UNIVERSIDADE FEDERAL DO CARIRI
CENTRO DE CIÊNCIAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA EM REDE NACIONAL

CARMOS FERNANDES DA COSTA

O USO DO LaTeX NA CONSTRUÇÃO E ANIMAÇÃO DE FIGURAS GEOMÉTRICAS COMO AUXÍLIO NO ENSINO DE GEOMETRIA

JUAZEIRO DO NORTE
2022

CARMOS FERNANDES DA COSTA

O USO DO LaTeX NA CONSTRUÇÃO E ANIMAÇÃO DE FIGURAS GEOMÉTRICAS COMO AUXÍLIO NO ENSINO DE GEOMETRIA

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Matemática em Rede Nacional do Centro de Ciências e Tecnologia da Universidade Federal do Cariri, como parte dos requisitos necessários à obtenção do título de Mestre em Matemática.

Orientador: Prof. Dr. Francisco Pereira Chaves

JUAZEIRO DO NORTE
2022

Dados Internacionais de Catalogação na Publicação.
Universidade Federal do Cariri.
Sistema de Bibliotecas

C837u Costa, Carmos Fernandes da.
O uso do LaTeX na construção e animação de figuras geométricas como auxílio no ensino de geometria / Carmos Fernandes da Costa. – 2022.
xvii, 213 f.: il. color. 30 cm.

Dissertação (Mestrado) - Universidade Federal do Cariri, Centro de Ciências e Tecnologia, Programa de Pós-Graduação em Matemática em Rede Nacional - PROFMAT, Juazeiro do Norte, 2022.
Orientação: Prof. Dr. Francisco Pereira Chaves.

1. Geometria analítica - animação. 2. LaTeX. 3. Áreas de figuras planas. I. Título.

CDD 516.3

Bibliotecária: Glacinésia Leal Mendonça
CRB 3/ 925

CARMOS FERNANDES DA COSTA

O USO DO LaTeX NA CONSTRUÇÃO E ANIMAÇÃO DE FIGURAS GEOMÉTRICAS COMO AUXÍLIO NO ENSINO DE GEOMETRIA

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Matemática em Rede Nacional do Centro de Ciências e Tecnologia da Universidade Federal do Cariri, como parte dos requisitos necessários à obtenção do título de Mestre em Matemática.

Aprovado em 25 de agosto de 2022.

BANCA EXAMINADORA

gov.br Documento assinado digitalmente
FRANCISCO PEREIRA CHAVES
Data: 06/09/2022 16:09:27-0300
Verifique em https://verificador.iti.br

Prof. Dr. Francisco Pereira Chaves
Orientador

gov.br Documento assinado digitalmente
FRANCISCO DE ASSIS BENJAMIM FILHO
Data: 08/09/2022 13:45:17-0300
Verifique em https://verificador.iti.br

Prof. Dr. Francisco de Assis Benjamim Filho
UFCA

gov.br Documento assinado digitalmente
VICENTE HELANO FEITOSA BATISTA SOBRI
Data: 09/09/2022 15:26:42-0300
Verifique em https://verificador.iti.br

Prof. Dr. Vicente Helano Feitosa Batista Sobrinho
UFCA

João Vitor da Silva

Prof. Dr. João Vitor da Silva
UNICAMP

Para minha mãe e meu pai.

Agradecimentos

A Deus, pela dádiva da vida.

Aos meus familiares: minha mãe Luiza Fernandes da Costa, meu pai Francisco Fernandes Filho e meus irmãos, pelo apoio e incentivo na progressão de meus estudos.

À minha esposa e companheira, Antônia Larissa Paulo Souza, pelo apoio emocional e incentivo nos momentos difíceis.

Às minhas amigas, Genoveva e Maria do Socorro, pela inspiração e apoio na produção desta dissertação.

A todos os meus colegas de curso, em especial aos meus colegas e amigos Thiago Alan e José Alves, que se propuseram a estudar comigo aos finais de semana, possibilitando a troca de experiências e proporcionando uma aprendizagem mais significativa.

A todos os meus professores e professoras, em especial ao meu orientador Francisco Pereira Chaves, pela paciência e orientações na produção desta dissertação.

Ao PROFMAT, por oportunizar o curso de mestrado.

À UFCA, pelo apoio.

Usar recursos digitais não é garantia de aprendizagem. A tecnologia é mais uma ferramenta, que precisa do talento do professor, interesse do aluno e o acompanhamento da família!

Prof. Rogério Joaquim

RESUMO

Geometria é notoriamente um ramo da Matemática que é essencialmente visual. Um texto, se bem redigido e com figuras bem desenhadas, certamente contribui para uma melhor compreensão da mensagem que se pretende transmitir, ainda mais quando o interlocutor é um estudante, que está no processo de construção do conhecimento, como é o caso dos alunos do Ensino Médio. Este trabalho propõe a utilização do LaTeX para produção de figuras e animações que auxiliem os professores do Ensino Médio no ensino de Geometria Plana. Exploramos de forma descritiva a produção das animações que fazem uso das macros pertencentes aos pacotes *TikZ*, *tkz-euclide* e *Animate*. Apresentamos uma Proposta de Sequência Didática que explora animações produzidas com tais pacotes, mostrando como utilizá-las efetivamente, e que visa auxiliar os professores nas demonstrações das fórmulas das áreas das principais figuras planas.

Palavras-chave: Animação. LaTeX. Áreas de figuras planas.

ABSTRACT

Geometry is notoriously a branch of Mathematics that is essentially visual. A text, if well written and with well-drawn figures, certainly contributes to a better understanding of the message that is intended to convey, even more so when the reader is a student during the process of building his knowledge, as is the case of high school students. This work proposes the use of LaTeX to produce figures and animations that help high school teachers in tutoring plane geometry. We descriptively explore the production of animations that make use of macros belonging to the packages *TikZ*, *tkz-euclide* and *Animate*. We present a Proposal for a Didactic Sequence that explores animations produced with such packages, where it is shown how to use them effectively and which aims to assist teachers in proving formulas for computing the area of some well-known plane figures.

Keywords: Animation. LaTeX. Areas of flat figures.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

O surgimento da Geometria data de cerca de 3.000 a.C e está associado ao cálculo de extensões de terras às margens do rio Nilo. Por conta das cheias do rio encobrir parte dos terrenos, para compensar o pagamento de impostos, o Rei Sesóstris ordenava que se calculasse as dimensões das terras não submersas a fim de diminuir proporcionalmente a quantia de impostos paga à coroa pelos proprietários [1]. Esta hipótese é reforçada pela origem da própria palavra Geometria. De acordo com Silva [2], esse nome é proveniente do grego e é a junção de "geo", que significa terra, e "metria", derivada da palavra "métron", que significa medir. Portanto, é plausível relacionar a Geometria com o ato de medir a terra.

Mesmo tendo surgido da necessidade de medir terras às margens do Nilo, a Geometria não se restringiu apenas a solucionar esse problema. Com o passar dos anos, a humanidade acumulou conhecimentos sobre esta parte da Matemática que se mostraram eficientes para as mais diversas situações. A exemplo disso, temos as construções arquitetônicas antigas como o Partenon, na Grécia, e as Pirâmides de Gizé, no Egito, que impressionam por tamanha beleza e que certamente não seriam construídas sem os conhecimentos geométricos daquela época.

Nos registros antigos, são observadas situações em que há, simultaneamente, representações geométricas e algébricas, de forma rudimentar e com a escrita disponível à época. A Figura 1.1 mostra dois exemplos disso, onde estão representados pelos babilônios, por volta de 1.800-1.600 a.C., um círculo e um trapézio que, de acordo com Carvalho e Roque [1], representam os cálculos das áreas destas duas figuras. Atualmente a Geometria é denominada como a parte da Matemática que se ocupa em estudar as formas e suas propriedades, e que, para isso, faz uso de procedimentos algébricos que evidentemente se relacionam diretamente com as formas estudadas.

Figura 1.1: Círculo e trapézio na Babilônia.

Fonte: Carvalho e Roque [1].

É natural que, quando nos propomos a resolver um problema de Geometria, imaginemos o desenho que representa a situação. Dependendo da complexidade do problema, é mais viável que se faça um desenho para que tenhamos uma visão mais ampla da situação. Esta possibilidade de interpretação visual dos problemas geométricos é de suma importância para o estudo da Geometria, pois evita a fadiga mental de pensar, simultaneamente, no desenho da situação e em estratégias de solução. Muitas vezes, o simples fato de desenhar já nos permite pensar em inúmeras estratégias de solução. Portanto, o uso de figuras em problemas geométricos auxiliam na busca por estratégias de solução e também contribuem para a compreensão dos procedimentos adotados.

Com o desenvolvimento das tecnologias atuais, como, computadores e *softwares*, a representação de problemas geométricos tornou-se mais precisa. Dentre estes programas, encontra-se o LaTeX, que, embora não tenha sido criado com a finalidade de produzir desenhos geométricos, possui atualmente pacotes de comandos que permitem desenhar com perfeição. Dentre estes pacotes, em relação à Geometria Plana, destacamos os pacotes `TikZ` e `tkz-euclide`, que, utilizados em conjunto, possibilitam a produção de figuras de alta qualidade.

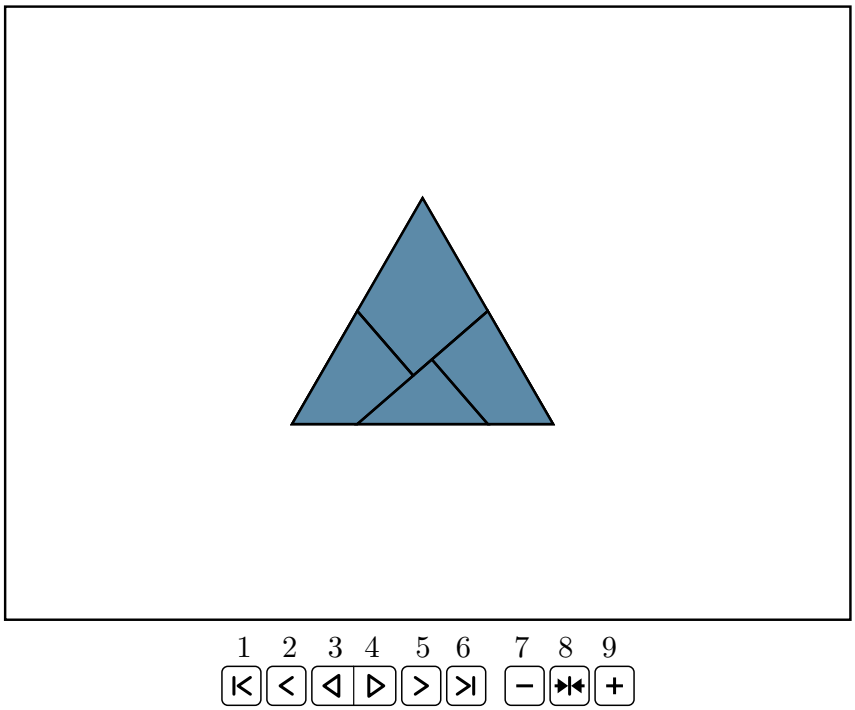
Quando o assunto é a produção de animações de figuras geométricas, um *software* que nos vem à mente é o *Geogebra* (para informações sobre este software consulte [3]). No entanto, o que muitos usuários do LaTeX, até mesmo professores de Matemática que usam esse programa no cotidiano, não sabem é que o LaTeX também permite a criação de animações de figuras. Essas animações podem ser feitas por meio de um pacote chamado `animate`. A vantagem de utilizar o LaTeX é que as animações são produzidas em arquivos no formato PDF, que proporciona facilidade no compartilhamento do arquivo e na reprodução das animações.

Juntando a importância de termos figuras em auxílio na compreensão da Geometria com a possibilidade de produção de animações no LaTeX, produzimos este trabalho, o qual tem o objetivo de explorar os pacotes supracitados, propondo o uso do LaTeX para a produção de animações que auxiliem professores do Ensino Médio

no ensino de Geometria, explorando de forma descritiva a utilização dos pacotes `TikZ`, `tkz-euclide` e `animate` para a produção de figuras e animações.

Este trabalho possui várias exemplos de animações produzidas no LaTeX, as quais permitem interação por meio de botões impressos no PDF. Por isso, para que o leitor tenha a experiência completa em sua leitura, é aconselhável que o faça por meio de um computador que tenha instalado leitores desse tipo de arquivo, de preferência um dos seguintes leitores: *Adobe Acrobat Reader*, *KDE Okular*, *PDF-XChange* ou *Foxit Reader*. Alertamos que, no sistema operacional Windows, as animações funcionam em qualquer um desses leitores de PDF, mas para o sistema Linux, funcionam apenas com o leitor *KDE Okular*. Para mais informações sobre a funcionalidade das animações, consulte a Seção 4.3. Para exemplificar a utilização de animações, considere a Figura 1.2.

Figura 1.2: O problema de Haberdasher.



Fonte: Elaborada pelo autor com base em Neves [4].

Sempre que uma figura deste trabalho apresentar os botões representados na Figura 1.2 pelos números 1, 2, 3, 4, 5, 6, 7, 8 e 9, significa que tal figura representa uma animação produzida no LaTeX e que, se o leitor estiver realizando a leitura deste trabalho por meio de um computador, será possível a interação com esses botões. O botão representado por 1, tem a função de voltar a animação diretamente para o início; 2 tem a função de voltar a animação um *frame* por vez; 3 reproduz a animação no sentido inverso, enquanto o botão 4 reproduz no sentido correto; 5 tem a função

semelhante a de 2, porém no sentido oposto; 6 leva a animação diretamente para o final; 7 diminui a velocidade da animação; 8 retorna a animação para a velocidade padrão; e 9 aumenta a velocidade. Caso o leitor pressione um dos botões 3 ou 4, aparecerá no lugar dos mesmos um novo botão que tem a função de pausar a animação, sendo possível retomá-la pressionando novamente o botão 3 ou 4.

Para alcançarmos o objetivo proposto, organizamos nossa pesquisa em sete capítulos, sendo o primeiro constituído por esta apresentação, na qual apresentamos a motivação, objetivo esperado com o seu desenvolvimento e sua estrutura.

O segundo capítulo se destina à apresentação e discussão dos assuntos de Geometria que fazem parte do currículo do Ensino Médio, tomando como referência a Base Nacional Comum Curricular (BNCC) [5] e a Matriz de Referência do Exame Nacional do Ensino Médio (Enem) [6].

Apresentamos no terceiro capítulo resultados e provas referentes a alguns assuntos de Geometria Plana, resultados esses que, além de imprescindíveis para a formação matemática do jovem do Ensino Médio, serviram de embasamento teórico para as orientações apresentadas nos Capítulos 5 e 6.

Já o quarto capítulo é dedicado ao programa LaTeX, destacando sua história e a importância para a produção de textos matemáticos. É apresentado um tutorial sobre a instalação dos programas necessários para o funcionamento correto do LaTeX além de orientações sobre o manuseio de comandos básicos do programa.

No quinto capítulo encontram-se explicações e orientações sobre como utilizar os pacotes `TikZ`, `tkz-euclide` e `animate` do LaTeX para o desenho de figuras geométricas planas e na construção de animações.

O sexto capítulo se destina à apresentação de uma proposta de sequência didática objetivando às demonstrações das fórmulas das áreas do quadrado, retângulo, paralelogramo, triângulo, losango, trapézio, triângulo equilátero, hexágono regular e do círculo, tomando como auxílio animações construídas em LaTeX.

Por fim, o sétimo, e último capítulo, é destinado às considerações finais.

Capítulo 2

A Geometria no Currículo do Ensino Médio

Neste capítulo, apresentaremos os assuntos de Geometria que estão presentes no currículo do Ensino Médio, bem como os assuntos, também da geometria, que figuram na Matriz de Referência do Exame Nacional do Ensino Médio (Enem), a qual é a mais importante avalição realizada pelo governo brasileiro e que várias universidades utilizam como meio para selecionar seus discentes.

2.1 A Geometria na Base Nacional Comum Curricular

A Base Nacional Comum Curricular (BNCC) é um documento normativo que estipula o conjunto de conhecimentos essenciais que todos os alunos brasileiros têm que desenvolver ao longo de suas carreiras escolares na Educação Básica. A BNCC foi construída pelo Conselho Nacional de Educação (CNE) e contou com a colaboração de vários especialistas em educação, entidades representativas das três esferas (federal, estadual e municipal), professores, escolas e da população em geral. Sendo por isso, considerada um documento democraticamente construído.

O processo de construção da BNCC foi demorado e se iniciou indiretamente com a promulgação da Constituição da República Federativa do Brasil de 1988, a qual afirma, em seu artigo 210, que "serão fixados conteúdos mínimos para o ensino fundamental, de maneira a assegurar formação básica comum e respeito aos valores culturais e artísticos, nacionais e regionais". Três décadas depois a versão final da BNCC foi homologada pelo Ministério da Educação em 14 de dezembro de 2018, que além de fixar os conteúdos mínimos para o Ensino Fundamental, também fixa conteúdos mínimos para o Ensino Infantil e Ensino Médio. A linha de tempo para se chegar ao documento final pode ser encontrada em Brasil [7].

A BNCC descreve *competências* como sendo a mobilização de conhecimentos, habilidades, atitudes e valores que um indivíduo possui para resolver demandas da vida cotidiana, do exercício da cidadania e do mundo do trabalho. Essas competências são divididas em *gerais* e *específicas*.

As competências gerais se referem às aprendizagens essenciais para a formação dos discentes. Essas competências destinam-se não apenas aos conteúdos abordados em cada componente curricular, mas também à formação de atitudes e valores éticos e morais que os discentes devem desenvolver ao longo das três etapas da Educação Básica: Educação Infantil, Ensino Fundamental e Ensino Médio.

As competências específicas se remetem ao conjunto de competências que os estudantes devem desenvolver em cada componente curricular para que se apropriem das competências gerais. Essas competências se relacionam com *habilidades*, as quais contêm os pré-requisitos para o desenvolvimento das competências. Segundo Brasil [5], embora cada habilidade esteja associada a determinada competência, isso não significa que ela não contribua para o desenvolvimento de outras, ou seja, as habilidades podem contribuir para o desenvolvimento de outras competências mesmo que não estejam especificamente relacionadas com elas, podendo até mesmo contribuir para competências de diferentes componentes curriculares.

A Tabela 2.1 apresenta as competências específicas constantes na BNCC, referentes ao componente curricular Matemática e suas Tecnologias do Ensino Médio. Na Tabela 2.2, listamos as habilidades associadas a essas competências que estão relacionadas diretamente a algum conhecimento de Geometria do Ensino Médio. As demais habilidades podem ser encontradas em Brasil [5].

Tabela 2.1: Competências específicas de Matemática do Ensino Médio.

COMPETÊNCIAS ESPECÍFICAS	DESCRIÇÃO DAS COMPETÊNCIAS ESPECÍFICAS
Competência Específica 1	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.
Competência Específica 2	Propor ou participar de ações para investigar desafios do mundo contemporâneo e tomar decisões éticas e socialmente responsáveis, com base na análise de problemas sociais, como os voltados a situações de saúde, sustentabilidade, das implicações da tecnologia no mundo do trabalho, entre outros, mobilizando e articulando conceitos, procedimentos e linguagens próprios da Matemática.
Competência Específica 3	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.
Competência Específica 4	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.
Competência Específica 5	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.

Fonte: Brasil [5].

Tabela 2.2: Habilidades de Matemática e suas Tecnologias do Ensino Médio que apresentam conhecimentos geométricos (Parte 1).

ANO	CÓD. HAB.	DESCRIÇÃO DAS HABILIDADES DE MATEMÁTICA E SUAS TECNOLOGIAS
1º, 2º, 3º	EM13MAT201	Propor ou participar de ações adequadas às demandas da região, preferencialmente para sua comunidade, envolvendo medições e cálculos de perímetro, de área, de volume, de capacidade ou de massa.
1º, 2º, 3º	EM13MAT307	Empregar diferentes métodos para a obtenção da medida da área de uma superfície (reconfigurações, aproximação por cortes etc.) e deduzir expressões de cálculo para aplicá-las em situações reais (como o remanejamento e a distribuição de plantações, entre outros), com ou sem apoio de tecnologias digitais.
1º, 2º, 3º	EM13MAT105	Utilizar as noções de transformações isométricas (translação, reflexão, rotação e composições destas) e transformações homotéticas para construir figuras e analisar elementos da natureza e diferentes produções humanas (fractais, construções civis, obras de arte, entre outras).
1º, 2º, 3º	EM13MAT308	Aplicar as relações métricas, incluindo as leis do seno e do cosseno ou as noções de congruência e semelhança, para resolver e elaborar problemas que envolvem triângulos, em variados contextos.
1º, 2º, 3º	EM13MAT309	Resolver e elaborar problemas que envolvem o cálculo de áreas totais e de volumes de prismas, pirâmides e corpos redondos em situações reais (como o cálculo do gasto de material para revestimento ou pinturas de objetos cujos formatos sejam composições dos sólidos estudados), com ou sem apoio de tecnologias digitais.
1º, 2º, 3º	EM13MAT504	Investigar processos de obtenção da medida do volume de prismas, pirâmides, cilindros e cones, incluindo o princípio de Cavalieri, para a obtenção das fórmulas de cálculo da medida do volume dessas figuras.

(Continua.)

Fonte: Brasil [5].

Tabela 2.3: Habilidades de Matemática e suas Tecnologias do Ensino Médio que apresentam conhecimentos geométricos (Parte 2).

(Continuação.)

ANO	CÓD. HAB.	DESCRIÇÃO DAS HABILIDADES DE MATEMÁTICA E SUAS TECNOLOGIAS
1º, 2º, 3º	EM13MAT505	Resolver problemas sobre ladrilhamento do plano, com ou sem apoio de aplicativos de geometria dinâmica, para conjecturar a respeito dos tipos ou composição de polígonos que podem ser utilizados em ladrilhamento, generalizando padrões observados.
1º, 2º, 3º	EM13MAT506	Representar graficamente a variação da área e do perímetro de um polígono regular quando os comprimentos de seus lados variam, analisando e classificando as funções envolvidas.
1º, 2º, 3º	EM13MAT509	Investigar a deformação de ângulos e áreas provocada pelas diferentes projeções usadas em cartografia (como a cilíndrica e a cônica), com ou sem suporte de tecnologia digital.

Fonte: Brasil [5].

Cada habilidade possui um código, o qual está descrito na segunda coluna das Tabelas 2.2 e 2.3, onde as duas primeiras letras EM indicam que essa habilidade deve ser desenvolvida no Ensino Médio, os dois dígitos 13 indicam que a habilidade pode ser apresentada aos alunos em qualquer ano do Ensino Médio (1º, 2º ou 3º), a sigla MAT refere-se ao componente curricular Matemática e sua Tecnologias, o primeiro dígito após a sigla MAT relaciona a habilidade com uma das competências específicas apresentadas na Tabela 2.1, e os dois últimos dígitos indicam simplesmente a posição da habilidade dentre as que compõem cada competência específica. Essa última numeração não representa uma ordem ou hierarquia esperada das aprendizagens. Cabe aos sistemas e escolas definirem a progressão das aprendizagens, em função de seus contextos locais [5]. Isso significa que a BNCC não especifica em qual ano (série) o estudante deve iniciar o contato com cada habilidade, ficando a cargo das escolas decidirem quando se dará o contato do aluno com as habilidades, de acordo com suas especificidades.

É importante observar que a BNCC não traz uma lista extensa de conteúdos a serem trabalhados em sala de aula, em vez disso, apresenta um conjunto de habilidades que para serem desenvolvidas pelos estudantes, necessitam que os alunos aprendam conteúdos implicitamente relacionados com as mesmas. Por exemplo,

para desenvolver a habilidade EM13MAT506, ou seja, para que o aluno represente graficamente a variação da área e do perímetro de um polígono regular quando o comprimento de seus lados variam e classifique as funções envolvidas, é esperado que ele saiba calcular áreas e perímetros de polígonos regulares, tenha consciência de como representar informações através de gráficos, conheça e saiba diferenciar as funções que venham a aparecer, etc.

As habilidades referentes à componente Matemática e suas Tecnologias são apresentadas no texto da BNCC de duas formas diferentes. A princípio são listadas de acordo com as competências específicas descritas na Tabela 2.1 tendo como critério de agrupamento a relação que as habilidades possuem com tais competências e, posteriormente, são organizadas em três grupos: Números e Álgebra; Geometria e Medidas; e Probabilidade e Estatística, e juntas somam 43, onde nota-se que em 4 das 9 habilidades relacionadas à Geometria, há em seu texto menção a algum tipo de auxílio tecnológico, representando um percentual de mais de 44%, e quando considera-se todas as 43 habilidades, verifica-se que 17 fazem alusão ao uso de tecnologias, o que representa um percentual aproximado de 40%. Isso reforça a importância da elaboração de trabalhos acadêmicos que relacionem recursos tecnológicos com o ensino de Geometria, tanto como auxílio ao estudante quanto ao professor.

2.2 A Geometria no Enem

De acordo com Brasil [8], o Exame Nacional do Ensino Médio (Enem) foi criado em 1998 com o objetivo de avaliar o desempenho escolar dos alunos concluintes do Ensino Médio, e em 2009 passou a ser utilizado como mecanismo de acesso à educação superior, principalmente, por meio do Sistema de Seleção Unificada (Sisu), do Programa Universidade para Todos (ProUni) e do Fundo de Financiamento Estudantil (Fies).

O fato de o Enem ter passado a ser uma das formas de acesso ao ensino superior, certamente impacta nos assuntos trabalhados em sala de aula pelos professores da Educação Básica, levando a uma maior atenção a assuntos mais recorrentes no exame ou que estejam presentes na Matriz de Referência do Enem [6].

A Matriz de Referência do Enem é um documento que tem por objetivo indicar as competências e habilidades nas quais um candidato se submeterá a teste, bem como servir de orientação para a elaboração dos itens da prova. No que diz respeito aos assuntos de Matemática que têm estreita relação com a Geometria, encontramos na Matriz de Referência as Competências e Habilidades listadas na Tabela 2.4.

Tabela 2.4: Competências e Habilidades em Matemática e suas Tecnologias (Enem) relacionadas a conhecimentos geométricos.

Competência de área 2 - Utilizar o conhecimento geométrico para realizar a leitura e a representação da realidade e agir sobre ela.
H6 - Interpretar a localização e a movimentação de pessoas/objetos no espaço tridimensional e sua representação no espaço bidimensional.
H7 - Identificar características de figuras planas ou espaciais.
H8 - Resolver situação-problema que envolva conhecimentos geométricos de espaço e forma.
H9 - Utilizar conhecimentos geométricos de espaço e forma na seleção de argumentos propostos como solução de problemas do cotidiano.
Competência de área 3 - Construir noções de grandezas e medidas para a compreensão da realidade e a solução de problemas do cotidiano.
H14 - Avaliar proposta de intervenção na realidade utilizando conhecimentos geométricos relacionados a grandezas e medidas.
Competência de área 5 - Modelar e resolver problemas que envolvem variáveis socioeconômicas ou técnico-científicas, usando representações algébricas.
H22 - Utilizar conhecimentos algébricos/geométricos como recurso para a construção de argumentação.

Fonte: Brasil [6].

Nota-se que, assim como na BNCC, a lista de habilidades presentes na matriz de referência do Enem não explicita os assuntos matemáticos a serem abordados, porém, apresenta em anexo uma lista de objetos de conhecimento, os quais descrevem os assuntos relacionados às habilidades de cada componente curricular.

No que diz respeito à Matemática e suas Tecnologias, esses objetos são divididos em cinco tópicos, distribuídos como a seguir:

1. **Conhecimentos numéricos**: operações em conjuntos numéricos (naturais, inteiros, racionais e reais), desigualdades, divisibilidade, fatoração, razões e proporções, porcentagem e juros, relações de dependência entre grandezas, sequências e progressões, princípios de contagem.

2. **Conhecimentos geométricos**: características das figuras geométricas planas e espaciais; grandezas, unidades de medida e escalas; comprimentos, áreas e volumes; ângulos; posições de retas; simetrias de figuras planas ou espaciais; congruência e semelhança de triângulos; teorema de Tales; relações métricas nos triângulos; circunferências; trigonometria do ângulo agudo.

3. **Conhecimentos de estatística e probabilidade**: representação e análise

de dados; medidas de tendência central (médias, moda e mediana); desvios e variância; noções de probabilidade.

4. **Conhecimentos algébricos**: gráficos e funções; funções algébricas do 1.º e do 2.º graus, polinomiais, racionais, exponenciais e logarítmicas; equações e inequações; relações no ciclo trigonométrico e funções trigonométricas.

5. **Conhecimentos algébricos/geométricos**: plano cartesiano; retas; circunferências; paralelismo e perpendicularidade, sistemas de equações.

Essa lista esclarece quais assuntos podem vir a figurar nas questões da prova, e por isso merece atenção por parte dos estudantes e professores do Ensino Médio. Contudo, para uma análise mais profunda dos assuntos que estão sendo cobrados no exame, torna-se necessário avaliar os cadernos de provas das edições do Enem, pois, através deles podemos verificar como se dá a abordagem dos temas, além da possibilidade de quantificar os itens que abordam cada objeto ou habilidade.

Siqueira [9], avaliou cada item dos cadernos de provas das edições do Enem de 2009 a 2019, destacando os conteúdos que cada um aborda e os quantificou de acordo com os objetos de conhecimento.

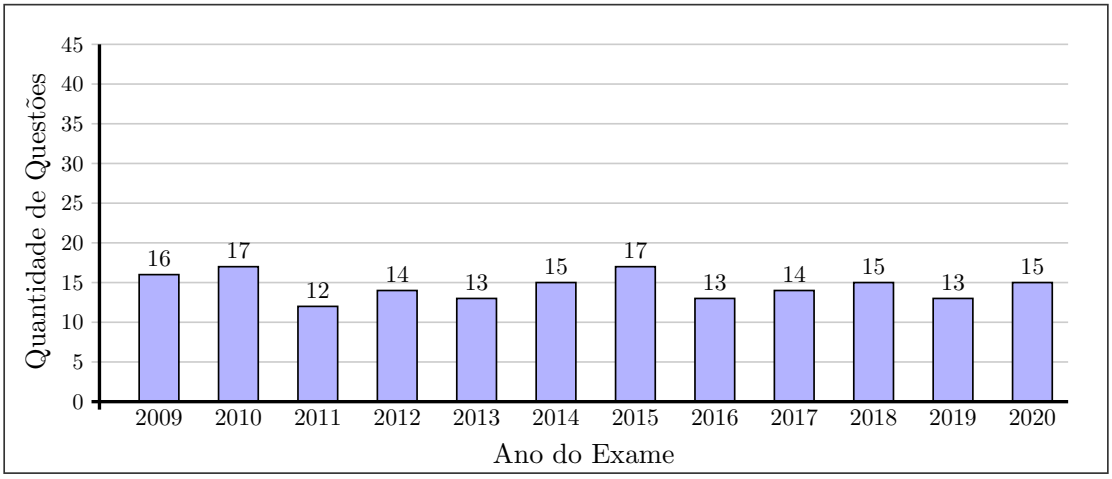
Tabela 2.5: Seleção das questões do Enem 2020 que abordam conhecimentos geométricos - Caderno Amarelo.

Questão Nº	Objeto	Conteúdo 1	Conteúdo 2	Conteúdo 3
137	Características de figuras planas e espaciais	Projeção		
138	Escala	Paralelepípedo		
145	Volume	Prisma		
147	Unidades de medida	Conversão		
150	Escala	Volume	Paralelepípedo	Conversão
152	Áreas	Retângulo	Quadrado	Conversão
154	Áreas	Retângulo		
158	Características de figuras planas e espaciais	Tronco de Pirâmide	Quadrado	Trapézio
159	Características de figuras planas e espaciais	Vista	Projeção	
168	Volume	Cilindro		
169	Áreas	Retângulo		
175	Volume	Paralelepípedo		
178	Características de figuras planas e espaciais	Triângulo	Quadrado	
179	Relações métricas no triângulo	Teorema de Pitágoras		
180	Trigonometria no ângulo agudo	Razão trogonométrica: Tangente		

Fonte: Autor.

O Gráfico 2.1 traz o quantitativo de itens que abordaram algum conhecimento geométrico desde 2009 até 2020, onde os dados de 2020 foram agregados aos dados levantados por Siqueira [9] levando em consideração a Tabela 2.5.

Gráfico 2.1: Quantitativo de questões do Enem relacionadas ao conhecimento geométrico.

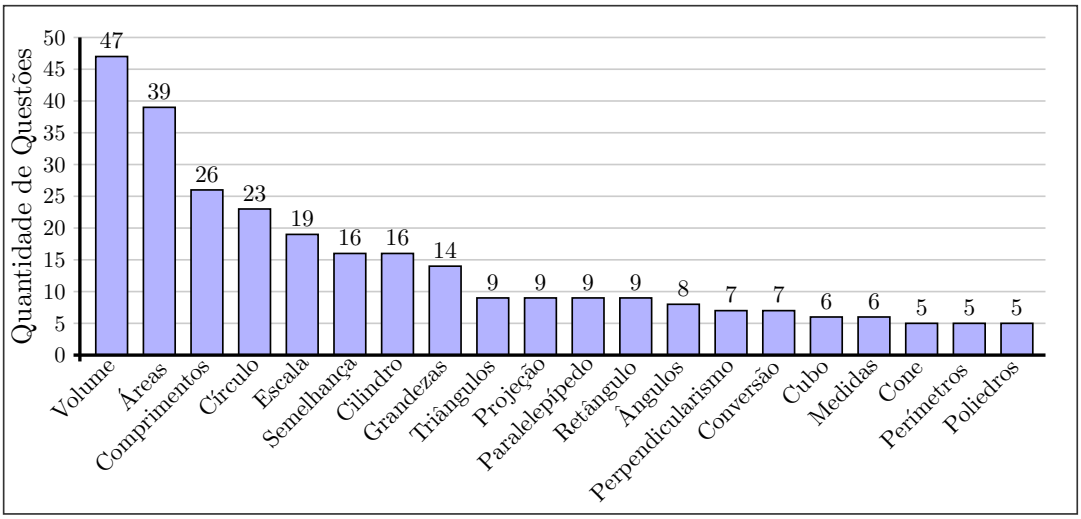


Fonte: Siqueira [9], adaptado pelo autor.

Como vemos no Gráfico 2.1, das 540 questões aplicadas nas 12 edições do Enem de 2009 a 2020, 174 abordaram conhecimentos geométricos. Em termos de porcentagem, isso representa aproximadamente 32,22%. Ao avaliarmos as provas de maneira individual, por edição, nota-se que das 45 questões, no mínimo 12, isto é, 26,66%, apresentaram conhecimentos geométricos.

Siqueira [9] constrói um *ranking* dos 20 assuntos geométricos mais abordados nas questões do Enem até o ano de 2019. O Gráfico 2.2 traz o *ranking* dos 20 assuntos de Geometria mais abordados no Enem até o ano de 2020.

Gráfico 2.2: Os 20 assuntos de Geometria com maior frequência nas provas do Enem.



Fonte: Siqueira [9], adaptado pelo autor.

É evidente o destaque que os assuntos Volumes e Áreas possuem em relação aos demais. Das 174 questões relacionadas aos objetos de conhecimentos geométricos, 39 abordaram o objeto Áreas, o que representa um percentual aproximado de 22,41%, enquanto questões relacionadas com Volumes totalizam 47, representando um percentual de 27,01%. Quando consideramos os dois assuntos, Volumes e Áreas, o percentual é de 49,44%, ou seja, das questões que abordaram algum conhecimento geométrico, quase metade se relacionaram com um desses dois objetos. Cabe ressaltar que uma questão pode abordar mais de um assunto, o que gera uma incompatibilidade com o total de questões que se relacionam com Geometria e a soma das vezes em que os objetos são abordados.

Os dados apresentados evidenciam a relevância da Geometria, em particular o assunto Áreas, no Enem e consequentemente no Currículo do Ensino Médio.

Capítulo 3

Geometria Plana

Em praticamente todo livro de Matemática, encontra-se uma grande quantidade de definições que fazem uso de conceitos apresentados previamente. Como exemplo, temos a definição de ângulos opostos pelo vértice, que são ângulos que possuem um mesmo vértice e cujos lados devem ser semirretas opostas. Nessa definição, faz-se uso de termos que necessitam de conceitos apresentados previamente, pois, para compreender o que é um ângulo oposto pelo vértice, é imperativo que se saiba o que são ângulos, lados e vértice de um ângulo e semirretas opostas.

Se tomarmos o raciocínio acima, cabe uma pergunta simples: é possível definir todos os conceitos matemáticos, em especial os de Geometria Plana? Segundo Santos e Viglioni [10], a resposta é não, pois a necessidade de definição de um conceito, fazendo o uso de outros predefinidos, nos coloca em um processo infinito, onde, para definir um termo, devemos usar outros termos, e para definir esses termos, devemos usar outros termos, e assim por diante. Sendo assim, como definir os conceitos de Geometria Plana de maneira segura e exata? A única saída é considerar que existem termos que não necessitam de definição, chamados de termos primitivos. Em se tratando da Geometria Plana, temos como conceitos primitivos, o ponto, a reta e o plano.

De acordo com Ávila [11], foi o grego Tales de Mileto, no início do século VI a.C, o primeiro matemático a se preocupar em demonstrar resultados matemáticos. Assim, foi a partir de Tales que a matemática grega foi assumindo organização propositiva logicamente ordenada, onde, para se demonstrar um dado resultado (proposição), faz-se necessário o uso de outros que já se sabe que são válidos. Isso parece nos levar ao mesmo problema a respeito das definições, isto é, nos levará a um processo infinito, onde, para se provar um resultado, faz-se necessário a utilização de outro já provado. Ainda de acordo com Ávila [11], os matemáticos gregos não demoraram a perceber a necessidade de considerar certas proposições como evidentes, as quais não careciam de prova por serem consideradas óbvias por si mesmas. A essas proposições damos o nome de "postulados" ou "axiomas".

O livro *Os elementos*, de autoria do matemático Euclides de Alexandria (século III a.C.), foi o primeiro construído com estrutura "axiomática", na qual inicialmente se estabelecem os axiomas e a partir destes, desenvolvem-se resultados mais elaborados, como proposições, teoremas, corolários, etc. Portanto, de acordo com Vieira [12], Euclides foi o primeiro a desenvolver a teoria baseada em axiomas ou postulados, ou seja, baseada em verdades absolutas.

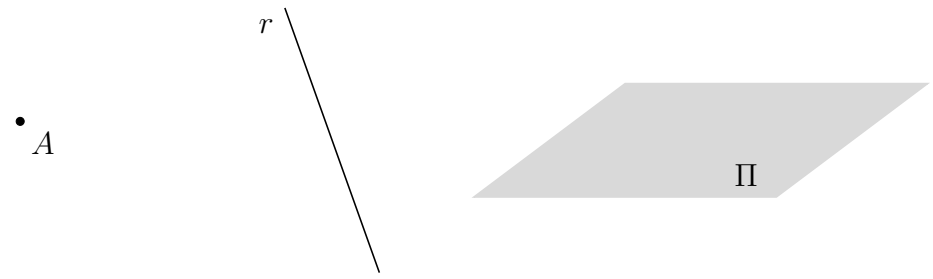
O protagonismo de Euclides frente ao estudo da Geometria, lhe rendeu o título de "pai da Geometria". Além disso, Vieira [12] destaca que *Os Elementos* foi o responsável pelo surgimento da Geometria Euclidiana, em homenagem àquele que o escreveu de modo tão simples e brilhante.

A Matemática, assim como toda ciência, deve ser pautada em resultados verdadeiros e possuir uma linguagem própria a qual visa facilitar a comunicação, e considerando o exposto nos parágrafos anteriores, é de suma importância que se siga uma ordem lógica de exposição. Dito isto, passamos agora a estabelecer as nomenclaturas que utilizaremos, bem como algumas definições e resultados básicos necessários para a compreensão dos assuntos abordados nas seções subsequentes. Em determinadas situações, apresentaremos definições e faremos exposições de resultados que não necessitem de prova (axiomas) logo antes das proposições ou teoremas para tornar a compreensão mais fácil. Reiteramos que os resultados que se seguem foram baseados nas referências [13], [14], [15], [16] e [17].

3.1 Alguns Conceitos Básicos de Geometria Plana

Utilizaremos letras latinas maiúsculas para representar pontos, letras latinas minúsculas para representar retas e letras gregas maiúsculas para representar planos.

Figura 3.1: Ponto, Reta e Plano.



Fonte: Elaborada pelo autor.

Nos termos da Figura 3.1, temos o ponto A, a reta r e o plano Π. Sugerimos ao leitor imaginar o plano como sendo a folha do papel na qual se escreve este trabalho,

estendida indefinidamente nas suas bordas, ou a tela do computador, também estendida indefinidamente, se for o caso do leitor estar lendo através de meios digitais.

A seguir, vamos enunciar os axiomas que utilizaremos no desenvolvimento deste capítulo.

Axioma 3.1.1. *Dados uma reta r e um ponto A, só há duas possibilidades: ou r incide em A ou r não incide em A.*

No primeiro caso, dizemos que "A pertence a r" (indica-se com $A \in r$) ou que A é um ponto de r, e no segundo caso diremos que "A não pertence a r" (indica-se com $A \notin r$).

Axioma 3.1.2. *Dados três pontos distintos de uma reta, só um deles localiza-se entre os outros dois.*

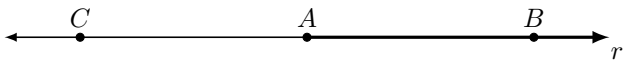
Axioma 3.1.3. *Dados dois pontos distintos A e B de uma reta, sempre existem: um ponto C entre A e B e um ponto D, tal que B está entre A e D.*

Esses Axiomas nos dão suporte para definirmos os conceitos de semirreta e segmento.

Definição 3.1.1. *Dada uma reta r e um ponto A dessa reta, A divide r em duas partes sendo cada uma delas denominadas* ***semirretas*** *de origem A.*

Como um ponto de uma reta a divide em duas semirretas, para diferenciar uma semirreta da outra consideramos dois pontos B e C ambos pertencentes à reta e de forma que A esteja entre eles. Assim, podemos diferenciar as semirretas sendo uma a que contém o ponto B e a outra que contém o ponto C.

Figura 3.2: Semirretas.



Fonte: Elaborada pelo autor.

Representaremos por $\overrightarrow{AB}$ a semirreta de origem A que contém o ponto B e por $\overrightarrow{AC}$ a semirreta de origem A que contém o ponto C.

Definição 3.1.2. *Dados dois pontos distintos A e B sobre uma reta r, chamamos de segmento AB o conjunto de pontos formado por A e B e por todos os pontos situados entre eles e que pertencem a r.*

Representaremos o comprimento do segmento AB por $\overline{AB}$.

Axioma 3.1.4. *Por dois pontos distintos incide uma única reta.*

Quando for conveniente podemos representar a reta que incide em dois pontos A e B por $\overleftrightarrow{AB}$.

Proposição 3.1.1. *Dadas duas retas distintas em um plano, só há duas possibilidades: ou elas não se intersectam ou se intersectam em um único ponto.*

Demonstração. Considere duas retas distintas e suponha que elas se intersectam em dois pontos distintos. Pelo Axioma 3.1.4 as duas retas coincidem, o que contradiz o fato das retas serem distintas. Portanto, ou as retas não se intersectam ou se intersectam em um único ponto. □

Quando duas retas no plano se intersectam, as chamamos de *retas concorrentes* e quando elas não se intersectam, as chamamos de *retas paralelas* (Figura 3.3).

Figura 3.3: Retas concorrentes e retas paralelas.

Fonte: Elaborada palo autor.

A essa altura é natural indagar se há um critério para identificar se determinadas retas são paralelas. Esse critério existe, porém necessita-se de alguns resultados sobre ângulos para torná-lo compreensível. Sendo assim, a seção seguinte traz resultados importantes sobre ângulos.

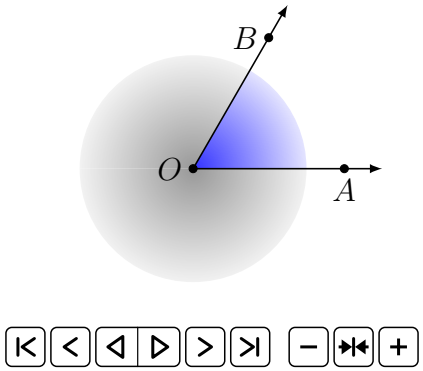
3.2 Ângulos

Definicão 3.2.1. *Um* ***ângulo*** *é uma região do plano delimitada por duas semirretas de mesma origem.*

Denotamos por $\angle AOB$ o ângulo delimitado por duas semirretas $\overrightarrow{OA}$ e $\overrightarrow{OB}$ de mesma origem O. Uma ilustração está na Figura 3.4.

O ponto O é denominado *vértice* do ângulo e as semirretas $\overrightarrow{OA}$ e $\overrightarrow{OB}$ são chamadas de *lados* do ângulo.

Figura 3.4: Ângulo.



Fonte: Elaborada pelo autor.

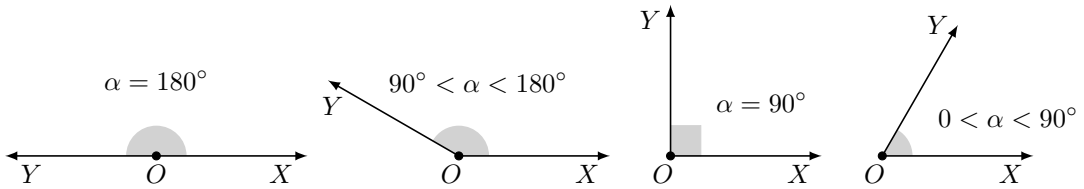
De acordo com a definição de ângulo, as semirretas consideradas determinam dois ângulos, sendo que fica a cargo do contexto qual deve ser considerado. Quando não houver perigo de confusão, utilizaremos a notação $\hat{O}$ para nos referirmos a um dos ângulos específicos de vértice O.

Para medirmos um ângulo, utilizaremos como unidade de medida o *grau*, representado pelo símbolo °. A definição a seguir formaliza essa unidade de medida.

Definição 3.2.2. *Dada uma circunferência de centro O e dois pontos X e Y posicionados sobre a circunferência, convenciona-se que* ***um grau*** *é a medida do ângulo $\angle XOY$ tal que o menor arco com extremidades em X e Y represente $\frac{1}{360}$ da circunferência.*

Usualmente se utilizam letras gregas minúsculas para representar a medida de um ângulo. Um ângulo que mede 90° é chamado ângulo *reto* e um ângulo cuja medida é 180° é chamado ângulo *raso*. Um ângulo *agudo* é aquele com medida α tal que $0 < \alpha < 90°$. Se um ângulo tem medida α, com $90° < \alpha < 180°$, ele será chamado de ângulo *obtuso*. Observe na Figura 3.5 a representação geométrica dessa classificação.

Figura 3.5: Ângulo: raso, obtuso, reto e agudo.



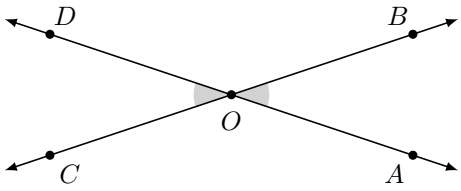
Fonte: Elaborada pelo autor.

Para simplificar a escrita, às vezes escreveremos *"ângulo α"* para indicar um ângulo $\angle XOY$ cuja medida é de α graus.

De acordo com a Definição 3.2.2, podemos considerar que 1° equivale a um arco de $\frac{1}{360}$ da circunferência e, com isso, um ângulo $\angle XOY$ de 180° equivale a um arco de $\frac{1}{2}$ da circunferência. Portanto, a reta $\overleftrightarrow{XY}$ contém o diâmetro da circunferência considerada. Diremos, então, que os lados de um ângulo raso $\angle XOY$ são duas semirretas opostas, ou seja, os pontos X, O e Y pertencem a uma mesma reta, com O entre X e Y. Essas informações proporcionam a definição de ângulos *opostos pelo vértice* que segue.

Definição 3.2.3. *Dois ângulos $\angle AOB$ e $\angle COD$ são* ***opostos pelo vértice*** *quando os lados de um são as semirretas opostas aos lados do outro.*

Figura 3.6: Ângulos opostos pelo vértice.

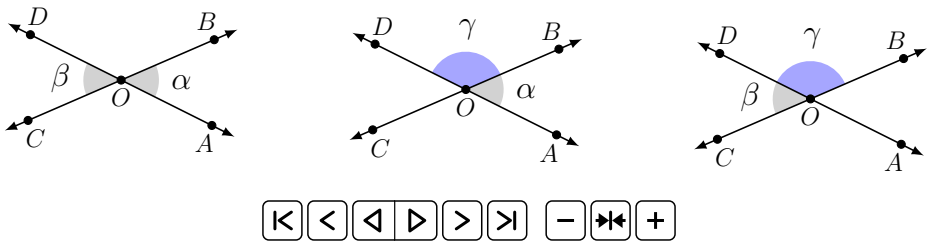


Fonte: Elaborada pelo autor.

Proposição 3.2.1. *Dois ângulos opostos pelo vértice possuem a mesma medida.*

Demonstração. Sejam $\angle AOB$, $\angle COD$ dois ângulos opostos pelo vértice, conforme a Figura 3.7. Denote por α, β e γ, respectivamente, as medidas dos ângulos $\angle AOB$, $\angle COD$ e $\angle BOD$.

Figura 3.7: Ângulos opostos pelo vértice: demonstração.



Fonte: Elaborada pelo autor.

Como $\angle AOB$ e $\angle COD$ são opostos pelo vértice, segue que $\overrightarrow{OA}$ e $\overrightarrow{OD}$ são semirretas opostas e, consequentemente, $\alpha + \gamma = 180^\circ$. Analogamente, temos que

$\beta + \gamma = 180°$. Dessas duas equações, obtemos

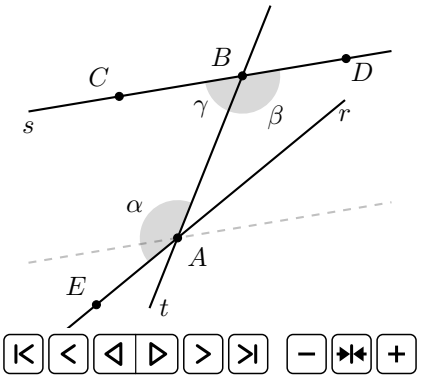
$$\alpha + \gamma = 180° = \beta + \gamma \Longrightarrow \alpha + \gamma = \beta + \gamma \Longrightarrow \alpha = \beta,$$

como queríamos demonstrar. □

Agora vamos enunciar um critério que permite identificar se duas retas são paralelas.

Proposição 3.2.2. *Sejam três retas distintas r, s e t de forma que t intersecta as outras duas, respectivamente, nos pontos A e B. Sejam C e D pontos de s tais que $B \in CD$ e E um ponto de r que esteja no mesmo semiplano que C em relação à reta t, de forma que $\angle EAB = \alpha$, $\angle DBA = \beta$ e $\angle CBA = \gamma$, conforme Figura 3.8. Nessas condições, tem-se que r e s são paralelas se, e somente se, $\alpha = \beta$ ou, equivalentemente, $\alpha + \gamma = 180°$.*

Figura 3.8: Critério para identificar retas paralelas.



Fonte: Elaborada pelo autor com base em Neto [13].

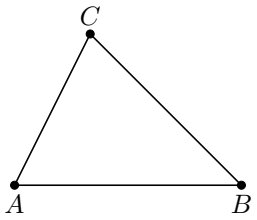
A demonstração desta proposição pode ser encontrada em Neto [13].

3.3 Estudo dos Triângulos

Dados três pontos A, B e C de um plano, diremos que eles são *colineares* se pertencerem a uma mesma reta. Caso contrário, diremos que eles são *não colineares*.

Definicão 3.3.1. *Um **triângulo** é a figura geométrica plana determinada pelos segmentos que unem três pontos não colineares.*

Figura 3.9: Triângulo ABC.



Fonte: Elaborada pelo autor.

Cada um dos pontos é denominado *vértice* e cada um dos segmentos que os unem é chamado de *lado*. Assim, um triângulo possui três vértices e três lados.

Os triângulos são classificados de duas maneiras: de acordo com os ângulos internos e de acordo com as medidas de seus lados.

Quanto aos lados, um triângulo é denominado: *equilátero*, se possuir todos os seus lados com medidas iguais; *isósceles*, se possuir exatamente dois lados com medidas iguais; e *escaleno*, se possuir todos os lados com medidas diferentes. Quanto aos ângulos, um triângulo é denominado: *acutângulo*, se possuir todos os seus ângulos internos agudos; *retângulo*, se possuir um ângulo reto; e *obtusângulo*, se possuir um ângulo interno obtuso.

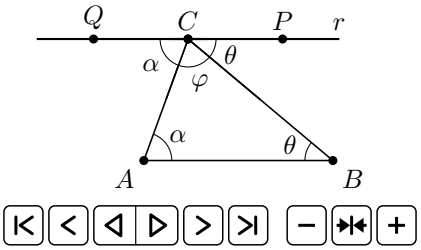
Existem muitos resultados que envolvem triângulos. Destacaremos nesta seção alguns dos mais importantes e que serão utilizados na criação da sequência didática, apresentada no Capítulo 6. Dentre eles, destacam-se os casos de congruências de triângulos. Começaremos estabelecendo um resultado que diz que a soma dos ângulos internos de um triângulo é igual a 180°. Para provarmos esse resultado, faremos uso do axioma a seguir, conhecido como *o quinto postulado de Euclides*.

Axioma 3.3.1. *Por um ponto fora de uma reta r, pode-se traçar uma única reta paralela à reta r.*

Proposição 3.3.1. *A soma das medidas dos ângulos internos de um triângulo é igual a 180°.*

Demonstração. Considere um triângulo ABC, com ângulos internos $\hat{A} = \alpha$, $\hat{B} = \theta$ e $\hat{C} = \varphi$. Trace pelo vértice C a reta r paralela a $\overleftrightarrow{AB}$ e marque os pontos P e Q sobre r de forma que $C \in QP$. Vamos mostrar que $\angle BCP = \theta$ e $\angle ACQ = \alpha$. Acompanhe a animação da Figura 3.10.

Figura 3.10: Soma dos ângulos internos de um triângulo.



Fonte: Elaborada pelo autor com base em Barbosa [15].

Sendo $r \parallel \overleftrightarrow{AB}$, segue imediatamente da Proposição 3.2.2 que $\angle BCP = \theta$ e $\angle ACQ = \alpha$. Sendo P, C e Q pontos sobre r, segue que $\angle PCQ$ é um ângulo raso e, portanto, $\theta + \varphi + \alpha = 180\,^{\circ}$. □

Definição 3.3.2. *Dois segmentos são ditos* ***congruentes*** *quando possuem a mesma medida. De forma análoga, dois ângulos são congruentes se possuem a mesma medida.*

Utilizaremos o símbolo "$\equiv$" para denotar congruência. Assim $AB \equiv CD$ e $\angle ABC \equiv \angle CDE$ indicam, respectivamente, que o segmento AB é congruente ao segmento CD e que o ângulo $\angle ABC$ é congruente ao ângulo $\angle CDE$.

Definição 3.3.3. *Dois triângulos são* ***congruentes*** *se for possível estabelecer uma correspondência biunívoca entre seus vértices, de modo que lados e ângulos correspondentes sejam congruentes.*

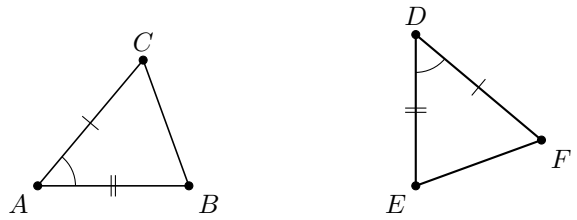
A Definição 3.3.3 sugere que para identificar se dois triângulos são congruentes, devemos verificar se os três lados e os três ângulos correspondentes dos dois triângulos são congruentes. Entretanto, não é necessário que façamos estas verificações em sua totalidade, pois existem os resultados conhecidos como *casos de congruência de triângulos* que facilitam a identificação de triângulos congruentes.

A partir das Definições 3.3.2 e 3.3.3, podemos apresentar o primeiro caso de congruência de triângulo, conhecido como caso LAL (lado, ângulo, lado), o qual trataremos como axioma. Em seguida, apresentaremos outros dois casos como teoremas.

Axioma 3.3.2. *(Caso de congruência LAL). Se em dois triângulos ABC e DEF ocorrerem $AB \equiv DE$, $\angle CAB \equiv \angle FDE$ e $AC \equiv DF$, então ABC é congruente a DEF.*

A Figura 3.11 representa de forma geométrica o Axioma 3.3.2.

Figura 3.11: Caso de congruência de triângulos LAL.

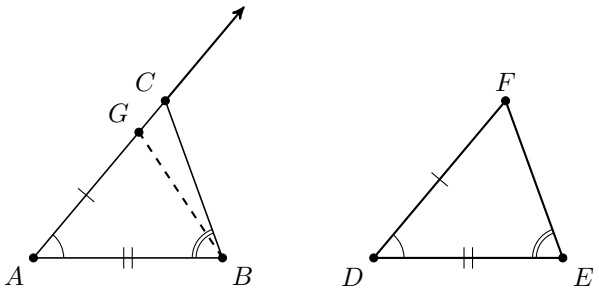


Fonte: Elaborada pelo autor.

Teorema 3.3.1. *(Caso de congruência ALA). Se em dois triângulos ABC e DEF ocorrerem $\hat{A} \equiv \hat{D}$, $AB \equiv DE$ e $\hat{B} \equiv \hat{E}$, então ABC é congruente a DEF.*

Demonstração. Sejam ABC e DEF dois triângulos, tais que $\hat{A} \equiv \hat{D}$, $AB \equiv DE$ e $\hat{B} \equiv \hat{E}$. Considere o pondo G sobre a semirreta $\overrightarrow{AC}$ tal que $AG \equiv DF$.

Figura 3.12: $ABC \equiv DEF$.



Fonte: Elaborada pelo autor com base em Barbosa [15].

Sendo, por construção, $AG \equiv DF$ e, por hipótese, $\hat{A} \equiv \hat{D}$ e $AB \equiv DE$, segue do Axioma 3.3.2 que $ABG \equiv DEF$. Consequentemente, $\angle ABG \equiv \hat{E}$. Mas, por hipótese, $\hat{E} \equiv \hat{B}$, logo $\angle ABG \equiv \hat{B}$. Isso mostra que as semirretas $\overrightarrow{BC}$ e $\overrightarrow{BG}$ coincidem, e, por consequência, os pontos C e G também coincidem, o que nos leva a concluir que os triângulos ABC e ABG são congruentes. Como $ABG \equiv DEF$, concluímos que ABC é congruente a DEF, por transitividade. □

O Teorema 3.3.1 nos permite estabelecer o seguinte resultado, que relaciona os triângulos equiláteros, isósceles e escalenos de acordo com os ângulos internos.

Proposição 3.3.2.

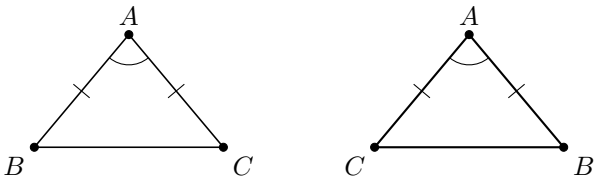
(a) Um triângulo é equilátero se, e somente se, possui os três ângulos internos com a mesma medida.

(b) Um triângulo é isósceles se, e somente se, possui exatamente dois ângulos internos congruentes.

(c) Um triângulo é escaleno se, e somente se, as medidas dos seus ângulos internos são todas distintas.

Demonstração. Iniciemos a demonstração pelo item (b). Vamos provar primeiro que se um triângulo é isósceles, então ele possui exatamente dois ângulos congruentes. Para tal, considere um triângulo isósceles ABC com $AB \equiv AC$. Compare este triângulo com ele mesmo, fazendo corresponder os vértices $A \leftrightarrow A$, $B \leftrightarrow C$ e $C \leftrightarrow B$, conforme a Figura 3.13.

Figura 3.13: ABC isósceles $\Longrightarrow \hat{B} \equiv \hat{C}$.

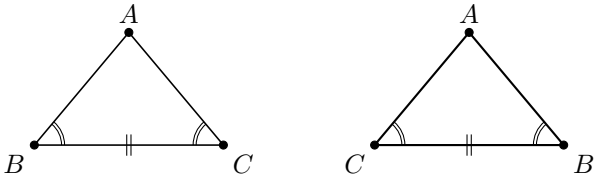


Fonte: Elaborada pelo autor com base em Barbosa [15].

Por hipótese $AB \equiv AC$ e, como $\hat{A} \equiv \hat{A}$, segue do caso de congruência LAL (Axioma 3.3.2), que a correspondência dos vértices determina uma congruência, e como consequência, $\hat{B} \equiv \hat{C}$, o que mostra a primeira parte da proposição.

Agora, considere a mesma correspondência de vértices e que os ângulos $\hat{B}$ e $\hat{C}$ do triângulo ABC sejam congruentes, como na Figura 3.14.

