



# INSTRUMENTAÇÃO ELETRÔNICA PARA AQUISIÇÃO DE DADOS EM ALTA VELOCIDADE

Centro Brasileiro de Pesquisas Físicas  
Laboratório de Sistemas de Detecção

**Bolsista:** Vitor Santos Pereira  
(Engenharia Elétrica/UERJ)  
**Orientador:** Herman Pessoa Lima Jr  
Julho/2019

# INTRODUÇÃO

- Dispositivos Lógicos Programáveis (PLD);
- Os PLDs são circuitos integrados que podem ser configurados pelo próprio usuário. Numa arquitetura baseada em dispositivo lógico programável, por exemplo, FPGA (Field Programmable Gate Array - HCPLD), um algoritmo de controle é implementado por hardware, sem necessidade de ciclos de busca e execução de instruções.

# OBJETIVO

- Software em python para geração de imagens a partir de fonte raio-X.

# FPGA

- Arquitetura:
  - Matriz bidimensional;
  - Blocos lógicos;
  - Chaves de interconexão.

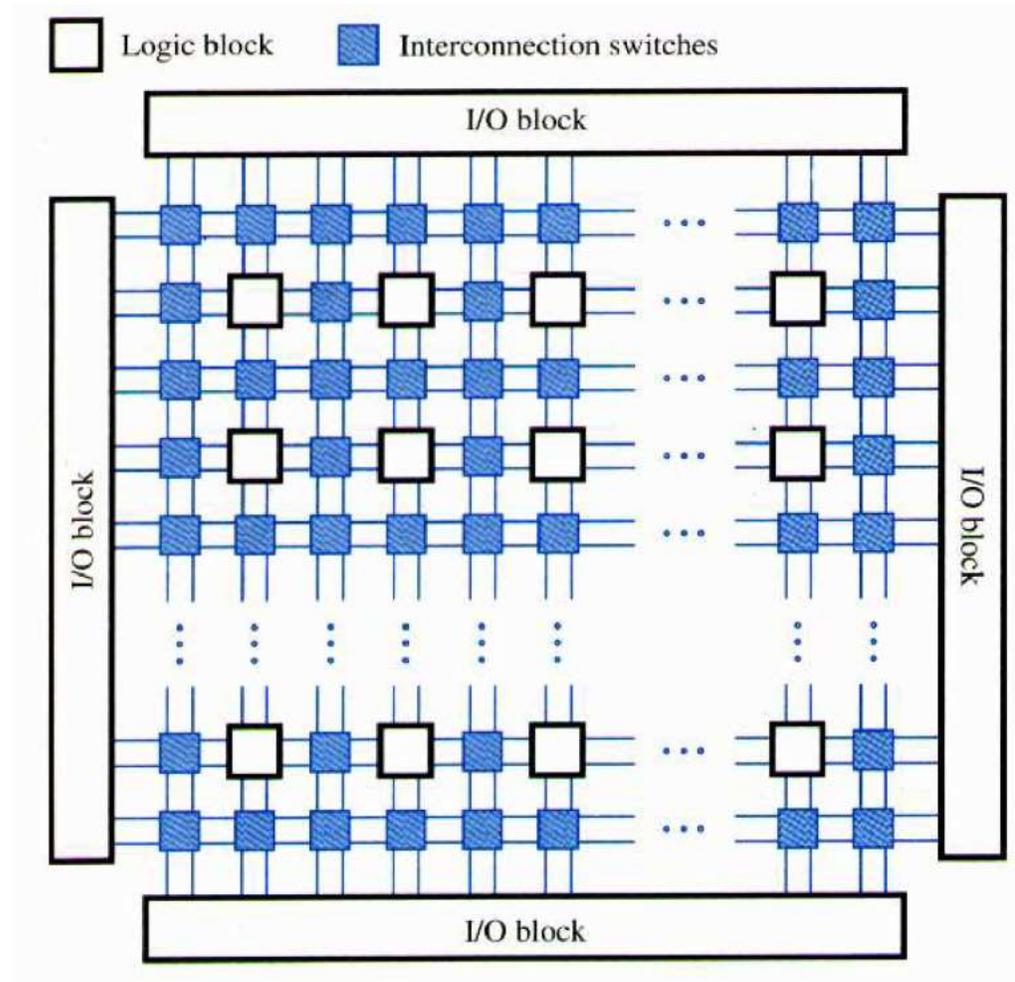


Figura 1: Matriz de blocos de uma FPGA.

