

# **Software com inteligência artificial para detecção automática de manchas na imagem de câmeras durante processo produtivo**

---

**Eduardo Camilo do Canto**

Orientador: Prof. Marcelo De Lellis Costa de Oliveira, Dr.  
Supervisor: Luiz Henrique Oro do Nascimento, Eng.

# Sumário

1. Introdução
2. Problematização
3. Objetivos
4. Solução proposta
5. Modelagem do software
6. Desenvolvimento
7. Resultados
8. Conclusões



# 1. Introdução

intelbras

- Fabricante e distribuidora de produtos eletro-eletrônicos e digitais desde 1976;
- Trabalho desempenhado no setor de câmeras de segurança;
- Matriz e filial Manaus.



*Fig. 1 - Filiais da Intelbras no Brasil*



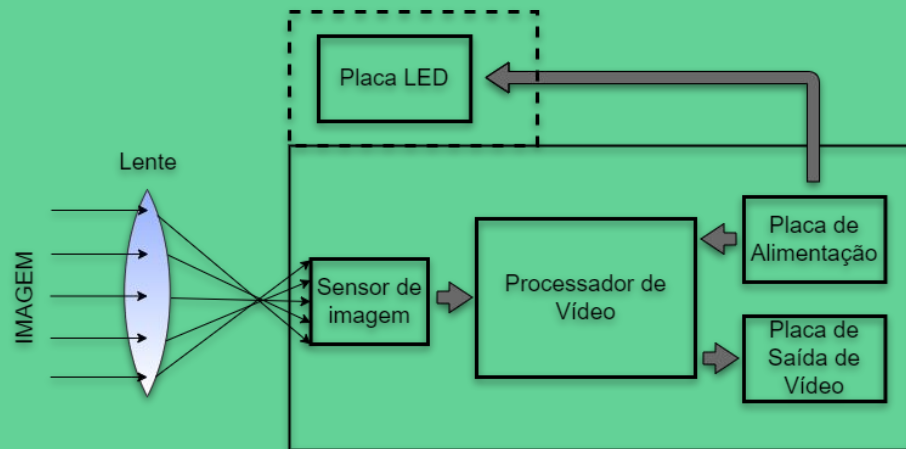
*Fig. 2 - Unidade Matriz - São José/SC*

# 1. Introdução

- Principais componentes de uma câmera de segurança;
- Processo produtivo:
  - Etapa 1: Montagem dos componentes eletrônicos na placa;
  - Etapa 2: Montagem da lente;
  - Etapa 3: Montagem final.



*Fig. 3 - Câmera de segurança*



*Fig. 4 - Esquemático de uma câmera*

## 2. Problematização

- Deposição de partículas no sensor de imagem;
- Problema da mancha escura;
- Dados do processo:

índice de mancha na entrada da 2ª etapa	20%
índice de mancha na saída da 2ª etapa	3%

- Acurácia do operador no teste de mancha: 85%.

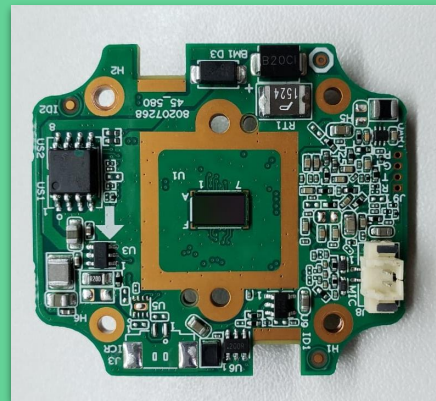


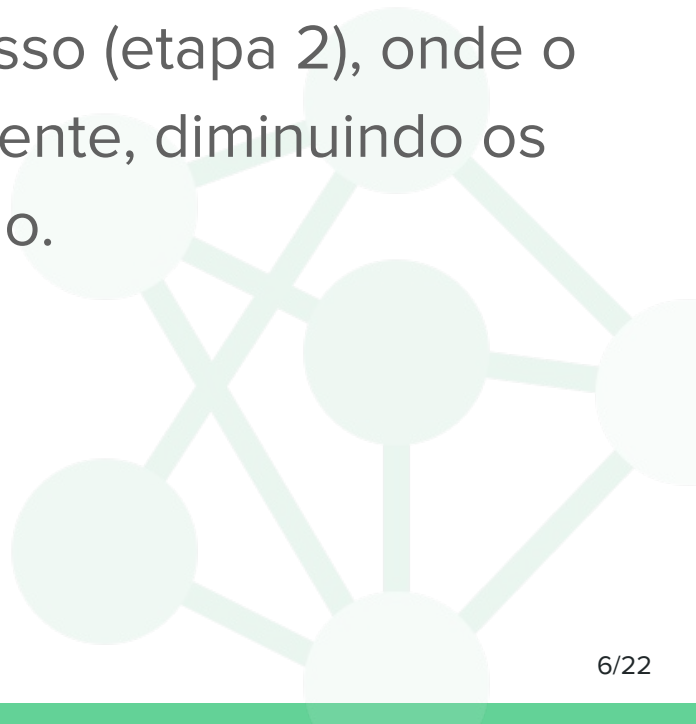
Fig. 5 - Placa principal com sensor ao centro



Fig. 6 - Imagem da câmera com mancha escura

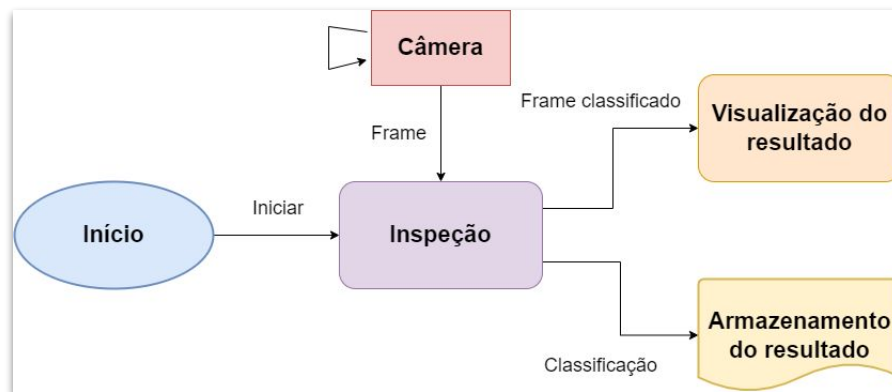
### 3. Objetivo

Detectar manchas no início do processo (etapa 2), onde o retrabalho é realizado mais rapidamente, diminuindo os custos de produção.



