INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO
CAMPUS SÃO PAULO

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

GUSTAVO HENRIQUE CORREIA LOPES
GUSTAVO HENRIQUE MANOEL DINIZ DE OLIVEIRA
RONALDO ROSSETTI DEARO

PROJETO DE SPOMVAL - CISALHAMENTO EM TRANSFORMAÇÕES LINEARES

SÃO PAULO
2025

GUSTAVO HENRIQUE CORREIA LOPES
GUSTAVO HENRIQUE MANOEL DINIZ DE OLIVEIRA
RONALDO ROSSETTI DEARO

PROJETO DE SPOMVAL - CISALHAMENTO EM TRANSFORMAÇÕES LINEARES

Trabalho apresentado ao Programa do Curso de Bacharelado em Sistemas de Informação, do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Campus de São Paulo, como requisito parcial para aprovação na disciplina SPOMVAL – Vetores, Geometria Analítica e Álgebra Linear, sob orientação da Professora Josceli Maria Tenorio.

SÃO PAULO
2025

Resumo

Este trabalho apresenta um estudo sobre a transformação linear de cisalhamento aplicada a objetos geométricos em duas e três dimensões. A partir da modelagem de um quadrado e de um cubo, definiram-se seus vértices em coordenadas cartesianas e aplicaram-se diferentes matrizes de cisalhamento para analisar os efeitos produzidos em cada caso. A implementação foi realizada em Python, permitindo observar de forma prática como a transformação altera a geometria original, preservando paralelismo e estrutura geral. Os resultados obtidos demonstram a coerência entre os fundamentos teóricos da Álgebra Linear e sua aplicação computacional, evidenciando a importância das transformações lineares em contextos de Sistemas de Informação, especialmente na representação e manipulação de formas geométricas.

Palavras-chaves: Transformações Lineares; Cisalhamento; Álgebra Linear; Computação Gráfica; Python; Modelagem Geométrica.

Sumário

1. Introdução

Transformações lineares são ferramentas fundamentais na modelagem matemática de problemas em Sistemas de Informação, Computação Gráfica e diversas áreas da tecnologia. Entre essas transformações, o cisalhamento (shear) se destaca por alterar a forma de objetos sem modificar áreas ou volumes, permitindo análises geométricas importantes e oferecendo aplicações práticas em animação, visão computacional, simulações e processamento gráfico.

Este trabalho investiga o comportamento do cisalhamento em objetos geométricos bidimensionais e tridimensionais, utilizando como base modelos simples, um quadrado no plano e um cubo no espaço. O estudo aplica fatores de cisalhamento horizontal e vertical variando entre 0.1 e 2.0, analisando como pequenas ou grandes deformações afetam a geometria original, destacando o caráter composicional das matrizes e a importância da Álgebra Linear na construção de pipelines gráficos.

O objetivo central é compreender, de forma prática e visual, como as matrizes de cisalhamento atuam sobre cada ponto dos objetos, reforçando a ligação entre teoria matemática e implementação computacional. A abordagem inclui modelagem dos objetos, aplicação das transformações, visualização gráfica antes e depois do cisalhamento e discussão dos efeitos observados. O trabalho também evidencia a utilidade dessas operações em contextos de TI, onde representações geométricas e manipulação de estruturas espaciais são essenciais.

2. Objetivos

O objetivo principal deste trabalho consiste em analisar e demonstrar, de forma prática e visual, os efeitos da transformação linear de cisalhamento (shear) em objetos geométricos bidimensionais e tridimensionais, relacionando teoria matemática desenvolvida na disciplina de Vetores, Geometria Analítica e Álgebra Linear e sua implementação computacional.

