

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

DANIEL SOUZA DOS SANTOS

REGISTRO DE ACESSO A SALAS DE DESCANSO MONITORADO POR
SISTEMAS EMBARCADOS E INTERNET DAS COISAS

São Luís
2023

DANIEL SOUZA DOS SANTOS

**REGISTRO DE ACESSO A SALAS DE DESCANSO MONITORADO POR
SISTEMAS EMBARCADOS E INTERNET DAS COISAS**

Monografia apresentada à Universidade Federal
do Maranhão como requisito parcial para
obtenção do título de Bacharel em Engenharia da
Computação.
Orientador: Prof. Dr. Pedro Baptista Fernandes

São Luís
2023

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Santos, Daniel Souza dos.

Registro de acesso a salas de descanso monitorado por sistemas embarcados e internet das coisas / Daniel Souza dos Santos. - 2023.

55 f.

Orientador(a): Pedro Baptista Fernandes.

Monografia (Graduação) - Curso de Engenharia da Computação, Universidade Federal do Maranhão, São Luís, 2023.

1. Computador de Placa Única. 2. Identificação por Radiofrequência. 3. Internet das Coisas. 4. Sistemas Embarcados. I. Baptista Fernandes, Pedro. II. Título.

DANIEL SOUZA DOS SANTOS

**REGISTRO DE ACESSO A SALAS DE DESCANSO MONITORADO POR
SISTEMAS EMBARCADOS E INTERNET DAS COISAS**

Monografia apresentada à Universidade Federal do Maranhão como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação.

Aprovada em ____ / ____ / ____

BANCA EXAMINADORA



Documento assinado digitalmente

PEDRO BAPTISTA FERNANDES

Data: 18/12/2023 21:27:56-0300

Verifique em <https://validar.iti.gov.br>

Prof. Pedro Baptista Fernandes (Orientador)

Doutor em Engenharia Elétrica
Universidade Federal do Maranhão



Documento assinado digitalmente

PAULO ROGERIO DE ALMEIDA RIBEIRO

Data: 18/12/2023 20:14:54-0300

Verifique em <https://validar.iti.gov.br>

Prof. Paulo Rogério de Almeida Ribeiro (Examinador)

Doutor em Neurociência
Universidade Federal do Maranhão

Prof. Adriano Mendes Magalhães (Examinador)

Mestre em Engenharia Elétrica
Universidade Federal do Maranhão

*A todos aqueles que nos
motivam a seguir em
frente.*

AGRADECIMENTOS

Aos meus pais, Paulo e Sione, e aos meus irmãos, Guido, Ednardo, e Junior pelo amor incondicional, pela confiança depositada e pelos sacrifícios para eu chegar aonde estou.

Aos colegas de curso pelo apoio livre e espontâneo durante o decorrer destes longos anos, pois, sem dúvida, sem eles eu não conseguiria chegar a este ponto. Agradecimento em especial a Leonardo Venâncio, Danilo Augusto Gomes, João Vitor Roma, João Baluz, Lucas Ferreira, Piercarlo Guinzani, Billyfranklim Pereira, Geovana Guterres, Vanessa Oliveira, José Victor, Jordan Pereira, Otávio Costa, André Cardoso e Thalles dos Anjos.

Aos colegas de profissão Eugênio Araújo, por ter me dado a minha primeira oportunidade real na área de tecnologia da informação e a Ricardo Frias, por acreditar no meu potencial e poder explorar mais o mundo de sistemas embarcados profissionalmente. A Moisés Correa, por ter dado todo o suporte e incentivo na elaboração deste trabalho e pela excelente gestão de pessoas ao longo dos anos. A Rodrigo Ramada, André Andrade, Francisco Carneiro e Paula Pedroso pela honra de poder dividir o cotidiano e as situações inusitadas do trabalho.

Aos excelentes professores que tive nessa jornada nesses longos anos, com destaque para Sérgio Costa, Bruno Feres, Luís Claudio Oliveira, Claudio Aroucha, Davi Viana e Alana Teixeira. Com menção honrosa a Pedro Fernandes, pela valiosa orientação na monografia.

Por fim, e de grandíssima importância, à minha companheira de todos os momentos, Karla Graziella Silva, pelo seu apoio, incentivo, cobrança e carinho durante todo este tempo em que estivemos caminhando lado a lado e construindo um futuro.

“[...] Só se vê bem com o coração. O essencial é invisível para os olhos.”

Antoine de Saint-Exupéry

RESUMO

Este estudo propõe a pesquisa sobre sistemas embarcados e Internet das Coisas para atender à necessidade de registrar o tempo de descanso de trabalhadores expostos ao calor. O protótipo desenvolvido utiliza um computador de placa única Raspberry Pi 4 integrando identificação por radiofrequência para capturar informações do usuário. Uma aplicação dedicada foi desenvolvida para coletar e enviar essas informações ao serviço de nuvem do Microsoft Azure. A fundamentação teórica abrange uma revisão de conceitos, incluindo arquitetura de computadores, microcontroladores, sistema em um chip, Raspberry Pi e Internet das Coisas. A pesquisa resultou na leitura e armazenamento local e na nuvem de informações críticas, como usuário, data e hora. Além disso, o sistema apresenta métricas de tempo de preparação e inicialização, juntamente com análise de custos. Os resultados indicam que o sistema atende aos requisitos propostos pelo cliente, evidenciando sua eficácia na monitorização do tempo de descanso em ambientes de trabalho expostos ao calor.

Palavras-chave: Sistemas Embarcados. Computador de Placa Única. Identificação por Radiofrequência. Internet das Coisas.

ABSTRACT

This study proposes research on embedded systems and the Internet of Things (IoT) to address the need for recording the rest time of workers exposed to heat. The developed prototype utilizes a Raspberry Pi 4 single-board computer, integrating radiofrequency identification to capture user information. A dedicated application has been developed to collect and send this information to the Microsoft Azure cloud service. The theoretical foundation includes a review of concepts, encompassing computer architecture, microcontrollers, system-on-chip, Raspberry Pi, and the Internet of Things. The research resulted in the local and cloud-based reading and storage of critical information, such as user, date, and time. Furthermore, the system provides metrics for preparation and startup times, along with cost analysis. The results indicate that the system meets the requirements proposed by the client, demonstrating its effectiveness in monitoring rest time in heat-exposed work environments.

Keywords: Embedded Systems. Single-Board Computer. Radiofrequency Identification. Internet of Things.

LISTA DE FIGURAS

Figura 3.1 - Componentes básicos de um computador	19
Figura 3.2 - Projeto <i>Whirlwind</i>	21
Figura 3.3 - Intel 4004.....	22
Figura 3.4 - Microcontrolador <i>Freescale MC9S12C32</i>	23
Figura 3.5 - Alguns microcontroladores comuns	23
Figura 3.6 - Placas de desenvolvimento de Microcontroladores.....	24
Figura 3.7 - Raspberry Pi OS	26
Figura 3.8 - Visão simbólica da Internet das Coisas.....	27
Figura 3.9 - Endereços de interface	29
Figura 3.10 - Modelo OSI para Protocolos IoT.....	30
Figura 3.11 - Funcionamento do RFID	32
Figura 4.1 - Tamanho do SBC em Comparação.....	34
Figura 4.2 - Raspberry Pi 4	35
Figura 4.3 - Leitor RFID USB.....	36
Figura 4.4 - Tela (1), Dissipadores (2) e Estojo (3).....	36
Figura 4.5 - Modem 4G.....	37
Figura 4.6 - Fluxograma de funcionamento do leitor RFID	38
Figura 4.7 - Fluxograma de funcionamento do sincronizador	40
Figura 5.1 - Tela inicial da aplicação.....	42
Figura 5.2 - Registro de entrada	43
Figura 5.3 - Registro de saída.....	43
Figura 5.4 - Listagem dos registros armazenados localmente.....	44
Figura 5.5 - Tempo Estimado de Gravação.....	47
Figura 5.6 - Arquivo de configuração do sistema.....	47
Figura 5.7 - Energização do SBC	49
Figura 5.8 - Tela Inicial.....	49

LISTA DE TABELAS

Tabela 4.1 - Materiais Utilizados.....	35
Tabela 5.1 - Dados resumidos por usuário.....	45
Tabela 5.2 - Dados de Funcionamento Normal.....	46
Tabela 5.3 - Custos dos componentes do Projeto.....	50

LISTA DE QUADROS

Quadro 3.1 - Comparação entre Microprocessador e Microcontrolador	24
Quadro 3.2 - Fabricantes e SBCs.....	25
Quadro 3.3 - Componentes de IoT.....	28
Quadro 3.4 - Camadas do modelo OSI	29
Quadro 3.5 - Protocolos e termos de IoT	30
Quadro 3.6 - Tecnologia wireless de curto alcance	31
Quadro 4.1 - Requisitos de usuário e sistema.....	33
Quadro 4.3 - Etapas do processo de aquisição de dados.....	39
Quadro 4.4 - Sincronização das informações do usuário	41
Quadro 4.5 - Sincronização de funcionamento	41
Quadro 5.1 - Tempo Estimado de Preparação.....	46
Quadro 5.2 - Campos de Configuração.....	48
Quadro 5.3 - Principais problemas e soluções.....	48

LISTA DE SIGLAS

CPU	Unidade Central de Processamento
E/S	Entrada e Saída
ULA	Unidade de Lógica e Aritmética
RAM	Memória de Acesso Aleatório
ROM	Memória Somente Leitura
ADC	Conversor Analógico-Digital
DAC	Conversor Digital-Analógico
PCB	Placa de Circuito Impresso
CI	Circuito Integrado
ASIC	Circuito integrado de aplicação específica
SoC	Sistema em um Chip
VLSI	Integração em muito larga escala
IoT	Internet das Coisas
OSI	Interconexão de Sistemas Abertos
IP	Protocolo de Internet
SBC	Computador de Placa Única
JSON	Notação de Objeto Javascript
API	Interface de Programação de Aplicação
HTTPS	Protocolo Seguro de Transferência de Hiper Texto
RFID	Identificação por Radiofrequência
USB	Barramento Serial Universal

SUMÁRIO

1 INTRODUÇÃO	15
1.1 TRABALHOS ANTERIORES	16
1.2 ESTRUTURA DO TRABALHO	16
2 OBJETIVOS	18
2.1 OBJETIVO GERAL	18
2.2 OBJETIVOS ESPECÍFICOS	18
3 FUNDAMENTAÇÃO TEÓRICA	19
3.1 ARQUITETURA DE COMPUTADORES.....	19
3.1.1 Sistemas Embarcados	20
3.1.2 Microcontroladores	22
3.1.3 Sistema em um Chip	25
3.1.4 Raspberry Pi	26
3.2 INTERNET DAS COISAS	27
3.3 PROTOCOLOS DE COMUNICAÇÃO	28
4 METODOLOGIA	33
4.1 REQUISITOS	33
4.2 MODELAGEM DO SISTEMA.....	34
4.3 MATERIAIS UTILIZADOS.....	34
4.3.1 Computador	35
4.3.2 Leitor de Dados	35
4.3.3 Estojo	36
4.3.4 Comunicação com a Internet	37
4.3.5 Azure	37
4.3.6 Desenvolvimento do Software	37
4.4 FUNCIONAMENTO	38
4.4.1 Aquisição e Armazenamento	38
4.4.2 Envio e Sincronização de Dados	39
4.4.2.1 Envio de Informações do Usuário	40
4.4.2.2 Envio das Informações de Funcionamento	41
5 RESULTADOS	42
5.1 TESTES COM O USUÁRIO	42

5.1.1	Testes de Leitura e Interação	42
5.1.2	Armazenamento e Sincronização	44
5.1.3	Status de Funcionamento	45
5.2	MÉTRICAS	46
5.2.1	Tempo de Preparação.....	46
5.2.2	Mapeamento de manutenção	48
5.2.3	Tempo de inicialização	48
5.2.4	Custos do projeto.....	50
6	CONSIDERAÇÕES FINAIS	51
6.1	SUGESTÕES PARA TRABALHOS FUTUROS	51
	REFERÊNCIAS.....	53
	APÊNDICE A – LISTAGEM COMPLETA DOS REGISTROS DE TESTE.....	56

1 INTRODUÇÃO

Locais de trabalho que operam em ambientes insalubres, nos quais a exposição ao calor é elevada, como na indústria, estão sujeitos a regulamentações específicas relacionadas às normas de trabalho, especialmente no que diz respeito ao descanso dos colaboradores. Diferentemente dos contratos de trabalho convencionais, nos quais os colaboradores geralmente trabalham por jornadas iguais ou superiores a seis horas e têm um período de descanso mínimo de uma hora, de acordo com suas funções [1], pessoas que trabalham expostas a altas temperaturas devem operar apenas por uma quantidade pré-determinada de minutos por hora, conforme estabelecido no Anexo 3 da Norma Regulamentadora nº 15 do Ministério do Trabalho e Emprego. Eles não devem exceder esse limite e devem dedicar o restante do tempo ao descanso, além de terem pausas regulares para refeições [2].

O colaborador deve desfrutar de períodos de descanso por hora, podendo variar entre 15, 30 ou 45 minutos por hora, dependendo da temperatura à qual está exposto e do esforço físico que realiza. É imperativo que essas pausas para recuperação térmica sejam rigorosamente respeitadas, e seus inícios e finais devem ser registrados, assegurando assim o monitoramento preciso do descanso total e diário. Dado que os colaboradores portam crachás equipados com chips de identificação por radiofrequência (RFID), existem oportunidades para explorar estratégias que permitam a obtenção das informações registradas nesses dispositivos.

A empresa em questão dispõe de salas de descanso, no entanto, estas estão localizadas em áreas onde não há conectividade com a internet por meio de rede cabeada ou sem fio.

No mercado, há diversas soluções para monitorar e controlar o acesso a esses locais, como o relógio de ponto eletrônico Henry Prima [3]. No entanto, esses equipamentos costumam ter um custo elevado e podem demandar um tempo significativo até a conclusão da sua implantação. Além disso, essas soluções são frequentemente limitadas em termos de gestão de dados e requerem investimentos adicionais em infraestrutura de rede para permitir a extração e o tratamento automatizado dos dados.

Entre as limitações do uso de soluções tradicionais, destaca-se a dependência do fornecedor para realizar modificações, além da possibilidade de o sistema ficar inoperante enquanto aguarda a chegada de manutenção especializada em caso de falha.

