FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS

FATEC PROFESSOR Jessen Vidal

FERNANDO PASSACANTILLI SILVA FERRARI

CCD CONTROLLER: SOFTWARE DE INTERFACE GRÁFICA PARA CONTROLE DE CÂMERA CCD

São José dos Campos

2018

FERNANDO PASSACANTILLI SILVA FERRARI

CCD CONTROLLER: SOFTWARE DE INTERFACE GRÁFICA PARA CONTROLE DE CÂMERA CCD

Trabalho de Graduação apresentado à Faculdade de Tecnologia São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas

**Orientador: Prof. Me. Fernando Masanori Ashikaga**

São José dos Campos

2018 (ano)

**Dados Internacionais de Catalogação-na-Publicação (CIP)**

**Divisão de Informação e Documentação**

FERRARI, Fernando Passacantilli Silva

CCD Controller: Software de Interface Gráfica Para Controle de Câmera CCD.

São José dos Campos, 2018.

999f. (número total de folhas do TG)

Trabalho de Graduação – Curso de Tecnologia em (Informática) com Ênfase em Análise e Desenvolvimento de Sistemas, FATEC de São José dos Campos: Professor Jessen Vidal, 2018.

Orientador: Prof. Me. Fernando Masanori Ashikaga.

1. Áreas de conhecimento. I. Faculdade de Tecnologia. FATEC de São José dos Campos: Professor Jessen Vidal. Divisão de Informação e Documentação. II. Programa Para Controle de Câmera CCD

**REFERÊNCIA BIBLIOGRÁFICA –**

FERRARI, Fernando Passacantilli Silva. **CCD Controller: Software de Interface Gráfica Para Controle de Câmera CCD.** 2018. 999f. Trabalho de Graduação - FATEC de São José dos Campos: Professor Jessen Vidal.

**CESSÃO DE DIREITOS –**

NOME DO AUTOR: Fernando Passacantilli Silva Ferrari

TÍTULO DO TRABALHO: CCD Controller: Software de Interface Gráfica Para Controle de Câmera CCD

TIPO DO TRABALHO/ANO: Trabalho de Graduação / 2018.

É concedida à FATEC de São José dos Campos: Professor Jessen Vidal permissão para reproduzir cópias deste Trabalho e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste Trabalho pode ser reproduzida sem a autorização do autor.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fernando Passacantilli Silva Ferrari

39465261-7

**Fernando Passacantilli Silva Ferrari**

CCD CONTROLLER: SOFTWARE DE INTERFACE GRÁFICA PARA CONTROLE DE CÂMERA CCD

Trabalho de Graduação apresentado à Faculdade de Tecnologia São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Informática com Ênfase em Análise e Desenvolvimento de Sistemas.

**Composição da Banca**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Nome do Componente da Banca, titulação e Instituição**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Nome do Componente da Banca, titulação e Instituição**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Fernando Masanori Ashikaga, mestre FATEC São José dos Campos**

\_\_\_\_\_/\_\_\_\_\_/\_\_\_\_\_

**DATA DA APROVAÇÃO**

RESUMO / ABSTRACT

Apresentação concisa dos pontos relevantes do documento deve ser exposta no resumo. No presente caso o resumo será informativo, assim deverá ressaltar o objetivo, a metodologia, os resultados e as conclusões do documento. A ordem desses itens depende do tratamento que cada item recebe no documento original. O resumo deve ser composto por uma sequência de frases concisas, afirmativas e não em enumeração de tópicos. Deve ser escrita em parágrafo único e espacejamento de 1,5. A primeira frase deve ser significativa, explicando o tema principal do documento. Deve-se usar o verbo na voz ativa e na terceira pessoa do singular. Quanto a sua extensão, o resumo deve possuir de 150 a 500 palavras..

O abstract é o resumo da obra em língua estrangeira, que basicamente segue o mesmo conceito e as mesmas regras que o texto em português. Recomenda-se que para o texto do abstract o autor traduza a versão do resumo em português e faça, se necessário, os ajustes referentes à conversão dos idiomas. É importante observar que o título e texto NÃO DEVEM estar em itálico.

