

# RECONHECIMENTO DE PLACA POR CÂMERA

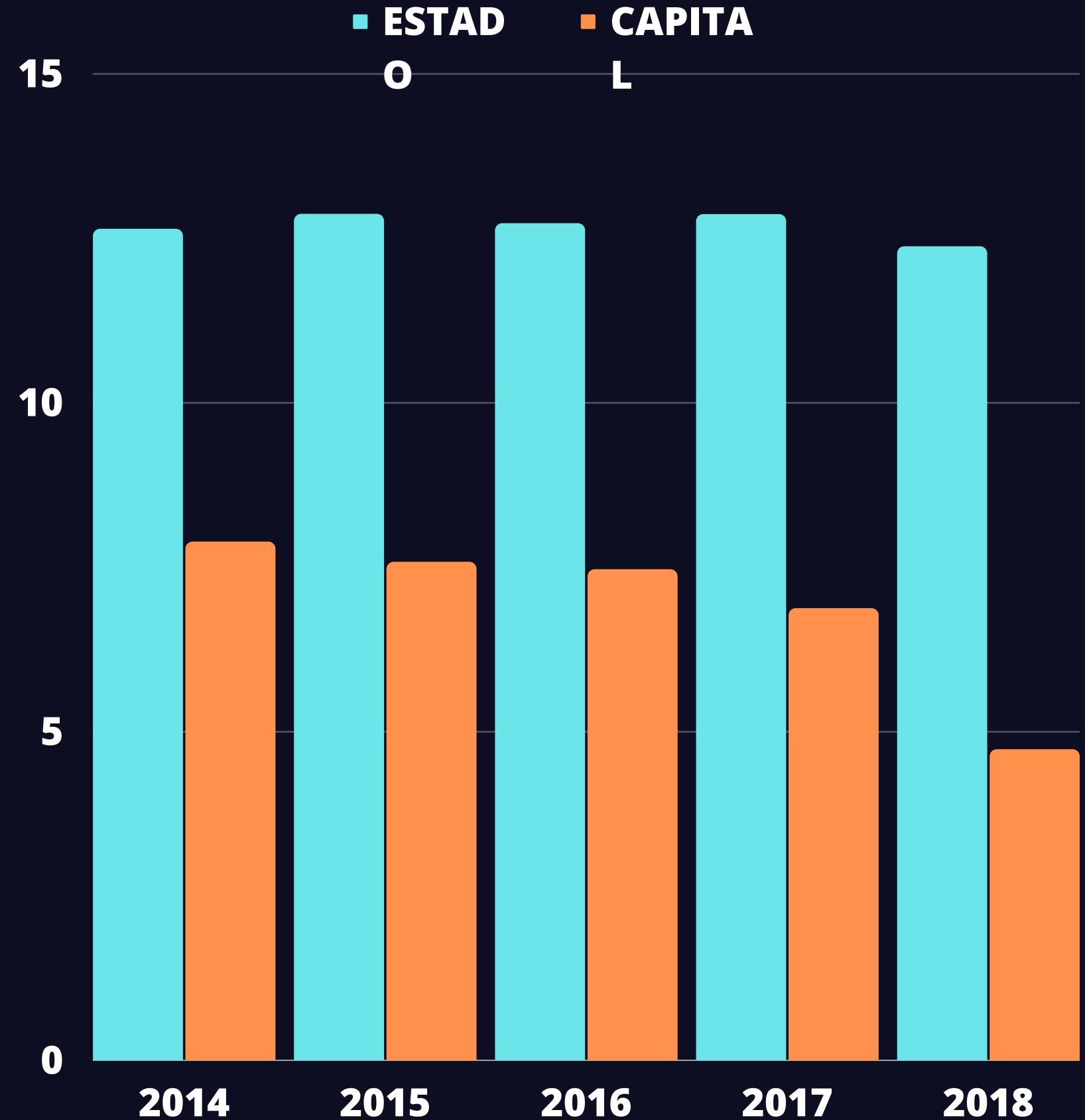
## PROJETO 1

- Thiago Henrique do Couto Pereira  
– 02010475
- Uriel Gonçalves Paiva da Conceição  
– 02010287



# INTRODUÇÃO

- Na atual situação do nosso país, o controle de acesso de usuários é indispensável, isto é, o gerenciamento de acesso em diferentes ambientes, garantindo a segurança e integridade das pessoas, com principal objetivo, bloquear os indivíduos não autorizados