## 4. Solução proposta

- Desenvolvimento de algoritmo de inteligência artificial para detecção automática de manchas através de análise de imagens;
- Aplicação no setor de testes após a montagem da lente (etapa 2).



*Fig. 7 - Esquemático do sistema proposto*

# 5. Modelagem do software

- Requisitos de projeto:
  - Funcionais;
  - Não-funcionais.
- Diagramas:
  - Sequência;
  - Classes.
- Objetivos: Auxiliar a implementação do *software* e, posteriormente, em novas funcionalidades e manutenção do código.



# 5. Modelagem do software

## Requisitos de projeto

- Adquirir e analisar as imagens da câmera de forma contínua;
- Apresentar resultados ao operador;
- Gerar relatórios mediante requisição.

Nome: F1 Detectar manchas escuras na imagem		Oculto (X)		
Descrição: O sistema deve analisar as imagens obtidas, tentando identificar a presença de manchas escuras.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF1.1 Tempo de análise	O tempo de análise deverá ser menor do que 1 segundo para cada imagem.	Temporal	()	(X)
NF1.2 Acurácia do sistema	A taxa de acerto do sistema na identificação da presença de mancha deverá ser superior à 99%.	Performance	()	(X)
NF1.3 Início da classificação	O processo de detecção será iniciado logo após a aquisição de uma imagem.	Especificação	()	(X)
NF1.4 Armazenamento da imagem	Mediante opção do operador, as imagens classificadas serão armazenadas para posterior consulta e validação.	Especificação	(X)	()

Nome: F2 Obter imagens da câmera		Oculto (X)		
Descrição: O sistema deve obter as imagens da câmera em análise.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF2.1 Intervalo de obtenção	O sistema obterá uma nova imagem assim que a imagem anterior tiver sido classificada.	Especificação	()	(X)

Nome: F3 Apresentar imagens com avaliação		Oculto ( )		
Descrição: O sistema deve apresentar as imagens obtidas para visualização do usuário.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF3.1 Forma de visualização	A classificação da imagem será disponibilizada em forma textual e visual em cor verde (sem mancha) ou vermelho (com mancha), logo após a classificação.	Interface	()	(X)

Nome: F4 Gerar relatórios		Oculto ( )		
Descrição: Disponibilizar a informação da quantidade de câmeras com mancha no período selecionado.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF4.1 Disponibilização do relatório	Relatórios disponibilizados mediante requisição.	Especificação	()	(X)

Fig. 8 - Tabela de requisitos do software

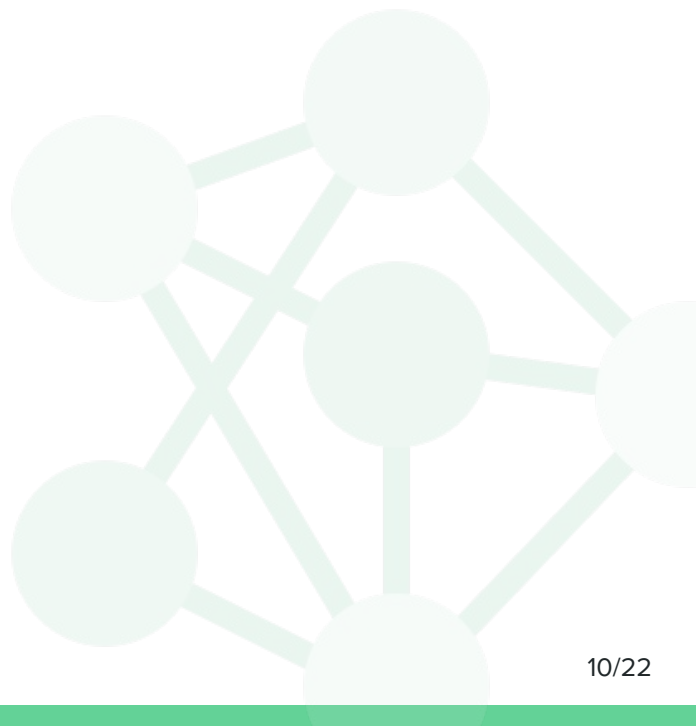
# 6. Desenvolvimento

## 1. Algoritmo de predição de manchas:

- Redes Neurais Convolucionais;
- Banco de imagens;
- Treinamento do modelo;
- Estudo de hiperparâmetros.

## 2. Implementação do *software*:

- Interfaces;
- Model-View-Controller.



# 6.1. Algoritmo de predição de manchas

## Redes Neurais Convolucionais

- Camadas pré-treinadas de convolução: extrair *features*;
- Camadas *Feed-Forward*: classificação;
- ResNets:
  - *Identity shortcut connection*;
  - Bons resultados a um custo computacional baixo.