**Palavras-Chave/ Keywords**:: <Um mínimo de 3 e um máximo de 10 palavras, separadas entre si por ponto e vírgula “;” e finalizadas por ponto. As palavras-chave sãopalavras representativas do conteúdo do documento. Recomenda-se que o autor traduza para o inglês as Palavras-Chave em português e faça, se necessário, os ajustes referentes à conversão dos idiomas.>

LISTA DE ABREVIATURAS E SIGLAS

CCD *Charge Coupled Device*

UTC *Universal Time Coordinated*

ADU *Analog-to-Digital Unit*

SUMÁRIO

[1](#_Toc438249388) INTRODUÇÃO 1

[1.1](#_Toc438249389) OBJETIVO GERAL 1

[1.2](#_Toc438249390) OBJETIVOS ESPECÍFICOS 1

[1.2.1](#_Toc438249391) ABORDAGEM METODOLÓGICA 1

[2](#_Toc438249392) CONTEXTUALIZAÇÃO TECNOLÓGICA 2

[2.1](#_Toc438249393) Tecnologias Utilizadas 2

[2.1.1](#_Toc438249394) Tecnologia 1 2

[2.1.2](#_Toc438249395) Tecnologia N 3

[2.2](#_Toc438249396) Soluções Existentes 3

[2.2.1](#_Toc438249397) Solução 1 3

[2.2.2](#_Toc438249398) Solução N 3

[2.3](#_Toc438249399) Levantamento de Requisitos 3

[2.3.1](#_Toc438249400) Definição dos Stakeholders <opcional> 3

[2.3.2](#_Toc438249401) Metodologia Utilizada 4

[2.3.3](#_Toc438249402) Requisitos Funcionais 4

[2.3.3.1](#_Toc438249403) Requisito 1 4

[2.3.3.2](#_Toc438249404) Requisito N 4

[2.3.4](#_Toc438249405) Requisitos Não Funcionais 4

[2.3.4.1](#_Toc438249406) Requisito 1 4

[2.3.4.2](#_Toc438249407) Requisito N 4

[3](#_Toc438249408) DESENVOLVIMENTO 5

[3.1](#_Toc438249409) Modelo de Dados 5

[3.2](#_Toc438249410) Arquitetura 5

[3.2.1](#_Toc438249411) Módulo 1 5

[3.3](#_Toc438249412) Deploy 5

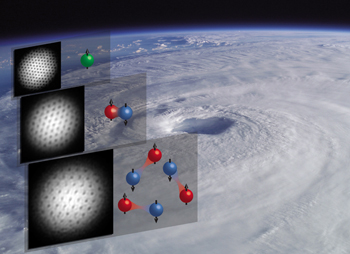
[4](#_Toc438249413) RESULTADOS E DISCUSSÃO 6

[5](#_Toc438249414) TRABALHOS FUTUROS 7

# INTRODUÇÃO

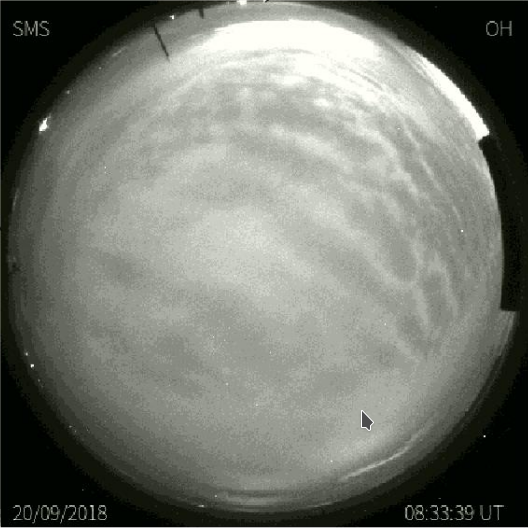
O trabalho é motivado pela necessidade da obtenção de imagens referentes à intensidade da vibração de determinadas moléculas atmosféricas. Pois este tipo de dado, disponível através das imagens atmosféricas, citadas anteriormente, apura movimentação ou perturbações na atmosfera, que normalmente se dá por precipitações de meteoritos ou indução térmica, por exemplo. (PRINCETON INSTRUMENTS, 2016) A imagem abaixo apresenta um exemplo teórico de tal situação.

