MVP de Monitoramento e Predição de Eventos Meteorológicos com ESP32 + Machine Learning

# Grupo 46 – FIAP

Thiago Scutari – RM562831 | thiago.scutari@outlook.com

Henrique Ribeiro Siqueira – RM565044 | henrique.ribeiro1201@gmail.com

Mariana Cavalcante Oliveira – RM561678 | mari.kvalcant@gmail.com  
  
Github: <https://github.com/ThiagoScutari/global_solution_1.git>

Video: https://youtu.be/lV0VRtBctgo

# 1. Introdução

Este projeto propõe o desenvolvimento de um sistema inteligente para monitoramento e predição de eventos climáticos extremos, como queimadas, geadas e tempestades. Utilizando sensores simulados e técnicas de aprendizado de máquina, o sistema permite prever fenômenos com base em dados ambientais, contribuindo para ações preventivas e de mitigação.  
  
A solução se baseia em um sensor DHT22 acoplado a um microcontrolador ESP32, ambos simulados por meio da plataforma Wokwi. Os dados são processados em tempo real e analisados por um modelo de Machine Learning previamente treinado em Python, possibilitando a classificação automática dos eventos meteorológicos.

# 2. Desenvolvimento

## 2.1 Arquitetura Geral

A arquitetura do sistema integra os seguintes componentes principais:  
  
- Coleta de dados: via sensor DHT22 simulado no ESP32;  
- Transmissão: utilizando protocolo RFC2217, que permite a leitura remota da porta serial;  
- Processamento: scripts em Python realizam a predição com base nos dados coletados;  
- Visualização: gráficos em Matplotlib.  
  
Fluxo de dados: ESP32 (DHT22) → RFC2217 → Python (preditivo) → CSV → Dashboard

## 2.2 Justificativas Técnicas

- O uso da plataforma Wokwi permite testes em ambiente 100% simulado, ideal para protótipos educacionais.  
- O protocolo RFC2217 viabiliza a comunicação serial remota, simulando um cenário real de IoT.  
- A escolha do algoritmo Random Forest justifica-se por sua robustez na classificação com dados tabulares.  
- O Scikit-Learn proporciona fácil treinamento e exportação de modelos com joblib.

## 2.3 Implementação

### a) Firmware ESP32 (main.ino)

O código simula a coleta de dados ambientais com envio via serial:  
Serial.print(id);  
Serial.print(",");  
Serial.print(temp);  
Serial.print(",");  
Serial.println(umid);

### b) Treinamento do Modelo (treinar\_modelo.py)

- Geração de dados sintéticos balanceados;  
- Treinamento com validação cruzada;  
- Salvamento em modelo\_evento.pkl.

### c) Predição em Tempo Real (prever\_evento\_metereologico.py)

- Leitura contínua da porta serial;  
- Classificação instantânea e gravação em CSV;  
- Confiabilidade exibida a cada predição.

### d) Dashboard Terminal (dashboard\_terminal.py)

- Gráficos temporal e de barras;  
- Interface simplificada com Matplotlib.

# 3. Resultados Esperados

- Classificação precisa de eventos meteorológicos com base em temperatura e umidade;  
- Atualização em tempo real dos dados sensoriais;  
- Visualização gráfica clara das condições ambientais;  
- Portabilidade e escalabilidade da solução para sensores reais em campo.

A lógica da classificação segue as seguintes regras:

|  |  |  |
| --- | --- | --- |
| Evento | Temperatura (°C) | Umidade (%) |
| Geada | < 4 | > 70 |
| Queimada | > 33 | < 40 |
| Tempestade | 25 - 32 | > 80 |
| Normal | outros casos |  |