



# Mais Fluxo Pix Force Vehicle Counter

*Tutorial configuração da aplicação de contagem de veículos*

Diorgenes de Abreu  
MAIS FLUXO

## Sumário

Informações Gerais.....	2
Scripts Python (*.py).....	2
Scripts MSDos (*.bat) .....	2
Hardware Servidor.....	3
Arquivos de configuração e área de contagem .....	3
Configuração Aplicação .....	5
Biblioteca OpenCV .....	5
Arquivo de configuração .....	5
Configuração Áreas de Contagem e Tempo de Permanência .....	6
Área de Contagem .....	6
Área de Tempo de Permanência .....	7
Execução da Aplicação.....	9
Exportação dos Dados .....	10
Fluxo .....	10
Tempo de Permanência.....	10
Banco de Dados Local .....	12
Tabelas.....	12
Consultas .....	13
Contatos .....	17
Pix Force Desenvolvedor .....	17
Mais Fluxo Suporte .....	17

## Informações Gerais

O sistema é desenvolvido em linguagem Python, rodando em um servidor com hardware especificado abaixo, consumindo o streaming de câmeras IP. Máximo de 3 câmeras por servidor, 2 faixas por câmera. A aplicação principal, bem como os scripts de apoio à configuração, é executada através de scripts .bat. Um script para cada câmera, ou seja, uma instância da aplicação principal é iniciada para cada câmera.

O diretório da aplicação: **C:\Users\maisfluxo\Desktop\Executavel**

### Scripts Python (\*.py)

- Aplicação: **yolo<versão\_app>.py**
- Configurador Área Contagem: **desenho.py**
- Configurador Área Tempo de Permanência: **desenhar\_area.py**
- Exportação Fluxo: **formatar\_dbread.py**
- Exportação Tempo de Permanência: **api\_tempopermanencia.py**

### Scripts MSDos (\*.bat)

- Aplicação: **mfvc\_camera<id\_camera>.bat**
- Configurador Área Contagem: **mfvc\_desenhar-area\_camera<id\_camera>.bat**
- Configurador Área Tempo de Permanência: **mfvc\_desenhar-area-tp\_camera<id\_camera>.bat**
- Exportação Fluxo: **mfvc\_db\_exportaTXT.bat**
- Exportação Tempo de Permanência: **mfvc\_db\_exportaTempoPermanencia.bat**
- Matar Aplicação Principal: **mfvc\_killapp.bat**

A aplicação escreve os registros de fluxo e o tempo de permanência (tabelas separadas) em um banco de dados local (SQLite). Cada evento é registrado individualmente no banco de dados. O fluxo é exportado para um arquivo .txt através de script .bat que executa um script python a cada 30min, agrupando as informações de fluxo por ponto de contagem por meia-hora. O arquivo .txt exportado é salvo no diretório C:\MaisFluxoLocal\RetornoTXT, onde o Windows Service MaisFluxoLocal fará o envio da informação para o MFWeb. O tempo de permanência é enviado via API.

Localização do banco de dados: **<diretório\_aplicação>\yolo8.db**

## Hardware Servidor

- Processador: Intel Core i5 12ª gen
- Memória RAM: 16GB
- GPU: NVidia RTX 3060
- Armazenamento: 1 SSD 240GB (Sistema Operacional) + 1 SSD 1TB (armazenamento vídeos)
- Obs.:
  - Este servidor está em um rack da Mais Fluxo, instalado na guarita da portaria, contendo um monitor de 15 polegadas, teclado, mouse, nobreak e switch.
  - A internet é fornecida pelo cliente, sendo um cabo de rede conectado ao switch no rack Mais Fluxo.
  - A placa de rede do computador está com o IP fixado de acordo com o IP que pegou da rede do cliente e com a adição de rota para a rede 192.168.1.0/24 (rede configurada nas câmeras IP)

## Arquivos de configuração e área de contagem

O arquivo de configuração e de área deverá ser único para cada câmera. No arquivo de configuração conterá a URL do streaming da câmera, bem como os pontos de contagem por classe de veículo, agrupados por faixa e no arquivo de área conterá as coordenadas (x,y) de cada área de contagem por faixa.