Figura 1: Exemplo de imagem obtida conforme agitação molecular



Fonte: princetonistruments.com [01]

Figura 2: Exemplo de imagem obtida por imageador CCD, exibida por meio de software.



Fonte: Software CCD Controller 1.2.0 [02]

## OBJETIVO GERAL

O objetivo geral deste trabalho é desenvolver um software constituído de interface gráfica para interagir com a câmera CCD Pixis, da companhia Princeton Instruments, visando, como funcionalidade principal, simplificar e tornar mais gerenciável o ato de obter imagens atmosféricas, de acordo com os parâmetros informados pelo usuário.

## OBJETIVOS ESPECÍFICOS

Para a consecução deste objetivo foram estabelecidos os objetivos específicos:

* Compreender o funcionamento de aplicações similares;
* Compreender a estrutura da biblioteca dinâmica constituída de instruções da câmera;
* Compreender o funcionamento de bibliotecas externas fornecedoras de dados;
* Analisar a capacidade de reutilizar comportamento de softwares similares;
* Rever conceitos de linguagem e programas auxiliares escolhidos.

### ABORDAGEM METODOLÓGICA

Buscando o êxito na realização do objetivo geral, o processo de criação do programa se dará, principalmente, por meio de escrita de código fonte de acordo com a lógica necessária para o melhor funcionamento do programa, tendo como principal diretriz o conjunto de objetivos específicos.

Para auxiliar na construção do software, serão utilizados trechos de código de trabalhos anteriores com a mesma premissa, destinados à equipamentos de câmera CCD diferentes.

# CONTEXTUALIZAÇÃO TECNOLÓGICA

Neste capítulo serão apresentadas as tecnologias utilizadas na solução do problema, no caso, linguagens de programação, softwares utilizados e um levantamento de requisitos.

## Tecnologias Utilizadas

Esta seção apresenta as principais tecnologias utilizadas na solução proposta para o problema, neste caso, linguagem Python e o software auxiliar QT Designer.

### Python

Python é atualmente, uma das linguagens de programação com maior popularidade, acarretada principalmente pela sua simplicidade e flexibilidade, com relação à escrita de código fonte e módulos auxiliares programados em Python previamente.(LIN, 2018)