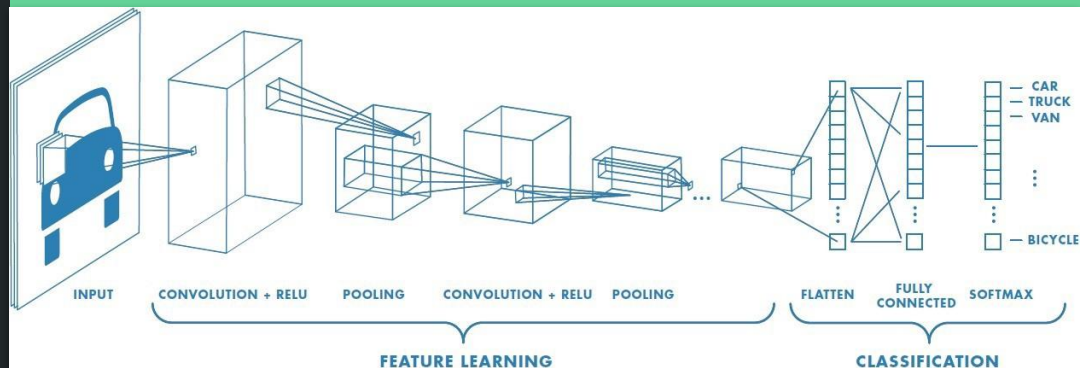


Fig. 9 - Arquitetura de RNCs

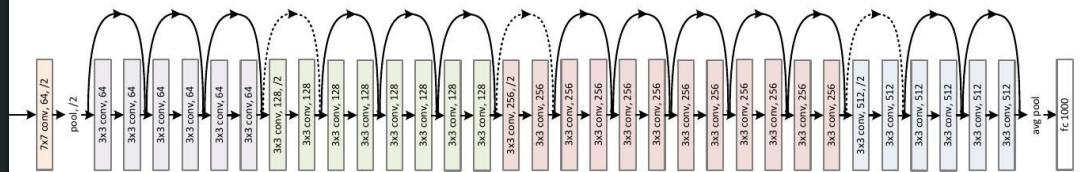


Fig. 10 - Arquitetura de uma ResNet34

# 6.1. Algoritmo de predição de manchas

## Banco de imagens

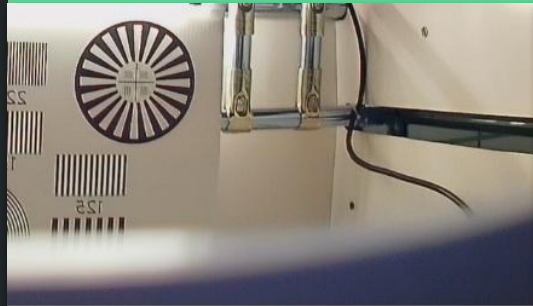
- 4 classes:
  - Mancha;
  - Normal;
  - Inconclusivo;
  - Troca.
- *Data augmentation*: Divisão da imagens em 4 e espalhamento das partes;
- Resolução: 448x252 pixels;
- Total: 1500 imagens.



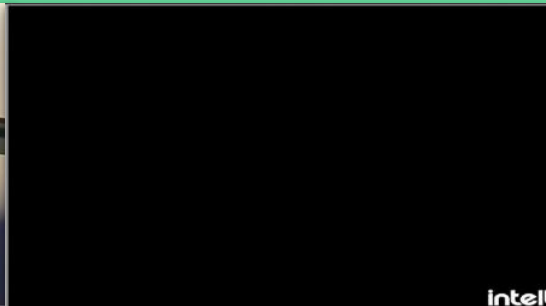
*Fig. 11 - Classe Mancha*



*Fig. 12 - Classe Normal*



*Fig. 13 - Classe Inconclusivo*

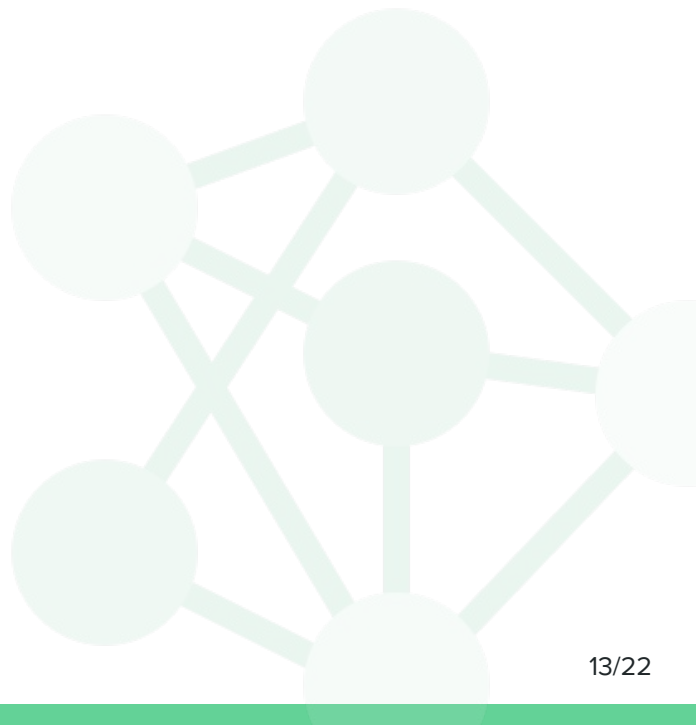


*Fig. 14 - Classe Troca*

## 6.1. Algoritmo de predição de manchas

### Treinamento do modelo

- Utilização da ResNet18;
- Banco de imagens:
  - 80% para treino;
  - 20% para validação.
- *Batch size*: 12 imagens;
- Taxa de aprendizagem variável;
- Ajuste fino do modelo.



# 6.1. Algoritmo de predição de manchas

## Estudo de hiperparâmetros

- Análise de diferentes valores de *batch size*: 1, 8, 16, 32 e 64;
- Baixos valores de *batch size* tendem a *underfitting*;
- Altos valores de *batch size* tendem a *overfitting*;
- Faixa ideal: 8 a 16.

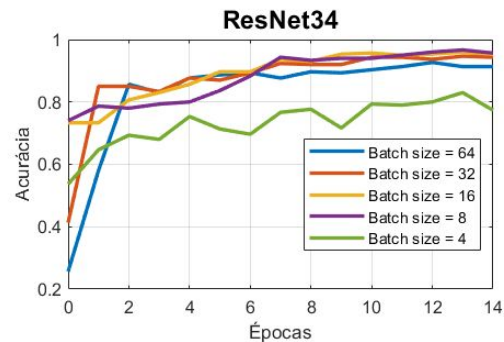
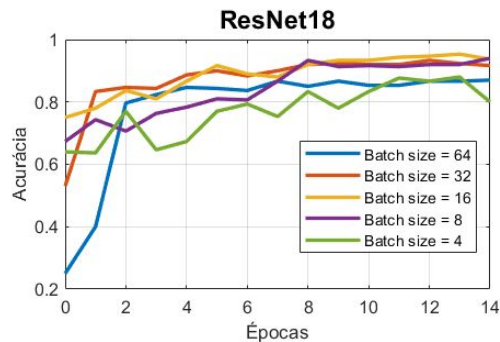