### **Localização:**

- Arquivos de Configuração: <diretório\_aplicação>\config
  - Padrão nome arquivos: camera<id\_camera>\_config.json
- Arquivos de Área de Contagem e Tempo de Permanência: <diretório\_aplicação>\area
  - Padrão Nome Arquivos:
    - Área de Contagem: camera<id\_camera>\_area.json
    - Área de Tempo de Permanência: camera<id\_camera>\_area\_tp.json

### Exemplo Arquivo de configuração:

```
{
  "codigocliente": 1724,
  "cameras": {
    "camera1": {
      "url": "rtsp://admin:mf6538dm@192.168.1.11:554/cam/realmonitor?channel=1&subtype=0",
      "faixas": {
        "faixa1": {
          "motorcycle": 26058,
          "cars": 26057,
          "truck": 26056,
          "bus": 26059,
          "vuc": 26060
        },
        "faixa2": {
          "motorcycle": 26053,
          "cars": 26052,
          "truck": 26051,
          "bus": 26054,
          "vuc": 26055
        }
      }
    }
  }
}
```

### Exemplo Arquivo de Área:

```
{
  "area_1": [
    [
      611,
      403
    ],
    [
      611,
      413
    ],
    [
      1007,
      415
    ],
    [
      1002,
      401
    ]
  ]
}
```

## Configuração Aplicação

### Biblioteca OpenCV

Para instalar a biblioteca de visão robótica, abra o msdos, com permissões administrativas e execute o comando abaixo.

```
# pip install opencv-python
```

### Arquivo de configuração

O arquivo de configuração é um arquivo .json (chave/valor) onde conterà o código do cliente, a URL de acesso a câmera e os pontos de contagem para cada classe de veículo. Os nomes das classes devem estar exatamente iguais ao exemplo abaixo. Caso contrário, a classe não será contabilizada pelo sistema.

```
{
  "codigocliente": 1724,
  "cameras": {
    "camera1": {
      "url":
"rtsp://admin:mf6538dm@192.168.1.11:554/cam/realmonitor?channel=1&subtype=0",
      "faixas": {
        "faixa1": {
          "motorcycle": 26053,
          "cars": 26052,
          "truck": 26051,
          "bus": 26054
        },
        "faixa2": {
          "motorcycle": 26058,
          "cars": 26057,
          "truck": 26056,
          "bus": 26059
        }
      }
    }
  }
}
```

## Configuração Áreas de Contagem e Tempo de Permanência

Repetir o processo para cada câmera

### Área de Contagem

1. Entrar no diretório da aplicação: **C:\Users\maisfluxo\Desktop\executavel**
  - a. Abra o script .bat **mfvc\_desenhar-area\_camera<id\_camera>.bat** e verifique os parâmetros câmera e url para se certificar de que estejam apontando para a câmera correta.

```
rem Parâmetros
set camera=cameral
set url="rtsp://admin:mf6538dm@192.168.1.11:554/cam/realmonitor?channel=1&subtype=0"
title MaisFluxo_PixForce_DesenharArea_%camera%
cls
```

2. Desenhar área de contagem (máximo 2 áreas por câmera):
  - a. Executar o script msdos: **mfvc\_desenhar-area\_camera<id\_camera>.bat**
  - b. Conhecendo os parâmetros informados no .bat para a execução do script python:
    - i. **--source**: fonte do arquivo de vídeo ou a URL do streaming da câmera.
    - ii. **--output**: onde o arquivo de configuração da área será salvo.
  - c. Exemplo:
    - i. **python.exe desenho.py --source "rtsp://admin:mf6538dm@192.168.1.11:554/cam/realmonitor?channel=1&subtype=0" --output "C:\Users\maisfluxo\Desktop\executavel\area\camera1\_area.json"**
  - d. Será carregado o frame do vídeo para que possamos desenhar a área de contagem. Esta área deverá ser um retângulo, de preferência com altura bem pequena. Deverá ser marcado os 4 pontos para delimitar a área, começando de cima para baixo, da esquerda para direita.
    - i. Exemplo para ordem dos cliques para delimitação da área:

<b>1</b>	<b>4</b>
<b>2</b>	<b>3</b>
  - e. Ao terminar de delimitar a primeira área, pressione **[r]** para habilitar a delimitação de nova área. Ao final, pressione **[s]** para salvar a configuração.

