



# **Aplicação de TinyML em Sensores Virtuais para Monitoramento da Qualidade do Ar em Ambientes Industriais**

Tayco Murilo Santos Rodrigues

Orientador: Prof. Dr. Thiago Damasceno Cordeiro  
Coorientador: Prof. Dr. Frede de Oliveira Carvalho

Maceió  
22 de Novembro de 2024

# Plano de apresentação



**01**

# **Motivação**

# Definição do Problema

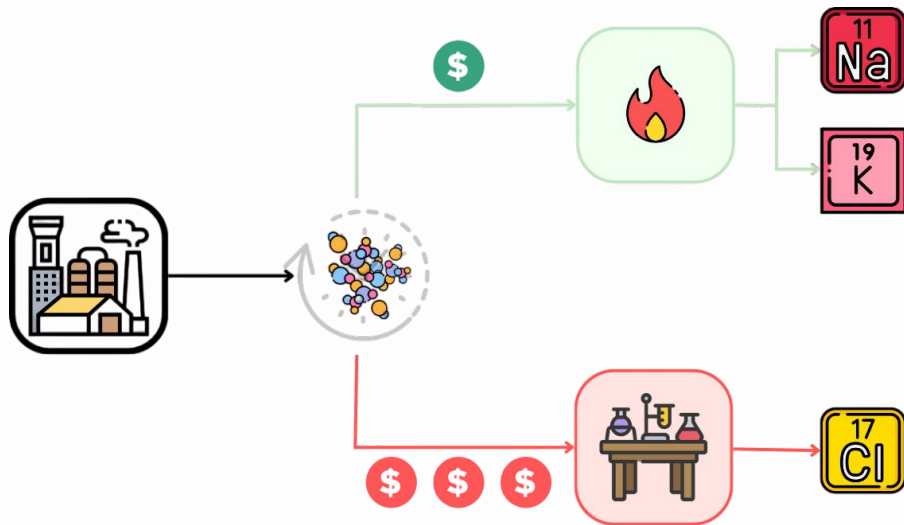


Figura 1: Metodologia usual. O fluxo em verde indica um processo mais simples e econômico (\$), enquanto o fluxo em vermelho representa um processo mais caro (\$\$\$), com maior complexidade e custo operacional. Fonte: Autor.

# Objetivos



**Criação de um  
sensor virtual**



**Barateamento  
do processo**



**Predição de  
variáveis**

**02**

# **Metodologia**

# Metodologia Proposta com TinyML

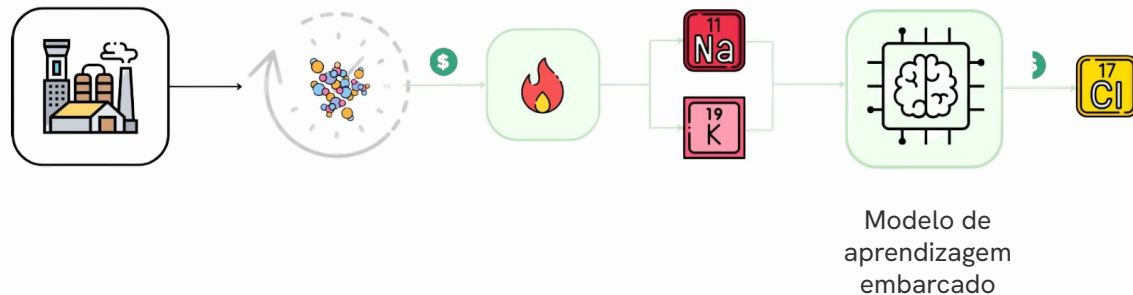


Figura 2: Metodologia proposta. O fluxo apresenta medições diretas de sódio (Na) e potássio (K), seguidas de análise preditiva com inteligência artificial embarcada (TinyML) para estimar os níveis de cloro (Cl). O processo é simplificado e otimizado em termos de custo (\$) em comparação com métodos tradicionais.

# Escolha e Análise dos Dados

## 1. Dados Utilizados

- Base de dados com 198 registros numéricos.
- Três variáveis principais: sódio, potássio e cloro.

## 2. Objetivo da Análise

- Verificar a adequação das variáveis como preditora.

## 3. Correlação entre Variáveis

- Alta correlação entre sódio e cloro (0,94).
- Correlação moderada entre potássio e cloro (0,48).