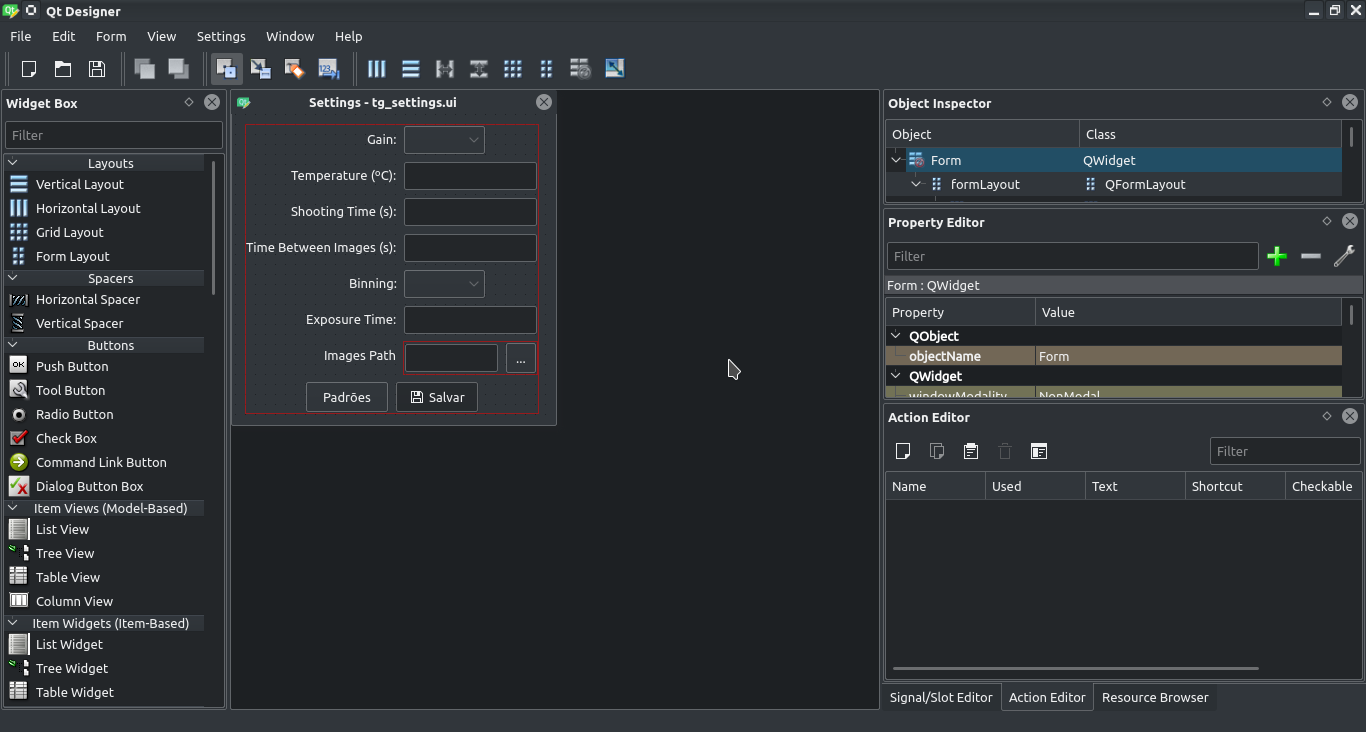
Além dos fatores citados previamente, Python é uma linguagem interpretada, ou seja, não só não requer um programa compilador para executar um conjunto de código, como também é possível testar trechos de código de forma dinâmica, por meio de linhas de comando.(LIN, 2018)

### QT Designer

QT Designer é um programa para criação de interfaces gráficas baseado no framework QT, construído na linguagem C++, porém, com suporte a compilação por meio de uma série de outras linguagens, entre estas se encontra Python.(QT, 2012)

O fato que torna seu uso conveniente, é que os arquivos de interface gráfica criados neste programa podem ser transformados em programas com código fonte escrito em linguagens das quais o compilador QT suporta, de forma a facilitar a associação entre a interface gráfica de determinado programa e diversas partes de seu código fonte, assim auxiliando a obtenção do produto final ideal.(QT, 2012)

**Figura 3: Ferramenta Para Construção de Interfaces de Usuário QT Designer**



## Soluções Existentes

Esta seção apresenta uma pesquisa mercadológica de soluções existentes.

## 2.2.1 Delmic Odemis (Open Delmic Microscope Software)

Projeto desenvolvido em linguagem Python para controlar uma série de câmeras CCD, entre estas, a câmera em questão. Por ser um projeto destinado a controlar uma série de imageadores, é mais completo, porém, com um procedimento de preparação para uso mais burocrático.

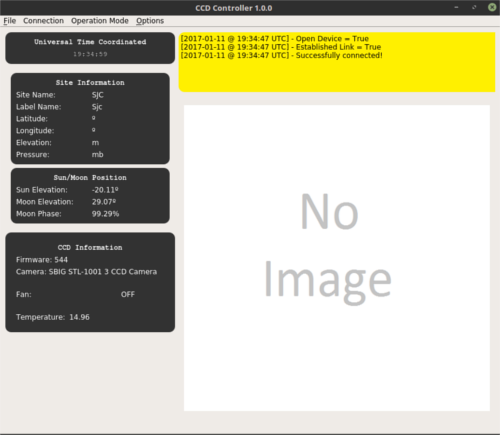
## 2.2.2 Micro-Manager

Projeto mais constituído, com o mesmo propósito do Odemis; Desenvolvido em uma série de linguagens, predominantemente em C++. Também controla as câmeras PIXIS.

## 2.2.3 CCD Controller Primeira Versão.

Não controla a câmera em questão, controla as câmeras SBIG, porém a solução é a mesma, destinada a um equipamento diferente, e empata com o Odemis em maior inspirador do trabalho atual.