Figura 3.14: $\hat{B} \equiv \hat{C} \Longrightarrow ABC$ isósceles.



Fonte: Elaborada pelo autor.

Sendo, por hipótese, $\hat{B} \equiv \hat{C}$ e como $BC \equiv CB$, segue do caso de congruência ALA (Teorema 3.3.1) que a correspondência estabelecida com os vértices determina uma congruência e, portanto, $AB \equiv AC$, o que prova a recíproca da proposição.

Para o item (a), suponha que o triângulo ABC seja equilátero. Assim, temos que $AB \equiv AC$, $BC \equiv BA$ e $CA \equiv CB$. Essas congruências, juntamente com o item (b), nos permitem concluir, respectivamente, que $\hat{B} \equiv \hat{C}$, $\hat{C} \equiv \hat{A}$ e $\hat{A} \equiv \hat{B}$, o que implica $\hat{B} \equiv \hat{C} \equiv \hat{A}$.

Reciprocamente, suponha que os ângulos internos do triângulo ABC possuem medidas iguais. Fazendo as correspondências $\hat{B} \equiv \hat{C}$, $\hat{C} \equiv \hat{A}$ e $\hat{A} \equiv \hat{B}$ e seguindo um

raciocínio análogo ao utilizado na demonstração da recíproca do item (b), concluímos que $AB \equiv AC$, $BC \equiv BA$ e $CA \equiv CB$, o que implica $AB \equiv AC \equiv BC$. Logo, ABC é equilátero.

Para o item (c), seja ABC um triângulo escaleno. Com isso, ABC não é equilátero nem isósceles, logo, pelas contrapositivas dos itens (a) e (b), concluímos que as medidas dos ângulos internos do triângulo ABC são todas distintas.

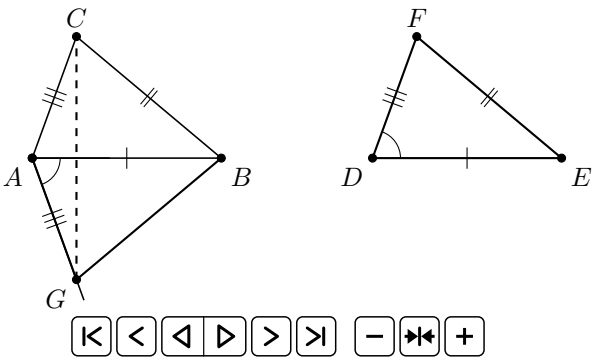
Agora, suponha que o triângulo ABC possui as medidas dos ângulos internos todas distintas. Novamente pelas contrapositivas dos itens (a) e (b), o triângulo ABC não é equilátero, nem isósceles, restando ser escaleno. □

Esta demonstração pode parecer enganosa, visto que um triângulo ABC é sempre congruente a sí mesmo, porém, o que a torna válida é a correspondência estabelecida entre os vértices ser utilizada em um triângulo isósceles. Se tentarmos seguir os mesmos passos supondo que o triângulo seja escaleno, certamente em algum momento aparecerá uma contradição.

Teorema 3.3.2. *(Caso de congruência LLL) Dados dois triângulos ABC e CDE, se $AB \equiv DE$, $BC \equiv EF$ e $CA \equiv FD$, então ABC é congruente a DEF.*

Demonstração. Considere dois triângulos ABC e DEF tais que $AB \equiv DE$, $BC \equiv EF$ e $CA \equiv FD$. Construa o ângulo congruente a $\hat{D}$ com vértice em A e que possui um dos lados sobre AB e o outro situado no semiplano oposto ao que contém o ponto C. Marque o ponto G sobre o lado que não contém B tal que $GA \equiv FD$. Para melhor compreensão, acompanhe a animação representada na Figura 3.15.

Figura 3.15: Caso de congruência LLL.



Fonte: Elaborada pelo autor com base em Barbosa [15].

Sendo, por hipótese, $AB \equiv DE$ e, por construção, $\angle BAG \equiv \hat{D}$ e $GA \equiv FD$. Segue, do caso LAL (Axioma 3.3.2), que os triângulos ABG e DEF são congruentes. Temos então que $CA \equiv FD \equiv GA$ e $BC \equiv EF \equiv BG$, o que nos permite constatar que os triângulos ACG e BCG são isósceles. Assim, pela Proposição 3.3.2, tem-se

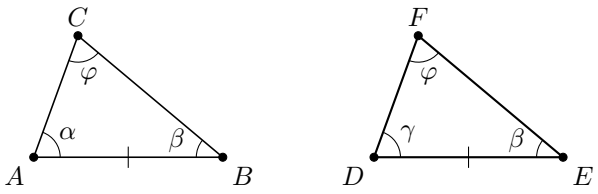
$\angle ACG \equiv \angle AGC$ e $\angle BCG \equiv \angle BGC$, logo, $\angle ACB \equiv \angle AGB$. Utilizando o caso de congruência LAL, concluímos que os triângulos ABG e ABC são congruentes. Como $ABG \equiv DEF$, concluímos que $ABC \equiv DEF$, por transitividade. □

Além dos três casos de congruência de triângulos LAL, ALA e LLL apresentados, existe um quarto resultado que é consequência direta do caso ALA, o qual chamamos de *caso de congruência LAA_o* (lado, ângulo, ângulo oposto).

Proposição 3.3.3. *(Caso de congruência LAA_o). Dados dois triângulos ABC e DEF, se $AB \equiv DE$, $\hat{B} \equiv \hat{E}$ e $\hat{C} \equiv \hat{F}$, então os triângulos ABC e DEF são congruentes.*

Demonstração. Sejam ABC e DEF dois triângulos, tais que $AB \equiv DE$, $\hat{B} \equiv \hat{E}$ e $\hat{C} \equiv \hat{F}$. Sejam $\alpha = \hat{A}$, $\beta = \hat{B} = \hat{E}$, $\gamma = \hat{D}$ e $\varphi = \hat{C} = \hat{F}$, conforme a Figura 3.16. Vamos provar que $ABC \equiv DEF$.

Figura 3.16: Caso de congruência $\mathrm{LAA_o}$.



Fonte: Elaborada pelo autor.

De acordo com a Proposição 3.3.1, a soma dos ângulos internos em cada triângulo resulta em 180°, o que nos leva a

$$\alpha + \beta + \varphi = 180^\circ = \gamma + \beta + \varphi \Longrightarrow \alpha = \gamma.$$

Isso nos permite concluir que $\hat{A} \equiv \hat{D}$ e, fazendo uso do caso ALA de congruência de triângulos, concluímos que os triângulos ABC e DEF são congruentes. □

A seguir, vamos apresentar as noções de *altura*, *bissetriz* e *mediana* de um triângulo.

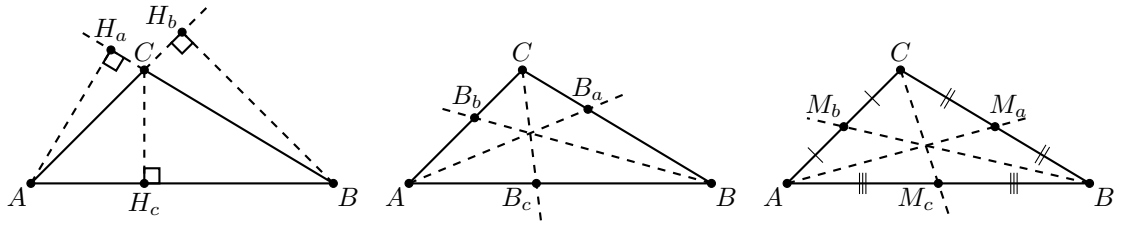
Definição 3.3.4. *Dado um triângulo ABC, define-se a **altura** relativa ao lado BC (ou ao vértice A) como sendo o segmento que une o vértice A ao pé da perpendicular baixada de A sobre a reta $\overleftrightarrow{BC}$.*

Definição 3.3.5. *Dado um triângulo ABC, define-se a **bissetriz interna** relativa ao lado BC (ou ao vértice A) como sendo o segmento que une o vértice A ao ponto de interseção da bissetriz do ângulo $\hat{A}$ e o lado BC.*

Definição 3.3.6. *Dado um triângulo ABC, define-se a* ***mediana*** *relativa ao lado BC (ou ao vértice A) como sendo o segmento que une o vértice A ao ponto médio do lado BC.*

Seguindo estas definições, é evidente que um triângulo ABC possui três alturas, três bissetrizes internas e três medianas, cada uma relativa a cada um dos lados do triângulo, conforme Figura 3.17, onde as alturas estão representadas à esquerda, as bissetrizes ao centro e as medianas à direita.

Figura 3.17: Altura, bissetriz e mediana.



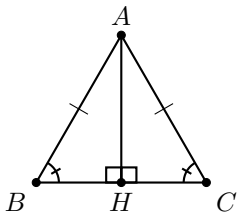
Fonte: Elaborada pelo autor.

Definição 3.3.7. *Em um triângulo isósceles, os lados congruentes são chamados de* ***laterais*** *e o terceiro lado, de* ***base****.*

Proposição 3.3.4. *A altura relativa à base de um triângulo isósceles coincide com a bissetriz interna e com a mediana relativas à mesma base.*

Demonstração. Considere um triângulo ABC isósceles de base BC. Vamos mostrar que a altura, a bissetriz interna e a mediana relativas a este lado coincidem.

Figura 3.18: Altura de um triângulo isósceles.



Fonte: Elaborada pelo autor.

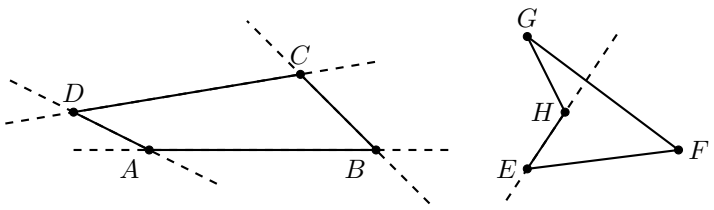
Seja H o pé da altura relativa ao lado BC. Como ABC é isósceles de base BC, segue que $\overline{AC} = \overline{AB}$ e $\angle ACH \equiv \angle ABH$. Sendo AH a altura relativa a BC, segue que $\angle AHB \equiv \angle AHC$ (pois ambos os ângulos são retos). Assim, pelo caso de congruência LAA_o, segue que $AHB \equiv AHC$ e esta congruência nos permite concluir que $\angle BAH \equiv \angle CAH$ e $\overline{HC} = \overline{HB}$, o que implica, respectivamente, em AH ser a bissetriz e a mediana relativa ao lado BC. □

3.4 Estudo dos Quadriláteros

Por *quadrilátero*, entendemos a figura geométrica plana formada pela junção de quatro pontos, três a três não colineares e posicionados de forma a determinar exatamente quatro segmentos de reta. Assim como nos triângulos, cada um dos pontos é chamado de *vértice* e cada um dos segmentos, de *lados*. Portanto, um quadrilátero possui quatro vértices e quatro lados.

Diremos que um quadrilátero é *convexo* se ao prolongarmos qualquer um de seus quatro lados, este prolongamento não possuir outro ponto do quadrilátero além dos que o determina. Caso contrário, diremos que ele é *côncavo*. Em outras palavras, um quadrilátero é *convexo* se qualquer prolongamento de seus lados não o divide em duas ou mais figuras. Observe a Figura 3.19, nela o quadrilátero da esquerda é convexo e o da direita é côncavo.

Figura 3.19: Quadriláteros convexo e côncavo.

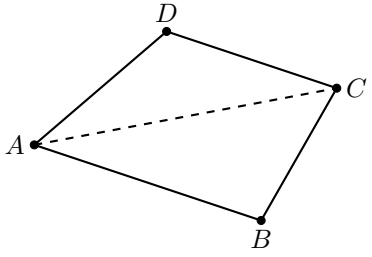


Fonte: Elaborada pelo autor.

Proposição 3.4.1. *A soma dos ângulos internos de um quadrilátero é igual a 360°.*

Demonstração. Dado um quadrilátero $ABCD$, trace o segmento AC, particionando-o nos dois triângulos ABC e ACD, conforme a Figura 3.20, de modo que a soma dos ângulos internos dos dois triângulos é igual à soma dos ângulos internos de $ABCD$. Como a soma dos ângulos internos de um triângulo é igual a $180°$, o resultado segue imediatamente. □

Figura 3.20: Partição do quadrilátero $ABCD$ em dois triângulos.



Fonte: Elaborada pelo autor.

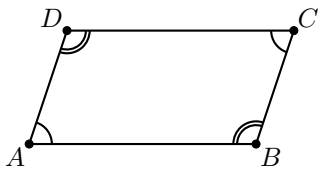
Definição 3.4.1. *Chama-se* ***paralelogramo*** *o quadrilátero convexo que possui os lados opostos paralelos.*

Sobre os paralelogramos destacamos as proposições que seguem. Elas são muitas vezes utilizadas no Ensino Médio de forma natural, mas é de suma importância que saibamos que elas são passíveis de demonstração.

Proposição 3.4.2. *Um quadrilátero convexo é um paralelogramo se, e somente se possui os ângulos opostos congruentes.*

Demonstração. Considere um quadrilátero convexo $ABCD$, e suponha primeiro que $\hat{A} \equiv \hat{C}$ e $\hat{B} \equiv \hat{D}$, conforme Figura 3.21. Vamos mostrar que este quadrilátero é um paralelogramo.

Figura 3.21: $\hat{A} \equiv \hat{C}$ e $\hat{B} \equiv \hat{D} \Longrightarrow ABCD$ paralelogramo.



Fonte: Neto [13].

Da congruência dos ângulos opostos é possível formar as seguintes equações: $\hat{A}+\hat{B} = \hat{D}+\hat{C}$ e $\hat{A}+\hat{D} = \hat{B}+\hat{C}$. Sabemos, da Proposição 3.4.1, que $\hat{A}+\hat{B}+\hat{C}+\hat{D} = 360\,^\circ$, agora basta utilizarmos convenientemente nesta equação as duas igualdades anteriores para obtermos

$$\begin{aligned} \hat{A}+\hat{B}+\hat{C}+\hat{D} = 360\,^\circ &\Longrightarrow \hat{D}+\hat{C}+\hat{C}+\hat{D} = 360\,^\circ \\ &\Longrightarrow 2\cdot(\hat{D}+\hat{C}) = 360\,^\circ \\ &\Longrightarrow \hat{D}+\hat{C} = 180\,^\circ \end{aligned}$$

e

$$\begin{aligned} \hat{A}+\hat{B}+\hat{C}+\hat{D} = 360\,^\circ &\Longrightarrow \hat{A}+\hat{A}+\hat{D}+\hat{D} = 360\,^\circ \\ &\Longrightarrow 2\cdot(\hat{A}+\hat{D}) = 360\,^\circ \\ &\Longrightarrow \hat{A}+\hat{D} = 180\,^\circ. \end{aligned}$$

Com isso, $\hat{A}+\hat{B} = \hat{D}+\hat{C} = 180\,^\circ$ e $\hat{A}+\hat{D} = \hat{B}+\hat{C} = 180\,^\circ$. Segue, da Proposição 3.2.2, que $\overleftrightarrow{AD} \parallel \overleftrightarrow{BC}$ e $\overleftrightarrow{AB} \parallel \overleftrightarrow{DC}$. Portanto, $ABCD$ é paralelogramo.

Agora suponha que $ABCD$ é um paralelogramo. Vamos mostrar que $\hat{A} \equiv \hat{C}$ e

$\hat{B} \equiv \hat{D}$. Da definição de paralelogramo e da Proposição 3.2.2, temos que

$$\overleftrightarrow{AB} \parallel \overleftrightarrow{DC} \Longrightarrow \hat{A} + \hat{D} = 180\,^\circ \quad \text{e} \quad \hat{B} + \hat{C} = 180\,^\circ, \tag{3.1}$$

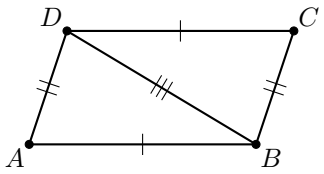
$$\overleftrightarrow{AD} \parallel \overleftrightarrow{BC} \Longrightarrow \hat{A} + \hat{B} = 180\,^\circ \quad \text{e} \quad \hat{D} + \hat{C} = 180\,^\circ. \tag{3.2}$$

Das equações em (3.1) e (3.2), obtemos $\hat{A} + \hat{B} = 180\,^\circ = \hat{B} + \hat{C} \Longrightarrow \hat{A} = \hat{C}$ e $\hat{A} + \hat{B} = 180\,^\circ = \hat{A} + \hat{D} \Longrightarrow \hat{B} = \hat{D}$, o que completa a demonstração. □

Proposição 3.4.3. *Um quadrilátero convexo é um paralelogramo se, e somete se possui os lados opostos congruentes.*

Demonstração. Considere um quadrilátero $ABCD$, com os lados AB e AD opostos a CD e CB, respectivamente. Suponha que se tenha $AB \equiv CD$ e $AD \equiv CB$. Vamos mostrar que este quadrilátero é um paralelogramo. Para tal, considere o segmento BD como na Figura 3.22.

Figura 3.22: $AB \equiv CD$ e $AD \equiv CB \Longrightarrow ABCD$ paralelogramo.

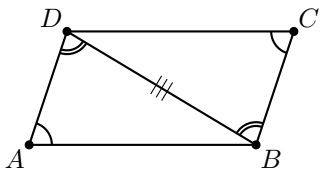


Fonte: Elaborada pelo autor com base em Neto [13].

Tem-se pelo caso de congruência LLL que $ABD \equiv CDB$ e consequentemente $\hat{A} \equiv \hat{C}$. Traçando o segmento AC e seguindo um raciocínio análogo, conclui-se que $\hat{B} \equiv \hat{D}$. Portanto $ABCD$ possui os ângulos opostos congruentes, logo é um paralelogramo.

Agora, suponha que $ABCD$ seja um paralelogramo, vamos mostrar que $AB \equiv CD$ e $AD \equiv CB$. Para tal, trace o segmento BD como na Figura 3.23.

Figura 3.23: $ABCD$ paralelogramo $\Longrightarrow AB \equiv CD$ e $AD \equiv CB$.



Fonte: Elaborada pelo autor com base em Neto [13].

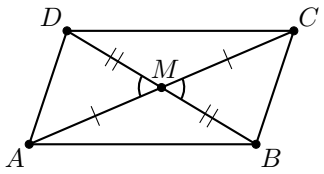
Sendo $ABCD$ um paralelogramo, segue que $\hat{A} \equiv \hat{C}$ e como $\overleftrightarrow{AD} \parallel \overleftrightarrow{BC}$, tem-se $\angle ADB \equiv \angle CBD$. Portanto, $ABD \equiv CDB$ pelo caso de congruência LAA_o, donde segue imediatamente que $AB \equiv CD$ e $AD \equiv CB$. □

Definição 3.4.2. *Chamamos de* ***diagonal*** *de um quadrilátero, o segmento que liga dois vértices não consecutivos.*

Proposição 3.4.4. *Um quadrilátero convexo é um paralelogramo se, e somete se suas diagonais se intersectam em seus pontos médios.*

Demonstração. Dado um quadrilátero $ABCD$, suponha que as diagonais AC e BD se intersectam no ponto M e que ele seja o ponto médio de ambas. Vamos provar que $ABCD$ é paralelogramo, conforme ilustrado na Figura 3.24.

Figura 3.24: M ponto médio de AC e $BD \Longrightarrow ABCD$ paralelogramo.

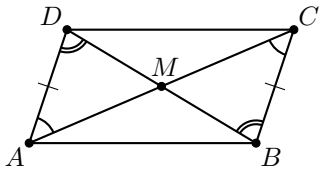


Fonte: Elaborada pelo autor com base em Neto [13].

Sendo M o ponto médio das diagonais segue que $AM \equiv CM$ e $BM \equiv DM$. Note que $\angle AMD \equiv \angle BMC$, pois são opostos pelo vértice. Logo, pelo caso de congruência LAL, os triângulos AMD e CMB são congruentes, donde segue imediatamente que $AD \equiv CB$. Analogamente, concluí-se que $AB \equiv CD$. Logo, $ABCD$ possui os lados opostos congruentes e, portanto, é um paralelogramo.

Agora suponha que $ABCD$ é um paralelogramo. Vamos provar que M é ponto médio das diagonais.

Figura 3.25: $ABCD$ um paralelogramo $\Longrightarrow M$ ponto médio de AC e BD.



Fonte: Elaborada pelo autor com base em Neto [13].

Sendo $ABCD$ paralelogramo, segue que $AD \equiv CB$. Por definição $\overleftrightarrow{AD}$ é paralela a $\overleftrightarrow{BC}$, portanto $\angle DAM \equiv \angle BCM$ e $\angle MDA \equiv \angle MBC$. Logo, AMD é congruente a CMB, pelo caso de congruência ALA, donde segue imediatamente que $AM \equiv CM$ e $BM \equiv DM$, isto é, M é ponto médio das diagonais de $ABCD$. □

Definição 3.4.3. *Um* ***retângulo*** *é um quadrilátero convexo que possui os quatro ângulos internos retos.*

Proposição 3.4.5. *Todo retângulo é paralelogramo.*

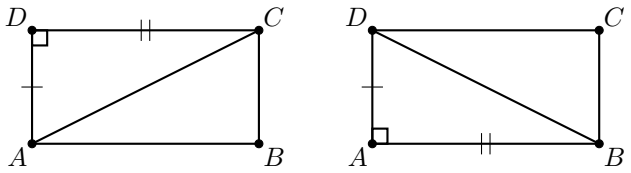
Demonstração. Como em um retângulo todos os ângulos internos são retos, os ângulos opostos são congruentes. Logo, pela Proposição 3.4.2, todo retângulo é paralelogramo. □

É importante ressaltar que a recíproca da Proposição 3.4.5 não é verdadeira, isto é, nem todo paralelogramo é retângulo, visto que para isso ocorrer é necessário que o paralelogramo possua os ângulos internos retos. Além disso, a proposição a seguir nos permite estabelecer uma condição necessária e suficiente para determinar quando um paralelogramo é também retângulo.

Proposição 3.4.6. *Um paralelogramo é retângulo se, e somente se, possui as diagonais congruentes.*

Demonstração. Suponha que um paralelogramo $ABCD$ seja retângulo e sejam AC e BD suas diagonais, como na Figura 3.26.

Figura 3.26: $ABCD$ um paralelogramo retângulo $\Longrightarrow AC \equiv BD$.

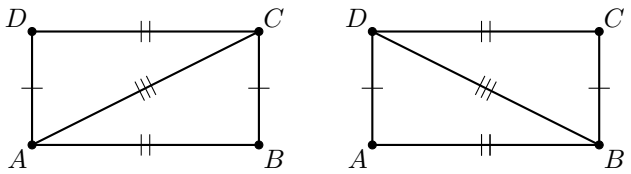


Fonte: Elaborada pelo autor.

Como $ABCD$ é paralelogramo e retângulo, segue que $DC \equiv AB$ e $\hat{D} = \hat{A} = 90^{\circ}$. Agora note que os lados AD e DA são comuns aos triângulos ACD e DBA. Assim, pelo caso de congruência LAL, concluí-se que $ACD \equiv DBA$, logo $AC \equiv BD$.

Agora suponha que $ABCD$ seja um paralelogramo com $AC \equiv BD$, como na Figura 3.27. Vamos provar que ele é retângulo.

Figura 3.27: $ABCD$ um paralelogramo com $AC \equiv BD \Longrightarrow ACBD$ retângulo.



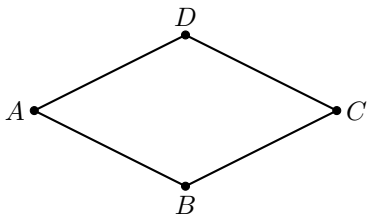
Fonte: Elaborada pelo autor.

Sendo $ABCD$ paralelogramo, segue que $AB \equiv CD \equiv DC \equiv BA$ e $AD \equiv CB \equiv BC \equiv DA$ e, como por hipótese, $AC \equiv BD$, isto é, $AC \equiv BD \equiv DB \equiv CA$,

segue do caso de congruência LLL que $BAD \equiv ABC \equiv DCB \equiv CDA$, logo, $\hat{A} \equiv \hat{B} \equiv \hat{C} \equiv \hat{D}$. Já sabemos, da Proposição 3.4.1, que a soma dos ângulos internos de um quadrilátero convexo é igual a 360°. Segue, daí, que $\hat{A} = \hat{B} = \hat{C} = \hat{D} = 90^{\circ}$, isto é, $ABCD$ é um retângulo. □

Definição 3.4.4. *Um **losango** é um quadrilátero convexo que possui os quatro lados congruentes.*

Figura 3.28: Losango $ABCD$ com $AB \equiv BC \equiv CD \equiv DA$.



Fonte: Elaborada pelo autor.

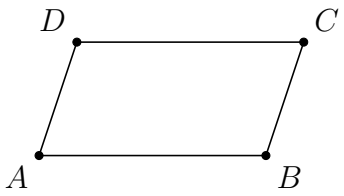
Proposição 3.4.7. *Todo losango é um paralelogramo.*

Demonstração. Por definição, um losango possui os quatro lados congruentes e, consequentemente, os lados opostos também o são. Logo, pela Proposição 3.4.3, todo losango é paralelogramo. □

A recíproca da Proposição 3.4.7 não é verdadeira, isto é, nem todo paralelogramo é um losango, visto que, para ele o ser, deve possuir, além dos lados opostos paralelos, os quadro lados congruentes, o que nem sempre ocorre em um paralelogramo, conforme Figura 3.29. Para além disso, uma condição necessária e suficiente para que um paralelogramo seja um losango é apresentada na proposição a seguir, cuja demonstração pode ser encontrada em Neto [13].

Proposição 3.4.8. *Um paralelogramo é um losango se, e somente se, suas diagonais forem perpendiculares.*

Figura 3.29: Paralelogramo $ABCD$ que não é losango.

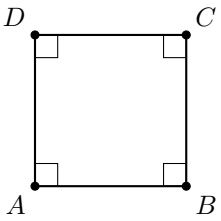


Fonte: Elaborada pelo autor.

O último paralelogramo que apresentaremos é o *quadrado* cuja definição é:

Definição 3.4.5. *Um* ***quadrado*** *é um quadrilátero que possui os quatro lados congruentes e os quatro ângulos internos retos.*

Figura 3.30: Quadrado $ABCD$.



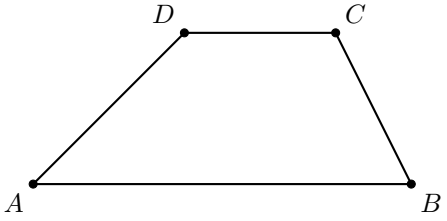
Fonte: Elaborada pelo autor.

É imediato, das Proposições 3.4.2 e 3.4.3, que todo quadrado é paralelogramo, e segue da definição de retângulo que todo quadrado é retângulo e da definição de losango que todo quadrado é losango. Também é fácil constatar que as recíprocas destas afirmações não são verdadeiras.

Encerramos esta seção com a definição de *trapézio*, quadrilátero muito estudado no Ensino Médio.

Definição 3.4.6. *Um* ***trapézio*** *é um quadrilátero convexo que possui exatamente um par de lados paralelos.*

Figura 3.31: Trapézio $ABCD$ com $AB \parallel CD$.



Fonte: Elaborada pelo autor.

3.5 Polígonos regulares e o círculo

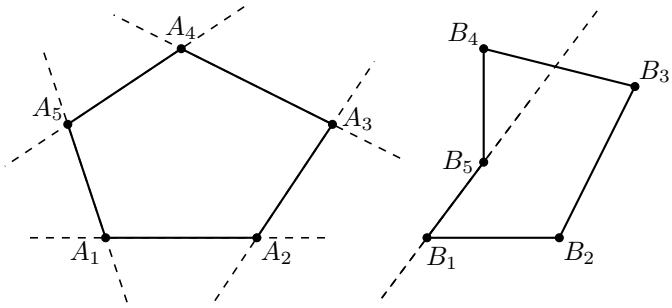
Nesta seção, trataremos das definições e principais resultados envolvendo polígonos regulares e o círculo. A estrutura da presente seção converge para a apresentação e demonstração do fato de que todo polígono regular é inscritível em um círculo, fato que será utilizado na proposta de sequência didática apresentada no Capítulo 6. Comecemos, então, pela definição de *polígono*.

Definição 3.5.1. *Dados n pontos distintos A_1, A_2, A_3, ..., A_n, com $n \geqslant 3$, chamamos de* ***polígono*** *a figura geométrica plana formada por n segmentos que unem estes n pontos de forma que nenhum segmento cruze outro em um ponto que não seja uma de suas extremidades.*

Cada um dos n pontos que determinam o polígono é chamado de *vértice* e cada segmento, de *lado*.

Assim como nos quadriláteros existem os polígonos *convexos* e os *côncavos* cuja definição é a mesma apresentada para os quadriláteros, isto é, um polígono é *convexo* se, ao prolongarmos qualquer um de seus lados, este prolongamento não contiver outro ponto do polígono, a não ser os pontos que determinam o segmento prolongado. Caso contrário o polígono é dito *côncavo*.

Figura 3.32: Polígono convexo e côncavo.



Fonte: Elaborada pelo autor.

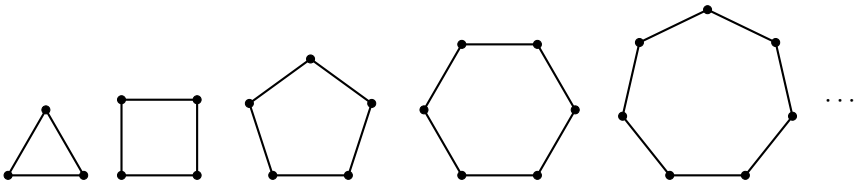
De acordo com Neto [13], é comum nomear um polígono com n vértices (ou n lados) de *n-ágono*, mas existem nomes específicos para alguns polígonos, tais como: *triângulo* ($n = 3$), *quadrilátero* ($n = 4$), *pentágono* ($n = 5$), *hexágono* ($n = 6$), *heptágono* ($n = 7$), *octógono* ($n = 8$), *eneágono* ($n = 9$), *decágono* ($n = 10$), etc.

A definição a seguir nos apresenta um grupo particular de polígonos, os chamados *polígonos regulares*.

Definição 3.5.2. *Um* ***polígono*** *é* ***regular*** *se for convexo e possuir todos os seus lados iguais e todos os seus ângulos internos congruentes.*

É óbvio que um triângulo equilátero e um quadrado são polígonos regulares. Observe na Figura 3.33 estes dois polígonos juntamente com um pentágono, um hexágono e um heptágono regulares.

Figura 3.33: Polígonos regulares.



Fonte: Elaborada pelo autor.

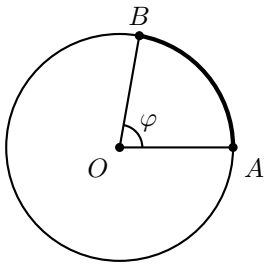
Definição 3.5.3. *Dado um ponto O e um número real $\boldsymbol{r}$, definimos um **círculo** de centro O e raio $\boldsymbol{r}$ como sendo a figura geométrica formada por todos os pontos do plano que estão a uma distância igual a $\boldsymbol{r}$ do ponto O.*

Definição 3.5.4. *Dados dois pontos distintos A e B sobre um círculo de centro O, chamamos $\angle AOB$ de **ângulo central**.*

É evidente que dois pontos distintos sobre um círculo determinam dois ângulos centrais.

Dois pontos A e B distintos sobre um círculo o dividem em duas partes, as quais chamaremos de *arcos* e os denotaremos por $\overset{\frown}{AB}$ e $\overset{\frown}{BA}$, sendo que esta notação diferencia um arco do outro levando-se em conta o sentido anti-horário de rotação, isto é, $\overset{\frown}{AB}$ é a porção do círculo que inicia em A e vai no sentido anti-horário em direção ao ponto B e o arco $\overset{\frown}{BA}$ é a porção do círculo que inicia em B e vai em direção, também no sentido anti-horário, ao ponto A. Observe na Figura 3.34 a representação do ângulo central $\angle AOB$ e do arco $\overset{\frown}{AB}$ (linha mais espessa do círculo) e o arco $\overset{\frown}{BA}$ (linha mais fina do círculo).

Figura 3.34: Ângulo central e arcos de um círculo.

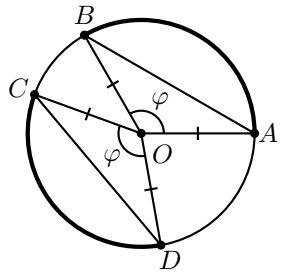


Fonte: Elaborada pelo autor.

Proposição 3.5.1. *Se dois ângulos centrais $\angle AOB$ e $\angle COD$ de um círculo de centro O têm medidas iguais, então as medidas dos segmentos AB e CD também são iguais.*

Demonstração. Suponha primeiro que $\angle AOB = \angle COD = \varphi < 180^{\,o}$, conforme Figura 3.35.

Figura 3.35: $\angle AOB = \angle COD < 180^{\,o} \Longrightarrow \overline{AB} = \overline{CD}$.

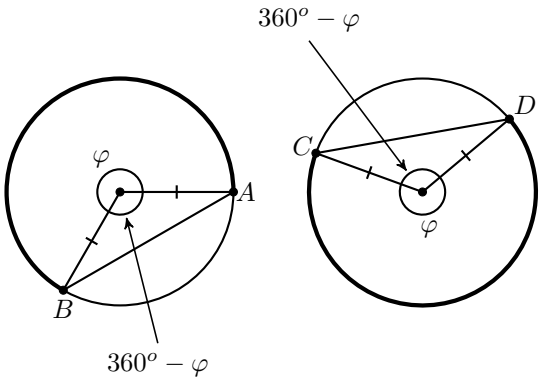


Fonte: Elaborada pelo autor com base em Neto [13].

Basta notar que $\overline{AO} = \overline{CO}$, $\overline{BO} = \overline{DO}$ (pois são raios do mesmo círculo) e $\angle AOB = \varphi = \angle COD$ (por hipótese), e, com isso, os triângulos AOB e COD são congruentes pelo caso LAL, donde segue imediatamente que $\overline{AB} = \overline{CD}$.

Agora suponha que $\angle AOB = \angle COD = \varphi > 180^{\,o}$, conforme Figuras 3.36 (considere que os círculos são o mesmo, construímos separados para facilitar a compreensão).

Figura 3.36: $\angle AOB = \angle COD > 180^{\,o} \Longrightarrow \overline{AB} = \overline{CD}$.



Fonte: Elaborada pelo autor.

Como os ângulos centrais que correspondem aos arcos $\widehat{AB}$ e $\widehat{CD}$ são congruentes e de medida φ, segue que os ângulos que correspondem aos arcos $\widehat{BA}$ e $\widehat{DC}$ medem $360^{\,o} - \varphi$. Ora, $\overline{AO} = \overline{CO}$ e $\overline{BO} = \overline{DO}$, pois são raios de um mesmo círculo e novamente, podemos concluir, pelo caso de congruência LAL, que os triângulos AOB e COD são congruentes, consequentemente $\overline{AB} = \overline{CD}$.

Por fim, se $\angle AOB = \angle COD = \varphi = 180°$, então $\overline{AB} = \overline{CD} = 2r$, onde r é o raio do círculo. □

Definição 3.5.5. *Diremos que um polígono é* ***inscritível*** *em um círculo se existir um círculo que passe por todos os seus vértices.*

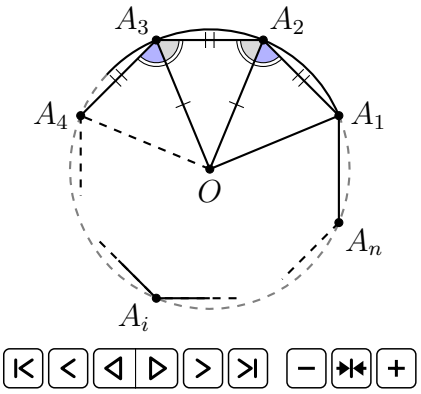
Proposição 3.5.2. *Todo triângulo é inscritível em um círculo.*

A demonstração da Proposição 3.5.2 pode ser encontrada em Barbosa [15].

Proposição 3.5.3. *Todo polígono regular é inscritível em um círculo.*

Demonstração. Seja $A_1A_2A_3\cdots A_n$ um polígono regular. Trace o círculo que inscreve o triângulo $A_1A_2A_3$ e seja O o centro deste círculo, conforme Figura 3.37.

Figura 3.37: Polígono regular inscritível.



Fonte: Elaborada pelo autor com base em Barbosa [15].

Como $OA_2 \equiv OA_3$, pois são raios de um mesmo círculo, segue que o triângulo OA_2A_3 é isósceles e, com isso, $\angle OA_3A_2 \equiv \angle OA_2A_3$. Sendo $A_1A_2A_3\cdots A_n$ um polígono regular, segue que os ângulos internos $\angle A_3A_2A_1$ e $\angle A_4A_3A_2$ são congruentes, logo, $\angle OA_2A_1 \equiv \angle OA_3A_4$. Temos, então, que $OA_2 \equiv OA_3$, $\angle OA_2A_1 \equiv \angle OA_3A_4$ e $A_2A_1 \equiv A_3A_4$, portanto, pelo caso de congruência LAL, os triângulos OA_2A_1 e OA_3A_4 são congruentes e, com isso, $OA_4 \equiv OA_1$, isto é, A_4 pertence ao círculo construído. De forma análoga conclui-se que os demais pontos A_5, A_6, $\cdots$, A_n pertencem ao mesmo círculo, o que implica o polígono $A_1A_2A_3\cdots A_n$ ser inscritível. □

3.6 Transformações Geométricas no Plano

Apresentamos nesta seção algumas transformações geométricas no plano, as quais servirão de base para a explicação das construções e animações das figuras apresentadas no Capítulo 5.

Inicialmente vamos definir formalmente uma transformação geométrica.

Definição 3.6.1. *Uma* ***transformação geométrica*** *no plano* Π *é uma função* $T : \Pi \longmapsto \Pi$ *que associa a cada ponto* P *do plano* Π *um ponto* $P' = T(P)$ *de* Π.

Se uma transformação no plano preserva distâncias, dizemos que ela é uma *isometria*. Simbolicamente, dizemos que T é uma isometria quando $d(T(P_1), T(P_2)) = d(P_1, P_2)$, para quaisquer P_1 e P_2 em Π, onde $d(A, B)$ denota a distância entre os pontos A e B no plano Π. É relevante ressaltar que, de acordo com Wagner [16], as isometrias possuem as seguintes propriedades:

i. a imagem de uma reta por uma isometria é sempre uma reta;

ii. uma isometria preserva paralelismo;

iii. uma isometria preserva ângulos.

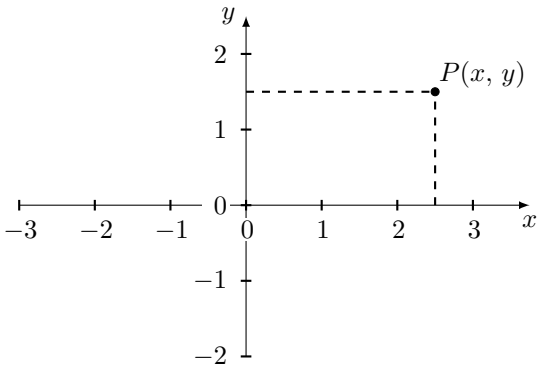
Por consequência da definição, e das três propriedades anteriores, a imagem de uma figura F por uma isometria, é uma figura F' congruente a F. A seguir, apresentaremos as seguintes transformações: *translação*, *rotação*, *simetria* e *reflexão*, as quais provaremos que são isometrias e, por consequência, assumiremos que satisfazem as propriedades listadas acima. As demonstrações para tais afirmações podem ser encontradas em Lima [17].

3.6.1 Translações

Nesta seção vamos assumir que o leitor é familiarizado com o sistema de coordenadas cartesianas, utilizado para localizar pontos no plano. Esse sistema tem por base duas retas numéricas perpendiculares, designadas por Ox e Oy, ambas com origem no ponto O. A reta Ox é chamada *eixo* Ox (ou *eixo das abscissas*), a reta Oy é chamada *eixo* Oy (ou *eixo das ordenadas*) e o ponto O, onde as retas concorrem, é denominado a *origem* do sistema de coordenadas.

No sistema de coordenadas cartesianas, também chamado de plano cartesiano, escrevemos $P(x, y)$ para indicar que o ponto P do plano é representado pelo par ordenado (x, y) de números reais (as coordenadas cartesianas de P), de forma que x, denominado *abscissa* de P, representa a projeção de P sobre o eixo Ox, e y, denominado *ordenada* de P, representa a projeção de P sobre o eixo Oy. Usualmente, ao desenharmos o plano cartesiano, colocamos o eixo das abscissas na horizontal, com a parte positiva para a direita, e o eixo das ordenadas na vertical, com a parte positiva para cima. Veja a Figura 3.38.

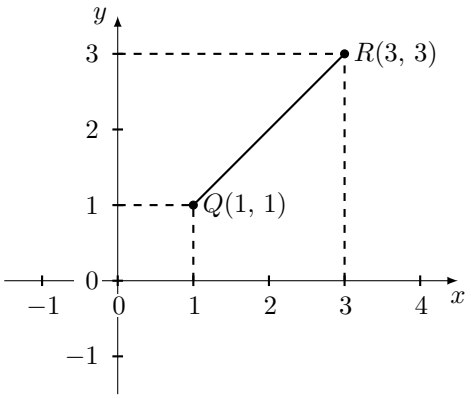
Figura 3.38: Representação do ponto $P(x, y)$ no plano cartesiano.



Fonte: Elaborada pelo autor.

Podemos também representar segmentos por meio de coordenadas cartesianas, basta dotar os pontos de suas extremidades de coordenadas. Observe a Figura 3.39, a representação do segmento QR cujas extremidades são os pontos $Q(1, 1)$ e $R(3, 3)$.

Figura 3.39: Representação do segmento QR no plano cartesiano.



Fonte: Elaborada pelo autor.

O sistema de coordenadas cartesianas nos permite calcular a distância entre pontos do plano, de acordo com a fórmula dada no teorema a seguir, cuja demonstração segue do Teorema de Pitágoras.

Teorema 3.6.1. *Dados os pontos $P_1(x_1, y_1)$ e $P_2(x_2, y_2)$, o comprimento do segmento P_1P_2 é dado por $\overline{P_1P_2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.*

No momento, estamos interessados na seguinte definição.

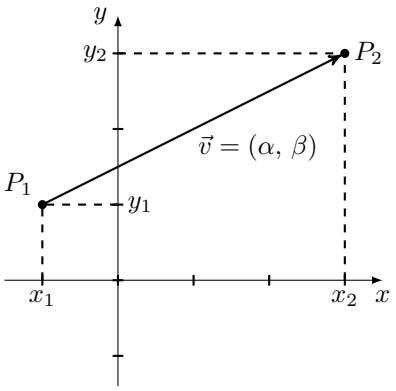
Definicão 3.6.2. *Dados dois pontos $P_1(x_1, y_1)$ e $P_2(x_2, y_2)$, definimos um **vetor** $\vec{v}$ como sendo o par de números (α, β) tais que $\alpha = x_2 - x_1$ e $\beta = y_2 - y_1$ e escrevemos $\vec{v} = (\alpha, \beta)$.*

Perceba que a definição de vetor o torna exclusivamente dependente dos pontos P_1 e P_2 considerados. Diremos então que o segmento P_1P_2 é um *representante* do vetor $\vec{v}$. Note também que um vetor pode ser representado por mais de um segmento. Por exemplo, o vetor $\vec{v} = (1,\, 2)$ pode ser representado por AB com $A(2,\, 3)$ e $B(3,\, 5)$ ou pelo segmento CD com $C(-1,\, 5)$ e $D(0,\, 7)$.

Estabeleceremos também que a ordem com que escrevemos os pontos do segmento representante do vetor é importante, visto que para calcularmos os valores do vetor subtraímos as coordenadas do primeiro ponto do segundo. Portanto, a partir de agora dois segmentos com mesmas extremidades, digamos AB e BA, serão tratados como segmentos que possuem orientações diferentes.

Geometricamente representaremos os vetores por um dos segmentos que o representa com uma seta em uma de suas extremidades para indicar a orientação estabelecida. A Figura 3.40 traz a forma geométrica do vetor $\vec{v}$ representado pelo segmento orientado P_1P_2.

Figura 3.40: Representação geométrica de um vetor.



Fonte: Elaborada pelo autor.

Agora temos condições de definir formalmente uma translação:

Definição 3.6.3. *Dado um vetor $\vec{v} = (\alpha,\, \beta)$ em um plano Π, chamamos de **translação** determinada por $\vec{v}$ a transformação $T_{\vec{v}} : \Pi \longmapsto \Pi$ que associa a cada ponto $P(x, y)$ do plano Π, um ponto P' tal que $P' = T_v(P) = (x + \alpha,\, y + \beta)$.*

Proposição 3.6.1. *Uma translação é uma isometria.*

Demonstração. Sejam $A'(x_a + \alpha,\, y_a + \beta)$ e $B'(x_b + \alpha,\, y_b + \beta)$ as imagens dos pontos $A(x_a,\, y_a)$ e $B(x_b,\, y_b)$ pela translação $T_{\vec{v}}$ determinada pelo vetor $\vec{v} = (\alpha,\, \beta)$. Vamos mostrar que $\overline{AB} = \overline{A'B'}$.

Fazendo uso do Teorema 3.6.1, obtemos

$$\overline{AB} = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2} \tag{3.3}$$

e

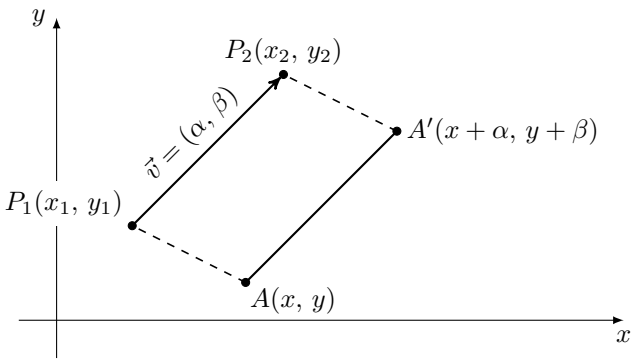
$$\begin{aligned}\overline{A'B'} &= \sqrt{(x_b+\alpha-(x_a+\alpha))^2+(y_b+\beta-(y_a+\beta))^2}\\ &= \sqrt{(x_b+\alpha-x_a-\alpha)^2+(y_b+\beta-y_a-\beta)^2}\\ &= \sqrt{(x_b-x_a)^2+(y_b-y_a)^2}. \qquad (3.4)\end{aligned}$$

Segue de (3.3) e (3.4) que $\overline{AB} = \overline{A'B'}$, logo $T_{\vec{v}}$ é uma isometria. □

Proposição 3.6.2. *Seja P_1P_2 um segmento representante de um vetor $\vec{v} = (\alpha, \beta)$. Se A' é a imagem do ponto A pela translação $T_{\vec{v}}$ determinada por $\vec{v}$, então $\overline{AA'} = \overline{P_1P_2}$ e $\overleftrightarrow{AA'} \parallel \overleftrightarrow{P_1P_2}$.*

Demonstração. Sejam A' a imagem de A pela translação $T_{\vec{v}}$ determinada pelo vetor $\vec{v} = (\alpha, \beta)$ e P_1P_2 um segmento representante de $\vec{v}$, conforme Figura 3.41.

Figura 3.41: Translações determinam segmentos congruentes e paralelos ao representante de $\vec{v}$.



Fonte: Elaborada pelo autor.

Pela definição de vetor e pelo Teorema 3.6.1, obtemos

$$\overline{P_1P_2} = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2} = \sqrt{\alpha^2+\beta^2}, \qquad (3.5)$$

$$\overline{AA'} = \sqrt{(x+\alpha-x)^2+(y+\beta-y)^2} = \sqrt{\alpha^2+\beta^2}, \qquad (3.6)$$

$$\overline{AP_1} = \sqrt{(x_1-x)^2+(y_1-y)^2} \qquad (3.7)$$

e

$$\begin{aligned}
\overline{A'P_2} &= \sqrt{(x_2 - (x+\alpha))^2 + (y_2 - (y+\beta))^2} \\
&= \sqrt{(x_2 - x - \alpha)^2 + (y_2 - y - \beta)^2} \\
&= \sqrt{(x_2 - x - (x_2 - x_1))^2 + (y_2 - y - (y_2 - y_1))^2} \\
&= \sqrt{(x_2 - x - x_2 + x_1)^2 + (y_2 - y - y_2 + y_1)^2} \\
&= \sqrt{(x_1 - x)^2 + (y_1 - y)^2} \qquad (3.8)
\end{aligned}$$

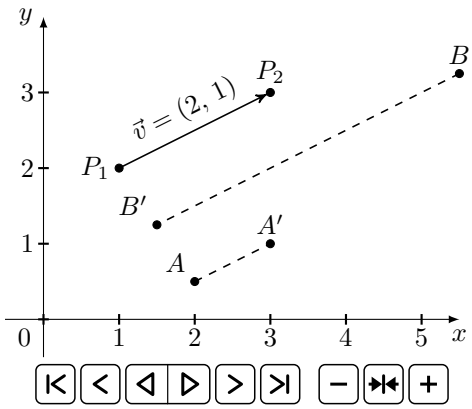
De (3.5) e (3.6), obtemos $\overline{AA'} = \overline{P_1P_2}$, de (3.7) e (3.8), obtemos $\overline{AP_1} = \overline{A'P_2}$, de sorte que $AA'P_2P_1$ é um quadrilátero com lados opostos congruentes e, portanto, é um paralelogramo, logo $\overleftrightarrow{AA'} \parallel \overleftrightarrow{P_1P_2}$. □

De acordo com a Proposição 3.6.2, numa translação, a distância entre os pontos e suas respectivas imagens é igual à medida do segmento representante do vetor considerado na translação. Para obtermos translações com distâncias diferentes entre ponto e imagem, basta tomar um novo vetor ou utilizar o que chamaremos de *múltiplo* do vetor considerado, que definiremos a seguir.

Definicão 3.6.4. *Dado um vetor $\vec{v} = (\alpha, \beta)$ e um número real não nulo λ, chamamos o vetor $\lambda\vec{v}$, dado por $\lambda\vec{v} = (\lambda \cdot \alpha, \lambda \cdot \beta)$, de* ***múltiplo*** *de $\vec{v}$.*

O valor de λ conserva a direção da translação, isto é, a translação continua a mover os pontos de forma paralela à reta suporte do segmento representante do vetor $\vec{v}$, porém modifica a distância entre os pontos e suas respectivas imagens, de forma que a imagem A' de um ponto A, determinada por uma translação segundo um vetor $\lambda\vec{v}$, é tal que $\overline{AA'} = |\lambda| \cdot \overline{P_1P_2}$ sendo P_1P_2 um representante de $\vec{v}$, e para valores negativos de λ, o sentido da translação é alterado. Observe na Figura 3.42 as translações dos pontos A e B determinadas, respectivamente, pelos vetores $\frac{1}{2}\vec{v}$ e $-2\vec{v}$ múltiplos de $\vec{v} = (2, 1)$.

Figura 3.42: Translação segundo múltiplos do vetor $\vec{v}$.



Fonte: Elaborada pelo autor.

O uso de múltiplos de vetores na realização de translações desempenha um papel de suma importância para a criação de desenhos e animações no Capítulo 5, pois será através deles que conseguiremos obter pontos específicos para criar e simular movimentos de figuras, assim como apresentado na Figura 3.42.

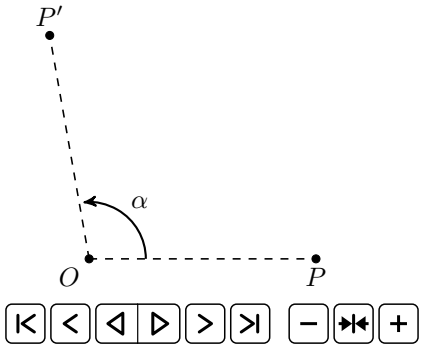
3.6.2 Rotação

Outra isometria que utilizaremos para a produção de animações é a *rotação*.

Definição 3.6.5. *Dado um ponto O e um ângulo de medida α, denomina-se* ***rotação*** *de centro O e amplitude α a transformação $R_{O,\alpha} : \Pi \longmapsto \Pi$ que associa a cada ponto P do plano Π o ponto $P' = R_{O,\alpha}(P)$ tal que $\overline{OP'} = \overline{OP}$ e cuja medida do ângulo $\angle POP'$ é α.*

Assumiremos que o ponto P' será obtido considerando o sentido anti-horário quando aplicada uma amplitude positiva, conforme animação da Figuras 3.43, e o sentido horário ao se utilizar uma amplitude negativa.

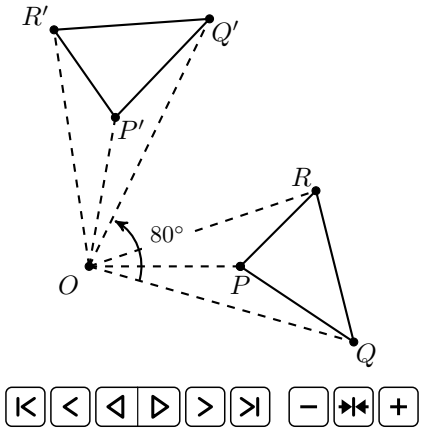
Figura 3.43: Rotação de centro O e amplitude α do ponto P.



Fonte: Elaborada pelo autor.

Quando aplicamos uma rotação em uma figura, obtemos outra congruente à primeira, isto é, a imagem de uma figura F por uma rotação é uma outra figura F' congruente a F. O que a rotação faz é simplesmente rotacionar a figura original para outra posição de acordo com o centro O e a amplitude considerada. Este fato é justificado por conta das rotações serem isometrias e, consequentemente, preservarem distâncias, paralelismo e ângulos, e transformarem retas em retas. A prova desses fatos podem ser encontradas em Lima [17]. Como exemplo, acompanhe a animação da Figura 3.44. Nela há uma rotação de centro O e amplitude $\alpha = 80°$ do triângulo PQR, onde tem-se $PQR \equiv P'Q'R'$.

Figura 3.44: Rotação de centro O e amplitude $80°$ do triângulo PQR.



Fonte: Elaborada pelo autor com base em Wagner [16].

3.6.3 Simetria

A definição da transformação simetria utiliza o conceito de ponto médio.

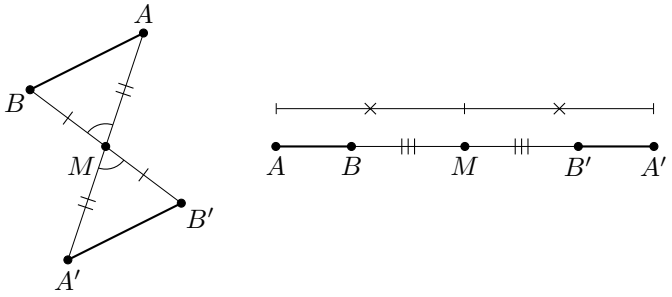
Definicão 3.6.6. *Dado um segmento AA' com $A \neq A'$, dizemos que M é o* ***ponto***

médio *de* AA' *se* $M \in AA'$ *e* $\overline{AM} = \overline{MA'}$. *Nesse caso, dizemos que* A' *é o* ***simétrico*** *de* A *em relação ao ponto* M.

Sabendo da definição de ponto médio, podemos agora apresentar a definição de simetria.

Definição 3.6.7. *Dado um ponto* M *no plano* Π, *definimos* ***simetria*** *em torno de* M *como sendo a transformação* $S_M : \Pi \longmapsto \Pi$ *que associa a cada ponto* P *do plano o ponto* $P' = S_M(P)$ *tal que* P' *é o simétrico de* P *em relação ao ponto* M *e para* $P = M$, $P' = S_M(P) = P$.

Figura 3.45: Simetria em torno do ponto M.



Fonte: Elaborada pelo autor.

Proposição 3.6.3. *Uma simetria é uma isometria.*

Demonstração. Sejam A' e B' as imagens dos pontos A e B pela simetria em torno de M. Quando pelo menos dois dos pontos A, B ou M coincidem não há o que provar. Sendo eles dois a dois distintos, consideraremos, então, dois casos. O primeiro quando A, B e M não são colineares, conforme Figura 3.45. Por definição, $\overline{AM} \equiv \overline{MA'} \equiv \overline{A'M}$ e $\overline{BM} \equiv \overline{MB'} \equiv \overline{B'M}$ e $\angle AMB \equiv \angle A'MB'$, pois eles são opostos pelo vértice. Portanto, pelo caso de congruência LAL, os triângulos AMB e $A'MB'$ são congruentes, logo $AB \equiv A'B'$.

No segundo caso, os pontos A, B e M são colineares e ambos pertencem à mesma semirreta de origem M. Quando eles estiverem em semirretas opostas, o raciocínio é análogo. Assim, basta fazer uso da definição de simetria e observar que

$$\begin{aligned} \overline{AM} = \overline{MA'} &\Longrightarrow \overline{AB} + \overline{BM} = \overline{MB'} + \overline{B'A'} \\ &\Longrightarrow \overline{AB} + \overline{BM} = \overline{BM} + \overline{B'A'} \\ &\Longrightarrow \overline{AB} = \overline{B'A'}. \end{aligned}$$

Em todo caso, $\overline{AB} = \overline{B'A'}$, o que mostra que uma simetria conserva distâncias, logo é uma isometria. □

Por ser uma isometria, a imagem de uma figura F por simetria é uma figura F' congruente a F. Porém, utilizaremos simetrias apenas para obtermos pontos auxiliares na produção das animações, pois, como veremos no Capítulo 5, os comandos em LaTeX que utilizaremos não possibilitam a simulação de movimento por meio de simetria. No entanto, se for necessário, pode-se utilizar as macros destinadas às translações para simular simetrias.

3.6.4 Reflexão

Para compreendermos a transformação denominada *reflexão*, precisamos de duas definições.

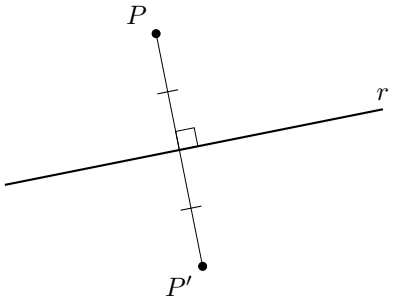
Definição 3.6.8. *Dado o segmento AB, definimos* ***mediatriz*** *de AB como sendo a reta r que passa pelo ponto médio de AB e que é perpendicular a $\overleftrightarrow{AB}$.*

Definição 3.6.9. *Dada uma reta r e um ponto A, diremos que A' é o* ***simétrico*** *de A em relação à reta r, quando r for a mediatriz de AA'. Se $A \in r$, diremos que A é o simétrico dele próprio em relação a r.*

De posse das Definições 3.6.8 e 3.6.9, definimos a transformação reflexão.

Definição 3.6.10. *Dada uma reta r no plano Π, chamamos de* ***reflexão*** *em torno de r a transformação $R_r : \Pi \longmapsto \Pi$ que associa a cada ponto P de Π o ponto $P' = R_r(P)$, tal que P' é o simétrico de P em relação à reta r.*

Figura 3.46: Reflexão do ponto P em torno da reta r.



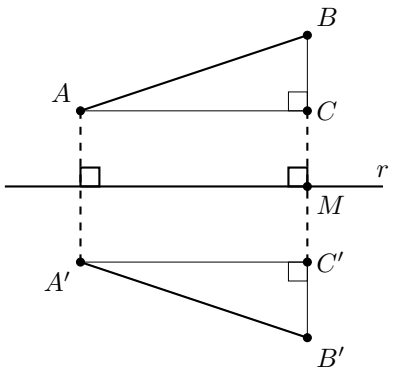
Fonte: Elaborada pelo autor com base em Lima [17].

Proposição 3.6.4. *Uma reflexão é uma isometria.*

Demonstração. Dados uma reta r e os pontos A e B, considere que A' e B' sejam as imagens desses pontos por reflexão em torno de r. Temos dois casos a considerar. O primeiro, quando A e B estão em um mesmo lado da reta r. Marque o ponto M de interseção entre r e $\overleftrightarrow{BB'}$ e trace duas paralelas à reta r, a que passa pelo ponto A e

a outra que passa por A' marcando, respectivamente, os pontos C e C' de interseção dessas retas com $\overleftrightarrow{BB'}$. Tome como base a Figura 3.47.

Figura 3.47: Reflexão de A e B em um mesmo lado de r.

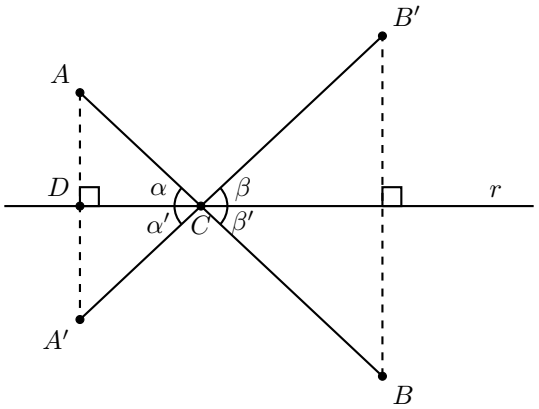


Fonte: Elaborada pelo autor com base em Lima [17].