# Estatísticas Descritivas das Variáveis

## 1. Variabilidade dos Dados

	Média	Mediana	Desvio Padrão
Sódio	8,37	7,90	5,75
Potássio	4,48	4,40	2,71
Cloro	16,55	15,30	9,40

## 2. Necessidade de Normalização

# Distribuição dos Dados

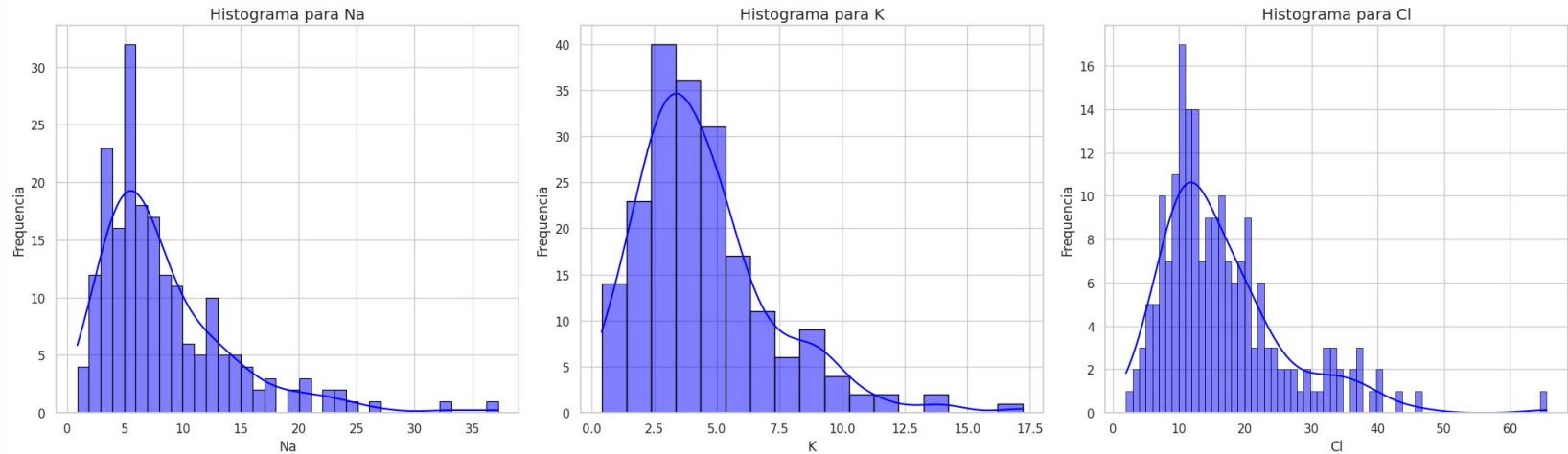


Figura 3: Distribuição dos Dados Originais. Fonte: Autor.