**Figura 4: Interface Gráfica do Software CCD Controller**

Fonte: Software CCD Controller 1.0.0 [03]

## Levantamento de Requisitos

Esta seção apresenta o levantamento de requisitos da solução proposta.

### Metodologia Utilizada

O principal método de levantamento utilizado consiste em consultar o cliente, por meio de entrevista. No entanto, há uma baixa frequência de pesquisas de opinião.

### Requisitos Funcionais

Estão listados abaixo os requisitos funcionais levantados por meio de casos de uso e anatomia de sistemas similares.

### 2.3.2.1 Conexão com a Câmera

O software deve possuir um meio de se conectar com a câmera, para que certas operações posteriores sejam possíveis.

### 2.3.2.2 Ajustar Parâmetros para Adquirir Imagens

O software deve possuir um meio de receber do usuário, os valores para os principais parâmetros de obtenção de imagens para depois serem utilizados pela rotina correspondente.

### 2.3.2.3 Adquirir Imagens

Trata-se da função mais essencial do software em questão, realizar a aquisição de imagens por meio da câmera. Esta função também deve possuir um meio de ser acionada.

### 2.3.2.4 Modo de Espera Padrão

Ao conectar com a câmera, ou interromper as aquisições, o sistema deve ajustar a temperatura do imageador para o valor ideal correspondente ao modo de espera, a fim de economizar a energia utilizada caso o equipamento não esteja obtendo fotos, no caso, 25ºC.

### Requisitos Não Funcionais

Os requisitos não funcionais, são essencialmente referentes ao acionamento de funções permitidas conforme as atividades atuais do software e atualização de informações referentes aos parâmetros de obtenção de imagens.

### 2.3.3.1 Persistir os Parâmetros Fornecidos

Tendo em mente que a função de aquisição de imagens necessita parâmetros fornecidos pelo usuário, os mesmos devem ser persistidos em um arquivo de configuração, por meio de uma janela de opções, na qual o susário digita os valores que o mesmo deseja persistir ao arquivo de configuração da aplicação, para que a mesma possa eventualmente utilizar os valores obtidos ao adquirir imagens.

### 2.3.3.2 Programação de Configurações Padrão Para os Parâmetros

Se os valores a serem persistidos estiverem vazios, o programa apresentará erros, para tal, o programa deve incluir um módulo de configurações padrão a serem persistidas.

### 2.3.3.3 Regular o Uso de Botões com Base nas Atividades Atuais do Software

O software não apresentará o comportamento ideal caso chamadas de função inválidas ou inviáveis sejam realizadas, por exemplo, se o botão para a aquisição de imagens for clicado sem que o equipamento esteja conectado ao computador, é necessário que o mesmo retorne uma mensagem dizendo que a opção é inválida para a situação atual da câmera, caso contrário, o programa poderá apresentar falhas de desempenho ou até mesmo travar.

# DESENVOLVIMENTO

Este capítulo apresenta o processo de desenvolvimento da solução proposta. Seu conteúdo pode variar, dependendo da metodologia adotada.

As seções a seguir são sugestões, baseadas nas soluções de software mais comuns.

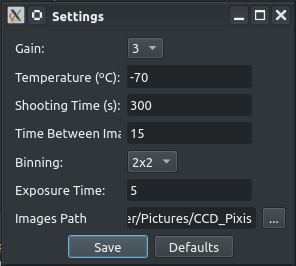
Buscando o êxito na realização do objetivo geral, o processo de criação do programa se dará, principalmente, por meio de escrita de código fonte de acordo com a lógica necessária para o melhor funcionamento do programa, tendo como principal diretriz o conjunto de objetivos específicos.

Para auxiliar na construção do software, serão utilizados trechos de código de trabalhos anteriores com a mesma premissa, destinados à equipamentos de câmera CCD diferentes.

## Modelo de Dados

Conforme dito anteriormente, os dados a serem persistidos consistem nos parâmetros de aquisição fornecidos pelo usuário, modelados conforme a imagem.

