

MONITORAMENTO DE SEGURANÇA POR IMAGENS

Documento TR-01

Versão 1.0 de janeiro de 2024

Projeto da disciplina STR

www.feelt.ufu.br

Execução:



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
www.ufu.br

Alan Petrônio Pinheiro

Coordenador do projeto – UFU/LRI

Execução e pesquisa:

GABRIEL ALVES CAIXETA CUSTÓDIO
MATEUS HONORATO DE ALMEIDA

| | | |
|---|--|--|
| 1. DATA VERSÃO ORIGINAL 23-3-2024 | 2. DATA ÚLTIMA ATUALIZAÇÃO 23-4-2024 | 3. DATA COBERTA JAN/24 ATÉ MAI/24 |
| 4. TÍTULO DESTE DOCUMENTO DETECÇÃO DE PESSOAS POR CÂMERAS | | 5a. PROCESSO SEI DO P&D - |
| | | 5b. NÚMERO PROJETO P&D 00123456789.... |
| 6. AUTOR(ES) GABRIEL ALVES CAIXETA CUSTÓDIO MATEUS HONORATO DE ALMEIDA | | 5c. ETAPA DO PROJETO TODAS |
| | | 5d. TIPO DE PRODUTO DOCUMENTAÇÃO TÉCNICA DE SOFTWARE DE DISCIPLINA STR |
| 7. ENDEREÇO AV. JOÃO NAVES DE ÁVILA, 2121, BLOCO 3N - UBERLÂNDIA - MG | | 8. NÚMERO DO DOCUMENTO TR-01 |
| 9. DISTRIBUIÇÃO DESTE DOCUMENTO DISTRIBUIÇÃO ABERTA A TODOS OS INTERESSADOS. | | |
| 10. NOTAS COMPLEMENTARES - | | |
| 11. RESUMO ESTE DOCUMENTO DESCREVE A MODELAGEM DOS ELEMENTOS DA APLICAÇÃO MU QUE GERA PACOTES DE DADOS PARA OUTROS ELEMENTOS DE SOFTWARE RESTRATADO POR OUTROS REPORTES TÉCNICOS. TODOS MÓDULOS CONSTITUEM UM SISTEMA SUPERVISÓRIO PARA O SETOR ELÉTRICO. | | |
| 12. PALAVRAS-CHAVE P&D; IOT; SISTEMA EM TEMPO REAL, SISTEMA SUPERVISÓRIO, DETECÇÃO DE OBJETOS. | | |
| 13. CLASSIFICAÇÃO SEGURANÇA: ABERTA | 14. NÚMERO DE PÁGINAS - | 15. NOME DO RESPONSÁVEL PRINCIPAL E CONTATO ALAN PETRÔNIO PINHEIRO. EMAIL: alan_petronio@yahoo.com.br. TELEFONE: (34) 3239-4701 |

HISTÓRICO DE VERSÕES DESTE TR

Tabela 1 – Histórico de versões deste reporte técnico.

| Versão | Data | Modificações |
|--------|------------|---|
| 1.0 | abril/2024 | <ul style="list-style-type: none">• Principais elementos de projeto• Requerimentos básicos• Modelagem de pacotes e fluxo de pacotes |

SUMÁRIO

| | |
|---|----|
| RESUMO GERAL | 4 |
| 1 – INTRODUÇÃO: VISÃO GERAL DA SOLUÇÃO | 5 |
| 1.1 – PROPÓSITO E ESCOPO | 5 |
| 1.2 – PRODUTO: PERSPECTIVAS E FUNÇÕES | 6 |
| 1.3 – RESTRIÇÕES DO PRODUTO E CONSIDERAÇÕES | 7 |
| 2 – REQUISITOS | 8 |
| 2.1 – CENÁRIOS DE USO | 8 |
| 2.2 – REQUISITOS E VALIDAÇÃO | 11 |
| 2.3 – VERSIONAMENTO | 12 |
| 2.4 – ELEMENTOS DE PROJETO | 12 |
| 2.4.1 – Máquina de estados | 12 |
| 2.4.2 – Interfaces de usuário | 13 |
| 3 – MODELAGEM | 14 |
| 3.1 – BLOCOS DE ELEMENTOS PRINCIPAIS | 14 |
| 3.2 – Fluxo geral de mensagens | 16 |
| 3.4 – Modelagem detalhada dos recursos | 16 |
| 3.4.1 – Envio de imagens da câmera | 16 |
| 3.4.2 – Envio de SMS | 18 |



RESUMO GERAL

O sistema projetado é uma solução integrada de segurança e monitoramento que utiliza duas câmeras IP para capturar imagens em tempo real, identificar a presença de humanos e, caso detecte tal presença, enviar um SMS de alerta para um número de telefone previamente cadastrado. A plataforma é destinada a melhorar a segurança de propriedades residenciais, comerciais ou industriais, permitindo uma resposta rápida a qualquer invasão ou atividade suspeita detectada pelas câmeras.

Componentes do Sistema

1. Câmeras IP com RTSP:

- As câmeras IP estão equipadas com a capacidade de transmitir vídeo usando o protocolo RTSP (Real Time Streaming Protocol), que permite a transmissão eficiente de vídeo ao vivo através de redes IP. Cada câmera é configurada para monitorar áreas específicas e capturar movimentos ou atividades em tempo real.

2. Detecção de Presença Humana:

- Utilizando algoritmos avançados de visão computacional, o sistema analisa os frames capturados pelas câmeras para detectar a presença de humanos. Essa detecção é feita em tempo real, com o sistema priorizando a rápida identificação para minimizar atrasos na resposta a uma invasão.

3. Envio de SMS:

- Em caso de detecção de presença humana, o sistema automaticamente dispara um alerta via SMS para números de telefone cadastrados. Este alerta informa o destinatário sobre a detecção, permitindo uma verificação rápida e, se necessário, ações subsequentes como contato com as autoridades.

4. Interface de Usuário:

- A interface de usuário é projetada para ser intuitiva e acessível, permitindo aos usuários cadastrarem e gerenciarem os números de telefone que receberão os alertas SMS. Além disso, a interface pode oferecer visualização ao vivo das câmeras, controle de configurações e revisão de logs de eventos.

5. Banco de Dados:

- Um banco de dados robusto armazena informações críticas, incluindo os números de telefone cadastrados, logs de eventos e configurações do sistema. Esse banco de dados é essencial para a operação eficiente do sistema, assegurando que todos os dados necessários para o envio de alertas e a gestão de usuários estejam disponíveis e seguros.

Funcionalidades e Operação

- **Robustez e Resiliência:** O sistema é projetado para ser robusto e operar de forma contínua, com medidas para garantir que permaneça funcional mesmo durante interrupções temporárias de rede. As câmeras tentam reconectar automaticamente e continuar a transmissão de dados assim que a rede estiver disponível novamente.