# Distribuição dos Dados

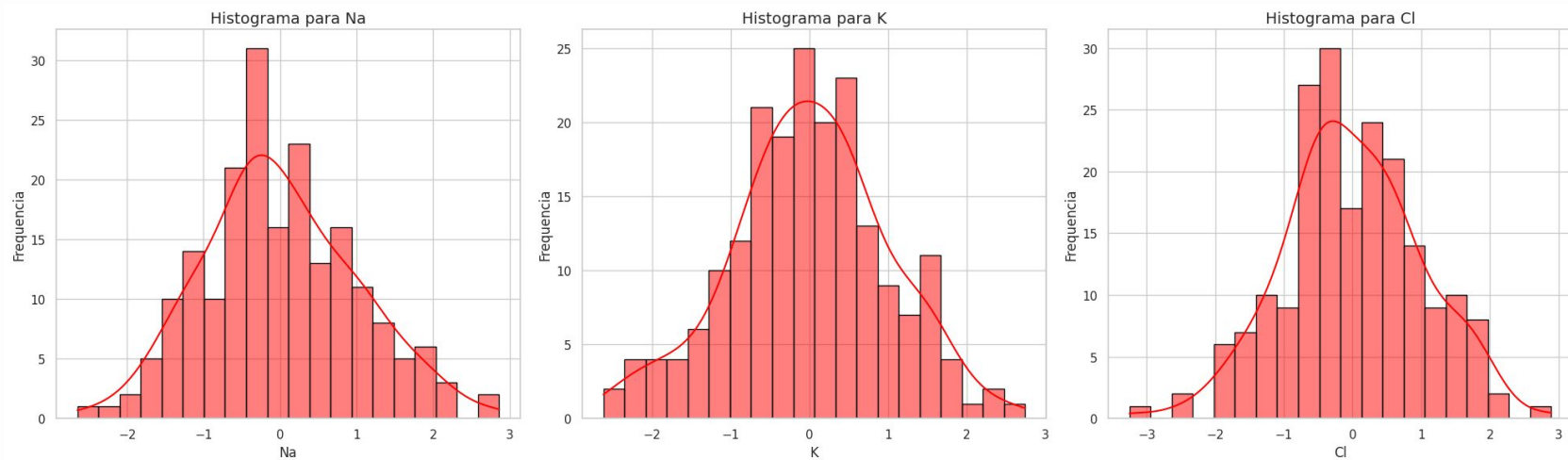
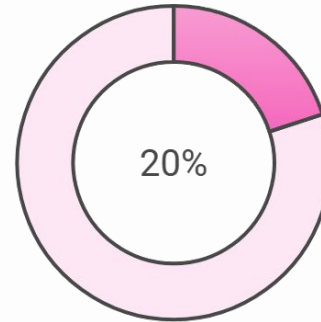
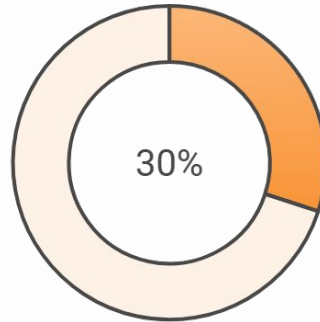
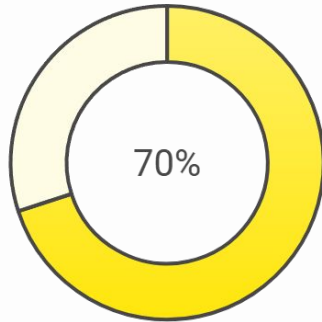


Figura 4: Distribuição dos Dados Após Normalização. Fonte: Autor.

# Divisão dos Dados para Treinamento e Teste

## 1. Estratégia de Divisão

- 70% Treinamento / 30% Teste: Baseia-se em uma divisão padrão para avaliar o modelo.
- Validação: 20% dos dados de treino foram reservados para validação.



# Escolha do Modelo: Multilayer Perceptron (MLP)



- Escolhido pela capacidade de aproximação universal
- Utilizado para resolver problemas de regressão

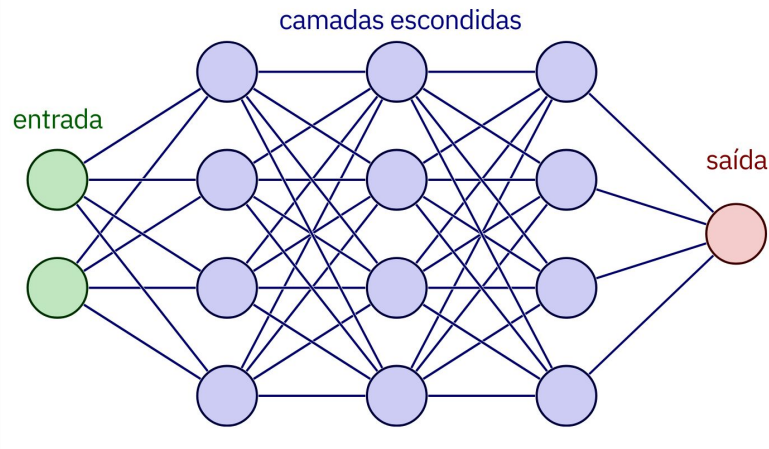


Figura 5: Diagrama de uma MLP com 3 camadas ocultas. Fonte: Autor.