Para alcançar o objetivo geral proposto, foram definidos os seguintes objetivos específicos:

- Modelar objetos geométricos simples (quadrado e cubo) utilizando coordenadas vetoriais.
- Implementar a transformação de cisalhamento horizontal e vertical com fator variável entre 0.1 e 2.0.
- Avaliar como o cisalhamento altera a geometria original dos objetos, observando deslocamentos, deformações e preservação de paralelismo.
- Representar graficamente os objetos antes e depois da aplicação do cisalhamento, facilitando a interpretação visual da transformação.
- Discutir a composição do cisalhamento com outras transformações lineares, destacando o papel das matrizes na construção de transformações mais complexas.
- Demonstrar a relevância prática do cisalhamento em áreas de TI, como computação gráfica, animação, CAD e processamento geométrico.
- Registrar os procedimentos, resultados e interpretações em um relatório técnico claro e fundamentado.

3. Metodologia

A metodologia adotada neste trabalho integrou fundamentos teóricos de transformações lineares com sua implementação prática em Python. Inicialmente, foram revisados os conceitos matemáticos relacionados ao cisalhamento em 2D e 3D, bem como suas matrizes correspondentes. Em seguida, modelaram-se um quadrado e um cubo por meio de scripts que definem seus vértices em coordenadas cartesianas.

Com os objetos definidos, aplicaram-se diferentes matrizes de cisalhamento aos conjuntos de pontos, realizando testes específicos para cada caso: cisalhamento horizontal e vertical no quadrado e cisalhamento no eixo X para o cubo. As transformações foram realizadas por multiplicação matricial ponto a ponto, permitindo observar de forma clara as deformações produzidas.

Por fim, os resultados foram analisados comparando a geometria original e transformada, verificando o comportamento esperado das matrizes de cisalhamento e registrando os efeitos observados. Essa abordagem garantiu a conexão direta entre teoria matemática e aplicação computacional.

4. Desenvolvimento Teórico

4.1. Cisalhamento

O cisalhamento é uma transformação linear que desloca pontos de um objeto em uma direção proporcional à sua posição em outra direção. Diferente de rotações ou escalas, o cisalhamento deforma a figura: linhas paralelas permanecem paralelas, mas ângulos e formas originais são alterados. Apesar da deformação, a transformação mantém propriedades essenciais da linearidade, como preservação da origem e operação por meio de multiplicação matricial.

Em duas dimensões, o cisalhamento pode ocorrer de duas formas principais: horizontal e vertical. No cisalhamento horizontal, a coordenada x sofre um deslocamento proporcional ao valor de y, enquanto no cisalhamento vertical ocorre o inverso — y é deslocado de acordo com x. As matrizes que representam essas transformações são:

Cisalhamento horizontal:

$$\begin{bmatrix} 1 & \boldsymbol{k} \\ 0 & 1 \end{bmatrix}$$

Figura 1: Cisalhamento horizontal

Cisalhamento vertical:

$$\begin{bmatrix} 1 & 0 \\ \boldsymbol{k} & 1 \end{bmatrix}$$

Figura 2: Cisalhamento vertical

O parâmetro *k* define a intensidade do cisalhamento: valores próximos de zero produzem deformações leves; valores maiores provocam distorções mais evidentes. Como a transformação é linear, cada ponto do objeto é modificado pela multiplicação da matriz pelo vetor posição (x,y)(x, y)(x,y), deslocando a geometria de forma proporcional e previsível.

Em três dimensões, o princípio permanece o mesmo, porém com mais combinações possíveis. O cisalhamento pode ocorrer ao longo de qualquer eixo, influenciado pelos outros dois. Um exemplo simples é o cisalhamento no eixo *x* proporcional a *y*:

$$\begin{bmatrix} 1 & k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figura 3: Cisalhamento no eixo *x* proporcional a *y*

Assim como nas versões 2D, o resultado é uma deformação linear que preserva paralelismo, mas altera ângulos e proporções.

