



**UNIVERSIDADE FEDERAL DO PIAUÍ
CENTRO DE CIÊNCIAS DA NATUREZA
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
DC/CCN032 - COMPUTAÇÃO GRÁFICA - 60h – 2025.2
Prof. Dr. Laurindo de Sousa Britto Neto
laurindoneto@ufpi.edu.br**

TRABALHO DE RASTERIZAÇÃO
Valor do Trabalho: 10,0 pontos

- 1) Desenvolva um Editor Gráfico em C/C++, estilo o Paint da Microsoft, utilizando as bibliotecas GLUT/OpenGL. Leia os comentários do programa “paint.cpp” e utilize o programa “paint.cpp” como referência. O foco deste trabalho consiste em implementar várias técnicas de Rasterização e Transformações Geométricas. Para isso, o programa deve atender, no mínimo, os requisitos listados abaixo:
 - a) Implemente o Algoritmo de Bresenham para traçado de linhas, utilizando apenas *GL_POINTS* como parâmetro da função *glBegin()*. Esse parâmetro indica que cada vértice deve ser tratado como um ponto simples;
 - b) O Algoritmo de Bresenham implementado só funciona no primeiro octante. Desenvolva funções com os algoritmos para redução ao primeiro octante e sua transformação inversa para que as linhas possam ser desenhadas em todos os octantes;
 - c) Desenvolva uma função para desenhar quadriláteros a partir das coordenadas do canto superior esquerdo e do canto inferior direito, utilizando o Algoritmo de Bresenham implementado;
 - d) Desenvolva uma função para desenhar triângulos a cada três coordenadas, utilizando o Algoritmo de Bresenham implementado;
 - e) Desenvolva uma função para desenhar polígonos a partir de quatro ou mais coordenadas, utilizando o Algoritmo de Bresenham;
 - f) Implemente funções para realizar as transformações geométricas de translação, escala, cisalhamento, reflexão e rotação das formas geométrica rasterizadas. Obs.: as funções de transformações geométricas da API do OpenGL ou de qualquer outra biblioteca não podem ser utilizadas;
 - g) Implemente uma função com o Algoritmo de Bresenham para rasterização de circunferências. Utilize apenas *GL_POINTS* como parâmetro da função *glBegin()*. A função de rasterização receberá como parâmetro as coordenadas do centro da circunferência e o tamanho do raio;
 - h) Implemente a função de rasterização para preenchimento de polígonos. Nesse caso, a função preencherá as formas geométricas rasterizadas com exceção da circunferência;
 - i) Implemente o algoritmo *Flood Fill* com vizinhança 4. Pode ser usada a função *glReadPixel()* do *OpenGL* para ler a cor do pixel (x, y) na janela *GLUT*, ou uma matriz auxiliar para a implementação. Nesse caso, a função preencherá todas as formas geométricas rasterizadas. Obs.: Ao utilizar as recursões do algoritmo, é importante ter cuidado para não estourar a pilha de memória (*stack memory*).