# FPGA

- Aplicações;
- Conversor Tempo-Digital (TDC).

# AQUISIÇÃO DE DADOS

- Sistema de coordenadas

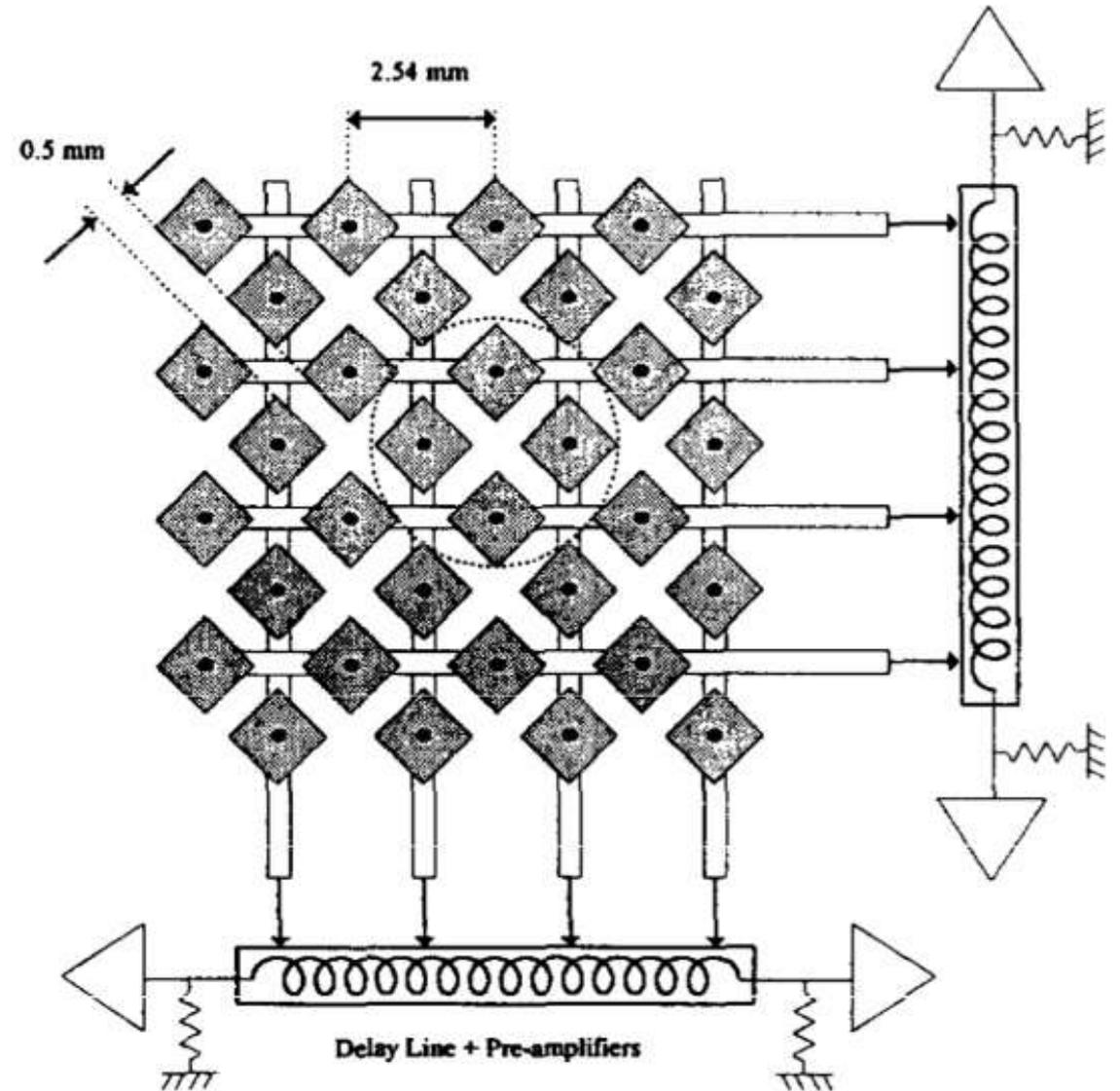


Figura 2: Ilustração para um arranjo para as posições X e Y.