Fig. 15 - Evolução da acurácia durante o treinamento da rede

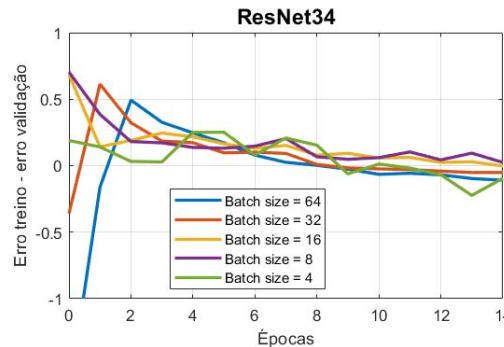
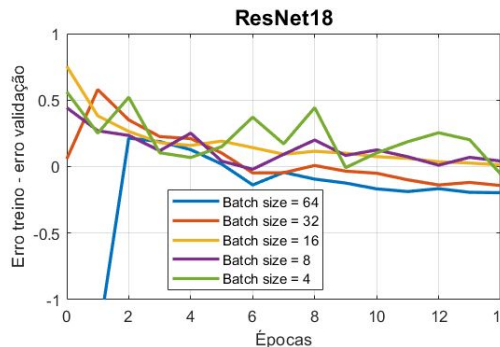


Fig. 16 - Evolução da diferença dos erros durante o treinamento da rede

# 6.1. Algoritmo de predição de manchas

## Estudo de hiperparâmetros

- Análise de diferentes valores de resolução: 640x360, 427x240 e 320x180 *pixels*;
- Diferença sutil de acurácia para diferentes resoluções;
- Desempenho similar entre a ResNet18 e ResNet34.

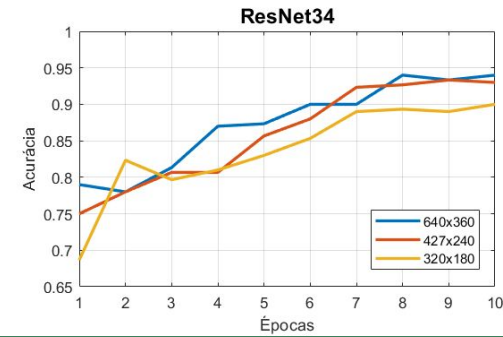
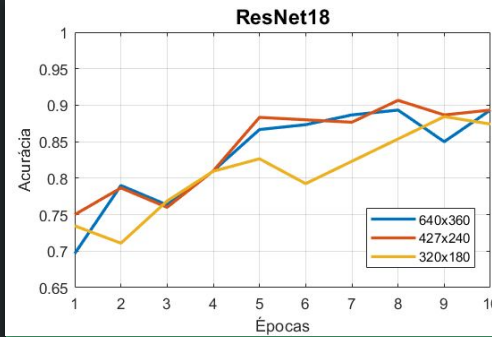


Fig. 17 - Evolução da acurácia durante o treinamento da rede

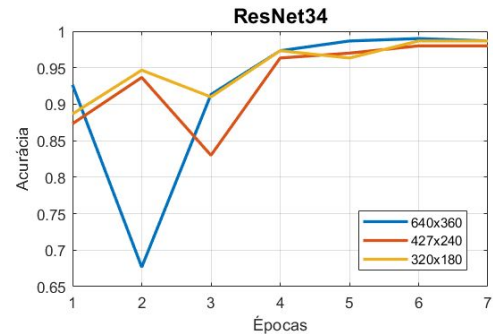
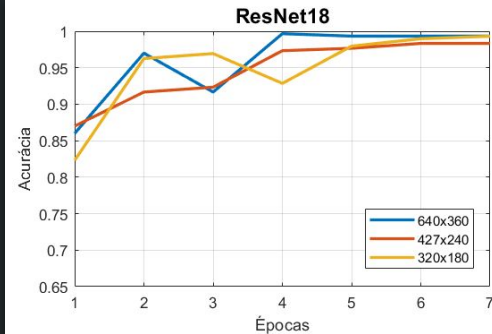
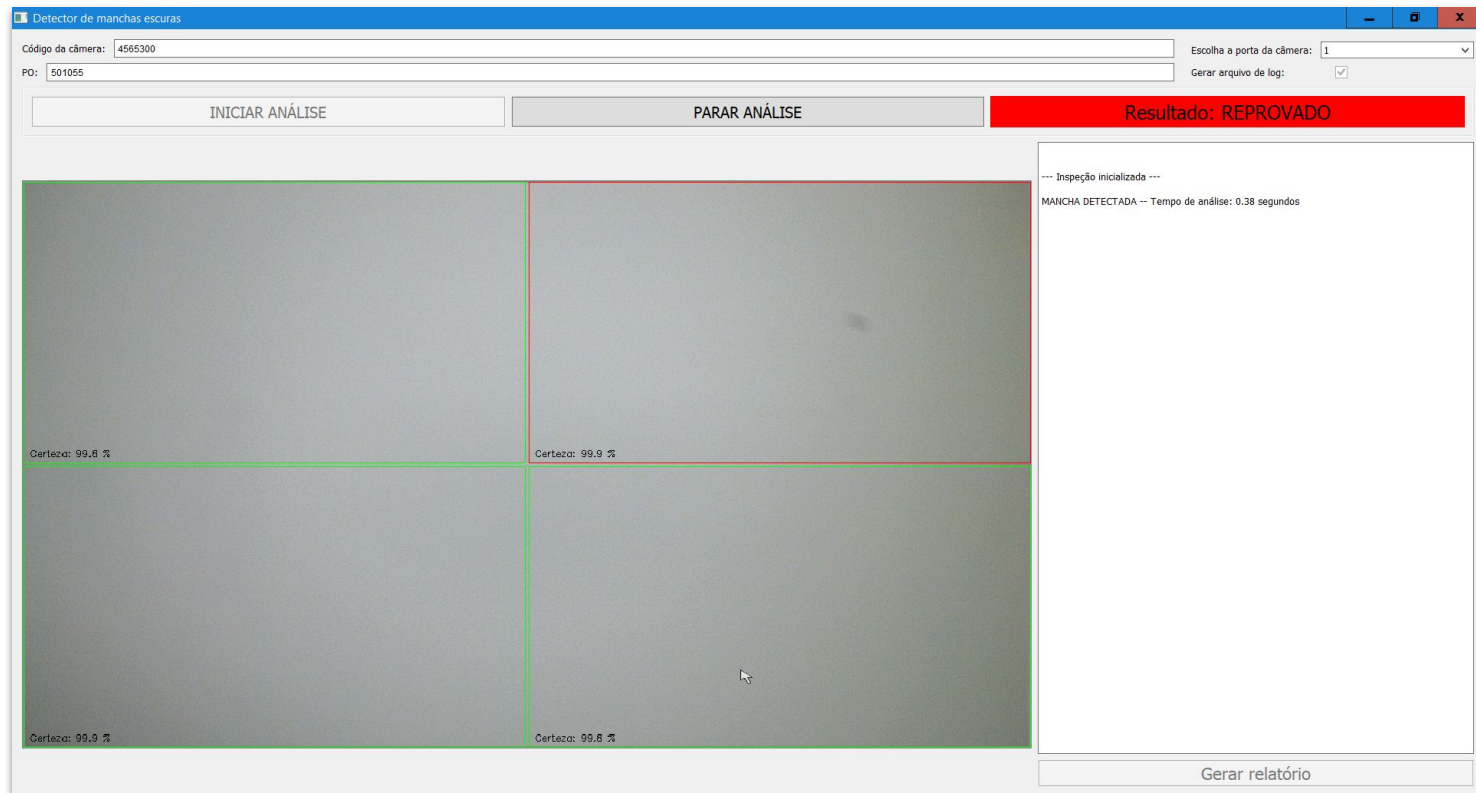