# Estrutura da Rede Neural

## 1. Camada Oculta 🧠

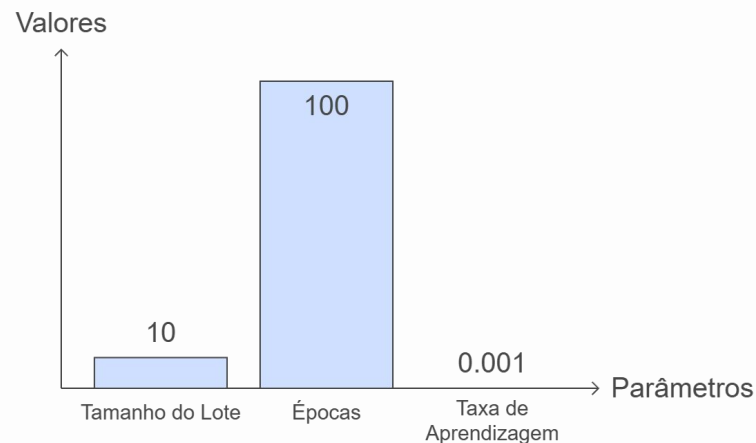
- Uma camada oculta com 7 neurônios.
- Função de ativação: ReLU, devido ao bom desempenho em problemas de regressão.

## 2. Camada de Saída 🔄

- Função de ativação linear, ideal para prever valores contínuos.

## 3. Função de Perda:

- Erro Quadrático Médio (MSE)



# Exportação do Modelo para TensorFlow Lite

## 1. Objetivo da Exportação

- Preparar o modelo para rodar no ESP32-S3.
- Otimizar o modelo.

## 2. Benefícios da Quantização

- Reduz tamanho e latência do modelo.
- Adequado para dispositivos com recursos limitados.

# Execução dos Modelos TFLite no Python

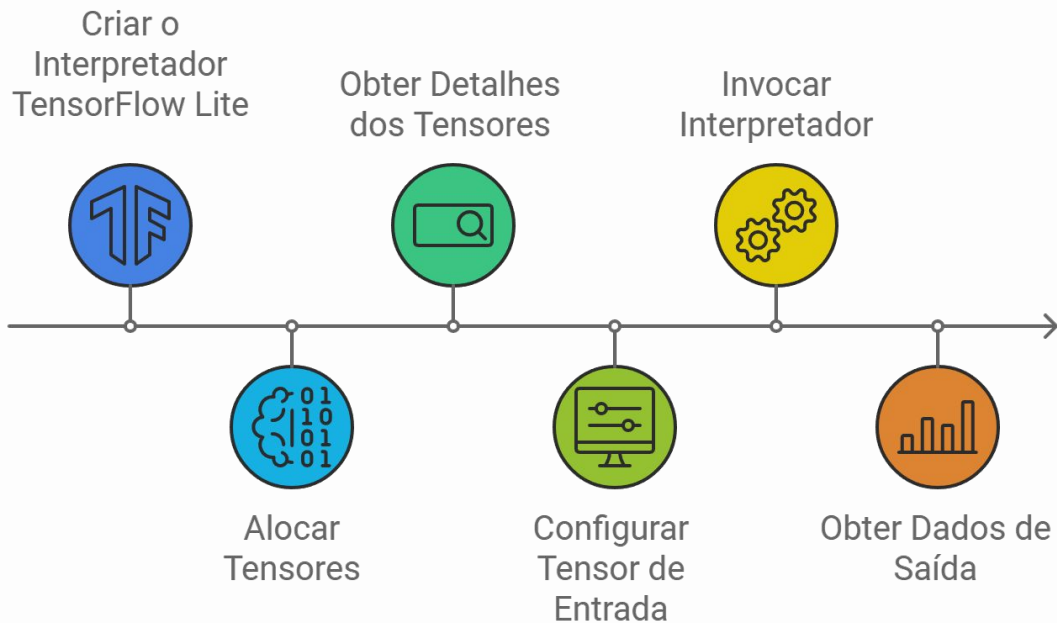


Figura 7: Distribuição dos Dados Originais. Fonte: Autor.



# Exportação do Modelo para o Microcontrolador

## Construção do Interpretador:

- Inicialização com as operações necessárias para o modelo (e.g., fully connected, ReLU).
- Configuração da arena de tensores e associação das operações ao modelo.

## Alocação dos Tensores:

- Alocação dos tensores de entrada, pesos e saída na memória.
- Verificação de erros durante a alocação para garantir a execução.