### Exemplo do script .bat

```
1  @echo off
2
3  rem =====
4  rem Mais Fluxo Vehicle Counter
5  rem Criado por: Diorgenes de Abreu <diorgenes.abreu@maisfluxo.com.br>
6  rem Data: 2024-08-12 11h00min
7  rem =====
8
9  rem Parâmetros
10 set camera=cameral
11 set url="rtsp://admin:mf6538dm@192.168.1.11:554/cam/realmonitor?channel=1&subtype=0"
12 title MaisFluxo_PixForce_DesenharArea_%camera%
13 cls
14
15 rem Execução App
16 cd C:\Users\maisfluxo\Desktop\executavel
17 python.exe desenho.py --source %url% --output "C:\Users\maisfluxo\Desktop\executavel\area\%camera%_area.json"
18 exit
```

### Área de Tempo de Permanência

1. Entrar no diretório da aplicação: **C:\Users\maisfluxo\Desktop\executavel**
  - a. Abra o script .bat **mfvc\_desenhar-area-tp\_camera<id\_camera>.bat** e verifique os parâmetros câmera e url para se certificar de que estejam apontando para a câmera correta.

```
rem Parâmetros
set camera=cameral
set url="rtsp://admin:mf6538dm@192.168.1.11:554/cam/realmonitor?channel=1&subtype=0"
set app_dir=C:\Users\maisfluxo\Desktop\executavel
title MaisFluxo_PixForce_DesenharArea_TempoPermanencia_%camera%
--
```

2. Desenhar área de tempo de permanência (máximo 2 áreas por câmera):
  - a. Executar o script msdos: **mfvc\_desenhar-area-tp\_camera<id\_camera>.bat**
  - b. Conhecendo os parâmetros informados no .bat para a execução do script python:
    - i. **--source**: fonte do arquivo de vídeo ou a URL do streaming da câmera.
    - ii. **--output**: onde o arquivo de configuração da área será salvo.
  - c. Exemplo:
    - i. **python.exe desenhar\_area.py --source "rtsp://admin:mf6538dm@192.168.1.11:554/cam/realmonitor?channel=1&subtype=0" --output "C:\Users\maisfluxo\Desktop\executavel\area\camera1\_area-tp.json"**
  - d. Será carregado o frame do vídeo para que possamos desenhar a área de contagem. Esta área deverá ser um retângulo. Deverá ser marcado os 4 pontos para delimitar a área, começando de cima para baixo, da esquerda para direita.
    - i. Exemplo para ordem dos cliques para delimitação da área:

1

4

2

3



- e. Ao terminar de delimitar a primeira área, pressione **[r]** para habilitar a delimitação de nova área. Ao final, pressione **[s]** para salvar a configuração.

Exemplo do script .bat

```
1  @echo off
2
3  rem =====
4  rem Mais Fluxo Vehicle Counter
5  rem Criado por: Diorgenes de Abreu <diorgenes.abreu@maisfluxo.com.br>
6  rem Data: 2024-08-12 11h00min
7  rem =====
8
9  rem Parâmetros
10 set camera=cameral
11 set url="rtsp://admin:mf6538dm@192.168.1.11:554/cam/realmonitor?channel=1&subtype=0"
12 set app_dir=C:\Users\maisfluxo\Desktop\executavel
13 title MaisFluxo_PixForce_DesenharArea_TempoPermanencia_%camera%
14 cls
15
16 rem Execução App
17 cd %app_dir%
18 python.exe desenhar_area.py --source %url% --output %app_dir%\area\%camera%_area_tp.json
19 exit
```

A área rosa é a área de contagem e área verde é a área de tempo de permanência.



## Execução da Aplicação

A aplicação será executada uma instância por câmera. Abaixo os nomes dos scripts .bat de inicialização do sistema bem como a explicação dos parâmetros.

- **mfvc\_camera<id\_câmera>.bat**
- Conhecendo os parâmetros informados no .bat para a execução do script python:
  - **--video\_path**: fonte do arquivo de vídeo ou a URL do streaming da câmera.
  - **--model\_path**: localização do arquivo de modelo de treinamento de IA.
  - **--config\_path**: localização do arquivo de configuração de pontos de contagem.
  - **--area\_config\_path**: localização do arquivo de configuração de áreas (linhas de contagem).
  - **--db\_path**: localização do arquivo de banco de dados local.
  - **--output\_dir**: diretório onde serão salvos os arquivos de vídeos.
  - **--save\_video**: flag para informar se é pra salvar ou não o vídeo processado. (True or False).
  - **--video\_interval**: intervalo de tempo que será gravado o vídeo.
  - **--output\_width**: resolução horizontal do vídeo gravado.
  - **--output\_height**: resolução vertical do vídeo gravado.
  - **--permanencia\_config\_path**: localização do arquivo com as áreas do tempo de permanência.