Computadores de placa única, como o Raspberry Pi 4, oferecem portas USB, armazenamento interno e placa de rede embutidos [4]. Embora seja possível desenvolver um sistema semelhante utilizando microcontroladores como o Arduino ou ESP32, optar por

ferramentas prontas para uso proporcionam ganho de tempo no desenvolvimento e simplificam a implementação, uma vez que não é necessário criar códigos adicionais e a quantidade de conexões é reduzida a apenas uma porta USB.

Para efeito de comparação, um módulo RC522 requer sete conexões às portas de um microcontrolador Arduino e, no mínimo, dez linhas de código em linguagem *Wiring* para realizar a leitura de informações presentes em um cartão RFID [5].

1.1 TRABALHOS ANTERIORES

Agus Bejo et al [6] apresentam um protótipo para registro de presença em sala de aula que utiliza um Raspberry Pi 3, um leitor RFID USB e um servidor web. Os autores desenvolvem aplicações que leem, escrevem e registram as informações dos crachás dos alunos por aproximação. Essas informações são salvas em um banco de dados web com o qual o Raspberry Pi se comunica através da rede WiFi.

Zeeshan Ali et al [7] desenvolveram um sistema que utiliza um Raspberry Pi B+, um leitor RFID do tipo RC522 e um servidor na rede para ler as etiquetas RFID dos produtos inseridos em um carrinho de supermercado. As informações do produto, como o nome e o preço, são exibidas na tela do usuário, com o objetivo de proporcionar controle de custos e facilitar o pagamento das compras.

A pesquisa de Omkar et al [8] mostra o desenvolvimento de um sistema utilizando um Raspberry Pi, um leitor RFID RC522 e um serviço em nuvem para capturar dados dos usuários do sistema de saúde, integrando diversos serviços de saúde. O usuário é cadastrado, e todos os exames e consultas realizados pelo usuário são armazenados em um sistema em nuvem, sendo o cartão do usuário a chave de acesso para todo o histórico.

1.2 ESTRUTURA DO TRABALHO

Com o intuito de organizar e facilitar a consulta e compreensão dos temas abordados, este trabalho está dividido em seis capítulos. O primeiro capítulo aborda aspectos introdutórios sobre o tema em questão.

O segundo capítulo apresenta o objetivo geral deste trabalho, além dos seus objetivos específicos.

No terceiro capítulo, realiza-se uma breve revisão da fundamentação teórica, explorando conceitos relacionados à arquitetura de computadores, sistemas embarcados, microcontroladores, sistema em um chip com o exemplo do Raspberry Pi, bem como internet das coisas e protocolos de comunicação.

O quarto capítulo discorre sobre a metodologia adotada para alcançar os objetivos propostos, abrangendo desde os requisitos de usuário e sistema até a modelagem e os materiais utilizados para a construção do primeiro protótipo. Além disso, é apresentada uma descrição passo a passo do seu funcionamento.

No quinto capítulo, são apresentados os resultados obtidos, englobando testes de leitura e interação, armazenamento e sincronização, juntamente com o status de funcionamento. O capítulo encerra-se com a apresentação de algumas métricas coletadas.

O sexto capítulo finaliza o trabalho, apresentando conclusões, contribuições e sugestões para futuros trabalhos. O capítulo é seguido pelas referências e pelo apêndice.

2 OBJETIVOS

2.1 OBJETIVO GERAL

O objetivo geral deste trabalho é apresentar um protótipo de um dispositivo destinado a registrar o tempo de repouso cumprido por um colaborador em uma sala de descanso. Essas salas estão localizadas em áreas desprovidas de infraestrutura de rede cabeada ou sem fio, dependendo exclusivamente de dados móveis.

O tempo de descanso é calculado a partir dos registros de horários de entrada e saída, os quais são posteriormente enviados a um serviço de gerenciamento de banco de dados em nuvem. O projeto abrange diversas etapas, desde o desenvolvimento do software de gerenciamento até a instalação em um dispositivo, incluindo seus respectivos periféricos.

2.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho são:

1. Utilizar um computador de placa única Raspberry Pi, executando seu sistema operacional próprio.
2. Desenvolver uma aplicação que, em primeiro plano, capture informações do usuário por identificação por radiofrequência e as armazene localmente.
3. Implementar dois serviços em segundo plano: o primeiro sincroniza as informações armazenadas localmente, enquanto o segundo envia dados de funcionamento normal do dispositivo.
4. Integrar ambos os serviços a uma estrutura de gerenciamento de dados na nuvem, utilizando a plataforma Microsoft Azure®, para a primeira versão funcional do protótipo.

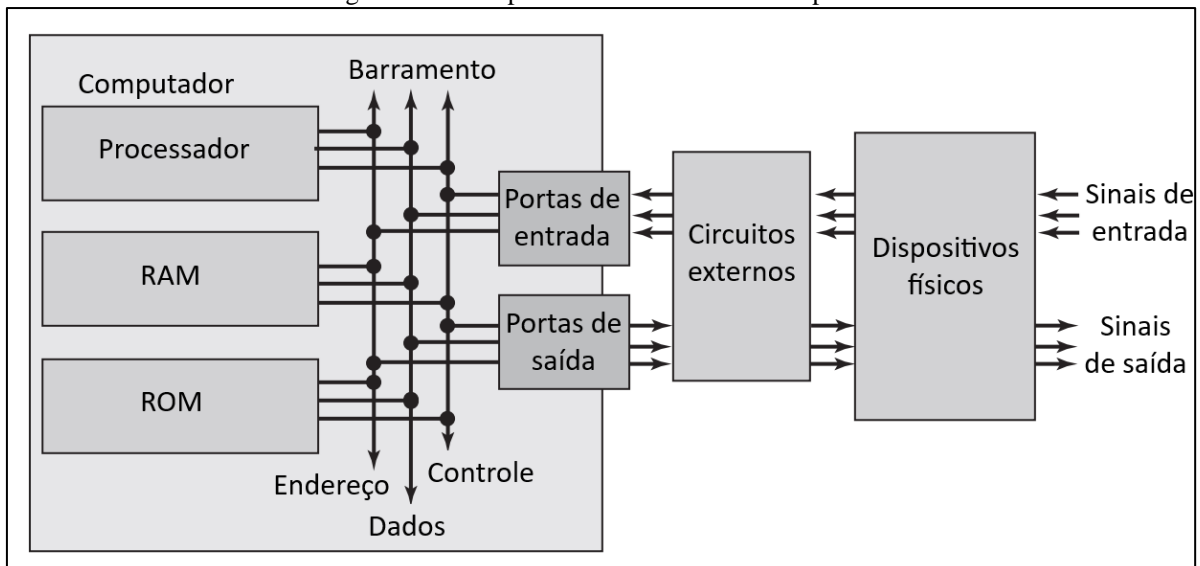
3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são discutidos os principais referenciais teóricos que serão empregados no desenvolvimento do sistema, abrangendo tanto aspectos relacionados ao hardware quanto ao software. A exposição tem início com uma concisa introdução acerca de computadores e sistemas embarcados, prosseguindo para uma análise da Internet das Coisas (IoT) e culminando na consideração de periféricos e protocolos de comunicação.

3.1 ARQUITETURA DE COMPUTADORES

Um sistema computacional é composto por um conjunto de elementos, incluindo um processador, uma memória de acesso aleatório (RAM), uma memória somente leitura (ROM) e unidades de entrada e saída. Um software, por sua vez, consiste em uma sequência ordenada de instruções específicas para realizar tarefas, armazenadas na memória ROM. O processador é responsável por executar as instruções do software, buscando e interpretando-as uma a uma [9]. A estrutura básica de um computador pode ser visualizada na Figura 3.1.

Figura 3.1 - Componentes básicos de um computador



Fonte: Tradução nossa de [9]

O computador exerce a coordenação de todos os dispositivos presentes, desempenhando diversas funções essenciais. Por exemplo, ele realiza a cópia do software da memória ROM para a memória RAM, executa as instruções presentes na RAM, facilita a comunicação entre as portas de entrada e saída e gerencia a transmissão de dados por meio do barramento. O barramento é uma coleção de sinais agrupados com um propósito comum,

podendo incluir informações como endereço dos dispositivos, dados de transferência, controle de horários e direção das informações [9].

3.1.1 Sistemas Embarcados

A definição precisa de um sistema embarcado ainda não atingiu um consenso unânime entre os estudiosos, uma vez que o tema está em constante evolução e os autores divergem em suas interpretações. Contudo, as características comumente encontradas em comum indicam que esses sistemas possuem restrições de hardware quando comparados aos computadores convencionais. Além disso, são projetados para executar uma tarefa ou conjunto específico de tarefas, diferenciando-se assim pela sua especialização e foco [10].

Conforme Marilyn Wolf [11], de maneira geral, um sistema embarcado é caracterizado como qualquer dispositivo que incorpora um computador programável, embora seu propósito não seja ser um computador de propósito geral. Essa definição destaca a natureza específica e a finalidade direcionada desses sistemas, que são projetados para atender a objetivos particulares além de simples funções de propósito geral.

Rodrigo Almeida et al [12] descrevem os sistemas embarcados como sistemas eletrônicos microprocessados, cuja função específica, uma vez programada, geralmente não pode ser alterada.

Michael Barr e Anthony Massa [13] definem sistemas embarcados como uma combinação entre hardware e software de computador, projetada para executar uma função dedicada, podendo incluir outros componentes de natureza eletrônica ou mesmo mecânica.

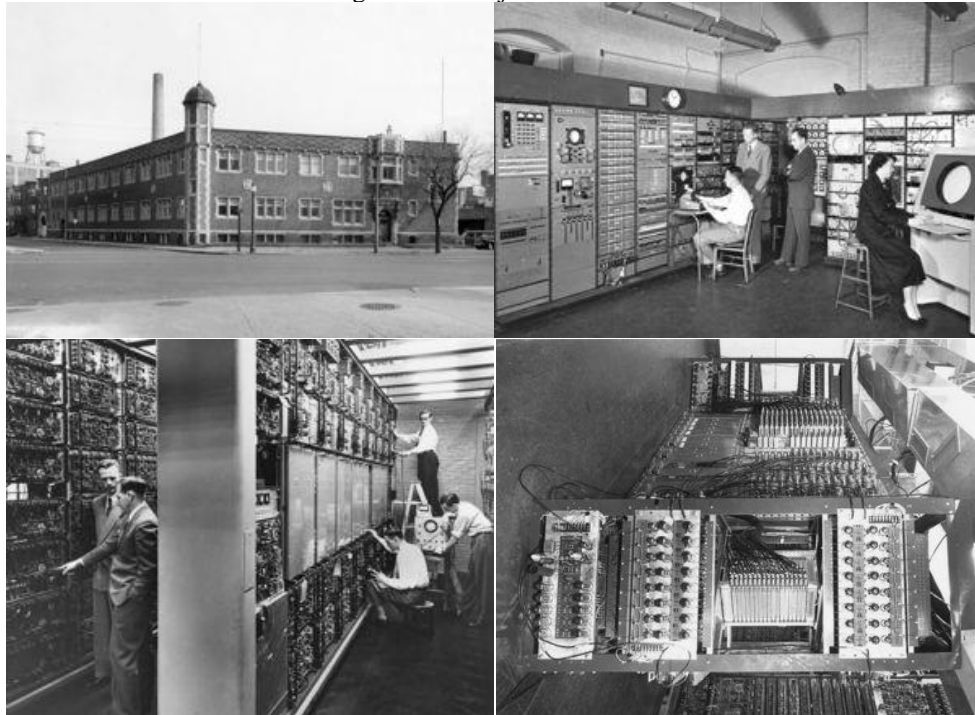
Segundo Jonathan Valvano [9], um sistema de computador embarcado é um microcomputador integrado a um sistema eletrônico, configurado para executar uma aplicação dedicada e específica. Essas definições destacam a natureza microprocessada, a função específica e a integração entre hardware e software dos sistemas embarcados.

Desde os primórdios da computação, os computadores têm sido incorporados em diversas aplicações, como exemplificado pelo *Whirlwind*. Desenvolvido pelo MIT entre o final da década de 1940 e início da década de 1950, o *Whirlwind* desempenhou um papel crucial no projeto de pesquisa "*Aircraft Stability and Control Analyzer*". Este projeto visava criar uma plataforma de treinamento universal para simulação de voo, evidenciando a presença de sistemas embarcados mesmo nas fases iniciais da computação [14].

Apesar de ser classificado como um sistema embarcado, é importante observar que o Whirlwind não se enquadrava necessariamente nas dimensões compactas típicas dos sistemas embarcados contemporâneos. Este sistema, por exemplo, era composto por mais de 4000 tubos a vácuo [11] e ocupava uma área superior a 200 metros quadrados (2500 pés quadrados), instalado no segundo andar do Prédio Barta, no Instituto de Tecnologia de Massachusetts [14]. Essas dimensões destacam as diferenças significativas entre os sistemas embarcados do passado e os modernos em termos de tamanho e complexidade.

A Figura 3.2 proporciona uma visão das instalações do primeiro computador embarcado e alguns de seus componentes no interior do prédio. Essas imagens ilustram não apenas a escala considerável do sistema, mas também a complexidade dos componentes envolvidos na época.

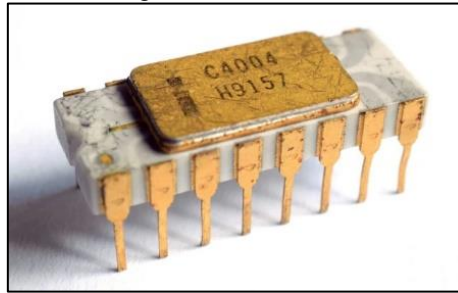
Figura 3.2 - Projeto *Whirlwind*



Fonte: [14]

Somente na década de 1970, foi possível consolidar um computador completo em um único chip. Em 1971, a Intel lançou o pioneiro microprocessador, o Intel 4004, como apresentado na Figura 3.3. Inicialmente concebido como um computador de propósito geral para atender às demandas de vários modelos de calculadoras da época, este microprocessador monolítico de 4 bits era capaz de endereçar 4096 locais na memória e possuía 45 instruções distintas [15].