1 – Introdução: visão geral da solução

1.1 – Propósito e escopo

O propósito da solução é fazer o monitoramento de uma área com câmeras, de forma constante ou programada, realizando a detecção de pessoas, e enviando uma sinalização via SMS quando ocorrer um evento, no caso, a própria detecção. Este sistema visa o monitoramento, segurança e proteção de um ambiente de possíveis agentes que possam agir contra o zelo deste, comunicando com o proprietário sobre o ocorrido em tempo real. A principal possível aplicação desta solução é em sistemas de segurança residenciais.

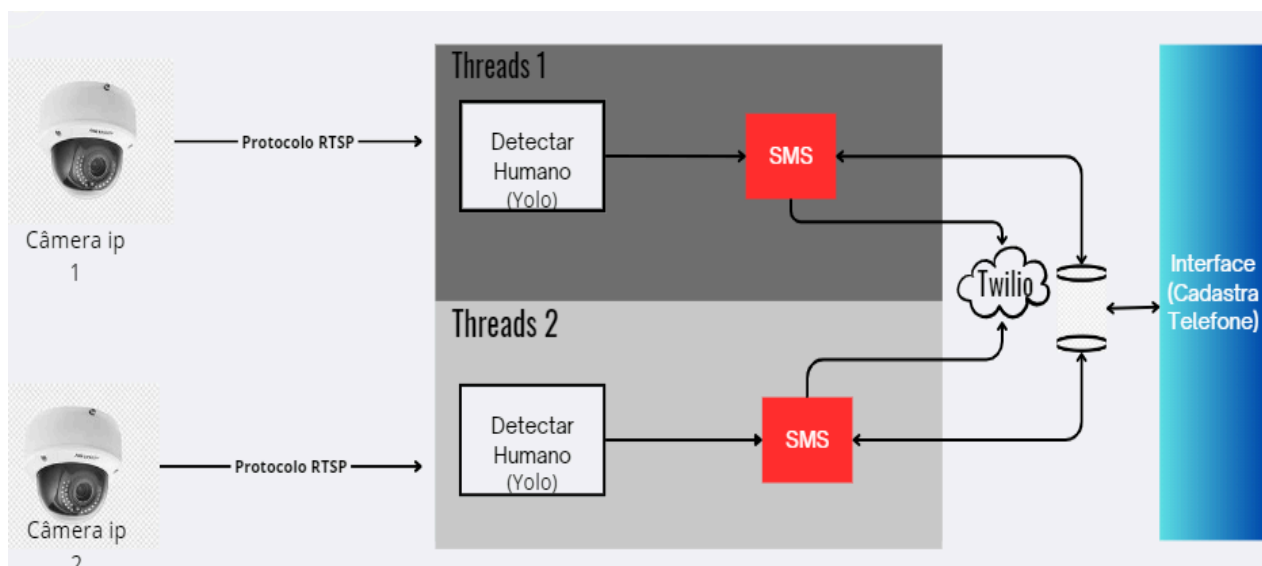


Figura 1.1.1: Visão geral de escopo.

Desta maneira, o escopo da solução abrange:

- 1) **Recebimento e Processamento de Dados:** O sistema deve ser capaz de receber imagens das câmeras em uma certa frequência de frames por segundo. A taxa deve ser suficiente para que ocorra a detecção de um ser humano. Ao escalar o projeto, utilizando mais de uma câmera, faz-se necessário o uso de threads.
- 2) **Identificação de objetos:** O sistema deve identificar a existência de um objeto específico (ser humano) em cada frame recebido pelas câmeras.
- 3) **Envio de sinalização:** No caso de detecção (trigger), o sistema, por meio de API, deve enviar uma sinalização SMS para o usuário.

Desta maneira, o sistema funcionará acerca destas três funções principais, projetado para funcionar em uma única rede, sendo visado para a segurança residencial, porém lidando com uma grande quantidade de frames por segundo, recebidos por mais de uma câmera.

1.2 – Produto: perspectivas e funções

Como apresentado anteriormente, a função deste sistema é o monitoramento e alerta de segurança de ambientes com câmeras, emitindo uma sinalização de alerta com base em um evento detectado em tempo real. Para o funcionamento correto do sistema, é necessário que todos os módulos se comuniquem adequadamente para que haja maior sincronia e menor latência possível entre a detecção do evento e a emissão do sinal. O sistema tem como principais funções:

- 1) Monitoramento por câmera (recebimento de dados em frames de imagens) ;
- 2) Enviar frames pela rede ao host do programa;
- 3) Tratamento dos frames recebidos pela câmera;
- 4) Detectar evento *trigger* (detectar objeto, no caso, ser humano);
- 5) Enviar alerta via SMS.

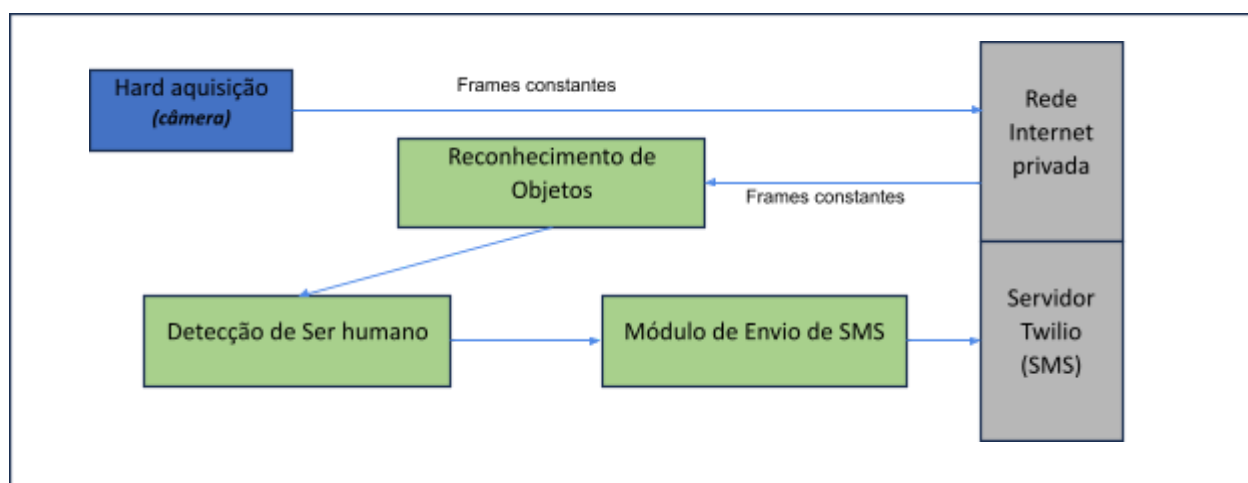


Figura 1.2.1: Principais elementos de projeto.

Para entender o sistema, começemos a análise observando a [Figura 1.3.1](#). Com base nisto, descreve-se os elementos:

- **Módulo hardware aquisição:** no caso deste projeto, são as câmeras, que fazem a captura das imagens em tempo real e enviam para a rede.
- **Módulo Reconhecimento de Objetos :** recebe os frames enviados pelas câmeras através da rede, por meio do endereço de IP de cada uma. Os frames são tratados pela biblioteca YOLO, que faz o reconhecimento de objetos com base em aprendizado de máquina (já inserido e treinado na biblioteca).