Exemplo do comando:

```
python.exe yolo13.py --video_path %url% --model_path %app_dir%\model_super.pt --config_path %app_dir%\config\%camera%_config.json --area_config_path %app_dir%\area\%camera%_area.json --db_path %app_dir%\yolo8.db --output_dir %video_dir%\%camera% --save_video true --video_interval 30 --output_width 640 --output_height 358 --permanencia_config_path %app_dir%\area\%camera%_area_tp.json
```

Exemplo do script .bat:

```
1 @echo off
2
3 rem =====
4 rem Mais Fluxo Vehicle Counter
5 rem Criado por: Diogenes de Abreu <diogenes.abreu@maisfluxo.com.br>
6 rem Data: 2024-08-02 16h00min
7 rem =====
8
9 rem Parâmetros
10 set camera=cameral
11 set app_dir=C:\Users\maisfluxo\Desktop\executavel
12 set video_dir=D:\PixForce\videos
13 set url="rtsp://admin:mf6538dm@192.168.1.11:554/cam/realmonitor?channel=1&subtype=0"
14 title MaisFluxo_PixForce_VehicleCounter_%camera%
15 cls
16
17 rem Execução App
18 cd %app_dir%
19 python.exe yolo13.py --video_path %url% --model_path %app_dir%\model_super.pt --config_path %app_dir%\config\%camera%_config.json --area_config_path %app_dir%\area\%camera%_area.json --db_path %app_dir%\yolo8.db --output_dir %video_dir%\%camera% --save_video true --video_interval 30 --output_width 640 --output_height 358 --permanencia_config_path %app_dir%\area\%camera%_area_tp.json
20 exit
```

Os atalhos para os arquivos .bat da aplicação estarão no diretório de inicialização do S.O.

C:\Users\maisfluxo\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup

O SystemScheduler está configurado para iniciar a aplicação todos os dias à 00h e matar às 23h59min.

## Exportação dos Dados

### Fluxo

A consolidação e exportação do dado no formato Mais Fluxo se dá por execução de script .bat agendado no System Scheduler para executar a cada 30min.

- **mfvc\_db\_exportaTXT.bat**
- Conhecendo os parâmetros informados no .bat para a execução do script python:
  - **--client\_code**: código do cliente.
  - **--db\_path**: localização do arquivo de banco de dados local.
  - **--output\_directory**: diretório onde serão salvos os arquivos consolidados e formatados no padrão Mais Fluxo. Padrão: C:\MaisFluxoLocal\RetornoTXT
  - **--days-to-keep**: quantidade de dias de histórico de fluxo a manter no banco de dados local.

Exemplo do comando:

```
python.exe formatar_dbread.py --client_code %empresa% --db_path %app_dir%\yolo8.db --output_directory %txt_dir% --days_to_keep %days_keep%
```

Exemplo do script .bat

```
1 @echo off
2
3 rem =====
4 rem Mais Fluxo Vehicle Counter
5 rem Criado por: Diorgenes de Abreu <diorgenes.abreu@maisfluxo.com.br>
6 rem Data: 2024-08-12 11h00min
7 rem =====
8
9 rem Parâmetros
10 set empresa=1724
11 set app_dir=C:\Users\maisfluxo\Desktop\executavel
12 set txt_dir=C:\MaisFluxoLocal\RetornoTXT
13 set days_keep=120
14 title MaisFluxo_PixForce_ExportTXT
15 cls
16
17 rem Execução App
18 cd %app_dir%
19 python.exe formatar_dbread.py --client_code %empresa% --db_path %app_dir%\yolo8.db --output_directory %txt_dir% --days_to_keep %days_keep%
20 exit
```

### Tempo de Permanência

O envio do tempo de permanência é feito através de script .bat, executando o script python, agendado no System Scheduler para ser executado a cada 3min.