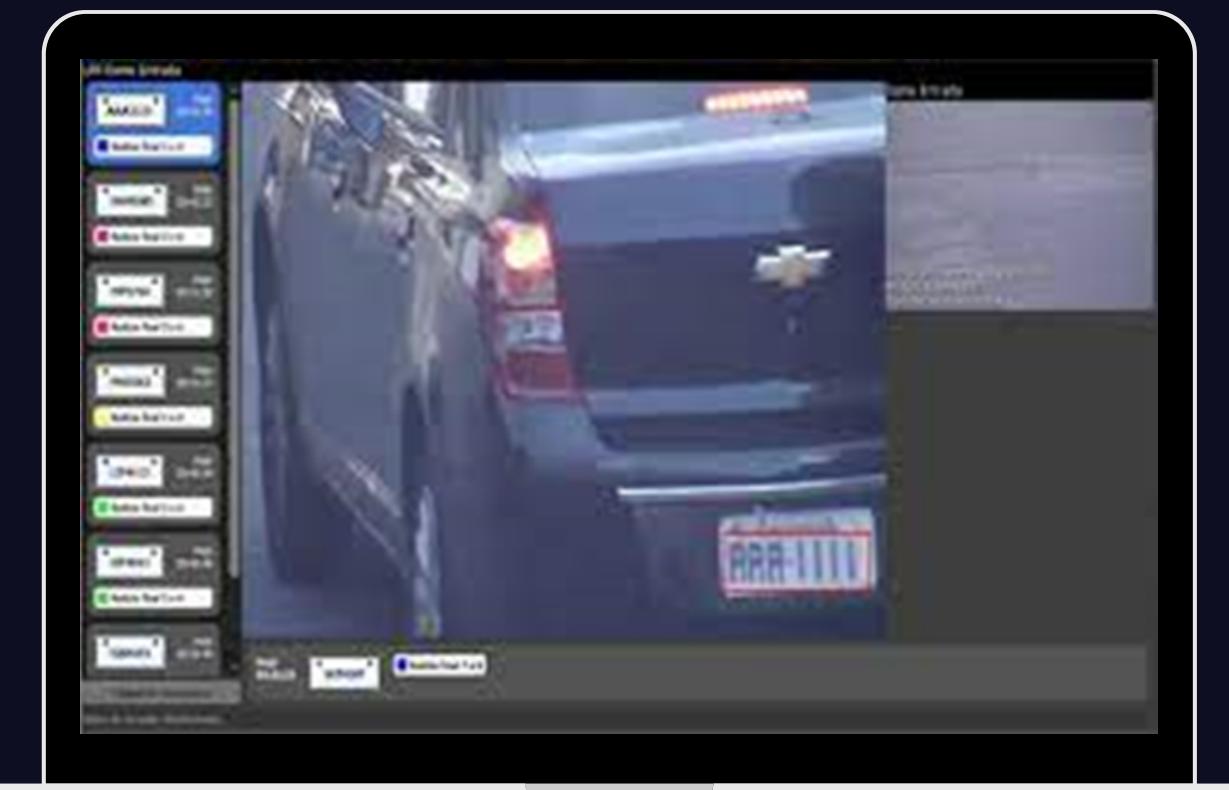
realizado pelo Fiquem Sabendo com informações da Polícia Civil do Estado de São Paulo. Os dados obtidos tiveram como base a Lei Federal nº 12.527/2011 (Lei de Acesso à Informação).

# INTRODUÇÃO

- o software de controle de acesso é mais utilizado em ambientes com uma grande quantidade de movimento como exemplo condomínios, hospitais, instituições de ensino e etceteras. Porem também pode se ter sua utilização em menor escala como exemplo o estacionamento de um estabelecimento ou a garagem de algum domicilio.
- O software tem como principal função identificar a placa e trabalhar a imagem com agilidade para comparar a leitura no presente momento com o código cadastrado no sistema pelo operador, para que assim o sistema identifique e mande uma resposta de liberação ou bloqueio de acordo com o banco de dados.

# OBJETIVO

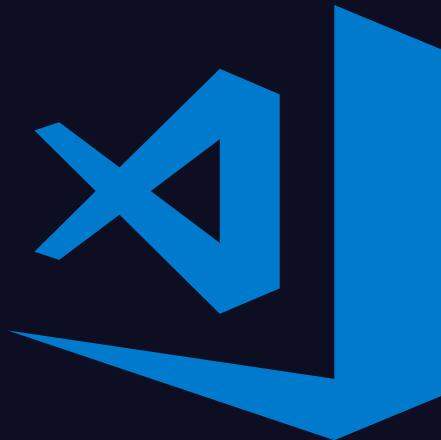
- O programa tem como objetivo processar imagens de carros e extrair informações relevantes, como a placa de identificação, para permitir uma melhor gestão e controle de tráfego.
- O programa foi desenvolvido em Python e utiliza técnicas de processamento de imagem e reconhecimento de padrões para identificar as placas de diferentes países e formatos.



