

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
MINAS GERAIS - *CAMPUS* BETIM
BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Bruno Alves de Sousa

**REALIDADE AUMENTADA NA INDÚSTRIA: Implementação de um
sistema de realidade aumentada em um smartphone para monitorar dados
de um CLP**

Betim
2025

BRUNO ALVES DE SOUSA

REALIDADE AUMENTADA NA INDÚSTRIA: Implementação de um sistema de realidade aumentada em um smartphone para monitorar dados de um CLP

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus* Betim, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Daniel Almeida Godinho

Betim
2025

FICHA CATALOGRÁFICA

S725r Sousa, Bruno Alves de

Realidade aumentada na indústria: implementação de um sistema de realidade aumentada em um smartphone para monitorar dados de um CLP / Bruno Alves de Sousa. – 2025.

46 f.: il.

Trabalho de conclusão de curso (Bacharelado em Engenharia de Controle e Automação) - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, Campus Betim, 2025.

Orientação: Prof. Dr. Daniel Almeida Godinho

1. Controladores programáveis. 2. Realidade aumentada. 3. Indústria 4.0. 4. Firebase. 5. Engenharia de Controle e Automação. I. Sousa, Bruno Alves de. II. Título.

CDU: 681.5



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS
Campus Betim
Diretoria de Ensino
Docentes Automação Industrial e Tecnologia da Informação
Rua Itamarati - CEP 32677-564 - Betim - MG
3135976360 - www.ifmg.edu.br

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Aos 21 dias do mês de fevereiro do ano de 2025, às 18:00, nas dependências do IFMG – *Campus* Betim , reuniu-se a banca examinadora presidida por mim, Arthur Hermano Rezende Rosa e demais membros, Daniel Godinho (orientador) e Fernando Thomé. Nesta ocasião o discente do curso de Bacharelado em Engenharia de Controle e Automação, do IFMG – *Campus* Betim , defendeu seu Trabalho de Conclusão de Curso (TCC) intitulado “Realidade Aumentada na Indústria: Implementação de um sistema de realidade aumentada em um smartphone para monitorar dados de um CLP” e foi aprovado , com Nota 83 pontos.

Este resultado reflete o cumprimento parcial dos critérios de avaliação estabelecidos pelo curso e reconhece os esforços e a dedicação do discente e seu orientador no desenvolvimento do seu TCC. O lançamento da nota e o conseqüente encerramento do respectivo processo está condicionado ao cumprimento dos procedimentos pós-defesa conforme previstos nos regulamentos vigentes. Tais procedimentos pós-defesa devem ser finalizados dentro do prazo limite de 20 dias, a contar da data desta ata. O descumprimento destes procedimentos até a data estipulada implicará em atribuição de nota 0 (zero) e conseqüente reprovação.

A sessão foi encerrada às 19:30. Para constar, eu, Arthur Hermano Rezende Rosa, redigi a presente ata que após lida publicamente, foi aprovada e assinada pelo discente e membros da banca examinadora.

Betim, 24 de fevereiro de 2025.



Documento assinado eletronicamente por **Arthur Hermano Rezende Rosa, Professor**, em 24/02/2025, às 22:39, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Daniel Almeida Godinho, Professor Substituto**, em 25/02/2025, às 02:07, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Fernando Thomé de Azevedo Silva, Professor EBTT**, em 05/03/2025, às 08:18, conforme Decreto nº 10.543, de 13 de novembro de 2020.

Dedico este trabalho à minha mãe, Ana Maria, que me ensinou a importância da perseverança e busca pelo conhecimento, apesar das dificuldades da vida. Seu amor e sacrifício me ensinaram a ser mais persistente e alcançar este e muitos objetivos.

À minha avó, Maria Lúcia, que sempre me apoiou e me ensinou à priorizar o mais importante. A sua luta foi o que me trouxe até aqui, onde posso construir meu futuro com conhecimento, não apenas suor.

Aos meus padrinhos, Simone e Jorge, que me acolheram e me guiaram no início da minha vida adulta, o que me deu a base necessária para, mais tarde, ingressar na faculdade. Seu cuidado e palavras de conselho me ensinaram que a grandeza não está no caminho que se escolhe, mas na coragem de trilhá-lo com amor e propósito.

À minha esposa, Grasielli, à sua família, e à todos os nossos amigos, que me trouxeram de volta toda vez que eu duvidava de mim, ou me perdia nas incertezas.

Hoje, cada linha deste trabalho carrega não só meu esforço, mas a força coletiva de quem acredita que sonhos se realizam quando se tem ao lado quem nunca desiste de você.

AGRADECIMENTOS

Agradeço, primeiramente, ao meu orientador, Prof. Daniel Almeida Godinho, pela orientação, paciência e disponibilidade em dedicar horas além do horário convencional ao desenvolvimento da parte prática deste trabalho.

À instituição IFMG - Campus Betim, por me conceder todos os auxílios necessários para que eu tivesse acesso ao melhor ensino possível. A minha permanência e dedicação foram viabilizadas, em grande parte, pelo auxílio financeiro oferecido.

Aos professores Fernando Thomé e Virgil Almeida, por me incentivar e guiar na formalização desta ideia, além do aconselhamento por parte das ferramentas e métodos à minha disposição.

Aos meus familiares e amigos, que me apoiaram durante todo esse processo. A compreensão e carinho de vocês foram fundamentais para que eu concluísse este trabalho.

RESUMO

A monitoração de Controladores Lógicos Programáveis (CLPs) em ambientes industriais frequentemente depende de interfaces estáticas, limitando a interpretação ágil de dados. Este trabalho propõe um sistema de realidade aumentada (RA) mobile, desenvolvido para gerar a visualização de medidores tridimensionais (3D) a partir da imagem da câmera de um *smartphone* em tempo real. A solução integra um aplicativo Android a um CLP Siemens S7-1200, utilizando comunicação via banco de dados em nuvem. Algoritmos de processamento de imagem rodam continuamente sobre a imagem da câmera para dispor medidores 3D (ex.: nível, temperatura) via reconhecimento de marcadores visuais. O sistema operou com estabilidade e baixa latência, demonstrando potencial para substituir interfaces tradicionais em cenários que exigem portabilidade. Foram utilizados conceitos de realidade aumentada, programação orientada a objetos, processamento de imagem e IoT.

Palavras-chave: Realidade Aumentada; Indústria 4.0; Firebase; Realtime; Snap-7.

ABSTRACT

The monitoring of Programmable Logic Controllers (PLCs) in industrial environments often relies on static interfaces, limiting agile data interpretation. This work proposes a mobile augmented reality (AR) system designed to generate the visualization of three-dimensional (3D) gauges from a smartphone camera image in real time. The solution integrates an Android application with a Siemens S7-1200 PLC, using cloud database communication. Image processing algorithms run continuously on the camera feed to display 3D gauges (e.g., level, temperature) via visual marker recognition. The system operated with stability and low latency, demonstrating potential to replace traditional interfaces in scenarios requiring portability. Concepts of augmented reality, object-oriented programming, image processing, and IoT were employed.

Keywords: Augmented Reality; Industry 4.0; Firebase; Realtime; Snap-7.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de fluxo geral pt. 1.	23
Figura 2 – Diagrama de fluxo geral pt. 2.	24
Figura 3 – Capturando a âncora e exibindo o medidor em tempo real.	27
Figura 4 – Ponto de vista do operador: valor em aprox. 50% ou 80°C	28
Figura 5 – Ponto de vista do operador: valor em aprox. 90% ou 112°C	29
Figura 6 – Ponto de vista da câmera do app: valor em aprox. 90% ou 112°C	30
Figura 7 – Ponto de vista do operador: valor em 0% ou 40°C	31
Figura 8 – Medidor ideal encontrado: Energymeter	33
Figura 9 – Medidor adaptado: Temperatura	34
Figura 10 – Medidor adaptado: Nível	34
Figura 11 – Medidor adaptado: RPM	35
Figura 12 – Medidor adaptado: Voltímetro	35
Figura 13 – Programação do CLP: leitura de um sensor de temperatura	36
Figura 14 – Programação do servidor: Código Python	37
Figura 15 – Firebase Realtime Database	38
Figura 16 – Escutando mudanças no banco de dados	39
Figura 17 – Encontrando o ponteiro	40
Figura 18 – Executando a transformação linear do valor para angulo de rotação do ponteiro	41
Figura 19 – Executando a transformação linear do valor para angulo de rotação do ponteiro	42

LISTA DE ABREVIATURAS E SIGLAS

IFMG	Instituto Federal de Minas Gerais
CLP	Controlador Lógico Programável
3D	Tridimensional
SDK	Kit de desenvolvimento de software
JVM	Java Virtual Machine
JSON	JavaScript Object Notation
IP	Internet Protocol
PCDs	Pessoas com deficiência

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Justificativa	14
<i>1.1.1</i>	<i>Acesso à informação</i>	<i>14</i>
<i>1.1.2</i>	<i>O uso das Tecnologias de Informação e Comunicação (TICs)</i>	<i>15</i>
<i>1.1.3</i>	<i>O uso de dispositivos moveis na Indústria 4.0</i>	<i>15</i>
1.2	Objetivos	15
<i>1.2.1</i>	<i>Objetivo geral</i>	<i>15</i>
<i>1.2.2</i>	<i>Objetivos específicos</i>	<i>16</i>
2	REVISÃO BIBLIOGRÁFICA	17
2.1	Controladores lógico programáveis (CLPs)	17
2.2	Protocolos de comunicação para o CLP siemens	17
2.3	Firestore	18
2.4	Python	18
2.5	Kotlin	19
2.6	Node.js	19
2.7	Código QR	19
2.8	JIDOKA	20
3	METODOLOGIA	21
3.1	Parametros de entrada	21
<i>3.1.1</i>	<i>Protocolo de comunicação</i>	<i>21</i>
<i>3.1.2</i>	<i>Biblioteca para o servidor local</i>	<i>21</i>
<i>3.1.3</i>	<i>Integração Servidor - Firestore</i>	<i>21</i>
<i>3.1.4</i>	<i>Integração Firestore - Dispositivo móvel</i>	<i>22</i>
<i>3.1.5</i>	<i>Método de reconhecimento das máquinas</i>	<i>22</i>
<i>3.1.6</i>	<i>Realidade aumentada</i>	<i>22</i>
3.2	Fluxo Geral	22
3.3	Fluxo específico do sistema	24
<i>3.3.1</i>	<i>Fluxo no dispositivo móvel</i>	<i>24</i>
<i>3.3.2</i>	<i>Fluxo no Servidor Local</i>	<i>26</i>