- **Módulo detecção de ser humano:** funciona como *trigger* para o envio de SMS, recebe o reconhecimento feito pelo módulo anterior e realiza o tratamento adequado, selecionando os frames em que houve o reconhecimento de ser humano e evitando o *flood* de muitos sinais de frames subsequentes para o módulo de envio de SMS.
- **Módulo de envio de SMS:** este módulo recebe sinalização do módulo de detecção para poder prosseguir com o envio de mensagem de alerta. Faz o tratamento da mensagem a ser enviada e a comunicação com a API do servidor do Twilio, o qual enviará a mensagem de fato, com uma conta, *token*, e o número pelo qual será enviado o SMS. Por fim, procura por números de celulares cadastrados em um banco de dados e faz o envio para cada um deles.

A divisão das etapas do projeto foi feita da seguinte maneira, Mateus Honorato ficou encarregado da parte de detecção de humano, e Gabriel do módulo de envio de SMS, porém, a integração e ajustes dos dois módulos foi feita de forma conjunta.

1.3 – Restrições do produto e considerações

A solução foi testada em apenas algumas condições, e, pelos testes, foram identificadas restrições e limitações detalhadas aqui:

Tabela 1.3.1: Restrições e limitações previstas para sistema.

| Nº | Restrição/limitação | Descrição/detalhamento |
|----|---|---|
| 1 | Dependente da latência da rede | Dependendo da latência da rede, há um atraso significativo no recebimento dos frames para o reconhecimento. A etapa de reconhecimento das imagens tem um atraso estimado entre 10 e 30 segundos entre o envio dos frames pela câmera até o reconhecimento. |
| 2 | Dependente da latência do sistema de processamento dos frames | Dependendo da capacidade da máquina hospedeira da solução, pode ocorrer atraso no processamento da detecção de um ser humano e envio do sinal para o módulo SMS. |
| 3 | Necessário limitação do número de frames que enviarão sinal para o módulo SMS | O módulo de Detecção de Ser Humano deve impedir que frames subsequentes solicitem envio de SMS. Sem a limitação, como há detecção em cada um dos frames, todos eles solicitarão um novo envio de SMS, mesmo que a diferença de tempo entre dois frames em sequência esteja na ordem de milissegundos. |
| 4 | Necessário uso de uma máquina ligada | Na implementação atual do projeto, é necessário que haja um computador sempre ligado rodando o programa para que funcione. |
| 5 | Novos números cadastrados necessitam de verificação | A versão gratuita do servidor e API do twilio permite a verificação de apenas um número tanto para enviar quanto para receber mensagens. Para contornar esta limitação, é possível utilizar mais de uma conta, assim, cada número que receberá um SMS deverá ter seu próprio <i>account_sid</i> , <i>auth_token</i> e número de envio |



| | | |
|---|--|--|
| | | twilio. Com a versão paga, isto é completamente solucionado, sendo necessário apenas um account_sid, auth_token e número twilio para todos os números que receberão SMS. |
| 6 | Dependência de Threads para Multiprocessamento | Ao se utilizar mais de uma câmera para a detecção do ser humano, é imprescindível o uso de Threads, para que o programa possa atuar sobre as duas em paralelo. |

2 – Requisitos

2.1 – Cenários de uso

Os seguintes cenários foram identificados para este sistema.

- a) **Cenário 1 – Condições Normais sem Detecção Humana:** Neste cenário, a câmera apenas continua capturando a imagem do ambiente e o módulo de reconhecimento recebe os frames, sem repassá-los.

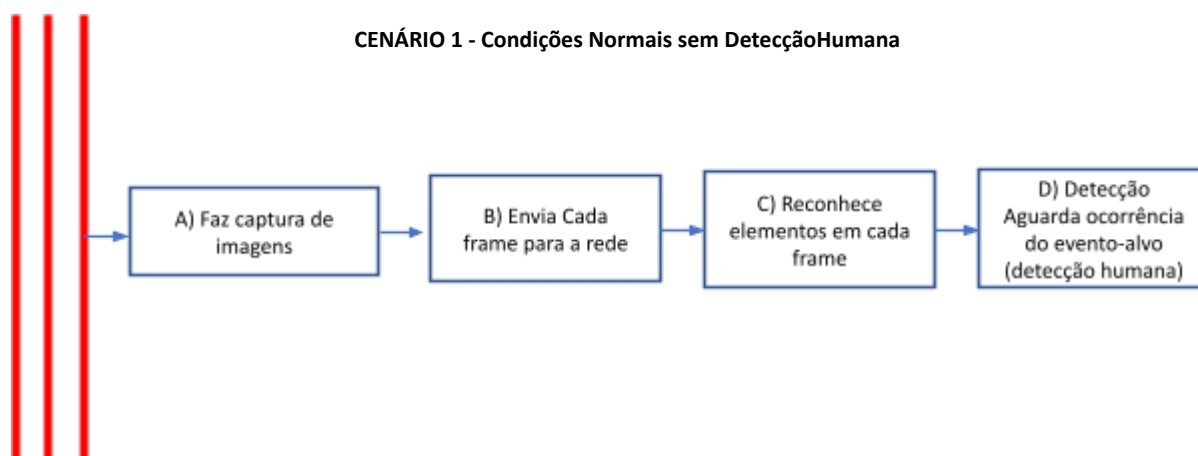


Figura 2.1.1: Cenário de aplicação.



- b) **Cenário 2 – Falha na Captura dos Frames:** Este cenário ocorre por uma falha de comunicação com duração curta entre o sistema de reconhecimento e o recebimento dos frames da câmera

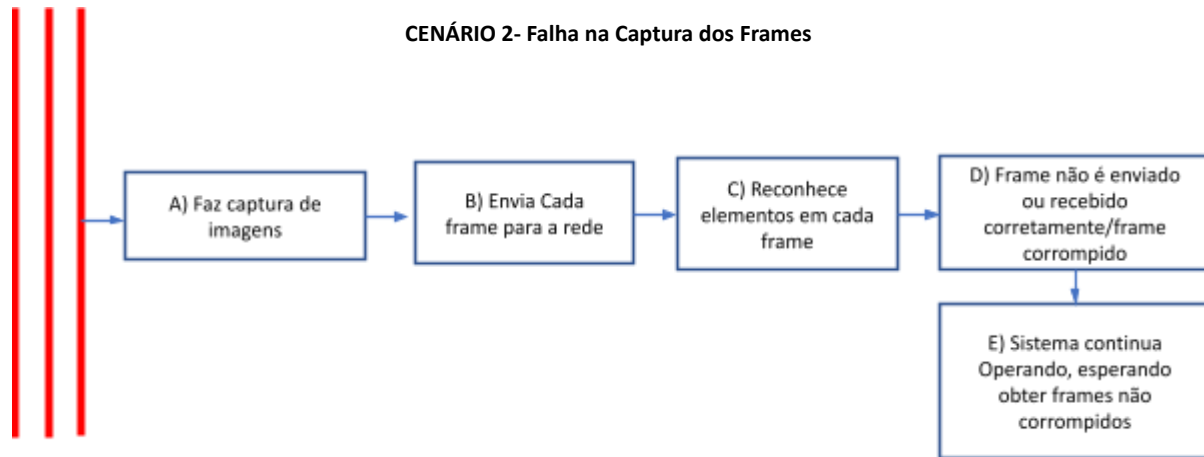


Figura 2.1.2: Cenário de aplicação.

