

**Uniruy  
Paralela**

**APLIC. DE CLOUD, IOT E INDÚSTRIA 4.0 EM PYTHON**

**Nome do(s) discente(s): Kauan Nunes Santos Vita  
Nome do(a) professor(a): VITOR EMMANUEL ANDRADE**

**2025  
Salvador/BA**

## Sumário

1. DIAGNÓSTICO E TEORIZAÇÃO.....	3
1.1. Identificação das partes interessadas e parceiros .....	3
1.2. Problemática e/ou problemas identificados.....	3
1.3. Justificativa.....	3
1.4. Objetivos/resultados/efeitos a serem alcançados (em relação ao problema identificado e sob a perspectiva dos públicos envolvidos).....	3
1.5. Referencial teórico (subsídio teórico para propositura de ações da extensão) .....	3
2. PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO .....	3
2.1. Plano de trabalho (usando ferramenta acordada com o docente) .....	3
2.2. Descrição da forma de envolvimento do público participante na formulação do projeto, seu desenvolvimento e avaliação, bem como as estratégias pelo grupo para mobilizá-los. ....	4
2.3. Grupo de trabalho (descrição da responsabilidade de cada membro) .....	4
2.4. Metas, critérios ou indicadores de avaliação do projeto .....	4
2.5. Recursos previstos .....	4
2.6. Detalhamento técnico do projeto.....	4
3. ENCERRAMENTO DO PROJETO .....	4
3.1. Relatório Coletivo (podendo ser oral e escrita ou apenas escrita).....	4
3.2. Avaliação de reação da parte interessada .....	4
3.3. Relato de Experiência Individual.....	5
3.1. CONTEXTUALIZAÇÃO.....	5
3.2. METODOLOGIA .....	5
3.3. RESULTADOS E DISCUSSÃO:.....	5
3.4. REFLEXÃO APROFUNDADA .....	6
3.5. CONSIDERAÇÕES FINAIS.....	6

## 1. DIAGNÓSTICO E TEORIZAÇÃO

### 1.1. Identificação das partes interessadas e parceiros

Interessados:

- Kauan Nunes Santos Vita: Interesse em IoT e aplicações práticas.
- VITOR EMMANUEL ANDRADE: Professor da disciplina, buscando identificar o nível de aprendizado do aluno.

### 1.2. Problemática e/ou problemas identificados

Dificuldade de implementar sistemas de captura e transmissão de imagens em tempo real com custo acessível.

### 1.3. Justificativa

Uma vez que iremos utilizar os conhecimentos obtidos na aula para desenvolver o projeto de IOT, se torna interessante o uso desses conhecimentos com o equipamento disponibilizado para o trabalho

### 1.4. Objetivos/resultados/efeitos a serem alcançados (em relação ao problema identificado e sob a perspectiva dos públicos envolvidos)

É necessário entregar um protótipo funcional, usando o ESP-32, para captura e transmissão de imagens, via WI-FI.

### 1.5. Referencial teórico (subsídio teórico para propositura de ações da extensão)

Os referenciais usados para a formulação do projeto, foi o professor da disciplina e pesquisas na internet na busca da resolução de problemas que foram ocorrendo durante o desenvolvimento.

## 2. PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO

### 2.1. Plano de trabalho (usando ferramenta acordada com o docente)

ETAPA	TAREFA	RECURSO
Pesquisa Teórica	Identificar qual firmware a ser utilizado e buscar cases de sucesso	Professor e internet (youtube, sites e foruns)
Prototipagem	Configuração do ESP32 e testes de captura de imagens	ESP32 e Pycharm
Transmissão	Enviar o resultado do reconhecimento facial pelo ESP32 para o ThingsBoard	Biblioteca do MicroPython (MQTT e Requests)
Teste de Envio	Enviar para o servidor do professor o arquivo em base64 da imagem capturada	Pycharm, ESP32 e MQTT

## 2.2. Descrição da forma de envolvimento do público participante na formulação do projeto, seu desenvolvimento e avaliação, bem como as estratégias pelo grupo para mobilizá-los.

A troca de informações foi feita no período da aula sendo comprovado pela minha presença nos dias de aula e envio do arquivo em base64 da captura feita pelo ESP32.

## 2.3. Grupo de trabalho (descrição da responsabilidade de cada membro)

Pesquisa e desenvolvimento: Pesquisar firmware, programar o ESP32-CAM e testar funcionalidades.

Documentação: Elaborar este relatório.