# FERRAMENTAS UTILIZADAS



|

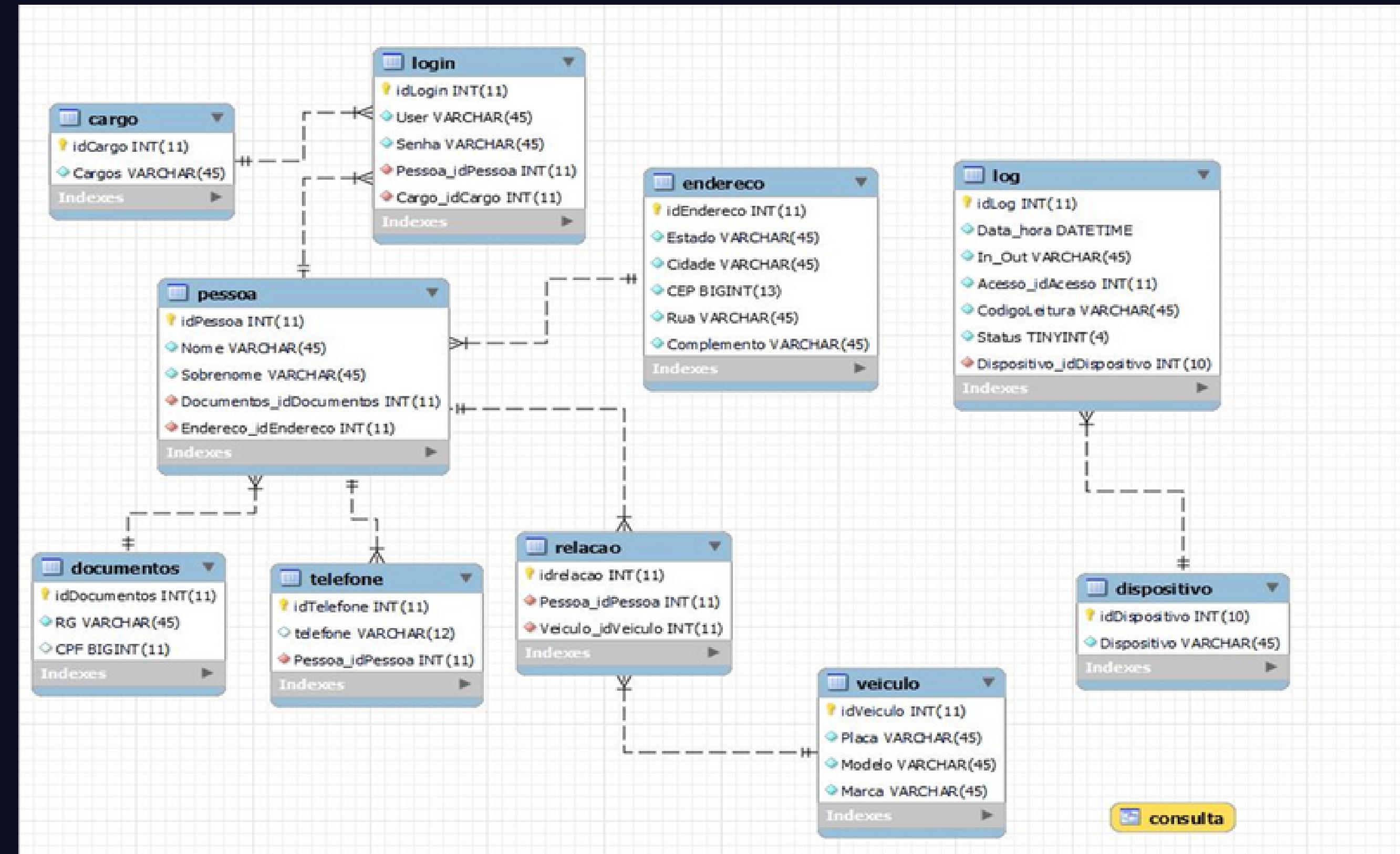
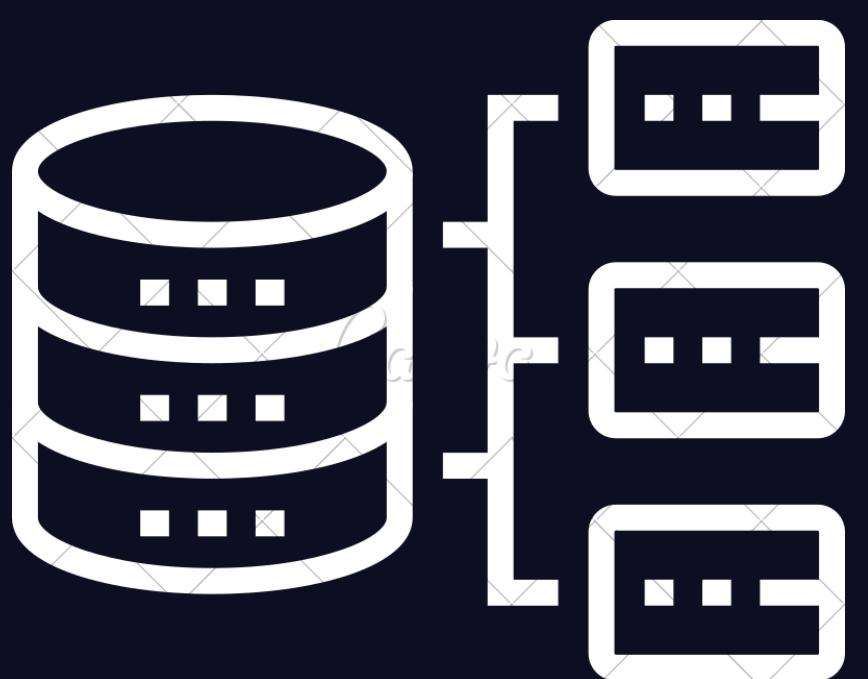