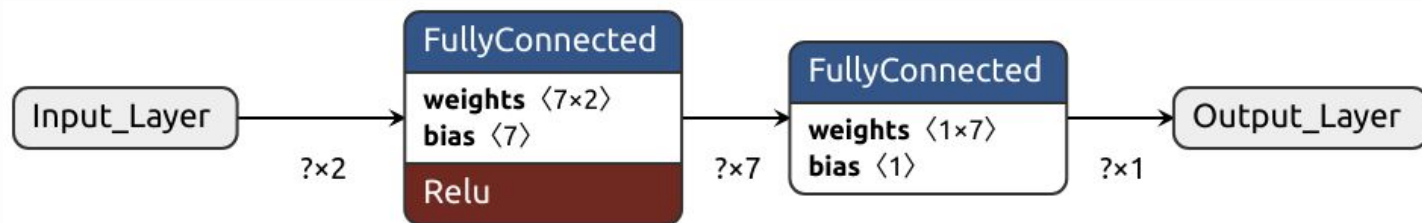


Figura 7: Arquitetura do modelo otimizado. Fonte: Autor.

# Inferência no Microcontrolador

```
1 interpreter = modelo.tflite
2 enquanto True faz:
3     Na, K = Valores presentes no ambiente
4     Na_norm, K_norm = normalize([Na, K], [Na_mean, K_mean], [Na_scale, K_scale])
5     input_buffer[0] = Na_norm
6     input_buffer[1] = k_norm
7     inference(interpreter, input_buffer*)
8     Cl_norm = output->data.f[0]
9     resultado = reverse(Cl_norm, Cl_mean, Cl_scale);
10    exibir(resultado)
11 fim enquanto
```

Figura 8: Pseudocódigo para o fluxo principal. Fonte: Autor.

**03**

# **Resultados**

# Volume dos Modelos Quantizados

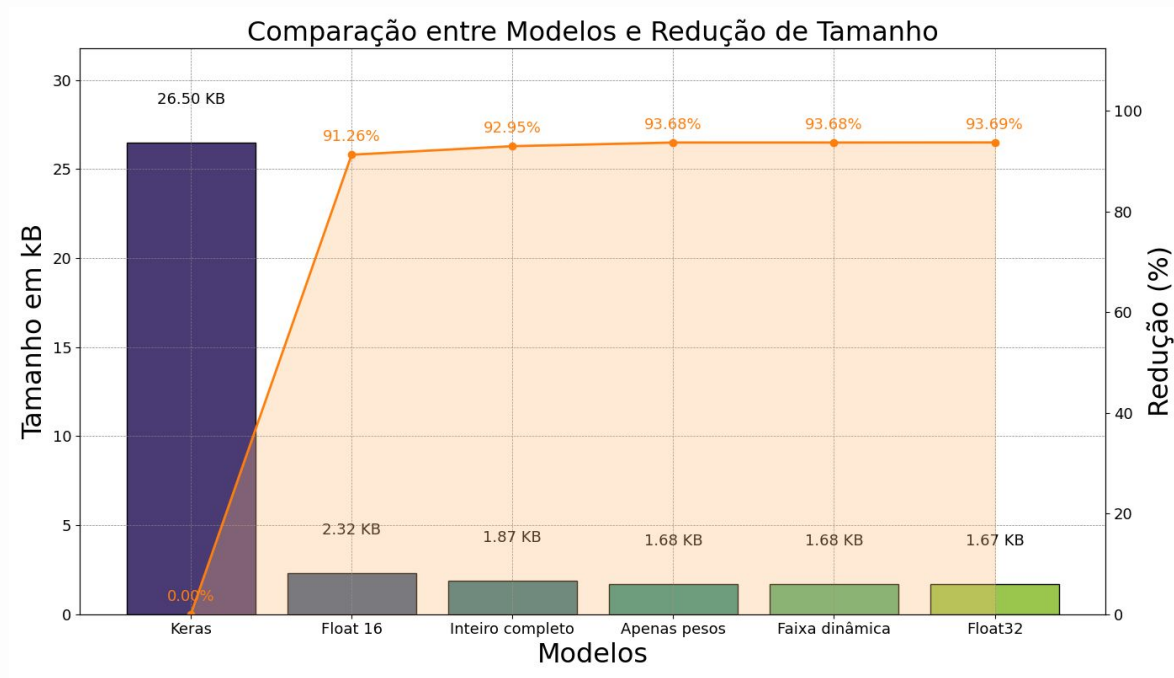


Figura 9: Tamanho dos modelos quantizados e sua porcentagem de redução. Fonte: Autor.