O cisalhamento é amplamente usado em computação gráfica, animações, engines 3D e sistemas de visualização exatamente por causa dessa característica: ele permite deformações controladas sem perder coerência geométrica. Em muitos pipelines gráficos, combina-se cisalhamento com rotação, escala e translação para gerar efeitos mais complexos. Em termos algébricos, isso é possível porque transformações lineares podem ser compostas via multiplicação matricial, o que mantém o modelo matemático simples e previsível.

No contexto de Álgebra Linear, o cisalhamento demonstra de maneira clara como as matrizes representam transformações geométricas e como a estrutura vetorial é manipulada para alterar formas preservando algumas propriedades fundamentais. Ele reforça a ligação entre conceitos abstratos como os espaços vetoriais, matrizes e linearidade e suas aplicações diretas em Sistemas de Informação.

4.2. Aplicações

Um caso representativo do uso do cisalhamento em Tecnologia da Informação ocorre na área de computação gráfica e modelagem geométrica, especialmente em sistemas que necessitam manipular objetos visuais de forma controlada, mantendo coerência estrutural. Uma aplicação frequente é o efeito de inclinação (skew) empregado em interfaces gráficas, animações e softwares de edição.

Nesse contexto, o cisalhamento é utilizado para produzir uma deformação linear que altera a orientação aparente de um objeto sem recorrer a rotações completas. A transformação desloca cada ponto do objeto de acordo com sua posição relativa, preservando paralelismo entre linhas e mantendo a estrutura geral da figura. Por exemplo, ao aplicar um cisalhamento horizontal a um elemento bidimensional — como um retângulo representando um botão ou painel — a coordenada x de cada ponto é modificada proporcionalmente ao seu valor de y. O resultado visual é uma inclinação controlada do objeto, frequentemente utilizada para criar efeitos dinâmicos, simular perspectiva simplificada ou transmitir sensação de movimento.

Esse tipo de transformação é expressa pela matriz:

$$\begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$$

Figura 4: Cisalhamento Horizontal

em que *k* representa o fator de cisalhamento. O modelo matemático permite prever o comportamento geométrico do objeto após a transformação e facilita sua integração a outras operações lineares, como rotação e escala.

O uso do cisalhamento nesse tipo de aplicação destaca sua relevância como ferramenta de manipulação geométrica em Sistemas de Informação, mostrando como conceitos de Álgebra Linear são incorporados a interfaces, mecanismos de renderização e motores gráficos para produzir efeitos visuais consistentes e computacionalmente eficientes.

4.3. Tecnologias

Para a implementação computacional das transformações lineares estudadas, utilizou-se a linguagem Python, devido à sua ampla adoção em contextos acadêmicos e à disponibilidade de bibliotecas voltadas à manipulação numérica e visualização de dados. Embora o código do projeto seja estruturado de forma a realizar operações matriciais manualmente, o ambiente Python oferece suporte a esse tipo de modelagem através de ferramentas consolidadas, como NumPy para operações vetoriais e Matplotlib para representação gráfica.

Neste contexto a biblioteca NumPy foi empregada para manipulação de vetores e matrizes, fornecendo operações numéricas eficientes e estruturas adequadas para representar os pontos e transformações aplicados aos objetos geométricos. Seu uso permitiu realizar multiplicações matriciais de forma precisa e organizada, garantindo a fidelidade dos resultados obtidos.

A biblioteca Matplotlib foi utilizada para a representação gráfica dos objetos antes e depois das transformações. No caso do quadrado, recorreu-se aos recursos de visualização bidimensional, enquanto para o cubo foi utilizado o módulo *mplot3d*, que possibilita a construção de gráficos tridimensionais. Essa visualização foi essencial para interpretar os efeitos do cisalhamento e validar o comportamento teórico das matrizes aplicadas.

Por fim, o módulo subprocess, integrante da biblioteca padrão do Python, foi utilizado no arquivo principal (*main.py*) para executar simultaneamente os scripts responsáveis pela visualização 2D e 3D. Essa abordagem permitiu abrir múltiplas janelas de forma coordenada, facilitando a análise comparativa entre as transformações realizadas em diferentes dimensões.

