



**UNIVERSIDADE FEDERAL DE RORAIMA**  
**CENTRO DE CIÊNCIA E TECNOLOGIA - CCT**  
**CURSO DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO**



**COMPUTAÇÃO GRÁFICA**

**PREENCHIMENTO**

**HENDRICK SILVA FERREIRA-2020026830**

Janeiro de 2025  
Boa Vista/Roraima

## **Resumo**

O propósito deste relatório é apresentar a aplicação dos algoritmos de preenchimento Flood Fill e varredura com análise geométrica em quatro formas diferentes.

## Conteúdo

1 Introdução .....	6
1.1 FloodFill .....	6
1.2 Análise Geométrica .....	6
2 Construção dos Programas .....	6
2.1 Configuração do Ambiente .....	6
2.2 Parâmetros da formas.....	7
3 Descrição dos Programas .....	7
3.1 FloodFill .....	7
3.2 Análise Geométrica .....	7
4 Resultados e Comparação .....	7
4.1 Comparativo .....	10
5 Conclusão .....	10
6 Referências .....	11

## **Figuras**

Figura 1 – Circulo .....	8
Figura 2 – Figura A .....	8
Figura 3 – Figura B .....	9
Figura 4 – Retângulo .....	9

## **1 Introdução**

Este relatório tem como objetivo apresentar a implementação de dois diferentes algoritmos para o preenchimento em Python. Os algoritmos abordados são o FloodFill e o Algoritmo de Análise Geométrica.

### **1.1 FloodFill**

O algoritmo Flood Fill é uma técnica fundamental no preenchimento de áreas em computação gráfica e processamento de imagens. Essa abordagem consiste em iniciar a coloração de um pixel de referência em uma área e, em seguida, se espalhar para os pixels vizinhos, preenchendo gradualmente a região com a cor desejada. O algoritmo continua a se propagar até que todos os pixels conectados à área inicial sejam preenchidos ou até que um limite predefinido seja atingido.

### **1.2 Análise Geométrica**

O Algoritmo de Varredura com Análise Geométrica é uma técnica utilizada no preenchimento de áreas em computação gráfica e processamento de imagens. Ao contrário do algoritmo Flood Fill, que começa o preenchimento a partir de um ponto específico e se espalha para os pixels vizinhos, o Algoritmo de Varredura com Análise Geométrica percorre a imagem em busca de contornos e fronteiras das áreas a serem preenchidas. Esse algoritmo varre a imagem linha por linha, identificando as interseções entre as retas que compõem o contorno das áreas. À medida que as fronteiras são encontradas, o preenchimento ocorre dentro dos limites identificados. Essa abordagem é particularmente útil em cenários onde é necessário preencher áreas delimitadas por formas geométricas complexas.

## **2 Construção dos Programas**

A construção dos programas que utilizam os algoritmos FloodFill e Analise Geométrica - envolve um conjunto comum de etapas. Que são:

### **2.1 Configuração do Ambiente**

A primeira coisa feita foi configurar o ambiente de desenvolvimento, no qual é a instalação de bibliotecas necessárias, como o pygame, que é usado para a exibição gráfica. Isso também envolve a configuração da janela de exibição, definição do título e dimensionamento da tela.

## **2.2 Parâmetros da Formas**

Para que os programas sigam o com pedido feito para realização, defini os parâmetros das formas que serem preenchidas. Que foi o mesmo para dois programas.

