge in arestas if edge not in arestas_laranja]
34
35 # Adicionando as arestas ao grafo
36 G.add_weighted_edges_from(arestas_preto)
37 G.add_weighted_edges_from(arestas_laranja)
38
39 # Definindo as posições dos vértices na imagem
40 pos = {
41     "PM": (0.0, 1.0),
42     "TA": (0.2, 0.85),
43     "TU": (0.5, 0.85),
44     "PN": (-0.1, 0.7),
45     "MC": (0.5, 0.7),
46     "NA": (0.8, 1.0),
47     "ST": (0.9, 0.85),
48     "LT": (1.2, 0.85),
```

```
    "PA": (1.0, 0.7),
    "MA": (1.5, 0.7),
    "SF": (1.4, 1.0),
    "X": (1.1, 0.77)
}

# Plotando o grafo
nx.draw(G, pos, with_labels=True, node_size=1000, node_color=[(154/255,
    198/255, 234/255)], font_weight="bold", edge_color="black", width=1.5)

# Desenhando as arestas laranja
nx.draw_networkx_edges(G, pos, edgelist=arestas_laranja, edge_color="orange
    ", width=1.5)

# Adicionando rótulos de peso nas arestas
labels = nx.get_edge_attributes(G, 'weight')
nx.draw_networkx_edge_labels(G, pos, edge_labels=labels, font_color='red',
    font_size=14)

plt.show()
```