Note que B' e C' são, respectivamente, as imagens de B e C por simetria em torno de M, e, como já sabemos que uma simetria preserva distâncias, segue que $\overline{BC} = \overline{B'C'}$. Por construção, $\hat{C}$ e $\hat{C'}$ são retos e $\overline{AC} = \overline{A'C'}$. Assim, pelo caso de congruência LAL, temos que $ABC \equiv A'B'C'$ e com isso $\overline{AB} = \overline{A'B'}$.

No segundo caso, os pontos A e B estão em lados opostos de r. Marque os pontos C e D de interseções da reta r com AB e AA' conforme a Figura 3.48.

Figura 3.48: Reflexão de A e B em lados opostos de r.



Fonte: Lima [17].

Os triângulos ADC e $A'DC$ compartilham o lado DC, são retângulos em D e, por conta da reflexão em torno de r, possuem os lados AD e $A'D$ congruentes. Logo, pelo caso de congruência LAL, temos que $ADC \equiv A'DC$ e, consequentemente, $\alpha = \alpha'$. Analogamente, conclui-se que $\beta = \beta'$. Por construção, $\alpha = \beta'$, pois são

medidas de ângulos opostos pelo vértice. Segue destas três igualdades que α, β, α' e β' são todos iguais, de forma que $\alpha + \alpha' = \beta + \beta'$.

Note que $\beta + \beta'$ é o que falta ao ângulo $\angle ACB'$ para 180° de modo que $\alpha + \alpha'$ também o é. Logo, A', C e B' são colineares e, portanto,

$$\overline{AB} = \overline{AC} + \overline{CB} = \overline{A'C} + \overline{CB'} = \overline{A'B'}.$$

Em todo caso uma reflexão conserva distância, consequentemente é uma isometria. □

Sendo uma reflexão uma isometria, a imagem de uma figura F por reflexão é outra figura F' congruente a F. Porém, assim como nas simetrias, os comandos que utilizaremos para animar com LaTeX não permitem simular movimento por meio de reflexões. Por isso, serão utilizadas apenas para obtermos pontos auxiliares para a construção de figuras. Todavia, se necessário, é possível utilizar as macros destinadas às translações para simular reflexões.

Capítulo 4

O programa LaTeX

Escrever equações e símbolos matemáticos por meio de editores de textos tradicionais como Microsoft Word ou Google Documents pode ser algo frustrante para um professor de Matemática do Ensino Médio, principalmente para aqueles mais perfeccionistas, pois as equações muitas vezes não agradam esteticamente ficando de forma desalinhada com o texto em que está inserida. Mas o que muitos não sabem é que existe um programa capaz de escrever textos simples, equações e símbolos de forma tão perfeita que chega a impressionar. Esse programa é o LaTeX.

Neste capítulo, apresentaremos o LaTeX ao leitor, destacaremos a importância do programa para a produção de trabalhos acadêmicos e científicos, instruiremos o que deve ser instalado no computador para que o programa funcione corretamente e apresentaremos instruções de manuseio de comandos básicos de forma a prepará-lo para produzir textos com desenhos e animações.

4.1 O que é o LaTeX?

Para entender o que é LaTeX, temos antes que entender o que é TeX. De acordo com Lourenço [18] e Oetiker et al. [19], TeX é um sistema computacional para edição de textos criado por Donald Ervin Knuth, entre os anos de 1977 e 1982, com o objetivo de explorar os potenciais dos equipamentos eletrônicos de impressão que começavam a ser utilizados nas publicações acadêmicas daquele tempo.

Ainda de acordo com Lourenço [18], para produzir textos de qualidade com o TeX, eram necessários conhecimentos específicos de editoração. A fim de contornar essa barreira, isto é, tornar a produção de texto mais acessível ao público, Leslie Lamport desenvolveu o LaTeX, no ano de 1985, que adicionou novas configurações ao TeX. Por meio do LaTeX tornou-se possível produzir textos de qualidade profissional, assim como era com o TeX, mas de forma mais simples.

Para compreendermos melhor a relação entre LaTeX e o TeX, destacamos de

Oetiker et al. [19], a seguinte metáfora:

> Para publicar alguma coisa, os autores dão um manuscrito dactilografado à companhia de publicação. Um dos seus paginadores decide o formato do documento (largura da coluna, tipos de letra, espaços antes e após os cabeçalhos,...). Este escreve as suas instruções no manuscrito, que é entregue ao tipógrafo que imprime o livro de acordo com estas instruções. [...] Num ambiente LaTeX, o paginador é o LaTeX, que usa o TeX como seu tipógrafo.

O LaTeX tem uma grande diferença dos editores de textos comuns e mais populares atualmente como Microsoft Word, LibreOffice Writer e Google Documents. Esses editores utilizam um sistema de escrita conhecido como WYSIWYG, fazendo alusão às iniciais das palavras na frase em inglês "What You See Is What You Get", que significa "o que você vê é o que você recebe", isto é, nesses editores escrevemos e vemos o resultado de forma imediata, pois o resultado é o que estamos escrevendo. Já o LaTeX utiliza o sistema WYSIWYM, que é acrônimo da frase em inglês "What You See Is What You Mean", que em tradução livre significa "o que você vê é o que você quer dizer". Nesse tipo de escrita, o que estamos escrevendo não representa o documento em si, mas sim o que queremos que o computador transforme em documento (geralmente em formato PDF), e, por esse motivo, além de escrevermos o texto do documento, utilizamos vários comandos (também chamados de macros), semelhantes à linguagem de programação, para orientar o computador nesse processo.

Outra grande diferença do LaTeX para os programas WYSIWYG, é que o LaTeX requer, no mínimo, dois *softwares* para que consigamos escrever: um compilador e um editor. O editor é utilizado para escrever os comandos e o texto que formará o documento, enquanto o compilador é responsável por gerenciar e instalar no computador os pacotes de comandos que são utilizados no editor.

Existem plataformas que possibilitam a utilização do LaTeX de forma *online*, como o Overleaf e o Papeeria, sem a necessidade de instalação de nenhum *software*, sendo necessário apenas realizar um registro de usuário. No entanto, para a criação de animações, a utilização do LaTeX por meio destas plataformas não é recomendada, visto que, como veremos na Seção 4.3, as animações só funcionam em leitores de PDF específicos e isso exige que, toda vez que se queira testar a animação produzida, será necessário realizar o *download* do arquivo PDF, isso pode representar um empecilho. Mesmo assim, se o leitor desejar utilizar uma das plataformas citadas, recomendamos os endereços eletrônicos www.overleaf.com ou www.papeeria.com.

Pelo fato de o LaTeX ser de domínio livre, qualquer um que tenha conhecimento para tal, pode desenvolver novos pacotes. Por isso, desde a sua criação, já foram desenvolvidos e disponibilizados de forma gratuita vários desses pacotes que pos-

suem macros para executar as mais diversas funções, tais como: inserção de figuras, produção de figuras geométricas, criação de gráficos e até produção de animações. Sendo assim, não é justo afirmar que o LaTeX seja um simples editor de texto, pois atualmente ele faz muito mais do que isso. É mais justo defini-lo como um sistema de produção de textos ou elementos gráficos que usa o TeX como base de formatação.

4.2 A importância e vantagens do uso do LaTeX

No LaTeX existem comandos que determinam, entre outras coisas, a estrutura de um documento como margens, distância entre linhas, notas de rodapé, numeração das páginas, sumários e citações, de forma a possibilitar a criação de modelos (também chamados de *templates*) que são de fácil compartilhamento. Essa possibilidade, juntamente com o fato do LaTeX construir documentos de qualidade profissional, fazem com que ele seja constantemente utilizado em eventos científicos, como em congressos, onde os organizadores disponibilizam *templates* para que os participantes os utilizem para produção dos trabalhos e os submetam para publicação.

É notória a importância da utilização do LaTeX para produção de textos científicos/acadêmicos. Mas por que utilizá-lo no Ensino Médio em vez de usar programas como Microssoft Word? Para responder essa pergunta, podemos nos basear no texto da BNCC [5], o qual trata da ocupação desse nível de ensino em desenvolver nos estudantes competências e habilidades que os permitam, dentre outras coisas:

> usar diversas ferramentas de software e aplicativos para compreender e produzir conteúdos em diversas mídias, simular fenômenos e processos das diferentes áreas do conhecimento, e elaborar e explorar diversos registros de representação matemática;

Além disso, a BNCC [5] exige que se deve desenvolver no aluno a habilidade de "utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática".

Com base nisso, vemos no LaTeX uma maneira de apresentar aos professores do Ensino Médio uma possibilidade de utilizarem uma nova ferramenta que além de ser capaz de produzir conteúdos com escrita matemática em qualidade profissional, desperta a busca por conhecimento sobre linguagem de programação o que não acontece com a utilização do Microsoft Word e similares.

Além da possibilidade de despertar interesse sobre a linguagem de programação TeX, o LaTeX possui outras vantagens em relação ao Microsoft Word. Lourenço [18] constrói uma lista de nove vantagens, das quais destacamos três: a edição de fórmulas matemáticas no LaTeX é mais robusta do que no Word, favorecendo a harmonização visual e agilidade na construção das mesmas; a possibilidade de incorporação de objetos multimídias como imagens, vídeos, áudios e animações no

arquivo de formato PDF; e a gratuidade dos programas utilizados.

É válido ressaltar que a segunda vantagem listada acima, possibilitou a produção deste trabalho acadêmico.

4.3 Instalando o LaTeX

Como vimos na Seção 4.1, para escrevermos em LaTeX precisamos de no mínimo dois programas: o compilador e o editor. Porém, como pretendemos produzir animações com o LaTeX em arquivos no formato PDF, devemos instalar um programa de leitor desse tipo de arquivo para visualizarmos as animações, pois, de acordo com Grahn [20], só poderemos visualizá-las com um dos seguintes leitores de PDF instalados no computador:

1. Adobe Acrobat Reader;
2. KDE Okular;
3. PDF-XChange;
4. Foxit Reader.

Avaliamos a funcionalidade das animações nos leitores de PDF listados, testando-os em computadores que possuem sistemas operacionais Windows e Linux (não testamos as animações com o sistema MacOS). Constatamos que, no sistemas Linux, as animações não funcionam em todos os leitores. A Tabela 4.1 informa em qual sistema operacional e sob qual leitor as animações funcionam corretamente.

Tabela 4.1: Leitores de PDF que reproduzem animações.

Sistema/Leitor	***Windows***	***Linux***
Adobe A. Reader	✓	×
KDE Okular	✓	✓
PDF-XChange	✓	×
Foxit Reader	✓	×

Legenda: ✓ – reproduz a animação;
× – não reproduz a animação ou indisponível para o sistema operacional.

Fonte: Elaborado pelo autor.

Em relação aos dispositivos móveis (celulares e tablets), as animações não funcionam em nenhum dos leitores de PDF listados. Portanto, as animações só funcionam em computadores com sistemas operacionais e leitores de PDF conforme a Tabela 4.1.

Nesta seção, trataremos das opções de compiladores e editores disponíveis para download, e apresentaremos um tutorial para instalação do compilador MiKTeX, do editor TeXstudio e do leitor de PDF Adobe Acrobat Reader.

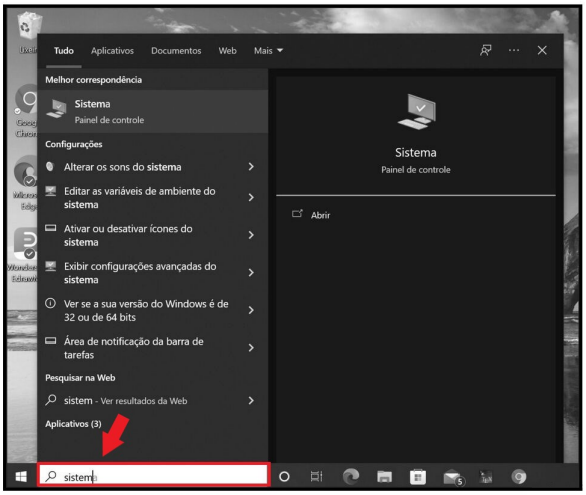
4.3.1 Instalando o compilador MiKTeX

Existem vários compiladores que podem ser usados no TEX, dos quais destacamos três: MiKTeX, TeXLive e o MacTeX. A escolha de um deles depende do gosto e/ou do sistema operacional que se utiliza no computador. O MiKTeX pode ser utilizado no Windows, Linux ou sistemas MacOS, o TeXLive pode ser usado em Windows ou Linux e o MacTeX é dedicado apenas ao sistema MacOS. O *download* desses programas pode ser realizado, respectivamente, em Miktex [21], Texlive [22] e Mactex [23].

Por ser possível instalar nos três sistemas operacionais, utilizaremos neste trabalho o MiKTeX e vamos explicar como instalá-lo no Windows. Orientações sobre como instalar o MiKTeX em sistemas MacOS ou Linux podem ser encontradas em MikTeX [21].

Existem sistemas Windows que possuem processadores baseados em 32 bits e outros em 64 bits. O primeiro passo é descobrir qual desses processadores a sua máquina possui. Esta identificação é importante, pois para instalar o MiKTeX devemos baixar arquivos diferentes a depender do tipo de processador. Para obter a informação se o processador usa 32 ou 64 bits, digite "sistema" no local de pesquisa da barra de tarefas da área de trabalho, como na Figura 4.1.

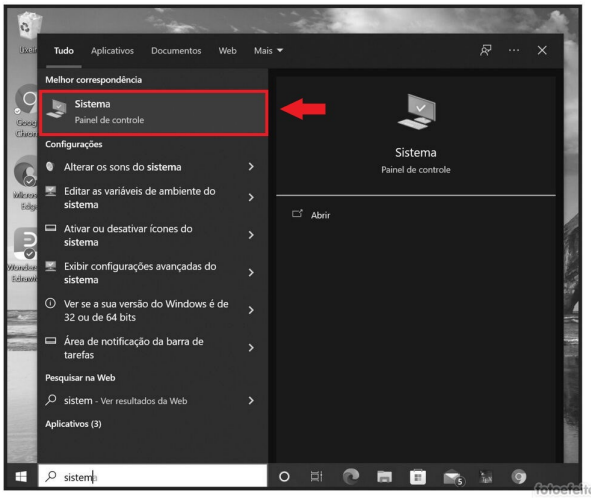
Figura 4.1: Identificando o tipo de sistema operacional 01.



Fonte: Elaborada pelo autor.

Clique na opção "Sistema" que aparece como um dos resultados da pesquisa.

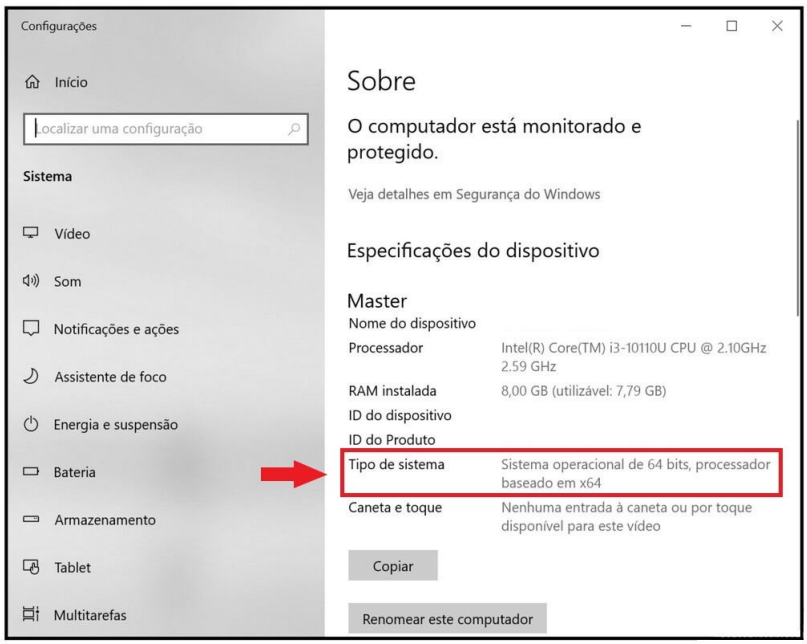
Figura 4.2: Identificando o tipo de sistema operacional 02.



Fonte: Elaborada pelo autor.

Após clicar em "Sistema" será exibida uma tela com as especificações do computador semelhante à imagem da Figura 4.3, onde é possível identificar sobre qual sistema o processador funciona: se é baseado em 32 ou 64 bits.

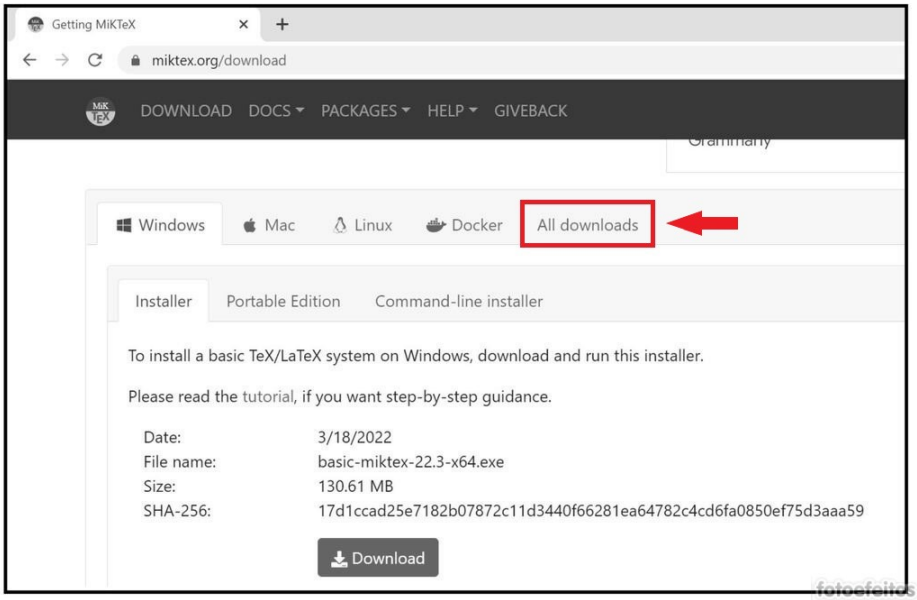
Figura 4.3: Identificando o tipo de sistema operacional 03.



Fonte: Elaborada pelo autor.

Após a identificação do tipo de sistema, acesse o site disponível em MikTeX [21] e clique em "*all downloads*" ou em português "todos os downloads".

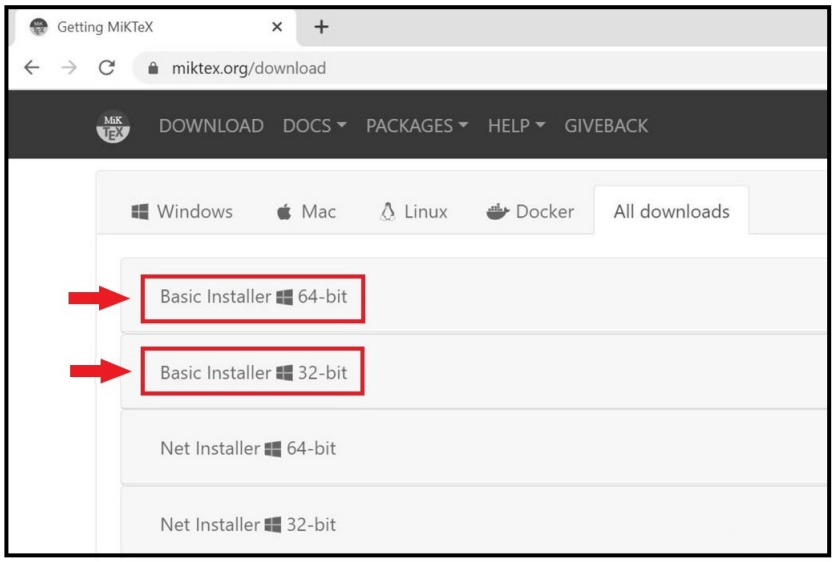
Figura 4.4: Baixando o MiKTeX 01.



Fonte: Elaborada pelo autor.

Será exibida uma lista de links para download do MiKTeX, organizada de acordo com os sistemas operacionais. Escolha "*Basic Installer 64-bit*" ou "*Basic Installer 32-bit*", conforme o tipo de sistema em que seu processador funciona. Observe, na Figura 4.5, a ilustração das duas escolhas mencionadas.

Figura 4.5: Baixando o MiKTeX 02.

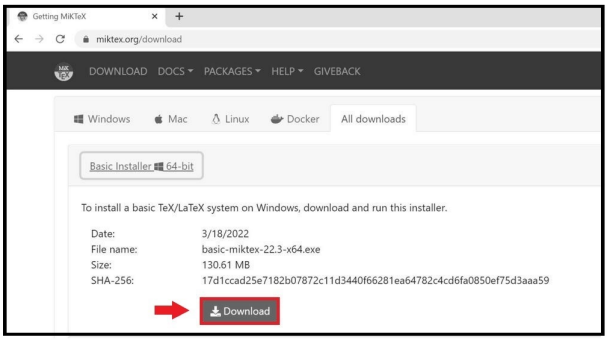


Fonte: Elaborada pelo autor.

Ao clicar em uma das opções (escolha a adequada para seu sistema) aparecerá

um botão "*Download*", clique nele que o MiKTeX será baixado automaticamente. Considere a Figura 4.6 como base.

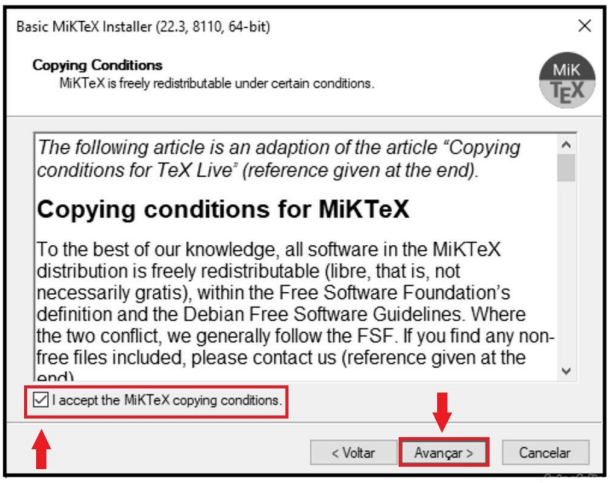
Figura 4.6: Baixando o MiKTeX 03.



Fonte: Elaborada pelo autor.

Após o arquivo ser baixado completamente, abra-o para iniciar o processo de instalação. Após aberto, será exibida na tala a janela representada na Figura 4.7, onde deve-se marcar o quadradinho da opção "*I accept the MiKTeX copying conditions*" para aceitar as condições de uso do MiKTeX e clicar em avançar para ir à próxima tela.

Figura 4.7: Instalando o MiKTeX 01.

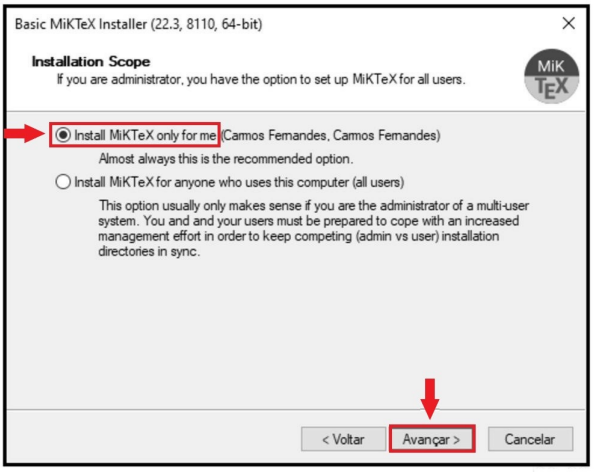


Fonte: Elaborada pelo autor.

É comum que em um computador com Windows se tenha vários perfis de usuários. Assim, após aceitar as condições de uso do programa, será exibida a tela representada na Figura 4.8, onde devemos escolher se a instalação será privada ou compartilhada, isto é, se o uso e administração do programa será de uso restrito ao perfil que o está instalando, ou se será compartilhada com todos os perfis presentes

no computador. Escolha a opção que se adéqua às suas necessidades. A Figura 4.8 ilustra a escolha de instalação privada. Após marcar uma das opções, clique em avançar.

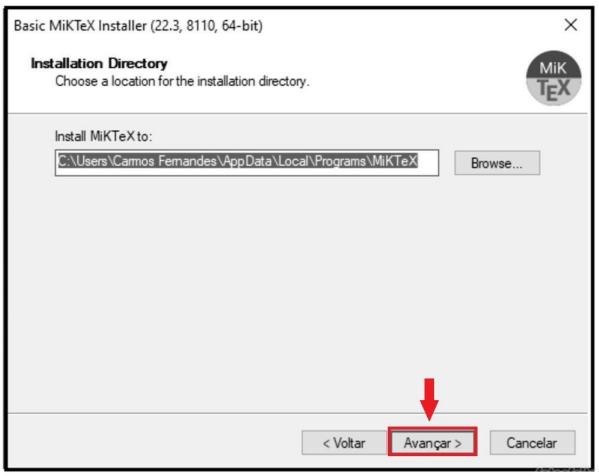
Figura 4.8: Instalando o MiKTeX 02.



Fonte: Elaborada pelo autor.

Na próxima tela escolhemos onde o programa será instalado. É apresentado de forma automática um local semelhante ao que aparece na Figura 4.9. Se quiser instalar o MiKTeX em outra pasta, clique em "*Browse*" e escolha uma pasta de sua preferência. Quando estiver satisfeito com o local de instalação clique em "Avançar".

Figura 4.9: Instalando o MiKTeX 03.

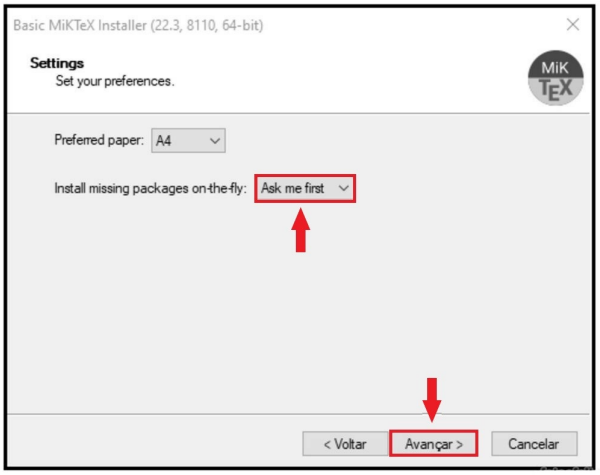


Fonte: Elaborada pelo autor.

Estamos instalando a versão básica do MiKTeX e, por isso, é possível que alguns pacotes estejam ausentes. Além disso, frequentemente são criados novos pacotes ou

os existentes sofrem atualizações. Assim, na próxima tela escolhemos como queremos que o MiKTeX realize essas atualizações. Há três opções: escolha "*Ask me first*" se desejar que o programa peça permissão para instalar os pacotes ausentes, escolha "*Always*" para que os pacotes ausentes sejam instalados sem pedir confirmação ou escolha "*Never*" para nunca instalar pacotes ausentes. Aconselha-se escolher a primeira opção, pois quando tentarmos usar um pacote que não esteja instalado, será pedida permissão para instalá-lo, o que será feito se o computador estiver conectado à internet. Portanto, escolha de acordo com a Figura 4.10 e clique em "Avançar".

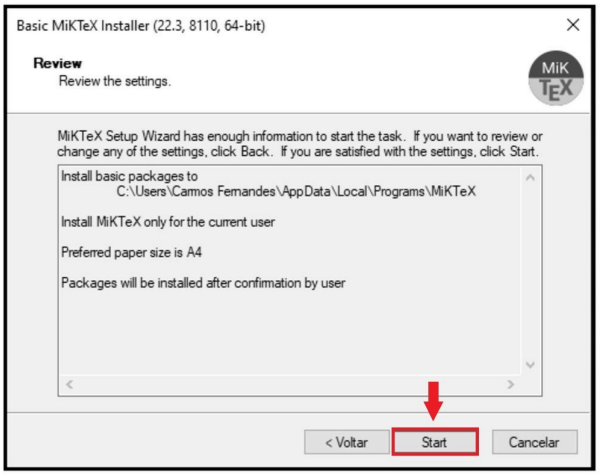
Figura 4.10: Instalando o MiKTeX 04.



Fonte: Elaborada pelo autor.

Na próxima tela será exibido um resumo das escolhas realizadas nos passos anteriores. Se estiver tudo certo, clique em *"Start"* para iniciar a instalação.

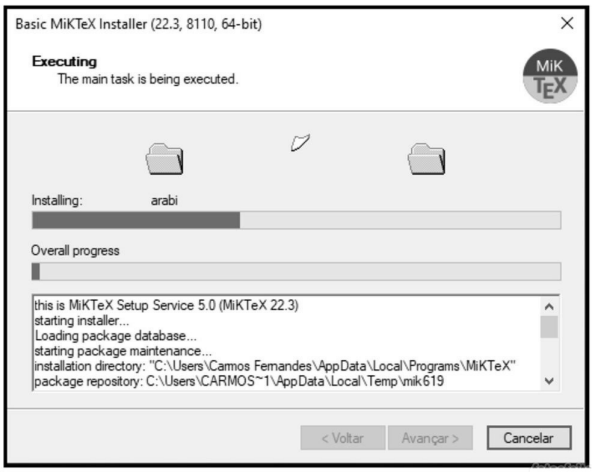
Figura 4.11: Instalando o MiKTeX 05.



Fonte: Elaborada pelo autor.

Após clicar em *"Start"* será exibida uma tela semelhante à Figura 4.12 mostrando o progresso da instalação.

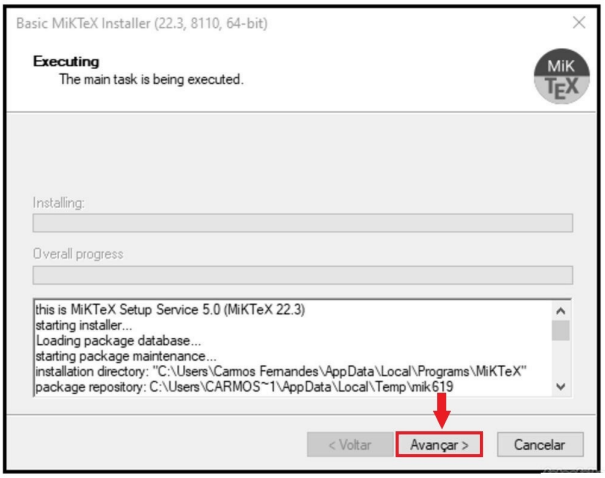
Figura 4.12: Instalando o MiKTeX 06.



Fonte: Elaborada pelo autor.

Após o progresso de instalação finalizar, clique em "Avançar", conforme a Figura 4.13.

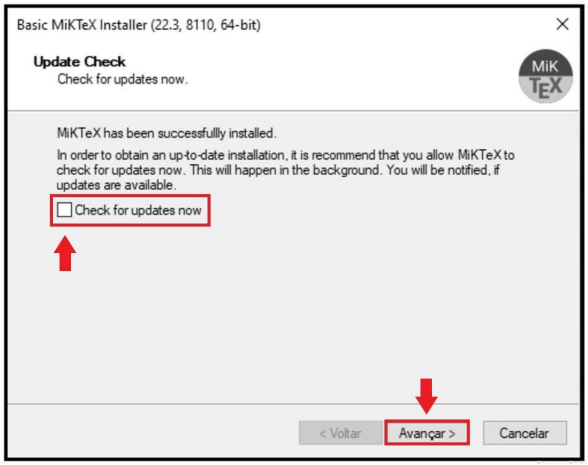
Figura 4.13: Instalando o MiKTeX 07.



Fonte: Elaborada pelo autor.

A próxima tela exibida é destinada à checagem de atualizações do programa. Como estamos instalando a versão mais recente, não é necessário procurar por atualizações. Portanto, desmarque a opção *"Check for updates now"* e clique em Avançar.

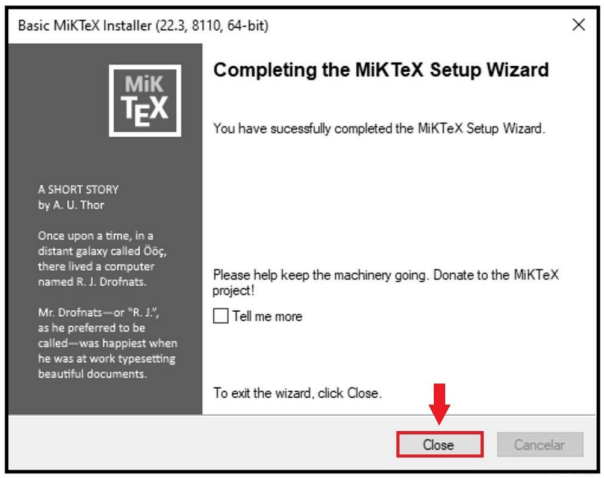
Figura 4.14: Instalando o MiKTeX 08.



Fonte: Elaborada pelo autor.

Finalmente, a última tela será exibida. Clique em "*Close*" e o MiKTeX estará instalado..

Figura 4.15: Instalando o MiKTeX 09.



Fonte: Elaborada pelo autor.

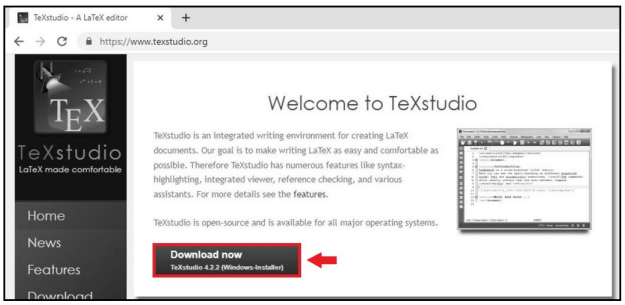
4.3.2 Instalando o editor TeXstudio

O MiKTeX é responsável por instalar os pacotes no computador. Ele funciona como um gerenciador e, portanto, é executado em segundo plano. O programa que utilizaremos de forma explícita é o editor, onde escrevemos textos e as macros (códigos) instaladas por meio do MiKTeX. Por isso, após a instalação do MiKTeX, devemos instalar o editor.

Existem vários editores LaTeX, dos quais destacamos três: TeXstudio, TeXmaker e o TeXworks, os quais podem ser obtidos, respectivamente, em [24], [25] e [26]. Todos eles podem ser instalados em computadores que utilizam sistemas operacionais Windows, Linux ou MacOs, porém, o TeXstudio se sobressai por possuir recursos que facilitam a escrita, como realce de erros ortográficos e de sintaxe, dicionário, assistência que sugere o autopreenchimento de comandos, etc. Por esse motivo utilizaremos o TeXstudio.

A instalação do TeXstudio é muito simples. Primeiro acesse o site disponível em [24] e clique no botão de download ilustrado na Figura 4.16 para baixar o arquivo de instalação.

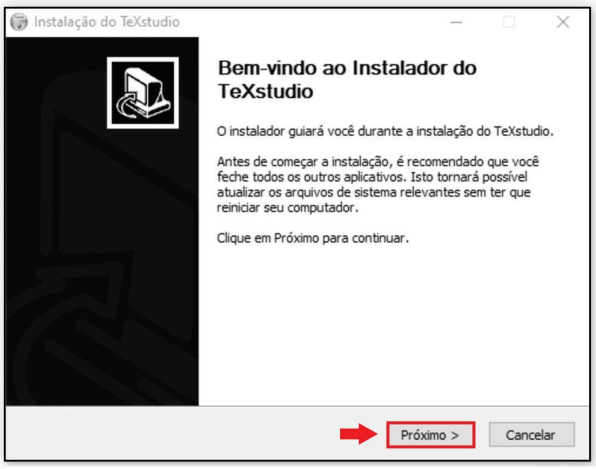
Figura 4.16: Instalando o TeXstudio 01.



Fonte: Elaborada pelo autor.

Após baixar o arquivo, abra-o que será exibida a tela representada na Figuras 4.17. Clique em “Próximo”.

Figura 4.17: Instalando o TeXstudio 02.

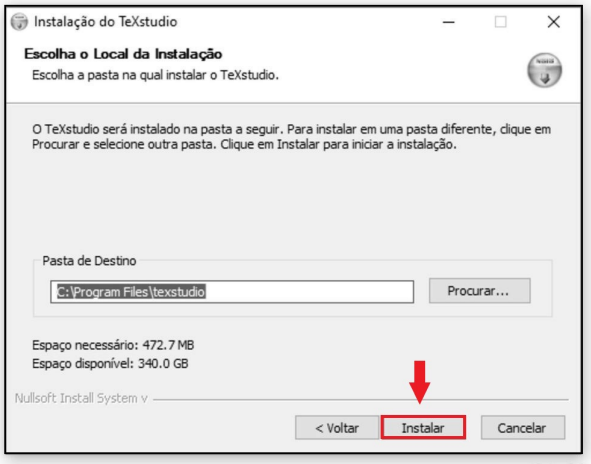


Fonte: Elaborada pelo autor.

A próxima tela é destinada à escolha do diretório da pasta em que será instalado

o TeXstudio. Clique em “Instalar”, se estiver satisfeito com a pasta padrão indicada na tela, ou clique em “Procurar”, para escolher outra pasta.

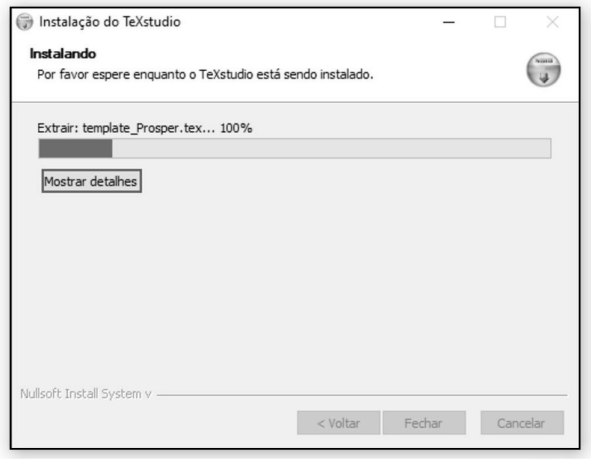
Figura 4.18: Instalando o TeXstudio 03.



Fonte: Elaborada pelo autor.

Após clicar em “Instalar”, será iniciado o processo de instalação, conforme a Figura 4.19.

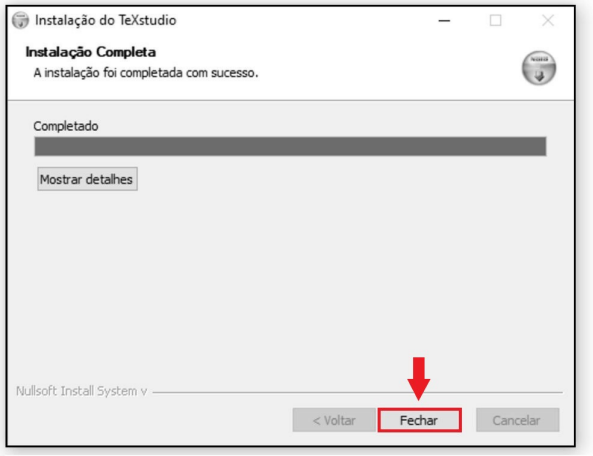
Figura 4.19: Instalando o TeXstudio 04.



Fonte: Elaborada pelo autor.

Quando a barra de progresso da instalação estiver totalmente preenchida, o TeXstudio estará instalado. Clique em “Fechar”.

Figura 4.20: Instalando o TeXstudio 05.



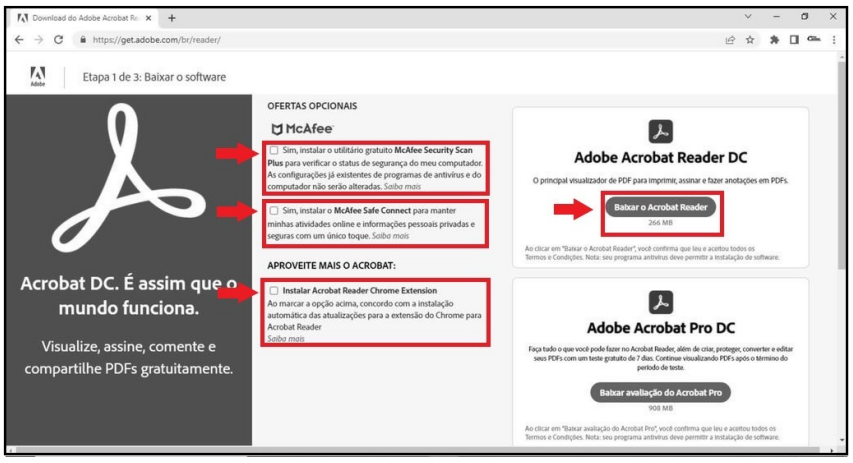
Fonte: Elaborada pelo autor.

4.3.3 Instalando o leitor de PDF Adobe Acrobat Reader

Como veremos, dentro do TeXstudio há um visualizador interno de PDF, porém, não é possível executar as animações por meio deste visualizador. Para isso é necessário que utilizemos um dos visualizadores externos mencionados no início desta seção. Da lista apresentada, escolhemos o Adobe Acrobat Reader por ser o mais conhecido e mais simples de instalar.

Para baixar o arquivo de instalação do Adobe Acrobat Reader, acesse o site disponível em Adobe [27] para ter acesso à tela representada na Figura 4.21.

Figura 4.21: Instalando o Adobe 01.



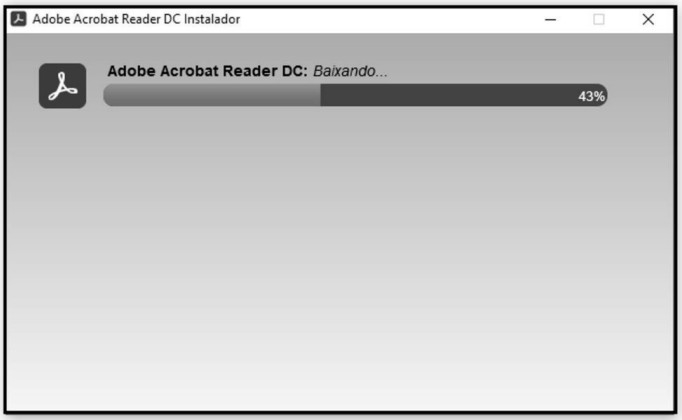
Fonte: Elaborada pelo autor.

Desmarque os quadradinhos indicados pelas três setas ao centro da imagem para

que seja instalado apenas o Adobe Acrobat Reader e clique no botão “Baixar o Acrobat Reader”.

Após o arquivo ser baixado completamente, abra-o para que a instalação seja iniciada. Será apresentada uma tela semelhante à Figura 4.22.

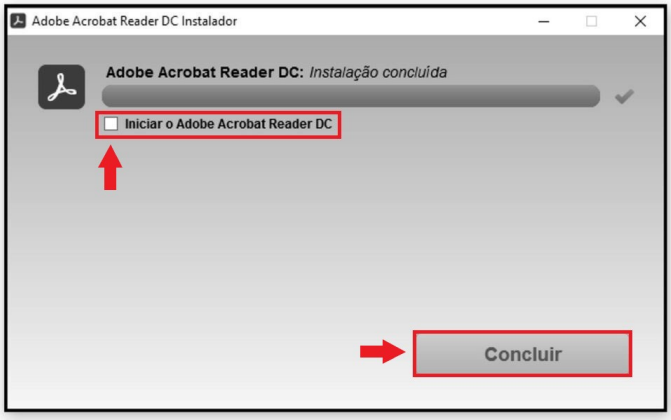
Figura 4.22: Instalando o Adobe 02.



Fonte: Elaborada pelo autor.

Espere a barra de progresso preencher por completo até aparecer a tela representada pela Figura 4.23.

Figura 4.23: Instalando o Adobe 03.



Fonte: Elaborada pelo autor.

Desmarque a opção “Iniciar o Adobe Acrobat Reader DC” e clique em “Concluir”.

4.4 Configurando e conhecendo o TeXstudio

Como dito anteriormente, o TeXstudio possui um visualizador integrado de PDF, porém, ele não serve para executar as animações que pretendemos produzir. Por isso, devemos configurar o TeXstudio para que utilize o Adobe Acrobat Reader como visualizador de PDF externo. Para realizar essa configuração, siga as instruções a seguir.

Abra o TeXstudio, para ter acesso às configurações do programa.

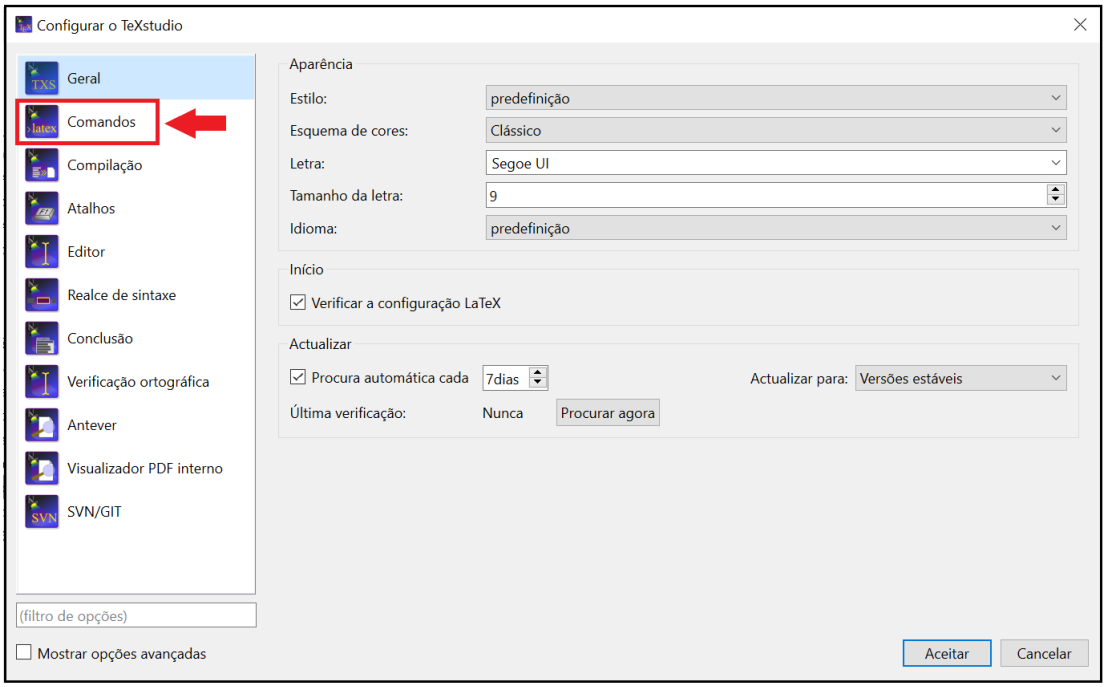
Ao abrir o TeXstudio, clique em "Opções" e posteriormente em "Configurar o TeXstudio". Veja na Figura 4.24 onde encontrar estas opções.

Figura 4.24: Configurando o TeXstudio 01.

Fonte: Elaborada pelo autor.

Será exibida a janela da Figura 4.25. Clique em "Comandos".

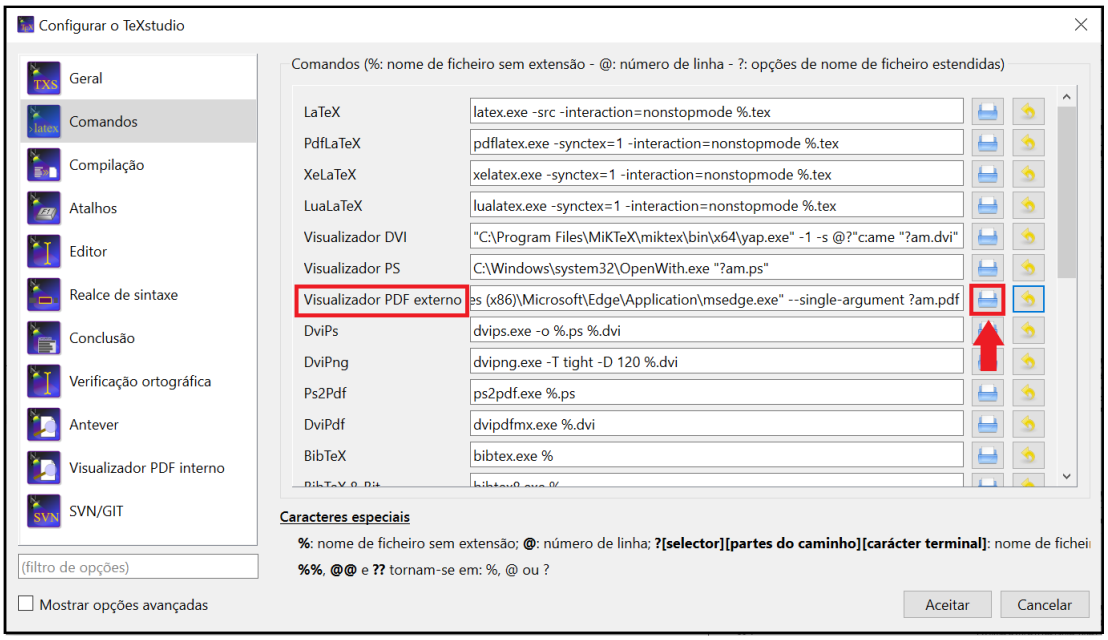
Figura 4.25: Configurando o TeXstudio 02.



Fonte: Elaborada pelo autor.

Será exibida uma lista com vários diretórios de arquivos que estarão preenchidos de forma automática pelo TeXstudio. Estes arquivos são, em sua maioria, oriundos do MiKTeX. Procure por "Visualizador PDF externo" e verifique se o diretório relacionado termina em "Acrobat.exe". Se assim for, ele leva ao programa Adobe Acrobat Reader. Caso contrário, clique na pasta indicada na Figura 4.26.

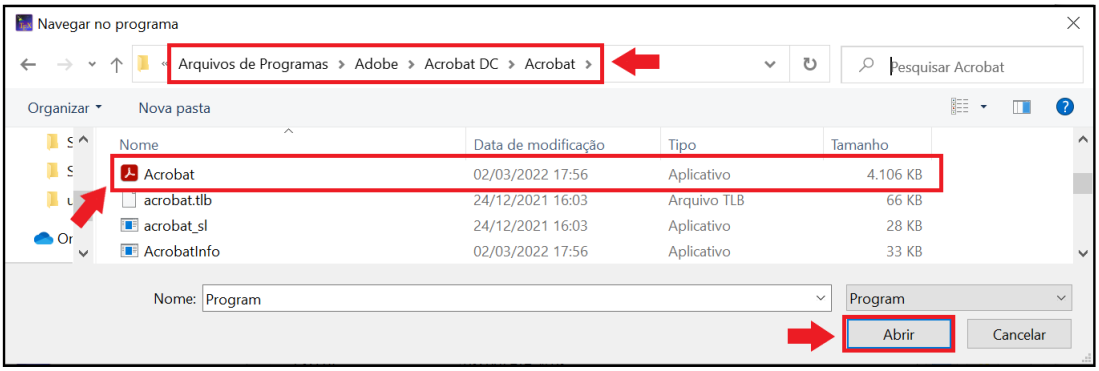
Figura 4.26: Configurando o TeXstudio 03.



Fonte: Elaborada pelo autor.

Nas telas que aparecerão, siga a sequência de pastas: Arquivos e Programas, Adobe, Acrobat DC e Acrobat. Esta sequência leva ao programa Adobe Acrobat Reader. Considere a Figura 4.27 como referência e clique em "Abrir".

Figura 4.27: Configurando o TeXstudio 04.

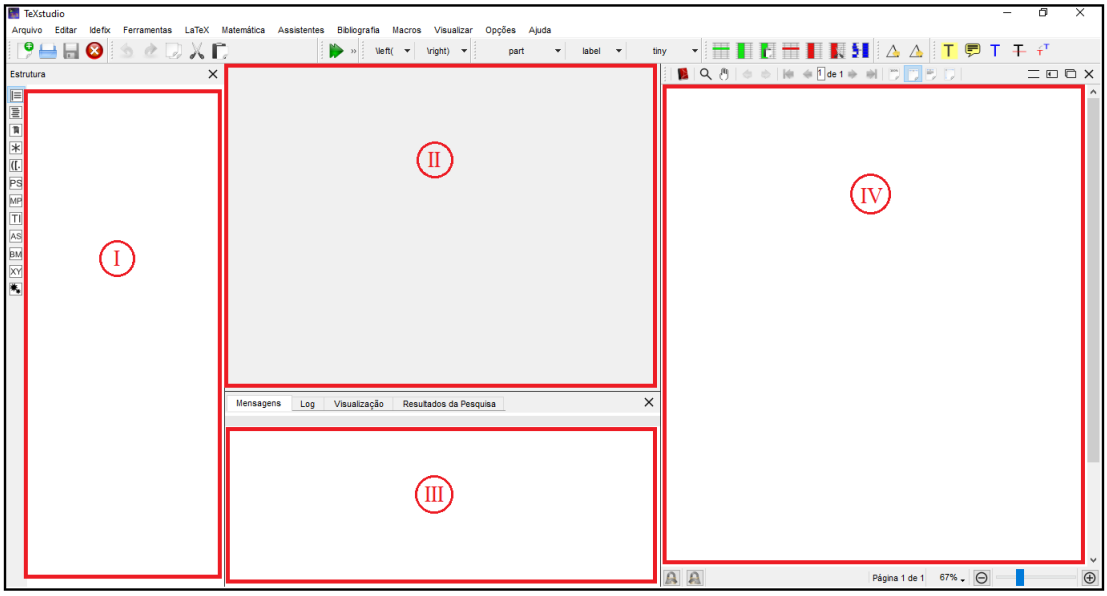


Fonte: Elaborada pelo autor.

Após este passo, o TeXstudio estará pronto para uso.

A interface do TeXstudio é dividida em quatro regiões, as quais estão representadas na Figura 4.28.

Figura 4.28: Conhecendo o TeXstudio 01.



Fonte: Elaborada pelo autor.

Na região I, aparecem os arquivos TEX caracterizados pelo final ".tex". Para criar um desses arquivos, clique em "Novo", representado pelo ícone da Figura 4.29.

Figura 4.29: Conhecendo o TeXstudio 02.

Fonte: Elaborada pelo autor.

Em seguida, clique no ícone "Gravar", representado pelo botão da Figura 4.30, para salvar o arquivo TEX. Escolha um nome para o arquivo e uma pasta onde o arquivo será salvo.

Figura 4.30: Conhecendo o TeXstudio 03.

Fonte: Elaborada pelo autor.

Após salvar o arquivo TEX, ele aparecerá na região I da Figura 4.28 com o nome escolhido no ato do salvamento.

Na região II, digitamos os códigos e os textos que controlam e compõem o arquivo PDF que se pretende produzir (ver Seção 4.5). Lembrando que o LATEX é um sistema WYSIWYM, o que digitamos nesta região não representará o arquivo em si, mas sim o que queremos que seja posto no PDF conforme especificamos com os comandos apropriados. Porém, a visualização do arquivo PDF pode ser feita a qualquer momento, clicando no botão "Compilar" representado pelo ícone da Figura 4.31.

Figura 4.31: Conhecendo o TeXstudio 04.

Fonte: Elaborada pelo autor.

Ao clicar no botão "Compilar", será iniciado o processo de compilação, isto é, o computador dará início ao processo de criação do PDF seguindo os comandos escritos na região II da Figura 4.28. Neste momento, será exibida na região III uma mensagem indicando que a compilação foi iniciada. Ao final da compilação, se estiver tudo certo com os códigos, será exibida uma mensagem informando que o processo foi finalizado normalmente, ou uma mensagem de erro, se existir algo de errado com os códigos digitados. No último caso, deve-se identificar o erro e consertá-lo, para recompilar o arquivo. Caso esteja tudo certo, o arquivo PDF será exibido na região IV com os resultados esperados.

A compilação do código é realizada por um compilador TEX. Atualmente os quatro compiladores mais utilizados são: *LaTeX*, *PdfLaTeX*, *LuaLaTeX* e *XeLaTeX*. A escolha de um compilador se relaciona diretamente com os pacotes utilizados, os tipos de imagens utilizadas no documento e o tipo de arquivo que se pretende produzir. O compilador *LaTeX* suporta somente imagens do tipo EPS (*Encapsulated PostScript*), o *PdfLaTeX* suporta imagens no formato PNG (*Portable Network Graphics*), JPEG (*Joint Photographics Experts Group*) e PDF. Ambos os compiladores, *LuaLaTeX* e *XeLaTeX*, suportam os quatro tipos de imagens citados. Dependendo dos comandos e pacotes que se utiliza na produção de um documento TEX pode ser necessário alterar o compilador. No entanto, para o que pretendemos produzir nesta dissertação, o compilador *PdfLaTeX* é suficiente, sendo que o TeXstudio é pré-configurado para utilizá-lo, e por isso não é necessário alterar esta configuração. Para informações sobre como alterar o compilador no TeXstudio, consulte a Seção A.3 do Apêndice A.

Para usar o leitor de PDF externo, basta, após a compilação, clicar no botão

"Visualizador Externo" representado pelo ícone da Figura 4.32, que o arquivo PDF será aberto automaticamente com o leitor externo sendo, no nosso caso, o Adobe Acrobat Reader.

Figura 4.32: Conhecendo o TeXstudio 05.

Fonte: Elaborada pelo autor.

Apresentamos nesta seção apenas o essencial para utilização do TeXstudio. Ele possui várias outras funções que facilitam a digitação e inserção de comandos. Encorajamos o leitor a explorar os ícones do programa, passando o mouse sobre eles e lendo suas descrições. Eles são, em sua maioria, autoexplicativos. Na próxima seção, exploraremos os comandos imprescindíveis para a criação de arquivos em PDF por meio do LaTeX. Todos os comandos e textos apresentados nos exemplos devem ser escritos na região representada por II na Figura 4.28.

4.5 Primeiros comandos

Um arquivo TeX é dividido em duas partes: o preâmbulo e o corpo do documento. No preâmbulo, escrevemos os comandos que cuidam da arquitetura geral do documento como margens, tamanho e tipo da fonte da letra utilizada. No corpo do documento, escrevemos tudo o que compõe o arquivo PDF (textos, figuras, tabelas, gráficos, etc) e comandos que são utilizados para formatação específica de um trecho do documento, por exemplo, a inclusão de texto em negrito, itálico ou sublinhado.

O corpo do documento é iniciado com `\begin{document}` e finalizado com `\end{document}`, tudo o que vier antes de `\begin{document}` é considerado como preâmbulo, o que estiver entre estes comandos formará o PDF e tudo o que estiver depois de `\end{document}` será totalmente ignorado.

O primeiro comando a ser escrito em um arquivo TeX é

```
\documentclass[opções]{classe}.
```
(4.1)

Este comando é obrigatório e tem a função de especificar a classe do documento que se pretende construir. Tal classe deve ser especificada entre chaves e escolhida conforme a finalidade do documento que se pretende produzir. A Tabela 4.2 traz as principais classes disponíveis e suas indicações, em especial, a classe *standalone* que utilizaremos no Capítulo 5.

Tabela 4.2: Classes de Documentos.

Classe	***Indicação***
article	Indicados para artigos científicos e pequenos relatórios.
report	Indicado para relatórios longos, pequenos livros e teses.
book	Indicado para livros longos.
slides	Indicado para criação de apresentações em slides (pouco utilizado).
beamer	Indicado para criações de slides de apresentações (muito utilizado).
standalone	Indicado para criação de desenhos geométricos e gráficos.

Fonte: Construído pelo autor com base em Lourenço [18] e Oetiker et al. [19].

No espaço "opções" do comando (4.1) podemos especificar algumas características do documento, como tamanho da fonte e tipo de papel. A Tabela 4.3 traz as principais opções que julgamos úteis, para as demais, consulte Oetiker et al. [19]. Estas opções devem ser digitadas dentro dos colchetes da macro (4.1) e, quando utilizadas mais de uma, deve-se separá-las por vírgulas.

Tabela 4.3: Principais opções para classes de documentos.

Opção	***Função***
10pt, 11pt, 12pt	Define o tamanho principal das letras do documento. Caso não especifique uma delas, é assumido 10pt.
a4paper, letterpaper, a5paper, b5paper, executivepaper, e legalpaper	Define o tamanho do papel. Por omissão, é utilizado o letterpaper.
twocolumn	Instrue o LaTeX a escrever o documento em duas colunas.

Fonte: Oetiker et al. [19].

Existem comandos que pertencem a um pacote específico de macros e, por esse motivo, só serão efetivamente reconhecidos no corpo do documento se o pacote for referenciado no preâmbulo. Portanto, devemos ficar atentos aos comandos que digitamos no corpo do documento, pois, caso utilizemos um comando que pertença a um pacote que não tenha sido referenciado no preâmbulo, ele ocasionará um erro na compilação, fazendo com que o PDF seja gerado inadequadamente ou até impedindo a sua criação.

Para referenciar um pacote e consequentemente liberar o uso de suas macros, escrevemos no preâmbulo o comando `\usepackage[opções]{nome do pacote}`, sendo colocado o nome do pacote entre chaves e entre colchetes as opções, se houver, que controlam suas propriedades.