No presente trabalho, essas tecnologias servem como base para a execução dos scripts responsáveis pela definição dos objetos geométricos, aplicação das matrizes de transformação e exibição dos resultados, garantindo uma implementação clara, modular e facilmente reprodutível. O uso de Python, portanto, compõe a parte prática do estudo, permitindo demonstrar de forma concreta os conceitos matemáticos discutidos anteriormente.

5. Desenvolvimento Prático

5.1. Descrição do Projeto

O código desenvolvido implementa uma estrutura básica para modelagem e transformação de objetos geométricos em duas e três dimensões. Ele está organizado de forma modular, permitindo a separação clara entre a definição dos objetos e a aplicação das transformações lineares.

O módulo square.py contém a definição de um objeto bidimensional simples, representado por um conjunto de vértices que descrevem um quadrado no plano. Cada vértice é modelado como um vetor, possibilitando sua manipulação por operações matriciais. De modo análogo, o módulo cube.py define um cubo tridimensional, especificando seus pontos no espaço 3D por meio de coordenadas vetoriais.

O arquivo main.py centraliza a execução do programa. Nele, os objetos definidos nos módulos anteriores são carregados e submetidos a transformações lineares. A transformação é realizada por multiplicação entre matrizes e vetores, seguindo os princípios da Álgebra Linear.

Dessa forma, cada ponto dos objetos é modificado de acordo com a matriz aplicada, permitindo observar como a transformação altera a forma e a posição do modelo original.

O arquivo requirements.txt reúne as dependências necessárias para execução do projeto, garantindo reprodutibilidade e organização do ambiente. No conjunto, o código demonstra de maneira clara como objetos geométricos podem ser representados e manipulados por meio de matrizes, servindo como base para a implementação do cisalhamento e demais transformações estudadas no trabalho.

5.2. Execução do Código

Para executar o projeto, basta rodar o arquivo **main.py**, que é o ponto de entrada da aplicação. Esse arquivo inicia automaticamente a execução dos scripts **square.py** e **cube.py**, responsáveis por gerar as visualizações das transformações aplicadas aos objetos geométricos.

Isso abrirá duas janelas de visualização: uma exibindo o quadrado transformado e outra mostrando o cubo após a aplicação das matrizes de transformação, porém caso o usuário deseje, é possível executar unitariamente o square.py ou o cube.py.

Para avaliar o comportamento da transformação de cisalhamento em duas dimensões, realizou-se um teste utilizando um quadrado unitário definido por quatro vértices: (0,0), (1,0), (1,1) e (0,1). Esse conjunto de pontos representa um quadrado simples no plano cartesiano, servindo como base adequada para observar deformações geométricas de forma clara.

