

# Documentação de Implementação

**Autor:** Marcelo Augusto Etcheverria

Este documento descreve brevemente a implementação de uma aplicação para reconstruir desenhos vetoriais a partir de arquivos SVG usando Séries de Fourier.

O trabalho proposto consistia em implementar uma aplicação gráfica com apoio de duas linguagens de programação à escolha. O desafio principal era realizar o uso conjunto dessas linguagens.

## 1. Visão Geral da Aplicação

O projeto foca na reconstrução de desenhos vetoriais, aplicando os seguintes passos:

- **Extração de Pontos:** O contorno do desenho é convertido em uma sequência de pontos.
- **Cálculo da Transformada de Fourier:** A Transformada Discreta de Fourier (DFT) é utilizada para decompor o contorno em suas componentes harmônicas.
- **Animação com Epícclos:** A reconstrução do desenho original é visualizada através de uma animação de epícclos, onde cada círculo rotativo representa uma harmônica, e a soma de seus movimentos forma o desenho.
- **Funcionamento da Animação com Epícclos:**

Cada coeficiente da DFT representa um vetor rotativo (epícclo), com raio proporcional à amplitude, rotação definida pela frequência e deslocamento angular pela fase.

O vetor de frequência zero define o ponto de partida.

A ponta de cada epícclo se torna o centro do próximo, formando uma cadeia de círculos conectados.

O último epícclo determina o ponto final, que é registrado para formar o traço do desenho.

A cada quadro da animação, as posições são recalculadas pelas equações:

$$x_{\text{novo}} = x_{\text{anterior}} + A \cdot \cos(2\pi ft + \varphi)$$

$$y_{\text{novo}} = y_{\text{anterior}} + A \cdot \sin(2\pi ft + \varphi)$$

onde  $A$  é a amplitude,  $f$  é a frequência e  $\varphi$  é a fase.

Veja uma explicação elegante e visual do conceito neste vídeo do 3Blue1Brown: [Epicycles by 3Blue1Brown](#)

## 2. Linguagens de programação utilizadas

A aplicação é desenvolvida utilizando Python e C++, com as seguintes responsabilidades:

### Python:

- **parser.py**: Responsável por ler arquivos SVG, converter seus caminhos em pontos igualmente espaçados, centralizá-los e normalizá-los. Os pontos processados são salvos em `input.csv` para uso pelo código C++.
- **animate\_fourier.py**: Lê os coeficientes gerados pelo C++ (contidos em `output.json`) e renderiza a animação dos epiciclos em tempo real usando a biblioteca Matplotlib.

### C++:

- **DFT.cpp**: Este módulo lê os pontos de `input.csv`, executa a Transformada Discreta de Fourier sobre as coordenadas e gera o arquivo `output.json`. Este arquivo contém os coeficientes (frequência, amplitude e fase) ordenados por amplitude decrescente.

## 3. Integração Python e C++

A comunicação entre as linguagens é realizada através de arquivos intermediários, garantindo simplicidade, portabilidade e independência de plataforma:

- **CSV (`input.csv`)**: Gerado pelo Python e consumido pelo C++.
- **JSON (`output.json`)**: Gerado pelo C++ e consumido pelo Python.

Este método permite que as partes do projeto sejam executadas de forma independente ou em conjunto, sem a necessidade de bibliotecas de *binding* específicas.