Também existem pacotes que alteram as características gerais do documento, como é o caso do `geometry` que controla as margens do documento. Sua estrutura é

```
\usepackage[top=Xcm, bottom=Ycm, left=Zcm, right=Wcm]{geometry},
```
(4.2)

onde os valores especificados em `X`, `Y`, `Z` e `W` redefinem as margens do documento. Estes valores representam, respectivamente, as medidas das margens superior (`top`), inferior (`bottom`), esquerda (`left`) e direita (`right`).

O LaTeX possui suporte para várias línguas, sendo o inglês a padrão. Para especificar com qual língua estamos escrevendo, utilizamos o pacote `babel`. A escolha da língua é muito importante, pois permite ao LaTeX escrever corretamente os textos segundo às normas de cada uma. Um exemplo disto é a hifenização, isto é, a separação das sílabas quando uma palavra não cabe completamente em uma linha do texto. Em português, esta separação não deve dividir letras que pertencem a uma mesma sílaba, se não especificarmos que estamos trabalhando com a língua portuguesa, a hifenização se dará de forma inadequada.

A Escolha da língua é feita pelo comando

```
\usepackage[opções]{babel},
```
(4.3)

onde, especificamos entre os colchetes em qual língua estamos trabalhando. Para escolhermos o português brasileiro basta escrevermos `brazil` no espaço de opções, isto é, utilizamos o comando `\usepackage[brazil]{babel}`.

Além dos pacotes `geometry` e `babel`, existe o `inputenc` que agrega propriedades a todo o documento e é responsável por reconhecer a codificação dos acentos das letras em línguas que os utilizam, como é o caso do português. Aconselha-se utilizar neste pacote, embora existam outras, a opção `latin1`, isto é, a escrita completa para referenciar o pacote no preâmbulo é `\usepackage[latin1]{inputenc}`.

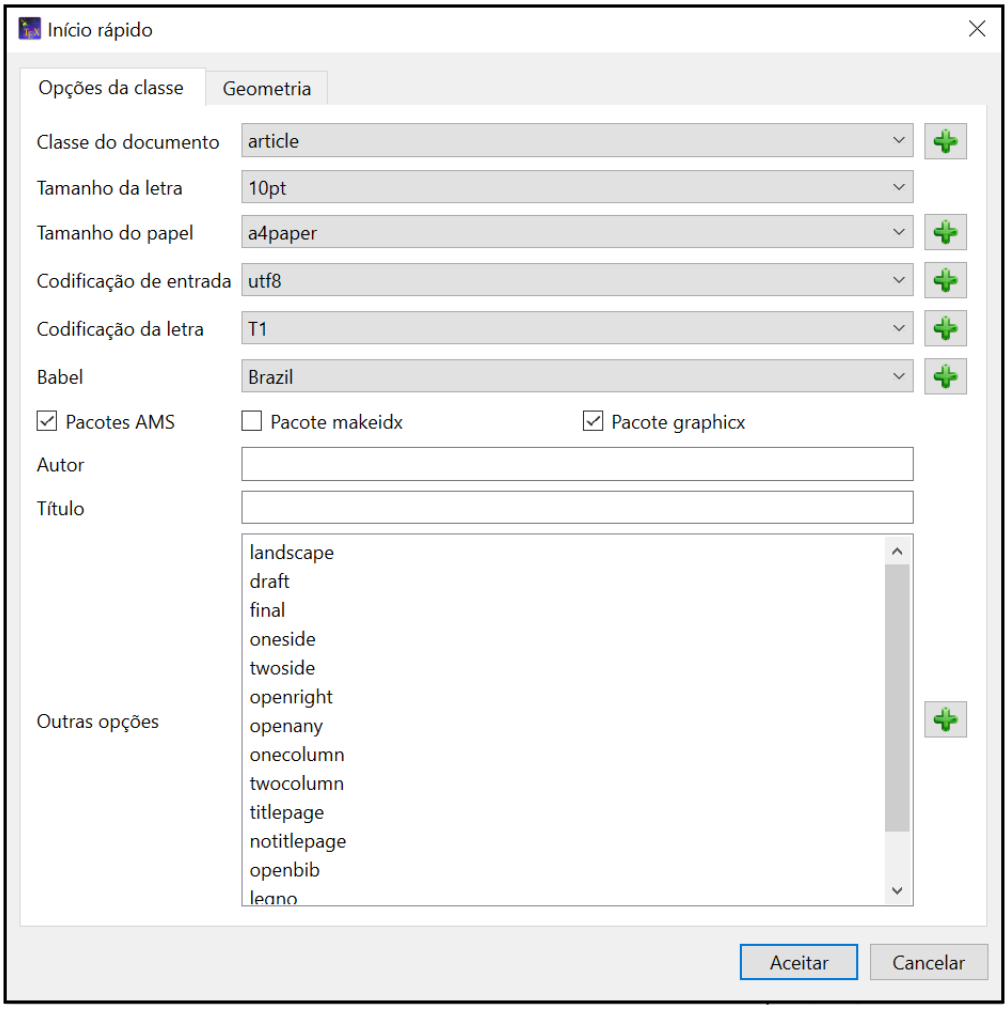
Assim, uma estrutura básica para a criação de um documento da classe `article` é o seguinte:

```
\documentclass[12pt,a4paper]{article}
\usepackage[brazil]{babel}
\usepackage[top=3cm, bottom=2cm, left=3cm, right=2cm]{geometry}
\usepackage[latin1]{inputenc}
\begin{document}
```

```
  O corpo do documento é construído aqui.
\end{document}
```

O TeXstudio possui a opção de início rápido de um arquivo TEX que permite sua criação de forma simples e rápida. Para utilizar esta opção, procure na parte superior do programa a opção "Assistentes", chique nela e depois em "Início rápido...", aparecerá a tela representada pela Figura 4.33. Nela é possível escolher, entre outras coisas, a classe do documento, o tamanho da letra, o tipo de papel e a língua.

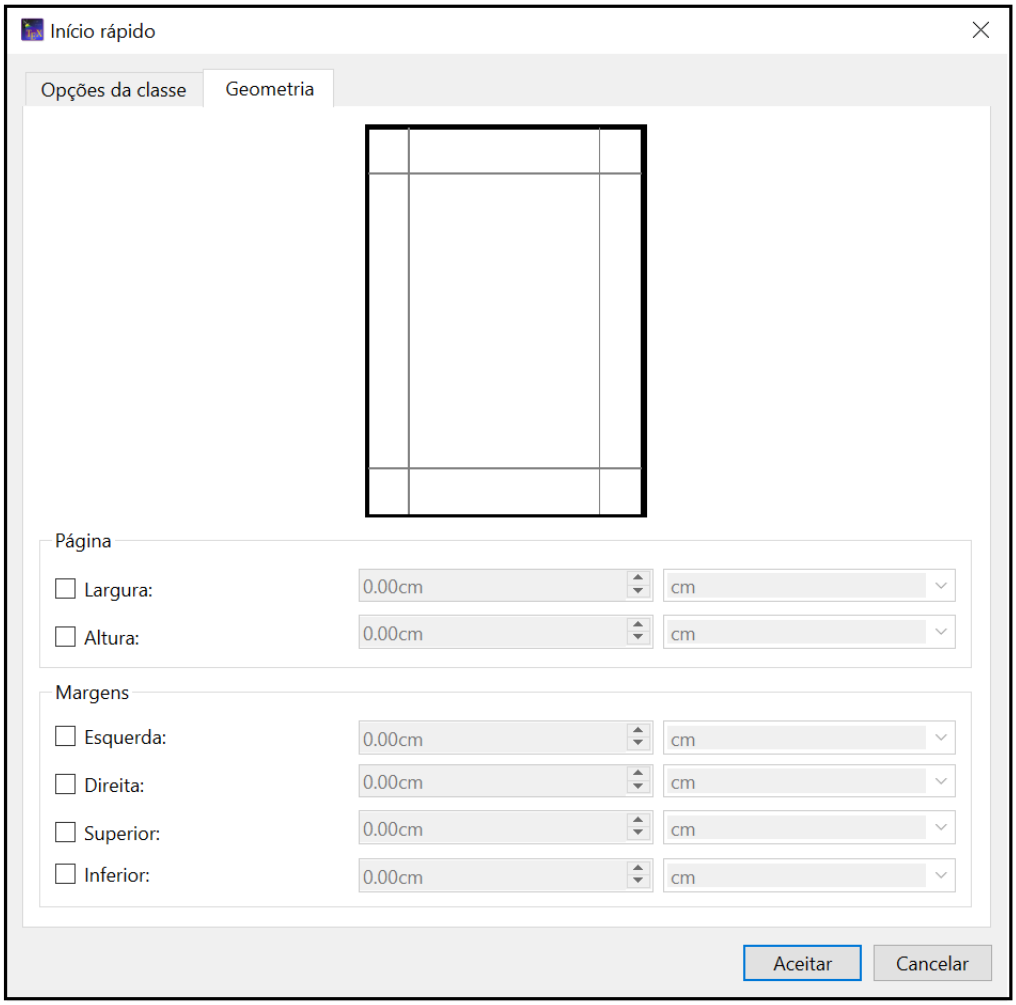
Figura 4.33: Usando o assistente do TeXstudio para início rápido de documentos.



Fonte: Elaborada pelo autor.

Na mesma janela de início rápido, pode-se especificar as margens do documento, basta clicar em "Geometria", que será exibida a tela representada na Figura 4.34. Para controlar as margens, basta clicar nos quadradinhos da margem que se pretende especificar e digitar o tamanho da margem em centímetros.

Figura 4.34: Usando o assistente do TeXstudio para definição das margens.



Fonte: Elaborada pelo autor.

Quando estiver tudo pronto, basta clicar em "Aceitar", que será criada uma estrutura com preâmbulo e o ambiente destinado ao corpo do documento TEX sendo possível adicionar, sempre que quiser, outros pacotes no preâmbulo.

4.6 Configurações textuais básicas

Na seção anterior, aprendemos a configuração básica para um arquivo TEX que servem para todo o documento. Nesta seção apresentaremos os principais comandos para alterar características do texto em trechos específicos.

4.6.1 Espaços entre palavras, criação de parágrafos e espaços horizontais e verticais

Uma característica básica da escrita TEX é o espaçamento entre palavras. Por mais que se adicione mais de um espaço entre elas, quando compilamos, o LATEX imprime apenas um no PDF. Se, por algum motivo, quisermos colocar um espaço diferente, podemos fazer uso de alguns comandos que alteram este espaçamento em intensidades diferentes. Alguns comandos estão representados na Tabela 4.4. Observe como utilizá-los e os resultados esperados.

Tabela 4.4: Comandos para alterar o espaçamento entre palavras.

Comando	***Uso***	***Resultado esperado***
\,	Texto\,texto texto.	Texto texto texto.
\:	Texto\:texto texto.	Texto texto texto.
\;	Texto\;texto texto.	Texto texto texto.
\quad	Texto{\quad}texto texto.	Texto texto texto.
\qquad	Texto{\qquad}texto texto.	Texto texto texto.
\hspace{x cm}	Texto\hspace{1cm}texto texto.	Texto texto texto.

Fonte: Construído pelo autor com base em Lourenço [18].

Outra ação importante é a criação de parágrafos. Uma maneira simples de se criar um parágrafo é deixar uma linha em branco entre os textos que compõem o parágrafo anterior do próximo. Porém, se a intenção for adicionar uma quebra de linha sem criar um parágrafo, basta adicionar duas barras invertidas no local em que se pretende quebrar a linha. É importante ressaltar que a utilização de várias linhas em branco entre textos, não adiciona espaços adicionais entre os parágrafos, enquanto a utilização de vários pares de barras invertidas criam no PDF uma linha em branco para cada par de barras utilizado.

Se a intenção for adicionar um espaço vertical vazio entre textos, é recomendado o uso do comando `\vspace{xcm}`, onde especificamos entre chaves a medida do espaço em centímetros.

Tabela 4.5: Comandos para criar parágrafos, quebra de linhas e espaço vertical.

Comando	***Uso***	***Resultado esperado***
Uma linha em branco.	Texto texto texto texto texto texto texto. Texto texto texto texto.	Texto texto texto texto texto texto texto. Texto texto texto texto.
\\	Texto.\\Texto.\\\\Texto.	Texto. Texto. Texto.
\vspace{x cm}	Texto texto texto. *\vspace{1 cm}* Texto texto texto.	Texto texto texto. Texto texto texto.

Fonte: Elaborado pelo autor.

4.6.2 Alterando o tamanho da fonte no corpo do documento

Embora o tamanho da letra seja definido no preâmbulo do arquivo, é possível alterá-lo por meio de alguns comandos que o redefine para tamanhos preestabelecidos. A Tabela 4.6 traz esses comandos, como utilizá-los e os resultados da utilização de cada um.

Tabela 4.6: Comandos para alterar o tamanho da fonte no corpo do arquivo.

Comando	***Uso***	***Resultado esperado***
\tiny	{\tiny Texto texto.}	Texto texto.
\scriptsize	{\scriptsize Texto texto.}	Texto texto.
\footnotesize	{\footnotesize Texto texto.}	Texto texto.
\small	{\small Texto texto.}	Texto texto.
\normalsize	{\normalsize Texto texto.}	Texto texto.
\large	{\large Texto texto.}	Texto texto.
\Large	{\Large Texto texto.}	Texto texto.
\LARGE	{\LARGE Texto texto.}	Texto texto.
\huge	{\huge Texto texto.}	Texto texto.
\Huge	{\Huge Texto texto.}	Texto texto.

Fonte: Elaborado pelo autor com base em Lourenço [18].

Observe na segunda coluna da Tabela 4.6 que colocamos os comandos, juntamente com o texto, entre chaves, isso faz com que apenas o texto no interior das chaves seja alterado pelos comandos.

Se nenhuma das opções apresentadas na Tabela 4.6 atender suas necessidades, há a opção de especificar o tamanho da fonte por meio do comando

```
\fontsize{size}{skip}\selectfont,
```
(4.4)

sendo `size` o tamanho da fonte e `skip` o distanciamento entre linhas e, para obtermos um resultado agradável, recomenda-se que o valor de `skip` seja 1.2 vezes o valor escolhido para `size`.

Ao utilizarmos o comando (4.4), todo texto que estiver posicionado depois dele será alterado para o tamanho especificado, sendo que, para voltarmos a escrever com o texto em tamanho original, devemos digitar `\normalsize`.

4.6.3 Negrito, itálico e sublinhado

É comum, quando se está escrevendo, enfatizar uma palavra ou frase por meio de realces do texto com negrito, itálico ou sublinhado. No LaTeX, estes realces podem ser produzidos, respectivamente, por meio dos comandos `\textbf{texto}`, `\textit{texto}` e `\underline{texto}`. Todo texto que estiver escrito dentro das chaves sofrerá alteração pelo comando, observe os resultados na Tabela 4.7.

Tabela 4.7: Comandos para negrito, itálico e sublinhado.

Comando	***Uso***	***Resultado esperado***
\textbf{texto}	Texto \textbf{texto} texto.	Texto **texto** texto.
\textit{texto}	Texto \textit{texto} texto.	Texto *texto* texto.
\underline{texto}	Texto \underline{texto} texto.	Texto <u>texto</u> texto.

Fonte: Elaborado pelo autor.

4.7 Modo matemático e escrita matemática básica

4.7.1 Caracteres reservados

Para escrevermos equações matemáticas por meio do LaTeX, devemos primeiramente compreender que alguns caracteres são reservados para determinadas funções e, caso não sejam utilizados adequadamente, podem ocasionar erros ou resultados indesejados. Alguns desses caracteres estão representados na Tabela 4.8.

Tabela 4.8: Lista de caracteres reservados.

Caractere	***Função***	***Imprimir no PDF***
\	Barra invertida: utilizada principalmente para iniciar um comando (macro) e escrever caracteres especiais. Já o utilizamos várias vezes.	\textbackslash
#	Jogo da velha: utilizado em criações de novos comandos. Não o utilizaremos neste trabalho.	\#
$	Cifrão: utilizado para delimitar o ambiente matemático.	\$
^	Acento circunflexo: utilizado para escrever textos sobrescritos ou expoentes. É usado em ambientes matemáticos.	\^{}
_	*Underline*: utilizado para escrever textos subscrito. É usado em ambientes matemáticos.	_{}
%	Porcentagem: utilizado para adicionar comentários. Ao utilizá-lo, todo texto que estiver entre ele e o próximo parágrafo, não será impresso no PDF.	\%
&	"e" estilizado: utilizado para separar colunas em quadros, tabelas e matrizes.	\&
~	"Til": adiciona um espaço inquebrável entre palavras, isto é, ele força a escrita de duas palavras em uma mesma linha.	\~{}
{}	Par de chaves: é utilizado em comandos, onde o que estiver em seu interior, é tratado como parte da macro. Também é chamado de bloco de processamento e já o utilizamos várias vezes.	\{ \}

Fonte: Elaborado pelo autor com base em Lourenço [18].

Para utilizar os caracteres apresentados na Tabela 4.8 de forma a exercerem as funções descritas, devemos digitá-los tais como estão escritos na primeira coluna da tabela, o modo como estão escritos na terceira coluna faz com que o caractere seja impresso no PDF ao se compilar o arquivo.

4.7.2 Modo matemático

Para escrevermos expressões matemáticas, é necessário que entremos no modo matemático. Este modo é iniciado e finalizado com o símbolo do cifrão `$`, isto é, os elementos matemáticos devem ser escritos entre cifrões. Por exemplo, a escrita `$ax+b y+c=0$` gera a equação $ax + by + c = 0$. Observe neste exemplo que a fonte da equação sofreu alterações, ela foi impressa de forma ligeiramente semelhante ao itálico e os sinais de adição e de igualdade sofreram uma ligeira alteração. Isto é comum, visto que o modo matemático possui fonte própria que difere dos demais elementos textuais do documento. Além disso, no modo matemático, o distanciamento simples entre textos é desconsiderado e é colocado um pequeno espaço automático entre símbolos e textos, observe que, mesmo colocando um espaço entre as letras `b` e `y` no código da equação, ele não apareceu na equação compilada e que foi adicionado um pequeno espaço entre a o símbolo de adição `+` e a letra `c`, mesmo este espaço não existindo no código. Se quiser adicionar espaços entre textos no modo matemático, deve-se utilizar as opões apresentadas na Tabela 4.4.

Também é possível escrever as equações de forma destacada em relação ao texto comum do PDF. Para isso, basta escrever cifrões duplos para iniciar e finalizar o modo matemático. Por exemplo, a equação descrita no exemplo anterior podeira ter sido escrita na forma `$$ax+by+c=0$$`. Neste caso, a equação apareceria na linha seguinte ao centro da página, do seguinte modo:

$$ax + by + c = 0.$$

4.7.3 Símbolos matemáticos

Além dos caracteres reservados, apresentados na Tabela 4.8, existem os símbolos matemáticos. O LaTeX possui suporte para uma grande quantidade de símbolos matemáticos, os quais possuem codificações específicas para serem impressos no PDF e para tal, devem ser escritos no modo matemático. Observe na Tabela 4.9 os principais símbolos e como escrevê-los no corpo do documento para que sejam exibidos no PDF.

Tabela 4.9: Lista dos principais símbolos matemáticos.

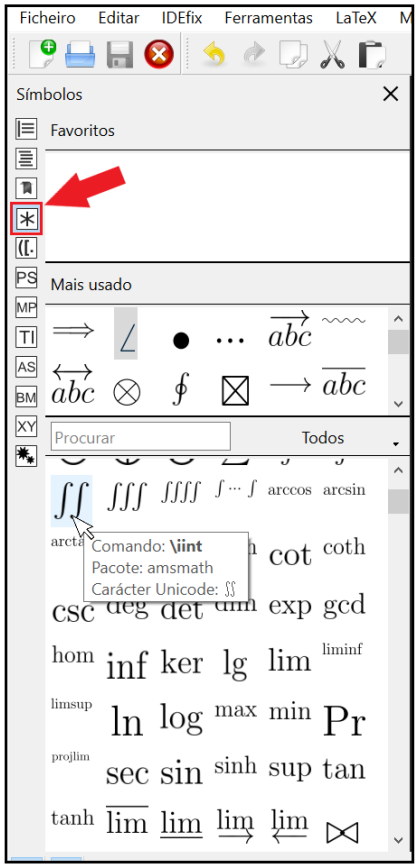
Nome do símbolo	***Uso***	***Resultado esperado***
Conjunto dos Naturais	$\mathbb{N}$	$\mathbb{N}$
Conjunto dos Inteiros	$\mathbb{Z}$	$\mathbb{Z}$
Conjunto dos Racionais	$\mathbb{Q}$	$\mathbb{Q}$
Conjunto dos Reais	$\mathbb{R}$	$\mathbb{R}$
Conjunto dos Complexos	$\mathbb{C}$	$\mathbb{C}$
Pertence	$\in$	$\in$
Não pertence	$\notin$	$\notin$
Está contido	$\subset$	$\subset$
Não está contido	$\not\subset$	$\not\subset$
Vazio	$\varnothing$	$\varnothing$
Menor do que, ou igual a	$\leqslant$	$\leqslant$
Maior do que, ou igual a	$\geqslant$	$\geqslant$
Menor do que	$\textless$	$<$
Maior do que	$\textgreater$	$>$
Diferente	$\neq$	$\neq$
Semelhante	$\sim$	$\sim$
Congruente	$\equiv$	$\equiv$
Aproximado	$\simeq$	$\simeq$
Paralelo	$\parallel$	$\parallel$
Ângulo	$\angle$	$\angle$
Vezes	$\times$	$\times$
Vezes (ponto)	$\cdotp$	$\cdotp$
Dividido	$\div$	$\div$

Fonte: Elaborado pelo autor.

É possível que algum dos símbolos apresentados na Tabela 4.9 pertença a um dos três pacotes: `amssymb`, `dsfont` ou `amsmath`. Portanto, para evitar problemas de compilação, é recomendado carregar estes pacotes no preâmbulo do arquivo TEX.

A quantidade de símbolos que o LATEX suporta é muito grande e é quase impossível que nos lembremos de todos os códigos que os representa. Felizmente, o TeXstudio possui uma lista de muitos símbolos que facilita a utilização. Para ter acesso a esta lista, basta clicar na opção "Símbolos", representada pelo ícone em forma de asterisco situado no lado esquerdo do programa. Veja a representação na Figuras 4.35.

Figura 4.35: Opções de símbolos matemáticos do TeXstudio.



Fonte: Elaborada pelo autor.

Observe que ao posicionar o mouse sobre o símbolo, além de indicar o comando, o TeXstudio também indica a que pacote o símbolo pertence.

Na mesma opção de símbolos, representada na Figura 4.35, encontram-se os comandos para imprimir no PDF todas as letras gregas comumente utilizadas em textos matemáticos.

4.7.4 Escrita de operações matemáticas

Em relação às operações de potenciação e radiciação, bem como à escrita de números em forma de fração e de elementos matemáticos com utilização de índices, utilizamos os comandos representados na Tabela 4.10. Alertamos que esses comandos devem ser escritos no modo matemático.

Tabela 4.10: Escrita de operações matemáticas.

Operação/Escrita	***Uso***	***Resultado esperado***
Potenciação	a^{b}	a^b
Raiz quadrada	$\sqrt{a}$	$\sqrt{a}$
Raiz n-ésima	$\sqrt[n]{a}$	$\sqrt[n]{a}$
Logaritmo	$\log_{b}(a)$	$\log_b(a)$
Logaritmo natural	$\ln(a)$	$\ln(a)$
Fração	$\frac{a}{b}$	$\frac{a}{b}$
Índices	a_{i}	a_i

Fonte: Elaborado pelo autor.

Esses comandos podem ser utilizados em conjunto para criar expressões mais complexas, como, por exemplo, o código

```
$$2^{\sqrt[4]{16}}+\frac{7+3}{\log_{2}(4)}$$,
```

que gera a expressão:

$$2^{\sqrt[4]{16}}+\frac{7+3}{\log_{2}(4)}.$$

Dependendo da necessidade, pode-se fazer uso dos comandos apresentados para criar outras expressões.

4.8 Ambientes

Na escrita TEX, existem o que chamamos de "ambientes", que são regiões destinadas à digitação e criação de elementos que possuem características específicas. Eles podem ser utilizados para criar tabelas, matrizes, inserir enumeração de itens e figuras, alinhamento de texto, inserção de equações enumeradas, etc. Existem muitos ambientes, sendo que, trataremos nesta seção, apenas dos ambientes `equation` e `figure`. Posteriormente, nas seções 5.2.1 e 5.3.2, exploraremos, respectivamente, os ambientes `tikzpicture` e `animateinline`, para informações sobre outros ambientes, consulte Oetiker et al. [19].

Para iniciar um ambiente, basta digitar `\begin{*nome-do-ambiente*}` e para finalizar, escrevemos `\end{*nome-do-ambiente*}`. Tudo que fizer parte do ambiente deve ser escrito entre estes dois comandos, sendo que, o nome do ambiente é escrito dentro das chaves no lugar de `*nome-do-ambiente*`. Em muitos casos é necessário que referenciemos o pacote ao qual o ambiente pertence no preâmbulo do arquivo, e, geralmente, o pacote e o ambiente, possuem o mesmo nome.

4.8.1 O ambiente `equation`

Como vimos na Seção 4.7, utilizamos o cifrão $ para delimitar o modo matemático, que é utilizado para escrever equações matemáticas, porém existe outra maneira de se escrever equações, que é utilizando o ambiente `equation`.

A vantagem de se utilizar este ambiente é que ele enumera as equações, o que é útil quando se pretende citá-las no texto. Por exemplo, o código

```
\documentclass[12pt,a4paper]{article}
\usepackage[brazil]{babel}
\usepackage[top=3cm, bottom=2cm, left=3cm, right=2cm]{geometry}
\usepackage[latin1]{inputenc}
\begin{document}
  \begin{equation}
    ax^{2}+bx+c=0.
  \end{equation}
\end{document}
```

gera um PDF com a equação

$$ax^2 + bx + c = 0. \tag{1}$$

Observe que a equação gerada recebeu a numeração (1), esta numeração é automática, o que significa que, ao digitarmos outra equação por meio do ambiente `equation`, ela receberá a numeração (2) e todas as outras que forem digitadas posteriormente continuarão a contagem. Mas caso queira colocar um nome específico para a equação basta digitar `\tag{nome da equação}` dentro do ambiente `equation` e especificar entre chaves o nome desejado para a equação. Observe que o código a seguir gera a mesma equação do exemplo anterior, mas com o nome (polinômio do 2º grau), perceba também que referenciamos no preâmbulo o pacote `amsmath` que é responsável por liberar, entre outras opções, a opção `tag`.

```
\documentclass[12pt,a4paper]{article}
\usepackage[brazil]{babel}
\usepackage[top=3cm, bottom=2cm, left=3cm, right=2cm]{geometry}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\begin{document}
  \begin{equation}
    ax^{2}+bx+c=0. \tag{polinômio do 2º grau}
  \end{equation}
```

```
\end{document}
```

Equação gerada no PDF:

$$ax^2 + bx + c = 0. \qquad \text{(polinômio do } 2^{\text{o}} \text{ grau)}$$

Outra observação importante é que não é necessário escrevermos entre cifrões as equações dentro do ambiente `equation`, na verdade se o fizermos, será apresentado um erro de compilação e o PDF não será gerado.

Também é possível utilizar o ambiente `equation` de forma que ele não enumere as equações, basta adicionar um asterisco no nome do ambiente, isto é, basta abrirmos o ambiente com `\begin{equation*}` e fecharmos com `\end{equation*}` que as equações não serão enumeradas.

4.8.2 O ambiente `figure`

Para inserirmos uma figura externa ao documento TEX utilizamos o comando

```
\includegraphics[opções]{nome_da_imagem},
```
(4.5)

o qual exige que se referencie no preâmbulo o pacote `graphicx`.

No espaço entre as chaves do comando (4.5) inserimos o nome do arquivo da imagem que pretendemos inserir no PDF, este arquivo deve estar salvo na mesma pasta em que se encontra o arquivo TEX e pode ser em formato PNG (*Portable Network Graphics*), JPEG (*Joint Photographic Experts Group*) ou até mesmo em formato PDF, dependendo do compilador que se utiliza, pois cada compilador possui uma lista própria de extensões de arquivos permitidas. Se o TeXstudio estiver configurado de acordo com as instruções apresentadas na Seção 4.4, será possível a inserção de aquivos nos três formatos supracitados. No espaço de opção do comando (4.5), é possível controlar características da imagem como escala, altura, largura ou rotacioná-la segundo um ângulo específico, a Tabela 4.11 traz um resumo destas opções e como utilizá-las.

Tabela 4.11: Resumo das opções para a macro (4.5).

\includegraphics[opções]{nome da imagem}		
Opção	**Função**	**Exemplos**
scale="valor"	Redimensiona de forma proporcional a imagem conforme o número especificado em *"valor"*, sendo 1 o tamanho original.	*scale=0.5* *scale=1.3*
height="valor"	Altera a altura da imagem conforme o número especificado em *"valor"* que pode ser um percentual da altura original ou um valor específico em centímetros. Esta opção pode deformar a imagem, gerando resultados estranhos.	*height=0.3* *height=4cm*
width="valor"	Controla a largura da imagem conforme o número especificado em *"valor"* que pode ser um percentual da largura original ou um valor específico em centímetros. Esta opção pode deformar a imagem, gerando resultados estranhos.	*width=0.7* *width=5cm*
rotate="ângulo"	Rotaciona a imagem no sentido anti-horário conforme o número especificado em *"ângulo"* que deve ser em graus	*rotate=30* *rotate=90*

Fonte: Elaborado pelo autor.

A macro (4.5) pode ser utilizada dentro do ambiente `figure` que, com o auxílio do pacote `caption`, possibilita adicionar legendas à imagem por meio dos comandos `\caption{legenda}` e `\caption*{legenda}`, de forma que, o primeiro escreve automaticamente a palavra "Figura" e enumera a imagem, também de forma automática, enquanto que o segundo comando escreve a legenda no PDF exatamente como a escrevemos. Observe o código a seguir, ele insere a Figura 4.36 tal qual está representada, perceba que referenciamos o pacote `caption` no preâmbulo para que as legendas sejam adicionadas corretamente, note também que adicionamos o comando `\centering`, esta macro é responsável por centralizar a imagem.

```
\documentclass[12pt,a4paper]{article}
\usepackage[brazil]{babel}
\usepackage[top=3cm, bottom=2cm, left=3cm, right=2cm]{geometry}
\usepackage[latin1]{inputenc}
\usepackage{graphicx}
```

```
\usepackage{caption}
\begin{document}
  \begin{figure}[h]
    \caption{Exemplo de inserção de figuras.}
    \centering
    \includegraphics[scale=0.1]{Simbolo_principal_da_UFCA}
    \caption*{Fonte: https://www.ufca.edu.br}
  \end{figure}
\end{document}
```

Figura 4.36: Exemplo de inserção de figuras.

Fonte: https://www.ufca.edu.br

É possível que ao inserir uma figura por meio do ambiente `figure`, ela seja posicionada em lugares indesejados na página do PDF, isso ocorre por que o LaTeX tenta posicionar a imagem de forma a equilibrar texto e figura, na tentativa de aproveitar ao máximo os espaços da página. Caso seja necessário alterar o posicionamento da imagem, podemos nos valer de algumas opções para o ambiente `figure`, perceba que no código responsável pela inserção da Figura 4.36, está escrito `[h]` na frente do comando `\begin{figure}`. Este "h" força o LaTeX a posicionar a imagem exatamente onde a posicionamos no arquivo TeX, isto é, exatamente entre os textos que digitamos.

Existem outras opções de posicionamentos, o uso de "t" no lugar do "h" instrui o LaTeX a posicionar a figura no topo da página, enquanto que o uso de "b", posiciona a imagem no final da página. Ainda é possível utilizar o sinal de exclamação "!" seguido de uma sequência das letras apresentadas para que o LaTeX posicione a imagem da melhor forma possível de acordo com a sequência estabelecida, isto é, se escrevermos `[!htb]` na frente de `\begin{figure}`, será estabelecida uma hierarquia de preferência para o posicionamento da imagem seguindo da esquerda para a direita.

O ambiente `figure` também pode ser utilizado quando criamos as imagens com o próprio LaTeX. A criação dessas figuras exige conhecimentos sobre os pacotes `TikZ` ou `tkz-euclide` e, por isso, passaremos a apresentar no capítulo a seguir como manosear as principais macros e opções destes dois pacotes que, utilizadas em conjunto com comandos do pacote `animate`, possibilitam a criação de animações

que também são compostas por imagens e, portanto, podem ser inseridas dentro de um ambiente `figure`.

Capítulo 5

Desenhando e Animando no LaTeX

5.1 Um pouco dos pacotes `TikZ`, `tkz-euclide` e `animate`

Neste capítulo, trataremos do manuseio de comandos que fazem parte dos pacotes `TikZ`, `tkz-euclide` e `animate`. O foco é explorar macros que permitam desenhar e animar elementos e figuras da Geometria Euclidiana Plana.

O `TikZ` foi criado pelo alemão Till Tantau em 2005 e é definido pelo próprio criador em [28] como sendo um conjunto de macros que definem uma série de comandos TeX que desenham gráficos. O autor alerta que o `TikZ` não é um programa de desenho, mas que permite programar desenhos onde se tem vantagens como criação rápida de gráficos simples e posicionamento altamente preciso dos elementos do desenho, no entanto, como desvantagem apresenta curva de aprendizagem muito íngreme e pequenas mudanças no código requer muito tempo de recompilação.

De acordo com Matthes [29], pensando em tornar a programação de desenhos geométricos mais simples, o francês Alain Matthes baseou-se no `TikZ` e construiu o `tkz-euclide`, tendo como público alvo os professores de Matemática. Isso facilitou bastante a criação de figuras geométricas, pois o `TikZ` não é focado nesse tipo de desenho. Por esta razão, utilizaremos principalmente o `tkz-euclide` para elaborar as figuras. Basicamente, usaremos as macros do `tkz-euclide` e as opções do `TikZ` para alterar as características dos elementos das figuras como, por exemplo, a espessura dos segmentos de reta.

Tanto o `TikZ` quanto o `tkz-euclide` estão em constante transformação e há várias versões produzidas, sendo que as mais recentes são mais aprimoradas e buscam correções de *bugs* que surgiram nas versões anteriores. Em relação a estas versões, destacamos que as macros apresentadas neste trabalho, são baseadas na versão 3.06c do `tkz-euclide`, produzida no ano de 2020, cujo manual está disponível em [29]. Essa versão, exige uma versão igual ou superior a 3.0 do `TikZ`. Utilizamos neste trabalho, o `TikZ 3.1.9a`, do ano de 2021, cujo manual pode ser encontrado em [28].

Versões mais recentes e seus manuais estão disponíveis em https://www.ctan.org.

Os pacotes `TikZ` e `tkz-euclide` serão utilizados para desenhar as figuras geométricas. Para a animação, utilizaremos as macros do pacote `animate`, criado por Alexander Grahn, o qual é definido em [20] pelo próprio criador como sendo um pacote LaTeX para gerar animações PDF e SVG, baseadas em JavaScript a partir de conjuntos de gráficos vetoriais ou arquivos de imagem *raster* ou de gráficos embutidos.

A proposta é utilizar as macros do `animate` para produzir animações quadro a quadro, isto é, animações que fazem uso de uma sequência de imagens que, ao serem sobrepostas em uma dada velocidade, simulam a movimentação de elementos dentro das imagens. Para isso, é necessário que saibamos desenhar utilizando as macros dos pacotes supracitados.

5.2 Desenhando no LaTeX

Nesta seção, explicaremos como desenhar figuras euclidianas com o LaTeX, através das principais macros do pacote `tkz-euclide`.

5.2.1 Preparando o ambiente para desenho e estrutura das macros

Para desenhar figuras geométricas no LaTeX, pode-se utilizar o pacote `TikZ` ou o pacote `tkz-euclide`, sendo que o primeiro pode ser utilizado tanto para desenhos bidimensionais quanto para desenhos tridimensionais e projetos mais complexos, enquanto o segundo oferece suporte para desenhos no plano de forma mais simples e acessível. Em algumas situações utilizaremos comandos em `TikZ`, mas na grande maioria dos casos, utilizaremos comandos do pacote `tkz-euclide`, por ser mais simples de se utilizar. Os comandos `TikZ` se caracterizam, entre outras coisas, por apresentarem em seu final o ponto e vírgula e os comandos `tkz-euclide`, por iniciarem com `\tkz`.

Para fazer o proposto nesse capítulo, não é necessário carregar o pacote `TikZ`. Embora utilizemos o ambiente `tikzpicture` para desenhar as figuras, o carregamento do pacote `tkz-euclide` é suficiente, pois ele carrega automaticamente o `TikZ`. A estrutura do arquivo que utilizaremos é a seguinte:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
```

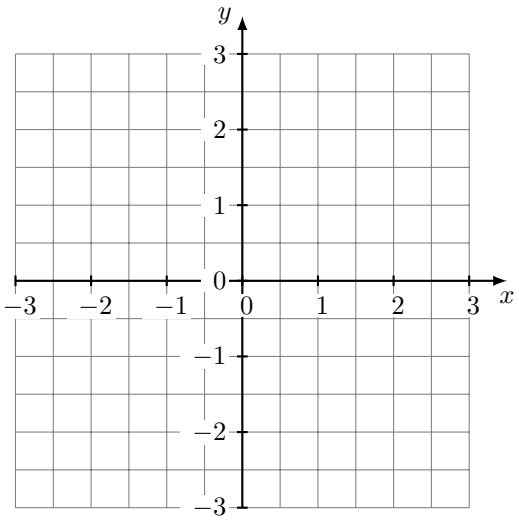
```
    % Os comandos para desenho serão escritos aqui.
  \end{tikzpicture}

\end{document}
```

Perceba que escolhemos a classe de documento `standalone`, pois nessa classe temos mais liberdade de espaço para desenhar, mas pode-se escolher naturalmente outra classe, sempre tomando cuidado para que o desenho fique dentro das margens da folha e que fique com tamanho adequado à visualização. Uma forma de controlar a área de desenho é construir uma "caixa delimitadora" que ficará em volta do desenho. Para mais informações veja a Subseção 5.3.1.

Ambos os pacotes utilizam tanto o sistema de coordenadas cartesianas quanto o sistema de coordenadas polares para as definições e desenho dos objetos geométricos. Optaremos pelo sistema cartesiano por ser mais utilizado no Ensino Médio. E para facilitar a construção dos desenhos geométricos, pode-se pedir ao LaTeX para desenhar os eixos coordenados Ox e Oy bem como uma malha quadriculada, utilizando para isso, respectivamente, os comandos `\tkzAxeXY` e `\tkzGrid`. Por padrão o comando `\tkzAxeXY` desenha os eixos Ox e Oy com valores mínimos iguais a zero e valores máximos iguais a dez centímetros, mas isso pode ser alterado pelo comando `\tkzInit[xmin=`$\mathtt{x}_0$`,xmax=`$\mathtt{x}_1$`,ymin=`$\mathtt{y}_0$`,ymax=`$\mathtt{y}_1$`]`, sendo $\mathtt{x}_0$ e $\mathtt{y}_0$ respectivamente os valores mínimos para os eixos Ox e Oy, e $\mathtt{x}_1$ e $\mathtt{y}_1$ os valores máximos. A malha quadriculada vem, por padrão, com quadrados de um centímetro de lado. Podemos alterá-la com o comando `\tkzGrid[step=x]`, sendo `x` o novo valor dos lados dos quadrados que compõem a malha. A Figura 5.1 traz os eixos com $\mathtt{x}_0 = \mathtt{y}_0 = -3$ e $\mathtt{x}_1 = \mathtt{y}_1 = 3$ e a malha quadriculada com `x=0.5cm`. Nela foram utilizados os seguintes comandos:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzInit[xmin=-3,xmax=3,ymin=-3,ymax=3]
    \tkzGrid[step=0.5cm]
    \tkzAxeXY
  \end{tikzpicture}
\end{document}
```

Figura 5.1: Eixos cartesianos e malha quadriculada.



Fonte: Elaborada pelo autor com base em Matthes [30].

A construção da malha quadriculada e dos eixos coordenados não é obrigatória para se desenhar no LaTeX, eles servem apenas de apoio para se orientar no plano. Nas próprias seções, não os desenharemos, a fim de simplificar os desenhos, evitando sobrecarregá-los. Mais informações sobre a malha e os eixos coordenados podem ser encontradas em Matthes [30].

De acordo com Matthes [29], os desenhos feitos com `tkz-euclide` são baseados em cinco ações: definir, criar, desenhar, marcar e rotular. Com exceção da segunda, essas serão as ações que mais utilizaremos e são produzidas com comandos que iniciam, respectivamente, com `\tkzDef`, `\tkzDraw`, `\tkzMark` e `\tkzLabel`. Os comandos relacionados às ações de criação não apresentam prefixo constante, mas geralmente possuem `\tkzGet` em sua estrutura. Esse tipo de comando é essencialmente dedicado à "criação" de pontos a partir de outros elementos predefinidos, como a interseção de duas retas ou a obtenção do ponto médio de um segmento de reta ou ainda a obtenção de pontos por transformações geométricas (translação, rotação, simetria ou reflexão).

5.2.2 Ponto

Definindo pontos

Os pontos são os elementos principais para se desenhar figuras geométricas. É a partir deles que será possível desenhar retas, semirretas, segmentos, polígonos, etc. Em `tkz-euclide`, só é possível desenhar ou nomear um ponto e, consequentemente, os objetos que se associam aos mesmos, se os definirmos previamente. Então, as primeiras macros a serem digitadas referentes aos desenhos devem ser as que de-

finem pontos, pois não poderemos desenhá-los se não estiverem definidos. Além disso, os comandos de definições devem ser digitados antes dos destinados a criar, desenhar, marcar ou rotular, e esta ordem não pode ser desrespeitada. Por exemplo, se pedirmos para desenhar um ponto A antes do comando que o define, será apresentado um erro com os dizeres *"Package pgf Erro: No shape named 'A' is known"*, que em tradução livre significa *"Erro no Pacote pgf: Nenhuma forma chamada 'A' é conhecida"*.

Os pontos são divididos em duas categorias: os "fixos" e os "criados". Vamos tratar inicialmente dos pontos fixos, os pontos criados serão abordados conforme a necessidade e de forma mais aprofundada na Subseção 5.2.10. Os pontos fixos são aqueles que definimos por meio de coordenadas, enquanto os pontos criados são obtidos às custas dos pontos fixos. Eles também possuem coordenadas, mas não precisamos conhecê-las para defini-los, desenhá-los ou nomeá-los.

Para se definir um único ponto fixo, pode-se utilizar o comando

$$\texttt{\textbackslash tkzDefPoint[opções](x,y)\{ponto}_1\texttt{\}}, \tag{5.1}$$

onde, no espaço de opções, entre colchetes, pode-se colocar `label="nome do ponto"` quando se pretende nomear o ponto. Esse argumento não é obrigatório e pode ser qualquer caractere aceito pelo TEX, até mesmo uma frase. Mais adiante veremos outra opção para nomear os pontos. As coordenadas cartesianas podem ser valores decimais e são colocadas em `x` e `y`, acomodadas entre parênteses e separados por vírgula (o ponto é reservado para separar inteiros de decimais). Por padrão, é assumido o centímetro como unidade de medida para as coordenadas. Entre chaves, colocamos a referência do ponto, a qual será usada para se fazer referência ao ponto posteriormente. Essa referência pode ser uma letra latina, um número, uma letra com texto subscrito ou sobrescrito (nesses casos a referência é escrita como no ambiente matemático, mas sem usar os cifrões) ou até mesmo uma frase, mas não são aceitos caracteres especiais, como $\otimes$, $\boxtimes$, $\divideontimes$ ou $\circledast$. É aconselhável que se use, sempre que possível, o mesmo símbolo para o nome e para a referência, pois isso evita confusão na hora de se referir ao ponto. Para este fim, usaremos letras latinas maiúsculas. Veja a seguir um exemplo de definição do ponto A:

```
\tkzDefPoint[label=A](-2,4){A}.
```

Para se definir um ou mais pontos fixos com uma única macro, pode-se fazer uso do comando

$$\texttt{\textbackslash tkzDefPoints[opções]\{x}_1\texttt{/y}_1\texttt{/n}_1\texttt{, x}_2\texttt{/y}_2\texttt{/n}_2\texttt{, ...,x}_i\texttt{/y}_i\texttt{/n}_i\texttt{\}}. \tag{5.2}$$

Nessa macro, podemos colocar na parte de opções, entre colchetes, os argumentos

```
[xshift="valor",yshift="valor"].
```
(5.3)

Essa opção move todos os pontos definidos com essa macro conforme os valores especificados em `"valor"`, sendo `xshift="valor"` referente à abcissa e `yxshift="valor"`, à ordenada (essa opção não é obrigatória e também pode ser utilizada na macro (5.1)). Os valores das abcissas e das ordenadas são colocados, respectivamente, em $\texttt{x}_i$ e $\texttt{y}_i$ e as referências em $\texttt{n}_i$. A seguir, definimos os pontos A, B, C e D utilizando essa macro, onde deslocamos a abcissa em 2 cm e a ordenada em -1 cm:

```
\tkzDefPoints[xshift=2cm,yshift=-1cm]{0/0/A,1/2/B,-1/-2/C,-1/2/D}.
```

Esse deslocamento das coordenadas utilizando `shift` é opcional, ele é útil quando se pretende, por algum motivo, deslocar os valores das coordenadas de todos os pontos evitando que se faça isso ponto a ponto.

Desenhando pontos

A definição de um ponto não o desenha. Para esse fim devemos utilizar a macro

$$\texttt{\tkzDrawPoint[opções](ponto}_1\texttt{)}, \tag{5.4}$$

se desejarmos desenhar um único ponto, ou

$$\texttt{\tkzDrawPoints[opções](ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\texttt{, ...,ponto}_i\texttt{)}, \tag{5.5}$$

para desenhar mais de um ponto.

Os comandos colocados no espaço de opções, em ambos os casos, modificam o ponto em relação ao formato, tamanho e cor. Para modificar o formato do ponto utilizamos `shape="formato"`, onde podemos usar `cross` que faz o ponto com a forma de uma cruz, `cross out` que faz o ponto no formato de "x" e `circle` para o ponto em forma de círculo. Por padrão, o ponto possui o formato de um círculo, isto é, se a opção do formato do ponto for omitida, será assumido o argumento `circle`.

Para modificar o tamanho do ponto, utilizamos `size="valor"`, onde se deve especificar a medida do diâmetro desejado do ponto. A unidade de medida padrão para esse fim da versão 3.06c do `tkz-euclide` é o "ponto", mas podemos utilizar as unidades métricas com a correspondência aproximada de 1 pt para cada 0.03527 cm. Se o tamanho do ponto não for especificado, como padrão, será assumido o diâmetro de 3 pt o que é equivalente a aproximadamente 0.10581 cm.

Por padrão, o ponto possui fronteira e interior na cor preta, mas podemos mudar essas cores usando, respectivamente, `draw="cor"` e `fill="cor"`.

Devemos especificar os pontos que desejamos desenhar, entre parenteses e separados por vírgulas, por meio das referências criadas ao definir os pontos.

A Figura 5.2 traz alguns exemplos de como desenhar pontos com possíveis combinações de opções para forma, tamanho e cores. Tais pontos foram obtidos por meio do código a seguir.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoint[label=A](0,0){A}
    \tkzDefPoint(1,0){B}
    \tkzDefPoints{2/0/C, 3/0/D, 4/0/E}
    \tkzDefPoints[xshift=4cm,yshift=-3cm]{1/3/F, 2/3/G}
    \tkzDrawPoint[size=0.2cm,fill=yellow,draw=black](A)
    \tkzDrawPoints(B,C)
    \tkzDrawPoints[shape=cross,draw=red,size=0.5cm](D)
    \tkzDrawPoints[shape=cross out,size=0.15cm,draw=blue](E,F)
    \tkzDrawPoints[size=0.15cm,draw=blue,fill=red](G)
  \end{tikzpicture}
\end{document}
```

Figura 5.2: Pontos com diferentes características.

Fonte: Elaborada pelo autor com base em Matthes [29].

Nomeando pontos

Observe que, na Figura 5.2, apenas o ponto A foi nomeado, pois foi o único que recebeu a opção `label` ao ser definido. Essa forma de nomear um ponto não é muito útil, pois não há como editar, por exemplo, sua posição ou alterar o tamanho do nome. Assim, para nomear um único ponto é aconselhável utilizar a macro

$$\texttt{\textbackslash tkzLabelPoint[opções](ponto}_1\texttt{)\{nome\}}. \tag{5.6}$$

Por meio da macro (5.6), podemos controlar características do nome como tamanho, cor e posição de ancoragem (onde o nome do ponto ficará posicionado considerando seu entorno), colocando opções específicas no local destinado.

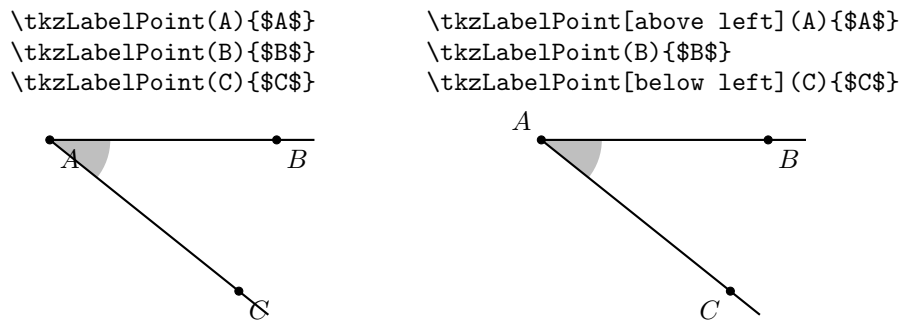
Para controlar o tamanho, podemos utilizar `scale="valor"`. Essa opção aumenta ou diminui o tamanho da etiqueta do ponto conforme o valor especificado em percentual, sendo obtido o tamanho normal quando se assume o valor 1 como argumento ou quando se omite essa opção.

Por padrão, a cor do nome é preta. Se quisermos alterá-la, basta escrever a cor desejada no local de opções da macro, que a etiqueta do ponto assumirá a cor especificada.

O controle da posição de ancoragem do nome do ponto é feito por meio de `below`, `above`, `right`, `left`, `below right`, `below left`, `above right` ou `above left`. Estas opções significam que o nome do ponto será ancorado, respectivamente, em regiões situadas ao sul, norte, leste, oeste, sudeste, sudoeste, nordeste e noroeste do ponto, sendo ainda possível especificar a distância entre o nome e o ponto, colocando o valor dessa distância após a definição da posição de ancoragem. Por exemplo, `below=3mm`, faz o nome ficar ancorado ao sul do ponto a uma distância de 3 milímetros.

Para um resultado ainda mais preciso para o posicionamento, pode-se utilizar as opções `xshift="valor"` e `yshift="valor"` que permitem deslocar o nome de sua posição padrão (`below right`), conforme valores especificados em `"valor"`. Estas opções são muito úteis, pois quando desenhamos uma figura muitas vezes o nome do ponto fica posicionado sobre um elemento desenhado como, por exemplo, um segmento ou no interior de um ângulo, essas opções possibilitam mover a etiqueta para um espaço adequado. A Figura 5.3 representa uma dessas situações, onde do lado esquerdo as macros nomeiam os pontos A e C em regiões indesejadas e, do lado direito, o problema é resolvido com as opções apresentadas.

Figura 5.3: Posicionamento adequado dos nomes de pontos.



Fonte: Elaborada pelo autor.

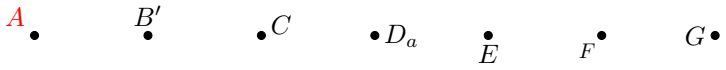
Após as opções na macro (5.6), colocamos entre parênteses a referência do ponto previamente definido que se pretende nomear. Depois da referência, colocamos entre chaves o nome desejado do ponto. Note que, na Figura 5.3, o nome do ponto foi

colocado entre cifrões ao utilizar essa macro. Isso é necessário para que o nome fique na forma matemática.

Observe, na Figura 5.4 e no código a seguir, exemplos de nomeação de pontos utilizando a macro (5.6).

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A,1.5/0/B',3/0/C,4.5/0/D_a,6/0/E,7.5/0/F,9/0/G}
    \tkzDrawPoints(A,B',C,D_a,E,F,G)
    \tkzLabelPoint[red,above left](A){$A$}
    \tkzLabelPoint[above](B'){$B'$}
    \tkzLabelPoint[above right,yshift=-0.1cm](C){$C$}
    \tkzLabelPoint[right](D_a){$D_a$}
    \tkzLabelPoint[below](E){$E$}
    \tkzLabelPoint[xshift=-0.4cm,scale=0.8](F){$F$}
    \tkzLabelPoint[left](G){$G$}
  \end{tikzpicture}
\end{document}
```

Figura 5.4: Algumas possibilidades para nomeação de pontos.



Fonte: Elaborada pelo autor.

É possível nomear vários pontos de uma só vez utilizando a macro

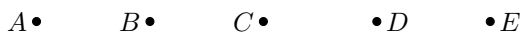
$$\texttt{\textbackslash tkzLabelPoints[opções](ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\texttt{, ...,ponto}_i\texttt{).} \qquad (5.7)$$

A vantagem de se utilizar esta macro é óbvia, economizamos linhas ao digitar um código, mas como desvantagem todos os pontos apresentarão as mesmas características especificadas no espaço de opções. Por isso, ele é indicado para nomear pontos que compartilham as mesmas opções utilizadas.

Observe que a macro (5.7) não apresenta lugar para se digitar o nome do ponto como é no comando (5.6), isto porque, ela assumirá como nome a referência dada quando o ponto foi definido, isto é, o ponto será nomeado conforme a referência. O código a seguir traz exemplos de utilização dessa macro e a Figura 5.5 representa o resultado esperado.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A,1.5/0/B,3/0/C,4.5/0/D,6/0/E}
    \tkzDrawPoints(A,B,C,D,E)
    \tkzLabelPoints[left](A,B,C)
    \tkzLabelPoints[right](D,E)
  \end{tikzpicture}
\end{document}
```

Figura 5.5: Nomeando mais de um ponto com um comando.



Fonte: Elaborada pelo autor.

É válido ressaltar que o nome do ponto será apresentado automaticamente no formato matemático quando se utiliza a macro (5.7). A Tabela 5.1 traz um resumo das opções para nomeação de pontos.

Tabela 5.1: Resumo das opções para as macros (5.6) e (5.7).

\tkzLabelPoint[opções](ponto$_1$){nome} ***\tkzLabelPoints[opções](ponto$_1$,ponto$_2$,ponto$_3$,...,ponto$_i$)***		
Opção	**Função**	**Exemplo**
scale="valor"	Controla o tamanho do nome.	*scale=0.4*
color	Altera a cor de preta para a especificada.	*red*
below	Posiciona o nome ao sul do ponto.	*below*
above	Posiciona o nome ao norte do ponto.	*above*
right	Posiciona o nome ao leste do ponto.	*right=-2mm*
left	Posiciona o nome ao oeste do ponto.	*left=1mm*
below right	Posiciona o nome ao sudeste do ponto.	*below right*
below left	Posiciona o nome ao sudoeste do ponto.	*below left*
above right	Posiciona o nome ao nordeste do ponto.	*above right*
above left	Posiciona o nome ao noroeste do ponto.	*above left*
xshift="valor"	Posiciona o nome conforme valor especificado.	*xshift=2mm*
yshift="valor"	Posiciona o nome conforme valor especificado.	*yshift=1mm*

Fonte: Elaborado pelo autor com base em Matthes [29].

5.2.3 Reta, Semirreta e Segmento

Desenhando retas, semirretas e segmentos

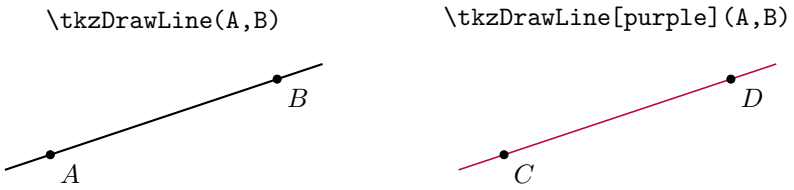
Para se desenhar uma reta, é necessário que dois pontos sejam definidos, a reta incidirá nos dois pontos. A macro destinada ao desenho da reta é

$$\texttt{\textbackslash tkzDrawLine[opções](ponto}_1\texttt{,ponto}_2\texttt{).} \tag{5.8}$$

Colocamos estre parênteses as referências dos dois pontos por onde a reta incidirá e no espaço de opções podemos colocar argumentos para controlar as características da reta como cor, espessura e estilo da linha.

Por padrão, a cor da linha que representa a reta é preta, mas podemos alterá-la colocando simplesmente o nome da cor no espaço de opções. A Figura 5.6 traz, à esquerda, o resultado padrão obtido com a macro sem nenhuma opção e, à direita, a macro com a opção que altera a cor de `black` (preto) para `purple` (roxo). Os pontos e seus nomes foram desenhados por meio dos comandos (5.5) e (5.7), respectivamente.

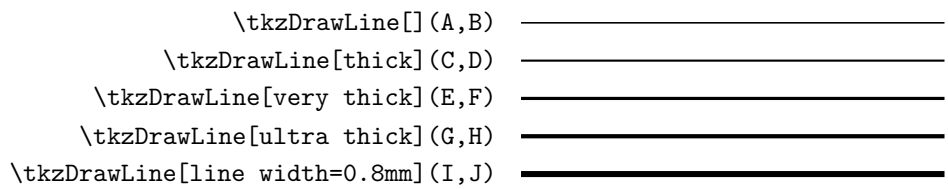
Figura 5.6: Exemplo de mudança na cor de uma reta.



Fonte: Elaborada pelo autor.

Pode-se alterar a espessura da linha por meio das opções `thick`, `very thick` ou `ultra thick` que alteram a espessura para valores predefinidos. Porem, se desejar uma espessura precisa, a opção `line width="valor"` é mais indicada onde especifica-se o valor desejado em centímetros (cm) ou milímetros (mm). Veja, na Figura 5.7, os resultados da utilização dessas opções.

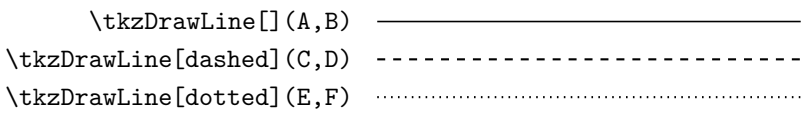
Figura 5.7: Exemplo controle da espessura de uma reta.



Fonte: Elaborada pelo autor.

O estilo padrão da reta é uma linha contínua. Esse estilo pode ser alterado para tracejado ou pontilhado, utilizando-se as opções `dashed` ou `dotted`, respectivamente. A Figura 5.8 ilustra os três resultados possíveis de estilo de linha.

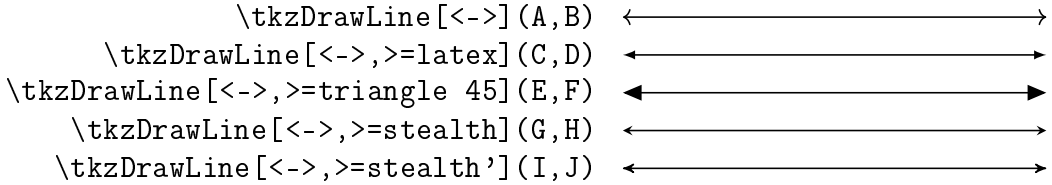
Figura 5.8: Exemplo estilo da linha de uma reta.



Fonte: Elaborada pelo autor.

Se desejarmos, é possível colocarmos uma seta na ponta da reta. Para isso, basta digitar `<->` no espaço de opções. Veja alguns resultados na Figura 5.9.

Figura 5.9: Exemplo de estilos das pontas de uma reta.



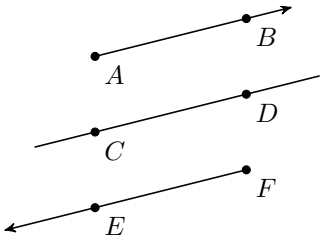
Fonte: Elaborada pelo autor.

Observe que, na Figura 5.6, as retas desenhadas excedem os segmentos AB e CD nas duas pontas. Por padrão, esse excesso é equivalente a 20% do tamanho do segmento, mas pode ser controlado com a opção `add="valor" and "valor"`. O primeiro valor, controla o excesso do primeiro ponto, e o segundo valor controla o do segundo ponto. Esses valores podem ser em percentuais (tomando como base o tamanho do segmento) ou em centímetros. Podemos nos valer desta opção para desenharmos semirretas, sendo que, basta considerar um dos pontos como origem da semirreta e adotar zero no valor que controla o excesso neste ponto. A Figura 5.10 mostra o resultado do código completo a seguir, exemplificando a obtenção de semirretas por meio da opção `add`.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{-3/1.5/A,-1/2/B,-3/0.5/C,-1/1/D,-3/-0.5/E,-1/0/F}
    \tkzDrawLine[add=0 and 0.3,->,>=stealth'](A,B)
    \tkzDrawLine[add=0.4 and 1cm](C,D)
```

```
    \tkzDrawLine[add=0.6 and 0,<-,>=stealth'](E,F)
    \tkzDrawPoints(A,B,C,D,E,F)
    \tkzLabelPoints(A,B,C,D,E,F)
  \end{tikzpicture}
\end{document}
```

Figura 5.10: Utilizando `add` para desenhar semirretas.