4	RESULTADOS	27
4.1	Desempenho e precisão	27
4.2	Desempenho do software	31
4.3	Âncoras	32
4.4	Modelos 3D	33
4.5	Desenvolvimento do software	36
4.5.1	<i>Programação do CLP utilizando o tia portal</i>	36
4.5.2	<i>Programação do servidor Python</i>	37
4.5.3	<i>Modelagem do banco de dados</i>	38
4.5.4	<i>Programação do App Android</i>	38
4.5.5	<i>Firebase</i>	39
4.5.6	<i>Encontrando o ponteiro no modelo 3D</i>	40
4.5.7	<i>Ajustando a escala</i>	41
4.5.8	<i>Executando a animação baseada nos dados do servidor</i>	42
4.6	Aplicações	42
4.7	Potencial retorno para a comunidade/empresa	43
4.7.1	<i>Ergonomia</i>	43
4.7.2	<i>Home office ou teletrabalho</i>	43
4.7.3	<i>Desburocratização da informação</i>	43
4.7.4	<i>Jidoka</i>	43
4.7.5	<i>Inclusão</i>	43
4.7.6	<i>Barateamento da mão de obra</i>	43
4.8	Viabilidade	43
4.9	Comparativo	44
5	CONCLUSÃO E TRABALHOS FUTUROS	45
5.1	Trabalhos Futuros	45
	REFERÊNCIAS	47

1 INTRODUÇÃO

“A Indústria 4.0 é um conceito que está ganhando relevância na indústria e na academia, especialmente pelas mudanças tecnológicas nos níveis de conectividade do sistema de produção” (CORDEIRO *et al.*, 2017). Com isso, estudos focados em aumentar a conectividade de sistemas dentro da indústria são sempre bem-vindos, já que uma das chaves de integração que classificam as diferentes características da indústria 4.0 é a utilização de sistemas conectados (KAGERMANN, 2013). Nesse sentido, a criação de um sistema supervisório móvel usando realidade aumentada pode trazer inúmeros benefícios para o setor industrial, além de estimular o uso de ferramentas mais tecnológicas no ambiente industrial.

Conceder acesso imediato, contínuo e de forma clara às informações das máquinas e processos de uma fábrica pode permitir uma resposta mais rápida a qualquer problema ou anomalia, reduzindo o tempo de reação e melhorando a eficiência geral. Para esse propósito, este trabalho tem como objetivo apresentar a criação de uma ferramenta, utilizando as mais recentes tecnologias para realizar a comunicação entre um CLP e um aplicativo móvel, apresentando essas informações na forma de realidade aumentada na câmera do celular. Com isso, espera-se contribuir para a aplicação prática das ferramentas mais tecnológicas na indústria, proporcionando melhoria contínua nos processos produtivos e de monitoramento.

1.1 Justificativa

Como já direcionado por Cordeiro *et al.* (2017), um dos grandes aspectos da indústria 4.0 é a conectividade, com isso, fazem-se presentes alguns conceitos, como o acesso à informação, o uso das tecnologias de informação e comunicação (TICs) e o uso de dispositivos móveis na indústria:

1.1.1 Acesso à informação

Em cenários ideais, sistemas já integrados e controlados por CLPs já possuem sistemas supervisórios. Porém, esses sistemas frequentemente se encontram centralizados em apenas um ambiente (normalmente uma sala de controle). Ao mesmo tempo, por uma questão de segurança e organização, somente quem tem acesso a esses ambientes são os operadores da sala de controle. Ou seja, ao mesmo tempo que essa centralização gera um sentimento de segurança, a mesma pode também gerar uma burocratização do acesso à informação, pois nem todos os indivíduos daquele ambiente industrial possuem acesso às informações tratadas e exibidas de forma tão clara quanto aqueles que têm acesso ao sistema supervisório.

Na Indústria 4.0, o mundo físico se funde com o virtual; baseados na Internet, produção, manutenção, fornecedores, compradores, distribuidores e até mesmo produtos e máquinas são integrados e conectados digitalmente, proporcionando uma cadeia de valor integrada como jamais visto antes (HERMANN M; PENTEK, 2018 apud SILVA, 2018).

Portanto, conceder acesso contínuo, imediato e de forma clara através de um dispositivo móvel é, também, desburocratizar de forma controlada o acesso à informação.

1.1.2 O uso das Tecnologias de Informação e Comunicação (TICs)

Em geral o uso das TICs no setor automobilístico, otimiza a operacionalização das inovações organizacionais e de gestão decorrentes da produção enxuta e do just-in-time, do controle de qualidade total e da engenharia simultânea. Também, a nível da engenharia de processos, o uso das TICs dá maior agilidade e qualidade na geração das informações permitindo desta maneira produzir dados que podem ser aproveitados por vários setores da empresa (DUMANS., 1998).

Nesse contexto, faz-se cada vez mais importante a criação de ferramentas mais tecnológicas para o setor industrial, já que quanto maior esse leque de ferramentas, maior é a versatilidade de suas aplicações, juntas ou separadas, criando assim uma estrutura mais customizável para cada tipo de contexto.

1.1.3 O uso de dispositivos moveis na Indústria 4.0

A integração entre Internet das Coisas e a Internet dos Serviços no processo de manufatura foi o que deu início à quarta revolução industrial (KAGERMANN, 2013 apud ALBERTIN, 2017). Essa tecnologia permite que “coisas” e “objetos” como sensores, atuadores e telefones celulares interajam entre si e cooperem com os componentes “inteligentes” ao seu redor para alcançar um objetivo comum (GIUSTO D.; ATZORI, 2010 apud ALBERTIN, 2017).

No mesmo sentido, a tecnologia mobile tende a propiciar um aumento da produtividade e maior disponibilidade dos ativos da empresa para a operação, em alinhamento com o paradigma da Indústria 4.0 (SILVA, 2018). Sendo assim, inovações que tragam cada vez mais direcionamento para o uso de soluções mobile contribuem para uma indústria mais conectada, caminhando cada vez mais para o ápice da Indústria 4.0.

1.2 Objetivos

1.2.1 Objetivo geral

Desenvolver uma solução que torne possível mostrar as informações de alguma variável de uma planta, como rotação, nível e temperatura na imagem da câmera de um dispositivo móvel através de medidores tridimensionais e realidade aumentada.

1.2.2 *Objetivos específicos*

- Construir um sistema capaz de consumir dados dos sensores de uma planta com um CLP Siemens;
- Tratar e transmitir essas informações da planta para o Firebase Realtime Database;
- Consumir as informações do Firebase Realtime Database no dispositivo móvel;
- Apresentar as informações das máquinas para o usuário;
- Identificar uma máquina pelo código QR (*Quick Response Code*) usando a câmera do celular;
- Gerar a visualização de instrumentos de medição animados ao lado do código QR apresentando as informações da máquina;

2 REVISÃO BIBLIOGRÁFICA

Nesse capítulo falaremos sobre os principais conceitos, técnicas e protocolos necessários para a realização desse trabalho, como dispositivos, protocolos de comunicação, Firebase, Python, *Kotlin* (linguagem moderna para desenvolvimento Android) e códigos QR, avaliando as suas importâncias e características.

2.1 Controladores lógico programáveis (CLPs)

Os controladores lógicos programáveis (CLPs) são hoje a tecnologia de controle de processos industriais mais amplamente utilizada. Um CLP é um tipo de computador industrial que pode ser programado para executar funções de controle (PETRUZELLA, 2014).

Sendo um ponto focal para qualquer planta industrial que precise de dispositivos mais robustos, são projetados para resistir a ambientes hostis. Usados em uma ampla gama de aplicações industriais, como: fabricação, embalagem e manuseio de materiais, onde o controle dos processos precisa dessa robustez para garantir qualidade e segurança em suas aplicações.

O seu uso, portanto, baseia-se principalmente na programação em *Ladder* (linguagem de programação gráfica baseada em diagramas de contato elétrico) focada no manuseio das entradas e saídas do CLP, podendo assim desempenhar diversas tarefas a partir dos valores recebidos dos sensores nas entradas e mandar sinais para os atuadores da planta, de forma contínua e completamente regulável.

Para programar e acessar os dados enviados para o CLP, são utilizados computadores comuns. E essa comunicação precisa ser feita de forma segura através de protocolos pré-definidos de acordo com o fabricante.

2.2 Protocolos de comunicação para o CLP siemens

Alguns dos protocolos de rede amplamente adotados pelos CLPs da Siemens são:

- PROFINET: Trata-se de um protocolo Ethernet industrial desenvolvido pela Siemens, largamente empregado em sistemas de controle e automação industrial. Reconhecido por sua capacidade de comunicação de alta velocidade, controle em tempo real e funções diagnósticas, o PROFINET é especialmente indicado para ambientes industriais que demandam alto desempenho.
- PROFIBUS: Este é um protocolo fieldbus bastante utilizado em sistemas de controle e automação industrial. Adequado para aplicações em automação de processos e fábricas, o PROFIBUS suporta uma ampla variedade de velocidades de comunicação e tipos de dados.

- **MODBUS:** Trata-se de um protocolo de comunicação industrial amplamente empregado, compatível tanto com comunicação serial quanto Ethernet. O MODBUS é frequentemente empregado para estabelecer a conexão entre CLPs e outros dispositivos, como sensores, atuadores e displays.
- **TCP/IP:** Este é o protocolo padrão de comunicação utilizado em redes da internet e é igualmente suportado pelos CLPs da Siemens. Possibilita a conexão com outros dispositivos por meio de redes Ethernet.
- **RFC 1006:** O protocolo S7 (RFC 1006) viabiliza a conexão de dispositivos Siemens com qualquer parceiro de comunicação, proporcionando acesso direto à memória. O RFC1006 foi concebido com o propósito de facilitar uma comunicação eficiente e confiável entre dispositivos, oferecendo suporte tanto para a transferência cíclica quanto acíclica de dados. A transferência de dados cíclica destina-se ao controle e monitoramento em tempo real de variáveis de processo, enquanto a transferência acíclica é utilizada para dados não críticos em relação ao tempo, como configuração e informações de diagnóstico (PEREIRA, 2023).

2.3 Firebase

O Firebase é uma plataforma de desenvolvimento de soluções de gerenciamento de dados em nuvem oferecida pela *Google*. Com um amplo leque de ferramentas focadas no desenvolvimento de aplicações web e mobile, é amplamente utilizado pelas big-techs para armazenar e gerenciar dados de maneira segura em suas soluções.

A grande vantagem do Firebase é a sua facilidade de integração com os mais diversos sistemas. Por exemplo: um Arduino programado em *C* pode atualizar dados direto no servidor e todos os dispositivos conectados com aquele servidor serão notificados, como um site web, um aplicativo Android e até mesmo outro Arduino, tudo isso usando requisições *HTTP* (*Hypertext Transfer Protocol*).

2.4 Python

Python é uma linguagem de programação de alto nível, orientada a objetos, de tipagem dinâmica. Conhecida por ser uma linguagem de programação com uma baixíssima curva de aprendizado, destaca-se por sua ampla gama de aplicações, já que pode ser facilmente integrada com praticamente qualquer ferramenta.

Considerando esses pontos fortes, permite que façamos a conexão com o CLP, podendo consumir as informações que o mesmo disponibilizar das máquinas que estiver conectado e, ainda, conectar diretamente com o Firebase, para disponibilizar os dados do CLP em um banco de dados criptografado para serem consumidos por outros dispositivos.