# AQUISIÇÃO DE DADOS

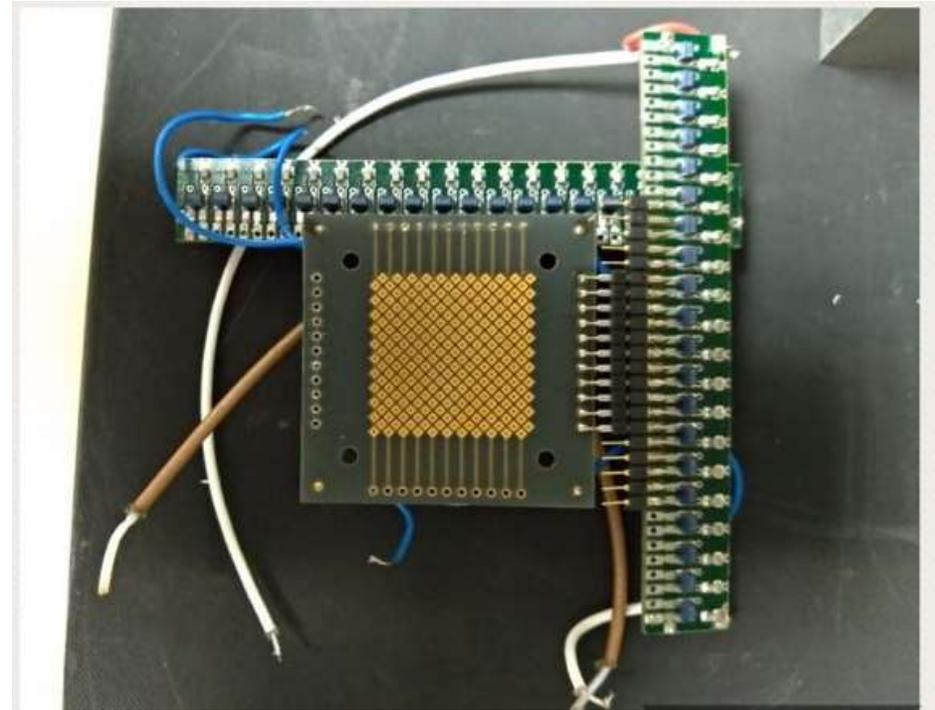
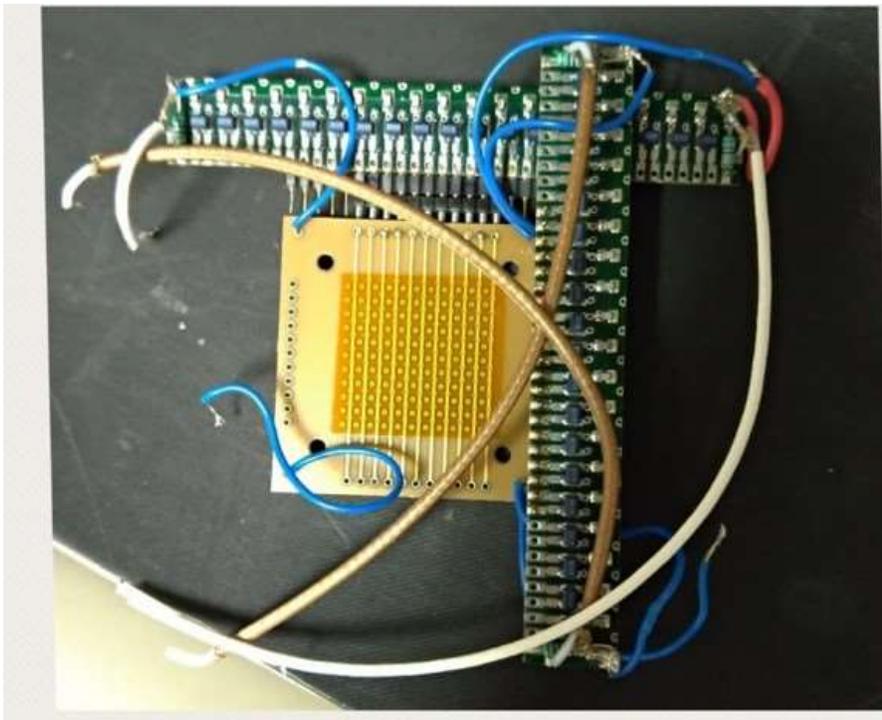


Figura 3 e 4: Linhas de retardo em frente/verso.