- c) **Cenário 3 – Câmera não detectada:** O sistema reconhece que não há câmeras na rede em que ele está, ou houve falha na procura. Nos dois casos, a solução é encerrada, sem acionar os demais módulos.

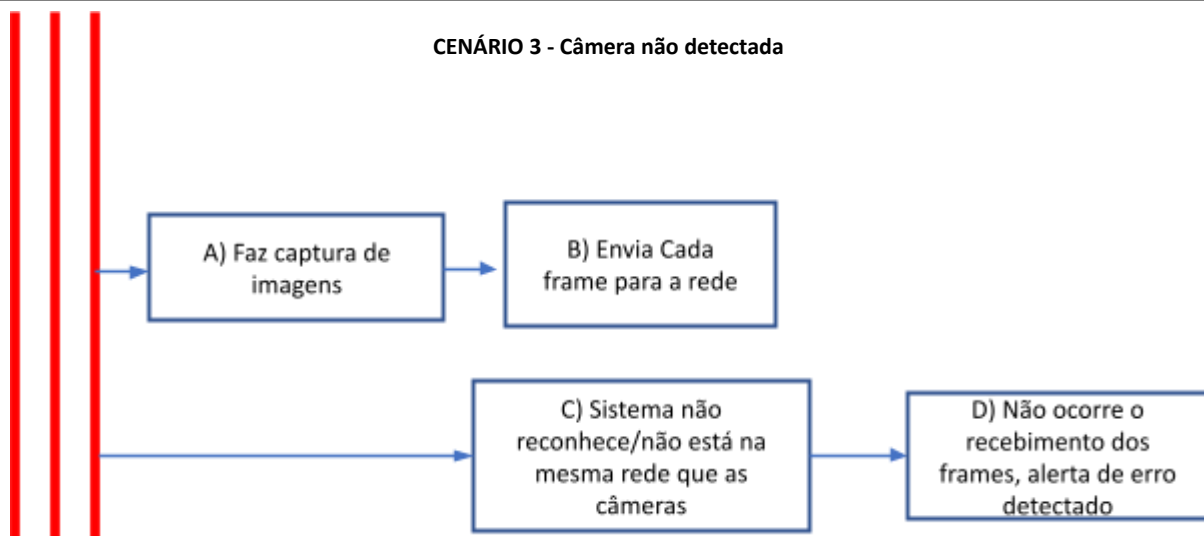


Figura 2.1.3: Cenário de aplicação.

- d) **Cenário 4 – Detecção Humana sem limitação de sinalização dos frames:** O sistema reconhece pessoas pela câmera, através da biblioteca YOLO, e o módulo de detecção aciona o módulo de envio de SMS para cada frame em que houve detecção humana. O módulo de SMS obtém as informações de número de recebimento, número do twilio, sid da conta e token de autorização do banco de dados e envia um SMS pela API do twilio para cada vez que é acionado.

CENÁRIO 4 - Detecção Humana sem limitação de sinalização dos frames:

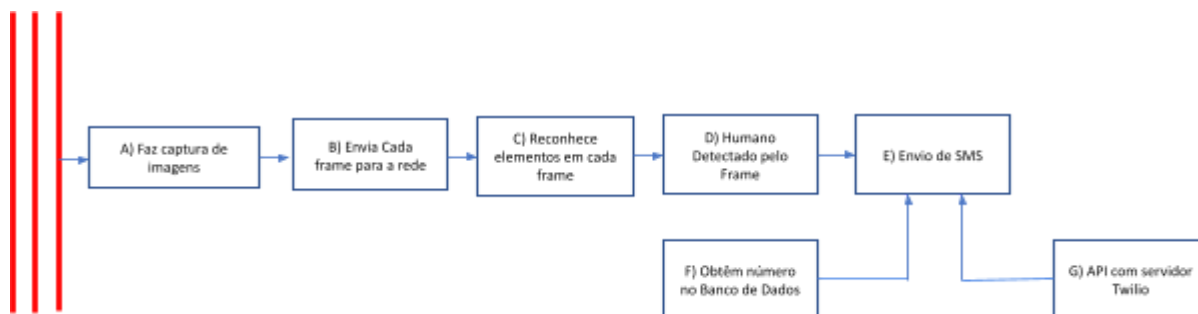


Figura 2.1.4: Cenário de aplicação.

- e) **Cenário 5 – Detecção Humana com limitação de sinalização dos frames:** O sistema reconhece pessoas pela câmera, através da biblioteca YOLO, e o módulo de detecção faz o tratamento da limitação dos frames, impedindo que frames subsequentes acionem o módulo SMS, dessa forma, ocorrerá apenas um acionamento para cada conjunto de frames. Esta limitação é então redefinida assim que não ocorre mais detecção de um ser humano no frame enviado pela câmera, permitindo outra detecção e acionamento em um intervalo de tempo diferente. O módulo de SMS obtém as informações de número de recebimento, número do twilio, sid da conta e token de autorização do banco de dados e envia um SMS pela API do twilio para cada vez que é acionado.

CENÁRIO 5 - Detecção Humana com limitação de sinalização dos frames

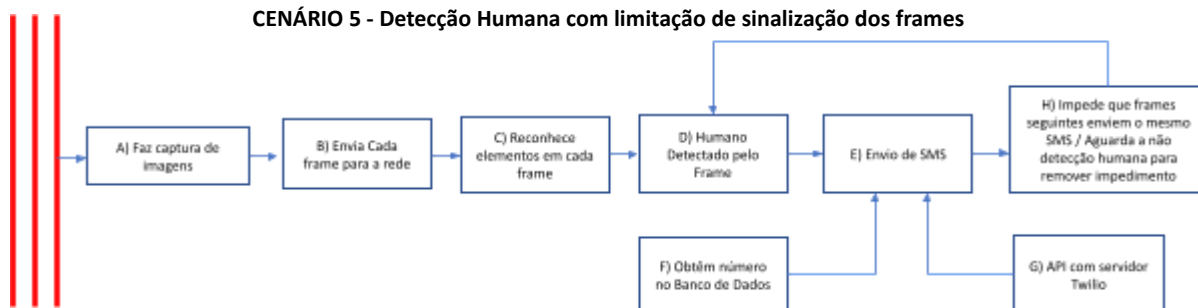


Figura 2.1.5: Cenário de aplicação.

- f) **Cenário 6 – Detecção Humana com mais de uma câmera:** Em um cenário com mais de uma câmera, são utilizadas mais de uma threads, uma para cada câmera, onde o programa irá rodar em cada uma delas. Quando ocorre um reconhecimento, seguido de detecção e acionamento, o módulo SMS reconhecerá a qual câmera pertence o acionamento, e na mensagem enviada, estará exibida explicitamente essa diferenciação.