Fonte: Elaborada pelo autor com base em Matthes [29].

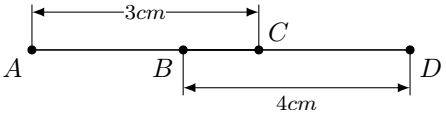
Podemos ainda utilizar a opção `add` na macro (5.8) para desenhar segmentos. Para isso, basta colocar zero em ambos os excessos. Porém, esta é apenas uma alternativa, pois existe uma macro específica para desenhar segmentos, a saber,

$$\texttt{\textbackslash tkzDrawSegment[opções](ponto}_1\texttt{,ponto}_2\texttt{)}. \tag{5.9}$$

Todas as opções apresentadas referentes à macro (5.8) podem ser utilizadas na macro (5.9). No entanto, uma opção muito útil para segmentos é a `dim` que serve para adicionar uma linha com setas nas pontas e indicar o comprimento do segmento. A Figura 5.11 representa a utilização desta opção.

Figura 5.11: Exemplo de utilização da opção *dim*.

```
\tkzDrawSegment[dim={$3cm$,0.5cm,}](A,C)
\tkzDrawSegment[dim={$4cm$,-0.5cm,below=1mm}](B,D)
```



Fonte: Elaborada pelo autor.

Observe, na Figura 5.11, que no segundo comando, usado para desenhar o segmento BD, foi utilizada a opção `dim={$4cm$,-0.5cm,below=1mm}`. O primeiro argumento deve ser o que pretendemos que apareça próximo à linha, geralmente é o comprimento do segmento, neste exemplo, `$4cm$`. O segundo argumento controla a

distância que a linha com setas aparecerá em relação ao segmento: valores positivos a colocarão acima ou à direita e valores negativos a colocarão abaixo ou à esquerda (isso também depende de como estão escritas as referências dos pontos: escrever (D, B) em vez de (B, D) alterará o sentido), `-0.5cm` indica que a linha com setas ficará abaixo do segmento BD a uma distância de 0.5 cm. O terceiro argumento controla a posição/ancoragem e a distância que o texto escrito no primeiro argumento aparecerá. Por exemplo, `below=1mm` indica que o texto ficará abaixo e a uma distância de 1 mm da linha com setas. Caso esse argumento seja omitido, o texto ficará posicionado no centro da linha como foi o caso do segmento AC. É importante ressaltar que, mesmo omitindo o terceiro argumento, é obrigatório colocar a vírgula antes de fechar a chave.

As macros (5.8) e (5.9) são utilizadas para desenhar uma única reta e um único segmento. Com o objetivo de diminuir o número de linhas do código, é possível desenhar várias retas e vários segmentos fazendo uso, respectivamente, das macros

$$\texttt{\textbackslash tkzDrawLines[opções](ponto}_1\texttt{,ponto}_2\ \texttt{ponto}_3\texttt{,ponto}_4\ \texttt{...ponto}_i\texttt{,ponto}_j\texttt{)} \tag{5.10}$$

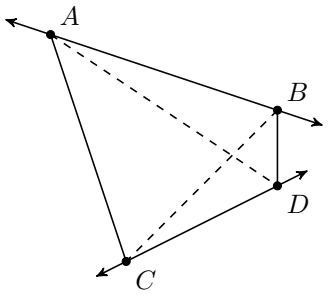
e

$$\texttt{\textbackslash tkzDrawSegments[opções](ponto}_1\texttt{,ponto}_2\ \texttt{ponto}_3\texttt{,ponto}_4\ \texttt{...ponto}_i\texttt{,ponto}_j\texttt{),} \tag{5.11}$$

onde os pares de pontos (referências) devem ser separados com um espaço simples e cada par representará um segmento. A desvantagem da utilização dessas macros é que todas as retas ou todos os segmentos ficarão com a mesma aparência, de acordo com as opções digitadas. A Figura 5.12 traz um exemplo, que foi obtido pelo seguinte código:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{-1/0/A, 2/-1/B, 0/-3/C, 2/-2/D}
    \tkzDrawLines[<->,>=stealth'](A,B C,D)
    \tkzDrawSegments(A,C B,D)
    \tkzDrawSegments[dashed](A,D B,C)
    \tkzDrawPoints(A,B,C,D)
    \tkzLabelPoints[above right](A,B)
    \tkzLabelPoints(C,D)
  \end{tikzpicture}
\end{document}
```

Figura 5.12: Desenhando mais de uma reta ou segmento com o mesmo comando.



Fonte: Elaborada pelo autor.

Nomeando retas e segmentos

Além de desenhar, podemos nomear uma reta ou um segmento, respectivamente, por meio dos comandos

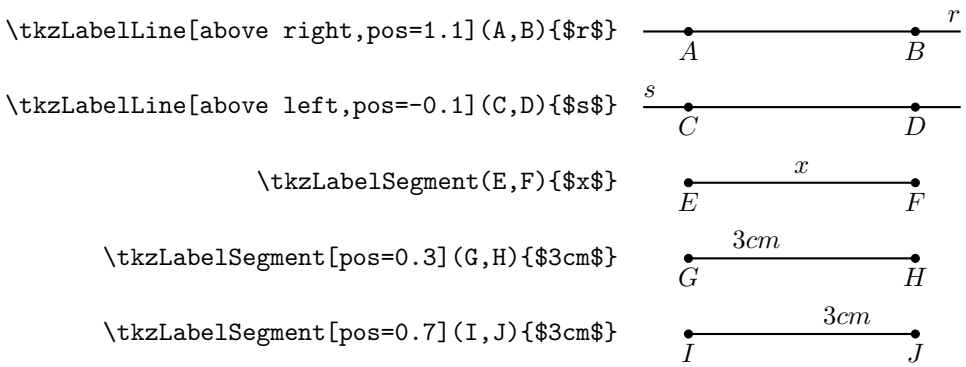
$$\texttt{\textbackslash tkzLebelLine[opções](ponto}_1\texttt{,ponto}_2\texttt{)\{nome\}} \tag{5.12}$$

ou

$$\texttt{\textbackslash tkzLebelSegment[opções](ponto}_1\texttt{,ponto}_2\texttt{)\{nome\}}. \tag{5.13}$$

Nestas macros, colocamos entre parênteses as referências dos pontos por onde a reta ou o segmento incidirá e entre chaves colocamos o nome que desejamos. Em relação às opções, são válidas todas as apresentadas na macro (5.6) e, além delas, podemos utilizar a opção `pos="valor"`, que controla a posição do nome ao longo da reta ou do segmento. Para esse controle, devemos indicar, em `"valor"`, o percentual em decimal do segmento definido pelos pontos de referência. Veja alguns resultados possíveis na Figura 5.13.

Figura 5.13: Nomeando retas e segmentos.



Fonte: Elaborada pelo autor.

Quando o segmento não estiver na horizontal, pode ser útil utilizar a opção `sloped` para alinhar o nome do segmento com a sua direção. Esse alinhamento pode ser controlado com precisão por meio da opção `rotate="valor"`, onde especificamos o ângulo de rotação do nome. Veja a Figura 5.14.

Figura 5.14: Rotacionando o nome de um segmento.

```
\tkzLabelSegment[sloped](A,B){3cm}  \tkzLabelSegment[rotate=-90,below](C,D){2cm}
```

Fonte: Elaborada pelo autor.

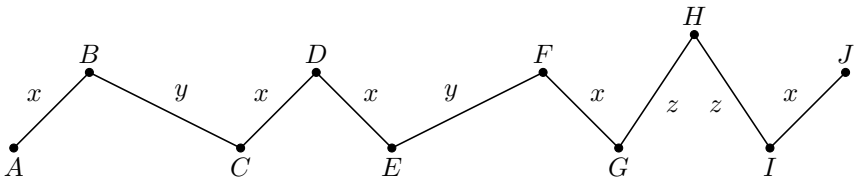
Quando os segmentos são congruentes, é conveniente nomeá-los com o mesmo valor ou com o mesmo símbolo. Nesse caso, podemos fazer uso da macro

$$\texttt{\textbackslash tkzLebelSegments[opções](ponto}_1\texttt{,ponto}_2\ \texttt{...}\ \texttt{ponto}_i\texttt{,ponto}_j\texttt{)\{nome\}}, \tag{5.14}$$

onde os segmentos são indicados pelos pares de referências (pontos) separados por um espaço simples e todos receberão o mesmo nome especificado entre chaves. Veja um exemplo na Figura 5.15, a qual foi construída por meio do código a seguir:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A, 1/1/B, 3/0/C, 4/1/D, 5/0/E, 7/1/F, 8/0/G,
    9/1.5/H, 10/0/I, 11/1/J}
    \tkzDrawSegments(A,B B,C C,D D,E E,F F,G G,H H,I I,J)
    \tkzDrawPoints(A,B,C,D,E,F,G,H,I,J)
    \tkzLabelPoints[below](A,C,E,G,I)
    \tkzLabelPoints[above](B,D,F,H,J)
    \tkzLabelSegments(A,B C,D D,E F,G I,J){$x$}
    \tkzLabelSegments(B,C E,F){$y$}
    \tkzLabelSegments(H,G I,H){$z$}
  \end{tikzpicture}
\end{document}
```

Figura 5.15: Nomeando segmentos congruentes.



Fonte: Elaborada pelo autor.

Marcando segmentos

Também é comum marcar os segmentos para indicar que eles são congruentes. Essa marcação é feita com as macros

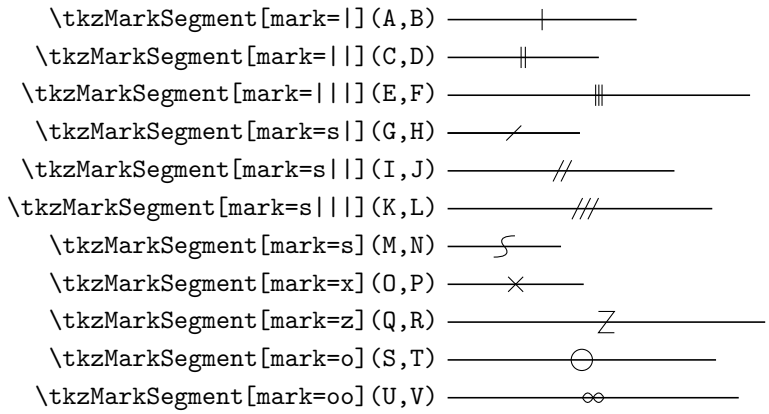
$$\texttt{\textbackslash tkzMarkSegment[opções](ponto}_1\texttt{,ponto}_2\texttt{)} \tag{5.15}$$

e

$$\texttt{\textbackslash tkzMarkSegments[opções](ponto}_1\texttt{,ponto}_2\ \texttt{...}\ \texttt{ponto}_i\texttt{,ponto}_j\texttt{).} \tag{5.16}$$

A macro (5.15) é indicada quando se pretende marcar apenas um segmento, enquanto a macro (5.16) é indicada para marcar vários segmentos com a mesma marcação. Em ambos os casos, é possível especificar no espaço de opções o formato da marcação do segmento colocando `mark=|`, `mark=||`, `mark=|||`, `mark=s|`, `mark=s||`, `mark=s|||`, `mark=s`, `mark=x`, `mark=z`, `mark=o` ou `mark=oo` para obter estilos diferentes de marcações. Veja, na Figura 5.16, o resultado de cada uma dessas opções.

Figura 5.16: Exemplo marcação de segmentos.



Fonte: Elaborada pelo autor.

5.2.4 Ângulo e Arco

Desenhando ângulos

Para desenharmos um ângulo, precisamos de três pontos e utilizamos a macro

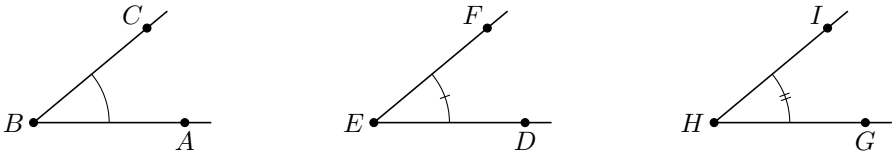
$$\texttt{\textbackslash tkzMarkAngle[opções](ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\texttt{)}, \tag{5.17}$$

onde especificamos, entre parênteses e em ordem anti-horária, as três referências dos pontos sendo, a central, o vértice do ângulo.

As opções permitem controlar características do ângulo, como tamanho do diâmetro, cor e estilo, além de colocar marcações como nas macros (5.15) e (5.16), sendo por padrão, colocada a marcação da opção `mark=|`, mesmo que não seja solicitada. Para desenharmos o ângulo sem marcação, devemos colocar no espaço de opções um dos comandos `mark=` ou `mark=none`. A Figura 5.17 traz três resultados com marcações diferentes.

Figura 5.17: Exemplo desenho e marcação de ângulos.

```
\tkzMarkAngle[mark=](A,B,C) \tkzMarkAngle(D,E,F) \tkzMarkAngle[mark=||](G,H,I)
```

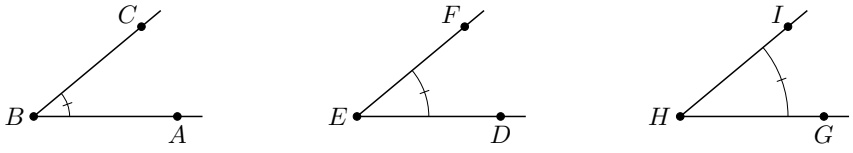


Fonte: Elaborada pelo autor.

O tamanho do diâmetro do ângulo é controlado pela opção `size="valor"` onde podemos utilizar valores em centímetros. Veja a Figura 5.18, onde o ângulo do centro apresenta o tamanho padrão de 1 cm de diâmetro.

Figura 5.18: Exemplo tamanho do diâmetro do ângulo.

```
\tkzMarkAngle[size=0.5cm](A,B,C) \tkzMarkAngle(D,E,F) \tkzMarkAngle[size=1.5cm](G,H,I)
```



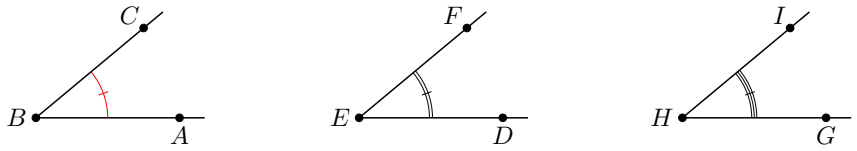
Fonte: Elaborada pelo autor.

A cor da linha do ângulo é alterada por meio da opção `color="cor"` ou, simplesmente, informando a cor desejada no espaço de opções, enquanto o estilo do ângulo é alterado por meio das opções `arc=l`, `arc=ll` ou `arc=lll`, sendo que cada letra "*l*"

equivale a uma linha desenhada no ângulo. A Figura 5.19 traz resultados com cores e estilos diferentes de ângulos.

Figura 5.19: Alterando a cor e o estilo do ângulo.

```
\tkzMarkAngle[red](A,B,C) \tkzMarkAngle[arc=ll](D,E,F)\tkzMarkAngle[arc=lll](G,H,I)
```



Fonte: Elaborada pelo autor.

A fim de economizar linhas, podemos utilizar a macro a seguir para desenhar vários ângulos de uma só vez:

$$\texttt{\textbackslash tkzMarkAngles[opções](ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\ \ldots\ \texttt{ponto}_i\texttt{,ponto}_j\texttt{,ponto}_k\texttt{)}, \tag{5.18}$$

sendo que, cada trio de referências dos pontos representa um ângulo e devem ser separados por um espaço simples. Além disso, todas as opções apresentadas para a macro (5.17) também são válidas, porém, todos os ângulos apresentarão as mesmas características conforme as opções utilizadas.

É comum desenhar o ângulo reto na forma de um quadrado. Essa forma é obtida utilizando-se a macro

$$\texttt{\textbackslash tkzMarkRightAngle[opções](ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\texttt{)} \tag{5.19}$$

para desenharmos um único ângulo reto, ou o comando

$$\texttt{\textbackslash tkzMarkRightAngles[opções](ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\texttt{...ponto}_i\texttt{,ponto}_j\texttt{,ponto}_k\texttt{)}, \tag{5.20}$$

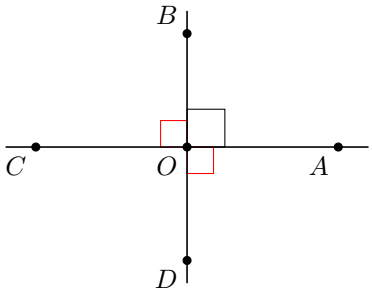
para desenharmos vários ângulos retos ao mesmo tempo.

As opções úteis para as macros (5.19) e (5.20) são `color="cor"` e `size="valor"` que alteram, respectivamente, a cor e o tamanho do raio do ângulo. Veja na Figura 5.20 o resultado do código a seguir, com alguns exemplos de utilização dessas macros e dessas opções.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{5/0/O, 7/0/A, 5/1.5/B, 3/0/C, 5/-1.5/D}
```

```
    \tkzMarkRightAngle[size=0.5](A,O,B)
    \tkzMarkRightAngles[color=red,size=0.35](B,O,C D,O,A)
    \tkzDrawLines[add=0 and 0.2](O,A O,B O,C O,D)
    \tkzDrawPoints(O,A,B,C,D)
    \tkzLabelPoints[below left](O,A,C,D)
    \tkzLabelPoints[above left](B)
  \end{tikzpicture}
\end{document}
```

Figura 5.20: Desenhando ângulos retos.



Fonte: Elaborada pelo autor.

Pintando ângulos

É possível pintar uma região angular com o comando

$$\texttt{\textbackslash tkzFillAngle[opções](ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\texttt{)}, \tag{5.21}$$

ou várias regiões angulares, com o comando

$$\texttt{\textbackslash tkzFillAngles[opções](ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\ \texttt{...ponto}_i\texttt{,ponto}_j\texttt{,ponto}_k\texttt{)}, \tag{5.22}$$

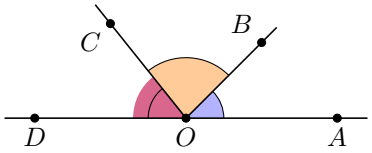
sendo que, devemos escrever a opção `fill="cor"` especificando a cor desejada em `"cor"`.

Caso as macros (5.17) e (5.18) sejam utilizadas em conjunto com as macros (5.21) e (5.22) para desenhar e pintar o mesmo ângulo, devemos nos atentar para a opção `size="valor"` que controla o tamanho do diâmetro do ângulo. O ideal é que se escolha o mesmo valor para evitar que a região angular pintada fique desencontrada com a linha do ângulo desenhado. A Figura 5.21 mostra um ângulo que apresenta esse problema. Isso acontece porque os valores utilizados para `size` nas macros referentes ao ângulo $\angle COD$ (linhas 8 e 9 do código a seguir) são diferentes.

```
\documentclass{standalone}
```

```
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{5/0/O, 7/0/A, 6/1/B, 4/1.25/C, 3/0/D}
    \tkzFillAngles[size=0.5cm,fill=blue!30](A,O,B)
    \tkzFillAngle[size=0.8cm,fill=orange!40](B,O,C)
    \tkzFillAngle[size=0.7cm,fill=purple!60](C,O,D)
    \tkzMarkAngles[size=0.5cm,mark=](A,O,B C,O,D)
    \tkzMarkAngle[size=0.8cm,mark=](B,O,C)
    \tkzDrawLines[add=0 and 0.2](O,A O,B O,C O,D)
    \tkzDrawPoints(O,A,B,C,D)
    \tkzLabelPoints[below](O,A,D)
    \tkzLabelPoints[above left](B)
    \tkzLabelPoints[below left](C)
  \end{tikzpicture}
\end{document}
```

Figura 5.21: Pintando ângulos.



Fonte: Elaborada pelo autor.

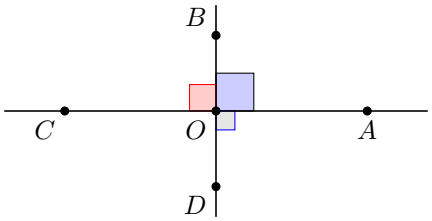
Perceba que nas macros das linhas 6, 7 e 8 do código da Figura 5.21 foi utilizada a opção `fill="cor"` especificando as cores azul (`blue`), laranja (`orange`) e roxo (`purple`) seguidas do sinal de exclamação e de um número. Essa exclamação e esse número controlam o percentual da cor utilizada. Por exemplo, `blue!30` indica que será utilizado 30% da cor azul padrão.

Para pintarmos os ângulos retos, devemos utilizar a opção `fill="cor"` nas macros (5.19) e (5.20). A Figura 5.22 traz um exemplo que foi construído por meio do seguinte código:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/O, 2/0/A, 0/1/B, -2/0/C, 0/-1/D}
    \tkzMarkRightAngle[size=0.5,fill=blue!20](A,O,B)
```

```
    \tkzMarkRightAngle[color=red,size=0.35,fill=red!20](B,O,C)
    \tkzMarkRightAngle[color=blue,size=0.25,fill=gray!20](D,O,A)
    \tkzDrawLines(C,A B,D)
    \tkzDrawPoints(O,A,B,C,D)
    \tkzLabelPoints[below](A)
    \tkzLabelPoints[above left](B)
    \tkzLabelPoints[below left](C,D,O)
  \end{tikzpicture}
\end{document}
```

Figura 5.22: Pintando ângulos retos.



Fonte: Elaborada pelo autor.

Nomeando ângulos

Podemos nomear um único ângulo por meio da macro

$$\texttt{\tkzLabelAngle[opções](ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\texttt{)\{nome\}} \tag{5.23}$$

ou nomear vários ângulos com o mesmo nome, através do comando

$$\texttt{\tkzLabelAngles[opções](ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\texttt{...ponto}_i\texttt{,ponto}_j\texttt{,ponto}_k\texttt{)\{nome\}}. \tag{5.24}$$

As referências dos pontos são organizadas como nas macros (5.17) e (5.18) e podemos controlar características como cor, tamanho e posicionamento do nome em relação ao ângulo desenhado.

O controle da cor é feito indicando a cor desejada no espaço de opções ou através do comando `color="cor"` especificando a cor desejada em `"cor"`.

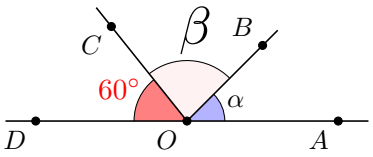
O controle do tamanho do nome é feito por meio da opção `scale="valor"`, sendo que o valor digitado em `"valor"` deve ser em percentual, a omissão dessa opção ou a utilização de `scale=1` desenhará o nome do ângulo em seu tamanho padrão.

Em relação ao posicionamento, podemos utilizar várias opções, como as utilizadas na macro (5.6) (`above`, `below`, `right`, etc.), sendo que para um posicionamento mais preciso aconselha-se utilizar as opções `xshift="valor"` e `yshift="valor"`.

Além disso, podemos fazer uso da opção `pos="valor"`, sendo esta mais indicada para controlar a distância do nome em relação à linha do ângulo desenhado. A Figura 5.23 traz o resultado do código abaixo com a utilização de algumas dessas opções.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{5/0/O, 7/0/A, 6/1/B, 4/1.25/C, 3/0/D}
    \tkzFillAngle[size=0.5cm,fill=blue!30](A,O,B)
    \tkzFillAngle[size=0.8cm,fill=pink!20](B,O,C)
    \tkzFillAngle[size=0.7cm,fill=red!50](C,O,D)
    \tkzMarkAngle[size=0.5cm,mark=](A,O,B)
    \tkzMarkAngle[size=0.8cm,mark=](B,O,C)
    \tkzMarkAngle[size=0.7cm,mark=](C,O,D)
    \tkzMarkAngle[size=0.3cm,mark=](E,O,A)
    \tkzDrawLines[add=0 and 0.2](O,A O,B O,C O,D)
    \tkzDrawPoints(O,A,B,C,D)
    \tkzLabelPoints[below left](A,C,D,O)
    \tkzLabelPoints[above left](B)
    \tkzLabelAngle[pos=0.7](A,O,B){$\alpha$}
    \tkzLabelAngle[scale=2,yshift=-0.4cm](B,O,C){$\beta$}
    \tkzLabelAngle[scale=1.1,pos=0.9,red](C,O,D){$60^\circ$}
  \end{tikzpicture}
\end{document}
```

Figura 5.23: Nomeando ângulos.

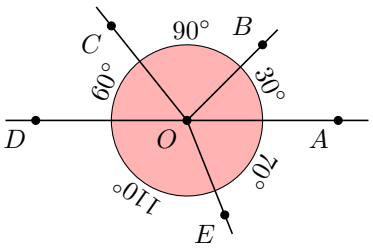


Fonte: Elaborada pelo autor.

Ainda é possível utilizar a opção `rotate="valor"` para rotacionar o nome do ângulo em torno do seu próprio eixo de acordo com o valor, em graus, especificado em `"valor"`. A Figura 5.24 traz o resultado do código completo a seguir, exemplificando o uso dessas opções.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{5/0/O,7/0/A,6/1/B,4/1.25/C,3/0/D,5.5/-1.25/E}
    \tkzFillAngles[fill=red!30](A,O,B B,O,C C,O,D D,O,E E,O,A)
    \tkzMarkAngles[mark=](A,O,B B,O,C C,O,D D,O,E E,O,A)
    \tkzDrawLines[add=0 and 0.2](O,A O,B O,C O,D O,E)
    \tkzDrawPoints(O,A,B,C,D,E)
    \tkzLabelPoints[below left](A,C,D,O,E)
    \tkzLabelPoints[above left](B)
    \tkzLabelAngle[pos=1.2,rotate=-65](A,O,B){$30^\circ$}
    \tkzLabelAngle[pos=1.2](B,O,C){$90^\circ$}
    \tkzLabelAngle[pos=1.2,rotate=65](C,O,D){$60^\circ$}
    \tkzLabelAngle[pos=1.2,rotate=145](D,O,E){$110^\circ$}
    \tkzLabelAngle[pos=1.2,rotate=-120](E,O,A){$70^\circ$}
  \end{tikzpicture}
\end{document}
```

Figura 5.24: Rotacionando nomes de ângulos.



Fonte: Elaborada pelo autor.

Desenhando arcos

A macro responsável por desenhar um arco é

$$\texttt{\textbackslash tkzDrawArc[opções](ponto}_1\texttt{,ponto}_2\texttt{)(ponto}_3\texttt{)}, \tag{5.25}$$

onde $\texttt{ponto}_1$ representa o ponto central do arco e a distância de $\texttt{ponto}_1$ ao $\texttt{ponto}_2$ equivale ao raio do arco que será desenhado no sentido anti-horário do $\texttt{ponto}_2$ para o $\texttt{ponto}_3$.

As opções para os arcos são diversas e controlam características como cor, estilo da linha, espessura e até o estilo das pontas do arco. A Tabela 5.2 apresenta as

principais opções e suas funções.

Tabela 5.2: Resumo das opções para a macro (5.25).

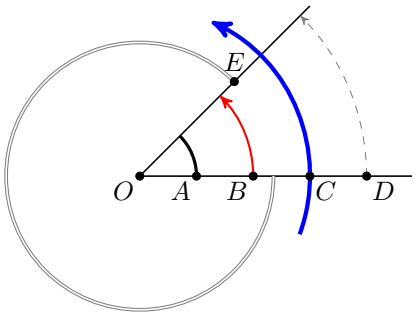
\tkzDrawArc[opções](ponto1,ponto2)(ponto3)		
Opções	**Função**	**Exemplos**
color="cor"	Controla a cor do arco.	*color=red*
dashed	O arco fica tracejado.	-
dotted	O arco fica pontilhado.	-
double	Duplica a linha do arco.	-
thick	O arco fica um pouco mais grosso.	-
very thick	O arco fica ainda mais grosso.	-
ultra thick	O arco fica bem mais grosso.	-
line width="valor"	Controla a espessura do arco.	*line width=0.6mm*
<->	Coloca uma seta nas pontas do arco.	->, <-, <->
delta="valor"	Faz o arco ultrapassar os pontos.	*delta=20*

Fonte: Elaborada pelo autor com base em Matthes [29].

Veja na Figura 5.25, resultados da utilização de algumas dessas opções na macro (5.25). Tal figura foi obtida por meio do seguinte código:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/O,0.75/0/A,1.5/0/B,2.25/0/C,3/0/D,1.25/1.25/E}
    \tkzDrawLine[add=0 and 0.8](O,E)
    \tkzDrawLine[add=0 and 0.2](O,D)
    \tkzDrawArc[color=black,very thick](O,A)(E)
    \tkzDrawArc[color=red,->,>=stealth',thick](O,B)(E)
    \tkzDrawArc[color=blue,->,>=stealth',delta=20,ultra thick](O,C)(E)
    \tkzDrawArc[dashed,->,>=stealth'](O,D)(E)
    \tkzDrawArc[double](O,E)(D)
    \tkzDrawPoints(O,A,B,C,D,E)
    \tkzLabelPoints[below left=-0.1cm](O,A,B)
    \tkzLabelPoints[below right=-0.1cm](C,D)
    \tkzLabelPoints[above](E)
  \end{tikzpicture}
\end{document}
```

Figura 5.25: Desenhando arcos.



Fonte: Elaborada pelo autor.

Existem variações da macro (5.25) que permitem desenhar arcos. Por exemplo a macro `\tkzDrawArc[rotate](ponto1,ponto2)(angulo)`, onde devemos especificar o valor da abertura do ângulo em graus correspondente ao arco. As demais macros e explicações de como utilizá-las podem ser encontrados em Matthes [29].

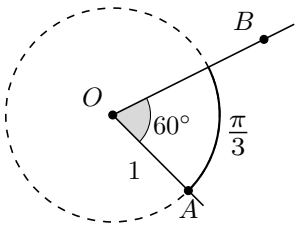
Nomeando arcos

A versão 3.06c do `tkz-euclide` não possui macro para nomear arcos, mas podemos utilizar as macros que nomeiam ângulos (5.23) e (5.24) e ajustar a posição do nome com a opção `pos="valor"`. A Figura 5.26 traz o resultado do código completo a seguir, que exemplifica essa situação, onde o arco $\overset{\frown}{AB}$ é "nomeado" (na verdade é indicada sua medida) com o comando da linha 16 do código.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{5/0/O, 6/-1/A, 7/1/B}
    \tkzFillAngle[size=0.5cm,fill=gray!30](A,O,B)
    \tkzMarkAngle[mark=,size=0.5cm](A,O,B)
    \tkzLabelAngle[pos=0.8](A,O,B){$60^\circ$}
    \tkzDrawCircle[dashed](O,A)
    \tkzDrawLines[add=0 and 0.2](O,A O,B)
    \tkzDrawPoints(O,A,B)
    \tkzLabelPoints[below](A)
    \tkzLabelPoints[above left](O,B)
    \tkzLabelSegment(A,O){1}
    \tkzDrawArc[color=black,thick](O,A)(B)
    \tkzLabelAngle[pos=1.2,scale=1.4](A,O,B){$\frac{\pi}{3}$}
```

```
  \end{tikzpicture}
\end{document}
```

Figura 5.26: Nomeando arcos.



Fonte: Elaborada pelo autor.

5.2.5 Poligonal e Polígono

Desenhando poligonais

A construção de poligonais é feita pela macro

$$\texttt{\textbackslash tkzDrawPolySeg[opções](ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\texttt{,...,ponto}_i\texttt{)}, \tag{5.26}$$

onde especificamos a sequência das referências dos pontos entre parênteses e separadas por vírgula. Essa sequência define por onde a poligonal passará.

Em relação às opções, podemos controlar a cor, estilo da linha, espessura da linha e mudar a cor do interior da poligonal (que é transparente por padrão). A Tabela 5.3 traz um resumo dessas e outas opções.

Tabela 5.3: Resumo das opções para a macro (5.26).

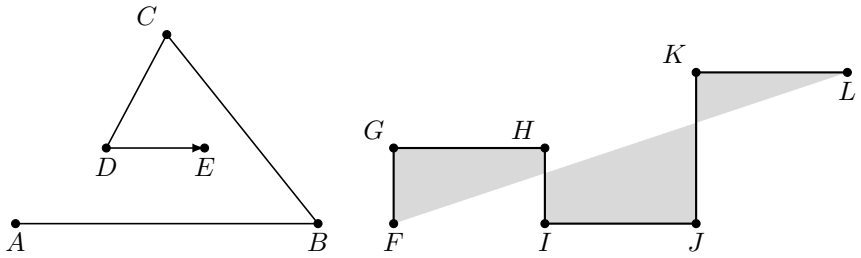
\tkzDrawPolySeg[opções](ponto$_1$,ponto$_2$,ponto$_3$,...,ponto$_i$)		
Opções	**Função**	**Exemplos**
color="cor"	Controla a cor da linha.	*color=blue*
dashed	A linha fica tracejada.	-
fill="color"	Pinta o interior da poligonal fechada.	*fill=gray!20*
dotted	A linha fica pontilhada.	-
double	Duplica a linha da poligonal.	-
thick	A linha fica um pouco mais grossa.	-
very thick	A linha fica ainda mais grossa.	-
ultra thick	A linha fica bem mais grossa.	-
line width="valor"	Controla a espessura da linha.	*line width=0.6mm*
->	Coloca uma seta na ponta da linha.	*->, ->>, -/, ->/*

Fonte: Elaborada pelo autor com base em Matthes [29].

A Figura 5.27 traz o resultado do código completo a seguir, a qual exemplifica a utilização da macro (5.26) e algumas de suas opções.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{-6/-3/A,-2/-3/B,-4/-0.5/C,-4.8/-2/D,-3.5/-2/E,
    -1/-3/F,-1/-2/G,1/-2/H,1/-3/I,3/-3/J,3/-1/K,5/-1/L}
    \tkzDrawPolySeg[->,>=latex](A,B,C,D,E)
    \tkzDrawPolySeg[thick,fill=gray!30](F,...,L)
    \tkzDrawPoints(A,...,L)
    \tkzLabelPoints[below](A,B,D,E,F,I,J,L)
    \tkzLabelPoints[above left](C,G,H,K)
  \end{tikzpicture}
\end{document}
```

Figura 5.27: Desenhando poligonais.



Fonte: Elaborada pelo autor com base em Matthes [29].

Note que, nas macros das linhas 8 e 9 do código que origina a Figura 5.27, algumas das referências dos pontos foram substituídas pela pontuação de reticências. Isso sempre é possível quando as referências estão dispostas em ordem alfabética. Observe também que a opção `fill=gray!30`, utilizada na macro da linha 8, pintou todas as regiões fechadas da poligonal admitindo-se o fechamento da poligonal pelo segmento FL.

Desenhando polígonos

A macro responsável por desenhar polígonos é

$$\texttt{\textbackslash tkzDrawPolygon[opções](ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\texttt{,...,ponto}_i\texttt{)}, \qquad (5.27)$$

onde especificamos, entre parênteses e separados por vírgula, as referências dos pontos representantes dos vértices do polígono.

Em relação às opções, destacamos as que controlam a cor, estilo e espessura da linha do contorno e cor do interior do polígono. A Tabela 5.4 traz um resumos dessas opções e exemplos de como utilizá-las. Caso não seja digitado nenhuma opção, o polígono será desenhado com as características padrões: bordas na cor preta, contínua e com espessura de 0.2 mm e interior totalmente transparente.

Tabela 5.4: Resumo das opções para a macro (5.27).

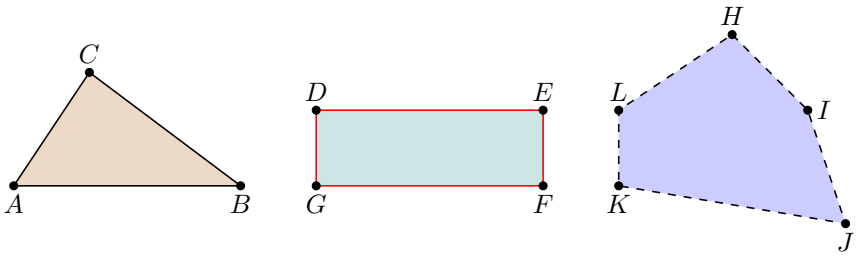
\tkzDrawPolygon[opções](ponto$_1$,ponto$_2$,ponto$_3$,...,ponto$_i$)		
Opções	**Função**	**Exemplos**
draw="cor"	Controla a cor da linha do contorno.	*draw=red*
fill="cor"	Pinta o interior do polígono.	*fill=blue!30*
dashed	A linha do contorno fica tracejada.	-
dotted	A linha do contorno fica pontilhada.	-
double	Duplica a linha do contorno.	-
thick	A linha fica um pouco mais grossa.	-
very thick	A linha fica ainda mais grossa.	-
ultra thick	A linha fica bem mais grossa.	-
line width="valor"	Controla a espessura da linha.	*line width=0.6mm*

Fonte: Elaborada pelo autor com base em Matthes [29] e Tantau [28].

A Figura 5.28 é resultado da compilação do código a seguir, ela ilustra os resultados esperados da utilização da macro (5.27) e algumas de suas opções.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{-1/-1/A, 2/-1/B, 0/0.5/C, 3/0/D, 6/0/E, 6/-1/F,
    3/-1/G,8.5/1/H, 9.5/0/I, 10/-1.5/J, 7/-1/K, 7/0/L}
    \tkzDrawPolygon[fill=brown!30](A,B,C)
    \tkzDrawPolygon[fill=teal!20,draw=red](D,E,F,G)
    \tkzDrawPolygon[fill=blue!20,dashed](H,I,J,K,L)
    \tkzDrawPoints(A,...,L)
    \tkzLabelPoints[below](A,B,F,G,J,K)
    \tkzLabelPoints[above](C,D,E,L,H)
    \tkzLabelPoints[right](I)
  \end{tikzpicture}
\end{document}
```

Figura 5.28: Desenhando polígonos.



Fonte: Elaborada pelo autor.

5.2.6 Triângulos

Para desenharmos um triângulo, temos duas estratégias: por meio de pontos fixos ou por meio de pontos fixos e criados. A primeira estratégia é indicada quando se pretende desenhar triângulos com vértices em coordenadas especificadas manualmente, onde, por ser desenhado utilizando pontos fixos, temos total controle do posicionamento dos vértices. A segunda estratégia é mais indicada quando pretendemos desenhar triângulos que possuam certas características, como os triângulos equiláteros, isósceles e retângulos, ou quando conhecemos alguns de seus elementos como, por exemplo, um ângulo, os vértices de um lado e um ângulo, respectivamente.

Com base em Tantau [28] e Matthes [29], destacamos a seguir como utilizar as duas estratégias.

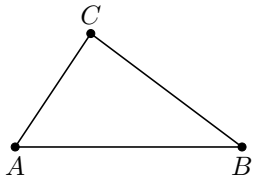
Desenhando triângulos utilizando pontos fixos

Para desenharmos um triângulo por meio de pontos fixos, basta definirmos os três pontos por meio da macro (5.2), os quais corresponderão aos vértices do triângulo. Após a definição dos pontos, desenhamos o triângulo por meio da macro (5.27). A Figura 5.29 representa um triângulo ABC obtido por meio do seguinte código:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{3/1/A, 6/1/B, 4/2.5/C}
    \tkzDrawPolygon(A,B,C)
    \tkzDrawPoints(A,B,C)
    \tkzLabelPoints[below](A,B)
    \tkzLabelPoints[above](C)
  \end{tikzpicture}
```

```
\end{document}
```

Figura 5.29: Desenhando um triângulo por meio de pontos fixos.



Fonte: Elaborada pelo autor.

Pode ser difícil desenharmos triângulos que possuam características específicas por meio de pontos fixos, pois, para isso, devemos saber exatamente as coordenadas dos pontos, por exemplo, dos vértices de um triângulo equilátero. A seguir destacamos as macros para a obtenção desse e de outros tipos de triângulos.

Desenhando triângulos por meio de pontos criados

Para desenharmos triângulos especiais como equiláteros, isósceles, retângulos, etc. podemos fazer uso da macro

$$\texttt{\textbackslash tkzDefTriangle[opções](ponto}_1\texttt{,ponto}_2\texttt{)}, \tag{5.28}$$

onde colocamos entre parênteses as referências de dois pontos predefinidos e, no espaço de opções, especificamos o tipo de triângulo que queremos. Portanto, para utilizarmos essa macro, é obrigatório que se tenha pelo menos dois pontos previamente definidos, o segmento formado pelos dois pontos será um dos lados do triângulo.

Para desenharmos um triângulo, é necessário que tenhamos definido três pontos; o que a macro (5.28) faz é definir (criar) um terceiro ponto que será posicionado convenientemente de forma a construir um triângulo conforme a opção digitada no espaço de opções. As principais opções, e a função de cada uma, estão descritas na Tabela 5.5 onde consideramos que A e B são os dois pontos predefinidos e C, o ponto criado.

Tabela 5.5: Resumo das opções para a macro (5.28).

\tkzDefTriangle[opções](A, B)	
Opções	**Função**
two angles=“ang_1” and “ang_2”	Determina C tal que ABC possua $\angle BAC =$ “ang_1” e $\angle ACB =$ “ang_2” com medidas em graus.
equilateral	Determina C para que ABC seja equilátero.
pythagore	Determina C para que ABC seja retângulo com lados proporcionais a 3, 4 e 5.
school	Determina C para que ABC possua ângulos internos com medidas 30°, 60°e 90°.
half	Determina C para que ABC seja retângulo em B e $AB = 2 \times BC$.

Fonte: Elaborada pelo autor com base em Matthes [29].

Para referenciar o ponto criado pela macro (5.28), devemos colocar imediatamente após seu uso, a macro

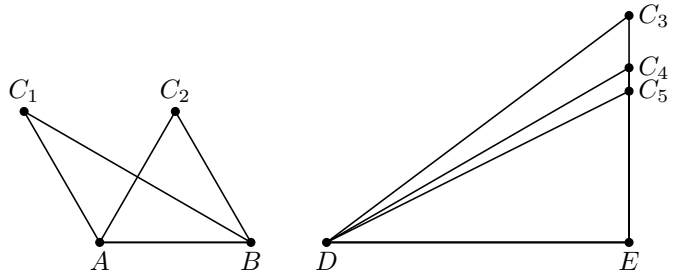
$$\texttt{\textbackslash tkzGetPoint\{ponto}_1\texttt{\}}, \tag{5.29}$$

onde colocamos estre chaves a referência do ponto. A Figura 5.30 traz o resultado do código a seguir.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/1/A, 2/1/B, 3/1/D, 7/1/E}
    \tkzDefTriangle[two angles=120 and 30](A,B)\tkzGetPoint{C_1}
    \tkzDefTriangle[equilateral](A,B)\tkzGetPoint{C_2}
    \tkzDefTriangle[pythagore](D,E)\tkzGetPoint{C_3}
    \tkzDefTriangle[school](D,E)\tkzGetPoint{C_4}
    \tkzDefTriangle[half ](D,E)\tkzGetPoint{C_5}
    \tkzDrawPolygon(A,B,C_1)
    \tkzDrawPolygon(A,B,C_2)
    \tkzDrawPolygon(D,E,C_3)
    \tkzDrawPolygon(D,E,C_4)
    \tkzDrawPolygon(D,E,C_5)
    \tkzDrawPoints(A,B,D,E,C_1,C_2,C_3,C_4,C_5)
```

```
    \tkzLabelPoints[below](A,B,D,E)
    \tkzLabelPoints[above](C_1,C_2)
    \tkzLabelPoints[right](C_3,C_4,C_5)
  \end{tikzpicture}
\end{document}
```

Figura 5.30: Definindo e desenhando triângulos especiais.



Fonte: Elaborada pelo autor com base em Matthes [29].

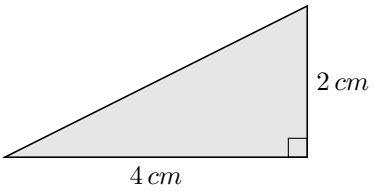
Caso não seja necessário referenciar o ponto criado, podemos desenhar o triângulo fazendo uso do comando

$$\texttt{\tkzDrawTriangle[opções](ponto}_1\texttt{,ponto}_2\texttt{)}, \tag{5.30}$$

onde devemos especificar o tipo de triângulo no espaço de opções, conforme a Tabela 5.5, e os dois pontos predefinidos entre parênteses. Se ainda assim quisermos utilizar o ponto criado, podemos utilizar o comando `tkzPointResult`, sem a barra invertida, para fazer referência ao mesmo dentro de outras macros que devem ser posicionadas imediatamente após a macro (5.30). Além disso, todas as opções apresentadas na Tabela 5.4 são válidas. Veja um exemplo na Figura 5.31, onde não desenhamos os pontos nem os nomeamos. Tal figura foi obtida por meio do código:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{2/1/A, 6/1/B}
    \tkzDrawTriangle[half,fill=gray!20](A,B)
    \tkzMarkRightAngle(A,B,tkzPointResult)
    \tkzLabelSegment[right](tkzPointResult,B){$2\,cm$}
    \tkzLabelSegment(B,A){$4\,cm$}
  \end{tikzpicture}
\end{document}
```

Figura 5.31: Exemplo de utilização do comando *tkzPointResult*.



Fonte: Elaborada pelo autor com base em Matthes [29].

5.2.7 Quadriláteros

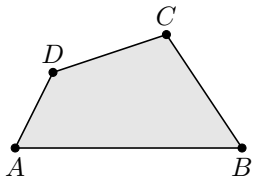
A obtenção de quadriláteros é feita de maneira semelhante aos triângulos: podemos utilizar pontos fixos ou pontos fixos e criados. A seguir descrevemos a utilização dos dois métodos.

Desenhando quadriláteros por meio de pontos fixos

Para desenharmos um quadrilátero por meio de pontos fixos, basta definir os quatro vértices com a macro (5.2) e, após a definição, utilizarmos a macro (5.27). A Figura 5.32 traz um exemplo de quadrilátero desenhado utilizando esse processo, o qual está representado no código a seguir.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/1/A, 3/1/B, 2/2.5/C, 0.5/2/D}
    \tkzDrawPolygon[fill=gray!20](A,B,C,D)
    \tkzDrawPoints(A,B,C,D)
    \tkzLabelPoints[below](A,B)
    \tkzLabelPoints[above](C,D)
  \end{tikzpicture}
\end{document}
```

Figura 5.32: Desenhando um quadrilátero por meio de pontos fixos.



Fonte: Elaborada pelo autor com base em Matthes [29].

Assim como no caso dos triângulos, esse método nos dá liberdade de especificar, por meio de coordenadas, a posição de cada vértice, mas isso pode ser inconveniente quando, por exemplo, desejarmos desenhar um quadrado ou um paralelogramo. Assim, para facilitar o desenho desse tipo de quadrilátero utilizamos macros específicas, as quais apresentamos a seguir.

Desenhando quadrados por meio de pontos fixos e criados

Dados dois pontos fixos, para definirmos um quadrado que os possuam como vértices, utilizamos a macro

$$\texttt{\textbackslash tkzDefSquare(ponto}_1\texttt{,ponto}_2\texttt{)}, \tag{5.31}$$

onde especificamos os dois pontos predefinidos entre parênteses. Como se sabe, para desenharmos um quadrado são necessários quatro vértices, o que essa macro faz é criar os outros dois pontos de forma que eles fiquem posicionados formando um quadrado juntamente com os pontos predefinidos, claro que o comprimento do lado do quadrado dependerá dos dois pontos iniciais.

Para referenciar os dois pontos criados, utilizamos a macro

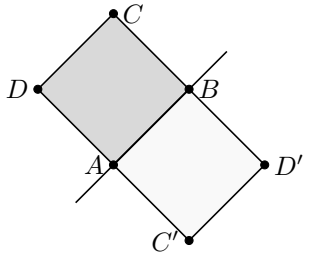
$$\texttt{\textbackslash tkzGetPoints\{ponto}_1\texttt{\}\{ponto}_2\texttt{\}} \tag{5.32}$$

imediatamente após a utilização da macro (5.31) e os pontos serão referenciados no sentido anti-horário, conforme os pontos especificados na macro (5.31) (ver Figura 5.33).

Após a definição dos pontos fixos, da criação dos pontos e da referenciação dos pontos criados, desenhamos o quadrado por meio da macro (5.27). A Figura 5.33 traz o resultado do código a seguir, onde são apresentadas as definições e os desenhos de dois quadrados utilizando esse processo, sendo que o quadrado $ABCD$ foi definido com `\tkzDefSquare(A,B)`, enquanto o quadrado $BAC'D'$ foi definido com `\tkzDefSquare(B,A)`. Observe que os vértices são dados no sentido anti-horário e que a troca da ordem de (A, B) para (B, A) altera o resultado.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/1/A, 1.5/2.5/B}
    \tkzDefSquare(A,B)\tkzGetPoints{C}{D}
    \tkzDrawPolygon[fill=gray!30](A,B,C,D)
    \tkzDefSquare(B,A)\tkzGetPoints{C'}{D'}
    \tkzDrawPolygon[fill=gray!5](B,A,C',D')
    \tkzDrawLine[add=0.5 and 0.5](A,B)
    \tkzDrawPoints(A,B,C,D,C',D')
    \tkzLabelPoints[left](A,C',D)
    \tkzLabelPoints[right](B,C,D')
  \end{tikzpicture}
\end{document}
```

Figura 5.33: Desenho de quadrados por meio de pontos fixos e criados.



Fonte: Elaborada pelo autor com base em Matthes [29].

Se quisermos referenciar separadamente cada ponto criado, devemos utilizar a macro

$$\texttt{\textbackslash tkzGetFirstPoint\{ponto}_1\texttt{\}} \tag{5.33}$$

para o primeiro ponto, e

$$\texttt{\textbackslash tkzGetSecondPoint\{ponto}_2\texttt{\}} \tag{5.34}$$

para o segundo, colocando, em ambos os casos, a referência entre chaves. Tomando como exemplo a Figura 5.33, o primeiro ponto criado pela macro `\tkzDefSquare(A,B)` é o ponto C, enquanto o segundo é o ponto D, os quais poderiam ter sido referenciados, respectivamente, pelos comandos

`\tkzGetFirstPoint{C}` e `\tkzGetSecondPoint{D}`.

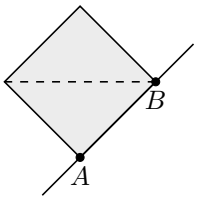
Caso não haja intenção de referenciar ou nomear os pontos criados, ou seja, se a intenção for apenas desenhar um quadrado cujo lado possui o comprimento do segmento formado pelos dois pontos predefinidos, podemos fazer uso do comando

$$\texttt{\textbackslash tkzDrawSquare[opções](ponto}_1\texttt{,ponto}_2\texttt{)}. \tag{5.35}$$

Os dois pontos criados, ainda podem ser utilizados em comandos posicionados especificamente após as macros (5.35) e (5.31) por meio de `tkzFirstPointResult` e `tkzSecondPointResult`, sem a barra invertida. Veja um exemplo na Figura 5.34, obtida com o uso do código completo a seguir, onde desenhamos uma das diagonais do quadrado fazendo uso de `tkzSecondPointResult`.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{1.75/1.5/A, 2.75/2.5/B}
    \tkzDrawSquare[fill=gray!15](A,B)
    \tkzDrawSegment[dashed](B,tkzSecondPointResult)
    \tkzDrawLine[add=0.5 and 0.5](A,B)
    \tkzDrawPoints(A,B)
    \tkzLabelPoints[below](A,B)
  \end{tikzpicture}
\end{document}
```

Figura 5.34: Desenhando um quadrado utilizando *tkzSecondPointResult*.



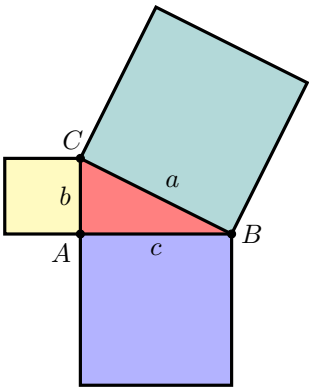
Fonte: Elaborada pelo autor com base em Matthes [29].

As opções apresentadas na Tabela 5.4 também são válidas para a macro (5.35). Veja a Figura 5.35 que representa a interpretação geométrica do Teorema de Pitágoras, obtida por meio do seguinte código:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
```

```
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A, 2/0/B, 0/1/C}
    \tkzFillPolygon[fill=red!50](A,B,C)
    \tkzDrawSquare[fill=teal!30,very thick](C,B)
    \tkzDrawSquare[fill=blue!30,very thick](B,A)
    \tkzDrawSquare[fill=yellow!30,very thick](A,C)
    \tkzDrawPoints(A,B,C)
    \tkzLabelPoints[below left](A)
    \tkzLabelPoints[right](B)
    \tkzLabelPoints[above,xshift=-0.1cm](C)
    \tkzLabelSegment(C,B){$a$}
    \tkzLabelSegment(A,C){$b$}
    \tkzLabelSegment(B,A){$c$}
  \end{tikzpicture}
\end{document}
```

Figura 5.35: Interpretação geométrica do Teorema de Pitágoras.



Fonte: Elaborada pelo autor com base em Matthes [29].

Desenhando paralelogramos por meio de pontos fixos e criados

Para definirmos um paralelogramo devemos ter três pontos fixos, o quarto ponto será obtido por meio da macro

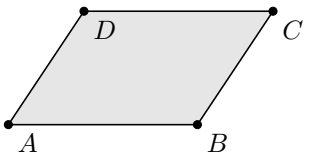
$$\texttt{\tkzDefParallelogram(ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\texttt{)}, \tag{5.36}$$

sendo que o paralelogramo será construído considerando a ordem dos pontos especificados na macro (5.36) e o ponto criado, isto é, o paralelogramo possuirá os vértices na ordem "ponto$_1$", "ponto$_2$", ponto$_3$" e "ponto criado".

Para referenciar o ponto criado, utilizamos a macro (5.29) que deve ser usada imediatamente após a macro (5.36). Observe o código a seguir, ele dá origem à Figura 5.36. Nesse código, os vértices do paralelogramo $ABCD$ são tais que A, B e C são os pontos fixos e D é o ponto criado.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A, 2.5/0/B, 3.5/1.5/C}
    \tkzDefParallelogram(A,B,C)\tkzGetPoint{D}
    \tkzDrawPolygon[fill=gray!20](A,B,C,D)
    \tkzDrawPoints(A,B,C,D)
    \tkzLabelPoints(A,B,C,D)
  \end{tikzpicture}
\end{document}
```

Figura 5.36: Desenhando um paralelogramo.



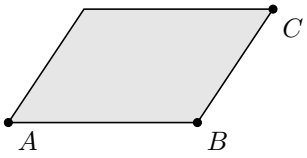
Fonte: Elaborada pelo autor com base em Matthes [29].

A versão 3.06c do `tkz-euclide` não possui uma macro para desenhar um paralelogramo sem antes defini-lo com o comando (5.36), mas é possível utilizar os pontos criados sem a necessidade de criar uma referência. Para isso, basta utilizar `tkzPointResult` dentro dos comandos posicionados após a macro (5.36). Por exemplo, poderíamos obter um paralelogramo idêntico ao da Figura 5.36, substituindo o código que o gera pelo código a seguir. O resultado do novo código está representado na Figuras 5.37.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A, 2.5/0/B, 3.5/1.5/C}
    \tkzDefParallelogram(A,B,C)
    \tkzDrawPolygon[fill=gray!20](A,B,C,tkzPointResult)
```

```
    \tkzDrawPoints(A,B,C)
    \tkzLabelPoints(A,B,C)
  \end{tikzpicture}
\end{document}
```

Figura 5.37: Desenhando um paralelogramo utilizando *tkzPointResult*.



Fonte: Elaborada pelo autor com base em Matthes [29].

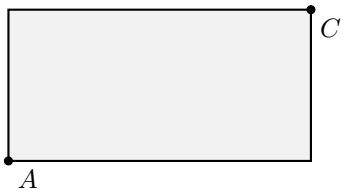
Para obtermos um retângulo, basta definir as coordenadas dos pontos fixos na macro (5.36) de forma a determinarem um ângulo reto ou, ainda, desenhar um retângulo por meio da macro

$$\texttt{\textbackslash tkzDrawRectangle(ponto}_1\texttt{,ponto}_2\texttt{)}, \tag{5.37}$$

sendo necessário definir previamente dois pontos e colocar entre parênteses suas referências. O retângulo será determinado pela diagonal representada pelo segmento formado pelos dois pontos fixos. Observe na Figura 5.38 a obtenção de um retângulo construído com base na diagonal AC, conforme a seguinte sequência de comandos:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A, 4/2/C}
    \tkzDrawRectangle[fill=gray!10,thick](A,C)
    \tkzDrawPoints(A,C)
    \tkzLabelPoints(A,C)
  \end{tikzpicture}
\end{document}
```

Figura 5.38: Desenhando um retângulo.



Fonte: Elaborada pelo autor com base em Matthes [29].

Para obter outros tipos especiais de quadriláteros, como losango e trapézio, podemos definir os seus vértices por meio de pontos fixos ou utilizar as macros da Subseção 5.2.10, pois não existem na versão 3.06c do `tkz-euclide` macros que possuam finalidades específicas para definição e desenho desses quadriláteros.

5.2.8 Polígonos Regulares

A maneira mais simples de obtermos um polígono regular é por meio da macro

$$\texttt{\textbackslash tkzDefRegPolygon[opções](ponto}_1\texttt{,ponto}_2\texttt{)}, \tag{5.38}$$

onde devemos especificar no espaço de opções a quantidade de lados do polígono por meio de `sides=n`, sendo `n` um número natural maior que 2.

Para a utilização dessa macro é necessário que sejam definidos previamente dois pontos que devem ser digitados entre parênteses, sendo que o segmento formado por eles pode representar o raio do polígono, se for digitado nas opções `center`, ou um dos lados do polígono, se for digitado `side`. Caso nenhuma dessas opções seja digitada, por padrão, será assumida a opção `center`.

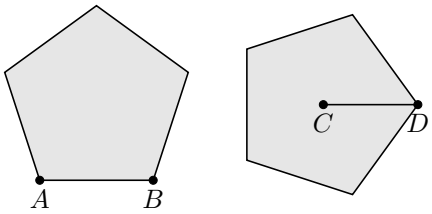
As referências dos vértices do polígono são definidas por padrão como `P1`, `P2`, ..., `Pn`, mas podem ser alteradas por meio da opção `name=X` passando a ser `X1`, `X2`, ..., `Xn`, conforme o que for digitado em `X`.

A macro (5.38) apenas cria os vértices do polígono, para desenhá-lo é necessário utilizar o comando (5.27). Veja a Figura 5.39, a qual é o resultado do código a seguir que define e desenha dois pentágonos regulares, um utilizando a opção `side` e outro utilizando a opção `center`.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A, 1.5/0/B, 3.75/1/C, 5/1/D}
    \tkzDefRegPolygon[side,sides=5](A,B)
```

```
    \tkzDefRegPolygon[center,sides=5,name=X](C,D)
    \tkzDrawPolygon[fill=gray!20](P1,P...,P5)
    \tkzDrawPolygon[fill=gray!20](X1,X...,X5)
    \tkzDrawPoints(A,B,C,D)
    \tkzLabelPoints[below](A,B,C,D)
    \tkzDrawSegment(C,D)
  \end{tikzpicture}
\end{document}
```

Figura 5.39: Desenhando polígonos regulares.



Fonte: Elaborada pelo autor com base em Matthes [29].

Quando utilizamos a opção `center`, o primeiro ponto especificado na macro (5.38) representa o centro do polígono. Caso queiramos determinar o ponto central quando utilizamos a opção `side`, devemos digitar a macro (5.29) imediatamente após a definição do polígono. Além disso, pode ser útil nomear os vértices do polígono, essa ação pode ser realizada utilizando a macro (5.6) para cada vértice. Podemos ainda utilizar a macro

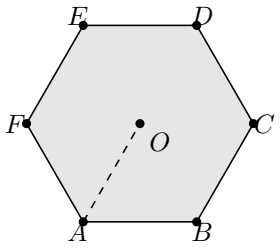
$$\texttt{\textbackslash tkzLabelRegPolygon(centro)\{nome}_1\texttt{,nome}_2\texttt{,...,nome}_n\texttt{\}}, \tag{5.39}$$

para nomear todos os vértices de uma só vez, colocando entre parênteses a referência do centro do polígono e os nomes dos vértices entre chaves. Observe o resultado no hexágono da Figura 5.40, a qual foi construída pelas seguintes macros:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A, 1.5/0/B}
    \tkzDefRegPolygon[side,sides=6](A,B)\tkzGetPoint{O}
    \tkzLabelRegPolygon(O){A,...,F}
    \tkzDrawPolygon[fill=gray!20](P1,P...,P6)
    \tkzDrawPoints(P1,P...,P6,O)
```

```
    \tkzDrawSegment[dashed](A,O)
    \tkzLabelPoints(O)
  \end{tikzpicture}
\end{document}
```

Figura 5.40: Nomeando os vértices de um polígono regular de uma só vez.