# AQUISIÇÃO DE DADOS

- Detector sensível a posição

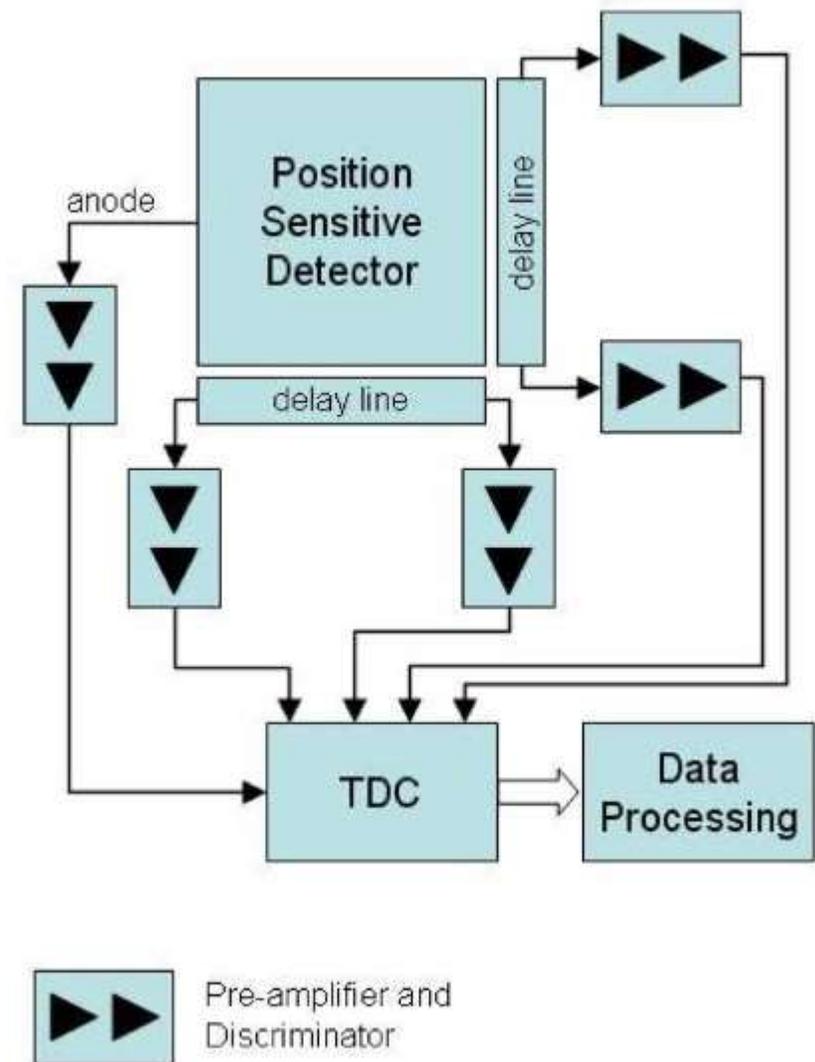


Figura 5: Diagrama de bloco de um detector sensível a posição.