**Figura 5: Janela de opções do software**



Gain: Trata-se do ganho da câmera, que define a quantia de fotoelétrons (elétrons criados a partir da emissão de luz em determinado meio) necessários para compor, nas imagens adquiridas uma ADU, isto é, uma unidade analógica, neste caso, fotoelétrons, digitalizada, ou seja, convertida em um pixel. (RICHMOND, 2017)

Os valores de ganho suportados pelo imageador testado são de 1, 2 e 4. (PRINCETON INSTRUMENTS, 2015)

Temperature: Indica ao imageador a temperatura que o mesmo deve manter durante a aquisição. O imageador suporta temperaturas entre -90ºC e 25ºC. Sendo que a temperatura de-70ºC é a mínima recomendada para a operação do imageador e a mínima aceita pelo software. (PRINCETON INSTRUMENTS, 2015)

Shooting Time: Se trata do tempo em que o imageador permanecerá adquirindo imagens, isto é, em seu modo de aquisição.

Time Between Images se trata do tempo de espera entre o término da aquisição de determinada imagem e a próxima, enquanto o imageador se encontrar em modo de aquisição. Este tempo de espera existe para que não sejam geradas imagens de maneira excessiva.

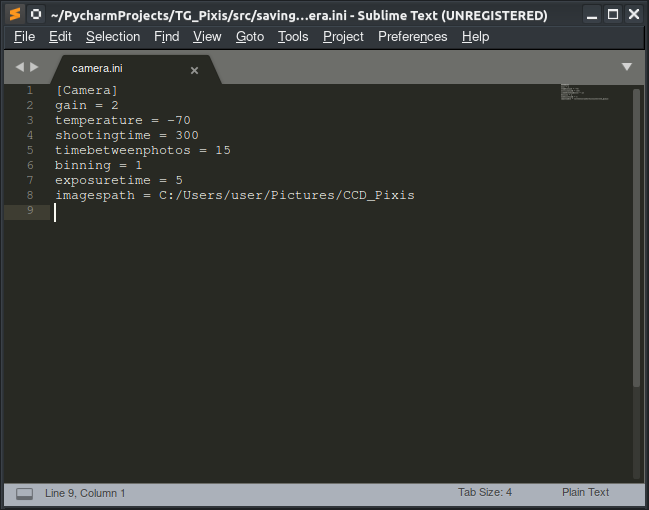
Binning: É um processo de otimização da qualidade de determinada imagem ao custo de resoluções de pixels maiores, que consiste em unir pixels de determinada imagem para obter pixels de melhor qualidade, o que por consequência reduz o tamanho da imagem. Por exemplo: se uma imagem de 1024x1024 pixels de largura passar pelo processo de binning de 2x2, a nova imagem possuirá 512x512 pixels de largura, porém com qualidade melhor. Os níveis de binning suportados pelo imageador em questão são de 1x1, 2x2 e 4x4. (PRINCETON INSTRUMENTS, 2015)

Exposure Time: Consiste no tempo em que a placa de carga acoplada ficará exposta à luz para que seja definida proporcionalmente a riqueza de pixels em cada imagem adquirida.

Images Path: Contém o diretório do computador pessoal do usuário, que o programa deverá guardar as imagens adquiridas.

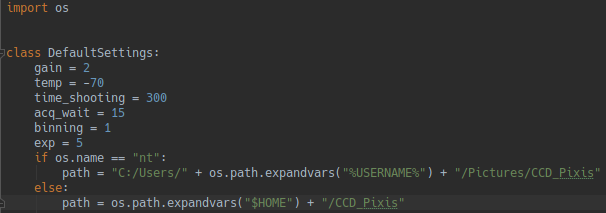
Os parâmetros são persistidos no arquivo camera.ini, conforme a imagem.

**Figura 6: camera.ini**

****

Ao clicar em Defaults na janela de configurações, os valores de todas as entradas de texto e caixas de seleção são alterados para os valores padrão, codificados no arquivo default\_settings.py, conforme a imagem.