Com isso, a biblioteca Python-Snap7 surge como uma ferramenta robusta para estabelecer comunicação com controladores Siemens S7 PLCs. Sua escolha foi fundamentada na sua compatibilidade com os protocolos de comunicação amplamente utilizados na indústria, bem como na sua interface amigável para desenvolvedores Python. A decisão de adotar o Python-Snap7 baseou-se, ainda, em sua popularidade, facilidade de integração e suporte ativo da comunidade.

2.5 Kotlin

Kotlin é uma linguagem de programação de alto nível, orientada a objetos e estaticamente tipada, desenvolvida pela empresa JetBrains. Conhecida principalmente pelo seu uso no desenvolvimento de aplicações Android, tem como principal característica o fato de compilar usando a JVM (*Java Virtual Machine*).

A grande vantagem do *Kotlin* é a sua baixa curva de aprendizado e seu robusto kit de desenvolvimento de software, ou software development kit (SDK), tornando assim o desenvolvimento Android muito mais fácil e direto, sem precisar lidar com o complexo nível de detalhamento do Java. Hoje, a linguagem oficial de desenvolvimento nativo de aplicações é o *Kotlin*, sendo assim uma escolha mais lógica ao lidar com aplicações Android mais complexas, pois permite que os desenvolvedores foquem mais nas soluções do que nas características específicas da linguagem.

2.6 Node.js

Representa uma poderosa plataforma de desenvolvimento. Essa tecnologia, amplamente conhecida por sua eficiência em operações de entrada/saída, é orientada a eventos, o que a torna ideal para aplicações em tempo real. Sua sintaxe amigável e a capacidade de usar JavaScript em ambientes de servidor consolidam o Node.js como uma escolha popular entre os desenvolvedores.

Considerando essa versatilidade, surge o Node S7, uma biblioteca para Node.js que oferece uma solução eficaz para a comunicação com controladores Siemens S7 PLCs. A escolha do Node S7 é fundamentada na integração perfeita com a arquitetura assíncrona do Node.js, proporcionando uma comunicação eficiente e de baixa latência com os CLPs Siemens.

2.7 Código QR

O código QR é um símbolo de duas dimensões e foi criado em 1994 por Denso, uma das maiores companhias Toyota. Este foi aprovado como Padrão internacional ISO (ISO/IEC18004) em junho de 2000 e foi inicialmente criado para ser usado no controle de produção de partes automotivas, mas espalhou-se para diversas outras áreas nos últimos tempos (SOON, 2008).

A sua principal vantagem é poder ser utilizado como identificador único, permitindo maior confiabilidade e segurança em soluções que precisam de identificadores, como acontece com os pagamentos usando código QR, em que cada pagamento precisa de um identificador

único e insubstituível, então é gerado um código QR em tempo real para o mesmo, garantindo que não haja dois pagamentos usando o mesmo identificador.

2.8 JIDOKA

Autonomação ("jidoka") consiste em facultar ao operador ou à máquina a autonomia de parar o processamento sempre que for detectada qualquer anormalidade no processamento (GHINATO, 1995). Proporciona uma maior capacidade de detecção de erros.

3 METODOLOGIA

Nesse capítulo, utilizando a revisão bibliográfica como alicerce conceitual e teórico, decidiremos quais ferramentas ou tecnologias utilizar, e como estas serão aplicadas no sistema como um todo.

3.1 Parametros de entrada

Os parâmetros de entrada foram definidos conforme requisitos técnicos e funcionais do sistema, como: protocolo de comunicação, biblioteca para servidor local (baseada em desempenho e compatibilidade), integração servidor-Firebase e *Firebase-smartphone* (configuração de autenticação), método de reconhecimento de máquinas e realidade aumentada.

3.1.1 Protocolo de comunicação

Para a comunicação entre planta e servidor, o protocolo S7 (padrão de comunicação industrial da Siemens) foi criteriosamente selecionado como o protocolo de comunicação para nossa solução de integração entre o sistema proposto e os CLPs Siemens. Essa decisão é fundamentada em diversos fatores que convergem para garantir uma comunicação eficiente e confiável no contexto industrial, como sua compatibilidade amplamente reconhecida, suporte à comunicação cíclica, acesso direto à memória, ampla aceitação na indústria e compatibilidade com as necessidades específicas do nosso projeto de integração de dados industriais. Essa decisão visa assegurar uma comunicação robusta, confiável e eficiente entre o sistema proposto e os CLPs Siemens.

3.1.2 Biblioteca para o servidor local

Para efetuar a comunicação entre o sistema proposto e os Controladores Lógicos Programáveis (CLPs) Siemens, é essencial selecionar a tecnologia mais adequada, considerando a necessidade de uma integração eficiente e a compatibilidade com o protocolo de comunicação S7. Portanto, após realizar um comparativo entre duas opções mais relevantes, Python-snap-7 e Node-S7, optou-se pelo uso da primeira, devido à sua facilidade de integração com o Firebase, interface amigável para desenvolvedores Python, e suporte ativo da comunidade Python.

3.1.3 Integração Servidor - Firebase

A integração entre o servidor e o Firebase desempenha um papel crucial na transmissão e armazenamento eficientes dos dados provenientes do CLP. Para isso, precisamos garantir a segurança, confiabilidade e acessibilidade das informações. Esse processo inicia-se com a configuração do ambiente Firebase. Isso inclui a criação de um projeto no Firebase Console, a

obtenção das credenciais necessárias e a configuração de um banco de dados em tempo real para armazenar os dados provenientes dos CLPs.

O próximo passo é definir tokens de autenticação exclusivos para esse projeto no Firebase. Esse token será utilizado para inicializar o cliente do Firebase no servidor local para possibilitar o envio dos dados obtidos pelo controlador. Essa inicialização do cliente será feita utilizando a interface Python chamada *Firebase-admin*. Portanto, após a inicialização do cliente, o servidor realizará uma leitura cíclica dos dados do controlador e enviará sempre o valor mais atualizado para o Firebase.

3.1.4 Integração Firebase - Dispositivo móvel

Para o desenvolvimento da aplicação mobile será utilizada a linguagem de programação *Kotlin*, devido à sua baixa curva de aprendizado e seu robusto kit de desenvolvimento de software, além de ser a linguagem oficial de desenvolvimento nativo de aplicações Android, tornando-se absolutamente a opção mais recomendada. A sua integração com o Firebase se assemelha com a integração Python - Firebase, utilizando dos SDKs disponíveis para *Kotlin* para inicializar o cliente Firebase e consumir os dados fornecidos no banco de dados em tempo real.

3.1.5 Método de reconhecimento das máquinas

Para reconhecer as máquinas da planta, utilizaremos a tecnologia de código QR, por possibilitar a criação de identificadores únicos e visuais, que são parâmetros essenciais para esse sistema. Cada máquina será atribuída a um código QR exclusivo, representando sua identificação única. Essa atribuição será feita previamente e registrada no Firebase Realtime Database, associando cada código QR à máquina correspondente.

Então, será criado um sistema de leitura de código QR utilizando a câmera de um dispositivo Android, e a partir do código QR e seu identificador único, poderemos consumir as informações específicas daquela máquina no Firebase.

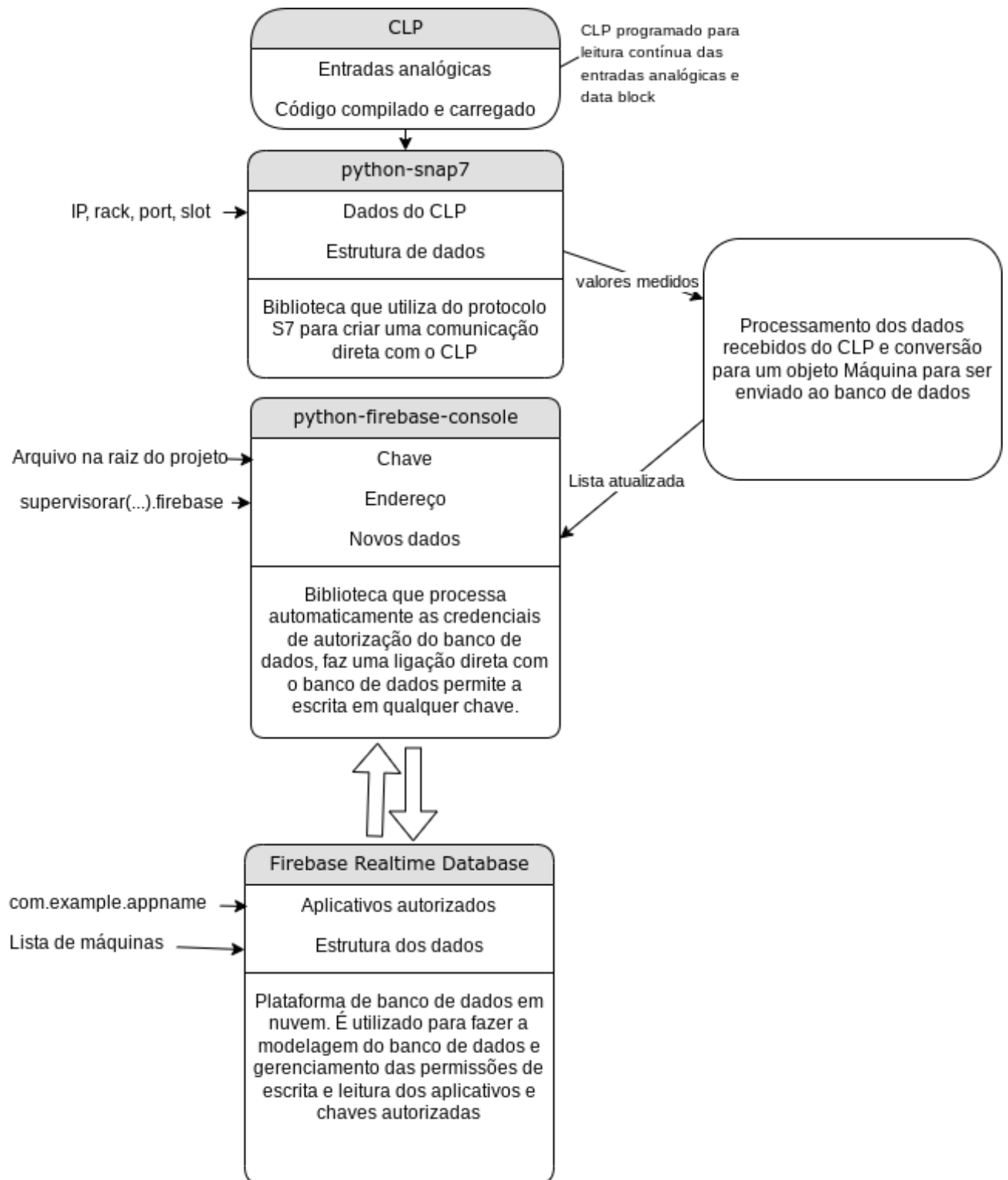
3.1.6 Realidade aumentada

Na implementação da realidade aumentada, códigos QR (dispostos nas máquinas) atuam como âncoras virtuais para exibir os modelos 3D no espaço físico.