# AQUISIÇÃO DE DADOS

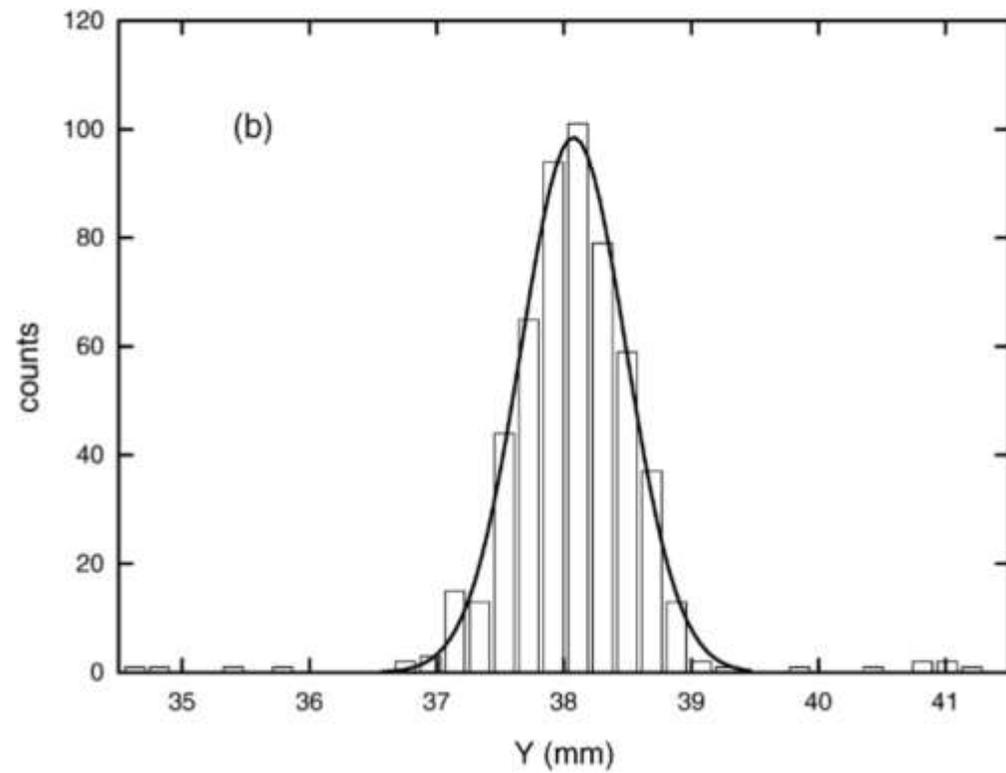
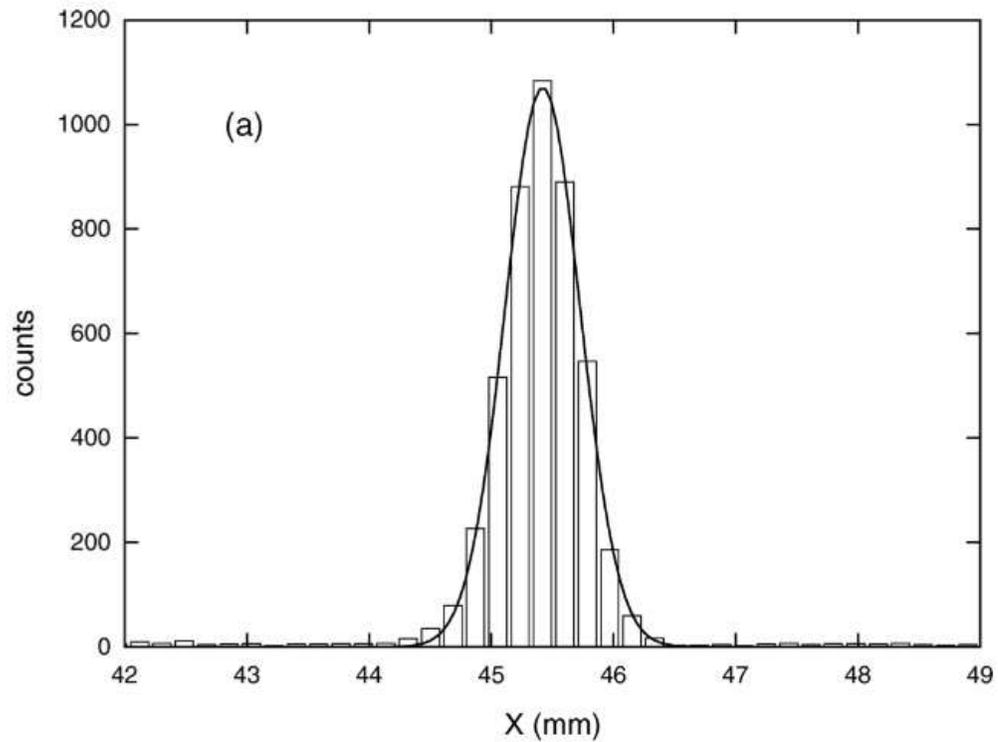


Figura 6 e 7: Contagem de eventos nas posições X e Y.