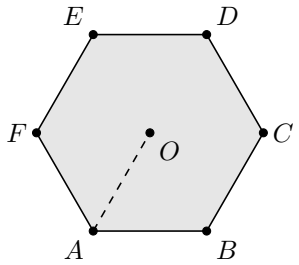


Fonte: Elaborada pelo autor.

A utilização da macro (5.39) pode gerar resultados indesejados em relação ao posicionamento dos nomes dos vértices. Observe, na Figura 5.40, que as letras ficaram muito próximas dos pontos. Para obtermos resultados mais exatos, nomeamos cada vértice por meio da macro (5.6) com o custo de aumentar a quantidade de linhas do código. Veja a Figura 5.41, ela foi obtida por meio dos seguintes comandos:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A, 1.5/0/B}
    \tkzDefRegPolygon[side,sides=6](A,B)\tkzGetPoint{O}
    \tkzDrawPolygon[fill=gray!20](P1,P...,P6)
    \tkzDrawPoints(P1,P...,P6,O)
    \tkzDrawSegment[dashed](A,O)
    \tkzLabelPoints(O)
    \tkzLabelPoint[below left](P1){$A$}
    \tkzLabelPoint[below right](P2){$B$}
    \tkzLabelPoint[right](P3){$C$}
    \tkzLabelPoint[above right](P4){$D$}
    \tkzLabelPoint[above left](P5){$E$}
    \tkzLabelPoint[left](P6){$F$}
  \end{tikzpicture}
\end{document}
```

Figura 5.41: Nomeando um vértice de cada vez de um polígono regular.



Fonte: Elaborada pelo autor.

5.2.9 Círculos

Desenhando círculos

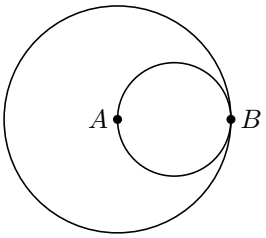
Para desenharmos um círculo, utilizamos a macro

$$\texttt{\textbackslash tkzDrawCircle[opções](ponto}_1\texttt{,ponto}_2\texttt{)}, \tag{5.40}$$

onde devemos especificar entre parênteses duas referências de pontos predefinidos. O segmento formado por esses pontos representará, por padrão, o raio do círculo, sendo o primeiro ponto o centro, e o segundo, um ponto sobre a circunferência. No entanto, é possível desenhar um círculo que possua o diâmetro representado pelo segmento formado pelos dois pontos. Para isso, basta escrever `diameter` no espaço de opções da macro. A Figura 5.42 foi obtida pelo código a seguir, onde obtivemos dois círculos, em um deles, o segmento AB representa o raio e no outro, o diâmetro.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture
    \tkzDefPoints{0/0/A, 1.5/0/B}
    \tkzDrawCircle(A,B)
    \tkzDrawCircle[diameter](A,B)
    \tkzDrawPoints(A,B)
    \tkzLabelPoints[left](A)
    \tkzLabelPoints[right](B)
  \end{tikzpicture}
\end{document}
```

Figura 5.42: Desenhando um círculo dados dois pontos.



Fonte: Elaborada pelo autor com base em Matthes [29].

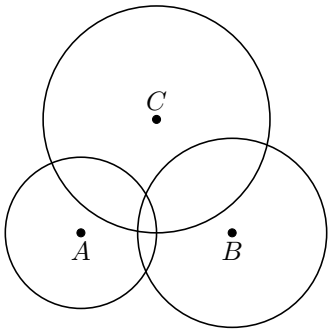
É possível desenhar um círculo com raio de medida específica. Para isso, basta escrever nas opções da macro (5.40) a letra R e especificar, no lugar da segunda referência do ponto predefinido e em centímetros, o tamanho desejado do raio, isto é, a macro (5.40) sofre uma alteração para

$$\texttt{\textbackslash tkzDrawCircle[R,opções](ponto}_1\texttt{,"valor").} \tag{5.41}$$

Veja um exemplo da utilização dessa opção na Figura 5.43 onde foram desenhados três círculos centrados em A, B e C com raios de comprimentos respectivamente iguais a 1 cm, 1.25 cm e 1.5 cm, conforme os comandos a seguir.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A, 2/0/B, 1/1.5/C}
    \tkzDrawCircle[R](A,1cm)
    \tkzDrawCircle[R](B,1.25cm)
    \tkzDrawCircle[R](C,1.5cm)
    \tkzDrawPoints(A,B,C)
    \tkzLabelPoints[below](A,B)
    \tkzLabelPoints[above](C)
  \end{tikzpicture}
\end{document}
```

Figura 5.43: Desenhando círculos dado o centro e a medida do raio.



Fonte: Elaborada pelo autor com base em Matthes [29].

Também é possível desenhar vários círculos de uma só vez utilizando a macro

$$\texttt{\textbackslash tkzDrawCircles[opções](ponto}_1\texttt{,ponto}_2\ \texttt{ponto}_3\texttt{,ponto}_4\ \texttt{...ponto}_i\texttt{,ponto}_j\texttt{)} \tag{5.42}$$

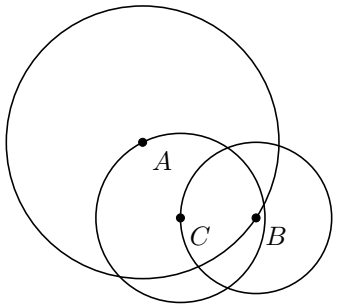
ou

$$\texttt{\textbackslash tkzDrawCircles[R,opções](ponto}_1\texttt{,"valor}_1\texttt{"}\ \texttt{...}\ \texttt{ponto}_i\texttt{,"valor}_i\texttt{"), } \tag{5.43}$$

onde devemos, nos dois casos, separar com um espaço simples os pares de referências $\texttt{ponto}_i\texttt{,ponto}_j$ e $\texttt{ponto}_i\texttt{,"valor}_i\texttt{"}$. Veja um exemplo na Figura 5.44, obtida por meio da sequência de comando a seguir.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{-0.5/1/A, 1/0/B, 0/0/C}
    \tkzDrawCircles(A,B B,C C,A)
    \tkzDrawPoints(A,B,C)
    \tkzLabelPoints(A,B,C)
  \end{tikzpicture}
\end{document}
```

Figura 5.44: Desenhando vários círculos.



Fonte: Elaborada pelo autor com base em Matthes [29].

Existem alguns círculos especiais que ficam situados em regiões específicas em relação a um triângulo. Por exemplo, os círculos circunscrito, inscrito e exinscrito de um triângulo. Para desenhar esses círculos utilizamos a macro

$$\texttt{\textbackslash tkzDrawCircle[opções](ponto}_1\texttt{,ponto}_2\texttt{,ponto}_3\texttt{)}, \tag{5.44}$$

sendo as referências dos vértices do triângulo colocados entre parênteses e, no espaço de opções, especificamos o círculo que queremos desenhar, conforme a Tabela 5.6.

Tabela 5.6: Resumo das opções para a macro (5.44).

\tkzDrawCircle[opções](ponto$_1$,ponto$_2$,ponto$_3$)	
Opções	**Função**
circum	Desenha o círculo circunscrito ao triângulo.
in	Desenha o círculo inscrito ao triângulo.
ex	Desenha o círculo ex-inscrito de um triângulo em relação ao lado formado pelo primeiro e terceiro ponto especificado na macro.

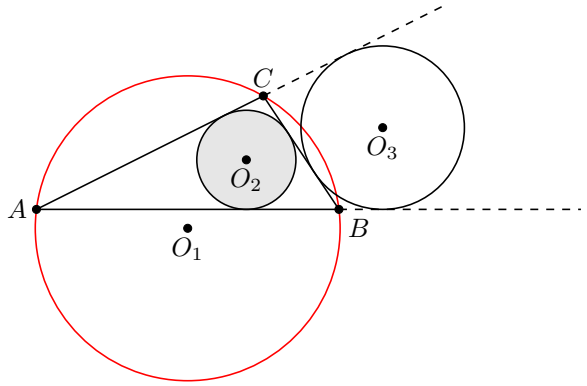
Fonte: Elaborada pelo autor com base em Matthes [29].

A Figura 5.45 traz o resultado do código completo a seguir, que exemplifica a obtenção dos triângulos circunscrito, inscrito e exinscrito (em relação ao lado BC) do triângulo ABC, onde os centros foram obtidos por meio da macro `\tkzGetPoint{ponto`$_1$`}` posicionada imediatamente após o comando (5.44).

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A, 4/0/B, 3/1.5/C}
    \tkzDrawCircle[circum,draw=red](A,B,C)\tkzGetPoint{O_1}
```

```
    \tkzDrawCircle[in, fill=gray!20](A,B,C)\tkzGetPoint{O_2}
    \tkzDrawCircle[ex](B,A,C)\tkzGetPoint{O_3}
    \tkzDrawPolygon(A,B,C)
    \tkzDrawSegments[add=-1 and 0.8,dashed](A,B A,C)
    \tkzDrawPoints(A,B,C,O_1,O_2,O_3)
    \tkzLabelPoints[left](A)
    \tkzLabelPoints(B)
    \tkzLabelPoints[above](C)
    \tkzLabelPoints[below](O_1,O_2,O_3)
  \end{tikzpicture}
\end{document}
```

Figura 5.45: Desenhando círculos circunscrito, inscrito e exinscrito.



Fonte: Elaborada pelo autor com base em Matthes [29].

É válido ressaltar que em todas as macros utilizadas para desenhar círculos são válidas as opções da Tabela 5.4, que alteram características como cor e estilo da linha da circunferência e preenchimento do círculo, conforme exemplificado na Figura 5.45.

Pintando círculos

Como vimos na Figura 5.45, podemos pintar o círculo com a opção `fill="cor"` nas macros destinadas ao desenho do mesmo, mas caso a intenção seja pintar apenas o interior do círculo e não desenhar a circunferência, devemos utilizar a macro

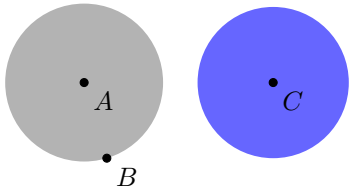
$$\texttt{\textbackslash tkzFillCircle[opções](ponto}_1\texttt{,ponto}_2\texttt{)}, \tag{5.45}$$

onde especificamos a cor nas opções por meio de `color="cor"`, `fill="cor"` ou simplesmente escrevendo a cor desejada. Essa macro funciona de forma semelhante ao comando (5.40) podendo ser adaptada para uma macro que use um ponto (centro)

e a medida do raio, assim como na macro (5.41), porém, não podemos utilizar a opção `diameter`. Veja como exemplo a Figura 5.46, ela foi obtida por meio dos comandos:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/2.5/A, 0.3/1.5/B, 2.5/2.5/C}
    \tkzFillCircle[color=gray!60](A,B)
    \tkzFillCircle[R,color=blue!60](C,1cm)
    \tkzDrawPoints(A,B,C)
    \tkzLabelPoints(A,B,C)
  \end{tikzpicture}
\end{document}
```

Figura 5.46: Pintando círculos.



Fonte: Elaborada pelo autor com base em Matthes [29].

Nomeando círculos

Geralmente nos referimos a um círculo por meio de seu centro ou através do segmento que representa seu raio, mas caso seja necessário nomear o círculo com uma letra (ou até mesmo uma frase), podemos fazer uso da macro

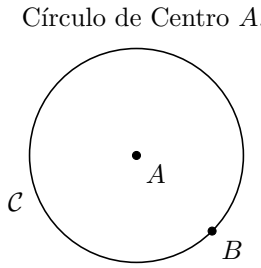
$$\texttt{\textbackslash tkzLabelCircle[opções](ponto}_1\texttt{,ponto}_2\texttt{)(ângulo)\{nome\}}, \tag{5.46}$$

onde especificamos entre parênteses, respectivamente, o centro e um ponto sobre a circunferência. Em seguida, também entre parênteses, especificamos o ângulo, em graus no sentido anti-horário de posicionamento do nome e, finalmente, entre chaves escrevemos o nome desejado do círculo.

Em relação às opções, podemos utilizar todas as descritas para pontos, onde controlamos a cor, o tamanho e o posicionamento do nome, de acordo com a Tabela 5.1. Veja um exemplo na Figura 5.47, a qual foi construída por meio dos seguintes comandos:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/1/A, 1/0/B}
    \tkzDrawCircle(A,B)
    \tkzDrawPoints(A,B)
    \tkzLabelPoints(A,B)
    \tkzLabelCircle[below left](A,B)(240){$\mathcal{C}$}
    \tkzLabelCircle[above,yshift=0.3cm,xshift=-0.5cm](A,B)(110){
    Círculo de Centro $A$.}
  \end{tikzpicture}
\end{document}
```

Figura 5.47: Nomeando círculos.



Fonte: Elaborada pelo autor com base em Matthes [29].

5.2.10 Comandos para definições e desenhos de objetos a partir de outros já definidos

Como dito no início desse capítulo, os pontos "criados" são aqueles obtidos por meio da utilização de outros pontos ou elementos predefinidos. Em algumas situações, a criação desses pontos auxiliam no desenho de figuras que possuem determinadas características. Essa subseção é destinada à exploração de macros que possuem fins de criação de pontos, bem como à apresentação de exemplos de utilização das mesmas em construções de figuras.

Determinando pontos sobre uma reta: ponto médio e outros

Dada uma reta definida por meio de dois pontos, podemos determinar um ponto sobre essa reta fazendo uso da macro

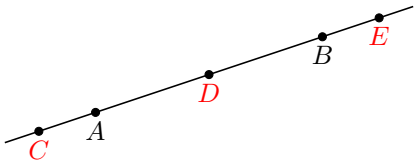
$$\texttt{\textbackslash tkzDefPointOnLine[opções](ponto}_1\texttt{,ponto}_2\texttt{)}, \tag{5.47}$$

sendo colocados entre parênteses as referências dos pontos predefinidos por onde a reta incide. No espaço de opções, colocamos `pos="valor"`, onde especificamos o percentual do segmento no qual o ponto será criado, sendo que o cálculo da posição será no sentido do primeiro ponto para o segundo, conforme especificado na macro.

Observe na Figura 5.48, obtida pelo código a seguir, que valores negativos para a opção `pos` criam pontos posicionados antes do primeiro ponto e valores maiores do que 1, criam pontos após o segundo. Além disso, a macro (5.47) apenas cria o ponto. Para referenciá-lo, utilizamos o comando `\tkzGetPoint{ponto}`$_1$.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A, 3/1/B}
    \tkzDefPointOnLine[pos=-0.25](A,B)\tkzGetPoint{C}
    \tkzDefPointOnLine[pos=0.5](A,B)\tkzGetPoint{D}
    \tkzDefPointOnLine[pos=1.25](A,B)\tkzGetPoint{E}
    \tkzDrawLine[add=0.4 and 0.4](A,B)
    \tkzDrawPoints(A,B,C,D,E)
    \tkzLabelPoints[below](A,B)
    \tkzLabelPoints[below,red](C,D,E)
  \end{tikzpicture}
\end{document}
```

Figura 5.48: Criando pontos sobre uma reta.



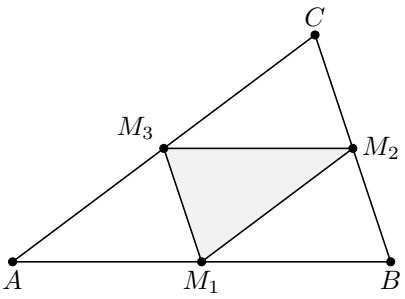
Fonte: Elaborada pelo autor com base em Matthes [29].

A macro (5.47) é útil para obtermos pontos médios de segmentos. Para isso, basta colocar a opção `pos=0.5`, como fizemos na linha 7 do código da Figura 5.48 quando criamos o ponto D. A obtenção do ponto médio é muito importante para desenhar várias figuras geométricas. Um exemplo está na Figura 5.49, onde obtivemos o triângulo médio $M_1M_2M_3$ em relação ao triângulo maior ABC, conforme a sequência das macros a seguir.

```
\documentclass{standalone}
```

```
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A, 5/0/B, 4/3/C}
    \tkzDefPointOnLine[pos=0.5](A,B)\tkzGetPoint{M_1}
    \tkzDefPointOnLine[pos=0.5](B,C)\tkzGetPoint{M_2}
    \tkzDefPointOnLine[pos=0.5](C,A)\tkzGetPoint{M_3}
    \tkzDrawPolygon(A,B,C)
    \tkzDrawPolygon[fill=gray!10](M_1,M_2,M_3)
    \tkzDrawPoints(A,B,C,M_1,M_2,M_3)
    \tkzLabelPoints[below](A,B,M_1)
    \tkzLabelPoints[above](C)
    \tkzLabelPoints[right](M_2)
    \tkzLabelPoints[above left](M_3)
  \end{tikzpicture}
\end{document}
```

Figura 5.49: Triângulo médio.



Fonte: Elaborada pelo autor com base em Matthes [29].

Como vimos, a obtenção de pontos sobre uma reta é simples e pode ser utilizada conforme a necessidade, como na obtenção de pontos médios ou determinação de pontos equidistantes sobre um reta. Nessa segunda situação, aconselha-se utilizar a opção `pos` com valores em forma de frações. Por exemplo, a utilização de `pos=1/3` e `pos=2/3` em macros do tipo (5.47) e com mesmas referências, criam dois pontos que dividem o segmento em três pedaços iguais.

Determinando interseções entre retas e círculos

Determinar interseções pode ser importante para desenhos específicos. Essas interseções podem se dar entre retas, entre círculos ou entre retas e círculos.

Para obtermos a interseção entre duas retas concorrentes utilizamos a macro

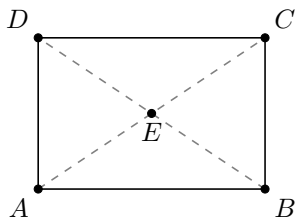
$$\texttt{\textbackslash tkzInterLL(ponto}_1\texttt{,ponto}_2\texttt{)(ponto}_3\texttt{,ponto}_4\texttt{)}, \tag{5.48}$$

onde colocamos entre parênteses, respectivamente, as referências dos pontos por onde as retas incidem.

Como a interseção entre duas retas concorrentes é um único ponto, utilizamos apenas `\tkzGetPoint{ponto`$_1$`}` para referenciá-lo ou `tkzPointResult` dentro de outras macros, caso não seja necessário nomeá-lo. Veja a Figura 5.50, onde o ponto E foi criado com objetivo de obter a interseção das diagonais do retângulo $ABCD$, conforme o código a seguir.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/0/A, 3/0/B, 3/2/C, 0/2/D}
    \tkzDrawPolygon(A,B,C,D)
    \tkzDrawSegments[dashed,draw=gray](A,C B,D)
    \tkzInterLL(A,C)(B,D)\tkzGetPoint{E}
    \tkzDrawPoints(A,B,C,D,E)
    \tkzLabelPoints[below left](A)
    \tkzLabelPoints[below right](B)
    \tkzLabelPoints[above right](C)
    \tkzLabelPoints[above left](D)
    \tkzLabelPoints[below](E)
  \end{tikzpicture}
\end{document}
```

Figura 5.50: Interseção entre retas.



Fonte: Elaborada pelo autor com base em Matthes [29].

É importante ressaltar que, caso as retas determinadas pelos pares de pontos informados na macro (5.48) sejam paralelas, será informado um erro, pois não existirá

ponto de interseção.

Em relação a interseções entre círculos, utilizamos a macro

$$\texttt{\textbackslash tkzInterCC(ponto}_1\texttt{,ponto}_2\texttt{)(ponto}_3\texttt{,ponto}_4\texttt{)}, \tag{5.49}$$

onde os pares de pontos entre parênteses representam os raios dos círculos, onde o primeiro ponto de cada par representa o centro.

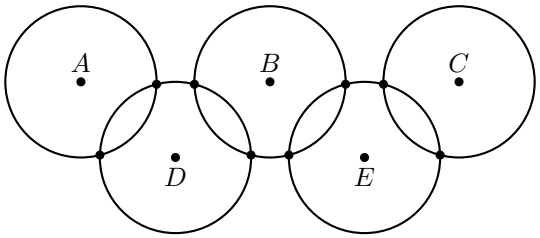
Também é possível obter as interseções de dois círculos quando conhecemos o centro e o valor do raio. Para isso, utilizamos a macro

$$\texttt{\textbackslash tkzInterCC[R](ponto}_1\texttt{,"valor")(ponto}_2\texttt{,"valor")}. \tag{5.50}$$

Em ambos os casos, os pontos de interseções são referenciados com o comando (5.32) ou, caso queira obter separadamente cada ponto, utilizamos os comandos (5.33) e (5.34). É claro que duas circunferências podem ter dois, apenas um ou nenhum ponto em comum. Quando o segundo caso ocorre, os dois pontos referenciados com as macros citadas coincidem. No terceiro caso, será exibida uma mensagem de erro ao compilar o código, pois não existe interseção. Observe, na Figura 5.51, a obtenção das interseções dos círculos dados. Tal figura foi construída por meio do seguinte código:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/3/A, 2.5/3/B, 5/3/C, 1.25/2/D, 3.75/2/E}
    \tkzDrawCircles[R,thick](A,1cm B,1cm C,1cm D,1cm E,1cm)
    \tkzInterCC[R](A,1cm)(D,1cm)\tkzGetPoints{F}{G}
    \tkzInterCC[R](B,1cm)(E,1cm)\tkzGetPoints{H}{I}
    \tkzInterCC(B,H)(D,F)\tkzGetPoints{J}{K}
    \tkzInterCC[R](C,1cm)(E,1cm)\tkzGetPoints{L}{M}
    \tkzDrawPoints(A,B,C,D,E,F,G,H,I,J,K,L,M)
    \tkzLabelPoints[above](A,B,C)
    \tkzLabelPoints[below](D,E)
  \end{tikzpicture}
\end{document}
```

Figura 5.51: Interseção entre círculos.



Fonte: Elaborada pelo autor com base em Matthes [29].

A obtenção das interseções entre uma reta e um círculo é feita de forma semelhante à apresentada nos comandos (5.49) e (5.50), porém devemos utilizar a macro

$$\texttt{\textbackslash tkzInterLC(ponto}_1\texttt{,ponto}_2\texttt{)(ponto}_3\texttt{,ponto}_4\texttt{)}, \tag{5.51}$$

se conhecemos dois pontos sobre a reta e o centro e um ponto sobre o círculo, ou a macro

$$\texttt{\textbackslash tkzInterLC[R](ponto}_1\texttt{,ponto}_2\texttt{)(ponto}_3\texttt{,"valor")}, \tag{5.52}$$

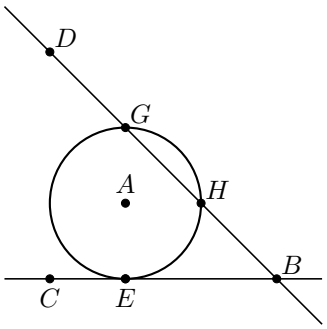
caso o círculo seja definido pelo centro e o valor do raio.

Assim como no caso de interseções de círculos, referenciamos os pontos obtidos na interseção entre uma reta e um círculo por meio do comando (5.32) ou, caso queiramos obter separadamente cada ponto, utilizamos os comandos (5.33) e (5.34). Há também três casos possíveis: a interseção ser composta por dois, um ou nenhum ponto. No segundo caso, os dois pontos referenciados coincidem e no terceiro caso será apresentada uma mensagem de erro ao compilar o código. A Figura 5.52 mostra um exemplo de utilização das macros (5.51) e (5.52), a qual foi produzida por meio da seguinte sequência de macros:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/3/A, 2/2/B, -1/2/C, -1/5/D, 3/2.25/E}
    \tkzDrawCircles[R,thick](A,1cm)
    \tkzDrawLines(C,B B,D)
    \tkzInterLC[R](C,B)(A,1cm)\tkzGetPoints{E}{F}
    \tkzInterLC(B,D)(A,E)\tkzGetPoints{G}{H}
    \tkzDrawPoints(A,B,C,D,E,F,G,H)
    \tkzLabelPoints[above](A)
    \tkzLabelPoints[below](C,E)
```

```
    \tkzLabelPoints[above right=-0.1cm](B,D,G,H)
  \end{tikzpicture}
\end{document}
```

Figura 5.52: Interseção entre retas e círculos.



Fonte: Elaborada pelo autor com base em Matthes [29].

Obtendo pontos por translação

Como vimos no Capítulo 3, a translação é uma isometria que depende de um vetor. Ela determina imagens dos pontos sobre um plano, segundo as características desse vetor. Assim, para obtermos por meio do `tkz-euclide` pontos por translações, devemos adotar um segmento de reta como representante do vetor a ser considerado na transformação. Isso significa que inicialmente é obrigatório que se tenha no mínimo três pontos predefinidos: os dois que determinam o segmento representante do vetor e um terceiro que sofrerá translação, isto é, do qual será calculada a imagem por translação. A macro responsável por determinar as imagens dos pontos é

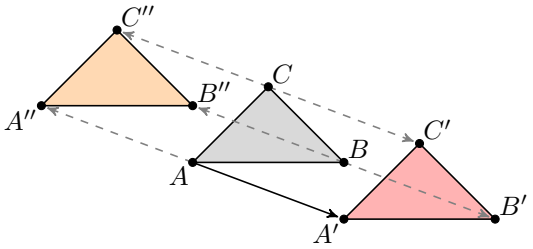
```
\tkzDefPointsBy[opções](lista de pontos){lista de pontos},
```
(5.53)

onde devemos especificar entre parênteses uma lista de pontos separados por vírgulas que sofrerão translação, enquanto as respectivas referências das imagens devem ser escritas também separadas por vírgulas, mas entre chaves. No espaço de opções, escrevemos `translation = from A to B`, sendo `A` e `B` as extremidades do segmento representante do vetor.

Devemos ter em mente que toda translação é totalmente dependente do segmento considerado e que alterações nas coordenadas ou no sentido de suas extremidades acarretarão resultados diferentes. Portanto, o segmento escolhido deve ser adequado para o que se pretende. Na Figura 5.53, construída por meio do código a seguir, os triângulos $A'B'C'$ e $A''B''C''$ são os respectivos resultados para as translações, segundo os segmentos AA' e $A'A$.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/3.5/A, 2/3.5/B, 1/4.5/C, 2/2.75/A'}
    \tkzDefPointsBy[translation=from A to A'](B,C){}
    \tkzDefPointsBy[translation=from A' to A](A,B,C){A'',B'',C''}
    \tkzDrawPolygon[fill=gray!30](A,B,C)
    \tkzDrawPolygon[fill=red!30](A',B',C')
    \tkzDrawPolygon[fill=orange!30](A'',B'',C'')
    \tkzDrawSegments[->,>=stealth',shorten >=0.8mm](A,A')
    \tkzDrawSegments[dashed,->,>=stealth',shorten >=0.8mm,gray](%
    B,B' C,C' A,A'' B,B'' C,C'')
    \tkzDrawPoints(A,B,C,A',B',C',A'',B'',C'')
    \tkzLabelPoints[above right=-1mm](B,B',B'',C,C',C'')
    \tkzLabelPoints[below left=-1mm](A,A',A'')
  \end{tikzpicture}
\end{document}
```

Figura 5.53: Translação de pontos.



Fonte: Elaborada pelo autor com base em Matthes [29].

Observe que na linha 6 do código da Figura 5.53, o interior das chaves está vazio. Isso foi feito de propósito para que a macro referencie automaticamente as imagens dos pontos B e C com B' e C'. Isso é sempre possível, mas deve ser utilizado com cautela para não correr o risco de referenciar pontos com o mesmo caractere.

Podemos não querer que a translação seja efetuada considerando o comprimento do segmento representante do vetor em sua totalidade, isto é, podemos querer realizar translações onde o vetor considerado seja múltiplo do representado pelo segmento. Sendo assim, antes de utilizar o comando (5.53), devemos fazer uso da macro

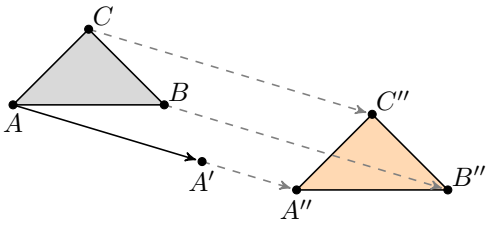
$$\texttt{\textbackslash tkzDefPointWith[opções](ponto}_1\texttt{|,ponto}_2\texttt{)}, \tag{5.54}$$

onde colocamos entre parênteses as referências dos pontos das extremidades do segmento representante do vetor, tal como na macro (5.53). No espaço de opções, colocamos, separados por vírgula, `linear` e `K="valor"`. O uso dessas opções criará um ponto sobre a reta suporte do vetor, de forma que o vetor representado pelo primeiro ponto especificado na macro (5.54) e o ponto criado, nessa ordem, será um múltiplo do vetor original conforme o valor de `K`. Por exemplo, suponha que foi digitado em um código `\tkzDefPointWith[linear,K=2](A,B)` e que o ponto criado tenha sido referenciado, por meio da macro (5.29), como C. Então, esse ponto seria tal que $\overrightarrow{AC} = 2 \times \overrightarrow{AB}$.

Após usar a macro (5.54) e referenciar o ponto criado pelo mesmo, utilizamos o comando (5.53) para determinar as imagens dos pontos desejados por translação, considerando o novo vetor. Observe um exemplo na Figura 5.54, a qual é resultado do código a seguir.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/4/A, 2/4/B, 1/5/C, 2.5/3.25/A'}
    \tkzDefPointWith[linear,K=1.5](A,A')\tkzGetPoint{A''}
    \tkzDefPointsBy[translation=from A to A''](B,C){B'',C''}
    \tkzDrawPolygon[fill=gray!30](A,B,C)
    \tkzDrawPolygon[fill=orange!30](A'',B'',C'')
    \tkzDrawSegment[->,>=stealth',shorten >=0.8mm](A,A')
    \tkzDrawSegments[gray,dashed,->,>=stealth',shorten >=0.8mm](%
    A',A'' B,B'' C,C'')
    \tkzDrawPoints(A,A',B,C,A'',B'',C'')
    \tkzLabelPoints[below](A,A',A'')
    \tkzLabelPoints[above right=-0.1cm](C,C'',B,B'')
  \end{tikzpicture}
\end{document}
```

Figura 5.54: Translação segundo um múltiplo de um vetor.



Fonte: Elaborada pelo autor com base em Matthes [29].

A macro (5.54) desempenha um papel semelhante à macro (5.47) utilizada para determinar pontos sobre uma reta. No entanto, a macro (5.54) possui outras opções de uso, como, por exemplo, ser utilizado para obter retas paralelas e perpendiculares. O leitor interessado pode encontrar uma apresentação dessas funções em Matthes [29]. Além disso, a macro (5.54), como foi apresentada aqui, desempenha papel crucial para a criação de animações como as que serão abordadas na Seção 5.3.

Obtendo pontos por rotação

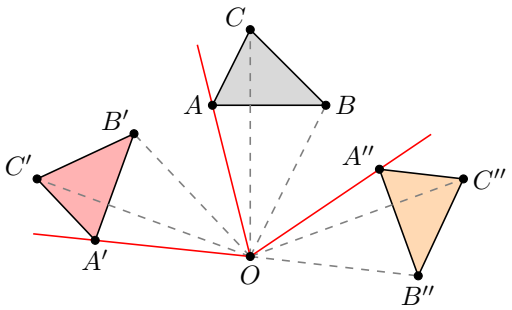
A rotação também é uma isometria, mas determina imagens de pontos sobre o plano segundo um ângulo de rotação considerando um centro referencial. Portanto, para determinarmos essa transformação no `tkz-euclide`, precisamos de um ponto que será o centro de rotação, de uma medida de ângulo em graus e de pontos para aplicar a rotação.

As imagens dos pontos aplicados nessa transformação são obtidas utilizando a mesma macro para translação (5.53), porém nas opções devemos escrever `rotation = center O angle "valor"` onde especificamos em `O` o ponto central da rotação e em `"valor"` a medida do ângulo. Para todo ponto colocado entre parênteses, será determinada uma imagem por rotação, a qual deverá ser referenciada entre chaves, de forma respectiva e, se houver mais de uma, separadas por vírgula. Veja exemplos na Figura 5.55, construída conforme a sequência de macros a seguir.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{-2/3/O, -2.5/5/A, -1/5/B, -2/6/C}
    \tkzDefPointsBy[rotation=center O angle 70](A,B,C){}
    \tkzDefPointsBy[rotation=center O angle -70](A,B,C){A'',B'',C''}
    \tkzDrawPolygon[fill=gray!30](A,B,C)
    \tkzDrawPolygon[fill=red!30](A',B',C')
```

```
    \tkzDrawPolygon[fill=orange!30](A'',B'',C'')
    \tkzDrawSegments[gray,dashed](O,B O,C O,B' O,C' O,B'' O,C'')
    \tkzDrawSegments[red,add=0 and 0.4](O,A O,A' O,A'')
    \tkzDrawPoints(O,A,B,C,A',B',C',A'',B'',C'')
    \tkzLabelPoints[above left=-1mm](C,B',C',A'')
    \tkzLabelPoints[left](A)
    \tkzLabelPoints[below](O,A',B'')
    \tkzLabelPoints[right](B,C'')
  \end{tikzpicture}
\end{document}
```

Figura 5.55: Rotação de um triângulo.



Fonte: Elaborada pelo autor com base em Matthes [29].

Observe que para valor positivo do ângulo informado na macro, o sentido de rotação é o anti-horário, e para valor negativo, é horário.

O uso dessas opções possibilitará animar figuras tendo o valor do ângulo de rotação como controle da animação, como veremos na Seção 5.3.

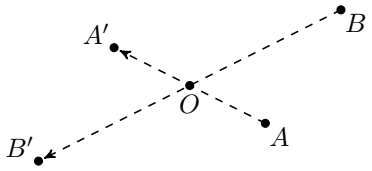
Obtendo pontos por simetria em relação a um ponto

Outra isometria muito importante para a criação de pontos e desenho de figuras é a simetria. Como sabemos do Capítulo 3, uma simetria em relação a um ponto é uma transformação geométrica que determina os pontos simétricos em relação ao ponto dado. Portanto, para determinarmos as imagens de pontos por simetria com o `tkz-euclide`, devemos ter um ponto previamente definido além dos pontos dos quais se deseja obter seus simétricos.

Assim como nas transformações anteriores, a obtenção de pontos por simetria é dada pela macro (5.53), porém com a utilização da opção `symmetry=center O`, sendo `O` a referência do ponto central da simetria. Observe na Figura 5.56 a obtenção dos simétricos A' e B' aos pontos A e B em relação a O, conforme as macros a seguir.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{-2/4/O, -1/3.5/A, 0/5/B}
    \tkzDefPointsBy[symmetry = center O](A,B){}
    \tkzDrawSegments[dashed,->,>=stealth',shorten >=0.8mm](A,A' B,B')
    \tkzDrawPoints(O,A,A',B,B')
    \tkzLabelPoints[below right=-0.1cm](A,B)
    \tkzLabelPoints[above left=-0.1cm](A',B')
    \tkzLabelPoints[below](O)
  \end{tikzpicture}
\end{document}
```

Figura 5.56: Determinação de pontos simétricos.



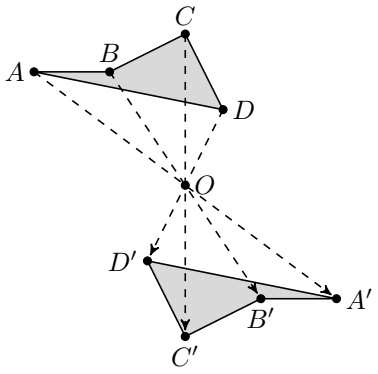
Fonte: Elaborada pelo autor com base em Matthes [29].

Também é possível obter figuras simétricas em relação a um ponto. Para isso, basta criar os pontos simétricos aos que determinam a figura. Veja um exemplo na Figura 5.57, a qual foi obtida por meio do seguinte código:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/4/O, -2/5.5/A, -1/5.5/B, 0/6/C, 0.5/5/D}
    \tkzDefPointsBy[symmetry = center O](A,B,C,D){}
    \tkzDrawPolygon[fill=gray!30](A,B,C,D)
    \tkzDrawPolygon[fill=gray!30](A',B',C',D')
    \tkzDrawSegments[dashed,->,>=stealth',shorten >=0.8mm](A,A'
    B,B' C,C' D,D')
    \tkzDrawPoints(O,A,A',B,B',C,C',D,D')
    \tkzLabelPoints[above](B,C)
    \tkzLabelPoints[below](C',B')
```

```
    \tkzLabelPoints[left](A,D')
    \tkzLabelPoints[right](A',O,D)
  \end{tikzpicture}
\end{document}
```

Figura 5.57: Obtendo figuras simétricas em relação a um ponto.



Fonte: Elaborada pelo autor.

Obtendo pontos por reflexão em relação a uma reta

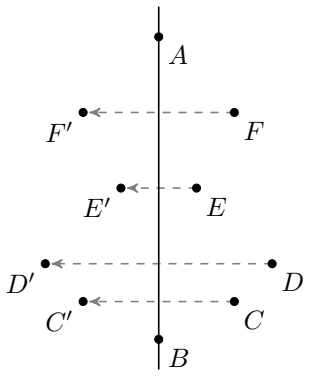
A reflexão em torno de um reta é uma transformação isométrica que associa a cada ponto do plano, seu simétrico em relação à reta dada. Portanto, é necessário que se tenha uma reta predefinida para que possamos aplicar a reflexão.

Para obtermos as imagens de pontos por meio de reflexões, utilizamos mais uma vez a macro (5.53), porém com as opções `reflection = over A--B`, sendo `A` e `B` pontos sobre a reta que será considerada na transformação. Observe na Figura 5.58, construída com o código a seguir, a obtenção dos pontos C', D', E' e F' por meio da reflexão dos pontos C, D, E e F em torno da reta $\overleftrightarrow{AB}$.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/6/A, 0/2/B, 1/2.5/C, 1.5/3/D, 0.5/4/E, 1/5/F}
    \tkzDefPointsBy[reflection = over A--B](C,D,E,F){}
    \tkzDrawSegments[gray,dashed,->,>=stealth',shorten >=0.8mm](
    C,C' D,D' E,E' F,F')
    \tkzDrawPoints(A,B,C,D,E,F,C',D',E',F')
    \tkzDrawLine[add=0.1 and 0.1](A,B)
    \tkzLabelPoints(A,B,C,D,E,F)
```

```
    \tkzLabelPoints[below left](C',D',E',F')
  \end{tikzpicture}
\end{document}
```

Figura 5.58: Aplicação de reflexão.



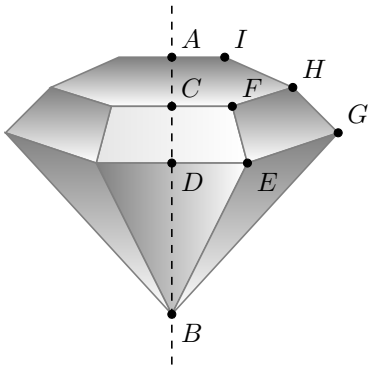
Fonte: Elaborada pelo autor.

A utilização de reflexões é muito útil para desenhar figuras que gozam de simetrias em relação a uma reta. Veja, na Figura 5.59, um exemplo da utilização dessa transformação na construção dada. Tal figura foi construída por meio do seguinte código:

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
  \begin{tikzpicture}
    \tkzDefPoints{0/5.4/A,0/2/B,0/4.75/C,0/4/D,1/4/E,0.8/4.75/F,
    2.2/4.4/G,1.6/5/H,0.7/5.4/I}
    \tkzDefPointsBy[reflection = over A--B](E,F,G,H,I){}
    \tkzDrawPolygon[draw=gray,top color=gray](H,I,I',H',F',F)
    \tkzDrawPolygon[draw=gray,left color=gray!20](E,F,F',E')
    \tkzDrawPolygon[draw=gray,top color=gray](G,H,F,E)
    \tkzDrawPolygon[draw=gray,top color=gray](G',H',F',E')
    \tkzDrawPolygon[draw=gray,left color=gray](B,E,E')
    \tkzDrawPolygon[draw=gray,top color=gray](B,G,E)
    \tkzDrawPolygon[draw=gray,top color=gray](B,G',E')
    \tkzDrawLine[dashed](A,B)
    \tkzDrawPoints(A,B,C,D,E,F,G,H,I)
    \tkzLabelPoints[above right](A,C,F,G,H,I)
    \tkzLabelPoints[below right](B,D,E)
```

```
  \end{tikzpicture}
\end{document}
```

Figura 5.59: Criando figura utilizando reflexão.



Fonte: Elaborada pelo autor.

5.3 Animações no LaTeX

Até o momento aprendemos como desenhar as figuras geométricas euclidianas por meio do pacote `tkz-euclide` e `TikZ`. Nessa seção, exploraremos macros do pacote `animate` com o objetivo de animar figuras geométricas por meio de algumas das transformações apresentadas anteriormente. Tomaremos como principal referência o manual do pacote `animate`, disponível em Grahn [20].

5.3.1 Preparando o ambiente de animação

De acordo com Grahn [20], as animações produzidas pelo `animate` podem ser criadas utilizando o ambiente `animateinline` ou a macro `\animategraphics`. Em ambos os casos, a animação é feita por meio de sucessões de imagens que, adequadamente ordenadas, se sobrepõem e simulam o movimento. Esse tipo de animação é chamado de animação "quadro a quadro". A diferença entre essas duas maneiras de animar é que a macro `\animategraphics` pode ser usada para montar animações de conjuntos de arquivos gráficos ou PDF de várias páginas, enquanto o ambiente `animateinline` destina-se a criar a animação a partir do material tipográfico que ele inclui. Este material pode ser imagens desenhadas no ambiente LaTeX 'picture' ou usando os recursos avançados do PSTricks ou pgf/TikZ; até mesmo material textual comum pode ser animado dessa maneira.

Portanto, a utilização da macro `\animategraphics` exige que as imagens já estejam produzidas em um arquivo composto por várias páginas ou de forma separada e que, não necessariamente, tenham sido produzidas com o LaTeX, enquanto para animar com o ambiente `animateinline`, precisamos desenhar as imagens dentro do corpo do ambiente de animação que, no nosso caso, serão produzidas com os pacotes `TikZ` e `tkz-euclide`.

A configuração completa da macro `\animategraphics` é

```
\animategraphics[opções]{frame rate}{file basename}{first}{last}.
```
(5.55)

Para essa macro funcionar, devemos incluir o pacote `animate` no preâmbulo, bem como, determinar a classe de documento, que mais uma vez adotaremos `standalone` por oferecer mais liberdade de espaço. No entanto, pode-se escolher outra classe e incluir outros pacotes, conforme a necessidade. Também é necessário carregar o pacote `graphicx` visto que trabalharemos com imagens. A estrutura do código seguirá a seguinte organização:

```
\documentclass{standalone}
\usepackage{animate}
\usepackage{graphicx}
\begin{document}
  \animategraphics[opções]{frame rate}{file basename}{first}{last}
\end{document}
```

Caso se faça uso de outra classe de documento, é aconselhável que se posicione a macro (5.55) dentro de um ambiente `figure`, já que o mesmo faz uso de imagens. A seguir, vamos explicar como utilizar essa macro.

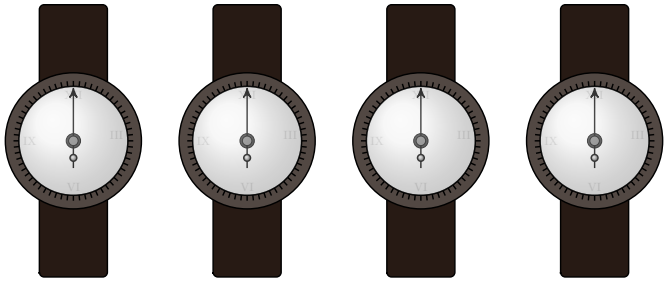
Como dito anteriormente, a animação causa a impressão de movimento por meio da sobreposição de imagens que possuem os elementos que se pretende animar em posições diferentes, simulando movimento. A cada uma dessas imagens damos o nome de *frame* ou quadro. A velocidade com que esses *frames* se sobrepõem é determinada por um número, geralmente natural, especificado no primeiro par de chaves designado por `frame rate` que em tradução livre significa "taxa de quadros". A unidade considerada no `frame rate` é o *fps* (do inglês *frames per second*), isto é, o número especificado nessa opção indicará quantos *frames* passará em uma unidade de tempo de um segundo.

O número de *fps* é muito importante para a criação da animação. Valores pequenos podem gerar animações que dão saltos bruscos, prejudicando a fluidez, enquanto valores muito altos, tornarão a animação muito rápida, prejudicando a visualização

em sua totalidade. Além disso, de acordo com Almeida [31], animações quadro a quadro devem evitar altos valores de *fps*, pois a produção da animação se torna muito lenta, visto que será necessária a criação de muitos quadros. Ainda de acordo com Almeida [31], na criação de animações costuma-se utilizar de 12 a 24 *fps*.

A Figura 5.60 mostra uma animação produzida com uso do `frame rate`. Em cada relógio que aparece na figura, foram utilizados 60 *frames*, com quantidades de *fps* distintas. As quantidades de *fps* utilizadas, olhando da esquerda para a direita, foram 1, 12, 24 e 60.

Figura 5.60: Relógios exemplificando o uso de 1, 12, 24 e 60 *fps*.



Fonte: Elaborada pelo autor.

A quantidade de quadros também é importante. Eles podem ser produzidos em imagens separadas, isto é, um arquivo para cada imagem, ou em um único arquivo com várias páginas, sendo que cada uma delas será considerada como um *frame*. Se as imagens forem produzidas em um arquivo único, deve-se digitar, na macro (5.55), o nome do arquivo no segundo par de chaves (no lugar do texto `file basename`). Tal arquivo deve estar salvo na mesma pasta do arquivo TEX da animação. No terceiro par de chaves, digita-se o número do quadro por onde se pretende iniciar a animação (no lugar de `first`) e no quarto e último par de chaves, digita-se o número do último quadro onde a animação terminará (no lugar de `last`). De acordo com Grahn [20], a primeira página do arquivo é equivalente ao quadro de número zero, ou seja, em um arquivo de n páginas, a primeira equivale ao *frame* de número zero e a última página equivale ao de número $n - 1$. Caso queira, é possível deixar os dois últimos pares de chaves em branco. Nesse caso, o `animate` iniciará a animação pela primeira página e terminará na última página do arquivo.

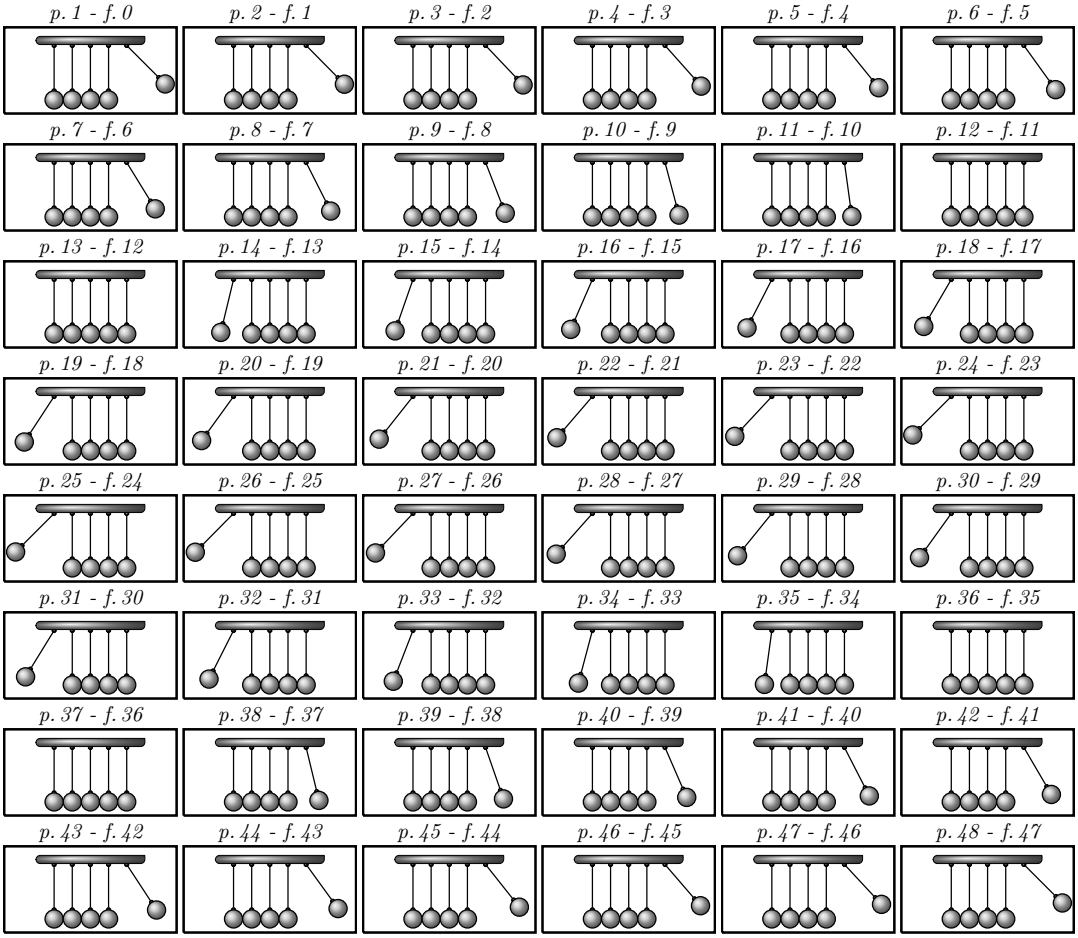
A Figura 5.62 é um exemplo de animação produzida por meio de um arquivo único. A mesma foi produzida com as imagens do arquivo de nome *PenduloDeNewton*, representadas na Figura 5.61, com o código

```
\documentclass{standalone}
\usepackage{animate}
```

```
\usepackage{graphicx}
\begin{document}
  \animategraphics[autoplay,loop]{30}{PenduloDeNewton}{0}{47}
\end{document}.
```

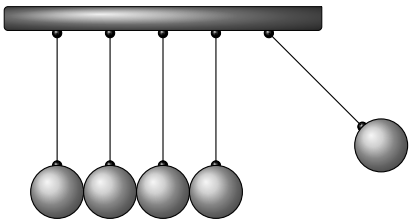
Esse código indica que a animação rodará a uma velocidade de 30 *fps*, iniciará pela primeira página (quadro 0 ou página 1) e terminará na última (quadro 47 ou página 48). Ao chegar no último quadro, a sequência de sobreposição das imagens reinicia.

Figura 5.61: Páginas do arquivo *PenduloDeNewton*.



Fonte: Elaborada pelo autor.

Figura 5.62: Pêndulo de Newton.



Fonte: Elaborada pelo autor.

Para animar utilizando arquivos separados, cada um deles deve possuir uma única página e todos devem ser nomeados de forma a possuírem a primeira parte do nome comum a todos. Após a parte comum coloca-se o símbolo de *underline* seguido de um número que indicará a posição do arquivo na sequência de sobreposição. Cada um dos arquivos representa um *frame* e serão exibidos na ordem como foram nomeados. Por exemplo, em uma sequência com n arquivos, os mesmos podem ser nomeados, respectivamente, com *frame_ 1*, *frame_ 2*, *frame_ 3*, *frame_ 4*, ..., *frame_ - n*, e a animação seguirá os números especificados nos nomes dos arquivos.

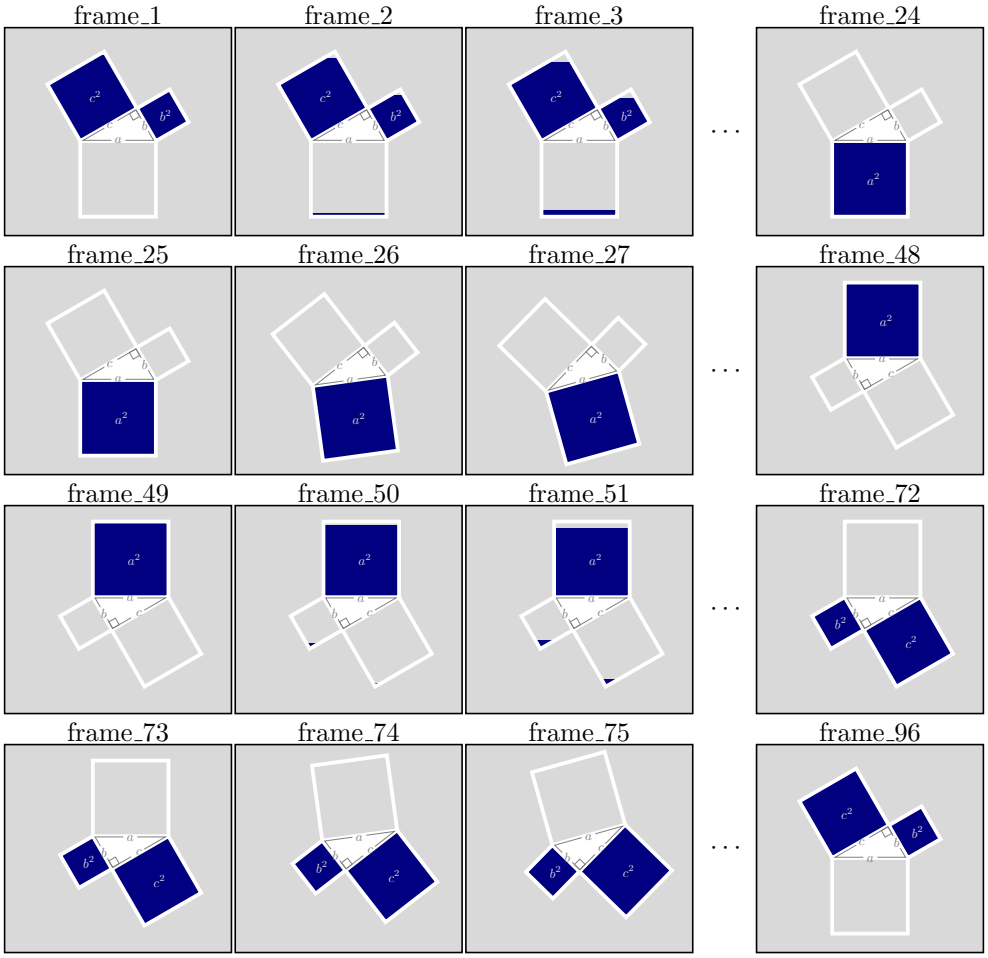
Diferente da animação com arquivo único, quando utilizamos arquivos separados, no segundo par de chaves da macro (5.55) digitamos apenas a parte comum dos nomes dos arquivos, seguido do símbolo de *underline* e nos dois últimos pares de chaves digitamos, respectivamente, o número do primeiro e do último arquivo por onde iniciará e terminará a animação.

Como exemplo de animação produzida com arquivos separados, temos a Figura 5.64, a qual foi produzida a partir dos arquivos da Figura 5.63, nomeados conforme os textos situados acima de cada uma. O código utilizado foi

```
\documentclass{standalone}
\usepackage{animate}
\usepackage{graphicx}
\begin{document}
  \animategraphics[autoplay,loop]{12}{frame_}{1}{96}
\end{document},
```

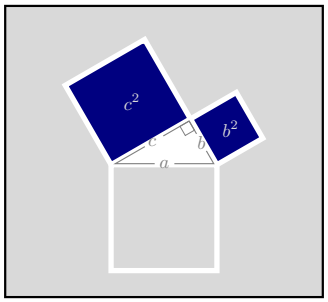
que usa os 96 arquivos da Figura 5.63, funcionando a uma velocidade de 12 *fps*, e estabelece uma sequência de sobreposição do arquivo `frame_1` até o arquivo `frame_96`. Ao chegar no último arquivo, a sequência reinicia, formando um ciclo infinito.

Figura 5.63: Frames Teorema de Pitágoras.



Fonte: Elaborada pelo autor com base em Warehouse [32].

Figura 5.64: Teorema de Pitágoras.

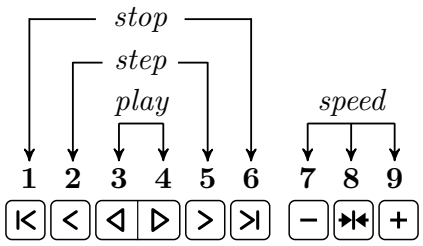


Fonte: Elaborada pelo autor com base em Warehouse [32].

Em relação às opções da macro (5.55), podemos utilizá-las para adicionar botões que controlam a reprodução da animação. Para isso, basta digitarmos `controls` ou `controls={all}` no espaço reservado para as opções. Com isso, serão adicionados

automaticamente nove botões abaixo da região de animação, conforme a Figura 5.65.

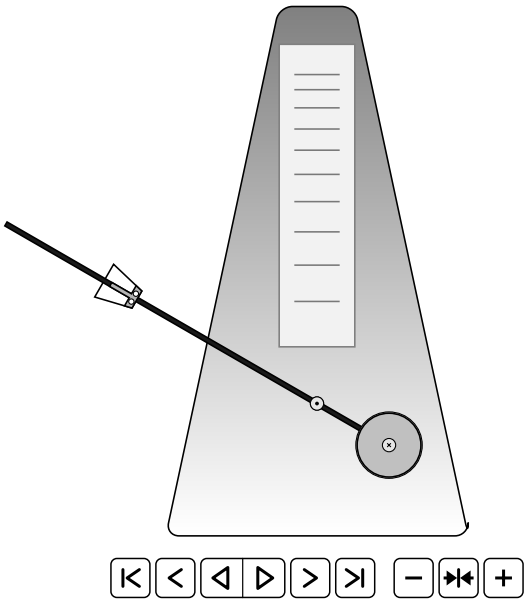
Figura 5.65: Botões do `animate`.



Fonte: Elaborada pelo autor com base em Grahn [20].

Segundo Grahn [20], o significado de cada botão é, da esquerda para a direita: parar e ir para o primeiro quadro, voltar um quadro, reproduzir para trás, reproduzir para frente, ir um quadro à frente, parar e ir para o último quadro, diminuir a velocidade, reproduzir em velocidade padrão e aumentar a velocidade. Ambos os botões "play" são substituídos por um grande botão "pause" enquanto a animação está sendo reproduzida. Sugerimos ao leitor interagir com os botões da animação da Figura 5.66, clique em um de cada vez, verificando o que ocorre com a animação.

Figura 5.66: Metrônomo animado.



Fonte: Elaborada pelo autor com base em Grahn [20].

Ainda de acordo com Grahn [20], é possível escolher grupos de botões dentre os nove disponíveis. Para isso, escrevemos `controls={}` no espaço de opções e

escolhemos os grupos digitando `stop`, `step`, `play` ou `speed`, dentro das chaves. Digitando `stop`, serão exibidos os botões 1 e 6, digitando `step`, apenas os botões 2 e 5 são exibidos, escrevendo `play`, os botões 3 e 4 são exibidos e `speed` exibe o trio de botões 7, 8 e 9. É possível ainda fazer combinações com esses grupos, basta escrever o texto que os controla separados por vírgulas.

Existem outras opções para a macro (5.55) que modificam as animações, como `autoplay` e `loop` que fazem, respectivamente, a animação reproduzir automaticamente e reiniciar em um ciclo infinito. Note que essas opções foram utilizadas nas animações das Figuras 5.62 e 5.64, pois elas iniciam automaticamente e repetem infinitamente. A Tabela 5.7 traz essas e outras opções que julgamos úteis para a macro (5.55). Para obter todas as opções, consulte Grahn [20].

Tabela 5.7: Resumo das opções para a macro (5.55).

\animategraphics[opções]{frame rate}{file basename}{first}{last}		
Opção	**Função**	**Exemplo**
poster= "número"	Define qual quadro deve ser exposto quando a animação não estiver em reprodução.	*poster=6*
autoplay	Reproduz automaticamente a animação assim que se abre a página em que ela está.	*autoplay*
loop	Faz a animação reiniciar automaticamente ao chegar ao final, produzindo um ciclo infinito.	*loop*
palindrome	Faz a animação ir para frente e para traz continuamente.	*palindrome*
controls	Adiciona botões para controle da animação.	*controls={all}* *controls={stop,play}*
controlsaligned="posição"	Controla o posicionamento dos botões de controle da animação, se aparecerão à esquerda, no centro ou à direita.	*controlsaligned=left*
buttonsize="tamanho"	Controla o tamanho dos botões de controle das animações.	*buttonsize=0.8cm*
nomouse	Faz a área de animação não responder aos comandos do mouse, os botões responderão.	*nomouse*

Fonte: Elaborada pelo autor com base em Grahn [20].

5.3.2 Animando com o ambiente `animateinline`.

Como dito na Subseção 5.3.1, o ambiente `animateinline` é utilizado para produzir animações onde as imagens são criadas dentro do corpo do documento TEX. Organizaremos este ambiente da seguinte maneira:

```
\begin{animateinline}[opções]{frame rate}
  \multiframe{nº de frames}{variáveis}{
    \begin{tikzpicture}
      \path[use as bounding box] (Xmax,Ymax) rectangle (Xmin,Ymin);
      %Comandos para a produção dos desenhos.
    \end{tikzpicture}}
\end{animateinline}
```

As opções do `animateinline` são as mesmas apresentadas na Tabela 5.7 e no interior do par de chaves, após o espaço de opções, colocamos o número de *fps* com que a animação funcionará.