3.2 Fluxo Geral

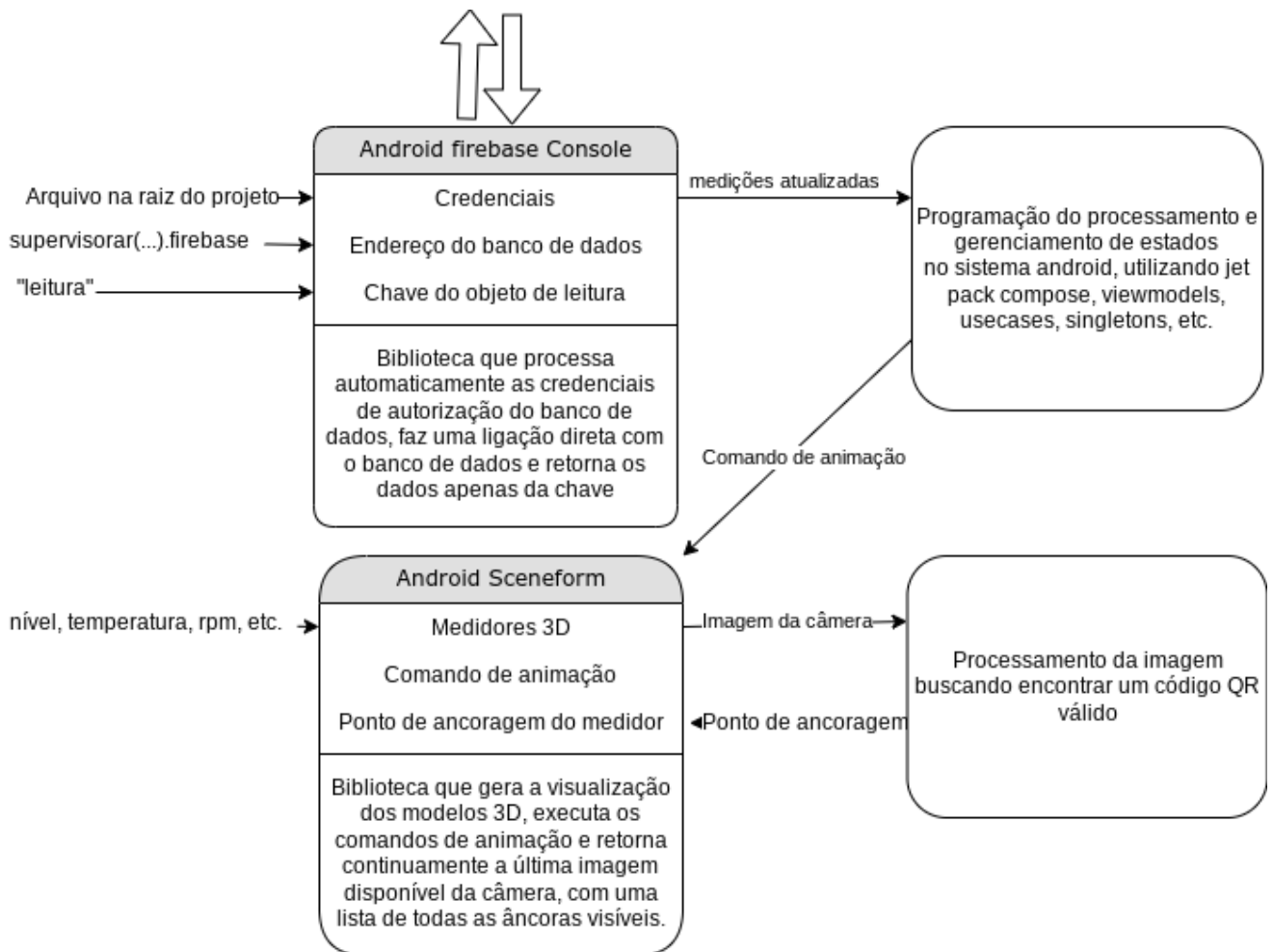
Portanto, esse será o fluxo geral do sistema:

Figura 1 – Diagrama de fluxo geral pt. 1.



Fonte: Elaborado pelo autor, 2025.

Figura 2 – Diagrama de fluxo geral pt. 2.



Fonte: Elaborado pelo autor, 2025.

3.3 Fluxo específico do sistema

O fluxo operacional do sistema foi segmentado em etapas críticas para garantir rastreabilidade e alinhamento com os requisitos. Cada subfluxo aborda aspectos técnicos específicos, conforme descrito a seguir.

3.3.1 Fluxo no dispositivo móvel

A programação do dispositivo móvel segue um paradigma de programação *frontend* chamado programação orientada a eventos. Portanto, todo o sistema no dispositivo é programado esperando eventos pré-descritos para que uma função seja executada.

Primeiro, é feita a inicialização e o processo de login. Acontece no momento em que o usuário abre o aplicativo pela primeira vez, e é responsável por uma série de validações iniciais para que o aplicativo funcione corretamente. Estes são:

- Apresentação da tela de Login e identificação do usuário;

- Representa uma tela estática com dois campos de entrada de texto para login e senha, os quais serão usados para identificar à quais máquinas esse usuário terá acesso nos próximos passos.
- Inicialização do cliente do Firebase;
 - Enquanto a tela carrega e o usuário interage com a tela, inicializamos o cliente do Firebase que será usado para validar os dados de login e fará uma chamada *HTTP* para pegar os identificadores alfanuméricos que representam cada máquina no banco de dados não relacional do Firebase realtime.

Depois, a partir do momento em que o usuário digita um usuário e senha válidos, é iniciado o processo em que o dispositivo começa a captar e processar a imagem com foco em encontrar um código QR para buscar suas informações no servidor.

- Identificação da imagem:
 - A câmera do dispositivo é aberta dentro do aplicativo e assincronamente fazemos a análise quadro a quadro da imagem fornecida. Em cada quadro verificamos primeiro se encontramos algum código QR e, se sim, marcamos aquele ponto da imagem como uma âncora para os nossos dispositivos de medição virtuais. Após encontrar o código QR, iniciamos o processo de leitura do mesmo, e seu conteúdo precisa ser exatamente um texto com o valor do identificador de uma máquina no servidor, caso contrário, o sistema apresentará um erro e irá iniciar a busca de outro código QR nas imagens da câmera.
- Conexão com as leituras captadas:
 - A partir do momento em que temos um identificador válido, podemos usar o cliente do Firebase realtime para pegar as informações de medição para aquele identificador. O servidor retorna um valor no formato JSON contendo as seguintes propriedades:
 - * Código da máquina, seguindo o padrão <nome><número>;
 - * Valor medido pelo dispositivo de medição naquele instante;
 - * Range do dispositivo de medição;
 - * Span do dispositivo de medição;

Então, a partir do momento em que um código QR válido é encontrado, é iniciado o processo de solicitar ao Firebase *client* as informações de uma máquina em tempo real, processar essa informação e exibir o instrumento de medição correspondente na interface do usuário. A função do cliente do Firebase realtime se chama "*getValue*" e recebe o parâmetro chamado *key* (chave). A partir do momento em que solicitamos os dados, o Firebase realtime nos retornará

uma string no formato JSON, então, convertemos os dados JSON em uma instância da classe *MachineData*. Após essa conversão, a instância é utilizada para exibir as informações na interface do usuário.

3.3.2 Fluxo no Servidor Local

A programação do servidor local será feita utilizando a biblioteca *Python-snap-7*, que permite a conexão com o CLP, e o *Firebase-python-sdk* para se conectar e mandar dados para o servidor em nuvem.

Primeiro, inicializamos o cliente do Firebase, fornecendo todas as informações necessárias para uma autenticação correta, como credenciais da licença e URL do banco de dados, para garantir uma conexão segura e constante.

Depois, fornecendo os dados de conexão com o CLP, como endereço de IP, rack, slot e port, podemos iniciar a leitura dos dados do data block interno do CLP. Então, em um loop while, fazemos a leitura dos dados do CLP e enviamos imediatamente para o banco de dados.

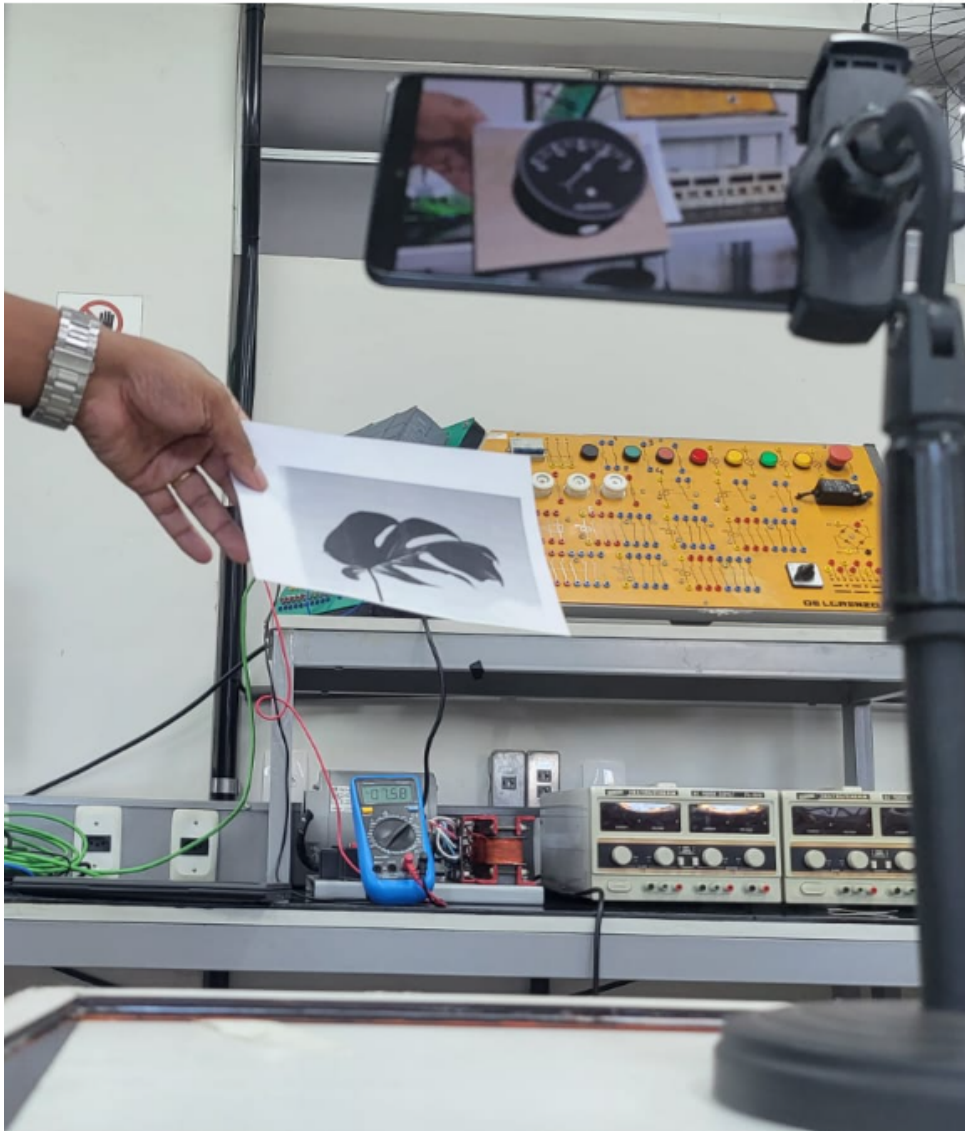
4 RESULTADOS

Nesse capítulo, utilizando a metodologia prevista, detalharemos como foi o andamento do desenvolvimento, bem como resultados práticos e comparativos.