Fig. 18 - Evolução da acurácia durante ajuste fino da rede



## 6.2. Implementação do *software*

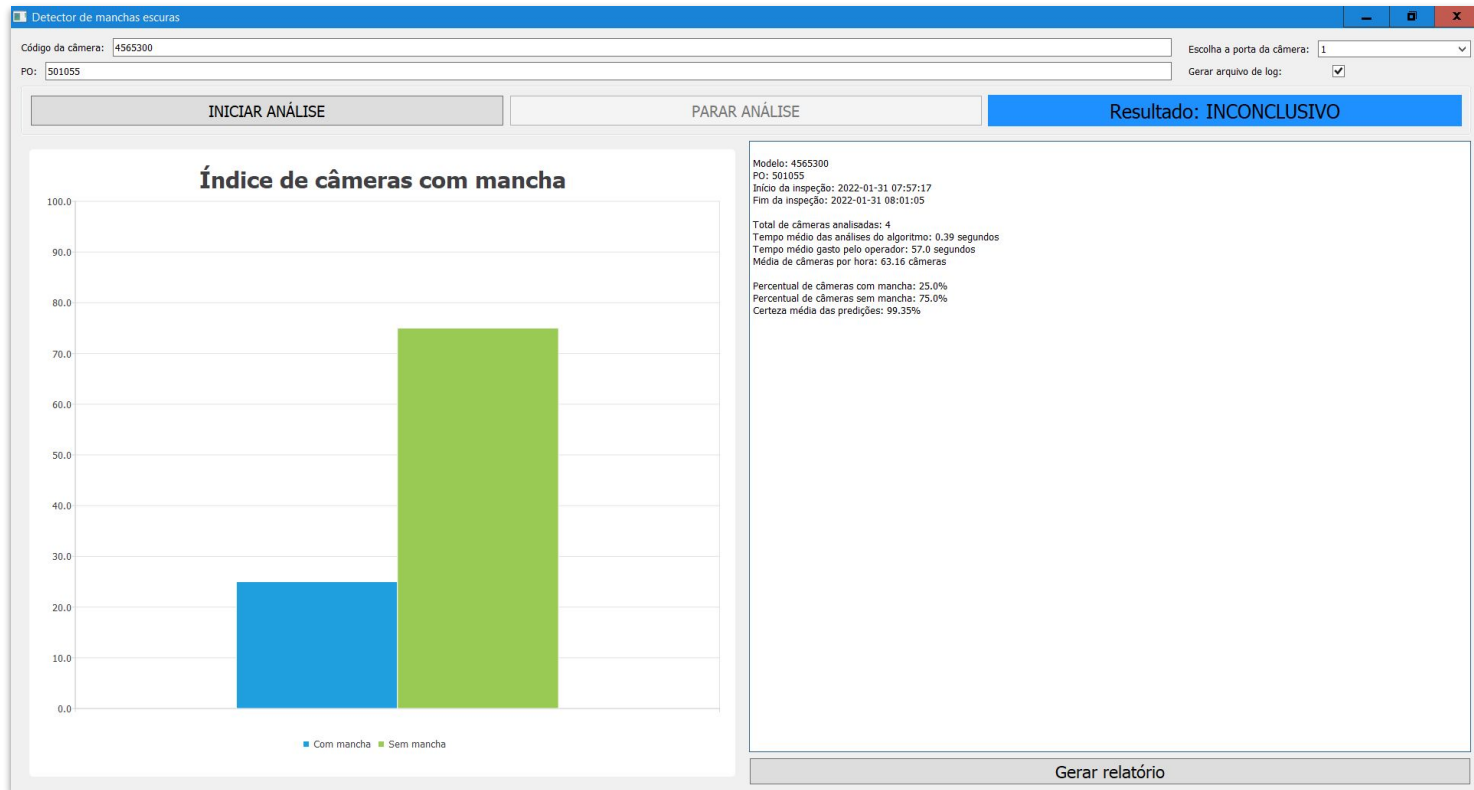
### Interfaces





# 6.2. Implementação do software

## Interfaces



## 6.2. Implementação do *software*

### **Model-View-Controller**

- Implementação conforme os requisitos e diagramas da Seção 5;
- Lógica para identificar a troca de câmera e evitar armazenamento duplicado dos resultados;
- Classificação de imagem: aquisição, pré-processamento e predição;
- Geração de relatórios (XML);
- Geração de arquivo .exe para distribuição.