Figura 3.3 - Intel 4004



Fonte: [16]

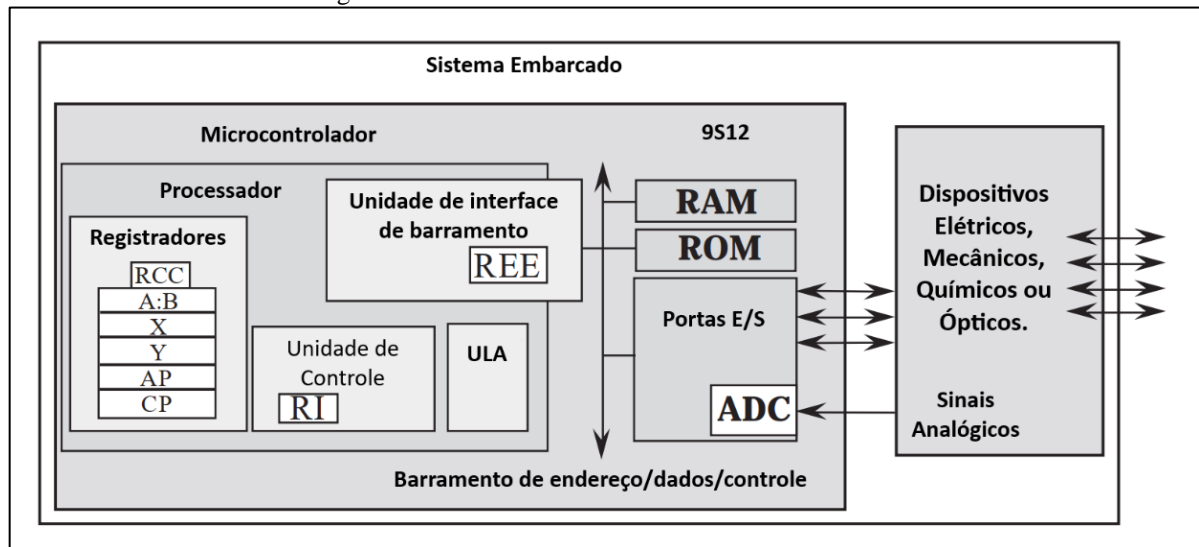
A abordagem de utilizar o mesmo processador, mas com programações específicas para cada modelo de calculadora, representou uma evolução significativa ao permitir a produção em massa de processadores monolíticos, reduzindo a necessidade de criar chips exclusivos para cada produto. Esse avanço contribuiu para a disseminação e a acessibilidade de sistemas embarcados em uma variedade de dispositivos de propósito específico [11], [13].

3.1.2 Microcontroladores

Na década seguinte ao lançamento do primeiro microprocessador, os fabricantes reconheceram a demanda por sistemas programáveis em um único chip, que fossem compactos e acessíveis para pequenas aplicações dedicadas. Surgiram, assim, os microcontroladores, como observa Kani [15].

Os microcontroladores são dispositivos semelhantes aos microprocessadores, porém projetados para operar como um único chip que engloba todos os componentes do sistema. Um microcontrolador típico inclui uma Unidade Lógica e Aritmética (ULA), registradores, temporizadores, unidades de controle (semelhantes aos microprocessadores) e integra unidades de entrada e saída, temporizador programável, memória RAM, memória ROM, além de conversores analógico-digital (ADC) e digital-analógico (DAC) [15].

Cada microcontrolador detém um conjunto único de instruções e componentes internos, apesar de compartilharem semelhanças notáveis em termos de arquitetura. A Figura 3.4 exemplifica o microcontrolador *Freescale* MC9S12C32.

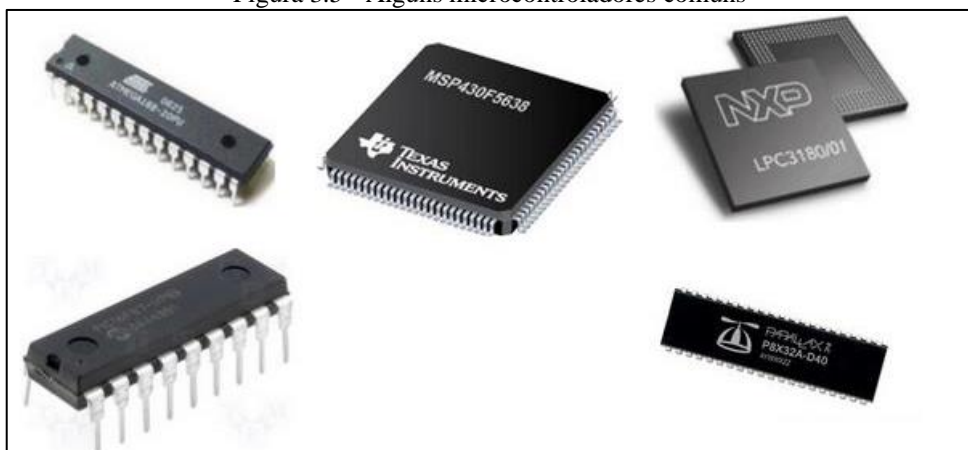
Figura 3.4 - Microcontrolador *Freescale MC9S12C32*

Fonte: Tradução nossa de [9]

No contexto fornecido, "REE" refere-se ao Registrador de Endereço Efetivo, responsável por conter os endereços de memória utilizados para buscar os dados necessários à instrução em execução. "RCC" é o Registrador de Código de Condição (também denominado registrador de flag em processadores Intel), que armazena o status da operação precedente. "AP" é o Apontador de Pilha, indicando a posição na RAM e definindo a pilha. "CP" é o Contador de Programa, apontando para a próxima instrução a ser executada. "RI" representa o Registrador de Instruções, contendo o código de operação da instrução em curso [9].

Diversos fabricantes oferecem uma ampla variedade de modelos de microcontroladores. Alguns dos destacados incluem o ARM *Cortex-M*, Atmel AVR / AVR 32, Intel 8051, Microchip PIC, NXP LCP 2000 / 3000 e MC9S12C32, *Parallax Propeller*, *Espressif ESP32*, *Texas Instruments MSP430* [12], [15], [17], [18]. A Figura 3.5 ilustra alguns desses Circuitos Integrados (CIs).

Figura 3.5 - Alguns microcontroladores comuns



Fonte: [17]

O Quadro 3.1 apresenta as diferenças fundamentais entre microcontroladores e microprocessadores de maneira abrangente.

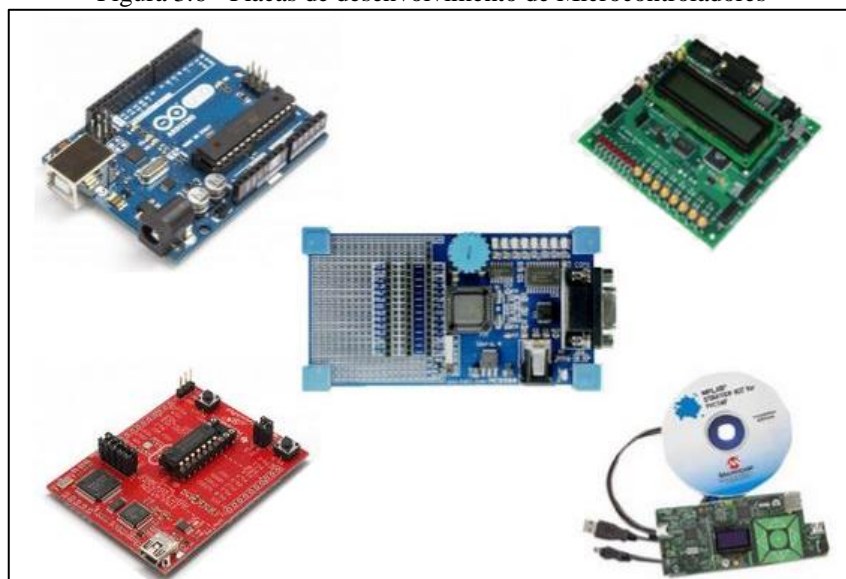
Quadro 3.1 - Comparação entre Microprocessador e Microcontrolador

Microprocessador	Microcontrolador
1. Os componentes funcionais fundamentais de um microprocessador incluem a Unidade Lógica e Aritmética (ULA), registradores e unidade de temporização e controle.	1. O microcontrolador incorpora os blocos funcionais de um microprocessador e, adicionalmente, dispõe de um temporizador, uma porta de E/S paralela, uma porta de E/S serial, bem como RAM e ROM internas. Alguns controladores incluem ADC e/ou DAC.
2. A principal preocupação de um microprocessador é a transferência eficiente de código e dados entre a memória externa e o próprio microprocessador. Conseqüentemente, apresenta uma variedade de instruções para movimentação de dados entre a memória externa e o microprocessador.	2. A ênfase do microcontrolador está na movimentação eficiente de código e dados dentro do próprio microcontrolador. Conseqüentemente, apresenta um número reduzido de instruções para transferência de dados entre a memória externa e o microcontrolador.
3. Microprocessadores operam predominantemente em dados de byte ou palavra, resultando em um número limitado de instruções para manipulação de bits.	3. Microcontroladores frequentemente lidam com manipulação de bits, possuindo, assim, diversas instruções para essa finalidade.
4. A implementação de sistemas baseados em microprocessadores geralmente exige a interface com um grande número de Circuitos Integrados (CI) adicionais, o que pode resultar em Placa de Circuito Impressas (PCB) extensas e, conseqüentemente, em custos elevados.	4. Os microcontroladores podem ser utilizados para formar um sistema único baseado em microcomputador, sem a necessidade de CIs adicionais. Dessa forma, a PCB de um sistema baseado em microcontrolador tende a ser compacta, resultando em um sistema mais econômico.
5. Os microprocessadores são empregados no projeto de sistemas de computação digital de propósito geral (ou computadores).	5. Microcontroladores são empregados no desenvolvimento de sistemas dedicados, específicos para determinadas aplicações.

Fonte: Tradução nossa e adaptado de [15]

Muitos desses microcontroladores disponibilizam suas próprias PCBs para estudo ou prototipagem [17]. Algumas das placas de desenvolvimento mais reconhecidas podem ser observadas na Figura 3.6, como a do Arduino no canto superior esquerdo.

Figura 3.6 - Placas de desenvolvimento de Microcontroladores



Fonte: [17]

Deve-se escolher o componente mais recomendado para o projeto. Embora uma PCB própria torne a prototipagem mais fácil, para sistemas embarcados, geralmente utiliza-se o microcontrolador em conjunto com uma placa específica para o projeto [12].

3.1.3 Sistema em um Chip

Em uma linha paralela, a partir da década de 1990, a tecnologia de Circuito Integrado de Aplicação Específica – ASIC (do inglês, *Application-Specific Integrated Circuit*) evoluiu de uma filosofia baseada em chips para uma abordagem de Sistema em um Chip – SoC (do inglês, *System on a Chip*). Define-se SoC como um CI, projetado unindo vários projetos de integração em muito larga escala – VLSI (do inglês, *Very Large Scale Integration*) autônomos, fornecendo funcionalidade completa para uma aplicação específica [19].

Os SoCs podem ser divididos em duas categorias: os microcontroladores de placa única e os computadores de placa única – SBC (do inglês, *Single-Board Computer*). Por um lado, os microcontroladores apresentam grande eficiência ao executar tarefas específicas, além de suporte a interrupções de hardware e eventos em tempo real. Por outro lado, eles requerem firmware dedicado. Já um sistema operacional (SO) executado em um SBC tem o potencial de oferecer maior flexibilidade e versatilidade, pois o cenário pode ser fluido e de rápida evolução. No entanto, há a possibilidade de perder o controle em tempo real em relação a interrupções [20].

Rahman et al [21] definem um SBC como um computador completo em uma única placa de circuito impresso, com custo reduzido em comparação com desktops. Seu uso abrange uma ampla gama de aplicações, desde controle industrial e instrumentos médicos até comunicação de rede e setor automotivo, entre outros.

Isikdag [22] lista alguns dos mais conhecidos SBCs até a época da concepção de seu artigo, dos quais podemos destacar no Quadro 3.2.

Quadro 3.2 - Fabricantes e SBCs

Fabricante	Produto
Arduino	Arduino Yún
BeagleBoard	BeagleBone Black
CubieBoard	CubieBoard
Raspberry Pi	Raspberry Pi
Orange Pi	Orange Pi
UDOO	UDOO Board
Netduino	Netduino
Intel	Galileo and Edison
Radxa Rock	Radxa Rock

Fonte: [22]

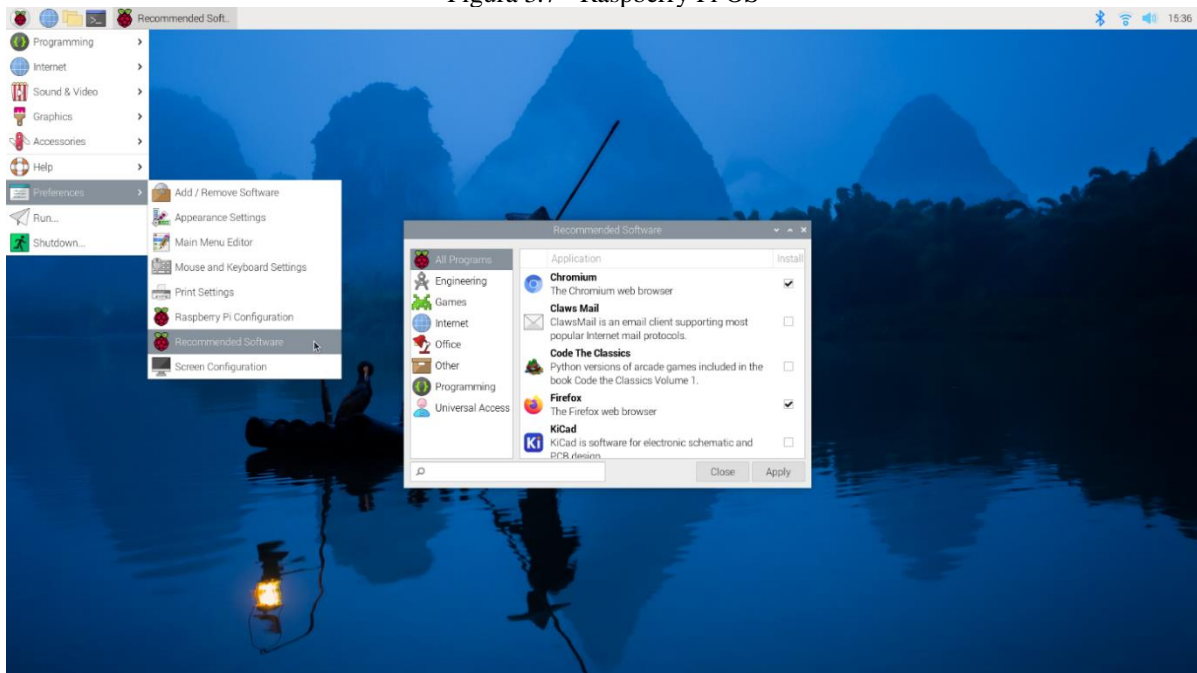
3.1.4 Raspberry Pi

Conforme mencionado por Isikdag [22], o Raspberry Pi é um SBC de custo acessível bastante conhecido. Ele oferece a capacidade de conectar periféricos de computador, como teclado, mouse e monitor. A concepção inicial do Raspberry Pi originou-se da iniciativa de funcionários da Universidade de Cambridge, que visavam desenvolver um computador compacto e acessível, voltado especialmente para atividades educacionais destinadas a crianças.