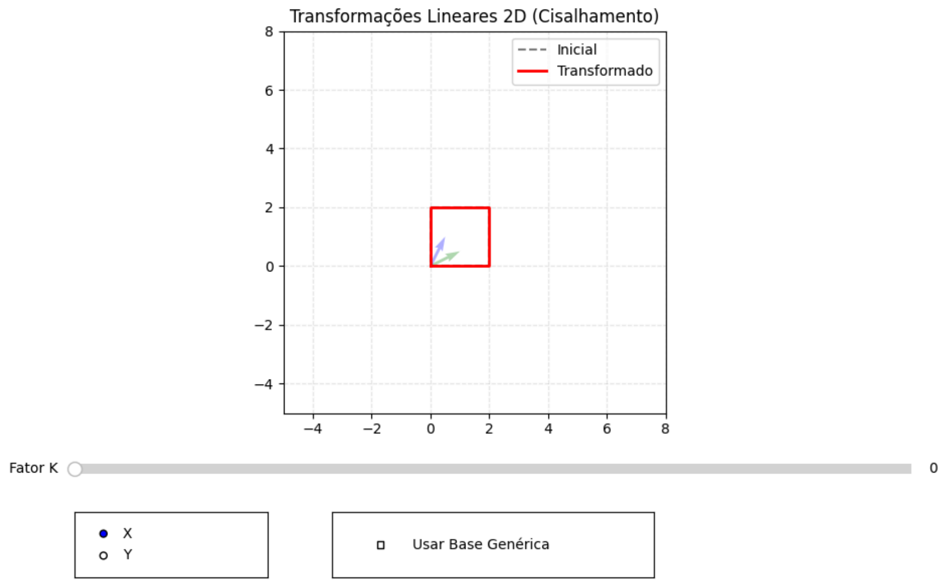

Figura 5: Quadrado de Teste 1 na Forma Original

A transformação utilizada foi um **cisalhamento horizontal** com fator k = 1,0. Nesse tipo de transformação, a coordenada x de cada ponto é deslocada proporcionalmente ao valor de y, enquanto as coordenadas y permanecem inalteradas. A operação é descrita pela matriz:

$$\begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Figura 6: Matriz de Transformação do Quadrado de Teste 1

Aplicando a matriz a cada vértice, obtém-se os seguintes pontos transformados:

- (0,0) → (0,0)
- (1,0) → (1,0)
- (1,1) → (2,1)
- (0,1) → (1,1)

  Os quais podem ser vistos na figura:

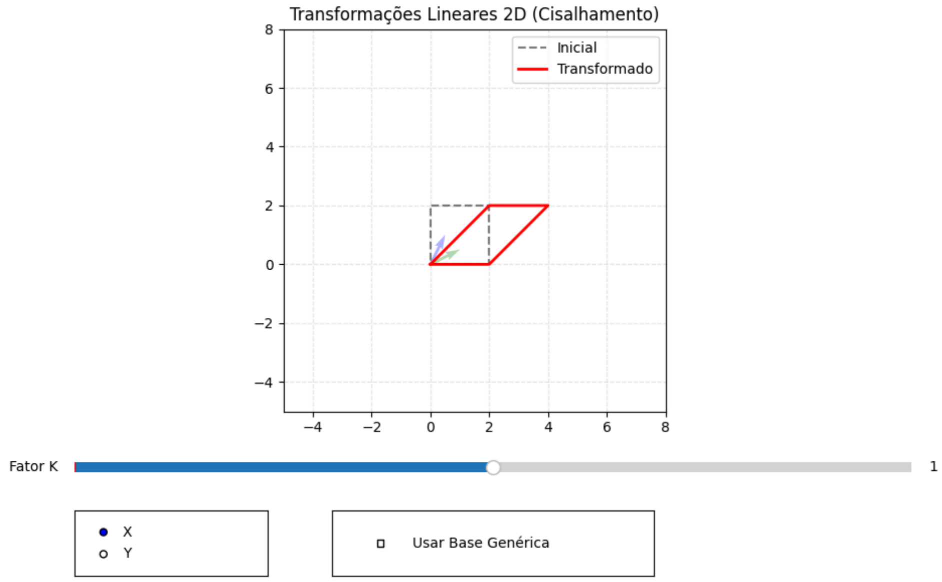

Figura 7: Quadrado de Teste 1 com Cisalhamento Aplicado

Os resultados mostram que os vértices com y = 0 permanecem inalterados, enquanto os vértices com y = 1 são deslocados horizontalmente em uma unidade. O efeito final é a conversão do quadrado original em um paralelogramo, mantendo o paralelismo entre as arestas, mas modificando os ângulos internos da figura.