Visual Studio Code



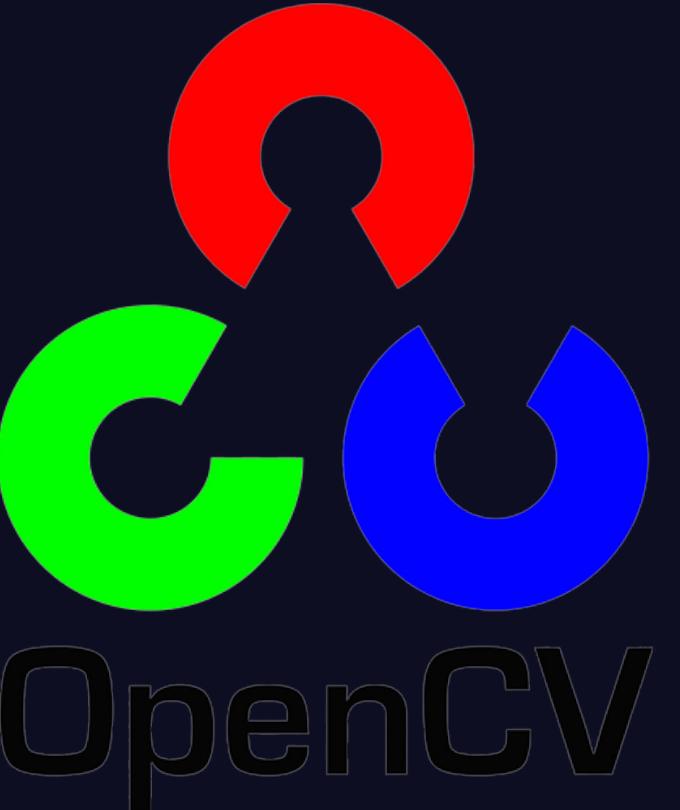
python™

# DIAGRAMA ENTIDADE-RELACIONAMENTO



# METODOLOGIAS

- No contexto do controle de acesso por placa de veículo, o OpenCV foi utilizado para extrair a placa de veículo das imagens capturadas pela câmera e ajustar a imagem salva em um arquivo na estação local para realizar a leitura pela biblioteca Tesseract.
- Após a captura da imagem foram realizadas as seguintes etapas de processamento da imagem contendo a placa de veículo



# METODOLOGIAS

## 1 Ajuste da imagem em escala de cinza:

- Nessa etapa, a imagem colorida é convertida para escala de cinza. Isso é feito utilizando a função cv2.cvtColor() do OpenCV, que converte a imagem para o espaço de cores em escala de cinza.



# METODOLOGIAS

2 Transformação da imagem em matriz binária:

- Após o ajuste em escala de cinza, a imagem é binarizada, convertendo os pixels em apenas duas cores: preto e branco.
- Isso é realizado por meio de técnicas de limiarização (thresholding) utilizando a função cv2.threshold() do OpenCV. O objetivo é tornar a placa de veículo mais destacada e facilitar a detecção dos caracteres.



# METODOLOGIAS

## 3 Ajuste da saturação em preto e branco:

- Nessa etapa, a imagem binarizada é ajustada para aumentar o contraste entre as áreas escuras (placa de veículo) e as áreas claras (fundo). Isso é feito utilizando técnicas de equalização de histograma, por meio da função cv2.Thresh\_otsu() mais o cv2.Thresh\_binary() do OpenCV. O objetivo é melhorar a legibilidade dos caracteres da placa.
- Essas etapas de pré-processamento da imagem no OpenCV ajudam a melhorar a qualidade e a legibilidade da placa de veículo, permitindo um reconhecimento mais preciso dos caracteres por