Atualmente, a Raspberry Pi Foundation oferece uma variedade de produtos distintos, abrangendo diferentes gerações e modelos. Inicialmente, houve o lançamento do Raspberry Pi 1, que incluía os modelos A e B, evoluindo posteriormente para o Raspberry Pi 5. Além disso, destacam-se a linha Zero por seu formato compacto, e os microcontroladores da série Pico [4].

Os SBCs Raspberry Pi operam com um sistema operacional dedicado, o Raspberry Pi OS, que é gratuito e baseado no Debian Linux. Esse sistema oferece uma ampla gama de funcionalidades, que vão desde acesso à internet até a reprodução de vídeos em alta definição, além de disponibilizar editores de texto, planilhas eletrônicas e jogos [4], [22]. A Figura 3.7 apresenta a tela inicial e o menu de aplicativos do sistema operacional Raspberry Pi OS.

Figura 3.7 - Raspberry Pi OS



Fonte: [4]

É evidente que a interface e a usabilidade são semelhantes às de sistemas operacionais convencionais, utilizados em desktops, smartphones e tablets.

3.2 INTERNET DAS COISAS

O termo "Internet das Coisas" surgiu da interconexão de vários sistemas eletrônicos em uma rede única, onde cada nó se comunica com uma central, trocando dados e descentralizando o processamento para permitir que cada extremidade opere de forma independente [23].

Marilyn Wolf [11] define Internet das Coisas – IoT como um sistema de computação embarcado em rede em tempo real, flexível, que abrange dispositivos de entrada e saída, processamento de dados e ciberfísicos.

Segundo Xiao [24]], a IoT é uma rede de objetos físicos, onde "coisas" podem variar desde objetos inanimados, como máquinas, mercadorias, eletrodomésticos e veículos, até edifícios, seres vivos como pessoas, animais e plantas. Este conceito difere da internet tradicional, pois envolve bilhões de dispositivos interconectados capazes de trocar dados entre si sem intervenção humana. A Figura 3.8 ilustra simbolicamente a visão dos dispositivos conectados na Internet das Coisas.

Figura 3.8 - Visão simbólica da Internet das Coisas



Fonte: [24]

Para Cirani et al [25], a IoT é uma visão abrangente do mundo onde bilhões de objetos com inteligência embarcada, meios de comunicação, e capacidade para sensores e atuadores se conectam via rede IP. As aplicações da IoT incluem automação residencial e predial, cidades inteligentes, redes inteligentes, IoT industrial e agricultura inteligente.

Wolf [11] acrescenta que os dispositivos modernos de IoT possuem capacidade de armazenamento e processamento, consomem pouca energia e são mais acessíveis. Ela destaca

que, ao longo do tempo, os avanços nas comunicações sem fio permitiram que esses dispositivos se comunicassem entre si e com sistemas tradicionais.

Singh et al [26] apresentam uma arquitetura de Internet das Coisas composta por dez componentes, conforme mostrado no Quadro 3.3.

Quadro 3.3 - Componentes de IoT

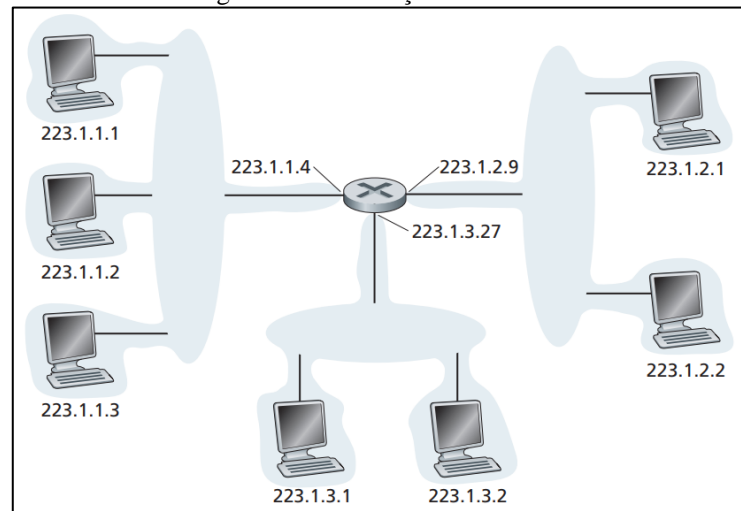
#	Componente	Descrição
1	Coisa	IoT está interligada com vários sensores para coletar os dados e atuadores para executar ações correspondentes aos comandos recebidos da nuvem.
2	Gateway	Filtragem de dados, pré-processamento e comunicação para a nuvem e vice-versa (recebendo os comandos da nuvem)
3	Gateway em nuvem	Transmissão dados entre os gateways e os servidores centrais da IoT.
4	Processador de dados de streaming	Distribuição dos dados provenientes dos sensores para os dispositivos relevantes conectados em rede.
5	<i>Data lake</i>	Armazenamento de todos os dados definidos e não definidos.
6	Armazém de Big data	Coleção de dados valiosos.
7	Aplicação de controle	Envio de comandos para os atuadores.
8	Aprendizagem de Máquina	Geração de modelos que aplicam algoritmos em dados, que podem ser usados para controlar aplicações.
9	Aplicação de usuário	Monitoramento dos dados pelos usuários e tomada de decisões sobre o controle de dispositivos conectados.
10	Análise de dados	Processamento manual de dados.

Fonte: Tradução nossa de [26]

3.3 PROTOCOLOS DE COMUNICAÇÃO

A comunicação na Internet ocorre por meio do Protocolo de Internet, ou IP (do inglês, *Internet Protocol*). Este protocolo estabelece um padrão para o envio de pacotes de dados pela Internet entre dois pontos, passando por vários nós intermediários, em uma estratégia de melhor esforço. Cada ponto e nó na rede possui um endereço IP. A Figura 3.9 ilustra uma rede de computadores, onde cada elemento é identificado por um endereço IP de 32 bits, ou quatro bytes [27].

Figura 3.9 - Endereços de interface



Fonte: [27]

Os endereços IP são representados pela notação decimal pontilhada, em que cada byte é expresso em sua forma decimal e separado por pontos. Essa representação varia de 0.0.0.0 a 255.255.255.255 [27]. Não se limitando exclusivamente a computadores, todos os dispositivos que se conectam à internet, incluindo os dispositivos de IoT, utilizam IP para estabelecer comunicação com o ambiente circundante [11].

A fim de padronizar a comunicação entre dispositivos, a Organização Internacional para a Normalização (ISO) desenvolveu o Modelo de Interconexão de Sistemas Abertos (OSI) para redes, composto por sete camadas [27]. O Quadro 3.4 oferece um resumo das responsabilidades de cada uma dessas camadas.

Quadro 3.4 - Camadas do modelo OSI

#	Camada	Responsabilidade
1	Física	Conexão física, propriedades elétricas, trocas de bits
2	Enlace de Dados	Detecção de erro, controle, transporte confiável de dados
3	Rede	Serviço de transmissão de dados ponto-a-ponto
4	Transporte	Transmissão de dados na ordem correta, sem erros e em vários links
5	Sessão	Mecanismo de controle de interação de serviços de usuário final
6	Apresentação	Formato dos dados
7	Aplicação	Interface de aplicação entre a rede e o usuário final

Fonte: [11]

No contexto da Internet das Coisas, o modelo OSI é composto apenas por cinco camadas, excluindo as camadas de sessão e apresentação [26]. A Figura 3.10 apresenta uma ilustração da adaptação do modelo OSI para IoT, destacando os protocolos e termos associados.

Figura 3.10 - Modelo OSI para Protocolos IoT

Camada de Aplicação	REST API, Objetos JSON-IPSO, Objetos Binários
Camada de Transporte	CoAP, MQTT, XMPP, AMQP, LLAP, DDS, SOAP, UDP, TCP, DTLS
Camada de Internet	6LoWPAN, IPv6, uIP, NanoIP
Camada de Enlace	IEEE802.15.4, IEEE802.11, ISO/IEC 8092:2004, NB-IoT, EC-GSM-IoT, Bluetooth, ANT, ISA100.11a, EnOcean, LTE-MTC
Camada Física	Dispositivos, objetos, coisas

Fonte: Tradução nossa de [26]

Os termos e protocolos essenciais, descritos na Figura 3.10, e que servem de base para este trabalho ou fornecem suporte, como o REST, estão resumidos no Quadro 3.5.

Quadro 3.5 - Protocolos e termos de IoT

Protocolo	Descrição
IPv6	O IPv6 é popular como um protocolo de camada de Internet para transmitir pacotes de informações em uma transmissão de ponta a ponta através de várias redes IP (<i>Internet Protocol</i>).
6LoWPAN	6LoWPAN significa IPv6 sobre redes pessoais sem fio de baixo consumo de energia. É uma camada estendida para IPv6 através de links IEEE802.15.4. Opera em 2,4 GHz de frequência com uma taxa de transmissão de dados de 250 kbps.
UDP	<i>User Datagram Protocol</i> , Este protocolo destina-se ao protocolo baseado em IP em rede entre cliente/servidor. O UDP é usado em aplicativos para desempenho em tempo real.
QUIC	<i>Quick UDP Internet Connections</i> . Suporta as conexões multiplexadas entre dois pontos de extremidade sobre (UDP). Ele foi projetado para proteção de segurança para reduzir a latência na conexão e transporte de informações através de uma rede.
DTLS	<i>Datagram Transport Layer Security</i> , protocolo DTLS fornece a privacidade de comunicação para protocolos de datagrama. Ele é usado para prevenção de adulteração, falsificação de mensagens ou espionagem em uma rede.
Web Física	A Web Física é uma abordagem para interconectar dispositivos e acessá-los perfeitamente.
MQTT	<i>Message Queuing Telemetry Transport</i> , O MQTT é um protocolo leve que permite um modelo de mensagens de publicação/assinatura. Usado para conexão remota em uma rede.
DDS	<i>Data Distribution System</i> , Serviço de Distribuição de Dados para sistemas em tempo real. É um código aberto e padrão internacional para abordar a comunicação entre sistemas embarcados e em tempo real.
REST	<i>REpresentational State Transfer</i> , Significa Transferência de Estado Representacional.
SOAP	<i>Simple Object Access Protocol</i> , significa Protocolo de Acesso a Objetos Simples.
Websocket	É um soquete full-duplex usado para se comunicar entre servidor e cliente.
RAML	<i>RESTful API Modelling Language</i> . Usado para projetar e compartilhar a API.
IoTivity	Fundada pela Linux Foundation para facilitar projetos de código aberto e patrocinada pela OIC.
IEEE P2413	É um padrão para uma estrutura arquitetônica para IoT.
OTrP	<i>Open Trust Protocol</i> , é usado para instalar, atualizar e excluir aplicativos. Gerencia a configuração de segurança em um TEE (<i>Trusted Execution Environment</i>)

Fonte: Tradução nossa de [26]

Por fim, é relevante destacar algumas outras tecnologias que oferecem suporte à Internet das Coisas. A rede de comunicação pode abranger diferentes tipos, não se limitando a uma única mídia ou dispositivo [26]. O Quadro 3.6 apresenta essas tecnologias, categorizando-as em grupos com base no alcance da comunicação. Os grupos 1, 2 e 3 abordam as tecnologias sem fio de curto, médio e longo alcance, respectivamente. Enquanto isso, o quarto grupo descreve tecnologias cabeadas.

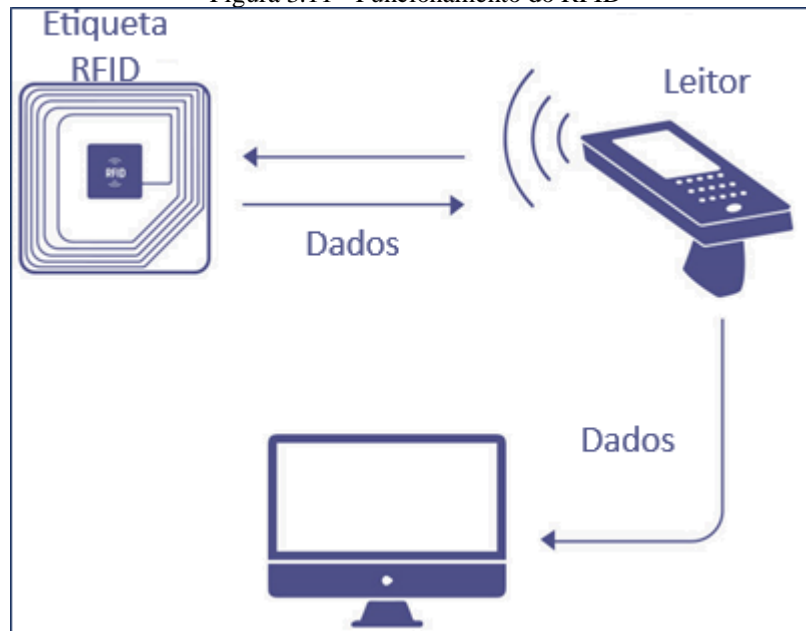
Quadro 3.6 - Tecnologia wireless de curto alcance

Tecnologia	Descrição	Grupo
Rede Mesh Bluetooth	Rede <i>mesh</i> compatível com Bluetooth de baixa energia (BLE) com um número maior de pontos.	1
Li-Fi	<i>Light-Fidelity</i> , semelhante ao Wi-Fi, mas usa o espectro de luz visível.	1
NFC	<i>Near-field Communication</i> , Comunicação de campo próximo. Protocolo que permite a comunicação entre dois dispositivos dentro de um alcance de 4 cm.	1
Códigos QR e de barras	Etiqueta óptica que pode ser lida por máquina, armazena as informações do item no qual está colada.	1
RFID	<i>Radio-Frequency Identification</i> , Identificação por radiofrequência Utiliza campos eletromagnéticos para ler as informações armazenadas em <i>tags</i> nos demais itens.	1
Thread	Protocolo de rede é baseado no padrão IEEE 802.15.4.	1
Wi-Fi	Para rede local, que é baseada no padrão IEEE 802.11.	1
Z-Wave	Protocolo de comunicação de baixa potência e latência e curto alcance. Melhor confiabilidade do que o Wi-Fi.	1
ZigBee	Protocolo que pode ser usado para uma rede de área pessoal, é baseado no padrão IEEE802.15.4.	1
HaLow	É a variante do padrão Wi-Fi. Fornece transmissão de baixa taxa de dados em uma ampla faixa.	2
LTE-Advanced	<i>Long-Term Evolution</i> , Tecnologia de evolução de longo prazo destinada a fornecer comunicação sem falha com alta taxa de dados.	2
LPWAN	<i>Low-Power Wide-Area Networking</i> , Rede de Longo Alcance de Baixa Potência. Rede sem fio que facilita uma ampla gama de comunicação, juntamente com baixa taxa de bits e menos energia.	3
VSAT	<i>Very Small Aperture Terminal</i> , Terminal de abertura muito pequena. Comunicação é usada em satélites que usam antena parabólica para dados de banda estreita.	3
Ethernet	Técnica de comunicação com fio que utiliza par trançado e fibra óptica com hubs ou switches.	4
MoCA	<i>Multimedia over Coax Alliance</i> , Aliança Multimídia sobre Coaxial. Tecnologia que melhora a qualidade de vídeo sobre o cabo existente.	4
PLC	<i>Power-line Communication</i> , Comunicação por linha de energia. Tecnologia de comunicação que utiliza a transmissão de energia elétrica e dados.	4

Fonte: Tradução nossa de [26]

Em destaque, cada vez mais presente nas tecnologias voltadas a conferências e controle de estoque, bem como em cartões de trabalho, a identificação por radiofrequência – RFID (do inglês, *Radiofrequency Identification* é uma tecnologia composta por duas partes principais: um emissor e um receptor. Essas duas partes são representadas na Figura 3.11 pelo leitor e pela etiqueta RFID, respectivamente.