Na segunda linha escrevemos `\multiframe`, seguido de três pares de chaves. No primeiro par, especificamos o número de quadros que serão construídos. O segundo par de chaves é reservado para a declaração das variáveis que controlarão a construção dos frames e, consequentemente, a animação. Essas variáveis podem assumir valores inteiros ou reais, sendo que, de acordo com Grahn [20], a primeira letra da variável define a qual conjunto ela pertence, "i" ou "I" são reservados para variáveis inteiras e "n", "N", "r" ou "R" são reservadas para valores reais. É aconselhável que se usem variáveis reais para evitar erros de compilação. O controle dos valores das variáveis é feito por meio de um número inicial somado a um incremento. Por exemplo, uma variável real pode ser declarada do seguinte modo: $Ra = 0 + 1/10$. Isso significa que a variável receberá os valores

$$0,\ 1/10,\ 2/10,\ 3/10,\ 4/10,\ ...,\ (x-1)/10,$$

sendo x o número de quadros especificado no primeiro par de chaves, e para cada valor assumido pela variável, será produzido um *frame*.

Dentro do terceiro par de chaves, escrevemos o ambiente `tikzpicture` que possuirá os comandos para a construção das figuras. Observe que dentro desse ambiente escrevemos o comando

```
\path[use as bounding box] (Xmax,Ymax) rectangle (Xmin,Ymin);.
```

Tal comando tem a função de determinar uma caixa delimitadora em forma de retângulo que possui diagonal determinada pelos pontos de coordenadas `(Xman,Ymax)`

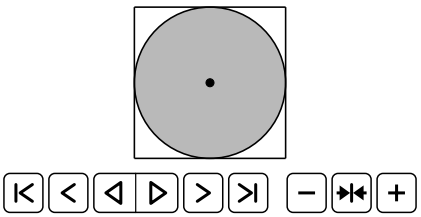
e `(Xmin,Ymin)`, os quais devem ser escolhidos de forma a posicionar todas as imagens desenhadas no interior do retângulo. A não utilização dessa caixa delimitadora distorce as imagens utilizadas na animação, a sua utilização evita tal problema.

Embora as variáveis sejam declaradas sem a barra invertida, quando as utilizarmos nas macros, dentro do ambiente `tikzpicture`, devemos fazer uso da barra e essas variáveis podem ser usadas nas mais diversas situações: em coordenadas de pontos, no controle da cor do interior de um polígono (porcentagem da cor), na espessura de uma linha, na medida do raio de um círculo, enfim, em toda situação em que o elemento substituído pela variável seja de cunho numérico.

Para exemplificar, observe o código completo a seguir e o resultado da animação na Figura 5.67.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\usepackage{animate}
\begin{document}
  \begin{animateinline}[poster=last, controls={all}]{12}
  \multiframe{12}{Ra=0+1/11, Rb=0+5}{
    \begin{tikzpicture}
      \path[use as bounding box] (3.5,2.2) rectangle (-1.5,-0.1);
      \tkzDefPoints{0/0/O, 1/1/A}
      \tkzDefPoint(0,2*\Ra){B} \tkzDefPoint(2*\Ra,0){C}
      \tkzDefPoint(2,2*\Ra){D} \tkzDefPoint(2*\Ra,2){E}
      \tkzFillCircle[R,fill=gray!\Rb](A,\Ra cm)
      \tkzDrawSegments(O,B O,C C,D B,E)
      \tkzDrawCircle[R](A,\Ra cm) \tkzDrawPoints(A)
    \end{tikzpicture}}
  \end{animateinline}
\end{document}
```

Figura 5.67: Círculo inscrito em um quadrado.



Fonte: Elaborada pelo autor.

Tudo o que estiver dentro do último par de chaves do comando `\multiframe`, e fizer uso das variáveis declaradas, será animado de maneira simultânea. Para animar de maneira sequencial, devemos definir novos "multiframes" por meio do comando `\newframe` e, logo em seguida, escrever um novo `multiframe` com novas variáveis. Por exemplo, o ambiente `animateinline` pode ser organizado da seguinte forma:

```
\begin{animateinline}[opções]{frame rate}
  \multiframe{nº de frames}{variáveis}{
    \begin{tikzpicture}
      \path[use as bounding box] (Xmax,Ymax) rectangle (Xmin,Ymin);
      %Comandos para a produção dos desenhos.
    \end{tikzpicture}}
  \newframe
  \multiframe{nº de frames}{variáveis}{
    \begin{tikzpicture}
      \path[use as bounding box] (Xmax,Ymax) rectangle (Xmin,Ymin);
      %Comandos para a produção dos desenhos.
    \end{tikzpicture}}
\end{animateinline}
```

Também podemos utilizar `\newframe*`, com asterisco, para adicionar uma pausa quando a animação chegar no fim do `multiframe` anterior a esse comando.

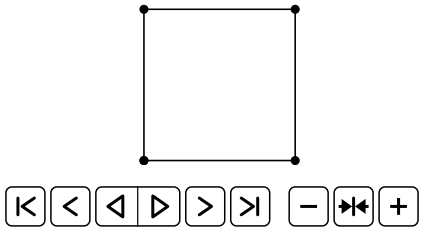
Cada `multiframe` funciona como se fosse uma cena da animação e atua de maneira independente, isto é, o que foi definido, desenhado ou nomeado no `multiframe` anterior deixa de existir no `multiframe` posterior. De acordo com Neves [33], para que o `animate` lembre dos pontos definidos nos multiframes anteriores, basta digitar `\tikzstyle{every picture}=[remember picture]` no preâmbulo do arquivo TEX. Em relação aos desenhos e nomes dos elementos, é necessário que seja desenhado novamente o que se deseja conservar de um `multiframe` para outro.

Observe a organização do código a seguir, o qual dá origem à animação da Figura 5.68, que faz uso de quatro multiframes para construir um quadrado, sendo que as variáveis são utilizadas nas coordenadas dos vértices do quadrado.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\usepackage{animate}
\tikzstyle{every picture}=[remember picture]
\begin{document}
  \begin{animateinline}[poster=last, controls={all}]{12}
  \multiframe{21}{Rb=0+0.1}{
```

```
    \begin{tikzpicture}
      \path[use as bounding box] (4,2.2) rectangle (-2.2,-0.25);
      \tkzDefPoint(0,0){A}
      \tkzDefPoint(\Rb,0){B}
      \tkzDrawPoints(A,B)
      \tkzDrawSegment(A,B)
    \end{tikzpicture}}
  \newframe
  \multiframe{21}{Rc=0+0.1}{
    \begin{tikzpicture}
      \path[use as bounding box] (4,2.2) rectangle (-2.2,-0.25);
      \tkzDefPoint(2,\Rc){C}
      \tkzDrawPoints(A,B,C)
      \tkzDrawSegments(A,B B,C)
    \end{tikzpicture}}
  \newframe
  \multiframe{21}{Rd=0+0.1}{
    \begin{tikzpicture}
      \path[use as bounding box] (4,2.2) rectangle (-2.2,-0.25);
      \tkzDefPoint(2-\Rd,2){D}
      \tkzDrawPoints(A,B,C,D)
      \tkzDrawSegments(A,B B,C C,D)
    \end{tikzpicture}}
  \newframe
  \multiframe{21}{Re=0+0.1}{
    \begin{tikzpicture}
      \path[use as bounding box] (4,2.2) rectangle (-2.2,-0.25);
      \tkzDefPoint(0,2-\Re){E}
      \tkzDrawPoints(A,B,C,D,E)
      \tkzDrawSegments(A,B B,C C,D D,E)
    \end{tikzpicture}}
  \end{animateinline}
\end{document}
```

Figura 5.68: Utilizando as variáveis em coordenadas.



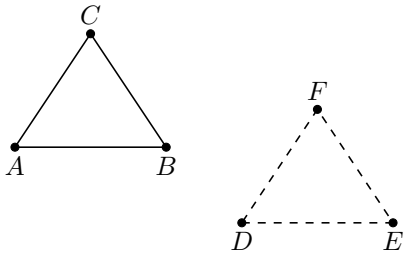
Fonte: Elaborada pelo autor.

Como vemos, é possível animar utilizando as variáveis nas coordenadas dos pontos fazendo-os se mover. Porém, mover uma grande quantidade de pontos pode ser um problema, pois devemos definir individualmente suas coordenadas, e isso, além de ser trabalhoso, pode aumentar a quantidade de linhas do código e, consequentemente, o tempo de compilação. Como alternativa, podemos utilizar as macros que criam os pontos por translação ou rotação (vistos na Subseção 5.2.10) que movem os vários pontos de uma só vez. É claro que a utilização dessas macros depende da intenção, isto é, translações são indicadas quando se pretende mover pontos em linha reta e rotações quando se pretende rotacioná-los em torno de um eixo. Nas subseções a seguir explicaremos como utilizar essas transformações geométricas para produzir animações.

5.3.3 Animando por Translação

Vamos explicar a utilização de translações para produção de animações por meio de um exemplo. Suponha que se queira transladar, por meio do ambiente `animateinline`, um triângulo ABC para a posição do triângulo DEF indicados na Figura 5.69.

Figura 5.69: Translação de um triângulo.



Fonte: Elaborada pelo autor.

A translação será realizado por meio do comando

```
\tkzDefPointWith[linear,K=\Ra](A,D)\tkzGetPoint{A'}.
```

Esse comando cria um ponto A' que será determinado seguindo múltiplos do vetor representado pelo segmento orientado AD, sendo que esses múltiplos são determinados pela variável declarada no `multiframe` como $Ra = 0+0.1$. Cada valor assumido pela variável determinará um quadro da animação, onde o ponto A' estará posicionado gradativamente mais próximo do ponto D e sempre sobre a semirreta $\overrightarrow{AD}$. Quando Ra assumir valor 0 o ponto A' coincidirá com o ponto A e quando Ra assumir valor 1, A' estará sobre o ponto D. Portanto a quantidade de *frames* a serem especificados no `multiframe` deve ser igual a 11. Essa quantidade de quadros garante o deslocamento do ponto A' da posição de A até D, pois os valores assumidos pela variável serão 0, 0.1, 0.2, 0.3, ..., 0.9 e 1.

O procedimento apresentado, apenas translada o vértice A do triângulo ABC. Para transladar os outros dois vértices, devemos fazer uso do comando

```
\tkzDefPointsBy[translation=from A to A'](B,C){}.
```

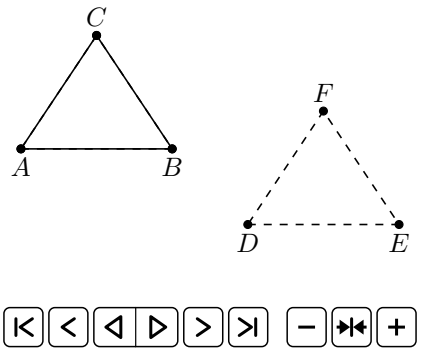
Esse comando cria os pontos B' e C' que são, respectivamente, imagens dos pontos B e C por translação segundo o vetor representado pelo segmento orientado de extremidades A e A'. Como A' é deslocado de maneira gradativa, pois depende da variável Ra, os pontos B' e C' também se deslocarão de forma gradativa, em direção, respectivamente, aos pontos E e F.

Após a criação dos pontos A', B' e C', basta desenhar o triângulo $A'B'C'$, que veremos o deslocamento do triângulo ABC em direção ao triângulo DEF. Veja o código completo a seguir e o resultado na animação da Figura 5.70.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\usepackage{animate}
\begin{document}
  \begin{animateinline}[poster=first, controls]{12}
  \multiframe{11}{Ra=0+0.1}{
    \begin{tikzpicture}
      \path[use as bounding box] (6,2) rectangle (-1,-2);
      \tkzDefPoints{0/0/A,2/0/B,1/1.5/C,3/-1/D,5/-1/E,4/0.5/F}
      \tkzDrawPolygon[dashed](A,B,C)
      \tkzLabelPoints[below](A,B,D,E)
      \tkzLabelPoints[above](C,F)
```

```
      \tkzDefPointWith[linear,K=\Ra](A,D)\tkzGetPoint{A'}
      \tkzDefPointsBy[translation=from A to A'](B,C){}
      \tkzDrawPolygon(A',B',C')
      \tkzDrawPolygon[dashed](D,E,F)
      \tkzDrawPoints(A,B,C,D,E,F,A',B',C')
    \end{tikzpicture}}
  \end{animateinline}
\end{document}
```

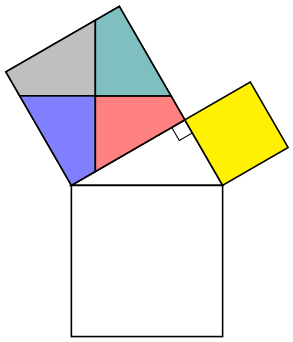
Figura 5.70: Animação translação de um triângulo.



Fonte: Elaborada pelo autor.

Esse procedimento pode ser usado para criar animações mais complexas, como, por exemplo, a apresentada na Figura 5.71, que ilustra uma interpretação do Teorema de Pitágoras.

Figura 5.71: Uso de translação no Teorema de Pitágoras.



Fonte: Elaborada pelo autor com base em Warehouse [32].

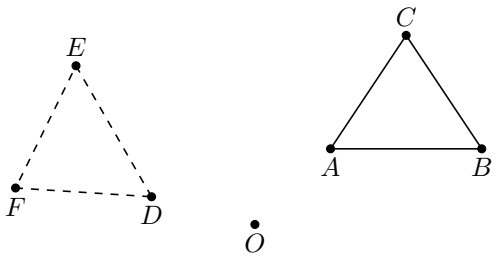
Nessa animação, o movimento de cada polígono foi realizado com um `multiframe`, fazendo com que eles se movam de maneira sequencial, isto é, o próximo polígono só

se moverá após o fim do movimento do anterior, quando um `multiframe` termina e inicia outro. O código completo da animação da Figura 5.71 está disponível na Seção A.2 do Apêndice A.

5.3.4 Animação por Rotação

Veremos agora como utilizar rotações para produzir animações através de um exemplo. Considere o triângulo ABC e o ponto O da Figura 5.72 e suponha que se pretende produzir uma animação que rotacione o triângulo ABC em um ângulo de 120° em torno do ponto O, de forma a coincidir com o triângulo DEF.

Figura 5.72: Rotação de um triângulo estático.



Fonte: Elaborada pelo autor.

Para realizar, de forma animada, o movimento do triângulo ABC por rotação, utilizamos o comando

```
tkzDefPointsBy[rotation=center O angle 120*\Ra](A,B,C){}
```

para criar os pontos A', B' e C'. A variável Ra, que deve ser definida como $Ra = 0 + 0.1$, controla o movimento dos pontos criados, conforme os valores determinados pela multiplicação de 120° pelos valores assumidos pela variável, os quais devem ir de 0 a 1, isto é, a quantidade de *frames* definida no `multiframe` deve ser igual a 11.

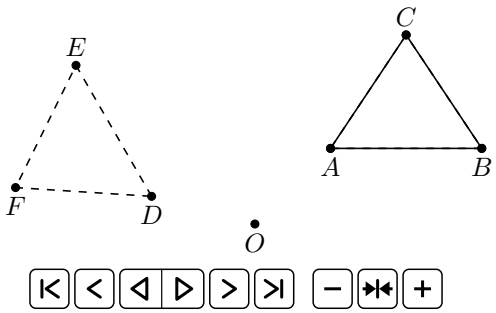
Portanto, a animação desejada será composta por 11 quadros, sendo que cada um deles representa os pontos A', B' e C' em posições diferentes, conforme as rotações dos pontos A, B e C determinadas pelos ângulos 0°, 12°, 24°, ..., 108°, 120° obtidos pelo produto `120*\Ra`.

Após a criação dos pontos, basta desenhar o triângulo $A'B'C'$ e renomeá-lo como DEF. Veja a seguir o código completo da animação e o resultado na Figura 5.73.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\usepackage{animate}
```

```
\begin{document}
  \begin{animateinline}[poster=first, controls]{12}
  \multiframe{11}{Ra=0+0.1}{
    \begin{tikzpicture}
      \path[use as bounding box] (6.5,2.25) rectangle (-1,-1.5);
      \tkzDefPoints{4/0/A, 6/0/B, 5/1.5/C, 3/-1/O}
      \tkzDrawPolygon[dashed](A,B,C)
      \tkzLabelPoints[below](A,B,O)
      \tkzLabelPoints[above](C)
      \tkzDrawPoints(A,B,C,O)
      \tkzDefPointsBy[rotation=center O angle 120](A,B,C){D,E,F}
      \tkzDrawPolygon[dashed](D,E,F)
      \tkzLabelPoints[below](D,F)
      \tkzLabelPoints[above](E)
      \tkzDrawPoints(D,E,F)
      \tkzDefPointsBy[rotation=center O angle 120*\Ra](A,B,C){}
      \tkzDrawPolygon(A',B',C')
      \tkzDrawPoints(A',B',C')
    \end{tikzpicture}}
  \end{animateinline}
\end{document}
```

Figura 5.73: Rotação de um triângulo animado.



Fonte: Elaborada pelo autor.

5.3.5 Animação com translação e rotação

Podemos ainda realizar o deslocamento de uma figura geométrica de forma que ela realize uma translação e, ao mesmo tempo, uma rotação em torno de um dos pontos transladados. Para isso, basta transladar os pontos que determinam a figura, escolher um dos pontos obtidos e rotacionar os outros em torno dele. Por exemplo,

considere os triângulos congruentes ABC e DEF da Figura 5.74 e suponha que se deseja transladar o triângulo ABC para DEF de forma que, ao mesmo tempo da translação, o triângulo sofra rotação de 180° em torno do seu vértice A.

Figura 5.74: Translação com rotação de um triângulo.

Fonte: Elaborada pelo autor.

Transladamos primeiro o vértice A para o ponto D por meio do comando

```
\tkzDefPointWith[linear,K=\Ra](A,D)\tkzGetPoint{A'}.
```
(5.56)

Em seguida, transladamos os pontos B e C por meio do comando

```
\tkzDefPointsBy[translation=from A to A'](B,C){}.
```
(5.57)

Os comandos (5.56) e (5.57) criam os pontos A', B' e C' sendo que, para realizar o movimento de rotação, devemos rotacionar B' e C' em torno do ponto A'. Essa rotação é determinada pelo comando

```
\tkzDefPointsBy[rotation=center A' angle 180*\Ra](B',C'){B'',C''}.
```
(5.58)

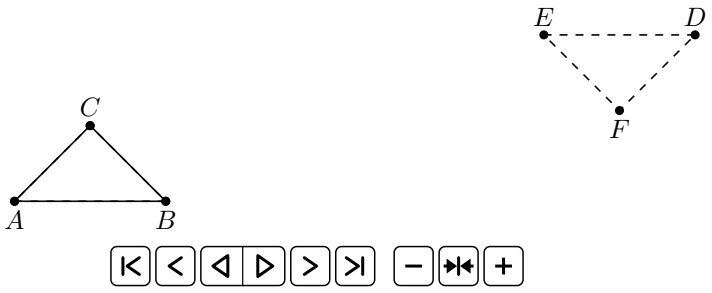
Obtemos então os pontos B'' e C'', que juntamente com A', compõem o triângulo $A'B''C''$ que se moverá de ABC até DEF, de forma a sofrer translação e rotação ao mesmo tempo.

É válido ressaltar que o número de *frames* e a variável Ra devem ser definidos, no `multiframe`, de forma que Ra assuma valores de 0 até 1 para que o ponto A' percorra todo o segmento AD, indo de A até D, e para que B'' e C'' rotacione 180° em torno de A'. Observe a seguir um código completo composto por 30 quadros cuja variável é definida como `Ra=0+1/29`. Verifique também o resultado esperado na Figura 5.75.

```
\documentclass{standalone}
\usepackage{tkz-euclide}
```

```
\usepackage{animate}
\tikzstyle{every picture}=[remember picture]
\begin{document}
  \begin{animateinline}[poster=first, controls]{12}
  \multiframe{30}{Ra=0+1/29}{
    \begin{tikzpicture}
      \path[use as bounding box] (4.5,2.25) rectangle (-6.5,-1.5);
      \tkzDefPoints{-5/-1/A,-3/-1/B,-4/0/C,4/1.2/D,2/1.2/E,
      3/0.2/F}
      \tkzDrawPolygon[dashed](A,B,C)
      \tkzDrawPolygon[dashed](D,E,F)
      \tkzDrawPoints(A,B,C,D,E,F)
      \tkzLabelPoints[below](A,B,F)
      \tkzLabelPoints[above](C,D,E)
      \tkzDefPointWith[linear,K=\Ra](A,D)\tkzGetPoint{A'}
      \tkzDefPointsBy[translation=from A to A'](B,C){}
      \tkzDefPointsBy[rotation=center A' angle 180*\Ra](B',C'){B'',C''}
      \tkzDrawPolygon(A',B'',C'')
      \tkzDrawPoints(A',B'',C'')
    \end{tikzpicture}}
  \end{animateinline}
\end{document}
```

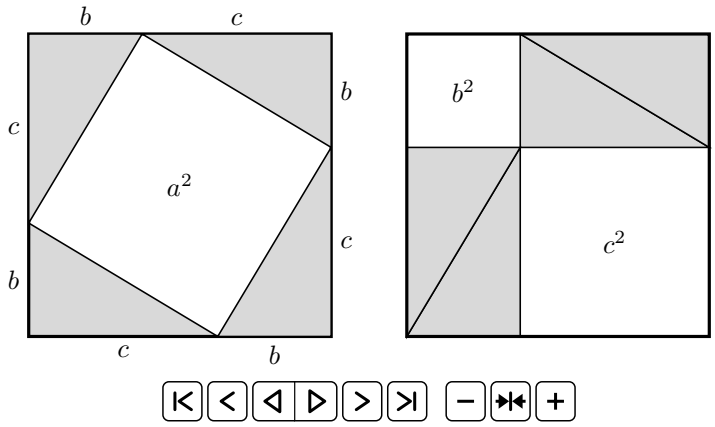
Figura 5.75: Translação com rotação de um triângulo (Animado).



Fonte: Elaborada pelo autor.

Esse procedimento pode ser utilizado para construir animações mais complexas, como na Figura 5.76 onde fizemos uso de vários multiframes para sua construção. Para mover o triângulo de lados a, b e c, dentro do quadrado do lado esquerdo, utilizamos translação com rotação, simultaneamente.

Figura 5.76: Uso de translação com rotação no Teorema de Pitágoras.



Fonte: Elaborada pelo autor.

5.3.6 Utilizando o ambiente `animateinline` para produzir imagens para animar com o `animategraphics`

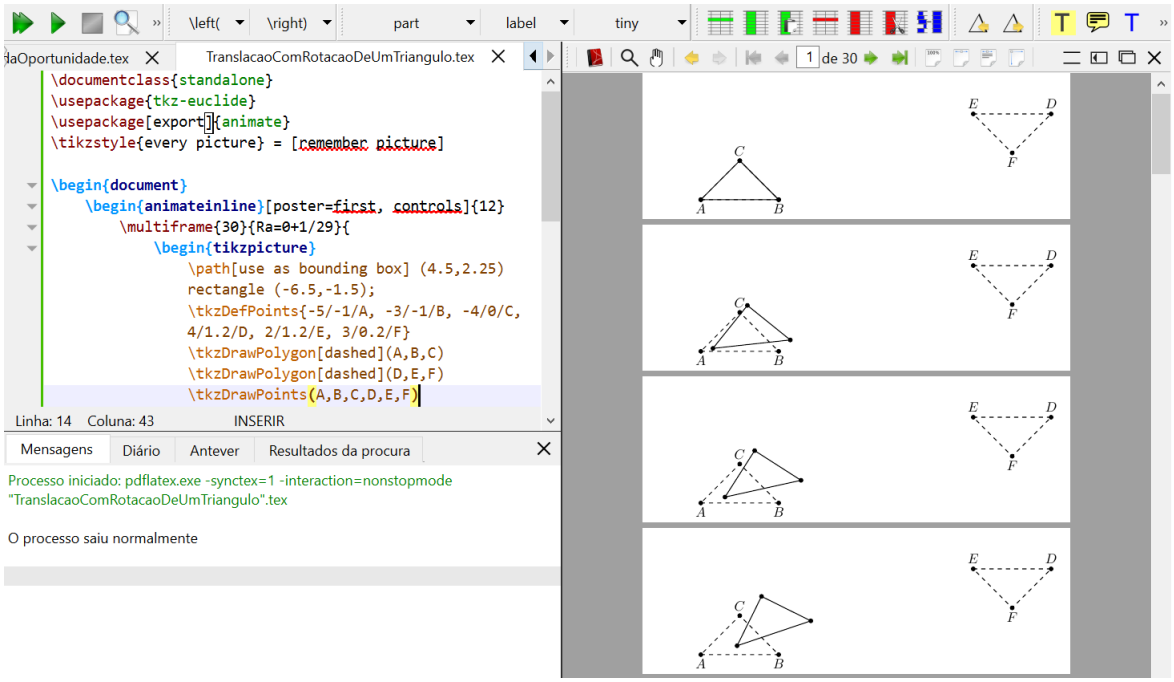
Quando produzimos animações com o ambiente `animateinline`, dependendo do tamanho da animação, o código pode ficar muito grande e, consequentemente, a compilação do arquivo demora muito tempo. Além disso, esse tempo ainda pode ser aumentado quando se produzem várias animações em um mesmo documento.

Como alternativa para diminuir o tempo de compilação, podemos utilizar o ambiente `animateinline` para produzir um arquivo com várias páginas, e utilizar esse arquivo para construir a animação com o comando `\animategraphics` por meio do procedimento descrito na Subseção 5.3.1. Para a produção desse arquivo, devemos escrever no preâmbulo

```
\usepackage[export]{animate}.
```
(5.59)

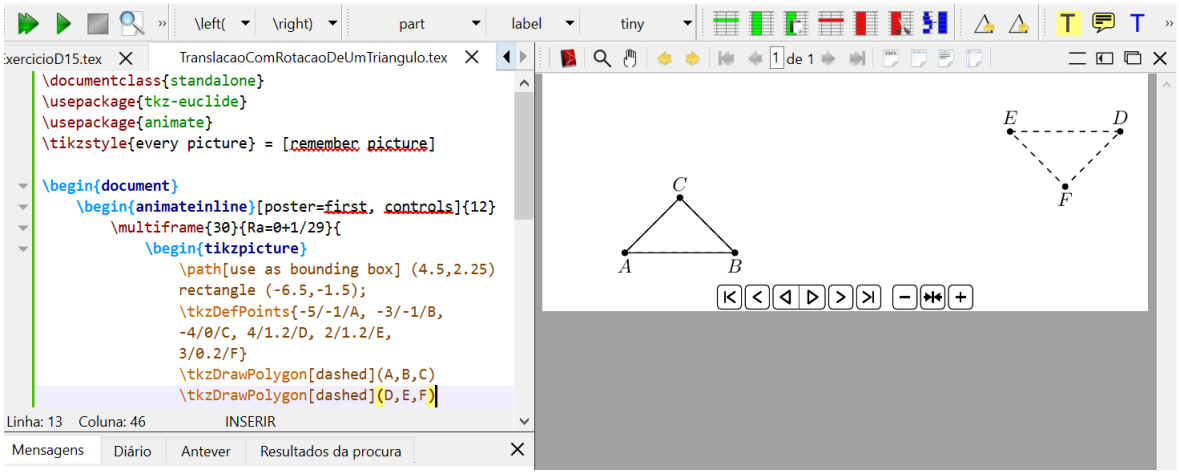
A opção `export` fará o `animate` produzir um arquivo com o número de páginas equivalente à soma de todos os *frames* declarados nos `multiframes` utilizados no código do `animateinline`, isto é, cada página representará um quadro da animação. A Figura 5.77 traz a interface do programa TeXstudio com parte do código que dá origem a um arquivo PDF com trinta páginas que foi utilizado para a produção da animação apresentada na Figura 5.75, enquanto que a Figura 5.78 traz parte do mesmo código, porém, sem a utilização da opção `export`. Compare os resultados obtidos observando o lado direito de cada figura.

Figura 5.77: Código com uso da opção `export`.



Fonte: Elaborada pelo autor.

Figura 5.78: Código sem uso da opção `export`.



Fonte: Elaborada pelo autor.

É importante lembrar que o arquivo PDF que possui as páginas da animação, produzido ao se utilizar a opção `export`, deve estar salvo na mesma pasta em que se encontra o arquivo TEX para a produção da animação com o comando `\animategraphics`.

Capítulo 6

Proposta de Sequência Didática

Esse capítulo é destinado à exposição de uma proposta de sequência didática para o ensino de Geometria Plana. Mais especificamente, para o ensino de áreas de figuras planas tais como: quadrado, retângulo, paralelogramo, triângulo, losango, trapézio, polígonos regulares e círculo. Serão propostas dez atividades, nas quais serão utilizadas animações produzidas no LaTeX para auxiliar no desenvolvimento das mesmas.

As atividades propostas objetivam levar ao aluno justificativas para as fórmulas obtidas. Essas justificativas muitas vezes são negligenciadas pelos autores ao elaborar os materiais didáticos.

6.1 Área de um Quadrado

6.1.1 Atividade 01: determinação da área de um quadrado de lado natural

Objetivo:

Esta atividade tem o propósito de justificar o fato da área de um quadrado de lado natural n ser dada por n^2.

Tempo estimado:

Duas aulas de 50 minutos.

Materiais:

Computador com leitor de PDF, projetor, régua (de preferência sem marcações), compasso, lápis, borracha, folhas com um quadrado de lado 3 cm desenhado e folhas com as instruções para o uso da régua e compasso.

Desenvolvimento:

Inicialmente o professor deve expor, por meio do projetor, a Definição 6.1.1 e

os Axiomas listados a seguir, os quais foram retirados de Neto [13]. Se necessário, explicar o que são Axiomas e discutir um pouco com os alunos sobre as informações que cada um traz.

Definicão 6.1.1. *Um quadrado é um quadrilátero que possui os quatro lados congruentes e os quatro ângulos internos retos.*

Axioma 6.1.1. *Polígonos congruentes têm áreas iguais.*

Axioma 6.1.2. *Se um polígono convexo é particionado em um número finito de outros polígonos convexos, então a área do polígono maior é a soma das áreas dos polígonos menores.*

Axioma 6.1.3. *Se um polígono (maior) contém outro (menor) em seu interior, então a área do polígono maior é maior que a área do polígono menor.*

Axioma 6.1.4. *A área de um quadrado de lado 1 cm é igual a 1 cm^2.*

Após a exposição e exploração dos axiomas, distribuir para os alunos as duas folhas, uma com um quadrado que possua exatamente 3 cm de lado e a outra com as instruções para os procedimentos a serem realizados com a régua e o compasso. Distribuir também as réguas e os compassos, caso não possua materiais suficientes para uso individual, optar por fazer esta atividade em grupo, dependendo da disponibilidade dos materiais.

Os procedimentos com régua e compasso a seguir visam particionar o quadrado desenhado no papel em nove quadradinhos de lados com comprimento 1 cm. Nesse momento, o professor deve mostrar a animação da Figura 6.1 e orientar os alunos para que sigam os seguintes passos:

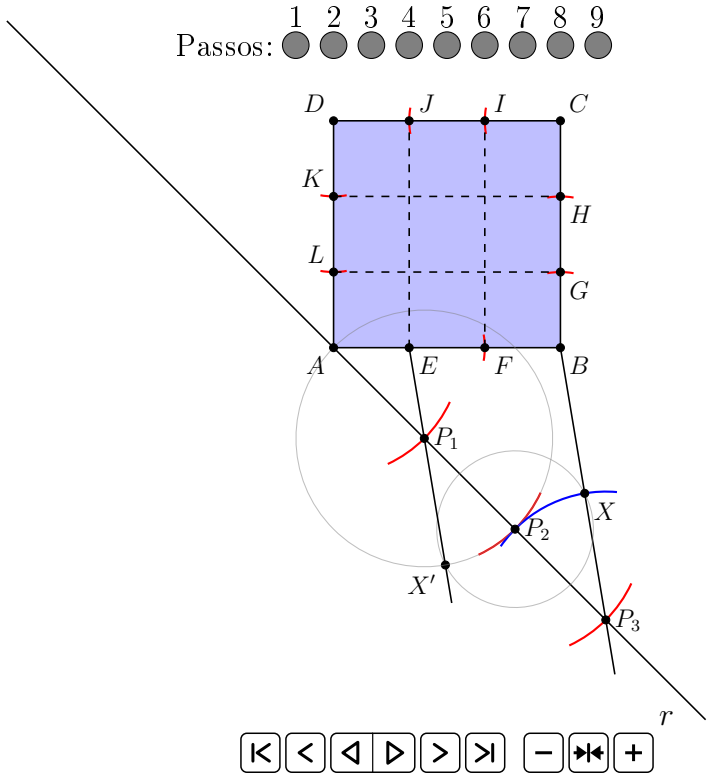
1. Trace uma reta r que incida no ponto A tal que $r \cap ABCD = \{A\}$;
2. Com o compasso com abertura arbitrária, trace um arco centrado em A marcando o ponto P_1 sobre a reta r. Conservando a abertura do compasso, centra-o em P_1 e marque o ponto P_2 sobre r. Ainda com a mesma abertura, centre o compasso em P_2 e marque o ponto P_3 sobre r;
3. Trace a reta $\overleftrightarrow{BP_3}$;
4. Com o compasso com abertura $\overline{P_3P_2}$, trace um arco centrado em P_3 marcando o ponto X sobre o segmento BP_3;
5. Construa a circunferência de raio $\overline{P_1P_2}$ centrada em P_1;
6. Trace a circunferência de centro P_2 e raio medindo $\overline{XP_2}$ e marque o ponto X' de interseção dessa circunferência com a construída no passo anterior de forma que X' esteja no semiplano oposto a X em relação à reta r;

7. Trace a reta $\overleftrightarrow{X'P_1}$ marcando o ponto E de interseção dessa reta com o segmento AB;

8. Com abertura $\overline{AE}$ use o compasso para dividir os lados do quadrado $ABCD$ em segmentos iguais, marcando os pontos F, G, H, I, J, K e L;

9. Ligue os pontos que estão em lados opostos do quadrado particionando-o em nove quadradinhos congruentes estre si.

Figura 6.1: Área de um quadrado de lado 3 cm.



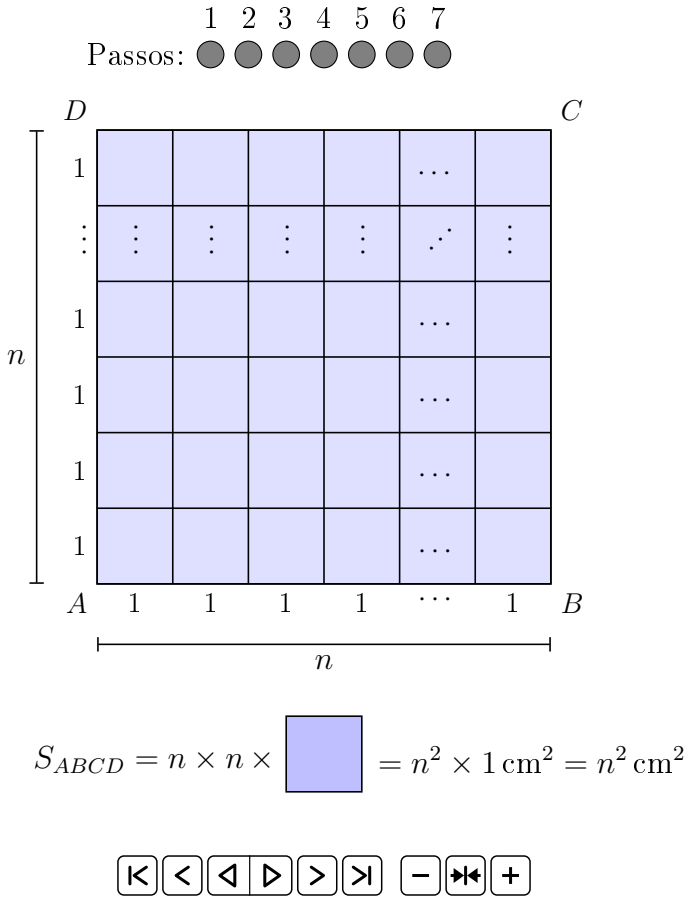
Fonte: Elaborada pelo autor.

É importante que os alunos não utilizem a graduação da régua caso ela possua, pois isso possibilitará o surgimento de questionamentos como: quem garante que o segmento AE mede 1 cm de comprimento? Esta questão e outras que possam surgir são importantes e devem ser discutidas com cautela juntamente com os alunos.

Após a construção com régua e compasso, de acordo com o Axioma 6.1.1 todos os nove quadradinhos de lado 1 cm possuem mesma área e, de acordo com o Axioma 6.1.4, as medidas dessas áreas são todas iguais a $1\,\text{cm}^2$. Sendo assim, fazendo uso do Axioma 6.1.2, a área do quadrado $ABCD$ é igual à soma das áreas dos nove quadradinhos de lado 1 cm, ou seja, a área de um quadrado de lado 3 cm é igual a $9\,\text{cm}^2$ o que equivale a $3^2\,\text{cm}^2$.

É importante que fique claro para o aluno que esse procedimento não constitui prova para a fórmula da área de um quadrado de lado natural n, mas serve para facilitar a compreensão da generalização do mesmo, isto é, para provarmos que a área de um quadrado de lado natural n é dada por n^2, devemos proceder de maneira semelhante, porém considerando um quadrado de lado natural arbitrário. Sendo assim, após expor aos alunos tal raciocínio e debater a respeito, o professor passará a apresentar, via projetor, a animação da Figura 6.2.

Figura 6.2: Área de um quadrado de lado natural.



Fonte: Elaborada pelo autor.

Simultaneamente à apresentação da animação da Figura 6.2, o professor deve explicar, fazendo uso dos Axiomas 6.1.1, 6.1.2 e 6.1.4, que um quadrado $ABCD$ de lado natural n pode ser particionado em quadradinhos de lado $1\,\text{cm}$, sendo que, para tanto, basta traçar sucessivas retas paralelas a AB e sucessivas retas paralelas a AD, sempre mantendo a distância de $1\,\text{cm}$. Esse procedimento particiona o quadrado $ABCD$ em n fileiras de n quadradinhos cada uma, totalizando $n \times n = n^2$ quadradinhos, com $1\,\text{cm}^2$ de área cada um. Isso prova que a área de um quadrado de lado natural n é dada pela fórmula n^2.

6.1.2 Atividade 02: determinação da área de um quadrado de lado racional

Objetivo:

Esta atividade objetiva justificar o fato da área de um quadrado de lado racional r ser dada por r^2.

Tempo estimado:

Duas aulas de 50 minutos.

Materiais:

Computador com leitor de PDF, projetor, régua (de preferência sem marcações), compasso, lápis, borracha, folhas com um quadrado de lado $\frac{5}{3}$ cm desenhado e folhas com as instruções para o uso de régua e compasso.

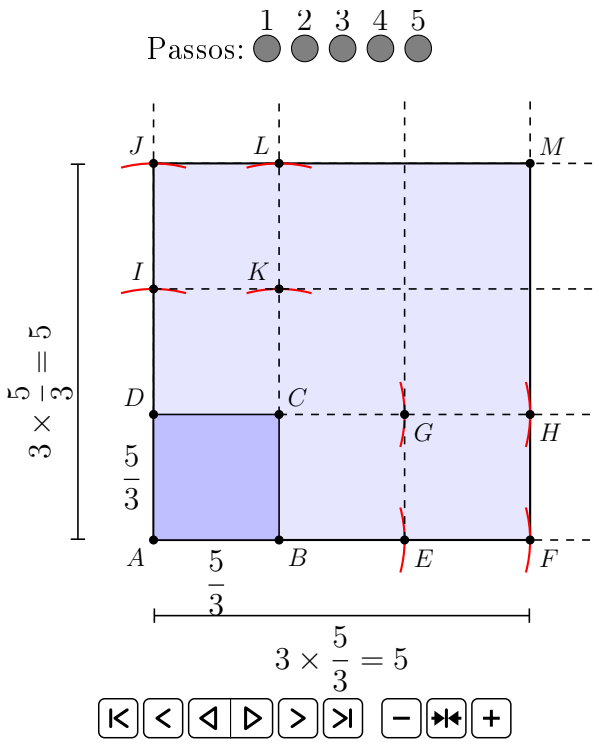
Desenvolvimento:

Distribuir aos alunos duas folhas, uma com um quadrado de lado $\frac{5}{3}$ cm e a outra com as instruções para os procedimentos com régua e compasso. Distribuir também as réguas e os compassos.

Os procedimentos com régua e compasso a seguir visão construir um quadrado de lado natural utilizando a justaposição de nove quadrados congruentes ao quadrado de lado $\frac{5}{3}$ cm. Neste momento, o professor deve orientar os alunos a utilizarem a régua e o compasso para realizarem os passos a seguir, mostrando, via projetor, a animação da Figura 6.3 que deve ser utilizada para facilitar a construção.

1. Prolongue as semirretas $\overrightarrow{AB}$, $\overrightarrow{DC}$, $\overrightarrow{AD}$ e $\overrightarrow{BC}$;
2. Com abertura igual a $\overline{AB}$, use o compasso para marcar os pontos E e F sobre a semirreta $\overrightarrow{AB}$, G e H sobre a semirreta $\overrightarrow{DC}$, I e J sobre a semirreta $\overrightarrow{AD}$ e K e L sobre a semirreta $\overrightarrow{BC}$, de forma que, $\overline{AB} = \overline{BE} = \overline{EF} = \overline{CG} = \overline{GH} = \overline{DI} = \overline{IJ} = \overline{CK} = \overline{KL}$ e $E \in BF$, $G \in CH$, $I \in DJ$ e $K \in CL$;
3. Prolongue as semirretas $\overrightarrow{EG}$ e $\overrightarrow{FH}$;
4. Prolongue as semirretas $\overrightarrow{IK}$ e $\overrightarrow{JL}$ e marque o ponto M tal que $\overrightarrow{FH} \cap \overrightarrow{JL} = \{M\}$;
5. Desenhe o quadrado $AFMJ$, ele possui 5 cm de lado.

Figura 6.3: Área de um quadrado de lado $\frac{5}{3}$ cm.



Fonte: Elaborada pelo autor.

Após a construção, o quadrado $AFMJ$ será formado por nove quadrados congruentes ao quadrado $ABCD$ e, de acordo com o Axioma 6.1.1, todos esses nove quadrados possuem a mesma área, a qual vamos chamar de S_{ABCD}. De acordo com o Axioma 6.1.2, a área S_{AFMJ} do quadrado $AFMJ$ é dada pela soma das áreas dos nove quadrados congruentes a $ABCD$ ou, equivalentemente,

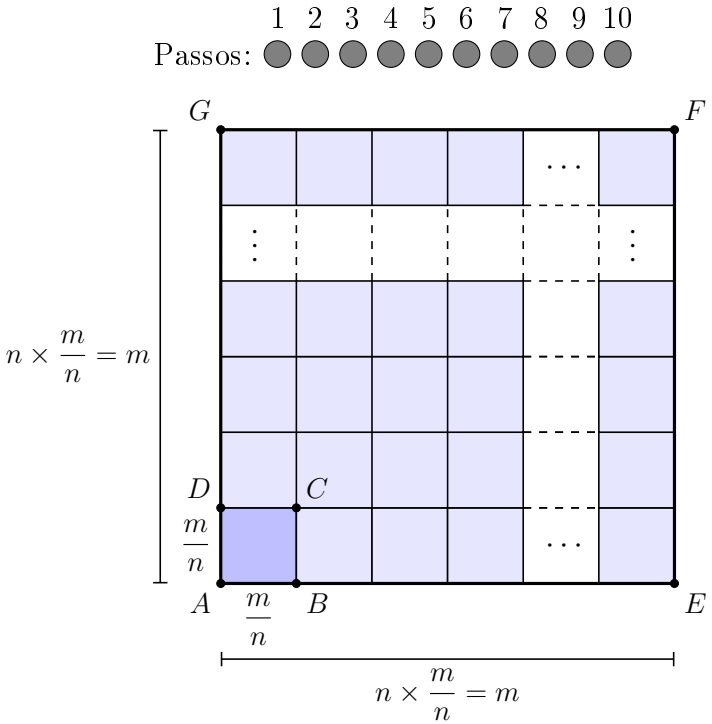
$$S_{AFMJ} = 3^2 \times S_{ABCD}.$$

Note que o lado do quadrado $AFMJ$ é natural e igual a 5 cm, visto que é dado por $3 \times \frac{5}{3}$ cm. Como já sabemos que a área de um quadrado de lado natural é dada pelo quadrado do seu lado, segue que $S_{AFMJ} = 5^2\,\text{cm}^2$ e com isso

$$5^2 cm^2 = 3^2 \times S_{ABCD} \Longrightarrow S_{ABCD} = \frac{5^2}{3^2} cm^2 \Longrightarrow S_{ABCD} = \left(\frac{5}{3}\right)^2 cm^2.$$

Esse procedimento apenas sugere que a área de um quadrado de lado racional r seja igual a r^2, servindo para preparar o aluno para a demonstração que consiste em tratar esse procedimento de maneira genérica. Sendo assim o professor passará a expor, via projetor, a animação da Figura 6.4.

Figura 6.4: Área de um quadrado de lado racional.



$$S_{AEFG} = n^2 \times S_{ABCD} \implies m^2 = n^2 \times S_{ABCD}$$
$$\implies S_{ABCD} = \frac{m^2}{n^2}$$
$$\implies S_{ABCD} = \left(\frac{m}{n}\right)^2$$

Fonte: Elaborada pelo autor.

Considerando o quadrado $ABCD$ de lado racional $r = \dfrac{m}{n}$ da Figura 6.4, é possível enfileirar, um ao lado do outro, n quadrados congruentes a $ABCD$. Após isso, empilhamos n dessas filas, formando um quadrado maior $AEFG$ com lado medindo $n \times \dfrac{m}{n} = m$. Com isso, o quadrado $AEFG$ será formado por n filas com n quadrados congruentes a $ABCD$, isto é, $AEFG$ será particionado em n^2 quadrados congruentes a $ABCD$. Assim, pelos Axiomas 6.1.1 e 6.1.2, segue que

$$S_{AEFG} = n^2 \times S_{ABCD}.$$

Dado que o quadrado $AEFG$ possui lado natural e igual a m e que a área de um quadrado de lado natural é o quadrado da medida de seu lado, segue que $S_{AEFG} = m^2$ e com isso

$$m^2 = n^2 \times S_{ABCD} \implies S_{ABCD} = \frac{m^2}{n^2} \implies S_{ABCD} = \left(\frac{m}{n}\right)^2.$$

Portanto a área de um quadrado de lado racional $r = \frac{m}{n}$, com m e n naturais, é igual a r^2.

6.2 Área de um Retângulo

De posse dos resultados apresentados na seção anterior propomos a utilização da animação apresentada na Figura 6.5 para auxiliar o professor na demonstração da fórmula da área de um retângulo.

6.2.1 Atividade 03: determinação da área de um retângulo

Objetivo:

Esta atividade tem o objetivo de justificar o fato da área de um retângulo de base b e altura h ser dada por $b \times h$.

Tempo estimado:

Uma aula de 50 minutos.

Materiais:

Computador com leitor de PDF, projetor, régua (de preferência sem marcações), compasso, lápis, borracha, folhas com um retângulo de base arbitrária b e altura arbitrária h desenhado e folhas com as instruções para o uso de régua e compasso.

Desenvolvimento:

Inicialmente, o professor apresenta aos estudantes a Definição 6.2.1 e a Proposição 6.2.1.

Definicão 6.2.1. *Um retângulo é um quadrilátero convexo que possui os quatro ângulos internos retos.*

Proposição 6.2.1. *A área S_{ABCD} de um retângulo $ABCD$ com base b e altura h, é igual a $b \times h$.*

Distribuir aos alunos duas folhas, uma com um retângulo $ABCD$ de base $\overline{AB} = b$ e altura $\overline{AD} = h$, e a outra com as instruções para o uso da régua e compasso. Distribuir também as réguas e os compassos.

Os procedimentos com régua e compasso a seguir visam construir um quadrado de lado $b+h$ que seja formado pela justaposição de dois quadrados, um com lado b e outro com lado h, e dois retângulos congruentes ao retângulo $ABCD$. Sendo assim, o professor passará a orientar os estudantes a utilizarem a régua e o compasso para seguirem os passos a seguir, mostrando via projetor a animação da Figura 6.5.

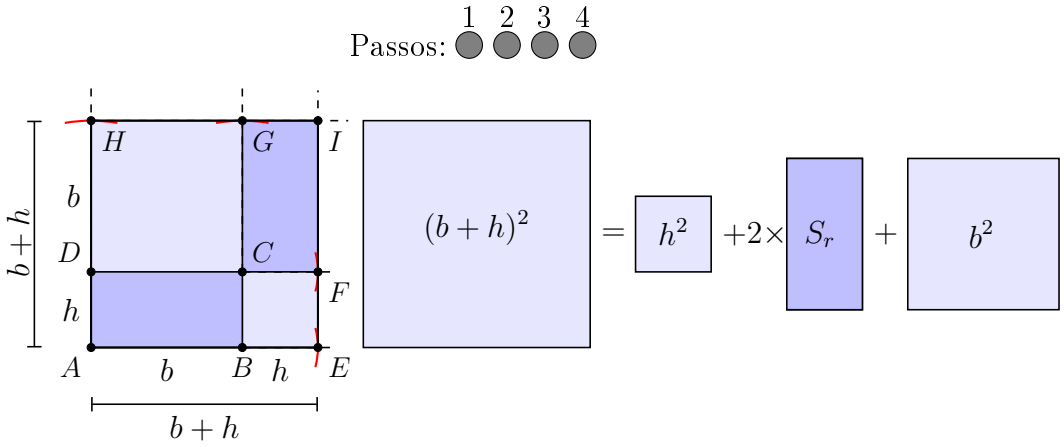
1. Prolongue as semirretas $\overrightarrow{AB}$, $\overrightarrow{DC}$, $\overrightarrow{AD}$ e $\overrightarrow{BC}$;

2. Com o compasso com abertura $\overline{AD} = h$, marque o ponto E sobre $\overrightarrow{AB}$ e o ponto F sobre $\overrightarrow{DC}$ de forma que $B \in AE$ e $C \in DF$. Com o compasso com abertura $\overline{AB} = b$, marque os pontos G e H, respectivamente, sobre $\overrightarrow{BC}$ e $\overrightarrow{AD}$ tais que $C \in BG$ e $D \in AH$;

3. Prolongue as semirretas $\overrightarrow{EF}$ e $\overrightarrow{HG}$, mancando o ponto I de interseção das mesmas;

4. Desenhe o quadrado $AEIH$, ele possui lado medindo $b + h$.

Figura 6.5: Área de um retângulo.



$$(b+h)^2 = b^2 + 2 \times S_r + h^2 \Longrightarrow b^2 + 2 \times b \times h + h^2 = b^2 + 2 \times S_r + h^2 \Longrightarrow S_r = b \times h$$

Fonte: Elaborada pelo autor com base em Lima [34].

Após a construção do quadrado $AEIH$ (de lado $b+h$), o professor deve continuar com a animação da Figura 6.5, mostrando que a área de tal quadrado é dada pela soma das áreas do quadrado $BEFC$ (de lado h), com o dobro da área do retângulo $ABCD$ e com a área do quadrado $DCGH$ (de lado b). Visto que já sabemos calcular a área de um quadrado e chamando a área do retângulo $ABCD$ de S_r, segue que

$$(b+h)^2 = b^2 + 2 \times S_r + h^2.$$

Desenvolvendo o produto notável no primeiro membro da equação, obtemos

$$b^2 + 2 \times b \times h + h^2 = b^2 + 2 \times S_r + h^2 \Longrightarrow S_r = b \times h.$$

Portando, a área S_r de um retângulo de base b e altura h é dada pelo produto $b \times h$, isto é, $S_r = b \times h$.

6.3 Área de um Paralelogramo

6.3.1 Atividade 04: determinação da área de um paralelogramo

Objetivo:

Esta atividade tem o objetivo de justificar o fato da área de um paralelogramo de base b e altura relativa a esta base h ser dada por $b \times h$.

Tempo estimado:

Uma aula de 50 minutos.

Materiais:

Computador com leitor de PDF; projetor e um arquivo PDF que possua a animação da Figura 6.6.

Desenvolvimento:

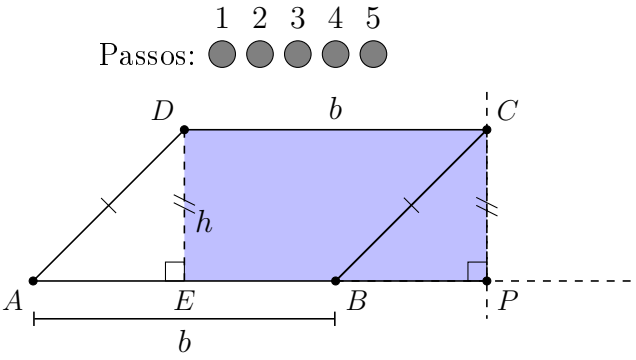
O professor deve apresentar aos estudantes, por meio do projetor, a definição de paralelogramo 6.3.1 e a Proposição 6.3.1. Aproveite também para definir a base e a altura de um paralelogramo, sendo estes, respectivamente, um dos lados e a distância entre tal lado e o que está na posição oposta. Nos termos da Figura 6.6, temos o paralelogramo $ABCD$, no qual se tem a base AB com medida b e a altura DE cuja medida é h.

Definicão 6.3.1. *Um paralelogramo é um quadrilátero que possui os lados opostos paralelos.*

Proposição 6.3.1. *A área S_{ABCD} de um paralelogramo $ABCD$ com base b e altura h, é igual $b \times h$.*

Após discutir as definições mencionadas e a Proposição 6.3.1, o professor passa a apresentar, via projetor, a animação da Figura 6.6 e, simultaneamente, a apresentação, explicar que é possível prolongar a base AB do paralelogramo e traçar a perpendicular à semirreta $\overrightarrow{AB}$ que passa pelo ponto C, marcando assim o ponto P, interseção dessa perpendicular com $\overrightarrow{AB}$.

Figura 6.6: Área de um Paralelogramo.



$$AED \equiv BPC \implies S_{ABCD} = S_{EPCD} \implies S_{ABCD} = b \times h$$

Fonte: Elaborada pelo autor.

Sendo $ABCD$ um paralelogramo, segue que $\overline{AD} = \overline{BC}$ e $\overline{DE} = \overline{CP}$. Com isso, e pelo fato dos ângulos $\angle AED$ e $\angle BPC$ serem retos, conclui-se, pelo caso lado, lado, ângulo oposto (LLA_o), que os triângulos AED e BPC são congruentes.

A congruência dos triângulos AED e BPC garante que a área S_{ABCD} do paralelogramo $ABCD$ seja igual à área S_{EPCD} do retângulo $EPCD$ que, por sua vez é igual a $b \times h$, isto é, $S_{ABCD} = b \times h$.

Portanto a área S_{ABCD} de um paralelogramo $ABCD$ com base b e altura h é dada por $S_{ABCD} = b \times h$, como queríamos demonstrar.

6.4 Área de um Triângulo

6.4.1 Atividade 05: determinação da área de um triângulo

Objetivo:

Esta atividade tem o objetivo de justificar o fato da área de um triângulo de base b e altura h ser dada por $\dfrac{b \times h}{2}$.

Tempo estimado:

Uma aula de 50 minutos.

Materiais:

Computador com leitor de PDF; projetor e um arquivo PDF que possua a animação da Figura 6.7.

Desenvolvimento:

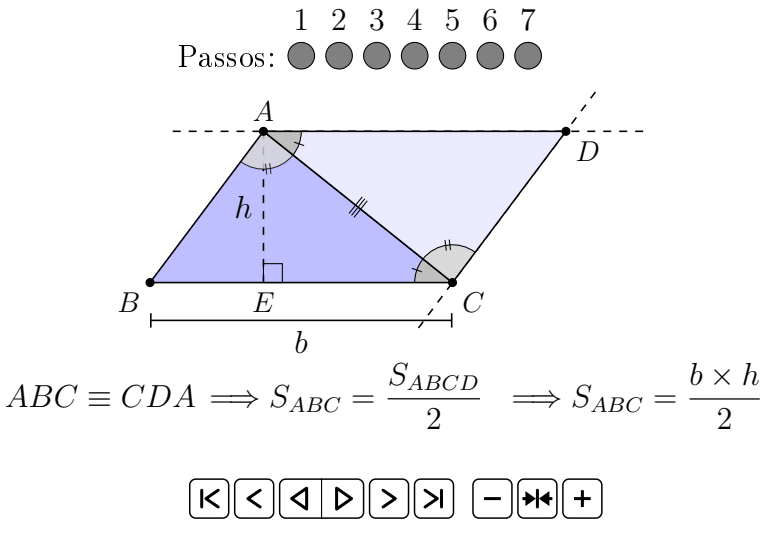
Apresentar, via projetor, a definição de triângulo, Definição 6.4.1, e a Proposição 6.4.1, definindo também a base e a altura de um triângulo como sendo, respectivamente, um dos lados do triângulo e a distância do vértice oposto ao lado considerado à reta suporte de tal lado. Nos termos da Figura 6.7, temos o triângulo ABC, no qual se tem a base BC e altura AE com medidas respectivamente iguais a b e h.

Definicão 6.4.1. *Um triângulo é um polígono formado por três vértices.*

Proposição 6.4.1. *A área S_{ABC} de um triângulo ABC com base b e altura h é igual a* $\dfrac{b \times h}{2}$.

Após discutir as definições e a Proposição 6.4.1, o professor passa a apresentar, via projetor, a animação da Figura 6.7 e, simultaneamente, a apresentação, explicar que é possível traçar as duas retas paralelas às retas $\overleftrightarrow{AB}$ e $\overleftrightarrow{BC}$ que passam, respectivamente, pelos pontos C e A, marcando o ponto D de interseção das retas traçadas.

Figura 6.7: Área de um triângulo.



$$ABC \equiv CDA \implies S_{ABC} = \frac{S_{ABCD}}{2} \implies S_{ABC} = \frac{b \times h}{2}$$

Fonte: Elaborada pelo autor.

Sendo $\overleftrightarrow{BC}$ paralela com $\overleftrightarrow{AD}$, segue que os ângulos $\angle BCA$ e $\angle DAC$ são congruentes e, pelo fato de $\overleftrightarrow{AB}$ ser paralela a $\overleftrightarrow{DC}$, conclui-se que $\angle BAC$ e $\angle DCA$ também são congruentes. Essas duas informações juntamente com o fato do lado AC ser comum aos triângulos ABC e CDA, garantem, pelo caso de congruência ângulo, lado, ângulo (ALA), que os dois triângulos ABC e CDA são congruentes, donde podemos concluir que a área do triângulo ABC é igual a metade da área do quadrilátero $ABCD$.

O quadrilátero $ABCD$ é um paralelogramo por construção. Portanto sua área é igual a $b \times h$. Assim, a área S_{ABC} do triângulo ABC é igual a metade do produto $b \times h$, isto é,

$$S_{ABC} = \frac{b \times h}{2},$$

como queríamos demonstrar.

6.5 Área de um Losango

6.5.1 Atividade 06: determinação da área de um losango

Objetivo:

Esta atividade tem o objetivo de justificar o fato da área de um losango de diagonais D_m e d ser dada por $\frac{D_m \times d}{2}$.

Tempo estimado:

Uma aula de 50 minutos.

Materiais:

Computador com leitor de PDF, projetor e um arquivo PDF que possua a animação da Figura 6.8.