Realizando um segundo teste com o mesmo quadrado utilizado no primeiro, porém agora, aplica-se um **cisalhamento vertical**, no qual a coordenada y é modificada proporcionalmente ao valor de x. A matriz correspondente a esse tipo de transformação é:

$$\begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

Figura 8: Matriz de Transformação do Quadrado de Teste 2

Neste contexto, a transformação resultante é:

- x' = x
- y' = x + y

E aplicando a matriz a cada vértice, obtém-se os seguintes pontos transformados:

- (0,0) → (0,0)

- (1,0) → (1,1)
- (1,1) → (1,2)
- (0,1) → (0,1)

Os quais podem ser vistos na figura:

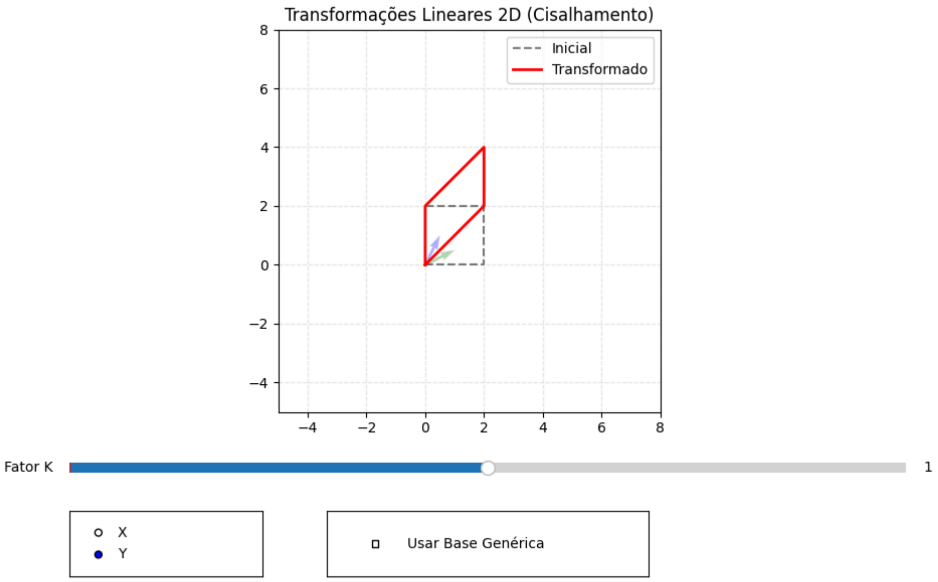

Figura 9: Quadrado de Teste 2 com Cisalhamento Aplicado

Neste caso, os pontos com x = 0 permanecem inalterados, enquanto os pontos com x = 1 sofrem deslocamento vertical proporcional ao valor de x. Assim como no primeiro teste, o quadrado original é transformado em um paralelogramo. Nesse caso, porém, a deformação ocorre no eixo vertical, preservando o paralelismo entre as arestas, mas alterando sua inclinação.

Seguindo então para um espaço tridimensional, o efeito do cisalhamento será utilizado em um cubo unitário definido pelos oito vértices tradicionais: V1 = (0,0,0), V2 = (1,0,0), V3 = (1,1,0), V4 = (0,1,0), V5 = (0,0,1), V6 = (1,0,1), V7 = (1,1,1), V8 = (0,1,1).

Neste teste, aplica-se um **cisalhamento no eixo X proporcional ao eixo Y**, com fator k = 1.0, sendo as respectivas matrizes:

$$\begin{bmatrix} 1 & k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figura 10: Matriz de Transformação do Cubo de Teste 1

A transformação afeta apenas o eixo X:

- x’ = x + y
- y’ = y
- z’ = z

Na aplicação ponto a ponto considerando o plano Z = 0, os pontos se modificam da seguinte forma:

(0,0,0)→(0,0,0)

$(1,0,0) \rightarrow (1,0,0)$

$(1,1,0) \rightarrow (2,1,0)$

$(0,1,0) \rightarrow (1,1,0)$

E considerando o plano $Z = 1$:

$(0,0,1) \rightarrow (0,0,1)$

$(1,0,1) \rightarrow (1,0,1)$

$(1,1,1) \rightarrow (2,1,1)$

$(0,1,1) \rightarrow (1,1,1)$

O cubo sofre uma deformação linear no sentido do eixo X, onde os vértices com $y = 1$ são deslocados em uma unidade. Os vértices com $y = 0$ permanecem inalterados. Assim como ocorre no plano, o cisalhamento transforma o cubo em um **paralelepípedo distorcido**, mantendo o paralelismo das arestas e preservando o volume relativo, mas alterando ângulos e faces.