Figura 3.11 - Funcionamento do RFID



Fonte: <https://mapsted.com/blog/rfid-positioning-explained>

Na unidade emissora, uma antena gera ondas em uma frequência específica, variando de 30 kHz a 3 GHz, e pode alcançar distâncias que vão de apenas alguns centímetros a dezenas de metros [28].

Na unidade receptora, há uma antena com a finalidade de ressoar na mesma frequência da emissora. Essa ressonância permite que o circuito gere energia elétrica para alimentar o chip integrado à etiqueta RFID, que contém informações. Essas informações podem ser lidas por um leitor e transmitidas para um computador. A variedade de informações vai desde uma simples identificação de 64 bits até dados mais complexos, como estruturas de 32 KB, por exemplo [29].

4 METODOLOGIA

Neste capítulo, é exposta a metodologia empregada para atingir os objetivos estabelecidos neste trabalho. Isso abrange a consideração dos requisitos tanto do usuário quanto do sistema, seguida da elaboração de um modelo para o sistema com base nesses requisitos. Posteriormente, apresenta-se o fluxograma de funcionamento das aplicações, fornecendo uma representação visual do processo. Por fim, são delineados os materiais utilizados durante o desenvolvimento do estudo.

4.1 REQUISITOS

De acordo com Sommerville [30], entende-se por requisitos de um sistema tudo o que este deve fazer, mais especificamente os serviços oferecidos e as devidas restrições. O autor ressalta que os requisitos podem ser separados entre requisitos de usuário, em linguagem natural e de maneira mais genérica, e os requisitos de sistema, que descrevem de maneira específica e tendem a ter uma relação “um para muitos” dos requisitos de usuário. Neste trabalho o cliente solicitou os requisitos de acordo com o Quadro 4.1.

Quadro 4.1 - Requisitos de usuário e sistema

Tipo	Requisito	Descrição
Usuário	1	Um aparelho que registre, de maneira automática, quando um funcionário entrar em um determinado ambiente juntamente com os horários de entrada e saída.
	2	Uma maneira de gerar relatórios diários de quanto tempo cada funcionário passou neste ambiente.
	3	Um dispositivo que consiga ser colocado em áreas mais afastadas, sem acesso a redes sem fio.
Sistema	1	Não ser intrusivo ou complexo, ou seja, não atrapalhe ou interfira no processo ou trabalho dos operadores.
	2	Tamanho reduzido.
	3	Capturar as informações do usuário.
	4	Velocidade de processamento das informações.
	5	Sinais visuais e sonoros de indicação.
	6	Comunicação em áreas mais afastadas.
	7	Possibilidade de funcionamento mesmo em situações de ausência de sinal e que volte a operar depois que a comunicação for restabelecida.
	8	Facilidade na implantação, utilização e na manutenção.
	9	Custo reduzido comparado a soluções de mercado.

Fonte: Coletado pelo autor

4.2 MODELAGEM DO SISTEMA

Como ideia inicial, foi proposto o uso de cartões de identificação dos funcionários, que também fornecem acesso à empresa, equipados com chips de tecnologia RFID. A principal unidade de processamento foi concebida para utilizar tecnologia de sistemas embarcados, com foco especial em SBCs, devido ao seu tamanho compacto em comparação com computadores de mesa convencionais. A Figura 4.1 ilustra o tamanho de um SBC, destacado em amarelo no canto inferior direito, em comparação com um computador portátil de tela 15,6 polegadas e um smartphone de 6,6 polegadas.

Figura 4.1 - Tamanho do SBC em Comparação



Fonte: Acervo do Autor

Em termos de custos, um SBC, como o modelo de entrada da família 4B mostrado na Figura 4.1, pode ser adquirido por a partir de US\$ 35.00 [4]. Por outro lado, um smartphone, que atenda às especificações mínimas para realizar as mesmas tarefas, pode custar várias vezes mais. O smartphone mostrado na Figura 4.1, homologado para uso na empresa, tinha um custo inicial de US\$ 259.99 no momento de seu lançamento [31].

4.3 MATERIAIS UTILIZADOS

Todo o hardware utilizado neste projeto está detalhado na Tabela 4.1 e com mais informações nas subseções subsequentes:

Tabela 4.1 - Materiais Utilizados

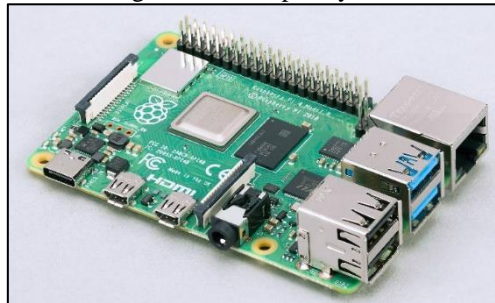
Item	Quantidade Unitária
Raspberry Pi 4 model B 4GB	1
Cartão de memória de 32GB	1
Fonte chaveada de 5 Volts e 3 Ampères	1
Dissipadores de calor	4
Tela tátil de 3,5 polegadas	1
Estojo de proteção	1
Leitor RFID USB	1
Par de caixas de som de 3 Watts RMS	1
Modem roteador WiFi 4G USB	1

Fonte: Elaborado pelo autor

4.3.1 Computador

O principal componente, e o cerne deste trabalho, é o computador de placa única Raspberry Pi 4B. Esta placa possui um processador ARM de 4 núcleos operando a 1,5 GHz e memória RAM de 4 gigabytes DDR4 2400MHz. Este modelo não possui armazenamento próprio, sendo necessária a utilização de armazenamento externo, um cartão Micro-SD de no mínimo 16GB. Esta placa deve ser alimentada por uma fonte de no mínimo 15W. A Figura 4.2 mostra o modelo de Raspberry Pi usado neste protótipo.

Figura 4.2 - Raspberry Pi 4



Fonte: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

4.3.2 Leitor de Dados

O circuito integrado MFRC522 que se conecta à GPIO (do inglês, General Purpose Input/Output, entrada e saída de propósito geral) do Raspberry Pi tornaria o desenvolvimento mais demorado, pois requer desenvolvimento específico de código para controlá-lo e obter os dados presentes nos cartões. Não obstante, haveria a necessidade de enclausurar este circuito, bem como adicionar componentes e códigos de programação para emissão de luz e som indicadores.

De forma a facilitar a aquisição dos dados e a manutenção do aparelho, optou-se por usar um leitor de RFID com conexão USB, mostrado na Figura 4.3.

Figura 4.3 - Leitor RFID USB



Fonte: <https://produto.mercadolivre.com.br/MLB-1253309191>

Este modelo, embora apresente um custo mais elevado em comparação com os componentes mencionados anteriormente, opera como um teclado USB. Ao aproximar o cartão RFID, o identificador associado a esse cartão será exibido na tela. Além de contar com proteção adequada e emitir sinais sonoros e luminosos quando o cartão é aproximado, sua implementação e manutenção são consideravelmente mais simples, oferecendo facilidade de substituição em caso de falhas.

4.3.3 Estojo

O estojo utilizado para proteger e encapsular o Raspberry Pi apresenta dissipadores de calor e uma tela tátil de 3,5 polegadas, com resolução de 480x320 pixels, os quais podem ser vistos na Figura 4.4. Essa tela desempenha o papel de interface humano-máquina, exibindo informações como o número de identificação do crachá do usuário, data, horário, se está entrando ou saindo, e a quantidade de vezes que entrou ou saiu do recinto.

Figura 4.4 - Tela (1), Dissipadores (2) e Estojo (3).



Fonte: <https://www.aliexpress.us/item/4000587542368.html>

Este conjunto também inclui uma caneta *stylus*, que poderia ser usada para interagir com a tela tátil. Entretanto, a única interação que deverá haver é em relação à aproximação do crachá do usuário ao leitor.

4.3.4 Comunicação com a Internet

Considerando que a área onde o estudo foi conduzido é remota e os protocolos de segurança impedem a conexão de dispositivos não homologados à rede, torna-se necessário o uso de comunicação por dados móveis. Um modem roteador 4G, ilustrado na Figura 4.5, é empregado para viabilizar essa comunicação.

Figura 4.5 - Modem 4G



Fonte: <https://www.amazon.com.br/gp/product/B09NN3ZWLX/>

Dado que não há necessidade de grande tráfego de dados advindo do SBC nem de alta taxa de transferência para esta aplicação, a quarta geração de internet móvel é suficiente para o seu correto funcionamento.

4.3.5 Serviço de Armazenamento em Nuvem

Para armazenar os dados referentes às entradas e saídas, foi escolhida a plataforma Azure da Microsoft. Esta plataforma conta com um banco de dados em nuvem, gerenciamento de múltiplas Interfaces de Programação de Aplicação – APIs para troca de dados, alta disponibilidade, segurança da informação, entre outras vantagens [32].

4.3.6 Desenvolvimento do Software

Para este trabalho, foi escolhida a linguagem de programação C# usando o framework Dotnet Core devido à sua facilidade de integração com os serviços do Azure, visto que ambos são de autoria da Microsoft.

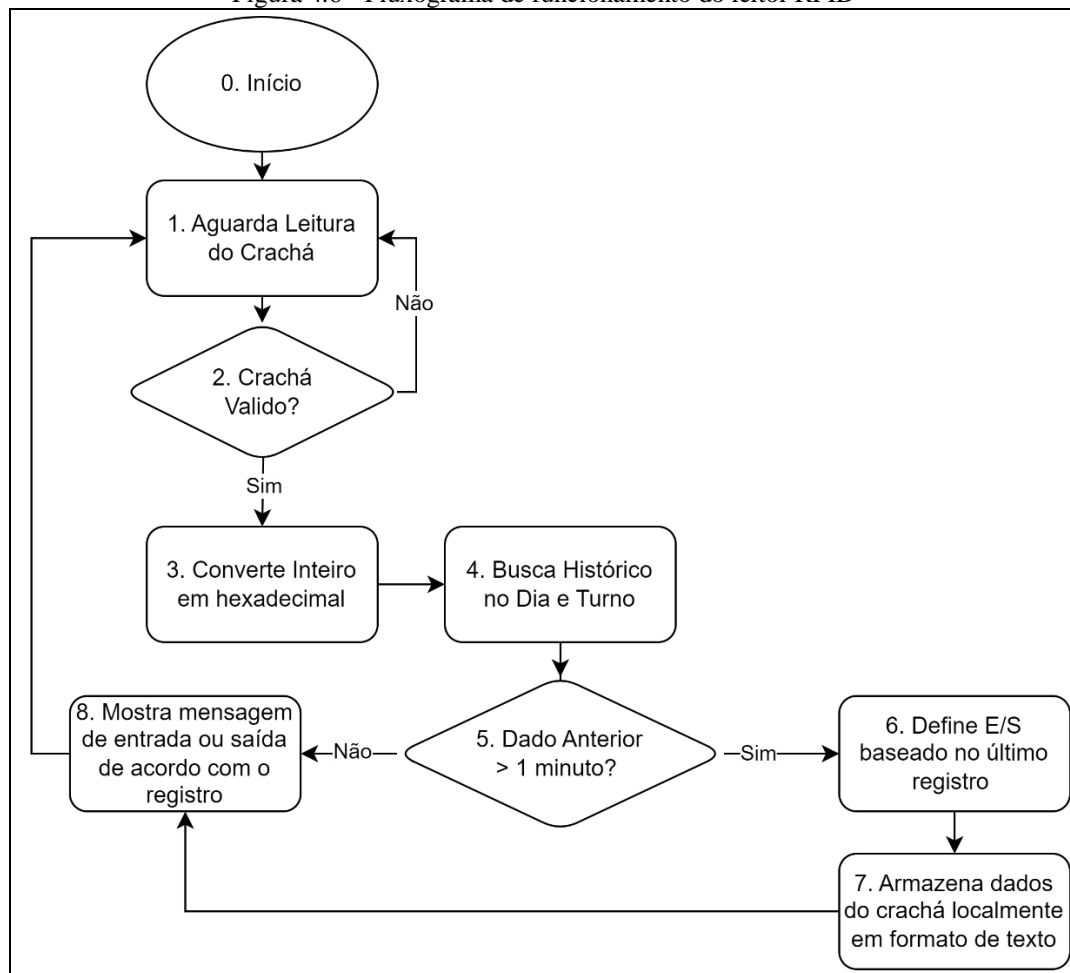
4.4 FUNCIONAMENTO

A solução de software foi dividida em duas partes. A primeira é uma aplicação em primeiro plano e tem como principal funcionalidade a aquisição e armazenamento dos dados. A segunda é um serviço em segundo plano para envio e sincronização dos dados armazenados localmente.

4.4.1 Aquisição e Armazenamento

A aplicação, que lê os dados de identificação presentes no cartão RFID e salva um arquivo de texto por leitura realizada, pode ser expressa em oito etapas, conforme a Figura 4.6.

Figura 4.6 - Fluxograma de funcionamento do leitor RFID



Fonte: Elaborado pelo autor em draw.io

As etapas de funcionamento deste sistema são descritas no Quadro 4.2 e trazem detalhes de cada um dos itens foram expostos na Figura 4.6 e devem ser analisadas como complementares.