## 2.4. Metas, critérios ou indicadores de avaliação do projeto

Protótipo Funcional: Informações enviadas sem falhas.

## 2.5. Recursos previstos

Materiais:

- ESP32-CAM– Disponibilizado pelo professor.
- Notebook – Recurso próprio.
- Wokwi – Emulador do ESP32.

Institucionais:

- Orientação do docente (reuniões semanais).

## 2.6. Detalhamento técnico do projeto

Hardware:

- ESP32-CAM

Software:

- Firmware em MicroPython para ESP32-CAM
- Transmissão de dados via MQTT e Requests.

# 3. ENCERRAMENTO DO PROJETO

## 3.1. Relato Coletivo:

Mesmo com o impeditivo da falta de espaço no esp32 para realizar a etapa do reconhecimento facial, foi possível contornar a situação fazendo o processamento em um servidor e enviando o resultado desse processamento para o esp32.

### 3.1.1. Avaliação de reação da parte interessada

Os objetivos acordados em sala foram alcançados, devido a isso, considero as partes interessadas satisfeitas com o projeto.

### 3.2. Relato de Experiência Individual (Pontuação específica para o relato individual)

#### 3.2.1. CONTEXTUALIZAÇÃO

O objetivo central do projeto foi desenvolver uma solução de baixo custo para captura de imagens e reconhecimento facial. A arquitetura se baseia em um microcontrolador **ESP32-CAM** atuando para capturar as imagens, que são então processadas por um servidor mais potente rodando em um computador pessoal.

#### 3.2.2. METODOLOGIA

O desenvolvimento ocorreu em um ambiente de *home office*, utilizando equipamentos próprios e da faculdade. O autor do projeto foi o próprio sujeito dos testes. O processo foi dividido nas seguintes etapas:

- **Configuração do Ambiente:**
  - O ambiente do servidor foi configurado em um PC com Python, utilizando bibliotecas como **Flask**, **Pillow**, **numpy** e, principalmente, **face-recognition** (baseada em **dlib**).
- **Desenvolvimento do Cliente (ESP32-CAM):**
  - **Conectividade:** O primeiro passo foi desenvolver o código para a conexão do ESP32 a uma rede Wi-Fi.
  - **Captura de Imagem:** Foi criada uma rotina para inicializar a câmera, configurando a resolução para **QQVGA (160x120)** e o formato para **JPEG**, visando a economia de memória.
  - **Codificação e Envio:** A imagem capturada foi codificada em **Base64** para ser enviada de forma segura dentro de um payload JSON via MQTT.
- **Desenvolvimento do Servidor (Computador):**
  - **Banco de Faces:** Uma pasta local chamada `known_faces` foi utilizada para armazenar as imagens de referência para o treinamento do algoritmo.
  - **Lógica de Reconhecimento:** O servidor decodifica a imagem em Base64, e comparar os rostos detectados com os rostos já conhecidos. O resultado (o nome da pessoa ou "Desconhecido") era retornado ao ESP32.
- **Integração e Testes:**
  - A fase de testes envolveu a execução simultânea do servidor no PC e do script no ESP32, com a depuração de erros comuns de comunicação.

#### 3.2.3. RESULTADOS E DISCUSSÃO:

Observações e Resultados: O sistema, funcionou como o esperado. O ESP32 capturava a imagem, enviava ao servidor, e em poucos segundos (geralmente de 2 a 4 segundos de latência total), eu recebia o resultado do reconhecimento no console do microcontrolador. A precisão do reconhecimento foi boa em condições de iluminação

favoráveis, mas decaía com ângulos desfavoráveis ou baixa luz, o que é esperado do modelo utilizado.

#### 3.2.4. REFLEXÃO APROFUNDADA

Esta experiência prática permitiu solidificar diversos conceitos teóricos. A teoria da arquitetura **Cliente-Servidor** foi a base de todo o projeto, com o ESP32-CAM atuando como um e o PC como um servidor.

#### 3.2.5. CONSIDERAÇÕES FINAIS

Extensão: Como próximo passo, o sistema poderia ser integrado a um atuador, como um relé para abrir uma fechadura eletrônica, criando um sistema de controle de acesso completo. Outra melhoria seria criar uma interface web no servidor para cadastrar novos rostos dinamicamente, sem precisar adicionar arquivos de imagem manualmente na pasta.

Pesquisa: Uma linha de pesquisa interessante seria otimizar o processo. Poderíamos investigar o uso do protocolo MQTT em vez de HTTP.