CENÁRIO 6 - Detecção Humana com mais de uma câmera:

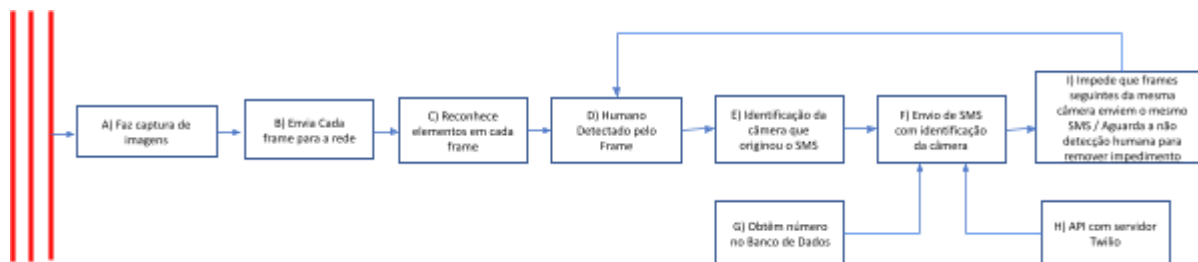


Figura 2.1.6: Cenário de aplicação.

2.2 – Requisitos e validação

Com base nas avaliações de cenário de uso, desenvolveu-se na sequência a seguinte lista de requerimentos, vista na tabela da sequência.

Tabela 2.2.1: Mapa de requerimentos

| Classe/Compo nente | Nº req. | Requisito | Origem requisito | Prio | Tipo validação | Divisão do projeto |
|--------------------|---------|--|----------------------------|------|---|--------------------|
| 1 - Funcional | 1.1 | Capturar frames usando RTSP de duas câmeras IP. | Funcionalidad e do sistema | 1 | Testar conectividade com câmeras IP usando o protocolo RTSP em diferentes condições de rede. | Mateus |
| | 1.2 | Detectar presença humana nos frames capturados. | Necessidade do usuário | 1 | Implementar algoritmo de detecção de presença humana e validar com diferentes cenários e condições de iluminação. | Mateus |
| | 1.3 | Enviar SMS ao detectar presença humana. | Requisito de alerta | 1 | Integrar com API do Twilio e testar o envio de SMS em tempo real ao detectar presença. | Gabriel |
| 2-Interfac e | 2.1 | Interface de usuário para cadastro de números de telefone. | Facilidade de uso | 2 | Desenvolver e validar interface de usuário para garantir facilidade no cadastro e gestão de números. | Gabriel |



| | | | | | | |
|------------------|-----|--|------------------------------------|---|--|----------------|
| 3-Banco de dados | 3.1 | Banco de dados para armazenar números de telefone cadastrados. | Integridade e recuperação de dados | 2 | Simular falhas de rede e validar a resiliência do sistema, garantindo que as câmeras reconectem automaticamente. | Gabriel/Mateus |
|------------------|-----|--|------------------------------------|---|--|----------------|

Aqui está uma tabela de requisitos e validação modificada para um sistema de duas câmeras IP que usa o protocolo RTSP para capturar frames, detectar presença humana e, caso detecte, enviar um SMS para um número de telefone cadastrado. O sistema também inclui uma interface de usuário para o cadastro de telefones e um banco de dados para armazenar esses dados.

Explicação dos Requisitos

- 1.1 - Captura de Frames: O sistema precisa estabelecer uma conexão estável com as câmeras IP usando RTSP para garantir a captura contínua de frames.
- 1.2 - Detecção de Presença: Um algoritmo de visão computacional será utilizado para analisar os frames e detectar a presença humana.
- 1.3 - Envio de SMS: Integração com a API do Twilio para enviar alertas via SMS quando uma presença for detectada.
- 2.1 - Interface de Usuário: Uma interface simples e clara para permitir ao usuário cadastrar e gerenciar os números de telefone para recebimento de alertas.
- 3.1 - Banco de Dados: Segurança e integridade dos dados são cruciais, pois o sistema depende da precisão dos números de telefone cadastrados para enviar os alertas.

2.3 – Versionamento

Os recursos do software são distribuídos em versões conforme estimado pela tabela na sequência.

Tabela 2.3.1: Tabela de recursos do sistema e versão.

| Versão | Recurso |
|-----------------|--|
| 1.0 (abr/24) | <ul style="list-style-type: none">■ O sistema ser capaz de reconhecer seres humanos pela câmera□ O sistema deve ser capaz de manipular corretamente os frames recebidos□ O sistema deve ser capaz de enviar adequadamente a mensagem SMS para os celulares do banco de dados |

2.4 – Elementos de projeto

2.4.1 – Máquina de estados

Baseado nos cenários identificados e requerimentos construídos, tem-se a seguinte proposição para a máquina de estados da solução, considerando o cenário 6, que é o cenário ideal de funcionamento do sistema.