# PROCESSAMENTO DE IMAGEM

- Software de geração de imagem (DELPHI)

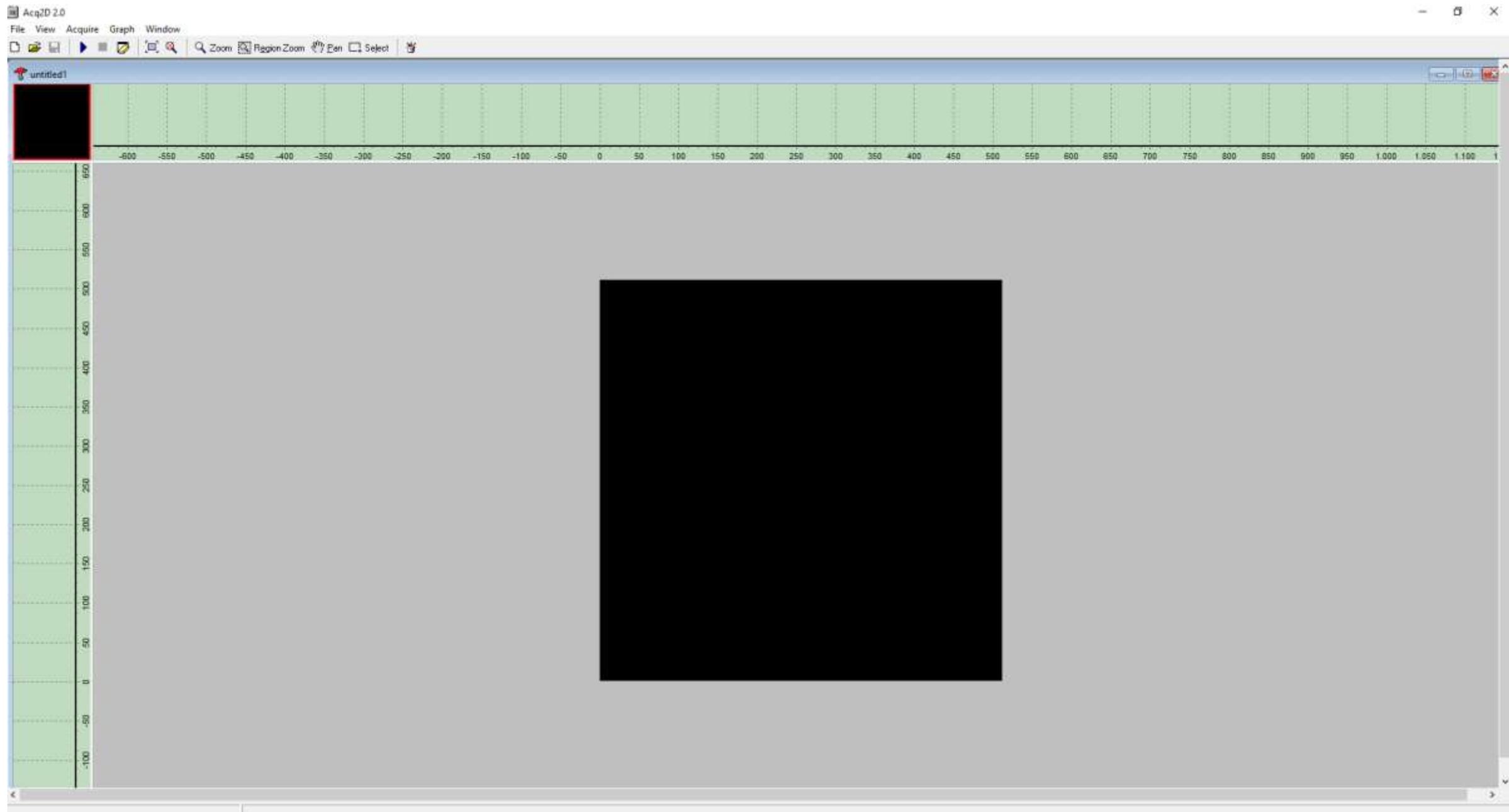
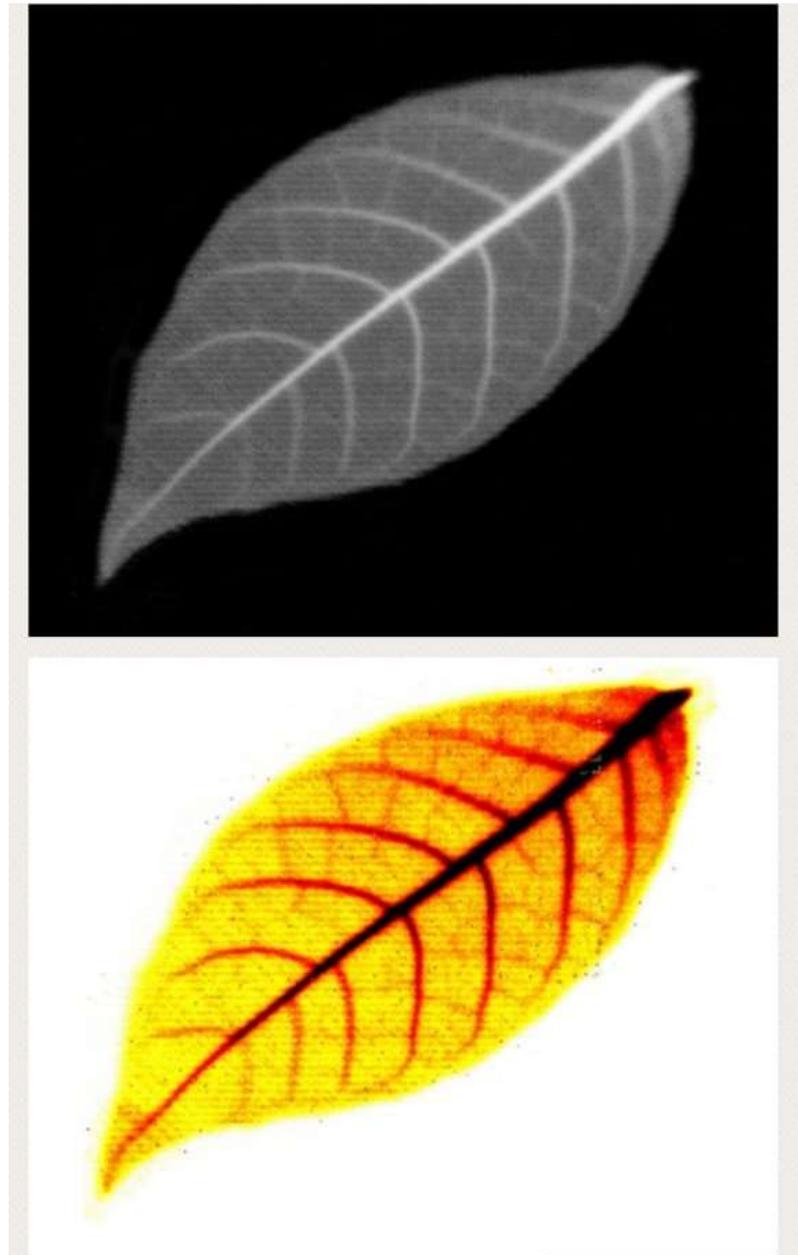


