

**UNIVERSIDADE FEDERAL DE PELOTAS  
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO  
CONCEITOS DE LINGUAGEM DE PROGRAMAÇÃO  
PROF. GERSON CAVALHEIRO**



**DOCUMENTAÇÃO DA IMPLEMENTAÇÃO DA APLICAÇÃO GRÁFICA  
UTILIZANDO DUAS LINGUAGENS DE PROGRAMAÇÃO**

BRUNO CHIM SILVEIRA - 20200909 - [bcsilveira@inf.ufpel.edu.br](mailto:bcsilveira@inf.ufpel.edu.br)  
DANIEL PÔRTO NÚÑEZ - 20202019 - [daniel.pnunes@inf.ufpel.edu.br](mailto:daniel.pnunes@inf.ufpel.edu.br)

Pelotas  
2024

## 1. Especificação

O objetivo do trabalho é desenvolver uma aplicação gráfica, sugerindo opções como o Fractal de Mandelbrot, Ray Tracing ou Simulação de Partículas, utilizando duas linguagens de programação: Python e C (ou C++). O desafio principal é combinar o uso dessas duas linguagens. Python será responsável por criar a interface com o usuário e exibir a imagem gerada, enquanto C (ou C++) será empregada para implementar os cálculos necessários.

## 2. Aplicação

Optamos por implementar o Fractal de Mandelbrot, realizando os cálculos e gerando o fractal usando a linguagem C, enquanto utilizamos Python para criar a interface gráfica. Na matemática, o conjunto de Mandelbrot é um fractal definido como o conjunto de pontos  $c$  no plano complexo para os quais uma sequência definida recursivamente não se expande indefinidamente. Na “Figura1” temos o exemplo de como seria essa sequência.

$$\begin{aligned}c &= x + iy \\Z_0 &= 0 \\Z_1 &= Z_0^2 + c \\&= x + iy \\Z_2 &= Z_1^2 + c \\&= (x + iy)^2 + x + iy \\&= x^2 + 2ixy - y^2 + x + iy \\&= x^2 - y^2 + x + (2xy + y)i \\Z_3 &= Z_2^2 + c = \dots\end{aligned}$$

*Figura 1*

Na sua representação gráfica, o Conjunto de Mandelbrot pode ser descrito como composto por um número infinito de figuras, sendo a maior delas um cardióide (uma curva gerada como um locus) situado no centro do plano complexo. Ao redor do cardióide, existem inúmeros quase-círculos que o tangenciam e cujo raio diminui progressivamente, aproximando-se assintoticamente de zero.

A biblioteca PyQt5 foi utilizada para desenvolver a interface gráfica, garantindo uma integração eficiente com Python. Ela oferece diversos widgets e ferramentas que facilitam a criação de interfaces. A programação orientada a eventos e a facilidade de uso de suas APIs facilitaram a decisão de utilizá-la no nosso projeto, resultando em uma interface que atendeu às nossas necessidades.

No programa escrito em C, geramos o fractal desejado e o cálculo é feito na

função “generate\_fractal”. Utilizamos valores definidos pelo usuário na interface e com base nesses valores serão definidos as cores e os detalhes. Em seguida, geramos uma biblioteca compartilhada com o comando “gcc -shared -o libMandelbrot.so -fPIC mandelbrot.c”, criando o arquivo “libMandelbrot.so”.

Esse arquivo é um “shared object”, uma biblioteca dinâmica que é carregada para a memória no tempo de execução, ao invés de ser copiada para o executável pelo linker na criação do programa. No código Python, no arquivo “main\_window.py”, usamos a biblioteca “ctypes” com o comando “CDLL(nome\_da\_biblioteca\_compartilhada.so)” para carregar a biblioteca compartilhada e acessar suas funções. O código contém comentários que explicam as funções e os trechos principais para facilitar o entendimento do funcionamento.

### 3. Revisão Bibliográfica

1. Fractalize: Modelagem Fractal nas Ciências e Engenharias (<https://www2.ufjf.br/fractalize/2020/12/23/conjunto-de-mandelbrot/>)
2. Mandelbrot Set in all in all programming languages ([https://rosettacode.org/wiki/Mandelbrot\\_set#C](https://rosettacode.org/wiki/Mandelbrot_set#C))
3. FRACTAIS EM C / C++ (<https://acervolima.com/fractais-em-c-c/>)
4. Fractal in C/C++ (<https://www.geeksforgeeks.org/fractals-in-cc/>)
5. Calling C Functions from Python [Step by Step Procedure] (<https://www.cstack.org/calling-c-functions-from-python/>)