



UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO
CIÊNCIA DA COMPUTAÇÃO

CONCEITOS DE LINGUAGENS DE PROGRAMAÇÃO

Relatório da Atividade: Implementação com visualização gráfica e duas linguagens de programação

Nome: Gustavo Raymundi Nygaard

Integração entre C e Python — Conjunto de Mandelbrot

1. Objetivo

Demonstrar a integração entre duas linguagens de programação com papéis distintos:

- C — responsável pelo cálculo numérico de alto desempenho.
- Python (Tkinter) — responsável pela interface gráfica com o usuário.

A aplicação carrega a biblioteca C via **ctypes** e exibe a imagem gerada do conjunto de Mandelbrot.

2. Aplicação Escolhida

O projeto consiste na visualização do Conjunto de Mandelbrot.

Para cada pixel da imagem, mapeia-se um ponto no plano complexo e aplica-se a iteração:

$$z_{n+1} = z_n^2 + c$$

O processo continua até que:

- O módulo $|z_n| > 4$ (divergência), ou
- Seja atingido o limite máximo de iterações.

O número de iterações de cada ponto é convertido em cores, resultando na imagem final do fractal.

3. Papéis das Linguagens

C

- Implementa a função `mandelbrot(...)`.
- Preenche um buffer de inteiros com o número de iterações por pixel.
- É compilado como biblioteca compartilhada:
 - `.so` no Linux
 - `.dll` no Windows
 - `.dylib` no macOS

Python / Tkinter

- Constrói a interface gráfica.
 - Aloca o buffer de dados e chama a função C via `ctypes`.
 - Converte o número de iterações em cores.
 - Desenha a imagem usando `PhotoImage.put`.
 - Permite interatividade simples: zoom, pan e clique para explorar.
-

4. Método de Integração

A biblioteca C é compilada como biblioteca dinâmica, permitindo que Python a carregue em tempo de execução.

Exemplo de carregamento em Python:

```
from ctypes import CDLL, POINTER, c_int
lib = CDLL("mandelbrot.so")
```

- Definem-se `argtypes` e `restype` para a função C.

- Ponteiros são passados com `ctypes.POINTER(c_int)`, permitindo acesso direto ao buffer de pixels.
 - O fluxo é síncrono: Python chama a função C → aguarda cálculo → renderiza imagem.
-

5. Compilação e Execução

Compilar a biblioteca compartilhada:

`make`

(usa `gcc -O2 -fPIC -shared` no Linux/WSL)

Executar a interface gráfica:

`make run`

Ajustes específicos para Windows, macOS e Linux estão descritos no README.md.

6. Observações

- Interface gráfica sem dependências externas (não utiliza Pillow, NumPy ou bibliotecas de terceiros).
- Todas as instruções de instalação, compilação e execução estão detalhadas no README.md.
- O projeto demonstra como C e Python se complementam:
 - C → performance no cálculo numérico
 - Python → interatividade e visualização