Quadro 4.2 - Etapas do processo de aquisição de dados

Etapa	Descrição
0	Definido como o início do processo, logo após o SBC ser iniciado e a aplicação carregada automaticamente.
1	A aplicação imprime a informação sobre a utilização na tela e fica em espera até que o usuário aproxime seu crachá;
2	A aplicação valida se o código apresentado é um número inteiro;
3	Se o número for válido, é convertido para hexadecimal, senão, retorna ao passo anterior;
4	Busca-se no histórico na pasta do dia e turno (“/Logs/[YYYY-MM-DD]/[A B C]”) se existe algum registro anterior deste usuário. Onde [YYYY-MM-DD] representa os oito dígitos das datas, onde “YYYY” são os dígitos do ano, “MM” os dois dígitos do mês e “DD” os dois dígitos do dia em que o programa está funcionando, e “[A B C]” representa o turno de operação da empresa que opera 24 horas por dia, podem ser “A”, “B” ou “C”;
5	Por questões de segurança contra possível registro em duplicidade, no caso de um colaborador erroneamente aproximar o seu crachá duas vezes em um curto período, o dado do último registro deve ser superior a um minuto. Se for superior a este tempo, o sistema prosseguirá normalmente para a etapa 6, senão avançará para a etapa 8;
6	Baseado no último registro, será registrada saída se o último registro for de entrada, e vice-versa. A cada novo registro de entrada, um contador registra a ordem de entradas e saídas.
7	Um arquivo é salvo na mesma pasta do histórico mencionada na etapa 4, com o padrão de título “[ID].[Contagem].[Horário].[Tipo]”, por exemplo “01-23-AB-CD.005.13-05-59.CHECKIN”, onde: <ul style="list-style-type: none"> [ID] é o identificador do cartão RFID, convertido em hexadecimal de 16 bits, separados por um hífen a cada quatro bits, por exemplo “01-23-AB-CD”; [Contagem] é o padrão utilizado para saber a quantidade de vezes que o usuário entrou e saiu daquele ambiente. É registrado com três dígitos, variando de 001 a 999; [Horário] é o horário deste registro. É formatado no padrão “HH-mm-ss”, onde “HH” são os dois dígitos a hora no formato de 24 horas, “mm” são os dois dígitos dos minutos e “ss” são os dois dígitos dos segundos. Por exemplo, “13-05-59”, seria um registro feito às 13 horas, 5 minutos e 59 segundos; [Tipo] é o tipo de registro e a extensão do arquivo. Se for de entrada, é “CHECKIN”, senão, “CHECKOUT”; O conteúdo do arquivo é um JSON que possui quatro campos: Número de ID, data completa de registro, tipo de registro e identificação do aparelho;
8	O sistema imprime na tela o registro de entrada e saída mais recente, sendo fundo verde para entrada e fundo azul para saída. O sistema também emite sonoramente a mensagem relativa ao registro, por exemplo “segunda saída registrada”. Logo após, volta à etapa 1.

Fonte: Elaborado pelo autor

O sistema foi concebido para operar de forma ininterrupta, em um laço infinito. Assim, não há um término predeterminado, uma vez que o sistema retornará à etapa inicial após exibir as informações do registro na tela.

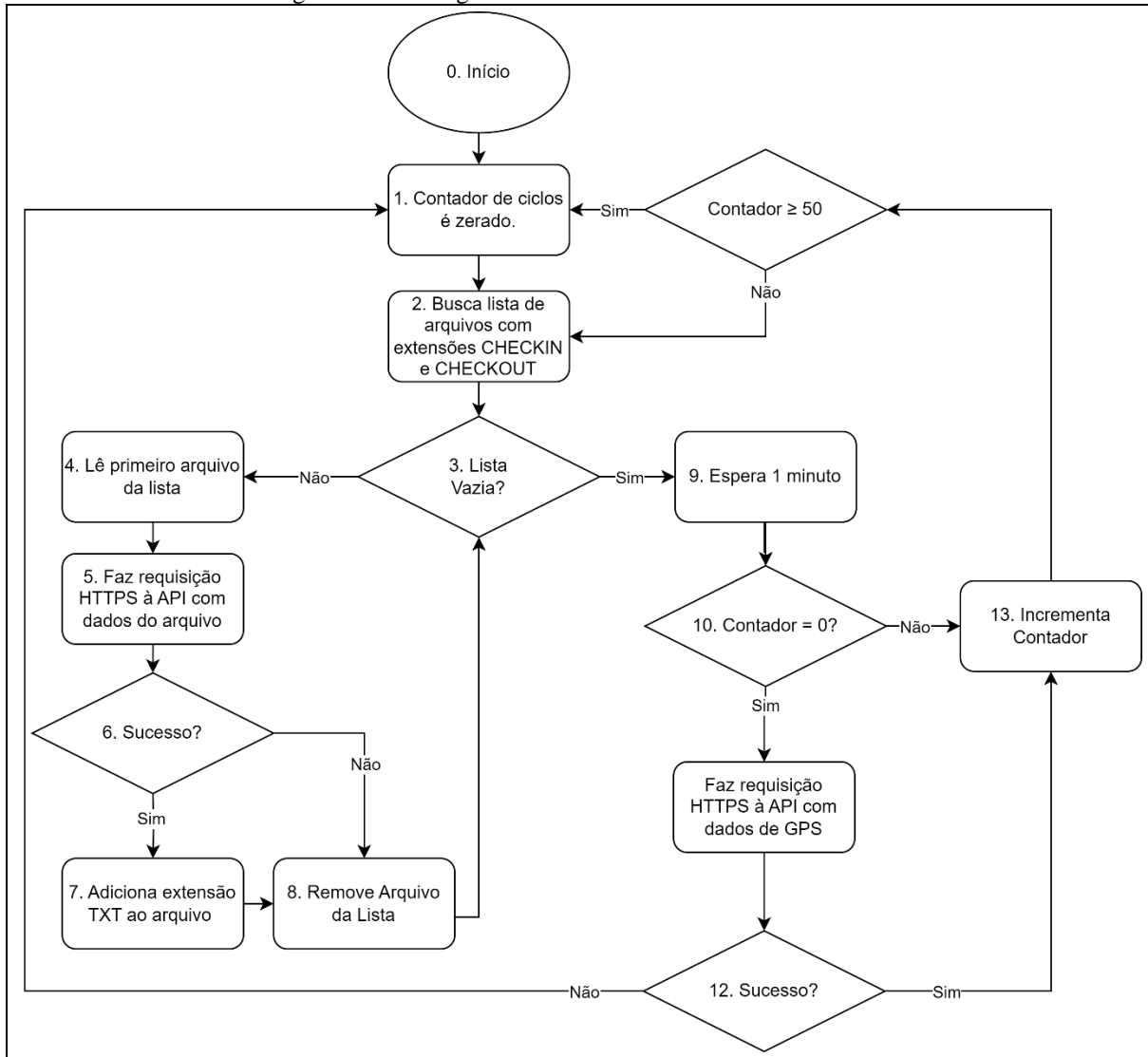
4.4.2 Envio e Sincronização de Dados

A segunda parte da solução é um serviço em segundo plano responsável por enviar as informações armazenadas no dispositivo para a nuvem. Existem dois tipos de informações a serem enviadas: a primeira refere-se às informações coletadas dos crachás dos funcionários, e a segunda diz respeito aos dados de funcionamento do dispositivo.

Os dois grupos de informações são processados de maneira serial, e suas etapas estão entrelaçadas, pois a segunda é de baixa prioridade. A primeira tarefa será executada na proporção de um para cinquenta em relação à segunda e terá um intervalo de espera de 1 minuto entre os processamentos.

A Figura 4.7 mostra em 14 etapas o seu funcionamento, sendo oito etapas para a sincronização de informações e seis para o registro de operação normal do aparelho.

Figura 4.7 - Fluxograma de funcionamento do sincronizador



Fonte: Elaborado pelo autor em draw.io

4.4.2.1 Envio de Informações do Usuário

O processo inicia logo após o sistema operacional iniciar e segue primeiramente as etapas referentes à sincronização dos arquivos previamente armazenados. Estas etapas são descritas no Quadro 4.3.

Quadro 4.3 - Sincronização das informações do usuário

Etapa	Descrição
0	Início do processo;
1	É atribuído o valor zero ao contador de ciclos, que servirá para indicar o momento para enviar os dados de funcionamento;
2	O serviço acessa a pasta do dia e do turno e busca todos os arquivos que não foram processados, considerando apenas aqueles com extensões CHECKIN e CHECKOUT. Outros arquivos são ignorados;
3	Se houver pelo menos um arquivo não processado na lista, avança para a etapa 4; caso contrário, avança para a etapa 9, pois não há arquivos para processar;
4	O serviço recebe todo o texto do primeiro arquivo e o atribui a uma variável do tipo JSON;
5	A informação coletada na etapa 4 é utilizada para fazer uma requisição HTTPS à API do Azure;
6	Se a informação for recebida pela nuvem com sucesso, avança para a etapa 7; caso contrário, avança para a etapa 8 para processamento na próxima iteração;
7	A extensão ".TXT" é adicionada para que o arquivo não seja incluído na próxima busca por arquivos e para facilitar a leitura por um ser humano;
8	Para evitar duplicidade de sincronização, o arquivo é removido da lista de processamento; o sistema volta à etapa 3 até que todos os arquivos da lista sejam analisados com tentativas de envio das informações à nuvem;
9	Após processar todos os arquivos, o sistema aguarda um minuto antes de prosseguir;

Fonte: Elaborado pelo autor

4.4.2.2 Envio das Informações de Funcionamento

Logo após terminar a espera do primeiro grupo, o sistema entrará no segundo. Ainda fazendo referência à Figura 4.7, a segunda parte da aplicação pode ser descrita em 6 etapas, as quais são:

Quadro 4.4 - Sincronização de funcionamento

Etapa	Descrição
10	Será realizado um teste para verificar se o contador de ciclos está zerado. Em caso afirmativo, o sistema prossegue para a etapa 11; caso contrário, vai para a etapa 13;
11	É feita uma requisição HTTPS à API do Azure com os dados pré-configurados daquele aparelho, incluindo o identificador do aparelho, as coordenadas de latitude e longitude do GPS, bem como a data e hora da requisição;
12	Se a requisição for bem-sucedida, o fluxo avança para a etapa 13; caso contrário, o programa retorna ao início da primeira aplicação com o contador zerado, permitindo uma nova tentativa após processar todos os registros de entrada e saída
13	O contador de ciclos é incrementado para que, no próximo ciclo, não haja nova tentativa;
14	Se o contador ultrapassar 50 ciclos, ele é zerado para uma nova atualização; caso contrário, a etapa de zerar o contador é ignorada;

Fonte: Elaborado pelo autor

Portanto, ocorrerá uma proporção de um para 50 entre a quantidade de vezes em que o envio de informações sobre o funcionamento do sistema acontecerá em relação ao envio das informações dos usuários.

5 RESULTADOS

Os resultados delineados nesta seção estão subdivididos em duas partes distintas. A primeira parte abrange os testes efetuados em campo, enquanto a segunda, e última, expõe algumas métricas obtidas ao longo desses procedimentos experimentais.

5.1 TESTES COM O USUÁRIO

Foram conduzidos dois ensaios com o propósito de verificar a operacionalidade da solução e avaliar sua capacidade de atendimento às demandas do cliente. O primeiro teste abrangeu a avaliação do funcionamento, desde a interação na interface do usuário até a etapa de aquisição e sincronização de dados. No segundo teste, foi realizada uma simulação de uso real, acompanhando os usuários em suas atividades cotidianas.

5.1.1 Testes de Leitura e Interação

Inicialmente, o dispositivo foi ativado, aguardando-se até que a mensagem solicitando que o usuário aproximasse o seu crachá do leitor fosse exibida na tela, conforme exemplificado na Figura 5.1. Nessa representação visual, observa-se que a luz do leitor está acesa na cor laranja, indicando que este se encontra em estado de espera.

Figura 5.1 - Tela inicial da aplicação

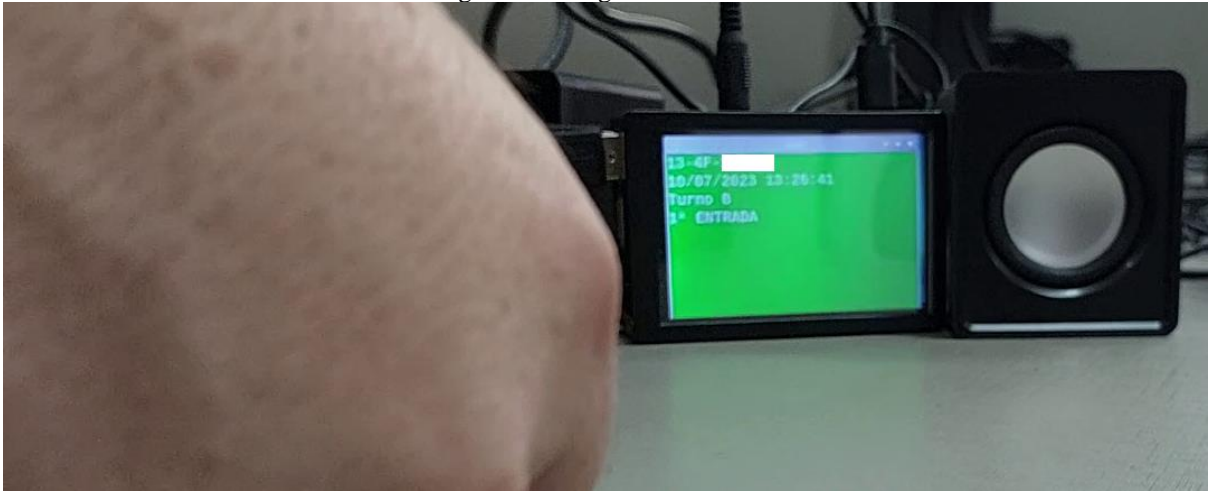


Fonte: Acervo do autor

Imediatamente após a aproximação do crachá pelo usuário, um sinal sonoro é emitido pelo leitor, acompanhado por uma transição momentânea da luz indicativa de laranja para verde, indicando a efetivação da leitura. Após alguns milissegundos, o fundo da tela se tornará

verde, denotando uma entrada registrada, seguida pela projeção sonora da mensagem “primeira entrada registrada” através dos alto-falantes. Além disso, serão exibidas as informações pertinentes ao crachá, data, hora e turno. No contexto específico da Figura 5.2, , destaca-se que esta é a primeira vez em que o usuário está aproximando o crachá. Após meio segundo, a tela retorna à exibição da mensagem representada na Figura 5.1.