4.1 Desempenho e precisão

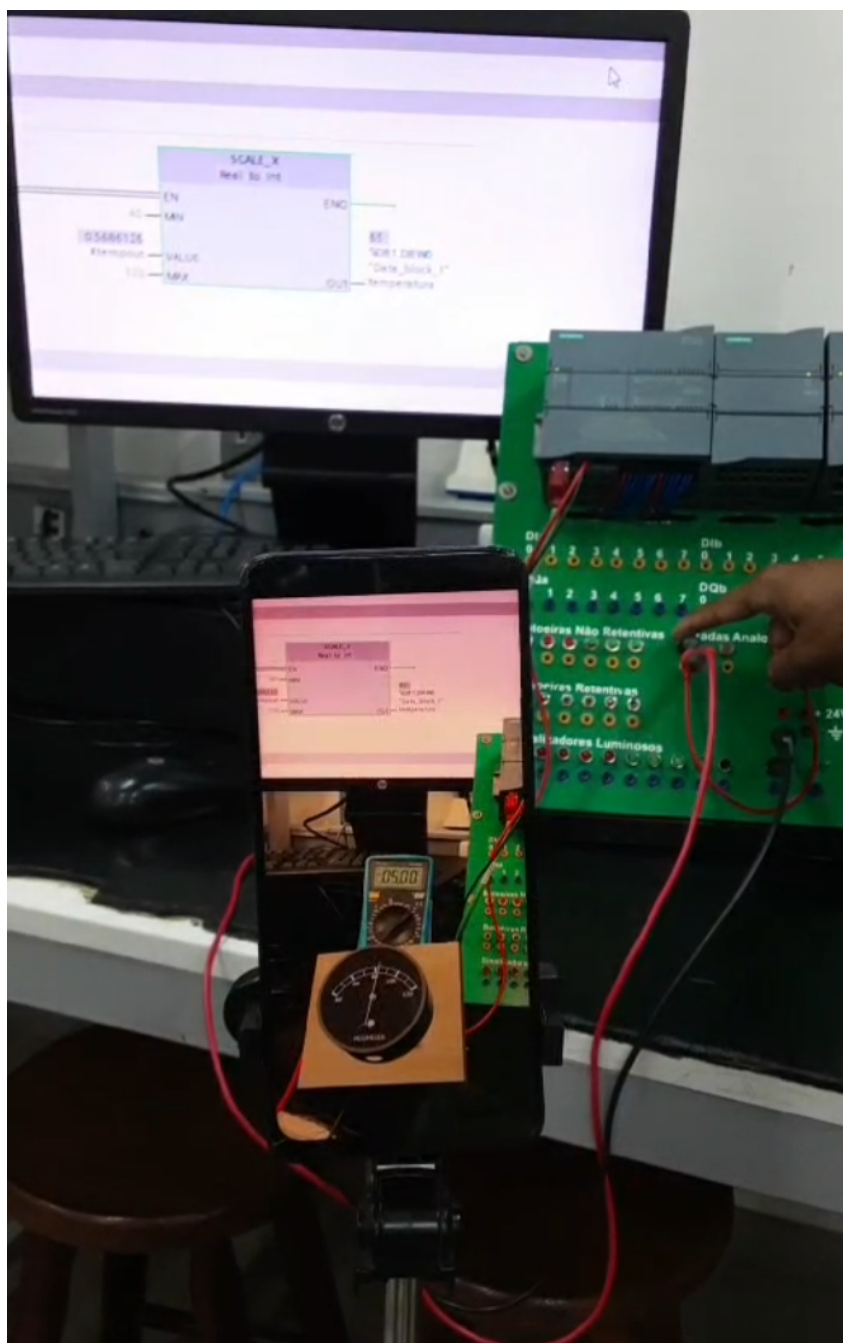
O desempenho do sistema mostrou-se satisfatório: os dados capturados no CLP são processados e exibidos ao usuário em menos de 1 segundo, garantindo clareza e intuitividade. A precisão é reforçada por medidores virtuais 3D, que, diferentemente dos analógicos convencionais, não possuem restrições mecânicas (ex.: engrenagens) e permitem ajustes contínuos de escala, superando a precisão de dispositivos físicos encontrados no mercado.

Figura 3 – Capturando a âncora e exibindo o medidor em tempo real.



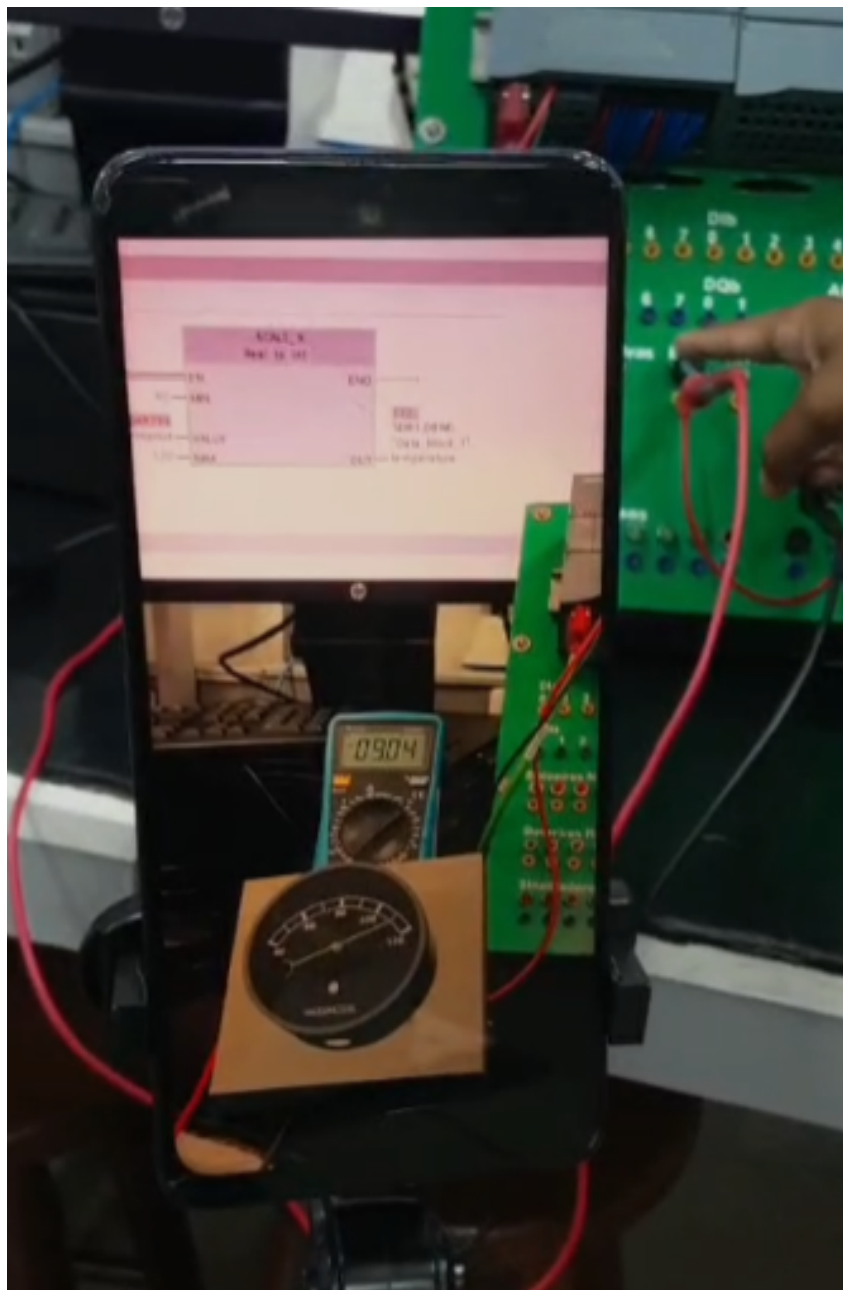
Fonte: Elaborado pelo autor, 2025.

Figura 4 – Ponto de vista do operador: valor em aprox. 50% ou 80°C



Fonte: Elaborado pelo autor, 2025.

Figura 5 – Ponto de vista do operador: valor em aprox. 90% ou 112°C



Fonte: Elaborado pelo autor, 2025.