- **mfvc\_db\_exportaTempoPermanencia.bat**
- Conhecendo os parâmetros informados no .bat para a execução do script python:
  - **--db\_path**: localização do banco de dados local

Exemplo de comando:

```
python.exe api_tempopermanencia.py --db_path %app_dir%\yolo8.db
```

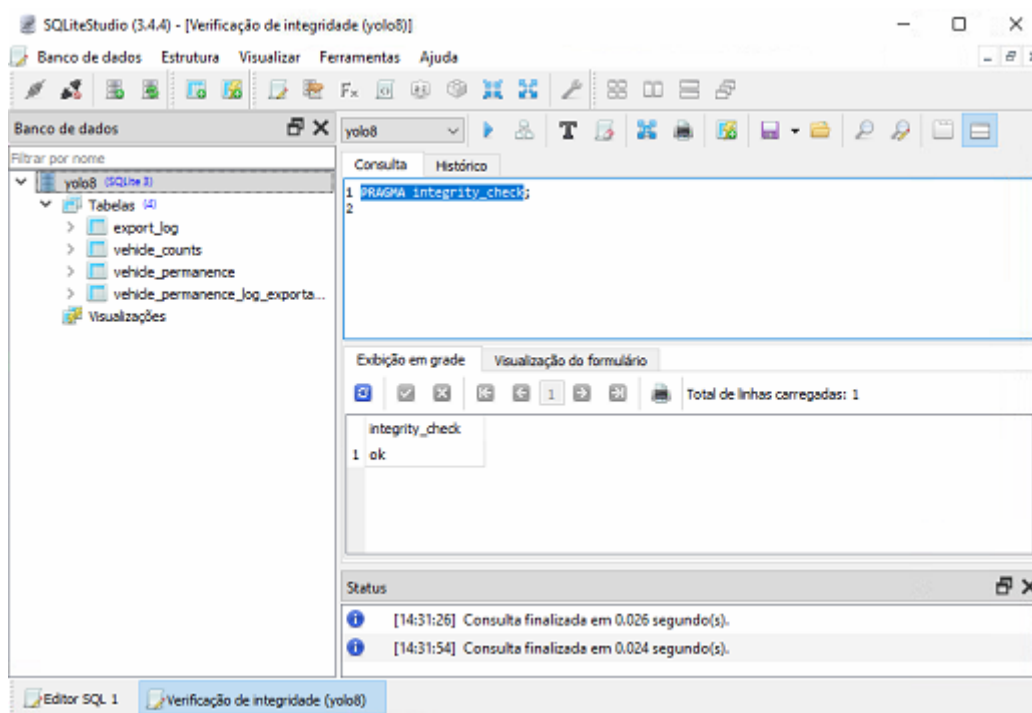
Exemplo do script .bat:

```
1  @echo off
2
3  rem =====
4  rem Mais Fluxo Vehicle Counter
5  rem Criado por: Diorgenes de Abreu <diorgenes.abreu@maisfluxo.com.br>
6  rem Data: 2024-08-12 11h00min
7  rem =====
8
9  rem Parâmetros
10 set app_dir=C:\Users\maisfluxo\Desktop\executavel
11 set days_keep=120
12 title MaisFluxo_PixForce_ExportTempoPermanencia
13 cls
14
15 rem Execução App
16 cd %app_dir%
17 python.exe api_tempopermanencia.py --db_path %app_dir%\yolo8.db
18 exit
```

## Banco de Dados Local

O arquivo de banco de dados local é um arquivo no formato SQLite (.db). Neste arquivo de banco, registramos o fluxo e tempo de permanência por evento, bem como o log da última exportação realizada. Este log faz o controle para que dados que já tenham sido exportados não o sejam novamente.

Para acessar a estrutura de banco de dados utilizamos o aplicativo *SQLite Studio*.



## Tabelas

- **export\_log**: registra a última exportação do fluxo de veículos;
- **vehicle\_counts**: registra os eventos de fluxo;
- **vehicle\_permanence**: registra os eventos de tempo de permanência; e
- **vehicle\_permanence\_log\_export\_exportacao**: registra a última exportação do tempo de permanência.