## **3 Descrição dos Programas**

### **3.1 FloodFill**

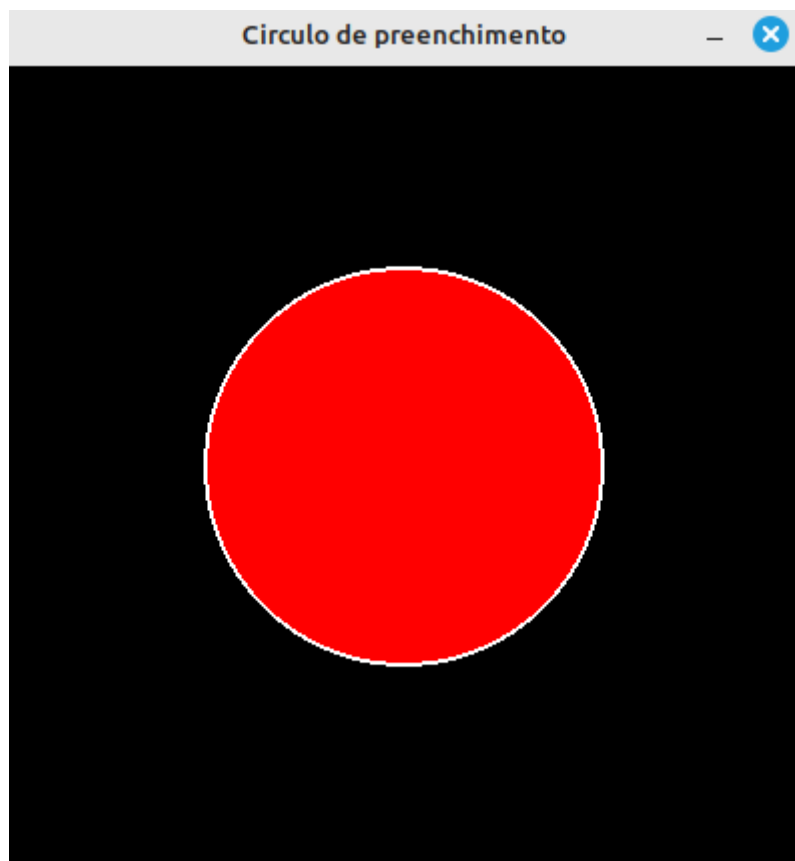
Este programa em Python usa o algoritmo Flood Fill para preencher uma área delimitada por um círculo com a cor azul em uma janela gráfica feita com a biblioteca Pygame. Ele configura a janela, desenha a forma, inicia o preenchimento a partir do centro da forma e aguarda o usuário fechar a janela para encerrar o programa. O código demonstra a aplicação prática do algoritmo de preenchimento Flood Fill em gráficos interativos.

### **3.2 Análise Geométrica**

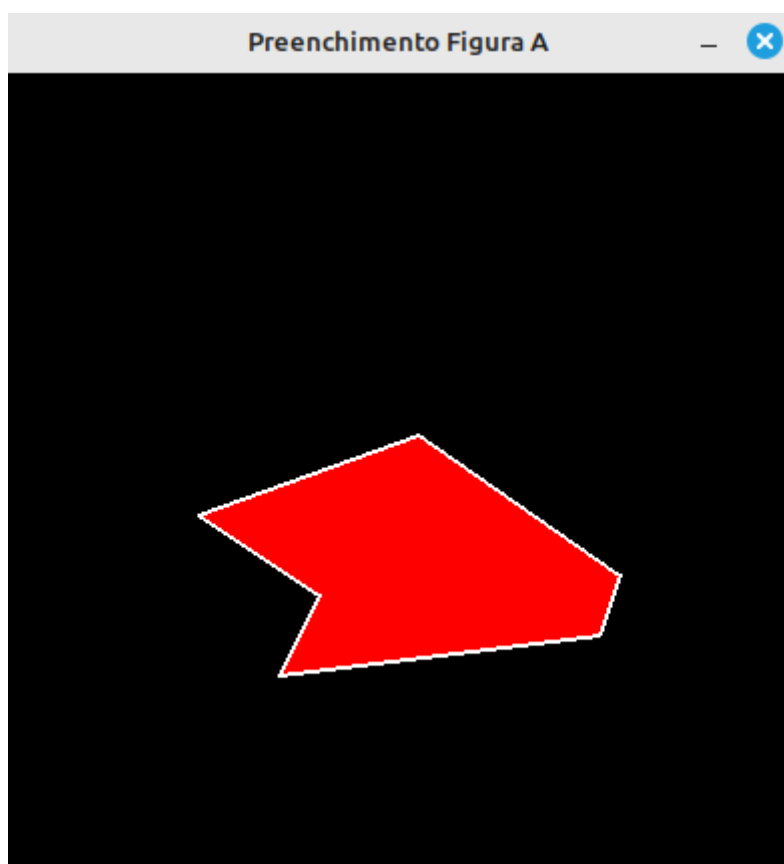
Este programa em Python usa análise geométrica para preencher uma área delimitada por um círculo com a cor azul em uma janela gráfica criada com o Pygame. Ele configura a janela, desenha uma forma, aguarda um curto período e, em seguida, preenche a área interna da forma com a cor azul, usando análise geométrica para determinar os limites da região a ser preenchida. O programa é encerrado quando o usuário fecha a janela. Esse código ilustra a aplicação da análise geométrica no preenchimento de áreas delimitadas em gráficos interativos.