Figura 6 – Ponto de vista da câmera do app: valor em aprox. 90% ou 112°C

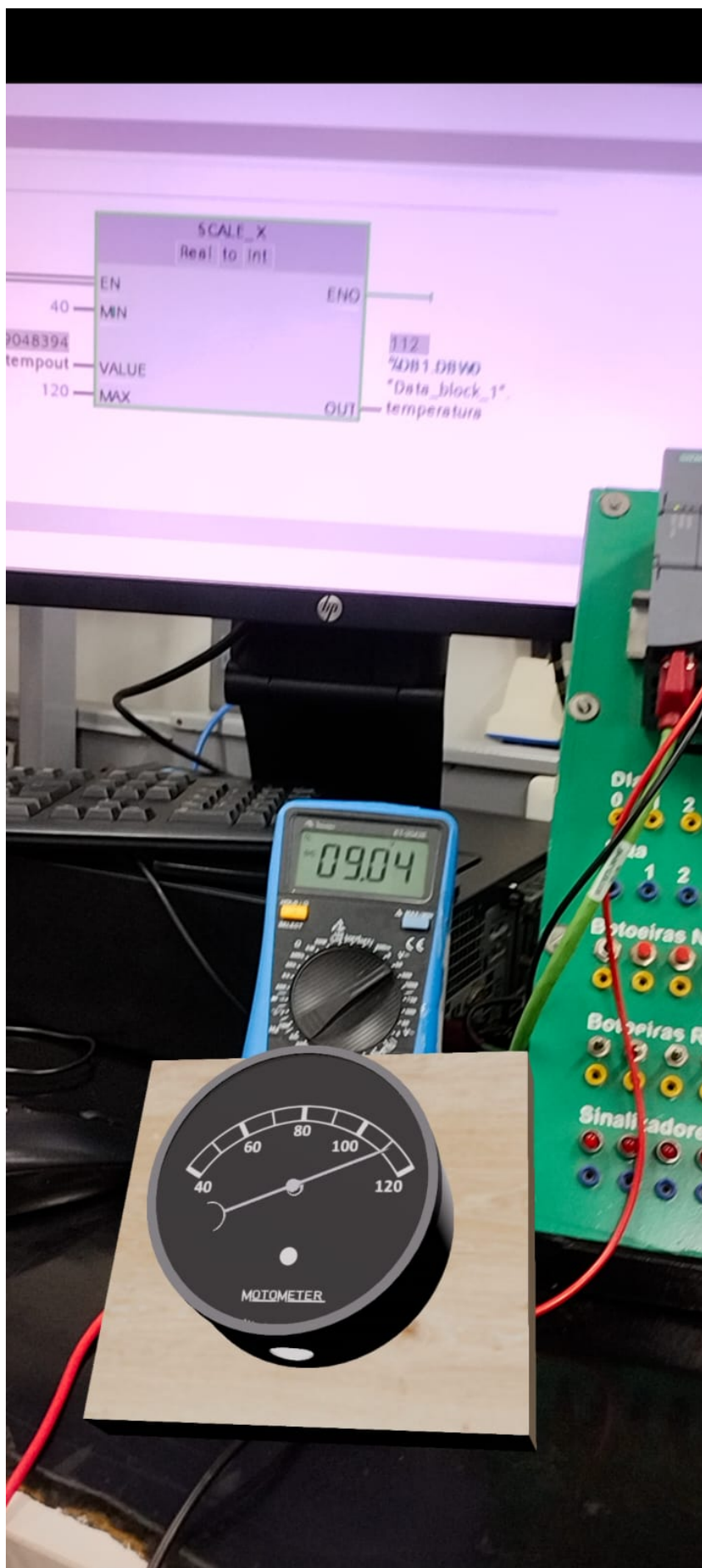
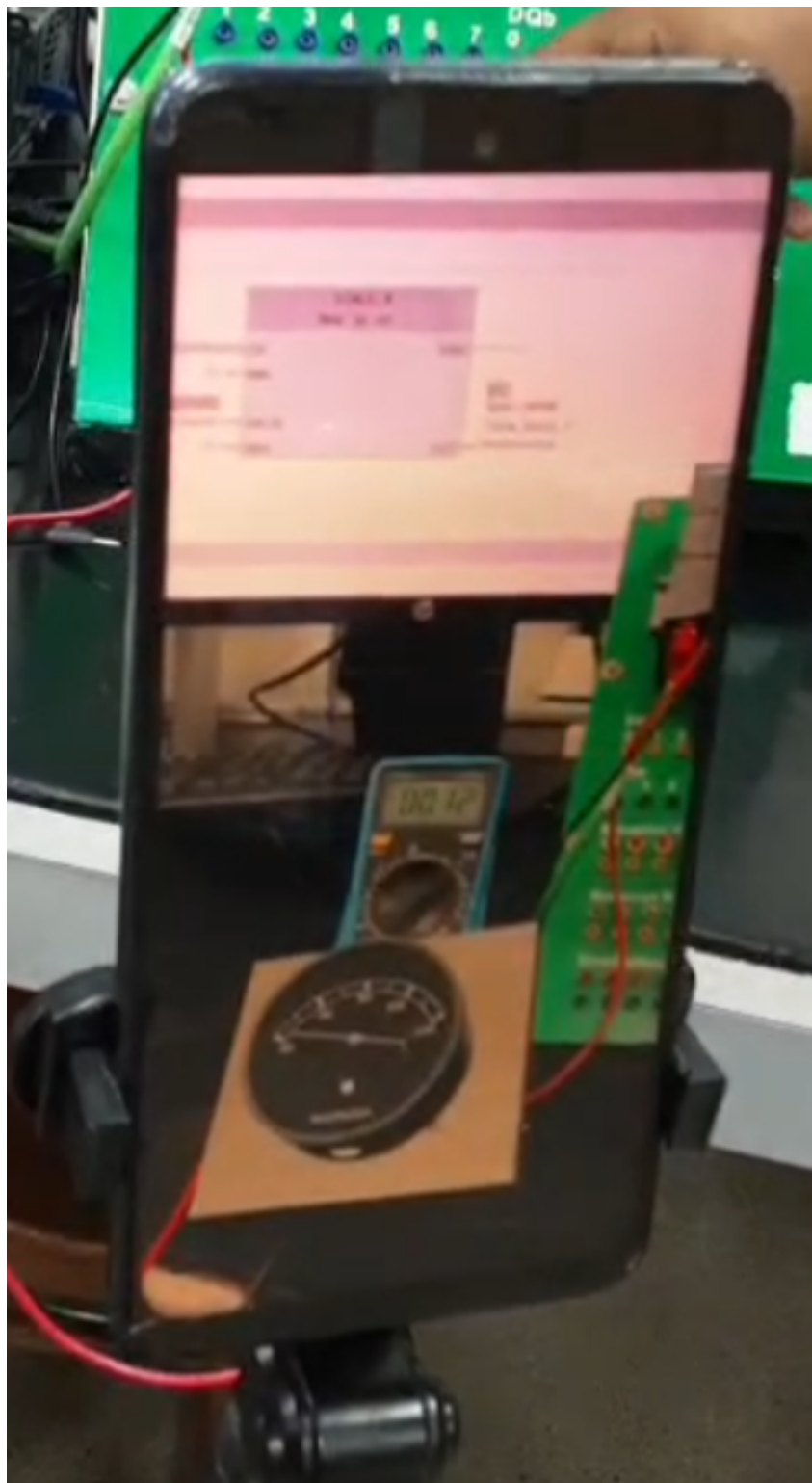


Figura 7 – Ponto de vista do operador: valor em 0% ou 40°C



Fonte: Elaborado pelo autor, 2025.

4.2 Desempenho do software

O desempenho do software de realidade aumentada desenvolvido mostrou-se satisfatório, mas com espaço para melhorias, pois, em condições de uso contínuo e prolongado, observou-

se um descarregamento consideravelmente rápido. Além disso, o smartphone utilizado (Moto G62) apresentou pequenos travamentos em situações de uso contínuo, prolongado na exibição simultânea de múltiplos modelos 3D.

4.3 Âncoras

Durante o desenvolvimento, diversos métodos de ancoragem foram utilizados e comparados para um traqueamento contínuo mais preciso do medidor 3D. Dentre eles, o que mais se destacou foi a utilização de imagens naturais em escala de cinza, devido ao seu alto contraste e maior riqueza de detalhes e profundidade.

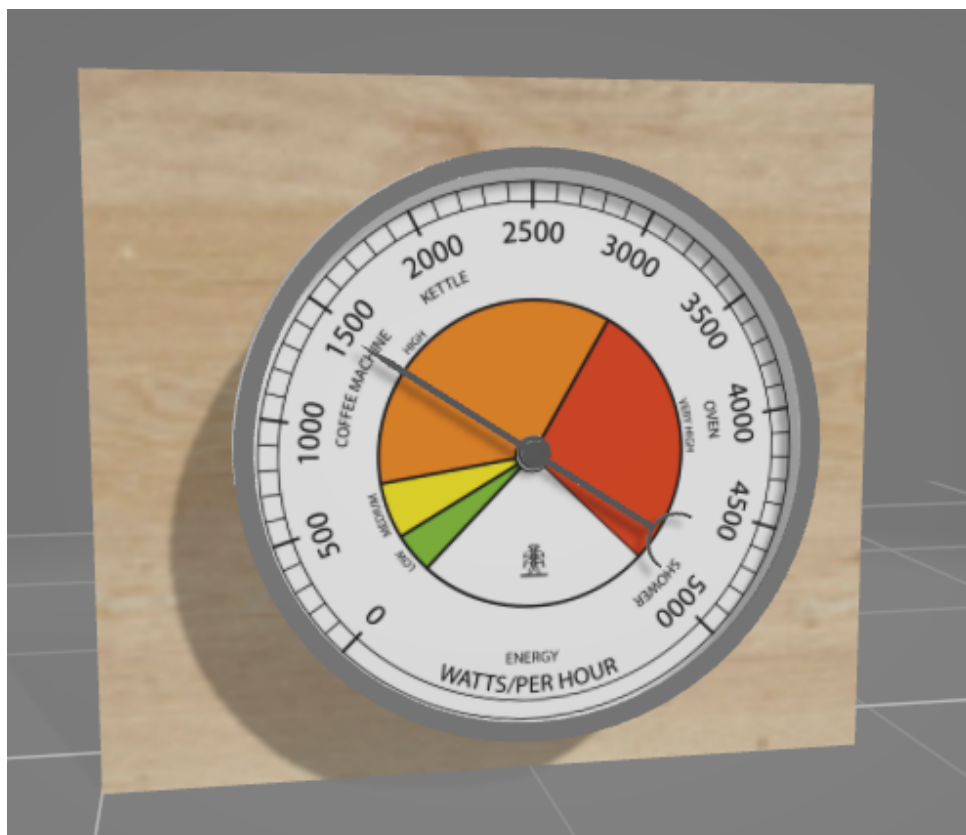
- Leitura de códigos QR em tempo real;
 - A leitura de códigos QR em tempo real apresentou-se ineficaz, pois exigia um processamento intensivo - executando a análise de imagem quadro a quadro, 30 a 60 quadros por segundo. Além disso, esse método mostrou-se ineficiente para o traqueamento contínuo do medidor 3D, posicionando o medidor virtual no local do código QR, mas, restringindo seu reposicionamento posterior.
- Cadastro de códigos QR como imagens aumentadas;
 - O método demonstrou uma melhoria em relação ao anterior, mas, devido a natureza geométrica muito similar e confundível dos códigos QR, o sistema apresentava falhas de identificação quando múltiplos medidores virtuais estavam visíveis ao mesmo tempo. Entretanto, uma vez identificado, o medidor passava a seguir - com dificuldade - a imagem do código QR no ar.
- Cadastro de arUco codes como imagens aumentadas;
 - O método apresentou melhoria em relação à natureza geométrica muito similar e confundível dos códigos QR. Entretanto, devido à baixa qualidade das imagens, o sistema demorava para encontrar e identificar as imagens aumentadas como âncoras. Comprometendo também o traqueamento contínuo.
- Cadastro de imagens naturais em escala de cinza como imagens aumentadas;
 - O cadastro de imagens naturais em escala de cinza como imagens aumentadas comprovou ser o método mais eficiente. Todos os aspectos desejados foram atendidos, proporcionando uma melhoria significativa no desempenho geral. Isso se deve ao fato de que as imagens naturais possuem maior riqueza de detalhes fotográficos - como contraste, profundidade e elementos únicos. Além disso convertê-las para escala de cinza foi um ponto chave para acelerar o processamento e facilitar a detecção. O

uso da escala de cinza para melhor detecção em softwares de processamento de imagens é uma técnica conhecida e consolidada, e o kit de desenvolvimento de software Sceneform é otimizado para trabalhar com imagens naturais.