# 7. Resultados

## Algoritmo de predição

- Estabilização da acurácia próximo a 92% com ajuste preliminar;
- Atingimento de 100% de acurácia após a realização do ajuste fino;

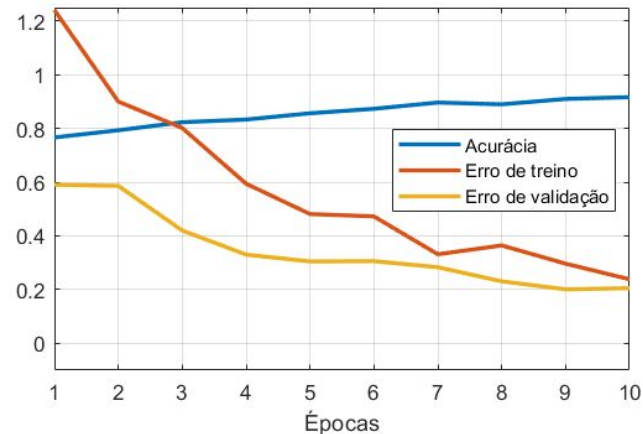


Fig. 19 - Resultados preliminares do algoritmo de predição

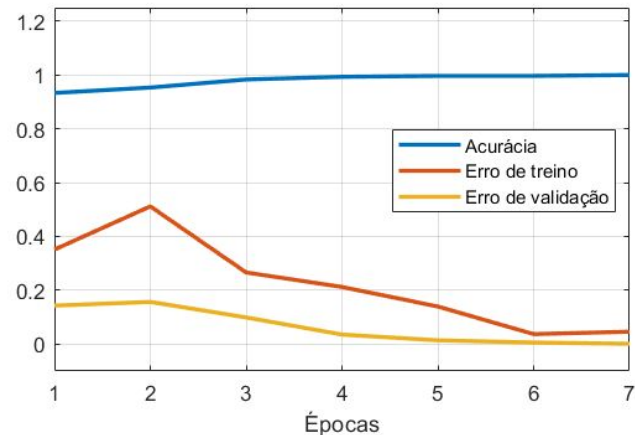


Fig. 20 - Resultados do algoritmo de predição após o ajuste fino

# 7. Resultados

## Algoritmo de predição

- Classificação correta de todas as 300 imagens;
- Erro nulo mesmo com a arquitetura com menor número de camadas (18) das ResNets;
- Tempo de predição: 0,4 s.

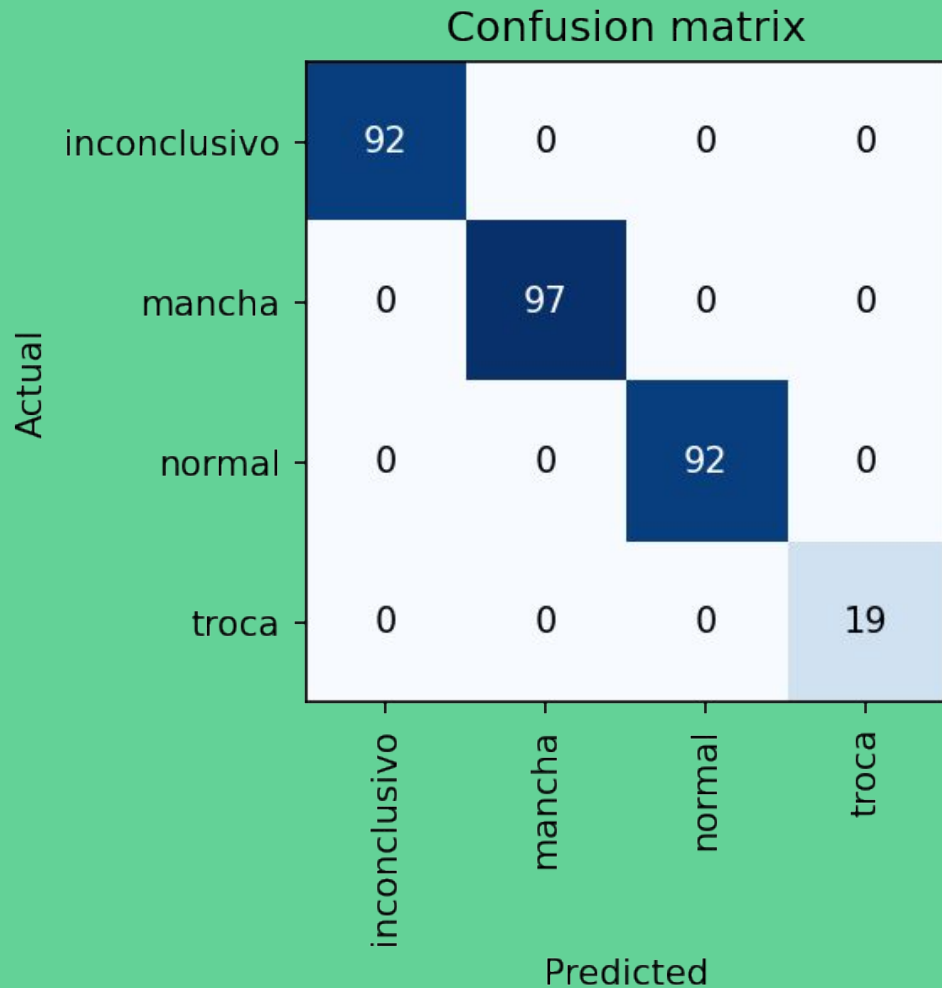


Fig. 21 - Matriz de confusão do conjunto de validação

# 7. Resultados

## Implementação na fábrica

- Acompanhamento do funcionamento do sistema na linha por 3 meses;
- Funcionamento em 2 das 14 células, atualmente, com implementação em andamento nas demais;
- **Economia anual estimada: R\$400 mil.**

# 8. Conclusões

- Cumprimento de todos os objetivos propostos;
- Sugestões de projetos futuros:
  - Replicação da abordagem na automatização de todos os testes da Montagem Final (etapa 3);
  - Automação completa na etapa 2 (teste de mancha + ajuste de foco).

# Obrigado!

# 5. Modelagem do software

## Requisitos

- Detectar manchas na imagem:
  - Acurácia maior que 99%;
  - Tempo de análise menor que 1 s;
  - Início após
- Obter imagens da câmera:
  - Nova obtenção após classificação anterior.
- Apresentar imagem classificada:
  - Resultado em forma textual e visual.
- Gerar relatórios:
  - Disponibilização mediante requisição.