Figura 5.2 - Registro de entrada



Fonte: Acervo do autor

Conforme evidenciado na Figura 5.3, observa-se que um usuário, distinto do antecessor, está realizando a aproximação do crachá junto ao dispositivo pela terceira vez. Nessa situação, o fundo da tela assume a coloração azul, enquanto a mensagem “terceira saída registrada” é anunciada, exibindo o padrão equivalente de informações, comparável ao registrado durante as entradas.

Figura 5.3 - Registro de saída



Fonte: Acervo do autor

5.1.2 Armazenamento e Sincronização

Após a conclusão dos testes, o dispositivo foi desligado e o cartão de memória foi removido para ser lido em um computador externo. A partição do cartão foi acessada e uma relação de todos os arquivos contidos na pasta referente ao dia e ao turno em questão foi extraída, ordenada por nome. A Figura 5.4, apresenta uma visão resumida dessa lista, cujo comprimento foi reduzido para se adequar a este trabalho. A relação completa encontra-se disponível no APÊNDICE A – LISTAGEM COMPLETA DOS REGISTROS DE TESTE.

Figura 5.4 - Listagem dos registros armazenados localmente

```
PS E:\Logs\2023-07-10\B> dir | Sort-Object Name
```

Diretório: E:\Logs\2023-07-10\B

Mode	LastWriteTime	Length	Name
-a----	10/07/2023 15:05	131	02-C1 .001.12-05-33.CHECKIN.txt
-a----	10/07/2023 15:08	132	02-C1 .001.12-08-49.CHECKOUT.txt
-a----	10/07/2023 12:18	131	10-93 .001.09-18-21.CHECKIN.txt
-a----	10/07/2023 12:18	131	13-4F .001.09-18-15.CHECKIN.txt
-a----	10/07/2023 12:50	132	13-4F .001.09-50-00.CHECKOUT.txt
-a----	10/07/2023 12:51	131	13-4F .002.09-51-01.CHECKIN.txt
-a----	10/07/2023 14:15	132	13-4F .002.11-15-50.CHECKOUT.txt
-a----	10/07/2023 15:09	131	13-4F .003.12-09-38.CHECKIN.txt
-a----	10/07/2023 16:25	132	13-4F .003.13-25-24.CHECKOUT.txt
-a----	10/07/2023 16:26	131	13-4F .001.13-26-41.CHECKIN.txt
-a----	10/07/2023 16:29	132	13-4F .001.13-29-15.CHECKOUT.txt
-a----	10/07/2023 14:25	131	13-51 .001.11-25-10.CHECKIN.txt
-a----	10/07/2023 15:04	132	13-51 .001.12-04-42.CHECKOUT.txt
-a----	10/07/2023 12:58	131	4B-75 .001.09-58-46.CHECKIN.txt
-a----	10/07/2023 13:12	132	4B-75 .001.10-12-25.CHECKOUT.txt
-a----	10/07/2023 12:42	131	4B-7C .001.09-42-41.CHECKIN.txt
-a----	10/07/2023 12:44	132	4B-7C .001.09-44-52.CHECKOUT.txt
-a----	10/07/2023 12:18	131	53-15 .001.09-18-07.CHECKIN.txt
-a----	10/07/2023 12:19	132	53-15 .001.09-19-22.CHECKOUT.txt
-a----	10/07/2023 15:05	131	53-15 .002.12-05-51.CHECKIN.txt
-a----	10/07/2023 15:08	132	53-15 .002.12-08-31.CHECKOUT.txt
-a----	10/07/2023 15:00	131	6F-52 .001.12-00-03.CHECKIN.txt
-a----	10/07/2023 15:09	132	6F-52 .001.12-09-28.CHECKOUT.txt
-a----	10/07/2023 12:18	131	9A-4F .001.09-18-36.CHECKIN.txt

Fonte: Acervo do autor

Para preservar a confidencialidade da identificação do usuário e evitar qualquer uso indevido ou transmissão não autorizada de informações pessoais, uma prática adotada neste trabalho consiste na omissão parcial do identificador. Nesse sentido, os quatro últimos dígitos são substituídos por espaços ou asteriscos. Além disso, ao analisar a imagem, observa-se que os arquivos possuem a extensão "TXT", indicando que foram processados e sincronizados com o banco de dados. Essa extensão também facilita a leitura manual dos arquivos.

A fim de realizar uma contraprova e comparar os dados registrados no banco de dados, uma consulta foi executada para recuperar todos os registros datados em 10 de julho de 2023 e originados pelo dispositivo objeto deste estudo.

Com o intuito de facilitar a análise dos dados, a Tabela 5.1 contendo informações consolidadas e agrupadas pelo identificador do cartão, foi exportada do relatório paginado do Power BI. Além disso, é possível estabelecer uma correlação entre os dados apresentados na Tabela 5.1 e as informações visualizadas nas Figura 5.2, Figura 5.3 e Figura 5.4, sendo esperado que todas essas fontes de informação compartilhem os mesmos dados.

Tabela 5.1 - Dados resumidos por usuário

Número de identificação	Entrada	Saída
4B-7C-**-**	10/07/2023 09:42:42	10/07/2023 09:44:53
A6-C7-**-**	10/07/2023 14:23:00	10/07/2023 14:24:34
13-4F-**-**	10/07/2023 09:18:15	10/07/2023 09:50:00
	10/07/2023 09:51:02	10/07/2023 11:15:50
	10/07/2023 12:09:38	10/07/2023 13:25:24
13-4F-**-**	10/07/2023 13:26:42	10/07/2023 13:29:16
BA-DE-**-**	10/07/2023 09:22:26	10/07/2023 09:26:01
BA-CE-**-**	10/07/2023 09:46:16	10/07/2023 09:49:33
	10/07/2023 09:51:21	10/07/2023 12:05:28
	10/07/2023 12:08:42	10/07/2023 13:26:45
9C-F4-**-**	10/07/2023 14:22:45	10/07/2023 14:24:27
4B-75-**-**	10/07/2023 09:58:46	10/07/2023 10:12:25
02-C1-**-**	10/07/2023 12:05:34	10/07/2023 12:08:49
13-51-**-**	10/07/2023 11:25:11	10/07/2023 12:04:42
6F-52-**-**	10/07/2023 12:00:03	10/07/2023 12:09:29
53-15-**-**	10/07/2023 09:18:08	10/07/2023 09:19:23
	10/07/2023 12:05:51	10/07/2023 12:08:32
F5-4F-**-**	10/07/2023 09:31:18	10/07/2023 09:34:40

Fonte: Coletado e montado pelo autor

5.1.3 Status de Funcionamento

Exclusivamente para os testes em questão, o contador inicial não foi configurado como zero, uma medida adotada para evitar o envio excessivo de sinais de funcionamento sempre que o sistema fosse inicializado, o que poderia interferir nos resultados obtidos.

Com o propósito de preservar a confidencialidade da identidade da empresa onde os testes foram conduzidos, as coordenadas de latitude e longitude foram mascaradas nos dados, uma vez que essas informações correspondem à localização física da empresa. Essa localização facilitaria a manutenção caso o aparelho deixasse de enviar atualizações por um período pré-determinado.

Durante o teste inicial, o dispositivo teve que ser desligado e armazenado por um período ligeiramente superior a uma hora, para a pausa do almoço e para o restabelecimento do

sistema, visto que não havia uma pessoa designada para monitorar as leituras durante esse intervalo. Apesar das circunstâncias, o aparelho conseguiu enviar três atualizações com sucesso durante o seu funcionamento, conforme evidenciado pela Tabela 5.2.

Tabela 5.2 - Dados de Funcionamento Normal

ID	ID Aparelho	Latitude	Longitude	Data
13403541	270	-2.*****	-44.*****	2023-07-10 10:22:47.367
13403889	270	-2.*****	-44.*****	2023-07-10 11:22:51.287
13404490	270	-2.*****	-44.*****	2023-07-10 14:27:31.860

Fonte: Coletado e montado pelo autor

É perceptível que o serviço de envio de atualizações de status esteve operacional por aproximadamente uma hora desde o início, com o primeiro registro do dia registrado às 09:18:08, conforme apresentado na Tabela 5.1. O segundo registro de status ocorreu aproximadamente uma hora após o primeiro, e o terceiro foi registrado também uma hora após o primeiro, logo após o horário de almoço. Essa observação ressalta uma boa aproximação à premissa de executar a rotina a cada 50 ciclos.

5.2 MÉTRICAS

A fim de agregar maior valor e apresentar informações adicionais pertinentes, são fornecidas métricas adicionais além dos resultados, destacando os tempos necessários para que o sistema esteja pronto para uso.

5.2.1 Tempo de Preparação

O tempo total para que uma nova unidade deste sistema entre em operação dependerá de fatores internos e externos. Considerando os fatores internos, pressupondo que todos os equipamentos e periféricos estejam disponíveis em estoque, podemos afirmar que o tempo médio gasto nas atividades a seguir seria conforme apresentado no Quadro 5.1.

Quadro 5.1 - Tempo Estimado de Preparação

Passo	Atividade	Tempo estimado
1	Baixar a imagem no cartão de memória	25 a 30 minutos
2	Configurar a unidade	5 a 7 minutos
3	Montagem do SBC no estojo	5 a 7 minutos
4	Conexão dos periféricos	2 a 4 minutos
5	Instalação da estrutura no local de uso	30 a 40 minutos

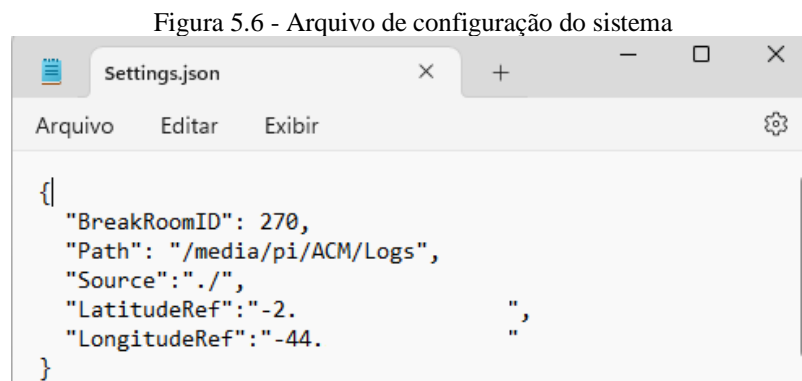
Fonte: Elaborado pelo autor

Ressalta-se que o passo um do Quadro 5.1 pode ser realizado concomitantemente aos passos 3, 4 e 5, pois são etapas físicas, enquanto as duas primeiras são etapas lógicas. Em outras palavras, todo o processo pode ser concluído em menos de uma hora. No canto inferior direito da Figura 5.5 é apresentado o tempo estimado pelo programa *Balena Etcher*[®] para replicar fielmente a imagem do disco na memória de armazenamento do SBC.



Fonte: Software *Balena Etcher*[®]

Essa cópia possibilita que a versão otimizada do programa, bem como todas as características de uso, estejam presentes em todos os dispositivos. Portanto, não há a necessidade de formatar o cartão de memória, realizar a instalação e configuração do sistema operacional, ou ainda, instalar o software separadamente. A única alteração necessária refere-se ao arquivo de configuração do dispositivo, conforme exemplificado na Figura 5.6.



Fonte: Elaborado pelo autor

Existem cinco campos que precisam ser preenchidos para assegurar o funcionamento adequado do sistema, conforme detalhados no Quadro 5.2.

Quadro 5.2 - Campos de Configuração

Item	Campo	Descrição
1	<i>BreakroomID</i>	Número identificador do equipamento a ser instalado. Cada dispositivo deve ter um registro único.
2	<i>Path</i>	Local onde os arquivos locais de registro devem ser salvos – Padrão para um SBC.
3	<i>Source</i>	Local onde estão os arquivos-fonte, como os de áudio usados nos avisos de entrada e saída – Padrão para um SBC.
4	<i>LatitudeRef</i>	Dado de latitude de GPS onde o equipamento será instalado
5	<i>LongitudeRef</i>	Dado de longitude de GPS onde o equipamento será instalado

Fonte: Elaborado pelo autor

Os dados das posições geográficas, adquiridos por meio de busca em GPS, têm o potencial de facilitar significativamente a manutenção de dispositivos com falhas, pois proporcionam a localização exata quando configurados de maneira adequada.

5.2.2 Mapeamento de manutenção

Considerando a disponibilidade de peças sobressalentes para substituir componentes eventualmente falhos, o tempo necessário conforme o Quadro 5.1, seria significativamente reduzido. Isso se deve ao pressuposto de que a maior parte da estrutura já estaria montada previamente, e seria necessário apenas substituir os elementos que não estiverem funcionando corretamente. O Quadro 5.3 destaca os principais problemas que surgiram durante o desenvolvimento e teste, juntamente com suas possíveis soluções.

Quadro 5.3 - Principais problemas e soluções

Item	Problema	Soluções
Leitor	Os dados do crachá não são capturados.	Verificar conexão; Trocar porta USB; Substituir o leitor.
Tela	Tela não mostra informações.	Verificar conexão; Substituir a tela; Substituir o cartão de memória; Substituir o SBC.
Dados Móveis	Não há envio de dados de usuário nem de posição por mais de duas horas.	Testar conexão do Modem; Substituir o Modem.
Caixas de Som	Som não é emitido após aproximação do crachá.	Verificar conexão e volume da caixa de som; Substituir a caixa de som; Reiniciar SBC; Substituir SBC.
Fonte	O aparelho não liga.	Verificar conexão de energia; Substituir fonte de alimentação; Substituir o SBC.
SBC	A luz de energia ou processamento não acendem.	Reiniciar o SBC; Testar a fonte; Substituir a fonte; Substituir o Cartão de Memória; Substituir o SBC.