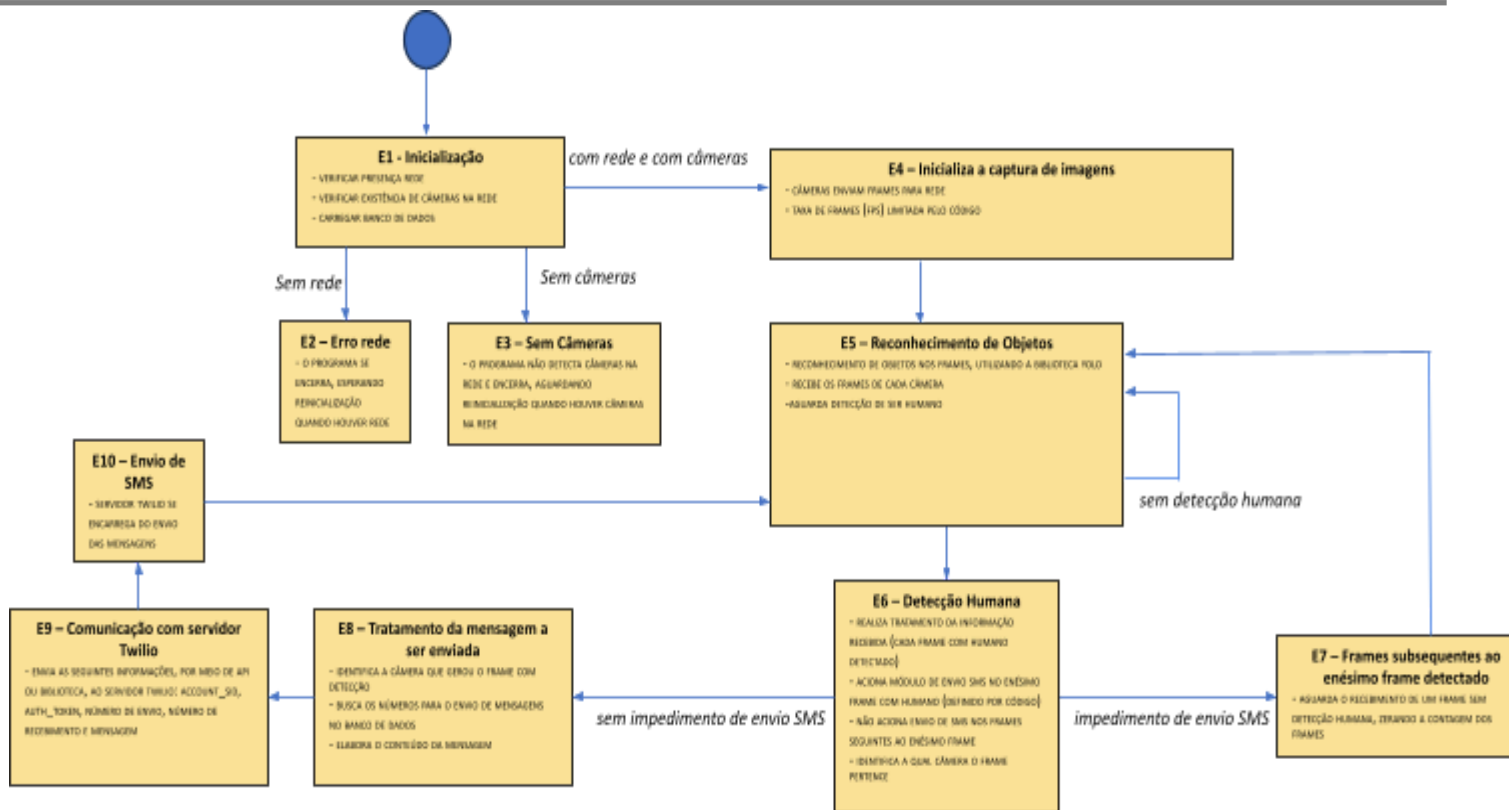


Figura 2.4.1.1: Máquina de estados de uma MU.

2.4.2 – Interfaces de usuário

Para fins de caracterização do sistema, a figura na sequência ilustra a interface desta aplicação indicando alguns de seus recursos previstos.

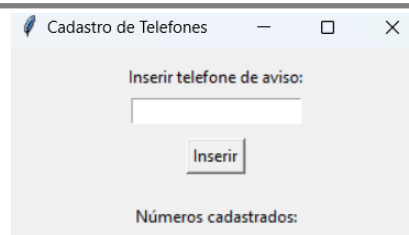


Figura 2.4.2.1: Interfacecadastro de Usuário

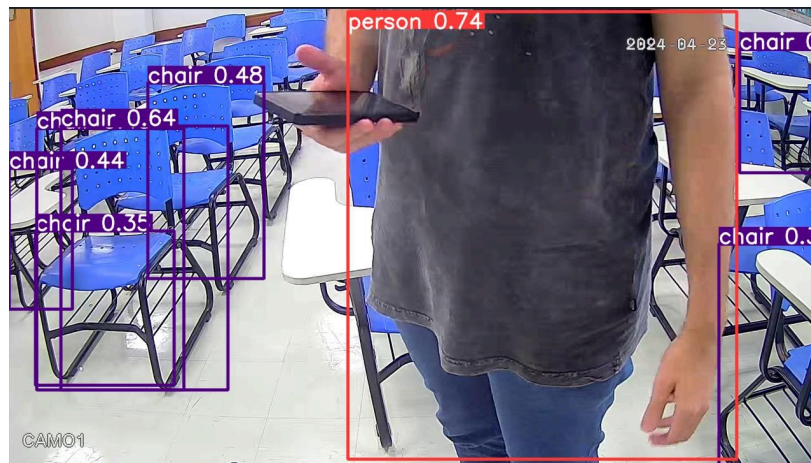


Figura 2.4.2.2: Interface da câmera

A primeira interface serve apenas para cadastrar novos números no banco de dados. É uma interface simples e fácil de entender, ao clicar no botão de inserir, o número escrito na lacuna de cima é adicionado ao banco, no bloco abaixo, são exibidos todos os números cadastrados no banco. Na versão apresentada, como usamos apenas um número devido à limitação do twilio, ela não foi necessária.

A segunda interface é gerada pelo frame com detecção humana captada pela biblioteca YOLO, constantemente os frames das câmeras são mostrados na tela do computador, e quando há detecção e reconhecimento humano, este frame é mostrado em uma aba diferente, possibilitando o usuário salvar a imagem capturada.

3 – Modelagem

3.1 – Blocos de elementos principais

Na sequência é mostrado um conjunto de diagramas de blocos para exemplificar a arquitetura do sistema. Cada bloco é um objeto e estes são os principais objetivos previstos na solução. As setas indicam o fluxo das informações.