Nome: F1 Detectar manchas escuras na imagem		Oculto (X)		
Descrição: O sistema deve analisar as imagens obtidas, tentando identificar a presença de manchas escuras.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF1.1 Tempo de análise	O tempo de análise deverá ser menor do que 1 segundo para cada imagem.	Temporal	()	(X)
NF1.2 Acurácia do sistema	A taxa de acerto do sistema na identificação da presença de mancha deverá ser superior à 99%.	Performance	()	(X)
NF1.3 Início da classificação	O processo de detecção será iniciado logo após a aquisição de uma imagem.	Especificação	()	(X)
NF1.4 Armazenamento da imagem	Mediante opção do operador, as imagens classificadas serão armazenadas para posterior consulta e validação.	Especificação	(X)	()

Nome: F2 Obter imagens da câmera		Oculto (X)		
Descrição: O sistema deve obter as imagens da câmera em análise.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF2.1 Intervalo de obtenção	O sistema obterá uma nova imagem assim que a imagem anterior tiver sido classificada.	Especificação	( )	(X)

Nome: F3 Apresentar imagens com avaliação		Oculto ( )		
Descrição: O sistema deve apresentar as imagens obtidas para visualização do usuário.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF3.1 Forma de visualização	A classificação da imagem será disponibilizada em forma textual e visual em cor verde (sem mancha) ou vermelho (com mancha), logo após a classificação.	Interface	( )	(X)

Nome: F4 Gerar relatórios		Oculto ( )		
Descrição: Disponibilizar a informação da quantidade de câmeras com mancha no período selecionado.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF4.1				
Disponibilização do relatório	Relatórios disponibilizados mediante requisição.	Especificação	()	(X)

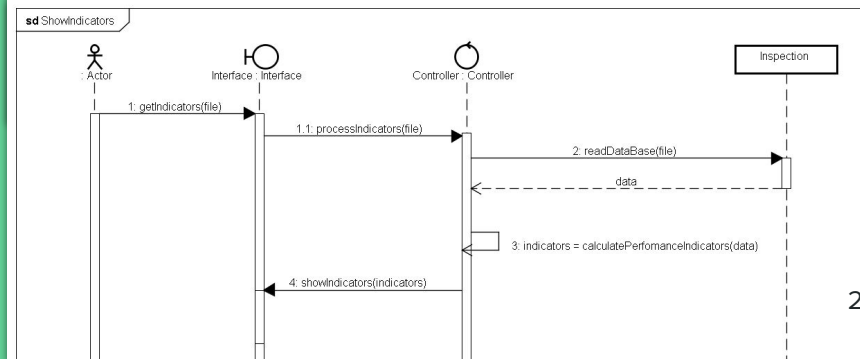
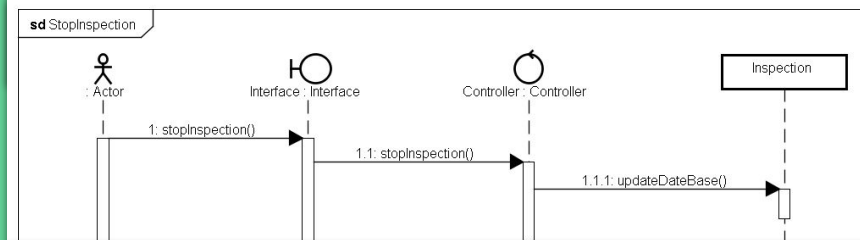
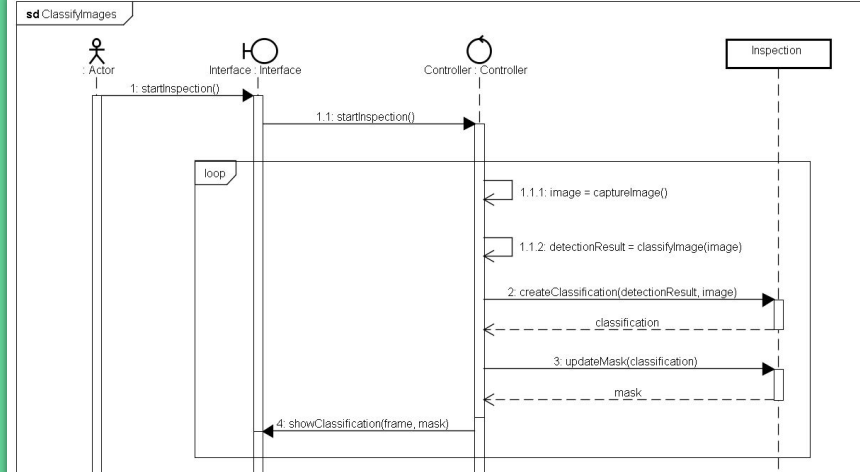
Tabela de requisitos do software



# 5. Modelagem do software

## Diagramas de sequência

- Obtenção da imagem, classificação e disponibilização do resultado.
- Dados da inspeção atual são salvos e o programa para de coletar e classificar imagens.
- Leitura dos dados de uma inspeção e cálculo dos indicadores para disponibilização.



# 5. Modelagem do software

## Diagrama de classes

- *Inspection*
  - *Mask*
  - *Classification*
    - *LocalClassification*
- Classes auxiliares:
  - *DetectionResult*;
  - *ImageTools*;
  - *Location*.

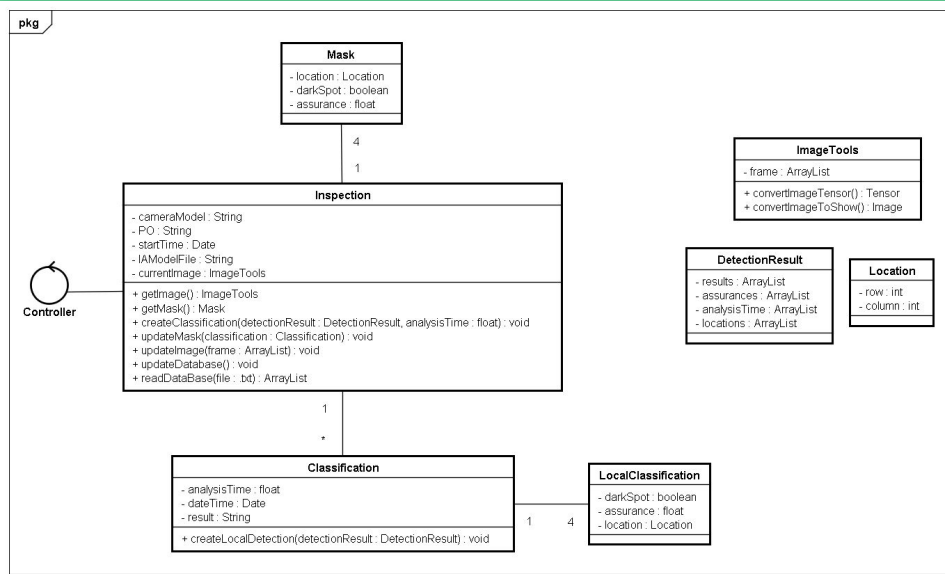


Diagrama de classes do software

## 6.2. Implementação do *software*

### Relatórios

- Geração de arquivo em XML ao terminar a inspeção;
- Leitura do arquivo e disponibilização dos resultados.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<inspection>
```