## **4 Resultados e Comparação**

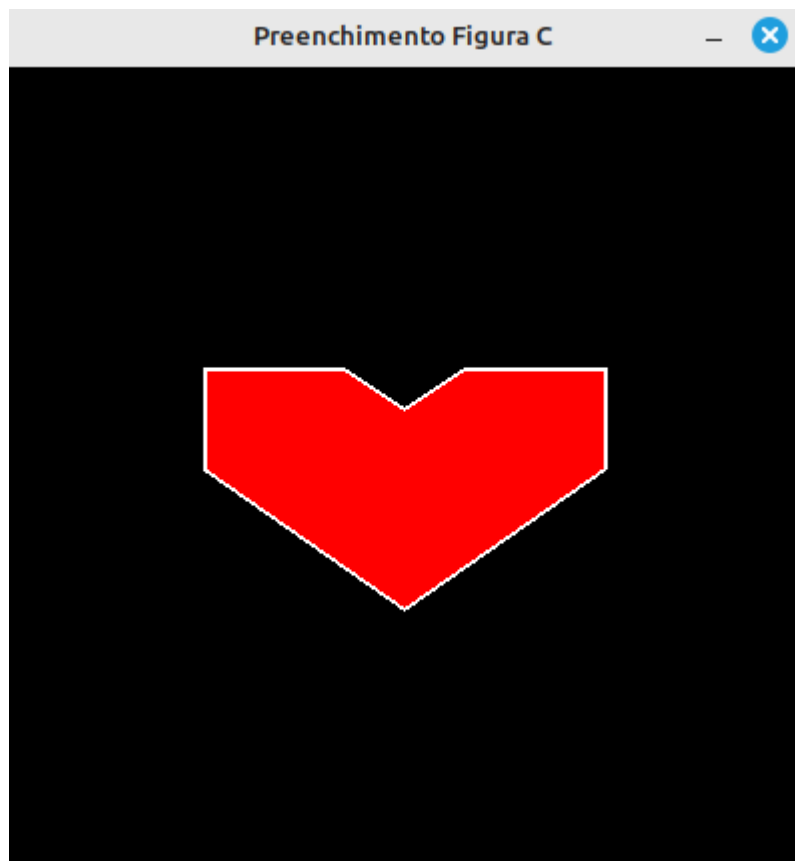
Os resultados obtidos ao executar cada um dos programas são bem interessantes, pois ambos têm meios diferentes de preencher as formas apresentadas. Podemos ver, por meio das figuras, que o resultado entre eles são os mesmos. A comparação pode ser mais detalhada quando falamos do percurso para aquele preenchimento acontecer.