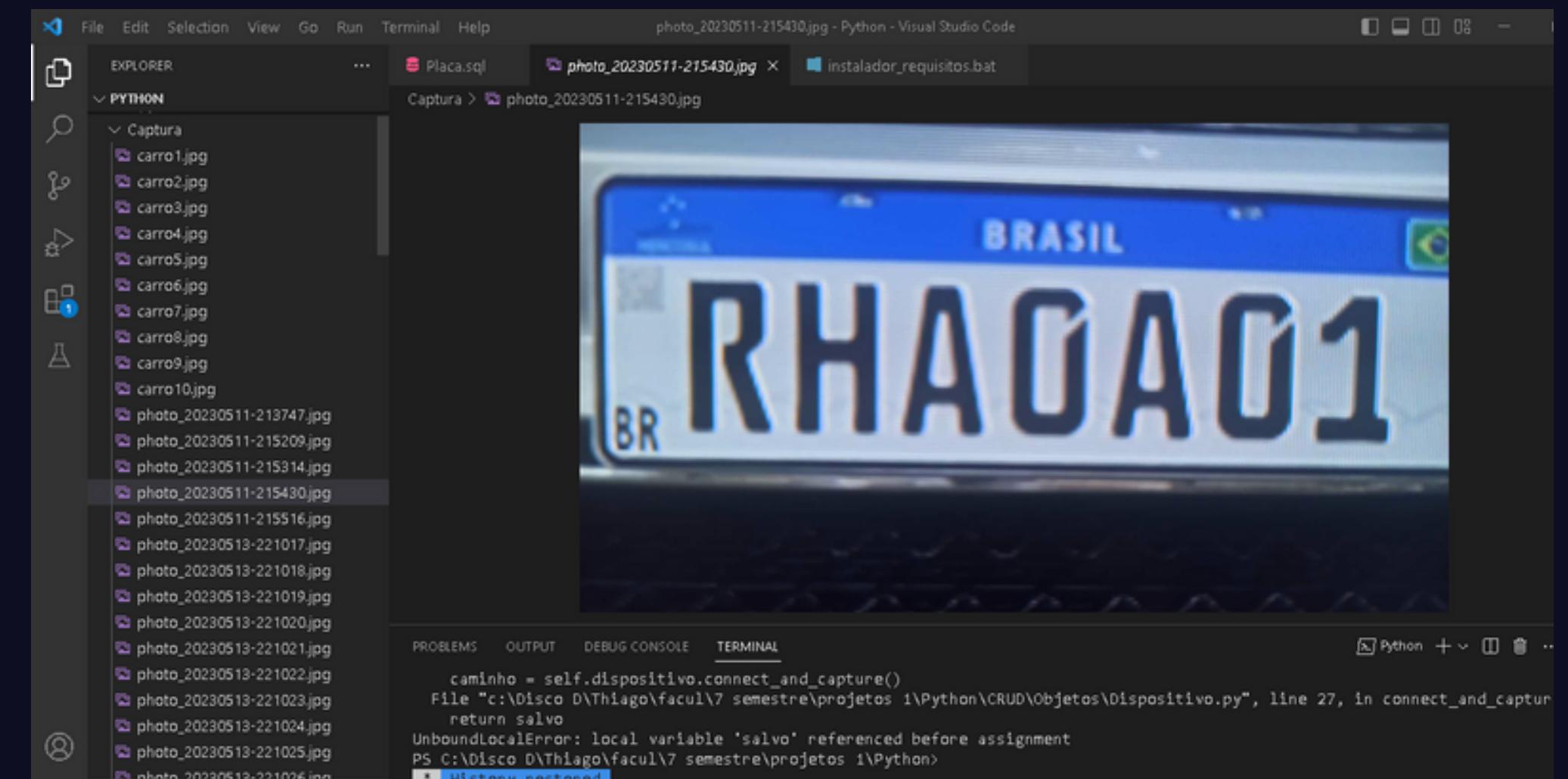
# METODOLOGIAS



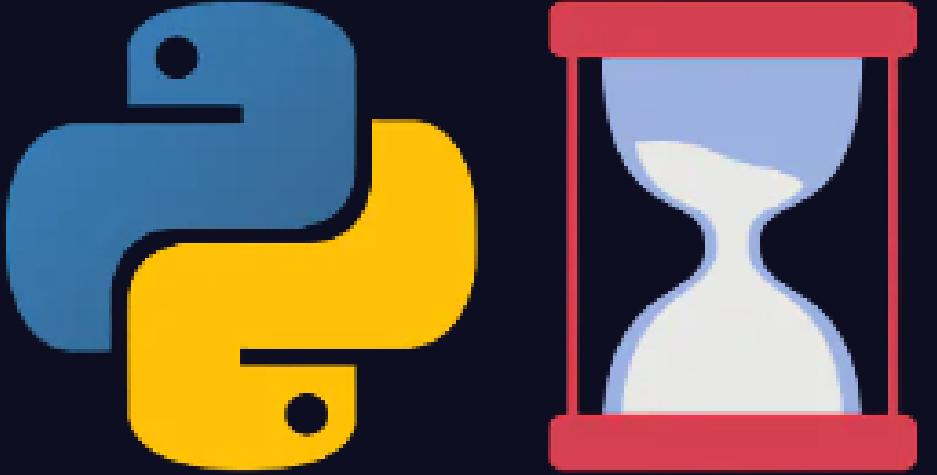
- O Tkinter é uma biblioteca gráfica padrão do Python que permite a criação de interfaces gráficas de usuário (GUI). Neste projeto, Tkinter foi utilizado para criar a interface do programa, incluindo janelas, botões e caixas de texto.
- Através do Tkinter, foi possível fornecer uma interface amigável para interação com o usuário

# METODOLOGIAS

- Foi utilizada a biblioteca OS pois a mesma fornece uma interface para interagir com o sistema operacional.
- Neste projeto, a biblioteca OS foi utilizada para acessar e manipular os arquivos do sistema, permitindo a leitura e gravação das imagens capturadas pela câmera.