Esse comportamento confirma as propriedades teóricas da transformação de cisalhamento, evidenciando como a aplicação matricial altera a geometria do objeto sem comprometer sua linearidade. Esse teste serve como base experimental para validar as etapas subsequentes do trabalho, que exploram o cisalhamento em outros objetos e em dimensões superiores.

5. Considerações e Conclusão

O desenvolvimento deste trabalho permitiu compreender de forma integrada como as transformações lineares, em especial o cisalhamento, atuam sobre objetos geométricos em duas e três dimensões. A partir da modelagem de um quadrado e um cubo, foi possível aplicar diferentes matrizes de cisalhamento e observar de maneira concreta as deformações produzidas em cada caso, verificando a preservação de características como paralelismo e proporcionalidade entre os vértices.

Os testes realizados evidenciaram que o cisalhamento horizontal e vertical em 2D geram paralelogramos distintos a partir do quadrado original, enquanto o cisalhamento em 3D transforma o cubo em um paralelepípedo inclinado, mantendo sua estrutura geral. Esses resultados confirmam o comportamento teórico esperado das matrizes de cisalhamento, demonstrando a linearidade da transformação e sua aplicabilidade em cenários computacionais.

Além disso, o uso de Python como ferramenta prática mostrou-se adequado, permitindo estruturar o código de forma modular e facilitar a aplicação das operações matriciais. A correspondência direta entre teoria matemática e implementação computacional reforça a relevância desses conceitos no contexto de Sistemas de Informação, onde transformações geométricas são amplamente empregadas em áreas como computação gráfica, visualização e simulação.

Conclui-se, portanto, que o cisalhamento é uma transformação linear simples, porém poderosa, capaz de demonstrar de forma clara como as matrizes podem alterar a geometria de objetos preservando relações fundamentais entre seus elementos. O trabalho atinge seus objetivos ao evidenciar, por meio de experimentos práticos, a conexão entre Álgebra Linear e aplicações reais em TI.

6. Referências Bibliográficas

ANTON, Howard; RORRES, Chris. **Álgebra Linear com Aplicações**. 10. ed. Porto Alegre: Bookman, 2012.

BOLDRINI, José Luiz et al. **Álgebra Linear**. 3. ed. São Paulo: Harbra, 1980.

FOLEY, James D. et al. **Computer Graphics: Principles and Practice**. 3. ed. Boston: Addison-Wesley Professional, 2013.

HARRIS, C. R. et al. Array programming with NumPy. **Nature**, v. 585, p. 357–362, 2020. Disponível em: https://doi.org/10.1038/s41586-020-2649-2. Acesso em: 01 dez. 2025.

HEARN, Donald; BAKER, M. Pauline. **Computer Graphics with OpenGL**. 3. ed. Upper Saddle River: Pearson Prentice Hall, 2004.

HUNTER, J. D. Matplotlib: A 2D graphics environment. **Computing in Science & Engineering**, v. 9, n. 3, p. 90-95, 2007. Disponível em: https://matplotlib.org. Acesso em: 01 dez. 2025.

PYTHON SOFTWARE FOUNDATION. **Python 3.12 Documentation**. Disponível em: https://docs.python.org/3/. Acesso em: 01 dez. 2025.

STEINBRUCH, Alfredo; WINTERLE, Paulo. **Álgebra Linear**. 2. ed. São Paulo: Pearson Makron Books, 1987.