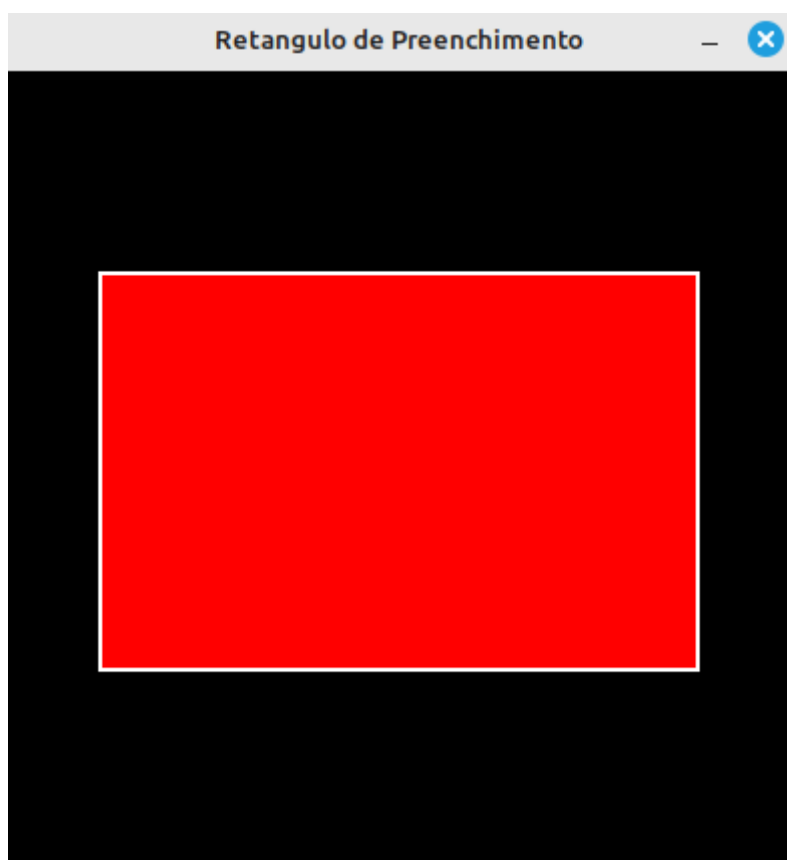
**Figura 1 – Circulo**



**Figura 2 – Figura A**



**Figura 3** – Figura C



**Figura 4** – Retângulo



## **4.1 Comparativo**

No que diz respeito aos resultados, ambos os programas preenchem as áreas de forma eficaz, mas a escolha entre eles depende das características específicas da aplicação. O Flood Fill se destaca em termos de versatilidade e aplicação em áreas complexas, enquanto a Análise Geométrica é ideal para formas geométricas regulares.

Ambos os algoritmos têm suas vantagens e desvantagens, e a escolha entre eles dependerá da natureza da aplicação gráfica em questão. Portanto, ao selecionar o algoritmo de preenchimento de áreas, é essencial considerar as características da área a ser preenchida e as necessidades específicas do projeto. Em última análise, a eficácia de cada algoritmo será determinada pelo contexto em que são aplicados.

## **5 Conclusão**

Em resumo, o uso dos algoritmos de Flood Fill e Análise Geométrica em programas gráficos implementados com o Pygame oferece soluções distintas para o preenchimento de áreas delimitadas. O Flood Fill se destaca pela sua versatilidade e eficácia na coloração de áreas complexas e irregulares, enquanto a Análise Geométrica é uma excelente escolha para o preenchimento de áreas delimitadas por formas regulares. A decisão sobre qual algoritmo empregar dependerá das características específicas da aplicação, levando em consideração o formato das áreas a serem preenchidas e as necessidades do projeto. Em última análise, a eficiência de cada algoritmo será determinada pelo contexto em que são utilizados, e a escolha deve ser guiada pela melhor correspondência às demandas do design gráfico em questão.

## 6 Referências

- <https://gist.github.com/IgnacioCorto/eb8a931479ce23d1e307e3540146bd69>
- <https://stackoverflow.com/questions/76435638/finding-coordinates-of-all-pixels-when-drawing-a-circle-in-pygame>
- <https://gamedev.stackexchange.com/questions/55045/pygame-circular-motion-with-bresenham-s-algorithm>