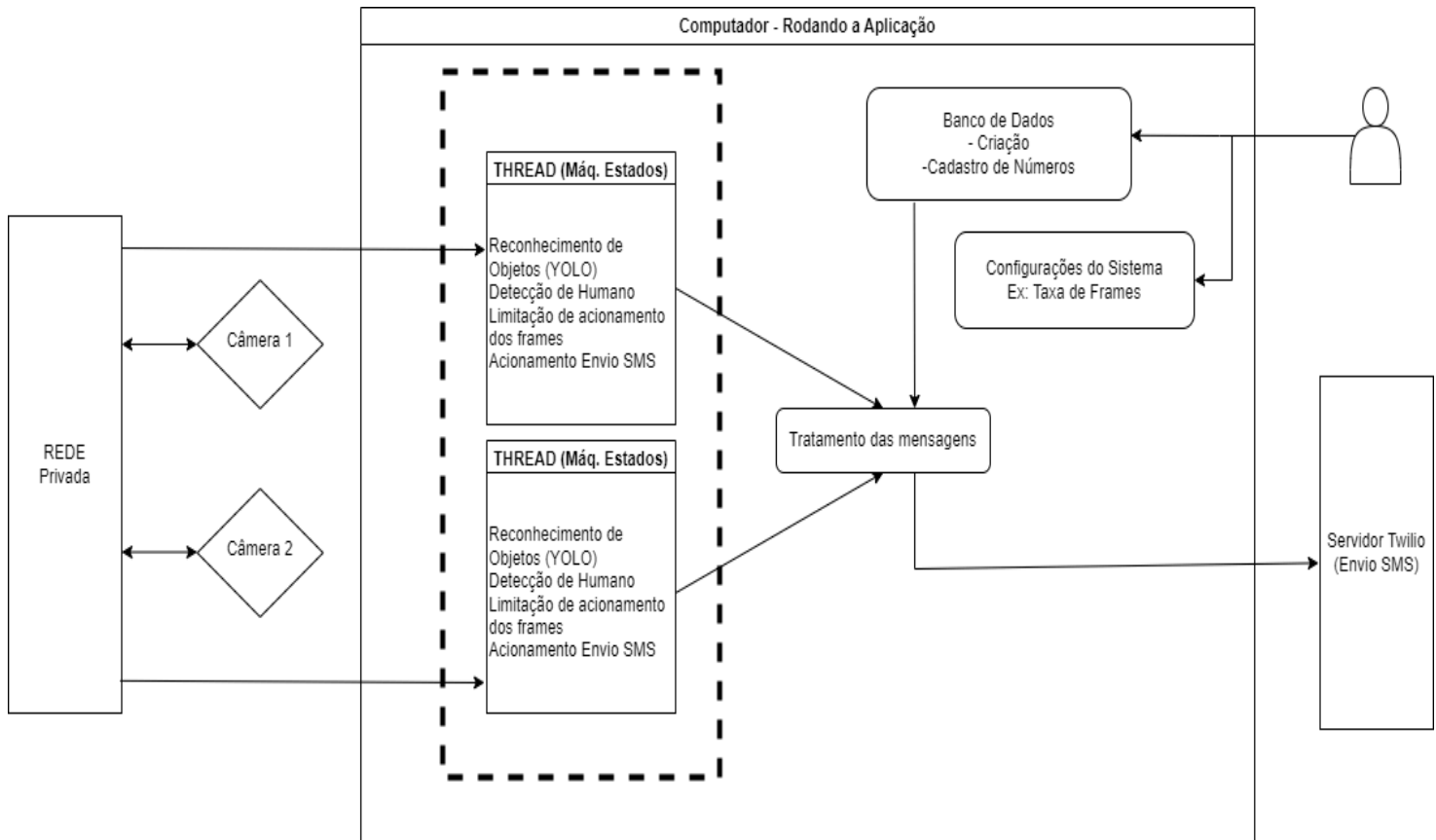


Figura 3.1.1: Diagrama de blocos



3.2 – Fluxo geral de mensagens

A figura na sequência ilustra resumidamente as mensagens que são transmitidas em todas as etapas da aplicação

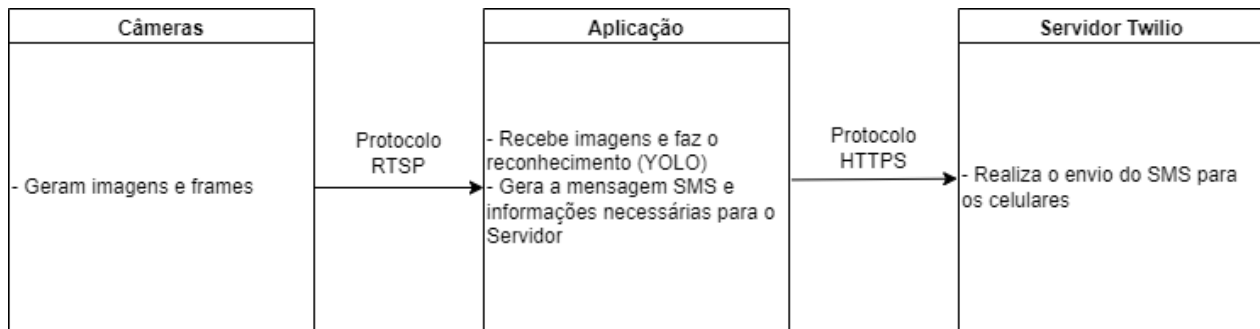


Figura 3.2.1: Fluxo geral das mensagens

Como visto anteriormente, as câmeras fazem a captura das imagens e geram os frames, que serão enviados para a rede, onde o computador que hospeda o sistema irá capturar os pacotes dos frames, comunicando utilizando o protocolo RTSP, que é tratado pela biblioteca YOLO, podendo ser transportado sobre TCP ou UDP.

Assim como também foi visto, a aplicação faz o tratamento e manipulação dos frames, acionando e limitando o envio de SMS quando necessário. Quando acionado, a aplicação faz a busca em um banco de dados para a obtenção dos elementos necessários para o envio da mensagem (já citados anteriormente). A comunicação com a plataforma de envio Twilio é feita com o protocolo https sobre TCP para a garantia de entrega dos dados da mensagem.

Para entender melhor esta dinâmica, elas são tratadas individualmente com mais detalhes nas próximas subseções.

3.4 – Modelagem detalhada dos recursos

3.4.1 – Envio de imagens da câmera

A figura na sequência ilustra como deve ser o comportamento do fluxo dos frames entre a câmera e aplicação:

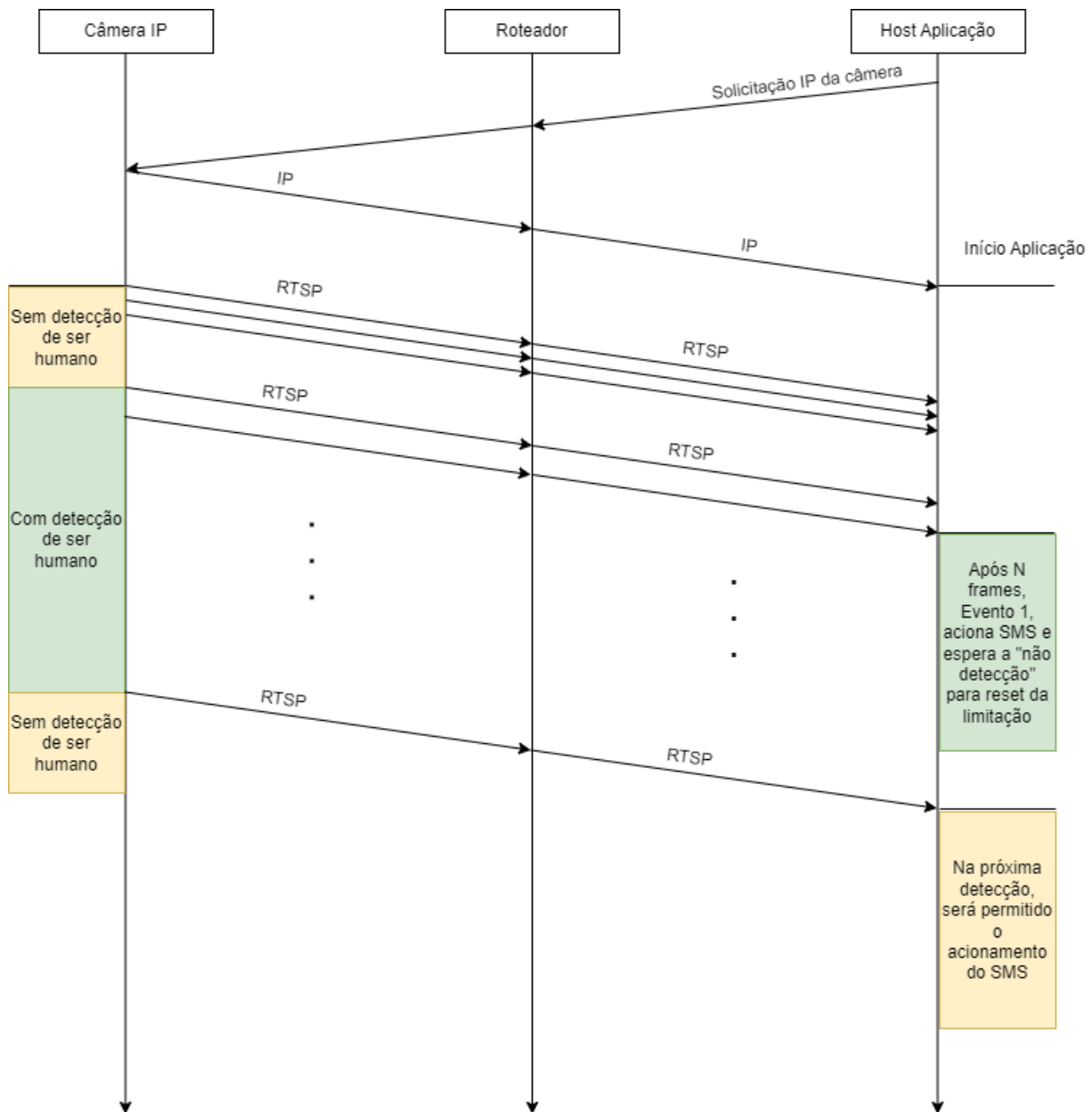


Figura 3.4.1.1: Diagrama pacotes e eventos associados à captura das imagens e reconhecimento de humano