# Gráfico de Dispersão - Modelo Original (Keras)

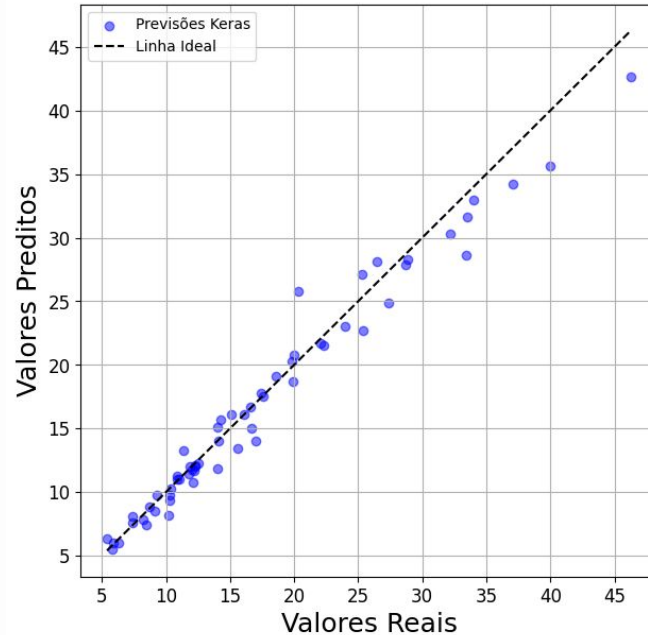
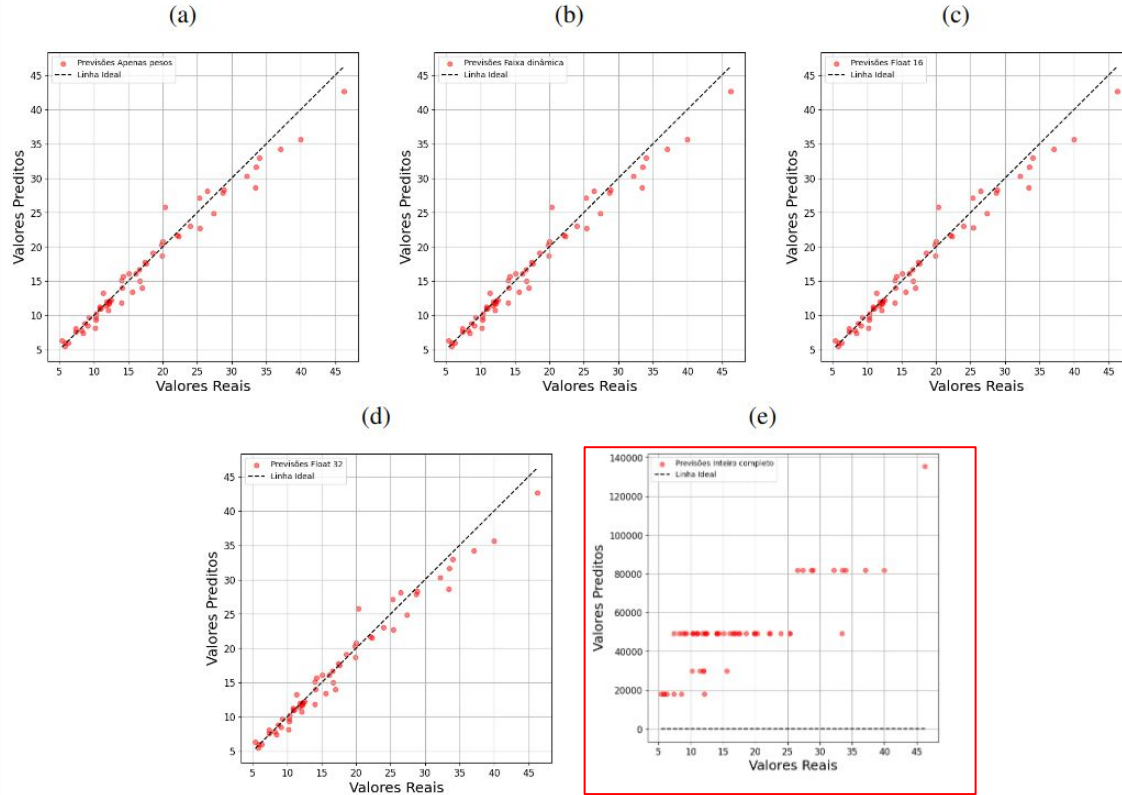


Figura 9: Gráfico de dispersão - Modelo Keras. Fonte: Autor.