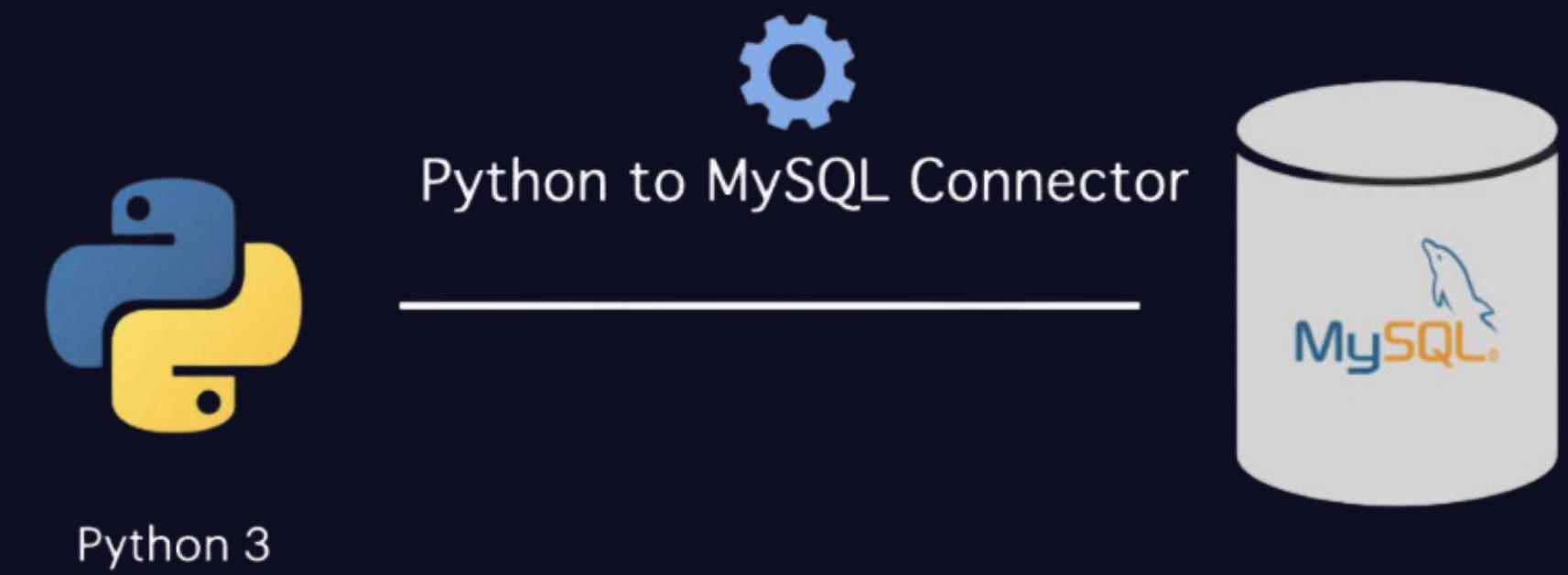
# METODOLOGIAS



- A biblioteca Time fornece funções relacionadas à medição e manipulação de tempo.
- Neste projeto, a biblioteca Time foi utilizada para implementar temporizações, como pausas e atrasos, para controlar o fluxo do programa.

# METODOLOGIAS

- A biblioteca MySQL.connector é essencial para se conectar e interagir com bancos de dados MySQL em Python.



- Neste projeto, a biblioteca foi utilizada para armazenar e recuperar dados relacionados aos veículos e seus respectivos acessos.

- Ela permite a execução de consultas SQL para inserção, atualização e recuperação de informações do banco de dados.

# METODOLOGIAS

- O Tesseract é uma biblioteca de OCR de código aberto amplamente utilizada, capaz de extrair texto de imagens.
- Ele examina cada região da imagem, identifica os padrões dos caracteres e os interpreta para gerar o texto correspondente
- No contexto do controle de acesso por placa de veículo, o Tesseract é empregado para identificar os caracteres presentes na imagem da placa binarizada e convertê-los em texto legível.



# METODOLOGIAS

- Com a Pillow, é possível redimensionar imagens, ajustar a resolução e realizar operações de escalamento.
- Essas funcionalidades são úteis para adaptar as imagens capturadas pela câmera para o tamanho adequado e resolução desejada, assim exibindo a imagem da câmera na tela de controle de acesso.



pillow

# METODOLOGIAS



- A biblioteca "RE" em Python é utilizada para trabalhar com expressões regulares, permitindo a manipulação e validação de padrões de texto. No contexto de manipulação de CPF (Cadastro de Pessoa Física).
- A biblioteca "RE" pode ser utilizada para verificar se um CPF fornecido está em um formato válido e para realizar manipulações específicas nesse tipo de dado.

# TELÀ DE LOGIN



Login

User:

Password:

Login

A screenshot of a Windows-style login window titled "Login". It contains two text input fields labeled "User:" and "Password:", both represented by empty white boxes with black outlines. Below the password field is a green rectangular button with the word "Login" in white capital letters.