Com base no diagrama mostrado na figura anterior, descreve-se agora a detecção de ser humano pelos frames e seu tratamento.

Tabela 3.4.1.1: Algoritmos equivalentes aos eventos vistos na Figura 3.4.1.1.

| Evento/Ação | Algoritmo |
|-----------------|---|
| Evento 1 | 1) Receber frames enviados pela câmera por RTSP # Utilizar threads para o uso de múltiplas câmeras |

| | |
|------------------------------------|---|
| (Reconhecimento e detecção Humana) | <ol style="list-style-type: none"> 2) Realizar o reconhecimento de Objetos (YOLO) <ol style="list-style-type: none"> i. Inicializar contagem de frames com detecção humana em 0; ii. Inicializar limitador de acionamento do módulo SMS em 1 (liberar acionamento) iii. Inicializar número mínimo de frames para acionamento do SMS em n (arbitrário) . <ol style="list-style-type: none"> a. Case Estado.E6 (Detectou humano): <ol style="list-style-type: none"> i. Incrementar contagem em 1. b. Case Estado.E5 (Não Detectou humano): <ol style="list-style-type: none"> i. Resetar contagem ii. Resetar limitador de acionamento 3) Case contagem de frames > número mínimo de frames: <ol style="list-style-type: none"> i. limitar o acionamento = 0 ii. acionar o módulo de SMS <p>#Printar/Salvar frame com detecção (opcional)</p> |
|------------------------------------|---|

3.4.2 – Envio de SMS

A figura na sequência ilustra como deve ser o fluxo da montagem até o envio do SMS..

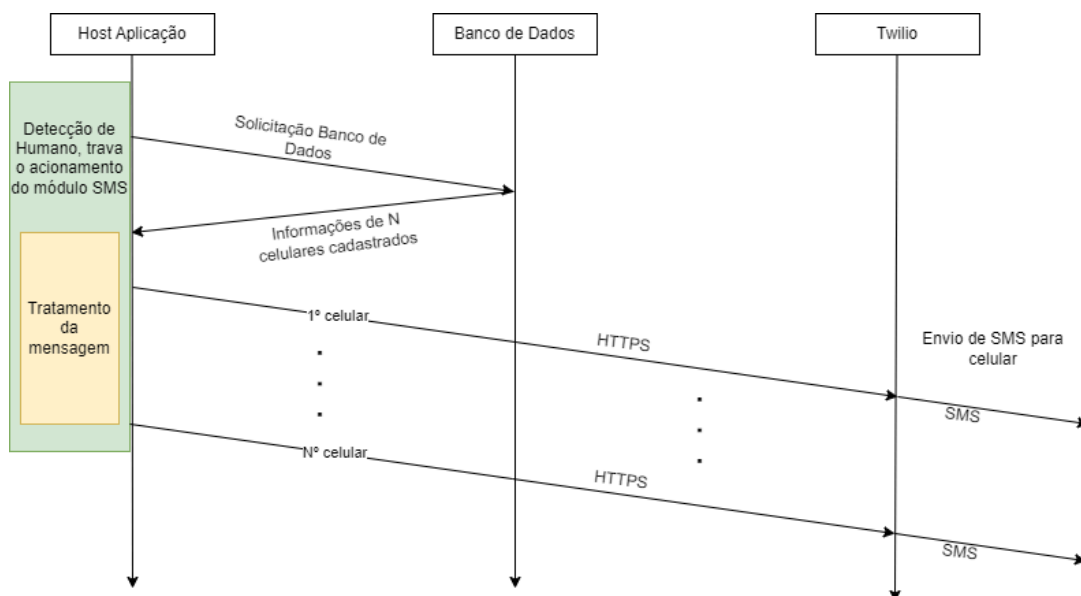


Figura 3.4.2.1: Diagrama pacotes e do envio de SMS

É de extrema importância ressaltar que este é o diagrama desejado quando temos vários celulares cadastrados. Como não foi possível utilizar uma conta twilio a não ser a trial, a versão apresentada envia a mensagem apenas para um celular, eliminando a necessidade de uma busca em um banco de dados, desta forma, com apenas um número, há apenas o envio de uma requisição pela API o twilio e apenas um SMS.

Com base no diagrama mostrado na figura anterior, descreve-se agora o algoritmo da montagem do sms. É importante ressaltar também, que neste sistema, o host da aplicação é o mesmo do banco de dados.



Tabela 3.4.2.1: Algoritmos equivalentes aos eventos vistos na Figura 3.4.2.1.

| Evento/Ação | Algoritmo |
|---|---|
| Evento 2 (envio de mensagens) | <ol style="list-style-type: none">1) Solicita dados dos números de telefone2) Com os dados recebidos, monta uma mensagem com a identificação da câmera e manipula os valores auth_token, account_sid, twilio_number e phone_number3) Para cada número, envia uma comunicação HTTPS pela API do servidor Twilio4) Servidor Twilio é encarregado de entregar os pacotes SMS para |

Os parâmetros necessários para o envio das mensagens, já citados anteriormente são:

- account_sid
- auth_token
- twilio_number
- phone_number
- message

No escopo da mensagem, é a mensagem que o usuário irá receber por SMS.

A API do Twilio pode ser utilizada via biblioteca do python ou por comando curl, realizando uma conexão HTTPS. A URL da API é :

<https://api.twilio.com/2010-04-01/Accounts/AC181d5aa4483d49b433293e0ecd022abf/Messages.json> e por ela enviamos as mensagens e os parâmetros para que o processo ocorra. O comando, no Windows, onde devemos substituir cada valor, é o seguinte:

```
curl {url} -X POST {phone_number} {twilio_number} {message} {account_sid}:{auth_token}
```

Tabela 3.4.2.2: Elementos necessários para o envio do SMS

| Campo | Valores | Significado |
|---------------|---------|--|
| account_sid | String | Identificador de conta do Twilio |
| auth_token | String | Token necessário para autenticação no servidor |
| phone_number | String | Número que receberá mensagem |
| twilio_number | String | Número pelo qual o Twilio enviará a mensagem |
| message | String | Mensagem a ser enviada por SMS |
| url | String | URL da API |