# Gráficos de Dispersão - Modelos Quantiz



# Comparação das Métricas de Desempenho

Modelo	MSE	RMSE	MAE	$R^2$	MEDAE	MAPE
Keras	2,78087	1,66759	1,14554	0,96778	0,70469	0,06412
Float32	2,78087	1,66759	1,14554	0,96778	0,70469	0,06412
Float16	2,77471	1,66575	1,14435	0,96786	0,70452	0,06409
Apenas Pesos	2,78087	1,66759	1,14554	0,96778	0,70469	0,06412
Faixa Dinâmica	2,78087	1,66759	1,14554	0,96778	0,70469	0,06412
Inteiro Completo	$2,96 \times 10^9$	$5,44 \times 10^4$	$5,04 \times 10^4$	$-3,43 \times 10^7$	$4,94 \times 10^4$	$3,22 \times 10^3$

Tabela 3: Comparativo de Métricas de Desempenho para Modelos Quantizados e Keras. Fonte: Autor.

# Resultados Obtidos do Microcontrolador

```
1 Iniciando casos de testes!  
2  
3 Na:6.20 K:4.80 Cl:13.22 Inf.Time:197μs  
4  
5 Na:14.20 K:17.20 Cl:35.19 Inf.Time:56μs  
6  
7 ...  
8  
9 Na:13.20 K:3.20 Cl:23.28 Inf.Time:54μs  
10  
11 Na:14.50 K:3.30 Cl:25.18 Inf.Time:43μs  
12  
13 Encerrando casos de teste!
```

Figura 11: Exemplo de Saída do Sensor Virtual. Fonte: Autor.



# Precisão do Modelo Embarcado

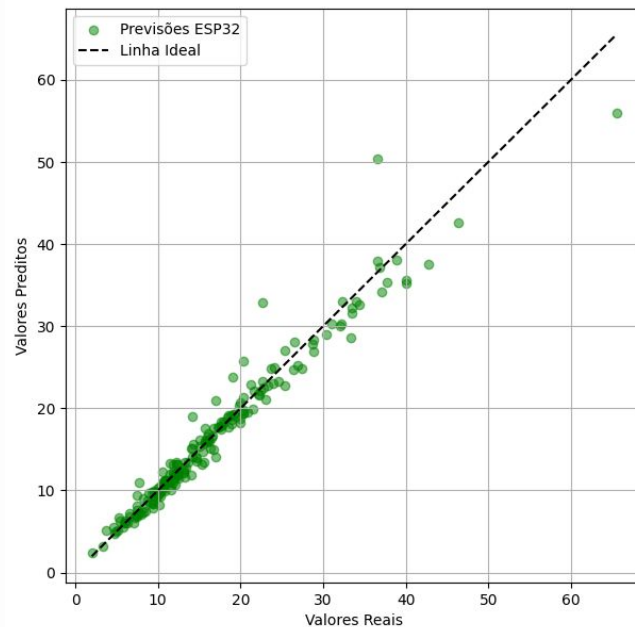


Figura 12: Gráfico de dispersão - Sensor virtual. Fonte: Autor.