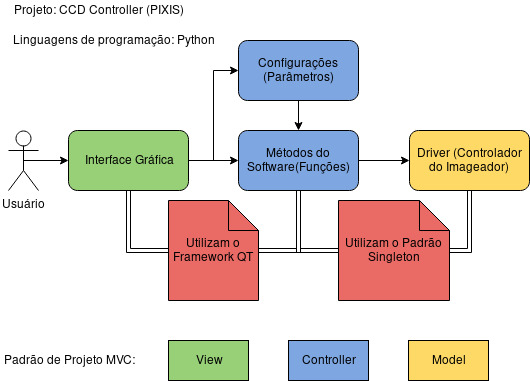
**Figura 7: default\_settings.py**



## Arquitetura

Segue abaixo o diagrama de arquitetura da aplicação.

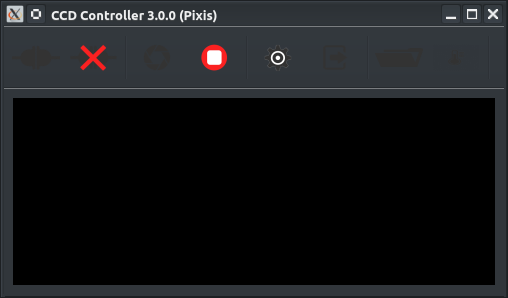
**Figura 8: Diagrama de Arquitetura do Software**



### Interface Gráfica

Contém as rotinas para a exibição da janela principal ao início do programa e do painel de configurações ao clicar no respectivo botão.

**Figura 9: Janela Principal**



(Apresente diagrama(s) de classe(s), comentando cada um das classes principais.)

Trechos de código podem ser apresentados para ilustrar elementos que o autor julgou complexos ou importantes. Tais trechos não devem ocupar mais de 1 página.

Apresenta também definição, execução e resultados de testes (unitários, de integração, etc).

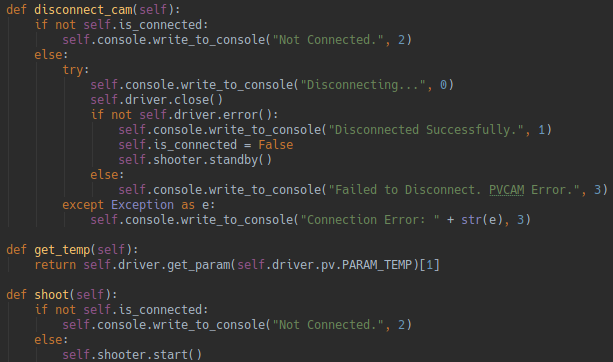
### Configurações

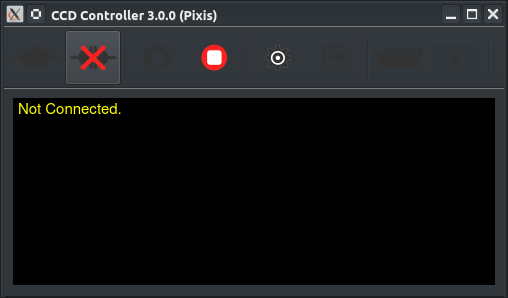
Contém as rotinas de persistência dos valores de parâmetros fornecidos pelo usuário e os valores padrão do programa, além do arquivo destinado a persistir ambos os tipos de valores.

### Métodos do Software

Contém a lógica principal do software, acessível pela janela principal do programa, no caso, as demais funções como conectar o programa ao imageador, disconectá-lo, iniciar aquisição, parar esta, configurações e entre outras.

**Figura 10: Função para desconectar o software à câmera**

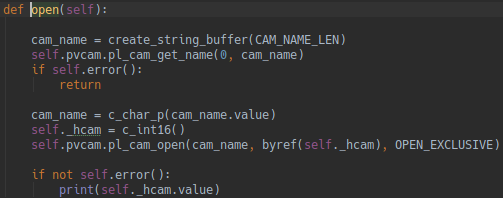
**Figura 11: Retorno da função, caso não haja equipamento conectado**



### Driver

É composto dos arquivos de comunicação com a biblioteca de controle da câmera, que trabalham com as funções e valores aceitáveis para a biblioteca, de forma a utilizá-los de maneira aceitável pelo resto do programa e legível pelo usuário.