```
<model>4565299</model>
```

```
<PO>500042</PO>
```

```
<start_time>2021-12-21 14:00:09.044115</start_time>
```

```
<stop_time>2021-12-21 14:08:16.066248</stop_time>
```

```
<classifications>
```

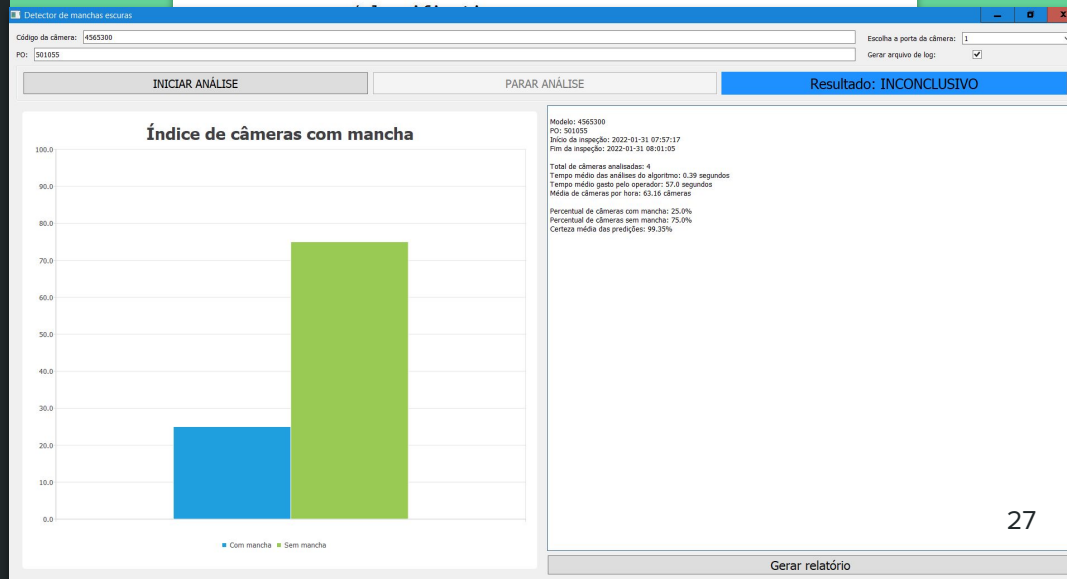
```
<classification>
```

```
<analysis_time>0.5954117774963379</analysis_time>
```

```
<date_time>2021-12-21 14:00:18.588241</date_time>
```

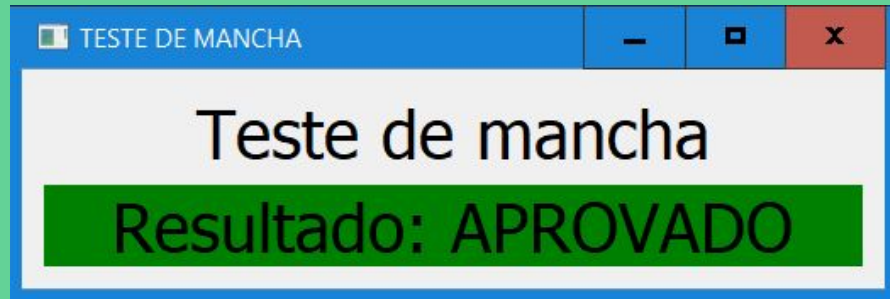
```
<result>mancha</result>
```

```
<assurance>0.9338309466838837</assurance>
```

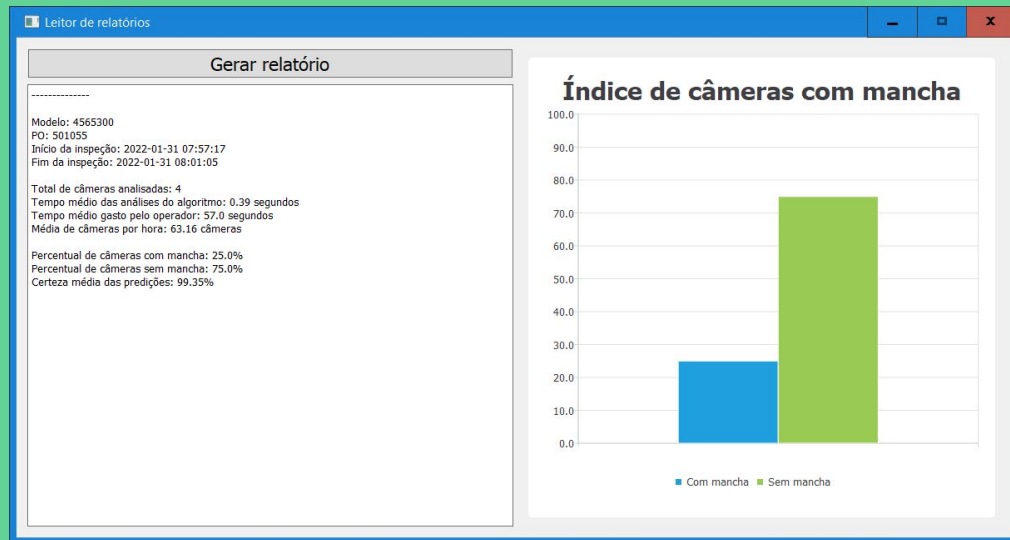


## 6.3. Implementação na fábrica

- Execução paralela ao sistema corrente;
- Interface compacta;
- Relatórios *software* separado;
- Geração de arquivos executáveis para execução na fábrica.



*Interface do software implementado na fábrica*



*Nova interface de relatórios*