# Medição de Corrente no Microcontrolador

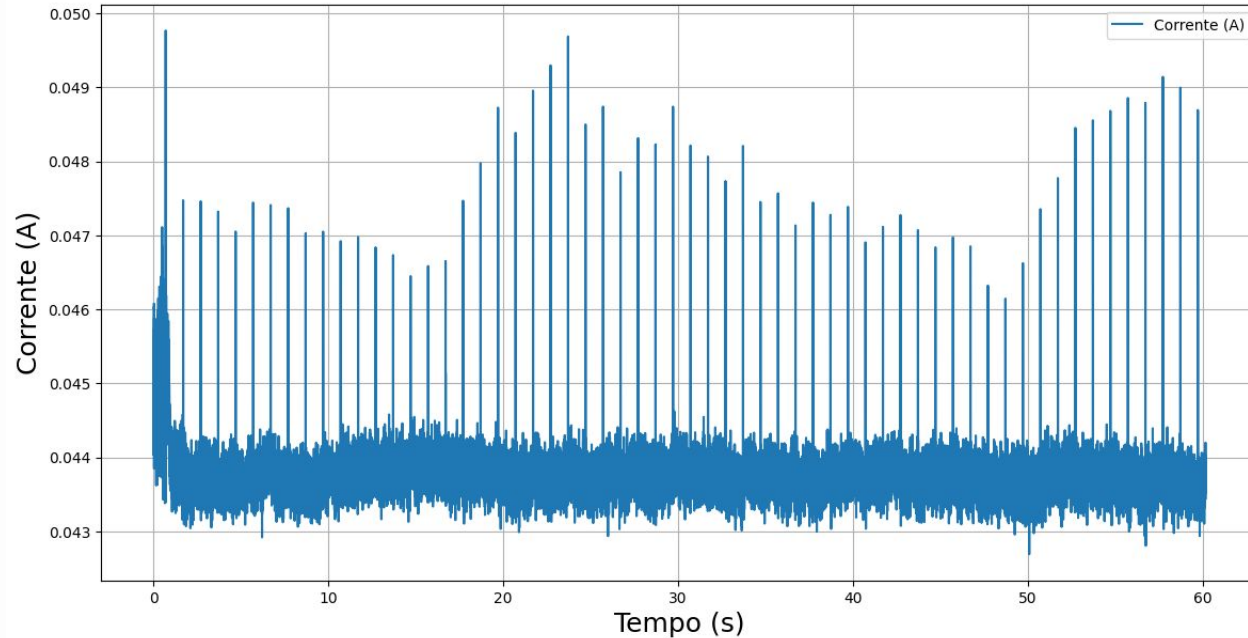


Figura 13: Consumo de Corrente ao Longo do Tempo. Fonte: Autor.

# Métricas

## Métricas de Precisão do Modelo Embarcado

Modelo	MSE	RMSE	MAE	$R^2$	MedAE	MAPE
Original	2,78087	1,66759	1,14554	0,96778	0,70469	0,06412
Embarcado	3.96343	1.99084	1.14136	0.95493	0.68500	0,06978

Tempo Médio de Inferência : 49,01 $\mu$ s

# Comparação entre Metodologia Usual e Proposta

## Objetivo da Comparação:

- Avaliar eficiência, custo e escalabilidade entre a metodologia tradicional e a abordagem com TinyML e sensores virtuais.

## Resumo das Diferenças:

- Metodologia Usual: Técnicas laboratoriais, alto custo, longo tempo de análise.
- Metodologia Proposta: Sensores e TinyML, monitoramento em tempo real, menores custos operacionais.

# Equipamentos Utilizados

Aspecto	Metodologia Usual	Metodologia Proposta
Coleta de Amostras	Amostrador de volume grande	Sensores em tempo real
Medição de Potássio	Fotômetro de chama	Sensores eletroquímicos ou óticos
Medição de Cloro	Titulação argentimétrica em laboratório	Modelo preditivo TinyML
Portabilidade	Equipamentos laboratoriais, não portáteis	Dispositivos portáteis, fácil instalação

Tabela 3: Equipamentos. Fonte: Autor.

# Comparação de Custos

Aspecto	Metodologia Usual	Metodologia Proposta
Investimento Inicial	Alto (equipamentos e reagentes)	Médio a alto (sensores e desenvolvimento)
Custo por Análise	Alto (preparo de amostras, reagentes)	Baixo (custo quase zero após implementação)
Manutenção	Reposição de reagentes e calibração	Calibração periódica dos sensores

Tabela 4: Custo. Fonte: Autor.

# Tempo de Análise

Aspecto	Metodologia Usual	Metodologia Proposta
Tempo de Coleta	Demorado (grande volume de ar)	Monitoramento em tempo real
Tempo de Análise	Demorado (preparo e execução da titulação)	Instantâneo (sensores e modelo TinyML)
Tempo de Resposta	Lento (análise laboratorial)	Rápido (dados em tempo real)

Tabela 5: Tempo de análise. Fonte: Autor.

# Complexidade Operacional

Aspecto	Metodologia Usual	Metodologia Proposta
Treinamento Necessário	Técnicos especializados	Treinamento básico para operadores
Número de Passos	Coleta, transporte, titulação, análise	Simplificado – sensores e processamento automático
Erros Humanos	Altos (várias etapas)	Baixos (automação reduz intervenção)

Tabela 6: complexidade operacional. Fonte: Autor.



# Precisão e Sensibilidade

Aspecto	Metodologia Usual	Metodologia Proposta
Precisão	Alta (titulação e fotometria)	Alta (modelo bem treinado)
Interferências	Possíveis interferências de outros íons	Sensores calibrados para minimizar interferências

Tabela 7: Precisão. Fonte: Autor.

**04**

# **Conclusão**

**OBRIGADO!!**