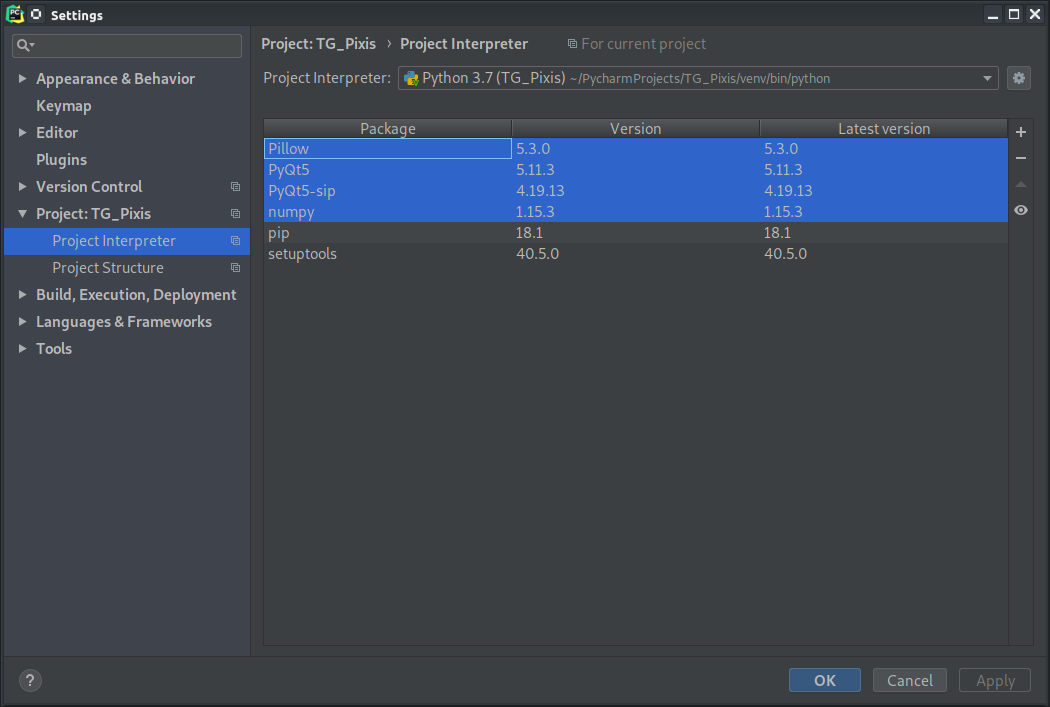
**Figura 12: Função de uso de biblioteca para conexão com a câmera**



## Deploy

O processo de deploy do software consiste em efetuar o download do mesmo em qualquer computador de uso pessoal, instalar os drivers e o SDK das câmeras Princeton contendo a biblioteca de controle PVCAM, para que as funções controladoras da câmera se tornem acessíveis, e também instalar o ambiente para Python e suas dependências necessárias.

**Figura 13: Módulos necessários para o funcionamento do software**



# RESULTADOS E DISCUSSÃO

Mostrar como a solução proposta cumpre cada um dos requisitos levantados.

Comparar a solução proposta com as soluções existentes, ressaltando suas vantagens e desvantagens.

# TRABALHOS FUTUROS

<Necessariamente 1 página>

Indicar possíveis trabalhos futuros. Podem ser funcionalidades que, por conta de limitações de tempo ou recursos, não foram desenvolvidas ou elementos que, no momento, não pertencem ao escopo.

AGRADECIMENTOS

Agradeço ao professor e orientador ..., pelo apoio e encorajamento contínuos pesquisa, aos demais professores, pelos conhecimentos transmitidos, aos meus pais... Na página de agradecimentos o autor dirige palavras de reconhecimento àqueles que contribuíram para a elaboração do trabalho. O conteúdo não deve ultrapassar uma página e por isso, é necessário que ele seja sucinto e objetivo.

APÊNDICES

APÊNDICE A

ANEXOS

ANEXO 1 –