# TELA DE MENU PRINCIPAL



# TELA DE OPERADORES



Cadastro de Operadores

buscar Funcionario (CPF):  buscar

Nome:  Sobrenome:

RG:  CPF:

Estado:  Cidade:

CEP:  Rua:

Complemento:  Telefone:

Usuario:

Cargo:  Senha:

# TEL A DE CADASTRO DE USUÁRIO



Cadastro de Placas

buscar Placa:	<input type="text"/>	buscar	
Nome:	<input type="text"/>	Sobrenome:	<input type="text"/>
RG:	<input type="text"/>	CPF:	<input type="text"/>
Estado:	<input type="text"/>	Cidade:	<input type="text"/>
CEP:	<input type="text"/>	Rua:	<input type="text"/>
Complemento:	<input type="text"/>	Telefone:	<input type="text"/>
Placa:	<input type="text"/>	Modelo:	<input type="text"/>
Marca:	<input type="text"/>		

# TELA DE CADASTRO DE CÂMERAS



Registro de Câmeras IP

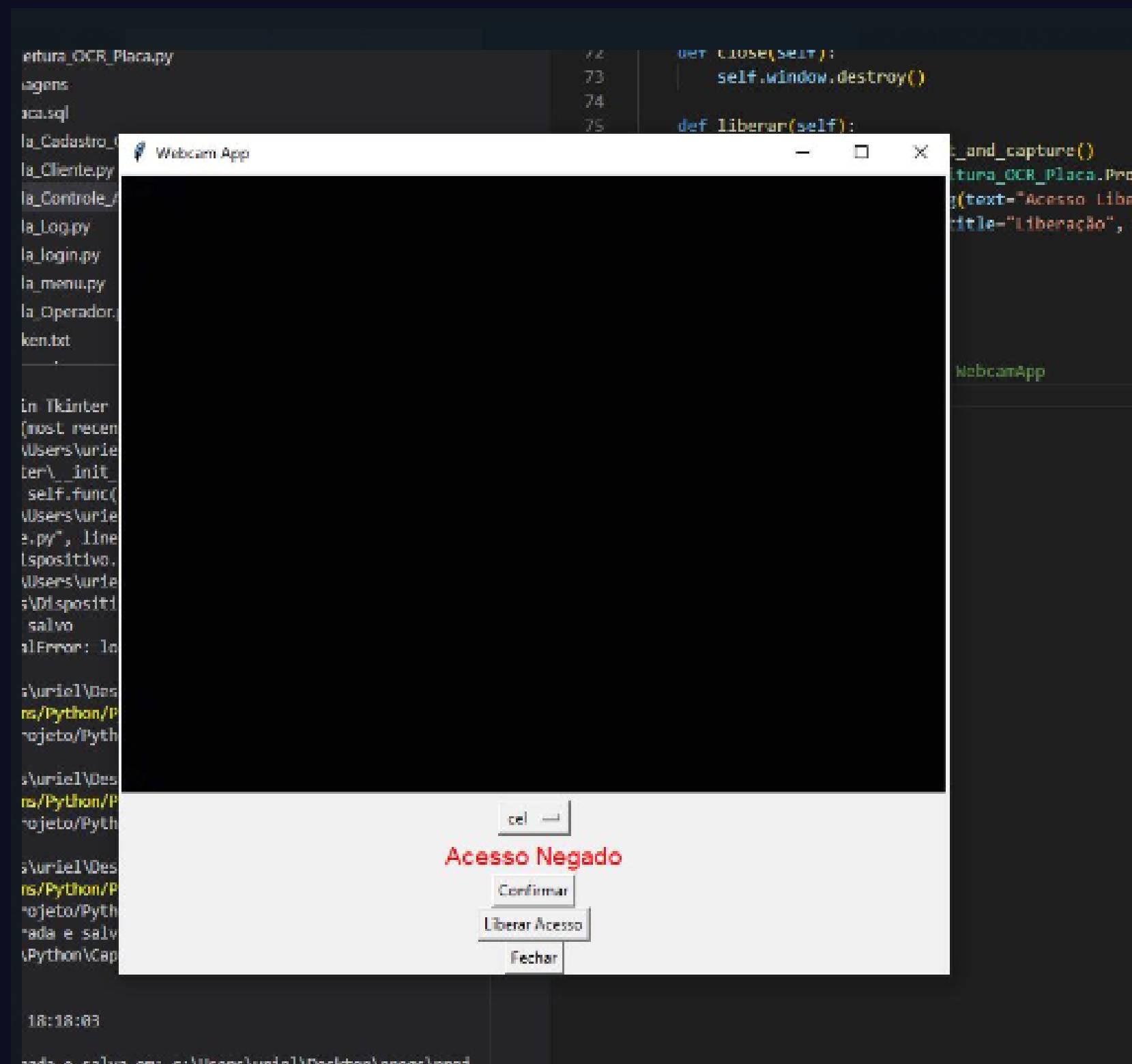
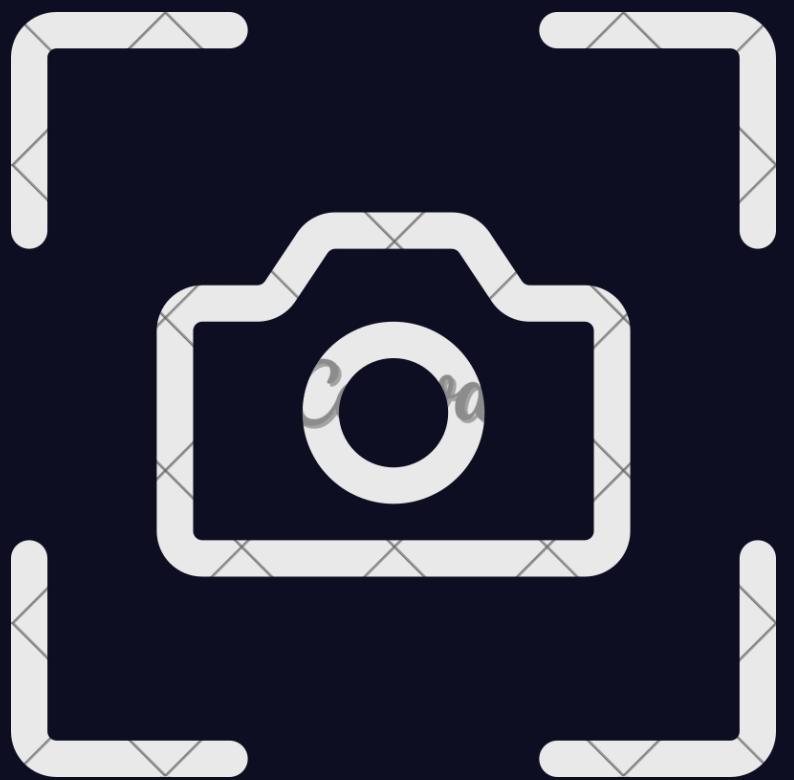
Nome:

IP:

Posição:

Registrar      Buscar      Atualizar      Deletar

# TELA DE MONITORAMENTO



# RESULTADOS

- Os resultados apresentados neste projeto foram satisfatórios.
- Foi possível realizar a leitura e o processamento de diversos modelos de placas de veículos diferentes com exido. Nos testes houve uma grande porcentagem de acertos no reconhecimento das letras e números das placas, em alguns casos ocorreram algumas divergências.
- É perceptível que nas placas de modelo da MERCOSUL há uma confusão no sistema para diferenciar o número 0 e a letra O, ocorrendo erros na leitura de certas placas.



# RESULTADOS

- Outro ponto que é valido ressaltar é o enquadramento da imagem, em fotos onde apresenta o para-choque inteiro do carro, em alguns casos o processamento da imagem reconhece sombras ou alguma característica física do carro como algum caractere, apresentando uma resposta errada.
- Para solução deste problema é necessário realizar a tratativa da cor do fundo reconhecido, assim verificando se o quadrado lido é uma placa ou não.



# CONCLUSÃO

- Este Projeto apresentou um sistema de reconhecimento de placas de veículos por câmera IP e webcam com exido, utilizando a linguagem de programação Python e suas devidas bibliotecas para o reconhecimento e processamento da imagem para tradução em caractere.
- O sistema pode ser implementado em diversos locais para o controle de acesso de veículos, como exemplo: condomínios, shopping, edifícios privados e públicos, empresas com grande número de funcionário entre outros. Também pode ser utilizado em pedágios, realizando a leitura da placa de veículo e por ela fazer a cobrança da taxa.
- A dificuldade encontrada é a instalação da câmera por conta do posicionamento, tendo em vista fatores externos que podem influenciar no erro de leitura.

**OBRIGADO!**

# BIBLIOGRAFIA

- Axmark, David. MySQL Connector/Python Developer Guide. MySQL. Disponível em: <https://dev.mysql.com/doc/connector-python/en/>. Acesso em: 17 maio 2023.
- Catunda, Heitor. JANELA PARA CÓDIGO COM PYSIMPLEGUI – TELA PARA CÓDIGOS EM PYTHON! Hashtag. Disponível em: <https://www.hashtagtreinamentos.com/janela-para-codigo-com-pysimplegui-python>. Acesso em: 19 maio 2023.
- Clark, Jeffrey. Python Imaging Library (Fork). Pillow.org. Disponível em: <https://pypi.org/project/Pillow/>. Acesso em: 19 maio 2023.
- Corporation, Intel. OpenCV Tutorials. OpenCV. Disponível em: [https://docs.opencv.org/4.x/d9/df8/tutorial\\_root.html](https://docs.opencv.org/4.x/d9/df8/tutorial_root.html). Acesso em: 14 maio 2023.
- McKinney, Wes. Miscellaneous operating system interfaces. Python. Disponível em: <https://docs.python.org/3/library/os.html>. Acesso em: 16 maio 2023.

# BIBLIOGRAFIA

- Neeleman, Daniel. O que é controle de acesso e para que serve?. Segware performance em segurança. Disponível em: <https://www.segware.com/post/o-que-e-controle-de-acesso-e-para-que-serve>. Acesso em: 12 maio 2023.
- Packard, Hewlett. Tesseract User Manual. Tesseract. Disponível em: <https://tesseract-ocr.github.io/tessdoc/>. Acesso em: 19 maio 2023.
- Van Rossum, Guido. Interface Python para Tcl/Tk. Python. Disponível em: <https://docs.python.org/pt-br/3/library/tkinter.html>. Acesso em: 15 maio 2023.
- Van Rossum. Operações com expressões regulares . Pynthon. Disponível em: <https://docs.python.org/pt-br/3/library/re.html>. Acesso em: 19 maio 2023.
- Van Rossum. Time access and conversions. Python. Disponível em: <https://docs.python.org/3/library/time.html>. Acesso em: 16 maio 2023.