## Consultas

Podemos consultar o fluxo de veículos usando uma query como a descrita abaixo. Para cada caso, é necessário alterar os valores de pontos de contagem no CASE WHEN.

```
SELECT
    strftime('%Y-%m-%d', timestamp) AS DATA,
    CASE
        WHEN vehicle_code = 26051 THEN 'ENT1-CAMINHÃO'
        WHEN vehicle_code = 26052 THEN 'ENT1-CARRO'
        WHEN vehicle_code = 26053 THEN 'ENT1-MOTO'
        WHEN vehicle_code = 26054 THEN 'ENT1-ÔNIBUS'
        WHEN vehicle_code = 26055 THEN 'ENT1-VUC'
        WHEN vehicle_code = 26056 THEN 'ENT2-CAMINHÃO'
        WHEN vehicle_code = 26057 THEN 'ENT2-CARRO'
        WHEN vehicle_code = 26058 THEN 'ENT2-MOTO'
        WHEN vehicle_code = 26059 THEN 'ENT2-ÔNIBUS'
        WHEN vehicle_code = 26060 THEN 'ENT2-VUC'
    END VEHICLE,
    CASE
        WHEN strftime('%H:%M', timestamp) BETWEEN '00:00' AND '00:29' THEN '00:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '00:30' AND '00:59' THEN '01:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '01:00' AND '01:29' THEN '01:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '01:30' AND '01:59' THEN '02:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '02:00' AND '02:29' THEN '02:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '02:30' AND '02:59' THEN '03:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '03:00' AND '03:29' THEN '03:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '03:30' AND '03:59' THEN '04:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '04:00' AND '04:29' THEN '04:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '04:30' AND '04:59' THEN '05:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '05:00' AND '05:29' THEN '05:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '05:30' AND '05:59' THEN '06:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '06:00' AND '06:29' THEN '06:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '06:30' AND '06:59' THEN '07:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '07:00' AND '07:29' THEN '07:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '07:30' AND '07:59' THEN '08:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '08:00' AND '08:29' THEN '08:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '08:30' AND '08:59' THEN '09:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '09:00' AND '09:29' THEN '09:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '09:30' AND '09:59' THEN '10:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '10:00' AND '10:29' THEN '10:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '10:30' AND '10:59' THEN '11:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '11:00' AND '11:29' THEN '11:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '11:30' AND '11:59' THEN '12:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '12:00' AND '12:29' THEN '12:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '12:30' AND '12:59' THEN '13:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '13:00' AND '13:29' THEN '13:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '13:30' AND '13:59' THEN '14:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '14:00' AND '14:29' THEN '14:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '14:30' AND '14:59' THEN '15:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '15:00' AND '15:29' THEN '15:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '15:30' AND '15:59' THEN '16:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '16:00' AND '16:29' THEN '16:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '16:30' AND '16:59' THEN '17:00'
```