4.4 Modelos 3D

A modelagem dos medidores 3D se mostrou um desafio, já que exigia conhecimento sólido e prática em softwares de modelagem 3D, como *Blender*. Com isso, para esse trabalho, optou-se por um modelo 3D de acesso livre na internet, em sites como Sketchfab.

Figura 8 – Medidor ideal encontrado: Energymeter



Fonte: Elaborado pelo autor, 2025.

Como visto na figura 2, o modelo Energymeter apresentava a maioria dos requisitos técnicos para este trabalho, como resolução adequada, representação fiel de um display analógico e um ponteiro animado, evidenciando a sua independência como objeto e com eixo de giro no ponto correto. Porém, esse modelo não está de acordo com as variáveis que desejamos medir (temperatura, velocidade e nível). Portanto, foi necessário um estudo básico sobre modelagem 3D para entender como mudar a escala do medidor. Com as devidas aplicações, obtivemos os seguintes medidores:

Figura 9 – Medidor adaptado: Temperatura



Fonte: Elaborado pelo autor, 2025.

Figura 10 – Medidor adaptado: Nível



Fonte: Elaborado pelo autor, 2025.

Figura 11 – Medidor adaptado: RPM



Fonte: Elaborado pelo autor, 2025.

Figura 12 – Medidor adaptado: Voltímetro



Fonte: Elaborado pelo autor, 2025.

4.5 Desenvolvimento do software

O desenvolvimento do software de realidade aumentada, integrado a um banco de dados com permissões de leitura e escrita para o servidor local e o aplicativo Android, fundamentou-se nos seguintes princípios:

- Utilização do SDK Sceneform do *Google*;
- Utilização do SDK Firebase Realtime da *Google*;
- Implementação do SDK Firebase Console por meio da plataforma Firebase;
- Utilização do SDK python-snap7 do autor Gijs Molenaar;
- Desenvolvimento de um banco de dados não relacional (Firebase Realtime Database) na interface Firebase Console;
- Utilização de uma licença gratuita limitada de utilização do Firebase Realtime;
- Utilização do Software Tia Portal;

Com os recursos mencionados e a aplicação de conceitos de programação orientada a soluções, o sistema foi integralmente implementado.

4.5.1 Programação do CLP utilizando o tia portal

Figura 13 – Programação do CLP: leitura de um sensor de temperatura



Fonte: Elaborado pelo autor, 2025.

Para a programação do CLP utilizamos dois blocos funcionais: NORM-X e SCALE-X. NORM-X foi utilizado para ler o valor de entrada físico do CLP e configurá-lo em uma escala real (0 a 1 float). Note que a entrada está no endereço de memória IW64, o que significa que é um endereço analógico de entrada, do tipo Word, que estará armazenado no byte 64 do barramento de memória do CLP. Embora contra-intuitivo, esse método nativo é um padrão utilizado para que se obtenha valores mais precisos em processos de leitura analógica, devido aos 16 bits disponíveis para armazenar um valor de leitura que normalmente só iria de 0 a 10V. Depois, utilizando o

bloco SCALE-X, podemos ajustar o valor real para a escala verdadeira do nosso sensor - no caso acima, é considerando um medidor que vai de 40 a 120 graus Celsius.

4.5.2 Programação do servidor Python

Figura 14 – Programação do servidor: Código Python

```
cred = credentials.Certificate("../Firebase/firebasekey.json")
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://(...).firebaseio.com/'
})

client = snap7.client.Client()
client.connect("192.168.0.16", 0, 1, 102)
client.get_connected()

while (True):
    data = client.db_read(1, 0, 4)
    temperatura = get_int(data, 0)

    db.reference('/leitura/0').set({
        "id": "maquina1",
        "type": "temperature",
        "value": temperatura
    })
```

Fonte: Elaborado pelo autor, 2025.

No desenvolvimento do servidor em Python, foram empregadas as bibliotecas `python-snap7` e `firebase-admin`.

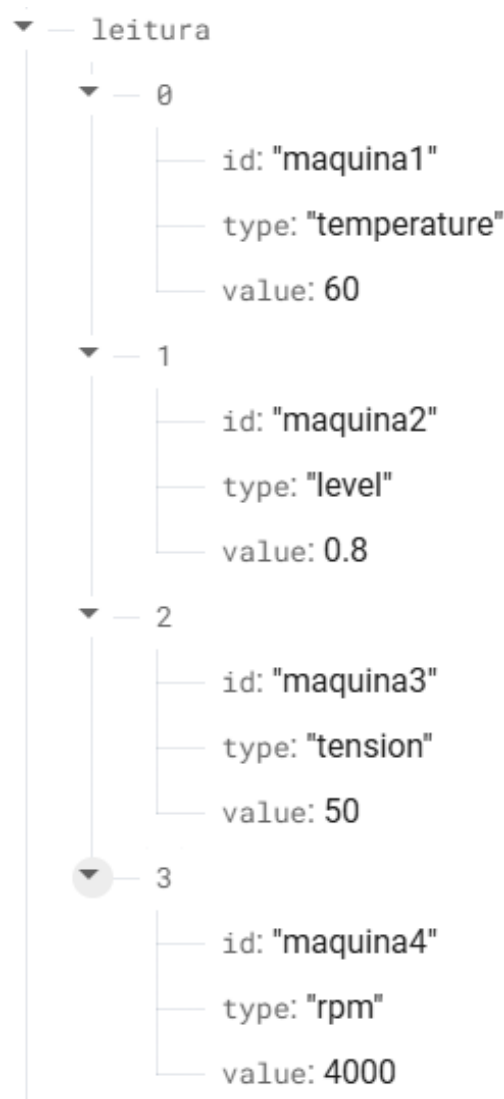
Para estabelecer uma conexão com o banco de dados, utilizamos as quatro primeiras linhas, onde foram fornecidos os dados e certificados necessários para autenticação segura no Firebase.

Posteriormente, a conexão com o CLP foi estabelecida via protocolo TCP/IP, mediante inserção de parâmetros como endereço IP, rack, slot e porta.

Depois, foi implementado um loop contínuo de leitura dos valores armazenados no bloco de dados 1 do CLP. Para isso, passamos as informações necessárias para obter os dados corretamente, como byte inicial para leitura e quantidade de bytes lidos após o byte inicial. Além disso, a função `get_int` de tipagem estática, assegura a conversão adequada dos valores para envio ao Firebase Realtime Database. Por fim, podemos utilizar a instância `"db"` para enviar os valores medidos para seus devidos objetos no banco de dados.

4.5.3 Modelagem do banco de dados

Figura 15 – Firebase Realtime Database



Fonte: Elaborado pelo autor, 2025.

A modelagem do banco de dados foi projetada para otimizar a recuperação de dados no aplicativo. Com isso, a estrutura utilizada foi em formato de lista de máquinas, onde cada máquina tem o seu próprio id, tipo e valor. Com isso em mãos, podemos deixar a implementação do código no app o mais genérico possível, se baseando apenas em um objeto. O que amplia a escalabilidade do projeto final.

4.5.4 Programação do App Android

Esta seção descreve os aspectos fundamentais da implementação em *Kotlin*.

4.5.5 Firebase

Figura 16 – Escutando mudanças no banco de dados

```
init {
    myRef.addValueEventListener(object : ValueEventListener {
        override fun onDataChange(dataSnapshot: DataSnapshot) {
            kotlin.runCatching {
                dataSnapshot.getValue<List<MachineInfo>>()?.let {
                    currentData.value = it
                }
            }.onFailure { error ->
                Log.d("FIREBASE", error.toString())
            }
        }

        override fun onCancelled(error: DatabaseError) {
            Log.d("FIREBASE", error.message)
        }
    })
}
```

Fonte: Elaborado pelo autor, 2025.

Nesse código, note o código contido dentro da função `onDataChange`. Veja que empregou-se `Kotlin.runCatching`, análogo ao bloco `try/catch` do Java, para interceptar exceções e prevenir interrupções no aplicativo. Em caso de falha, esse erro é apenas logado no console. Além disso, note que utilizamos a função `getValue<List<MachineInfo>>()`, que é capaz de recuperar o dado JSON que foi modelado no banco de dados e já fazer a conversão para uma lista de máquinas. Então, utilizamos o operador Elvis (`?:`), que verifica a existência do dado de forma concisa. Ou seja, o retorno dessa função é não nulo? Se a resposta para essa pergunta for sim, então o bloco mais interno é executado, e o dado atual exibido nas máquinas é atualizado.

4.5.6 Encontrando o ponteiro no modelo 3D

Figura 17 – Encontrando o ponteiro

```
private fun Node.findChildRecursive(name: String): Node? {
    if (this.name == name) return this

    for (child in this.childNodes) {
        val result = child.findChildRecursive(name)
        if (result != null) return result
    }

    return null
}
```

Fonte: Elaborado pelo autor, 2025.

A hierarquia do modelo 3D é representada por um nó raiz, que pode conter subnós, os quais, por sua vez, também admitem subdivisões. Com isso, essa função recursiva procura encontrar qualquer nó dentro de toda essa hierarquia, contendo apenas a informação do seu nome dentro do modelo 3D. Essa função é utilizada para encontrar o nó que representa o ponteiro dentro do modelo 3D.

4.5.7 Ajustando a escala

Figura 18 – Executando a transformação linear do valor para angulo de rotação do ponteiro

```
override fun calculateAngle(info: Medidores3d, objectiveValue: Float): Float {
    val valor = if (objectiveValue > info.valueMax) {
        info.valueMax
    } else if (objectiveValue < info.valueMin) {
        info.valueMin
    } else objectiveValue

    // 1. Normaliza o valor para um número entre 0 e 1.
    val valorNormalizado = (valor - info.valueMin) / (info.valueMax - info.valueMin)

    // 2. Calcula a amplitude (diferença) do intervalo de ângulos.
    val amplitudeAngulo = info.angleMax - info.angleMin

    // 3. Escala o valor normalizado para o intervalo de ângulos.
    val anguloEscalado = valorNormalizado * amplitudeAngulo

    // 4. Ajusta o ângulo escalado para iniciar a partir do ângulo mínimo.
    val anguloFinal = anguloEscalado + info.angleMin

    return anguloFinal
}
```

Fonte: Elaborado pelo autor, 2025.

A função aplica uma transformação linear para converter o valor do servidor no ângulo de rotação do ponteiro.

4.5.8 Executando a animação baseada nos dados do servidor

Figura 19 – Executando a transformação linear do valor para angulo de rotação do ponteiro

```
LaunchedEffect(serverData) {
    presentationData = presentationData.map {
        it.copy(valor = serverData.find { data ->
            data.id == it.id }?.value ?: 0f
        )
    }.onEach {
        it.ponteiro?.rotation = Quaternion.fromAxisAngle(
            Float3(0f, 1f, 0f),
            viewModel.calculateAngle(it.info, it.valor)
        ).toRotation(RotationsOrder.YXZ)
    }
}
```

Fonte: Elaborado pelo autor, 2025.