Figura 8: Software em DELPHI

# PROCESSAMENTO DE IMAGEM

- Resultados do software em DELPHI.

Figura 9: Imagem de uma folha processada pelo Software DELPHI.



# PROCESSAMENTO DE IMAGEM

- Novo Software de geração imagem (Python);

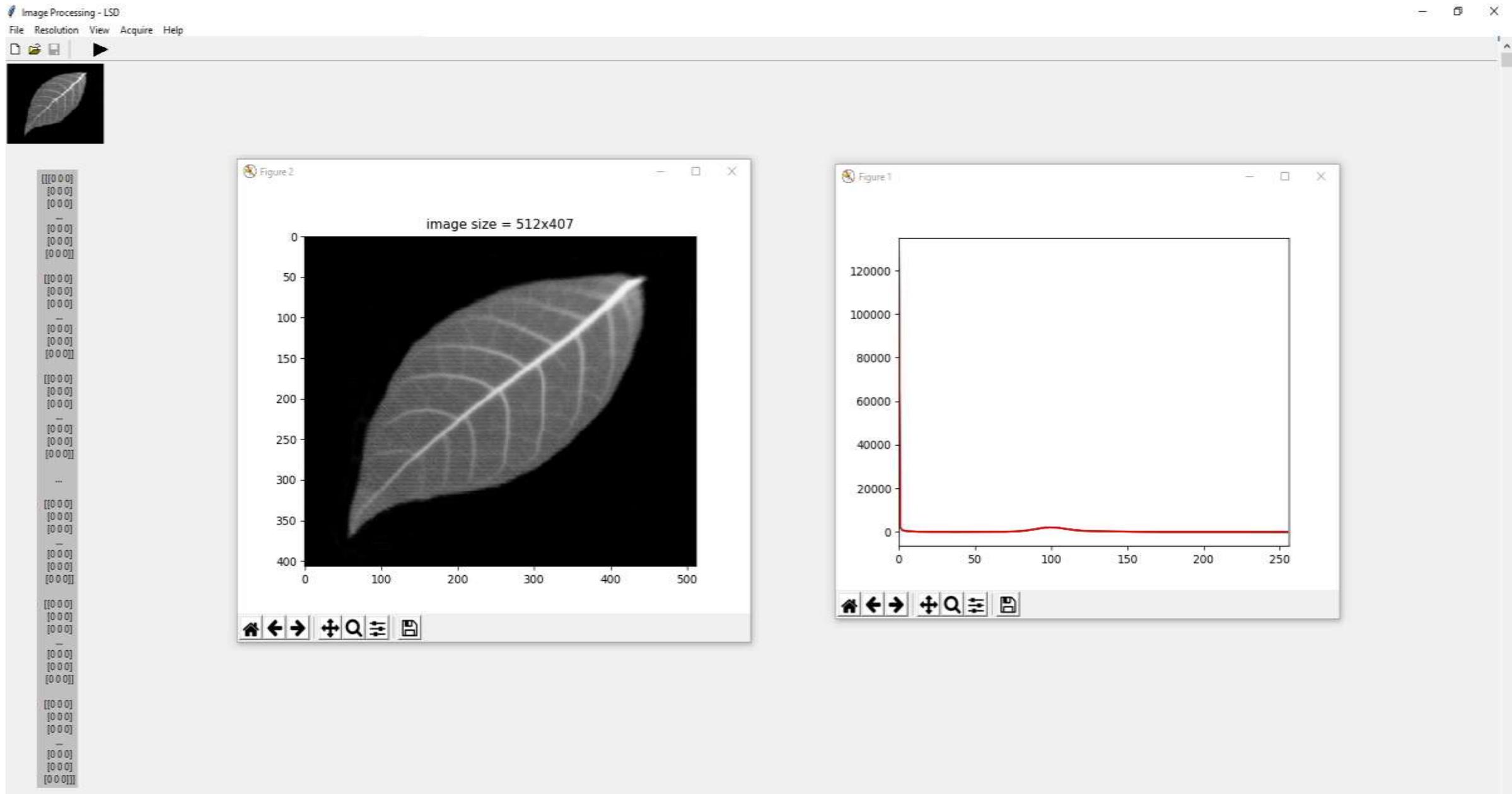


Figura 10: Software em Python

# PROCESSAMENTO DE IMAGEM

- Código do Software;

Figura 11: Código Principal em Python para Leitura

```
*PyCOD.py - /home/vitor/Área de Trabalho/PyCOD.py (3.6.8)*
File Edit Format Run Options Window Help
import usb.core
import usb.util
import sys
import os

dev = usb.core.find(idVendor=0x09FB, idProduct=0x6001)

interface=0
endpoint = dev[0][0][0]
if dev.is_kernel_driver_active(interface) is True:
    dev.detach_kernel_driver(interface)
    usb.util.claim_interface(dev,interface)

collected = 0
attempts = 50
while collected < attempts:
    try:
        data = dev.read(endpoint.bEndpointAddress,endpoint.wMaxPacketSize)
        collected += 1
        print (data)
    except usb.core.USBError as e :
        data = None
        if e.args == ('Operation timed out',):
            continue
usb.util.release_interface(dev,interface)
dev.attach_kernel_driver(interface)
```



# CONCLUSÃO

- Um software com uma linguagem de programação atualizada, como Python, trará melhores resultados para o projeto, tendo em vista maiores quantidades de funções que podem ser adicionadas no software, aprimorando a coleta de dados.

# REFERÊNCIAS BIBLIOGRÁFICAS

- S. Brown *et al.* , Fundamentals of Digital Logic with VHDL Design, (2009).
- S. Dey, Hands-On Image Processing with Python (2018).
- BARBOSA, Ademarlaudo França. Use of a multilayer printed circuit board as the position sensing electrode in an MWPC. Nuclear Instruments and Methods in Physics Research A 371 (1996) 368-374.
- L.M. de Andrade Filho, A.F. Barbosa, H.P. Lima Jr e P.R.B. Marinho. A TDC-Based System for X-Ray Imaging Detectors.