## Mais Fluxo Pix Force Vehicle Counter

### Tutorial configuração da aplicação de contagem de veículos

```
        WHEN strftime('%H:%M', timestamp) BETWEEN '17:00' AND '17:29' THEN '17:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '17:30' AND '17:59' THEN '18:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '18:00' AND '18:29' THEN '18:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '18:30' AND '18:59' THEN '19:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '19:00' AND '19:29' THEN '19:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '19:30' AND '19:59' THEN '20:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '20:00' AND '20:29' THEN '20:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '20:30' AND '20:59' THEN '21:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '21:00' AND '21:29' THEN '21:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '21:30' AND '21:59' THEN '22:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '22:00' AND '22:29' THEN '22:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '22:30' AND '22:59' THEN '23:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '23:00' AND '23:29' THEN '23:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '23:30' AND '23:59' THEN '23:59'
    END AS TIMESTAMP,
    SUM(count_in) AS ENT,
    SUM(count_out) AS SAI
FROM vehicle_counts
WHERE
    timestamp BETWEEN '2024-11-09 00:00:00' AND '2024-11-10 23:59:59'
    AND vehicle_code IN (26051, 26052, 26053, 26054, 26055, 26056, 26057, 26058,
    26059, 26060)
GROUP BY
    strftime('%Y-%m-%d', timestamp),
    CASE
    WHEN vehicle_code = 26051 THEN 'ENT1-CAMINHÃO'
    WHEN vehicle_code = 26052 THEN 'ENT1-CARRO'
    WHEN vehicle_code = 26053 THEN 'ENT1-MOTO'
    WHEN vehicle_code = 26054 THEN 'ENT1-ÔNIBUS'
    WHEN vehicle_code = 26055 THEN 'ENT1-VUC'
    WHEN vehicle_code = 26056 THEN 'ENT2-CAMINHÃO'
    WHEN vehicle_code = 26057 THEN 'ENT2-CARRO'
    WHEN vehicle_code = 26058 THEN 'ENT2-MOTO'
    WHEN vehicle_code = 26059 THEN 'ENT2-ÔNIBUS'
    WHEN vehicle_code = 26060 THEN 'ENT2-VUC'
    END,
    CASE
        WHEN strftime('%H:%M', timestamp) BETWEEN '00:00' AND '00:29' THEN '00:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '00:30' AND '00:59' THEN '01:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '01:00' AND '01:29' THEN '01:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '01:30' AND '01:59' THEN '02:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '02:00' AND '02:29' THEN '02:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '02:30' AND '02:59' THEN '03:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '03:00' AND '03:29' THEN '03:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '03:30' AND '03:59' THEN '04:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '04:00' AND '04:29' THEN '04:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '04:30' AND '04:59' THEN '05:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '05:00' AND '05:29' THEN '05:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '05:30' AND '05:59' THEN '06:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '06:00' AND '06:29' THEN '06:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '06:30' AND '06:59' THEN '07:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '07:00' AND '07:29' THEN '07:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '07:30' AND '07:59' THEN '08:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '08:00' AND '08:29' THEN '08:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '08:30' AND '08:59' THEN '09:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '09:00' AND '09:29' THEN '09:30'
        WHEN strftime('%H:%M', timestamp) BETWEEN '09:30' AND '09:59' THEN '10:00'
        WHEN strftime('%H:%M', timestamp) BETWEEN '10:00' AND '10:29' THEN '10:30'
```

```
WHEN strftime('%H:%M', timestamp) BETWEEN '10:30' AND '10:59' THEN '11:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '11:00' AND '11:29' THEN '11:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '11:30' AND '11:59' THEN '12:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '12:00' AND '12:29' THEN '12:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '12:30' AND '12:59' THEN '13:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '13:00' AND '13:29' THEN '13:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '13:30' AND '13:59' THEN '14:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '14:00' AND '14:29' THEN '14:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '14:30' AND '14:59' THEN '15:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '15:00' AND '15:29' THEN '15:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '15:30' AND '15:59' THEN '16:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '16:00' AND '16:29' THEN '16:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '16:30' AND '16:59' THEN '17:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '17:00' AND '17:29' THEN '17:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '17:30' AND '17:59' THEN '18:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '18:00' AND '18:29' THEN '18:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '18:30' AND '18:59' THEN '19:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '19:00' AND '19:29' THEN '19:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '19:30' AND '19:59' THEN '20:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '20:00' AND '20:29' THEN '20:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '20:30' AND '20:59' THEN '21:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '21:00' AND '21:29' THEN '21:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '21:30' AND '21:59' THEN '22:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '22:00' AND '22:29' THEN '22:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '22:30' AND '22:59' THEN '23:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '23:00' AND '23:29' THEN '23:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '23:30' AND '23:59' THEN '23:59'

END
ORDER BY
  strftime('%Y-%m-%d', timestamp),
CASE
  WHEN strftime('%H:%M', timestamp) BETWEEN '00:00' AND '00:29' THEN '00:30'
  WHEN strftime('%H:%M', timestamp) BETWEEN '00:30' AND '00:59' THEN '01:00'
  WHEN strftime('%H:%M', timestamp) BETWEEN '01:00' AND '01:29' THEN '01:30'
  WHEN strftime('%H:%M', timestamp) BETWEEN '01:30' AND '01:59' THEN '02:00'
  WHEN strftime('%H:%M', timestamp) BETWEEN '02:00' AND '02:29' THEN '02:30'
  WHEN strftime('%H:%M', timestamp) BETWEEN '02:30' AND '02:59' THEN '03:00'
  WHEN strftime('%H:%M', timestamp) BETWEEN '03:00' AND '03:29' THEN '03:30'
  WHEN strftime('%H:%M', timestamp) BETWEEN '03:30' AND '03:59' THEN '04:00'
  WHEN strftime('%H:%M', timestamp) BETWEEN '04:00' AND '04:29' THEN '04:30'
  WHEN strftime('%H:%M', timestamp) BETWEEN '04:30' AND '04:59' THEN '05:00'
  WHEN strftime('%H:%M', timestamp) BETWEEN '05:00' AND '05:29' THEN '05:30'
  WHEN strftime('%H:%M', timestamp) BETWEEN '05:30' AND '05:59' THEN '06:00'
  WHEN strftime('%H:%M', timestamp) BETWEEN '06:00' AND '06:29' THEN '06:30'
  WHEN strftime('%H:%M', timestamp) BETWEEN '06:30' AND '06:59' THEN '07:00'
  WHEN strftime('%H:%M', timestamp) BETWEEN '07:00' AND '07:29' THEN '07:30'
  WHEN strftime('%H:%M', timestamp) BETWEEN '07:30' AND '07:59' THEN '08:00'
  WHEN strftime('%H:%M', timestamp) BETWEEN '08:00' AND '08:29' THEN '08:30'
  WHEN strftime('%H:%M', timestamp) BETWEEN '08:30' AND '08:59' THEN '09:00'
  WHEN strftime('%H:%M', timestamp) BETWEEN '09:00' AND '09:29' THEN '09:30'
  WHEN strftime('%H:%M', timestamp) BETWEEN '09:30' AND '09:59' THEN '10:00'
  WHEN strftime('%H:%M', timestamp) BETWEEN '10:00' AND '10:29' THEN '10:30'
  WHEN strftime('%H:%M', timestamp) BETWEEN '10:30' AND '10:59' THEN '11:00'
  WHEN strftime('%H:%M', timestamp) BETWEEN '11:00' AND '11:29' THEN '11:30'
  WHEN strftime('%H:%M', timestamp) BETWEEN '11:30' AND '11:59' THEN '12:00'
  WHEN strftime('%H:%M', timestamp) BETWEEN '12:00' AND '12:29' THEN '12:30'
  WHEN strftime('%H:%M', timestamp) BETWEEN '12:30' AND '12:59' THEN '13:00'
  WHEN strftime('%H:%M', timestamp) BETWEEN '13:00' AND '13:29' THEN '13:30'
```



```
WHEN strftime('%H:%M', timestamp) BETWEEN '13:30' AND '13:59' THEN '14:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '14:00' AND '14:29' THEN '14:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '14:30' AND '14:59' THEN '15:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '15:00' AND '15:29' THEN '15:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '15:30' AND '15:59' THEN '16:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '16:00' AND '16:29' THEN '16:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '16:30' AND '16:59' THEN '17:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '17:00' AND '17:29' THEN '17:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '17:30' AND '17:59' THEN '18:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '18:00' AND '18:29' THEN '18:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '18:30' AND '18:59' THEN '19:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '19:00' AND '19:29' THEN '19:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '19:30' AND '19:59' THEN '20:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '20:00' AND '20:29' THEN '20:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '20:30' AND '20:59' THEN '21:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '21:00' AND '21:29' THEN '21:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '21:30' AND '21:59' THEN '22:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '22:00' AND '22:29' THEN '22:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '22:30' AND '22:59' THEN '23:00'
WHEN strftime('%H:%M', timestamp) BETWEEN '23:00' AND '23:29' THEN '23:30'
WHEN strftime('%H:%M', timestamp) BETWEEN '23:30' AND '23:59' THEN '23:59'
END,
vehicle_code, timestamp;
```