O algoritmo implementado atualiza a rotação dos ponteiros conforme os dados recebidos do servidor. Ou seja, toda vez que o sistema recebe novos dados do servidor, a função *LaunchedEffect* executa o código interno. Este, por sua vez, recupera a lista de medidores que já estão exibidos e procura qual daqueles possui um identificador que corresponde ao identificador que veio da lista do servidor.

Ao encontrar, ele atualiza os dados desse medidor para ficar igual ao dado do servidor. Se não encontrar, ele usa o operador Elvis (?:) para definir um valor padrão (0). Depois de fazer esse mapeamento, a função *onEach* é chamada para executar a rotação de todos os ponteiros para os valores atualizados. Veja que, nesse ponto, utilizamos a função *calculateAngle* mencionada anteriormente.

4.6 Aplicações

Esse sistema é flexível e aplicável em múltiplos contextos, como em ambientes industriais onde máquinas não dispõem de interfaces locais para monitoramento, dependendo exclusivamente de sistemas supervisórios remotos - cuja distância física dificulta análises pontuais ou contínuas. Além disso, há casos em que a planta até possui os medidores físicos locais, mas esses se encontram sujos, quebrados ou simplesmente inacessíveis para o operador.

Nesses casos, ter acesso a esse aplicativo em mãos permitiria acesso contínuo e fiel aos valores de interesse de maneira intuitiva e de fácil acesso. Tudo que precisaria seria um cartão contendo a imagem cadastrada para a máquina ou sensor em questão.

4.7 Potencial retorno para a comunidade/empresa

4.7.1 Ergonomia

O sistema promove ergonomia ao permitir que o operador posicione medidores virtuais no espaço 3D conforme suas necessidades físicas.

4.7.2 Home office ou teletrabalho

O sistema viabiliza que operadores de supervisão atuem remotamente com o mesmo nível de precisão de um monitoramento presencial, desde que implementado adequadamente.

4.7.3 Desburocratização da informação

Esse sistema permite o acesso aos dados do CLP a quaisquer colaboradores com acesso ao aplicativo e ao cartão âncora para aquela máquina, garantindo agilidade na manutenção e até implementação do Jidoka.

4.7.4 Jidoka

O Jidoka, princípio da filosofia Lean que integra automação e intervenção humana, delega a todos os colaboradores a responsabilidade pela qualidade em cada etapa do processo. Ao garantir acesso universal aos dados do sistema supervisorio, o monitoramento de parâmetros críticos deixa de ser centralizado, permitindo que a identificação de anomalias seja compartilhada e resultando em melhoria contínua na qualidade do produto.

4.7.5 Inclusão

Esse sistema pode atuar também como um facilitador para habilitar os espaços para condição de trabalho de PCDs (Pessoas com Deficiência), já que eles podem dispor os medidores da maneira que lhes for mais confortável e ergonômica no ar. Pessoas com deficiência visual, por exemplo, podem ampliar medidores virtuais para escalas adaptativas (ex.: proporções compatíveis com a sua necessidade visual), garantindo acessibilidade na interpretação dos dados.

4.7.6 Barateamento da mão de obra

A simplicidade do sistema pode reduzir a demanda por mão de obra especializada, diminuindo custos operacionais.

4.8 Viabilidade

Alguns passos são fundamentais para a viabilidade e bom funcionamento desse sistema:

- Requer capacitação técnica;
 - Embora aparentemente seja uma interface intuitiva para usuários leigos, pois se utiliza de uma ferramenta que todos usamos no nosso dia a dia - um smartphone e sua câmera - esse sistema requer tempo e prática para ser utilizado, já que o conceito de ter um medidor analógico "no ar" pode causar certa confusão e estranheza por parte do operador.
- Instalação/Setup;
 - Esse sistema não necessita de ponto de energia, cabos ou quaisquer outras ligações físicas que um sistema supervisório comum precisaria. Ao invés disso, basta cadastrar as imagens âncora e carregar os modelos 3D associados a ela. Além disso, o nome da imagem âncora cadastrado deve ser igual ao id da máquina cadastrado no banco de dados, para que a identificação ocorra corretamente.
- Exige modelagem 3D;
 - Para um bom funcionamento desse sistema é preciso bons modelos 3D. com boa resolução, escalas bem ajustadas e nós bem definidos. Felizmente, esse é um serviço relativamente barato para se contratar na internet. Em sites como SketchFab.com é possível encontrar modelos de medidores prontos na faixa de 6 a 15 dólares. Também é possível contatar os autores originais dos modelos e solicitar um medidor personalizado, focado exatamente nas necessidades do seu projeto.
- Exige um smartphone com bom processamento;
 - Embora exija baixa manutenção e simples instalação, esse sistema necessita de um smartphone com um bom poder de processamento, para que não ocorram travamentos durante o uso. Os testes utilizaram um Motorola Moto G62, que demonstrou desempenho satisfatório e com poucos travamentos.

4.9 Comparativo

Embora inovador, o sistema não substitui soluções supervisórias tradicionais, mas pode atuar como alternativa complementar para pequenas plantas ou redundância em sistemas existentes. É preciso ressaltar que a exibição de valores de sensores é apenas uma das funções de um sistema supervisório robusto.

5 CONCLUSÃO E TRABALHOS FUTUROS

Os resultados demonstram a viabilidade de implementar um sistema de realidade aumentada em *smartphones* para monitoramento de dados de CLPs. Apesar da complexidade técnica, a solução possui integração simplificada, adaptável a plantas existentes mediante execução do servidor *Python* em um computador conectado ao CLP e configuração dos endereços de memórias correspondentes. A partir disso, os dados já estarão disponíveis no servidor e no app, sendo necessário apenas editar as imagens âncora e os modelos 3D.

códigos QR demonstraram limitada eficácia para realidade aumentada devido à semelhança entre seus padrões geométricos, que geram ambiguidade no reconhecimento de imagem. Sendo assim, o uso de imagens naturais em escala de cinza foi mais efetivo em todos os aspectos, facilitando implementações futuras.

Embora os testes realizados tenham sido simplificados (ex.: potenciômetro de 0-10V), a metodologia assegura compatibilidade com sensores analógicos convencionais, dada a padronização de CLPs para essa faixa de tensão. Sendo assim, os resultados não seriam diferentes se fosse conectado na entrada analógica um sensor comum do mercado.

Embora ofereça transmissão ágil de dados, o sistema não atende a requisitos de tempo real, essenciais em ambientes industriais com controle rigoroso; nesses cenários, torna-se imprescindível adotar uma abordagem focada na minimização da latência e na otimização dos tempos de resposta. Para mitigar a latência, propõe-se substituir o servidor online por um *broker* MQTT local ou comunicação serial direta via USB.

Por fim, a solução apresenta viabilidade para pequenas empresas com restrições orçamentárias, além de potencial como redundância em sistemas supervisórios consolidados, visando agilizar o acesso à informação e possibilitar a implementação do Jidoka.

O trabalho alinha-se aos pilares da indústria 4.0 e IoT, ao integrar tecnologias emergentes (realidade aumentada e conectividade) a processos industriais. Se popularizado, esse sistema pode abrir espaço para a utilização de *smartphones* e realidade aumentada na indústria, como traz benefícios no âmbito de análise de dados, desempenho e estratégias preventivas. Além de abrir espaço para mais pesquisas focadas em realidade aumentada na indústria.

5.1 Trabalhos Futuros

Para a continuação desse trabalho, seria interessante evoluir o sistema para a comunicação bidirecional, utilizando processamento de imagem para identificar gestos manuais (como o movimento de uma mão na imagem da câmera) e realizar cliques em botões virtuais, possibilitando o acionamento remoto de comandos na máquina monitorada.

Além disso, seria relevante avaliar a adaptação do sistema para suportes de *smartphone* compatíveis com realidade virtual (VR), como os *headsets* de baixo custo, e comparar a sua

eficiência, usabilidade e precisão com dispositivos de realidade aumentada (AR) já consolidados no mercado, como os óculos Orion da Meta. Essa análise poderia incluir métricas de desempenho (ex.: latência, taxa de erro) e estudos de ergonomia.

Também seria interessante viabilizar o uso desse sistema em locais sem conectividade à internet, implementando processamento local via *MQTT*, ou comunicação serial cabeada. Essa versão offline poderia ser validada em ambientes industriais, comparando parâmetros como velocidade, praticidade, precisão e interferência.

REFERÊNCIAS

ALBERTIN, M. R. e. a. **Principais Inovações Tecnológicas da Indústria 4.0 E Suas Aplicações E Implicações Na Manufatura**. Bauru, São Paulo, Brasil: [s.n.], 2017.

CORDEIRO, G. A.; ORDONEZ, R. E. C.; FERRO, R.; NICOLELA, P. B.; STELLA, B. C. Etapas para implantação da indústria 4.0: uma visão sob aspectos estratégicos e operacionais. **XXXVII Encontro Nacional De Engenharia De Produção**, 2017.

DUMANS., L. M. S. O. F. M. R. G. Inovações tecnológicas no setor automobilístico: impactos e tendências. 1998. Disponível em: <https://d1wqtxts1xzle7.cloudfront.net/88098299/ENEGEP1998_ART191-libre.pdf?1656525049=&response-content-disposition=inline%3B+filename%3DInovacoes_tecnologicas_no_setor_automobi.pdf&Expires=1694150267&Signature=Mf4pJ3nCgQkk4VCOMMyQVOnXdEYSLCnPCC-ZMW~Hd0MRrvZDi03oi00pDVQOaUBIZ3eDGJGAhM_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA>.

GHINATO, P. Sistema toyota de produção: mais do que simplesmente just-in-time. **Production**, SciELO Brasil, v. 5, p. 169–189, 1995.

GIUSTO D., A. I. G. M.; ATZORI, L. The internet of things. **Springer**, 2010.

HERMANN M; PENTEK, T. O. B. Design Principles for Industrie 4.0 Scenarios: A Literature Review. 2018. Disponível em: <https://repositorio.unip.br/wp-content/uploads/tainacan-items/198/10782/eng_edsonpereiradasilva.pdf>.

KAGERMANN, H. e. a. Recommendations for implementing the strategic initiative industrie 4.0: Final report of the industrie 4.0 working group. **acatech**, 2013.

PEREIRA, J. A. **Comunicação bidirecional entre CLP e aplicativo móvel para automação de processos industriais**. Dissertação (B.S. thesis), 2023.

PETRUZELLA, F. D. **Controladores Lógicos Programáveis - 4ed.** [S.l.]: AMGH Editora, 2014.

SILVA, E. P. D. **A TRANSIÇÃO DA MANUTENÇÃO INDUSTRIAL PARA O MODELO DO NOVO PARADIGMA DA INDÚSTRIA 4.0**. Dissertação (Mestrado) — UNIVERSIDADE PAULISTA, 2018.

SOON, T. J. Qr code. **synthesis journal**, v. 2008, p. 59–78, 2008.