Desenvolvimento:

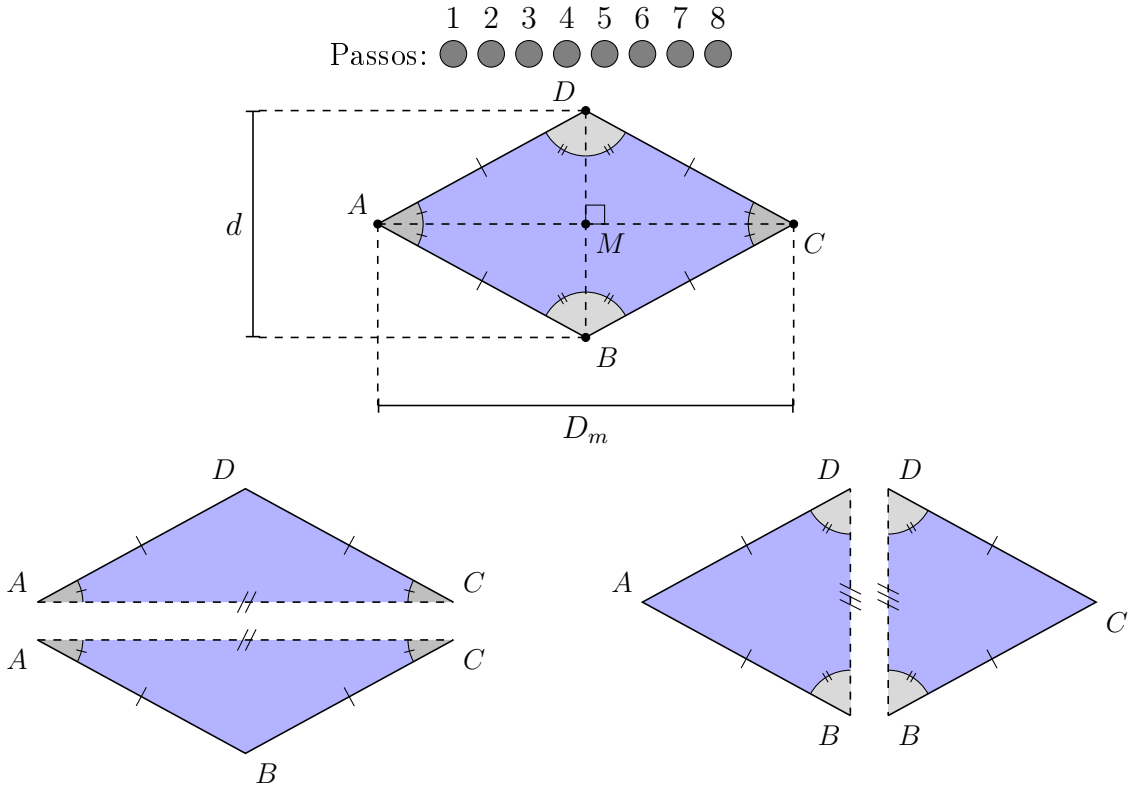
Apresentar a Definição 6.5.1 e a Proposição 6.5.1.

Definicão 6.5.1. *Um losango é um quadrilátero que possui os quatro lados com a mesma medida.*

Proposição 6.5.1. *A área de um losango $ABCD$ com diagonais medindo D_m e d, é igual a $\frac{D_m \times d}{2}$.*

Na Figura 6.8 temos o losango $ABCD$ cujas diagonais são AC, com medida D_m, e BD, com medida d.

Figura 6.8: Área de um losango.



$AMD \equiv CMD \equiv CMB \equiv AMB \Longrightarrow \overleftrightarrow{AC} \perp \overleftrightarrow{BD}$ e M ponto médio das diagonais AC e BD

$$S_{ABCD} = 4 \times S_{AMD} \implies S_{ABCD} = 4 \times \frac{\frac{D_m}{2} \times \frac{d}{2}}{2} \implies S_{ABCD} = \frac{D_m \times d}{2}$$

Fonte: Elaborada pelo autor.

Após a exibição da Definição 6.5.1 e da Proposição 6.5.1, o professor passa a apresentar, via projetor, a animação da Figura 6.8, explicando simultaneamente à apresentação que dado um losango $ABCD$, podemos traçar as diagonais AC e BD representando suas medidas por D_m e d, respectivamente. Marca-se o ponto M de interseção das duas diagonais.

Sendo $ABCD$ um losango, segue que os lados AB, BC, CD e DA dos triângulos ABC e ADC possuem a mesma medida. Com isso, e pelo fato do lado AC ser comum aos dois triângulos, concluímos, pelo caso de congruência lado, lado, lado (LLL), que ABC e ADC são congruentes e que ambos são triângulos isósceles de base AC. Assim, os ângulos $\angle DAC$, $\angle DCA$, $\angle BCA$ e $\angle BAC$ possuem a mesma medida. Considerando os triângulos ABD e CBD e utilizando um raciocínio análogo, concluímos que os ângulos $\angle ADB$, $\angle CDB$, $\angle CBD$ e $\angle ABD$ possuem medidas iguais. A igualdade desses ângulos juntamente com o fato do quadrilátero $ABCD$

ser um losango garantem, pelo caso de congruência ângulo, lado, ângulo (ALA), a congruência dos quatro triângulos AMD, CMD, CMB e AMB. Essa congruência implica a perpendicularidade das diagonais AC e BD e o fato de M ser o ponto médio dessas diagonais.

Assim, a área S_{ABCD} do losango $ABCD$ é igual à soma das áreas dos quatro triângulos AMD, CMD, CMB e AMB. Como esses triângulos são congruentes, a área do losango é igual a quatro vezes a área de um desses triângulos, isto é,

$$S_{ABCD} = 4 \times S_{AMD} \Longrightarrow S_{ABCD} = 4 \times \frac{\frac{D_m}{2} \times \frac{d}{2}}{2}.$$

Portanto, obtemos $S_{ABCD} = \dfrac{D_m \times d}{2}$, como queríamos demonstrar.

6.6 Área de um Trapézio

6.6.1 Atividade 07: determinação da área de um trapézio

Objetivo:

Esta atividade tem o objetivo de justificar o fato da área de um trapézio de bases B e b e altura h, ser dada por $\dfrac{(B+b) \times h}{2}$.

Tempo estimado:

Uma aulas de 50 minutos.

Materiais:

Computador com leitor de PDF, projetor e um arquivo PDF que possua a animação da Figura 6.9.

Desenvolvimento:

Inicialmente, o professor deve apresentar, via projetor, a definição de trapézio, Definição 6.6.1. Adiantar aos estudantes que um trapézio possui duas bases, as quais são os lados paralelos, e uma altura que é a distância entre as duas bases. Nos termos da Figura 6.9, temos o trapézio $CDEF$ com bases CD e EF, cujas medidas são respectivamente B e b, a altura está representada pelo segmento FG com medida h. Apresentar também a Proposição 6.6.1.

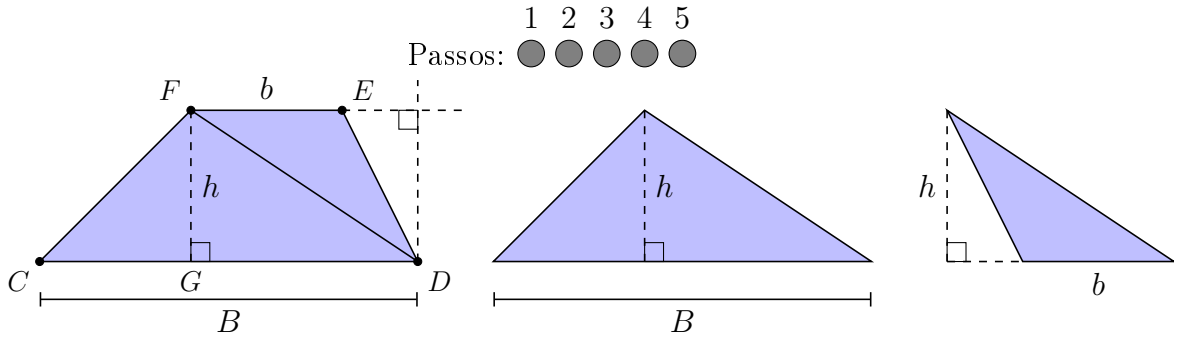
Definicão 6.6.1. *Um trapézio é um quadrilátero que possui um par de lados paralelos.*

Proposição 6.6.1. *A área S_{CDEF} de um trapézio $CDEF$ com bases e altura medindo, respectivamente, B, b e h, é igual a $\dfrac{(B+b) \times h}{2}$.*

Após a apresentação da Definição 6.6.1 e da Proposição 6.6.1, o professor passa a apresentar a animação da Figura 6.9, explicando ao estudante que o trapézio $CDEF$

pode ser particionado em dois triângulos: traçando-se a diagonal DF, construímos os triângulos CDF e DEF, que possuem a mesma altura h e, respectivamente, bases iguais a B e b.

Figura 6.9: Área de um trapézio.



$$S_{CDEF} = S_{CDF} + S_{EFF} \Longrightarrow S_{CDEF} = \frac{B \times h}{2} + \frac{b \times h}{2} \Longrightarrow S_{CDEF} = \frac{(B+b) \times h}{2}$$

Fonte: Elaborada pelo autor.

A área S_{CDEF} do trapézio $CDEF$ é igual à soma das áreas dos triângulos CDF e DEF. Como já sabemos que a área de um triângulo é igual a metade do produto entre a sua base e a sua altura, segue que

$$S_{CDEF} = \frac{B \times h}{2} + \frac{b \times h}{2} \Longrightarrow S_{CDEF} = \frac{(B+b) \times h}{2},$$

como queríamos demonstrar.

6.7 Área de um Polígono Regular

As atividades desta seção visam auxiliar o professor do Ensino Médio na demonstração das fórmulas das áreas de um triângulo equilátero e de um hexágono regular. Para os outros polígonos regulares pode-se seguir um raciocínio totalmente análogo.

Para a realização plena dessas atividades é necessário que os estudantes saibam identificar e calcular as razões trigonométricas no triângulo retângulo.

6.7.1 Atividade 08: determinação da área de um triângulo equilátero

Objetivo:

Esta atividade tem o objetivo de justificar o fato da área de um triângulo equilátero ABC de lado l, semiperímetro p e apótema a, ser dada por $S_{ABC} = p \times a$ ou $S_{ABC} = \dfrac{\sqrt{3} \times l^2}{4}$.

Tempo estimado:

Duas aulas de 50 minutos.

Materiais:

Computador com leitor de PDF, projetor e um arquivo PDF que possua as animações das Figuras 6.10, 6.11 e 6.12.

Desenvolvimento:

Inicialmente, apresentar a definição de polígino regular 6.7.1 e a proposição 6.7.1.

Definicão 6.7.1. *Um polígono é regular se for convexo e possuir todos os seus lados com a mesma medida.*

Proposição 6.7.1. *Todo polígono regular é inscritível em uma circunferência.*

Apresentar, via projetor, a animação da Figura 6.10 destacando que todo polígono regular possui os seguintes elementos:

1. Centro: ponto central do polígono que coincide com o centro da circunferência que o circunscreve;

2. Raio: segmento que liga o centro do polígono a um de seus vértices, é o mesmo raio da circunferência que o circunscreve;

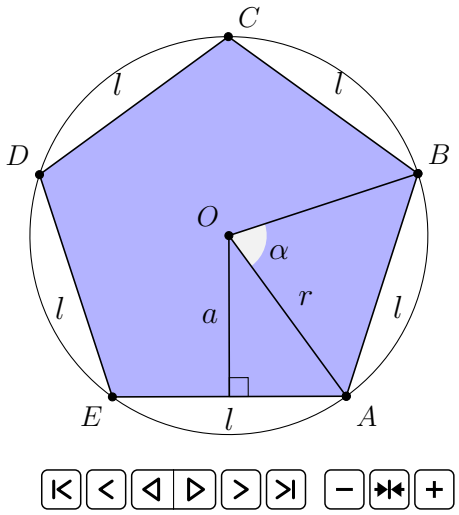
3. Semiperímetro: medida da metade da soma das medidas dos lados de um polígono;

4. Apótema: segmento que representa a distância de um lado ao centro do polígono;

5. Ângulo central: ângulo que possui o centro do polígono como vértice e dois raios consecutivos como lados;

Nos termos da Figura 6.10, o polígono $ABCDE$ é regular (pentágono regular) e possui centro O, raio r, apótema a e ângulo central α.

Figura 6.10: Elementos de um polígono regular.



Fonte: Elaborada pelo autor.

Após expor essas informações e debater com os estudantes, apresentar a Proposição 6.7.2 e enfatizar que um triângulo equilátero é um polígono regular e, portanto, possui todos os elementos citados anteriormente.

Proposição 6.7.2. *A área S_{ABC} de um triângulo equilátero ABC de semiperímetro p e apótema a é dada por $S_{ABC} = p \times a$.*

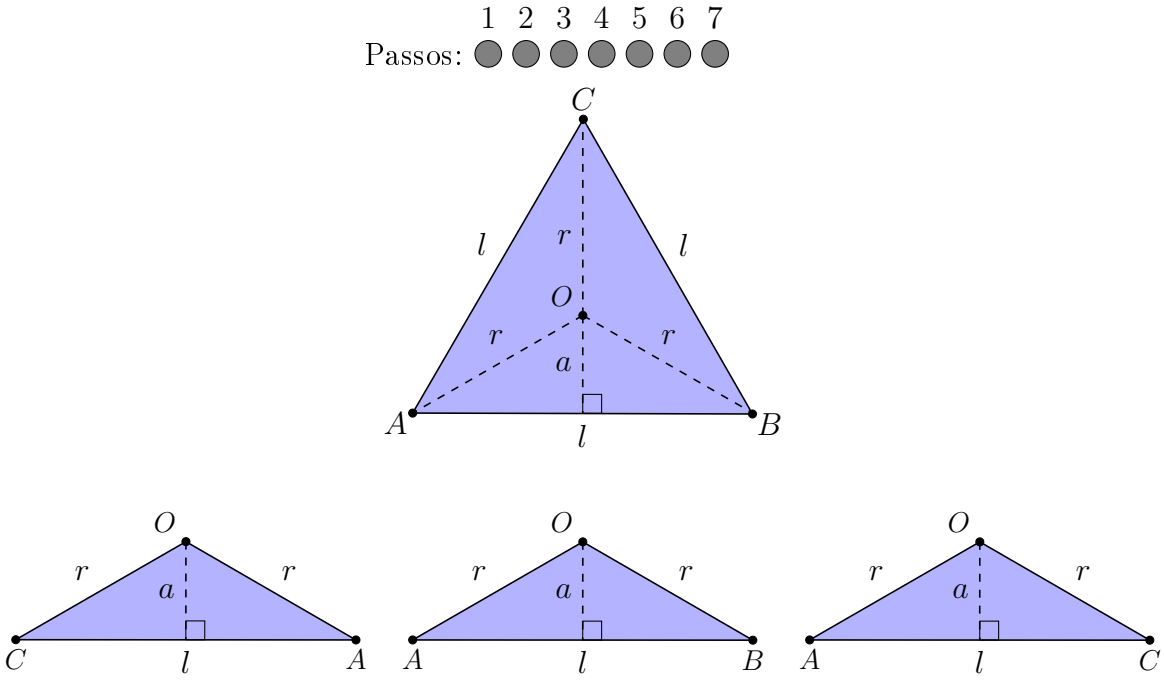
Passe a apresentar, via projetor, a animação da Figura 6.11 descrevendo simultaneamente que o triângulo equilátero ABC é particionado pelos seus raios em três triângulos ABO, BCO e CAO, e que tais triângulos são congruentes pelo caso lado, lado, lado (LLL).

A congruência dos triângulos ABO, BCO e CAO nos leva a concluir que a área S_{ABC} é igual a três vezes a área de um deles. Mas como o apótema do triângulo ABC é igual a altura dos triângulos ABO, BCO e CAO, segue que

$$S_{ABC} = 3 \times S_{ABO} = 3 \times \frac{l \times a}{2} = \frac{3 \times l}{2} \times a \Longrightarrow S_{ABC} = p \times a,$$

como queríamos demonstrar.

Figura 6.11: Área de um triângulo equilátero, dado seu lado e apótema.



$$S_{ABC} = 3 \times S_{ABO} \Longrightarrow S_{ABC} = 3 \times \frac{l \times a}{2} \Longrightarrow S_{ABC} = \frac{3 \times l}{2} \times a \implies S_{ABC} = p \times a$$

Fonte: Elaborada pelo autor.

A área de um triângulo equilátero também pode ser calculada por outra fórmula que está indicada na Proposição 6.7.3. Para a demonstração dessa proposição, faz-se necessário o uso de um outro resultado indicado na Proposição 6.7.4. Portanto, o professor deve apresentar tais proposições aos alunos e discuta um pouco sobre elas.

Proposição 6.7.3. *A área S_{ABC} de um triângulo equilátero ABC de lado l é dada por* $S_{ABC} = \dfrac{\sqrt{3} \times l^2}{2}$.

Proposição 6.7.4. *A altura relativa à base de um triângulo isósceles coincide com a bissetriz do ângulo do vértice e com a mediana relativa à mesma base.*

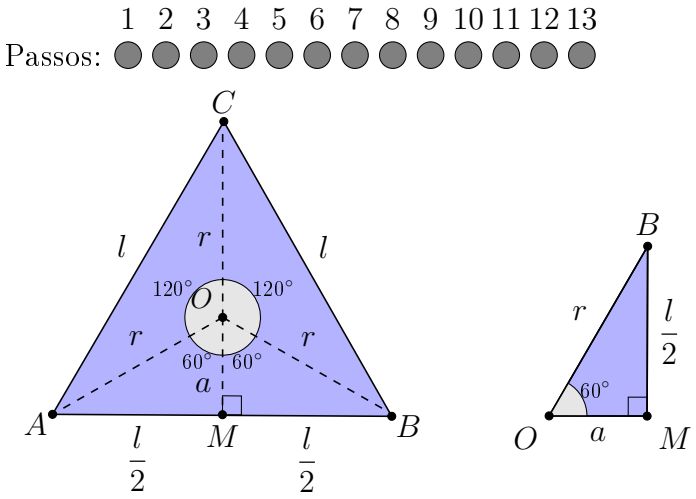
Agora, o professor passa a apresentar a animação da Figura 6.12 e simultaneamente à apresentação, deve mostrar novamente para os estudantes que, pelo caso de congruência lado, lado, lado (LLL), os triângulos ABO, BCO e CAO são congruentes. Essa congruência garante que os ângulos centrais do triângulo ABC sejam todos congruentes e medem 120°, visto que, ao somá-los, devemos encontrar o ângulo de uma volta, cuja medida é 360°.

Ao traçarmos o apótema do triângulo ABC, ele também representará a altura do triângulo ABO que, por sua vez, é isósceles de base AB e, pela Proposição 6.7.4, essa altura coincide com a bissetriz do ângulo $\angle AOB$ e com a mediana relativa a AB, ou seja, o ângulo $\angle MOB$ mede $60°$ e o segmento MB mede $l/2$, sendo M o pé do apótema relativo ao lado AB.

Temos então, o triângulo retângulo MBO, donde podemos determinar o valor de a em função de l por meio da tangente do ângulo $\angle MOB$,

$$tg\,60° = \frac{l/2}{a} \Longrightarrow \sqrt{3} = \frac{l/2}{a} \Longrightarrow a = \frac{\sqrt{3} \times l}{6}.$$

Figura 6.12: Área de um triângulo equilátero, dado seu lado.



$$tg\,60° = \frac{l/2}{a} \Longrightarrow \sqrt{3} = \frac{l/2}{a} \Longrightarrow \sqrt{3} \times a = \frac{l}{2} \Longrightarrow a = \frac{l}{2\sqrt{3}} \times \frac{\sqrt{3}}{\sqrt{3}} \Longrightarrow a = \frac{\sqrt{3} \times l}{6}$$

$$S_{ABC} = p \times a \quad \Longrightarrow S_{ABC} = \frac{3 \times l}{2} \times \frac{\sqrt{3} \times l}{6} \Longrightarrow S_{ABC} = \frac{\sqrt{3} \times l^2}{4}$$

Fonte: Elaborada pelo autor.

Como já sabemos que a área de um triângulo equilátero ABC é dado por $p \times a$, segue que

$$S_{ABC} = p \times a = \frac{3 \times l}{2} \times \frac{\sqrt{3} \times l}{6} \Longrightarrow S_{ABC} = \frac{\sqrt{3} \times l}{4},$$

como queríamos demonstrar.

6.7.2 Atividade 09: determinação da área de um hexágono regular

Objetivo:

Esta atividade tem o objetivo de justificar o fato da área de um hexágono regular $ABCDEF$ de lado l, semiperímetro p e apótema a, ser dada por $S_{ABCDEF} = p \times a$ ou $S_{ABCDEF} = \dfrac{3\sqrt{3} \times l^2}{2}$.

Tempo estimado:

Duas aulas de 50 minutos.

Materiais:

Computador com leitor de PDF; projetor e um arquivo PDF que possua as animações das Figuras 6.13 e 6.14.

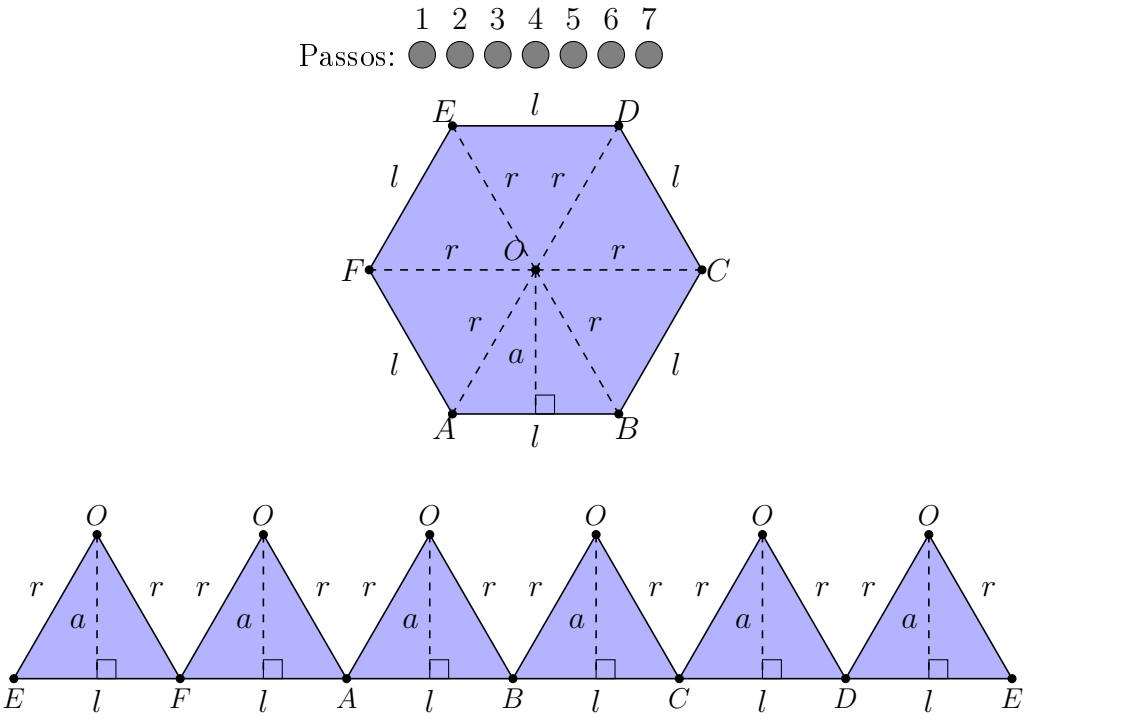
Desenvolvimento:

Apresentar aos estudantes a Proposição 6.7.5. Se julgar necessário, revisar os elementos de um polígono regular apresentando novamente a animação da Figura 6.10.

Proposição 6.7.5. *A área S_{ABCDEF} de um hexágono regular de semiperímetro p e apótema a é dada por $S_{ABCDEF} = p \times a$.*

Após a exposição da Proposição 6.7.5 e das devidas revisões, passar a apresentar a animação da Figura 6.13.

Figura 6.13: Área de um hexágono regular dado seu lado e apótema.



$$S_{ABCDEF} = 6 \times S_{ABO} \implies S_{ABCDEF} = 6 \times \frac{l \times a}{2} \implies S_{ABCDEF} = \frac{6 \times l}{2} \times a \implies S_{ABCDEF} = p \times a$$

Fonte: Elaborada pelo autor.

Simultaneamente à apresentação, explicar que o hexágono regular $ABCDEF$ é particionado pelos seus raios em seis triângulos EFO, FAO, ABO, BCO, CDO e DEO, que são congruentes, pelo caso lado, lado, lado (LLL), e, por esse motivo, possuem áreas iguais. Essa congruência garante que a área do hexágono é igual a seis vezes a área do triângulo ABO, que possui altura igual ao apótema a do hexágono. Assim,

$$S_{ABCDEF} = 6 \times S_{ABO} = 6 \times \frac{l \times a}{2} = \frac{6 \times l}{2} \times a \implies S_{ABCDEF} = p \times a,$$

como queríamos demonstrar.

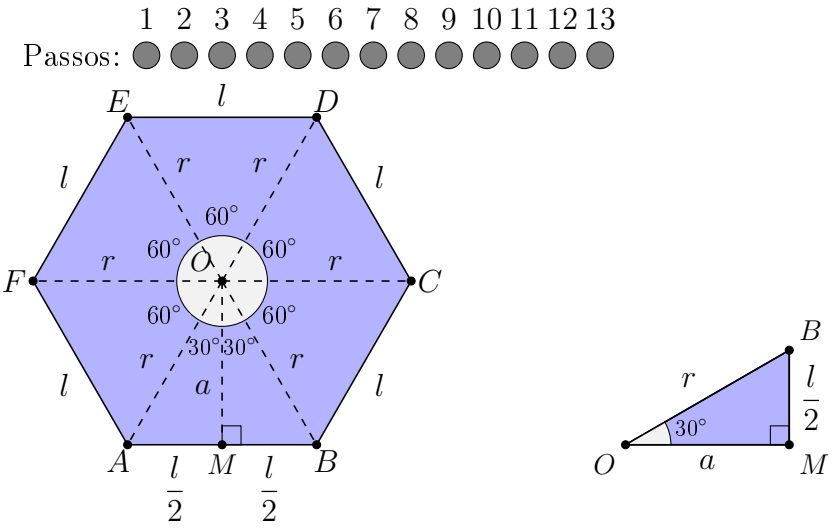
Assim como o triângulo equilátero, o hexágono regular possui uma outra fórmula para se calcular a área, a qual está representada na Proposição 6.7.6, apresente-a aos estudantes e relembre a Proposição 6.7.4, pois ela será utilizada novamente.

Proposição 6.7.6. *A área de um hexágono regular $ABCDEF$ de lado l é dada por*

$$S_{ABCDEF} = \frac{3\sqrt{3} \times l^2}{2}.$$

Após a exposição e discussão da proposição anterior, o professor passa a apresentar, via projetor, a animação da Figura 6.14.

Figura 6.14: Área de um hexágono regular, dado seu lado.



$$tg\,30° = \frac{l/2}{a} \Longrightarrow \frac{\sqrt{3}}{3} = \frac{l/2}{a} \Longrightarrow \sqrt{3} \times a = \frac{3 \times l}{2} \Longrightarrow a = \frac{3 \times l}{2\sqrt{3}} \times \frac{\sqrt{3}}{\sqrt{3}} \Longrightarrow a = \frac{\sqrt{3} \times l}{2}$$

$$S_{ABCDEF} = p \times a \Longrightarrow S_{ABCDEF} = \frac{6 \times l}{2} \times \frac{\sqrt{3} \times l}{2} \Longrightarrow S_{ABCDEF} = \frac{3\sqrt{3} \times l^2}{2}$$

Fonte: Elaborada pelo autor.

Simultaneamente à apresentação, explicar que o hexágono $ABCDEF$ é particionado em seis triângulos congruentes EFO, FAO, ABO, BCO, CDO e DEO. Essa congruência garante que cada ângulo central do hexágono é congruente e com medida de 60°, pois ao somá-los devemos encontrar o ângulo de uma volta, cuja medida é de 360°.

Ao traçarmos o apótema do hexágono, ele representará a altura do triângulo ABO, que é isósceles de base AB e, pela Proposição 6.7.4, essa altura coincide com a bissetriz do ângulo $\angle AOB$ e com a mediana relativa ao lado AB, isto é, o ângulo $\angle MOB$ mede 30° e o segmento MB mede $l/2$, onde M é o pé do apótema do hexágono. Temos com isso o triângulo retângulo MBO donde podemos determinar o valor do apótema a em função do lado l por meio da tangente do ângulo $\angle MOB$,

$$tg\,30° = \frac{l/2}{a} \Longrightarrow \frac{\sqrt{3}}{3} = \frac{l/2}{a} \Longrightarrow a = \frac{\sqrt{3} \times l}{2}.$$

Como já sabemos que a área de um hexágono regular é dada por $p \times a$, segue que

$$S_{ABCDEF} = p \times a = \frac{6 \times l}{2} \times \frac{\sqrt{3} \times l}{2} \Longrightarrow S_{ABCDEF} = \frac{3\sqrt{3} \times l^2}{2},$$

como queríamos demonstrar.

6.8 Área de um Círculo

6.8.1 Atividade 10: determinação da área de um círculo

Objetivo:

Esta atividade tem o objetivo de justificar o fato da área de um círculo de raio r ser dada por $S_c = \pi \times r^2$.

Tempo estimado:

Uma aula de 50 minutos.

Materiais:

Computador com leitor de PDF; projetor e um arquivo PDF que possua as animação da Figura 6.15.

Desenvolvimento:

Apresentar a Proposição 6.8.1 e discutir um pouco a respeito.

Proposição 6.8.1. *A área S_c de um círculo de raio r é dada por $S_c = \pi \times r^2$.*

Após a apresentação e discussão da proposição anterior, passar a apresentar a animação da Figura 6.15 pausando-a antes que apareça as equações. Indagar os estudantes à respeito das seguintes questões:

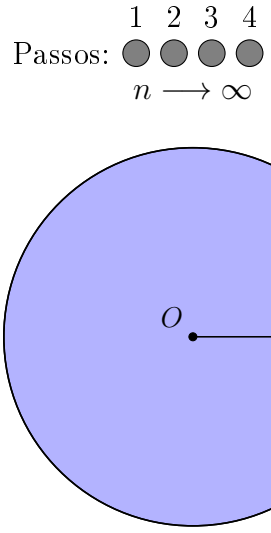
1. O que acontece com o perímetro do polígono regular em relação ao comprimento da circunferência à medida que se aumenta a quantidade n de lados do polígono?

2. O que acontece com o comprimento do apótema do polígono regular em relação ao raio da circunferência à medida que se aumenta a quantidade de lados n do polígono?

3. O que acontece com a área do polígono regular em relação à área do círculo ao passo que se aumenta a quantidade n de lados do polígono?

Se necessário, repetir a animação do início várias vezes para que os alunos reflitam sobre as questões.

Figura 6.15: Área de um hexágono regular dado seu lado.



$$S_p = p \times a \implies S_c = \frac{2 \times \pi \times r}{2} \times r \implies S_c = \pi \times r^2$$

Fonte: Elaborada pelo autor.

As conclusões dos alunos devem ser próximas das seguintes:

1. À medida que se aumenta a quantidade de lados do polígono, o seu perímetro se aproxima do comprimento da circunferência.

2. À medida que se aumenta a quantidade de lados do polígono, o comprimento do apótema se aproxima da medida do raio da circunferência.

3. À medida que se aumenta a quantidade de lados do polígono, a área do polígono se aproxima da área do círculo.

Continuar a animação da Figura 6.15 e destacar que, considerando uma quantidade muito grande de lados para o polígono regular, o semiperímetro do polígono se aproximará do semiperímetro do círculo, o apótema do polígono se aproximará do raio e consequentemente a área do polígono se aproximará da área do círculo, isto é, quando $n \longrightarrow \infty$, e considerando que o comprimento de uma circunferência é igual a $2 \times \pi \times r$, teremos

$$S_p = p \times a \implies S_c = \frac{2 \times \pi \times r}{2} \times r \implies S_c = \pi \times r^2,$$

como queríamos demonstrar.

Capítulo 7

Considerações Finais

Ao realizar nossa pesquisa, constatamos que o ensino de Geometria ainda deixa a desejar em relação às demonstrações dos resultados. Nesse sentido, o referido trabalho apresenta uma ferramenta que enfatiza o uso de animações que buscam facilitar a compreensão dos assuntos abordados, uma vez que as animações atraem, motivam e estimulam a curiosidade dos alunos, relacionando as fórmulas apresentadas com figuras geométricas de maneira dinâmica que, utilizadas em conjunto com as orientações do professor, levam a uma aprendizagem significativa.

A partir dos dados e das discussões apresentadas no Capítulo 2, ao explorarmos como os assuntos de Geometria são apresentados na Base Nacional Comum Curricular (BNCC) e, ao quantificarmos o considerável percentual de questões relacionadas a conhecimentos geométricos abordadas no Exame Nacional do Ensino Médio (Enem), evidenciamos a importância da Geometria no currículo do Ensino Médio, bem como a importância da utilização de ferramentas tecnológicas a serviço do ensino da Matemática.

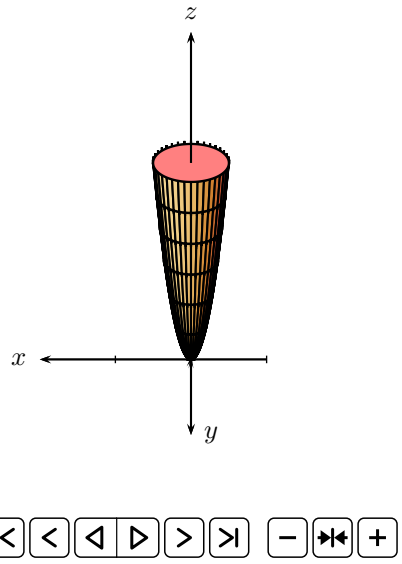
Os conceitos e instruções apresentados nos Capítulos 3, 4 e 5, contribuem para a formação profissional dos professores de Matemática do Ensino Médio e os encorajam a utilizar o LaTeX para a produção de materiais que os auxiliem no desenvolvimento de suas aulas na disciplina de Geometria Plana. Enquanto a proposta de sequência didática, apresentada no Capítulo 6, mostra efetivamente como trabalhar as atividades de forma sistemática, onde o trabalho do professor é facilitado.

Com a realização desta dissertação, compreendemos o potencial do programa LaTeX, entendendo que ele é muito mais do que um simples editor de textos científicos. Enxergamos também a capacidade deste programa de produzir imagens e animações de qualidade, que se mostram proveitosas no ensino de Geometria Plana. Percebemos ainda a escassez de trabalhos acadêmicos nacionais que explorem esta vertente do LaTeX, por meio dos pacotes `TikZ`, `tkz-euclide` e `animate`.

O uso das animações não se restringe aos assuntos apresentados nesta dissertação, elas podem ser utilizadas em outros ramos da Matemática que, no desenvolvimento

da teoria ou na resolução de problemas, envolvam alguma figura. Uma possibilidade é utilizar o pacote `pst-3dplot` em conjunto com o `animate` para construir animações em três dimensões. Veja um exemplo na Figura 7.1, onde é feita a representação de um paraboloide de revolução (o código que origina a Figura 7.1 está disponível na Seção A.3 do Apêndice A).

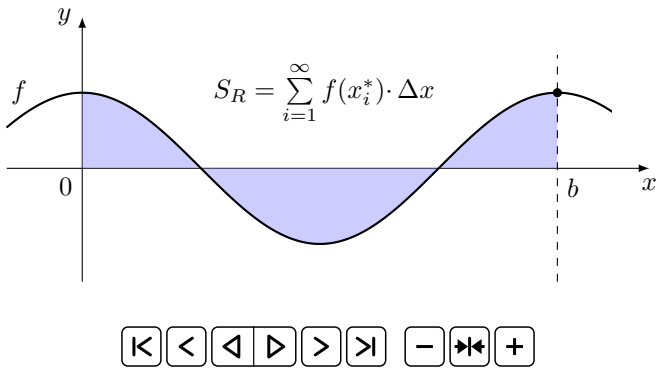
Figura 7.1: Animação com os pacotes `pst-3dplot` e `animate`.



Fonte: Elaborada pelo autor.

Pode-se considerar o uso das animações para auxiliar a compreensão de conceitos que envolvam representações gráficas de funções de variáveis reais ou até mesmo de integrais, como é o caso da Figura 7.2, que ilustra a soma de Riemann no intervalo fechado $[0, b]$ da função $f : \mathbb{R} \longmapsto \mathbb{R}$, cuja lei de formação é $f(x) = \cos(x)$. O código para a obtenção da Figura 7.2, está disponível na Seção A.4 do Apêndice A e para sua construção, utilizamos macros dos pacotes `tkz-euclide` e `tkz-fct` em conjunto com o `animate`.

Figura 7.2: Soma de Riemann com `tkz-euclide`, `tkz-fct` e `animate`.



Fonte: Elaborada pelo autor.

Portanto, esperamos que este trabalho sirva de incentivo à produção de materiais didáticos com uso do LaTeX para auxiliar os professores no ensino de Geometria Plana no Ensino Médio, como também, estimule a busca por informações relacionadas ao tema aqui discutido e a criação de materiais semelhantes aos que expomos nesta dissertação, proporcionando o aumento de referencial teórico para novos trabalhos que auxiliem no ensino e aprendizagem da Matemática.

Referências Bibliográficas

[1] CARVALHO, J. B. P.; ROQUE, T. M. *Tópicos de história da Matemática*. 1. ed. Rio de Janeiro: Sociedade Brasileira de Matemática, 2012.

[2] SILVA, C. M. R. *A modelagem matemática como estratégia para o ensino de Geometria: uma proposta para eletiva de exatas*. Dissertação (Mestrado Profissional em Matemática - PROFMAT) — Universidade Estadual do Sudeste da Bahia, Vitória da Conquista , Bahia, Brasil, 2021.

[3] GEOGEBRA. *GeoGebra - Aplicativos Matemáticos*. Disponível em: <https://www.geogebra.org>. Acesso em: 29 de out. de 2022.

[4] NEVES, A. *Geometria com tkz-euclide: O Problema do Haberdasher (Henry Dudeney)*. 2021. Disponível em: <https://www.youtube.com/watch?v=b2LSpci4jqs&t=1200s>. Acesso em: 14 de jun. de 2022.

[5] BRASIL, M. d. E. *Base Nacional Comum Curricular*. 2018. Disponível em: <http://basenacionalcomum.mec.gov.br/>. Acesso em: 02 de fev. de 2021.

[6] BRASIL, I. N. E. P. *Matriz de Referência do ENEM*. 2020. Disponível em: <https://download.inep.gov.br/download/enem/matriz_referencia.pdf>. Acesso em: 02 de fev. de 2021.

[7] BRASIL, M. d. E. *Histórico da Base Nacional Comum Curricular*. 2020. Disponível em: <http://basenacionalcomum.mec.gov.br/historico>. Acesso em: 02 de fev. de 2021.

[8] BRASIL, M. d. E. *Exame Nacional do Ensino Médio: Apresentação*. 2020. Disponível em: <https://www.gov.br/inep/pt-br/areas-de-atuacao/avaliacao-e-exames-educacionais/enem>. Acesso em: 02 de fev. de 2021.

[9] SIQUEIRA, V. F. de. *Tópicos de Geometria Plana em provas do Enem*. Dissertação (Mestrado Profissional em Matemática - PROFMAT) — Universidade Federal do Cariri, Juazeiro do Norte, Ceará, Brasil, 2020.

[10] SANTOS, A. R. S.; VIGLIONI, H. H. d. B. Geometria euclidiana plana. *Aracaju: UFS*, 2011.

[11] ÁVILA, G. Euclides, geometria e fundamentos. *Revista do professor de matemática*, v. 45, 2001.

[12] VIEIRA, C. A. do N. *Sobre Geometrias Não-Euclidianas*. Dissertação (Mestrado Profissional em Matemática - PROFMAT) — Universidade Federal do Cariri, Juazeiro do Norte, Ceará, Brasil, 2018.

[13] NETO, A. C. M. *Geometria, Coleção PROFMAT*. 1. ed. Rio de Janeiro: Sociedade Brasileira de Matemática, 2013.

[14] NETO, A. C. M. *Tópicos de Matemática Elementar: volume 2 Geometria Euclidiana Plana*. 2. ed. Rio de Janeiro: Sociedade Brasileira de Matemática, 2013.

[15] BARBOSA, J. L. M. *Geometria Eucludiana Plana*. 11. ed. Rio de Janeiro: Sociedade Brasileira de Matemática, 2012.

[16] WAGNER, E.; CARNEIRO, J. P. Q. *Construções geométricas*. 6. ed. Rio de Janeiro: Sociedade Brasileira de Matemática, 2007.

[17] LIMA, E. L. *Isometrias*. 2. ed. Rio de Janeiro: Sociedade Brasileira de Matemática, 2007.

[18] LOURENÇO, B. C. G. *Apostila LaTeX: do básico ao avançado*. 2021. Disponível em: <https://petmecanica.ufes.br/latex>. Acesso em: 19 de mar. de 2022.

[19] OETIKER, T. et al. *Uma não tão pequena introdução ao LaTeX2ε*. 2011. Disponível em: <https://mirror.las.iastate.edu/tex-archive/info/lshort/portuguese/pt-lshort-a5.pdf>. Acesso em: 19 de mar. de 2022.

[20] GRAHN, A. *The animate Package*. 2022. Disponível em: <http://mirrors.ctan.org/macros/latex/contrib/animate/animate.pdf>. Acesso em: 10 de fev. de 2022.

[21] MIKTEX. *Getting MiKTeX*. 2022. Disponível em: <https://miktex.org/download>. Acesso em: 22 de mar. de 2022.

[22] TEXLIVE. *TeX Live: All the ways to acquire TeX Live*. 2022. Disponível em: <https://www.tug.org/texlive/>. Acesso em: 22 de mar. de 2022.

[23] MACTEX. *The MacTeX-2021 Distribution*. 2021. Disponível em: <https://tug.org/mactex/>. Acesso em: 22 de mar. de 2022.

[24] TEXSTUDIO. *Welcome to TeXstudio*. 2022. Disponível em: <https://www.texstudio.org/>. Acesso em: 27 de mar. de 2022.

[25] TEXMAKER. *TEXMAKER: Free cross-platform LaTeX editor since 2003*. 2022. Disponível em: <https://www.xm1math.net/texmaker/download.html>. Acesso em: 27 de mar. de 2022.

[26] TEXWORKS. *TeXworks: lowering the entry barrier to the TeX world*. 2022. Disponível em: <https://tug.org/texworks/>. Acesso em: 27 de mar. de 2022.

[27] ADOBE. *Etapa 1 de 3: Baixar o software*. 2022. Disponível em: <https://get.adobe.com/br/reader/>. Acesso em: 27 de mar. de 2022.

[28] TANTAU, T. *The Tikz and PGF Packages: Manual for version 3.1.9a*. 2021. Disponível em: <https://pgf-tikz.github.io/pgf/pgfmanual.pdf>. Acesso em: 03 de jan. de 2022.

[29] MATTHES, A. *AlterMundus: tkz-euclide tool for Euclidean Geometry V3.06c*. 2020. Disponível em: <https://altermundus.fr/files/euclide/TKZdoc-euclide.pdf>. Acesso em: 25 de dez. de 2021.

[30] MATTHES, A. *AlterMundus: tkz-base V4.00b*. 2022. Disponível em: <http://altermundus.fr/files/tkz-base/tkz-base.pdf>. Acesso em: 25 de dez. de 2021.

[31] ALMEIDA, D. M.; LEITE, A. J. M. J.; MURAKAMI, L. C. *O emprego da técnica de animação por recorte digital para a criação de personagens de jogos digitais*. Teresina: SBC – Proceedings of SBGames: XIV SBGames, 2015. 526–535 p. ISSN: 2179-2259. Disponível em: <http://www.sbgames.org/sbgames2015/anaispdf/artesedesign-full/147538.pdf>. Acesso em: 12 de fev. de 2022.

[32] WAREHOUSE, M. *Pythagorean Theorem: How to Use The Pythagorean Theorem*. Disponível em: <https://www.mathwarehouse.com/geometry/triangles/how-to-use-the-pythagorean-theorem.php>. Acesso em: 21 de fev. de 2022.

[33] NEVES, A. *Construção e animação de figuras: tkz-euclide e animate parte 2*. 2020. Disponível em: <https://www.youtube.com/watch?v=1Pd1G10pDrI>. Acesso em: 01 de mar. de 2022.

[34] LIMA, E. L. *Medida e Forma em Geometria: Comprimento, Área, Volume e Semelhança*. 4. ed. Rio de Janeiro: Sociedade Brasileira de Matemática, 2011.

Apêndice A

Códigos de Algumas Figuras

A.1 Código da Figura 1.2

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\usepackage{animate}
\def\xmin{-3.375}
\def\xmax{6.5}
\def\ymin{-2.7}
\def\ymax{4.8}
\definecolor[]{azul}{rgb}{0.36,0.54,0.66}
\definecolor[]{cinza}{rgb}{0.83,0.83,0.83}

\begin{document}
  \begin{animateinline}[poster=last,controls={all},scale=1]{24}
  \multiframe{120}{ra=0+1/119}{
    \begin{tikzpicture}
      \path[use as bounding box] (\xmin,\ymin) rectangle (\xmax,\ymax);
      \tkzDefPoints{0/0/a,3/0/b,-3.5/-2.5/A,6.75/5/B,-0.55/-2.8/1,
      3.65/-2.8/9}
      \tkzDrawTriangle[thick,two angles=60 and 60](a,b)\tkzGetPoint{c}
      \tkzDefMidPoint(a,c)\tkzGetPoint{d}
      \tkzDefMidPoint(b,c)\tkzGetPoint{e}
      \tkzDefPointBy[projection=onto a--b](d)\tkzGetPoint{f}
      \tkzDefPointBy[projection=onto a--b](e)\tkzGetPoint{g}
      \tkzDefPointBy[projection=onto e--f](d)\tkzGetPoint{h}
      \tkzDefPointBy[projection=onto e--f](g)\tkzGetPoint{i}
      \tkzDefPointsBy[rotation=center e angle 180*\ra](i,g,b,f,h,d,
```

```
      a){i',g',b',f',h',d',a'}
      \tkzDefPointsBy[rotation=center g' angle 180*\ra](i',f',h',d',
      a'){i'',f'',h'',d'',a''}
      \tkzDefPointsBy[rotation=center f'' angle 180*\ra](h'',d'',
      a''){h''',d''',a'''}
      \tkzDrawSegments[thick](f,e g,i d,h)
      \tkzDrawPolygon[thick](h''',d''',a''',f'')
      \tkzDrawPolygon[thick](e,i',g',b')
      \tkzDrawPolygon[thick](g',i'',f'')
      \pgfmathtruncatemacro\cor{90*\ra}
      \tkzFillPolygon[fill=cinza!\cor!azul!](h,e,c,d)
      \tkzDrawPolygon[thick](h,e,c,d)
      \tkzFillPolygon[fill=cinza!\cor!azul!](h''',d''',a''',f'')
      \tkzFillPolygon[fill=cinza!\cor!azul!](e,i',g',b')
      \tkzFillPolygon[fill=cinza!\cor!azul!](g',i'',f'')
      \tkzDrawPolygon[thick](h''',d''',a''',f'')
      \tkzDrawPolygon[thick](e,i',g',b')
      \tkzDrawPolygon[thick](g',i'',f'')
      \tkzDrawSegments[thick](h,e h,d)
    \end{tikzpicture}}
  \end{animateinline}
\end{document}
```

A.2 Código da Figura 5.71

```
\documentclass{standalone}
\usepackage{tkz-euclide}
\usepackage{animate}
\tikzstyle{every picture} = [remember picture]
\def\xmax{3.5}
\def\ymax{2.5}
\def\xmin{-1.5}
\def\ymin{-2.5}

\begin{document}
  \begin{animateinline}[poster=first, controls, palindrome]{12}
  \multiframe{4}{Ri=0+0.1}{
    \begin{tikzpicture}
      \path[use as bounding box] (\xmax,\ymax) rectangle (\xmin,\ymin);
```

```
    \tkzDefPoints{0/0/A, 2/0/B}
    \tkzDrawTriangle[two angles=30 and 60](A,B)\tkzGetPoint{C}
    \tkzMarkRightAngle[size=0.2](B,C,A)
    \tkzDrawSquare(B,A)\tkzGetPoints{M}{N}
    \tkzDrawSquare[fill=yellow](C,B)\tkzGetPoints{O}{P}
    \tkzDrawSquare(A,C)\tkzGetPoints{D}{E}
    \tkzInterLL(A,D)(C,E)\tkzGetPoint{F}
    \tkzDefParallelogram(B,A,F)\tkzGetPoint{G}
    \tkzInterLL(A,E)(F,G)\tkzGetPoint{H}
    \tkzInterLL(C,D)(F,G)\tkzGetPoint{I}
    \tkzDefSquare(H,F)\tkzGetFirstPoint{J}
    \tkzInterLL(E,D)(F,J)\tkzGetPoint{K}
    \tkzInterLL(A,C)(F,J)\tkzGetPoint{L}
    \tkzDrawPolygon[fill=gray!50](E,H,F,K)
    \tkzDrawPolygon[fill=blue!50](A,L,F,H)
    \tkzDrawPolygon[fill=red!50](L,C,I,F)
    \tkzDrawPolygon[fill=teal!50](F,I,D,K)
  \end{tikzpicture}}
\newframe
\multiframe{11}{Ra=0+0.1}{
  \begin{tikzpicture}
    \path[use as bounding box] (\xmax,\ymax) rectangle (\xmin,\ymin);
    \tkzDrawTriangle[two angles=30 and 60](A,B)\tkzGetPoint{C}
    \tkzMarkRightAngle[size=0.2](B,C,A)
    \tkzDrawSquare(B,A)\tkzGetPoints{M}{N}
    \tkzDrawSquare[fill=yellow](C,B)\tkzGetPoints{O}{P}
    \tkzDrawSquare(A,C)\tkzGetPoints{D}{E}
    \tkzInterLL(A,D)(C,E)\tkzGetPoint{F}
    \tkzDefParallelogram(B,A,F)\tkzGetPoint{G}
    \tkzInterLL(A,E)(F,G)\tkzGetPoint{H}
    \tkzInterLL(C,D)(F,G)\tkzGetPoint{I}
    \tkzDefSquare(H,F)\tkzGetFirstPoint{J}
    \tkzInterLL(E,D)(F,J)\tkzGetPoint{K}
    \tkzInterLL(A,C)(F,J)\tkzGetPoint{L}
    \tkzDrawPolygon[fill=blue!50](A,L,F,H)
    \tkzDrawPolygon[fill=red!50](L,C,I,F)
    \tkzDrawPolygon[fill=teal!50](F,I,D,K)
    \tkzDefPointWith[linear,K=\Ra](F,N)\tkzGetPoint{F'}
    \tkzDefPointsBy[translation=from F to F'](K,E,H){}
```

```
    \tkzDrawPolygon[fill=gray!50](E',H',F',K')
  \end{tikzpicture}}
\newframe
\multiframe{11}{Rb=0+0.1}{
  \begin{tikzpicture}
    \path[use as bounding box] (\xmax,\ymax) rectangle (\xmin,\ymin);
    \tkzDrawTriangle[two angles=30 and 60](A,B)\tkzGetPoint{C}
    \tkzMarkRightAngle[size=0.2](B,C,A)
    \tkzDrawSquare(B,A)\tkzGetPoints{M}{N}
    \tkzDrawSquare[fill=yellow](C,B)\tkzGetPoints{O}{P}
    \tkzDrawSquare(A,C)\tkzGetPoints{D}{E}
    \tkzDrawPolygon[fill=red!50](L,C,I,F)
    \tkzDrawPolygon[fill=teal!50](F,I,D,K)
    \tkzDefPointWith[linear,K=\Rb](F,B)\tkzGetPoint{F''}
    \tkzDefPointsBy[translation=from F to F''](A,L,H){A'',L'',H''}
    \tkzDrawPolygon[fill=blue!50](A'',L'',F'',H'')
    \tkzDrawPolygon[fill=gray!50](E',H',F',K')
  \end{tikzpicture}}
\newframe
\multiframe{11}{Rc=0+0.1}{
  \begin{tikzpicture}
    \path[use as bounding box] (\xmax,\ymax) rectangle (\xmin,\ymin);
    \tkzDrawTriangle[two angles=30 and 60](A,B)\tkzGetPoint{C}
    \tkzMarkRightAngle[size=0.2](B,C,A)
    \tkzDrawSquare(B,A)\tkzGetPoints{M}{N}
    \tkzDrawSquare[fill=yellow](C,B)\tkzGetPoints{O}{P}
    \tkzDrawSquare(A,C)\tkzGetPoints{D}{E}
    \tkzDrawPolygon[fill=red!50](L,C,I,F)
    \tkzDefPointWith[linear,K=\Rc](F,M)\tkzGetPoint{F'''}
    \tkzDefPointsBy[translation=from F to F'''](I,D,K){I''',D''',
    K'''}
    \tkzDrawPolygon[fill=blue!50](A'',L'',F'',H'')
    \tkzDrawPolygon[fill=gray!50](E',H',F',K')
    \tkzDrawPolygon[fill=teal!50](F''',I''',D''',K''')
  \end{tikzpicture}}
\newframe
\multiframe{11}{Rd=0+0.1}{
  \begin{tikzpicture}
    \path[use as bounding box] (\xmax,\ymax) rectangle (\xmin,\ymin);
```

```
    \tkzDrawTriangle[two angles=30 and 60](A,B)\tkzGetPoint{C}
    \tkzMarkRightAngle[size=0.2](B,C,A)
    \tkzDrawSquare(B,A)\tkzGetPoints{M}{N}
    \tkzDrawSquare[fill=yellow](C,B)\tkzGetPoints{O}{P}
    \tkzDrawSquare(A,C)\tkzGetPoints{D}{E}
    \tkzDefPointWith[linear,K=\Rd](F,A)\tkzGetPoint{F''''}
    \tkzDefPointsBy[translation=from F to F''''](L,C,I){L'''',
    C'''',I''''}
    \tkzDrawPolygon[fill=blue!50](A'',L'',F'',H'')
    \tkzDrawPolygon[fill=gray!50](E',H',F',K')
    \tkzDrawPolygon[fill=teal!50](F''',I''',D''',K''')
    \tkzDrawPolygon[fill=red!50](L'''',C'''',I'''',F'''')
  \end{tikzpicture}}
\newframe
\multiframe{11}{Re=0+0.1}{
  \begin{tikzpicture}
    \path[use as bounding box] (\xmax,\ymax) rectangle (\xmin,\ymin);
    \tkzDrawTriangle[two angles=30 and 60](A,B)\tkzGetPoint{C}
    \tkzMarkRightAngle[size=0.2](B,C,A)
    \tkzDrawSquare(B,A)\tkzGetPoints{M}{N}
    \tkzDrawSquare(C,B)\tkzGetPoints{O}{P}
    \tkzDrawSquare(A,C)\tkzGetPoints{D}{E}
    \tkzDefPointWith[linear,K=\Re](P,C'''')\tkzGetPoint{P'''''}
    \tkzDefPointsBy[translation=from P to P'''''](C,B,O){C''''',
    B''''',O'''''}
    \tkzDrawPolygon[fill=blue!50](A'',L'',F'',H'')
    \tkzDrawPolygon[fill=gray!50](E',H',F',K')
    \tkzDrawPolygon[fill=teal!50](F''',I''',D''',K''')
    \tkzDrawPolygon[fill=red!50](L'''',C'''',I'''',F'''')
    \tkzDrawPolygon[fill=yellow](C''''',B''''',O''''',P''''')
  \end{tikzpicture}}
\newframe
\multiframe{4}{Rf=0+0.1}{
  \begin{tikzpicture}
    \path[use as bounding box] (\xmax,\ymax) rectangle (\xmin,\ymin);
    \tkzDrawTriangle[two angles=30 and 60](A,B)\tkzGetPoint{C}
    \tkzMarkRightAngle[size=0.2](B,C,A)
    \tkzDrawSquare(B,A)\tkzGetPoints{M}{N}
    \tkzDrawSquare(C,B)\tkzGetPoints{O}{P}
```

```
      \tkzDrawSquare(A,C)\tkzGetPoints{D}{E}
      \tkzDrawPolygon[fill=blue!50](A'',L'',F'',H'')
      \tkzDrawPolygon[fill=gray!50](E',H',F',K')
      \tkzDrawPolygon[fill=teal!50](F''',I''',D''',K''')
      \tkzDrawPolygon[fill=red!50](L'''',C'''',I'''',F'''')
      \tkzDrawPolygon[fill=yellow](C'''''',B'''''',O'''''',P'''''')
    \end{tikzpicture}}
  \end{animateinline}
\end{document}
```

A.3 Código da Figura 7.1

Para que o código funcione corretamente é necessário configurar o editor TEX para utilizar o compilador *LaTeX*. Para realizar esta configuração no TeXstudio, vá em "Opções" e depois em "Configurar o TeXstudio...". Na janela que aparecer, clique em "Compilação". Na opção "Compilador predefinido", aparecerá quatro tipos de compiladores: *LaTeX*, *PdfLaTeX*, *LuaLaTeX* e *XeLaTeX*. Escolha "LaTeX" e depois confirme a configuração clicando em "Aceitar".

```
\documentclass{standalone}
\usepackage{animate}
\usepackage{amsmath,pst-all,amsfonts,graphicx,pstricks,graphics,color}
\usepackage{amssymb}
\usepackage{pst-3dplot}

\begin{document}
  \begin{animateinline}[poster=last,palindrome,controls,scale=1]{7}
  \multiframe{31}{ra=0+12,rb=0.5+0.04}{
    \begin{pspicture}(-3.2,-2)(3.2,5.2)
      \psset{Beta=15}
      \psset{Alpha=\ra}
      \psset{Beta=30}
      \pstThreeDCoor[linecolor=black,xMax=2.5,yMax=2.5,zMin=0,zMax=5,
      IIIDticks]
      \pstThreeDLine[]{->}(-1,0,0)(2.5,0,0)
      \pstThreeDLine[]{->}(0,-1,0)(0,2.5,0)
      \pstParaboloid{3}{\rb}
      \pstThreeDLine[]{->}(0,0,3)(0,0,5)
      \pstThreeDDot(0,0,3)
```

```
    \end{pspicture}}
  \end{animateinline}
\end{document}
```

A.4 Código da Figura 7.2

Para utilizar todas as funcionalidades do pacote `tkz-fct` é necessário instalar o programa Gnuplot que pode ser obtido gratuitamente no endereço: http://www.gnuplot.info/download.html.

Após a instalação do Gnuplot, é necessário configurar o editor TEX para que ele o utilize para plotagem. Para configurar o TeXstudio, vá em "Opções" e depois em "Configurar o TeXstudio...". Na janela que aparecer, clique em "Comandos". Na opção "PdfLaTeX", digite o seguinte texto: `pdflatex.exe -synctex=1 -shell-escape -enable-write18 -interaction=nonstopmode`. Após isso, confirme a configuração clicando em "Aceitar".

É importante ressaltar que o TeXstudio deve estar configurado para utilizar o compilador PdfLaTeX. Para isso, siga passos semelhantes aos utilizados na Seção A.3.

```
\documentclass{standalone}
\usepackage{animate}
\usepackage{tikz}
\usepackage{tkz-euclide}
\usepackage{tkz-fct}

\begin{document}
  \begin{animateinline}[poster=last,controls=all]{12}
  \multiframe{18}{rb=0+1/17}{
    \begin{tikzpicture}
      \path[use as bounding box] (-1.5,-2) rectangle (8,2.25);
      \tkzDefPoints{1/0/B,0/1/D,1.6/1/P,0/0/O}
      \tkzInit[xmin=-1,xmax=7,ymin=-1.5,ymax=1.5]
      \tkzDrawXY[noticks]
      \tkzFct[thick,domain=-1:-1+4*pi*\rb]{cos(x)}
      \tkzLabelPoint[below left](O){$0$}
      \tkzLabelPoint[left=0.6cm,opacity=\rb](D){$f$}
    \end{tikzpicture}}
  \newframe
  \multiframe{18}{rb=0+1/17}{
```

```
    \begin{tikzpicture}
      \path[use as bounding box] (-1.5,-2) rectangle (8,2.25);
      \tkzDefPoints{2*pi*\rb/0/C}
      \tkzInit[xmin=-1,xmax=7,ymin=-1.5,ymax=1.5]
      \tkzDrawXY[noticks]
      \tkzFct[thick,domain=-1:2.5*pi]{cos(x)}
      \tkzDrawRiemannSumMid[fill=blue!40,opacity=0.5,
      interval=0:0+2*pi*\rb,number=6]
      \tkzDefPointByFct[draw,with=a](0+2*pi*\rb)\tkzGetPoint{A}
      \tkzLabelPoint[right,opacity=\rb](P){$S_R=\sum\limits_{i=1}^{6}
      f(x_i^*)\cdotp\Delta x$}
      \tkzVLine[dashed]{2*pi*\rb}
      \tkzLabelPoint[below right](C){$b$}
      \tkzLabelPoint[below left](O){$0$}
      \tkzLabelPoint[left=0.6cm](D){$f$}
    \end{tikzpicture}}
  \newframe[2]
  \multiframe{55}{ic=6+1}{
    \begin{tikzpicture}
      \path[use as bounding box] (-1.5,-2) rectangle (8,2.25);
      \tkzInit[xmin=-1,xmax=7,ymin=-1.5,ymax=1.5]
      \tkzDrawXY[noticks]
      \tkzFct[domain=-1:2.5*pi]{cos(x)}
      \tkzDrawRiemannSumMid[fill=blue!40,opacity=0.5,interval=0:2*pi,
      number=\ic]
      \tkzLabelPoint[right](P){$S_R=\sum\limits_{i=1}^{\ic}f(x_i^*)
      \cdotp\Delta x$}
      \tkzFct[thick,domain=-1:2.5*pi]{cos(x)}
      \tkzDrawPoint(A)
      \tkzLabelPoint[below right](C){$b$}
      \tkzLabelPoint[below left](O){$0$}
      \tkzVLine[dashed]{2*pi}
      \tkzLabelPoint[left=0.6cm](D){$f$}
    \end{tikzpicture}}
  \end{animateinline}
\end{document}
```