Exemplo de resultado da consulta acima:

Exibição em grade		Visualização do formulário			
		Total de linhas carregadas: 9			
	DATA	VEHICLE	TIMESTAMP	ENT	SAI
1	2024-11-09	ENT2-CARRO	05:30	2	0
2	2024-11-09	ENT2-CARRO	06:00	17	0
3	2024-11-09	ENT2-CARRO	06:30	16	0
4	2024-11-09	ENT2-CARRO	07:00	15	0
5	2024-11-09	ENT2-CARRO	07:30	9	0
6	2024-11-09	ENT2-MOTO	07:30	1	0
7	2024-11-09	ENT2-CAMINHÃO	08:00	1	0
8	2024-11-09	ENT2-CARRO	08:00	2	0
9	2024-11-10	ENT2-CARRO	07:30	1	0



## Mais Fluxo Pix Force Vehicle Counter

*Tutorial configuração da aplicação de contagem de veículos*

### Contatos

#### Pix Force Desenvolvedor

- **Desenvolvedor:** Rafael Santi
- **E-mail:**
- **Cel.:** (11)97765-5095

#### Mais Fluxo Suporte

- **Suporte:** Diorgenes de Abreu
- **E-mail:** [diorgenes.abreu@maisfluxo.com.br](mailto:diorgenes.abreu@maisfluxo.com.br)
- **Cel.:** (21)96746-4915
-