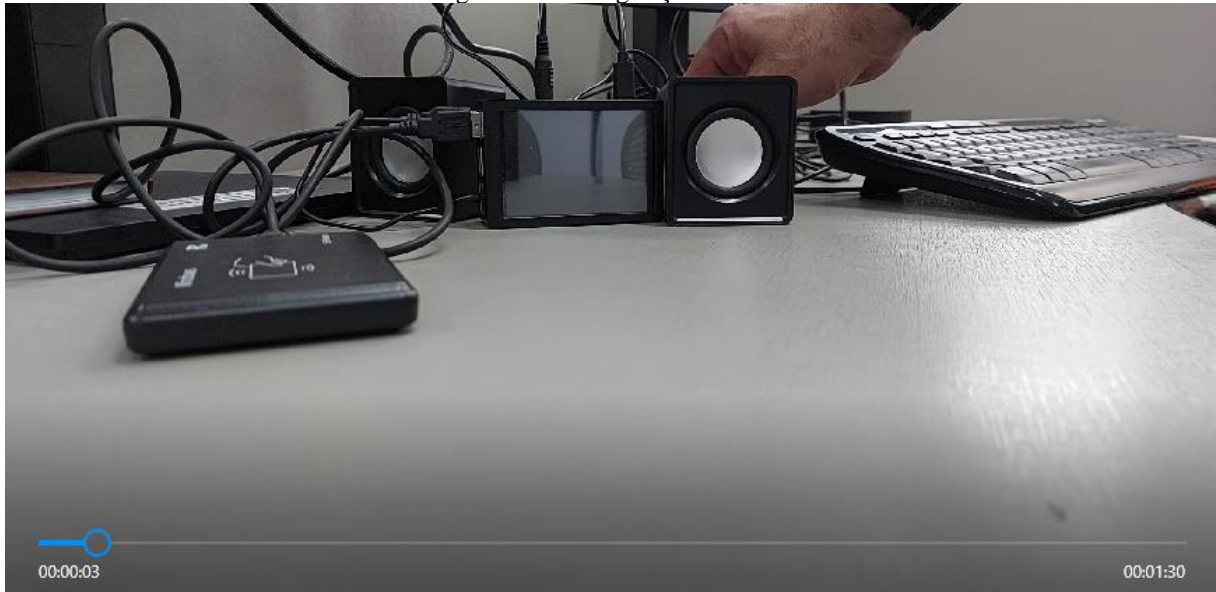
Fonte: Elaborado pelo autor

5.2.3 Tempo de inicialização

Durante os testes conduzidos, constatou-se que o tempo necessário para que o sistema esteja pronto para operar, após ser ligado, é inferior a trinta segundos. Um desses testes é exemplificado nas Figura 5.7 e na Figura 5.8, as quais são recortes de um vídeo com duração aproximada de 63 segundos.

Na Figura 5.7, temos o momento em que o botão de energia é pressionado, ocorrendo três segundos após o início do vídeo, conforme evidenciado na parte inferior da imagem.

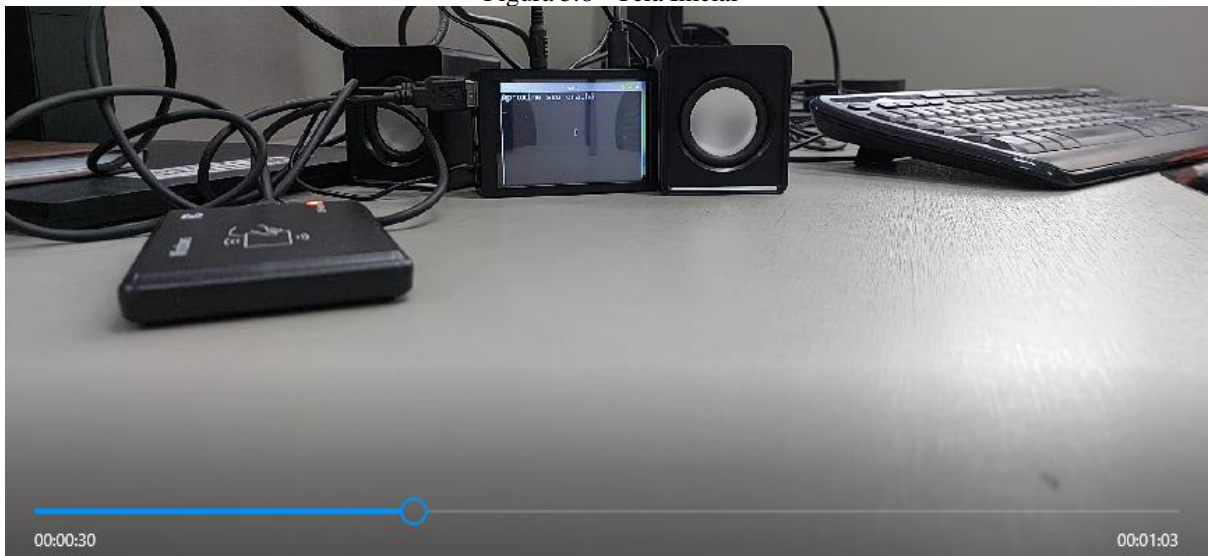
Figura 5.7 - Energização do SBC



Fonte: Acervo do autor

Na Figura 5.8, a mensagem inicial com a instrução “Aproxime seu crachá” é exibida quando o vídeo se aproxima dos 30 segundos.

Figura 5.8 - Tela Inicial



Fonte: Acervo do autor

Ao subtrair o valor inicial do valor final, observa-se que o tempo total foi aproximadamente 27 segundos. Este intervalo representa um tempo extremamente breve, especialmente considerando o ambiente onde o estudo foi conduzido.

5.2.4 Custos do projeto

De maneira complementar ao que foi abordado na Seção 4.3 na Tabela 4.1, é possível consultar os custos individuais de cada componente utilizado na construção do hardware deste protótipo na Tabela 5.3.

Tabela 5.3 - Custos dos componentes do Projeto

Item	Custo Total
Raspberry Pi 4B + Cartão de Memória + Fonte Chaveada	R\$ 353,13
Estojo + Tela + Dissipadores	R\$ 79,27
Leitor RFID USB	R\$ 45,12
Par de caixas de som de 3 Watts RMS	R\$ 51,00
Modem roteador WiFi	R\$ 74,99
TOTAL	R\$ 603,51

Fonte: Acervo do autor

É importante ressaltar que os valores mencionados na Tabela 5.3 podem estar sujeitos a alterações, uma vez que foram adquiridos em épocas distintas, de diferentes fabricantes e fornecedores ao redor do mundo. Além disso, os custos podem ser influenciados por diversos fatores externos, tais como variações nas taxas de câmbio e condições de oferta e demanda.

6 CONSIDERAÇÕES FINAIS

Este trabalho ofereceu uma exploração abrangente sobre sistemas embarcados, iniciando pela arquitetura de computadores e seguindo uma trajetória evolutiva que abrange sistemas dedicados, como microcontroladores e SBCs, com ênfase especial no Raspberry Pi. Destaca-se que esse dispositivo possui uma aplicação proeminente no contexto da Internet das Coisas.

A metodologia escolhida desempenhou um papel crucial na concepção deste protótipo, servindo como guia para delimitar o projeto final. Os requisitos, tanto do usuário quanto do sistema, receberam destaque especial, influenciando a seleção dos materiais nesta primeira versão e o desenvolvimento do software. Essa abordagem metodológica foi essencial para assegurar que as escolhas de projeto estivessem alinhadas com as necessidades e expectativas identificadas.

Os fluxogramas de funcionamento estabeleceram uma base sólida para que o software desenvolvido possa ser reescrito em outra linguagem ou framework, se necessário. Essa estrutura também possibilita a expansão de funcionalidades, como a integração de um módulo externo para a aquisição de dados de GPS, por exemplo.

Os testes com os usuários indicam que o sistema atendeu aos requisitos estabelecidos, destacando-se pelo tamanho não intrusivo e pela capacidade de capturar e enviar informações de forma correta. Vale ressaltar que o processamento dos dados coletados ocorreu em velocidade praticamente imperceptível ao usuário, e os gestores têm acesso em tempo real a essas informações por meio de relatórios de Inteligência de Negócios no *Power BI*.

Do ponto de vista da manutenção, observa-se que o sistema possui indicativos próprios de funcionamento, minimizando a necessidade de intervenção manual. Caso seja necessário realizar correções, os principais problemas estão mapeados com suas soluções correspondentes.

Considerando os custos do projeto e a facilidade de implementação, este sistema demonstra ser uma solução eficaz, tanto em termos de hardware quanto de software, para monitorar o acesso dos colaboradores a salas de descanso.

6.1 SUGESTÕES PARA TRABALHOS FUTUROS

Com o intuito de fornecer diretrizes para futuras pesquisas e desenvolvimentos, apresentam-se as seguintes recomendações:

- Explorar o uso de outros modelos do Raspberry Pi, como os da linha zero, que possuem dimensões e custos inferiores ao modelo utilizado neste trabalho, embora apresentem menor capacidade de processamento e memória.
- Conduzir testes em larga escala, permitindo que o equipamento permaneça ligado e operacional por centenas de horas consecutivas.
- Investigar abordagens relacionadas ao aprendizado de máquinas, processamento de big data e criação de relatórios de inteligência de negócios.
- Desenvolver uma estrutura unificada para acomodar todos os componentes em um único bloco.
- Replicar o mesmo sistema embarcado utilizando microcontroladores e placas de circuito impresso dedicadas.
- Realizar um estudo comparativo da relação custo-benefício e desempenho entre o Raspberry Pi e outros Computadores de Placa Única (SBCs) disponíveis no mercado, como o Orange Pi e o Rock Pi.
- Adaptar o software para outras áreas que também demandam o registro de entrada e saída de pessoas, tais como salas de aula, ponto eletrônico, espaços confinados, áreas de risco e ambientes restritos.
- Pesquisar sobre redes de longo alcance que permitam a comunicação entre dispositivos e uma central, possibilitando que esta se comunique com a nuvem, reduzindo assim os custos associados a dados móveis.

REFERÊNCIAS

- [1] TST - Tribunal Superior do Trabalho, “Jornada de Trabalho,” SECOM - Secretaria de Comunicação, [Online]. Available: <https://www.tst.jus.br/jornada-de-trabalho>. [Acesso em 11 15 2022].
- [2] DIÁRIO OFICIAL DA UNIÃO, “PORTARIA Nº 1.359, DE 9 DE DEZEMBRO DE 2019,” Ministério da Economia/Secretaria Especial de Previdência e Trabalho, 09 12 2019. [Online]. Available: <https://www.in.gov.br/en/web/dou/-/portaria-n-1.359-de-9-de-dezembro-de-2019-232663857>. [Acesso em 15 11 2022].
- [3] I9 Automação Comercial, “Relógio Ponto Eletrônico Henry Prisma c/ 10 - Crachás RFID,” 15 08 2023. [Online]. Available: <https://www.i9automacaocomercial.com.br/relogio-ponto-eletronico-henry-prisma-c-10-crachas-rfid/>. [Acesso em 12 10 2023].
- [4] Raspberry Pi, “Raspberry Pi 4,” [Online]. Available: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>. [Acesso em 25 07 2023].
- [5] A. Thomsen, “Controle de Acesso usando Leitor RFID com Arduino,” www.makehero.com, 22 04 2014. [Online]. Available: <https://www.makehero.com/blog/control-access-leitor-rfid-arduino/>. [Acesso em 26 10 202].
- [6] A. Bejo, R. Winata e S. S. Kusumawardani, “Prototyping of Class-Attendance System Using Mifare 1K Smart Card and Raspberry Pi 3,” *2018 International Symposium on Electronics and Smart Devices (ISESD)*, pp. 1-5, 2018.
- [7] Z. Ali, M. Waqas, M. Waqas e A. Shahzad, “Fast and Low-Cost, Wireless Controlled Smart Shopping Cart with Raspberry Pi and RFID Technology,” em *2022 3rd International Informatics and Software Engineering Conference (IISEC)*, Ankara, Turkey, 2022.
- [8] A. V. Omkar, J. Goyal, R. Ranjeet, A. H. Nalband e M. R. Ahmed, “Design and Development of secure RFID Based Health Card for integrating Various Health Services,” em *2023 IEEE International Conference on Contemporary Computing and Communications (InC4)*, Bangalore, India,, 2023.

- [9] J. W. Valvano, *Embedded Microcomputer Systems: Real Time Interfacing*, 2nd ed., Stamford: Cengage Learning, 2010, p. 822.
- [10] M. Arora, *Embedded System Design: Introduction to SoC System Architecture*, 1 ed., Austin, Texas: Learning Bytes Publishing, 2016, p. 213.
- [11] M. Wolf, *Computers as Components: Principles of Embedded Computing System Design*, 4th ed., Cambridge: Elsevier, 2017.
- [12] R. M. A. Almeida, C. H. V. Moraes e T. F. P. Seraphim, *Programação de Sistemas Embarcado: Desenvolvendo software para microcontroladores em linguagem C*, 1ª ed., Rio de Janeiro: LTC, 2021, p. 437.
- [13] M. Barr e A. Massa, *Programming Embedded Systems: with C and GNU Development Tools*, 2nd ed., Sebastopol, CA: O'Reilly Media, Inc., 2006, p. 328.
- [14] Engineering and Technology History Wiki, "IEEE Milestone:Whirlwind Computer, 1944-59," 21 11 2022. [Online]. Available: https://ethw.org/Milestones:Whirlwind_Computer,_1944-59. [Acesso em 07 11 2023].
- [15] A. N. Kani, *Microprocessors and microcontrollers*, 2nd ed., New Delhi: Tata McGraw-Hill, 2012, p. 805.
- [16] W. R. Plaza, "hardware.com.br," 16 11 2021. [Online]. Available: <https://www.hardware.com.br/artigos/50-anos-do-mitologico-intel-4004-o-processador-que-mudou-o-mundo/>. [Acesso em 07 11 2023].
- [17] F. d. Reis, "Introdução aos Microcontroladores," *BÓSON TREINAMENTOS EM CIÊNCIA E TECNOLOGIA*, 24 04 2015. [Online]. Available: <http://www.bosontreinamentos.com.br/electronica/electronica-geral/introducao-aos-microcontroladores/>. [Acesso em 08 11 2023].
- [18] Espressif, "ESP32," Espressif, 2023. [Online]. Available: <https://www.espressif.com/en/products/socs/esp32>. [Acesso em 08 11 2023].
- [19] R. Rajsuman, *System-on-a-Chip: Design and Test*, Santa Clara: Artech House Publishers, 2000, p. 292.
- [20] C. G. C. Carducci, G. Lipari, N. Giaquinto, F. Ponci e A. Monti, "Error Model in Single-Board Computer-Based Phasor Measurement Units," *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, vol. 69, nº 9, pp. 6155-6164, September 2020.

- [21] M. Rahman, Z. I. A. Khalib e R. B. Ahmad, “Performance Evaluation of PNtMS: A Portable Network Traffic Monitoring System on Embedded Linux Platform,” *International Conference on Computer Engineering and Technology*, pp. 108-114, 2009.
- [22] U. Isikdag, “Internet of Things: Single-Board Computers,” em *Enhanced Building Information Models*, Istanbul, Springer, Cham, 2015, pp. 43-53.
- [23] N. Gershenfeld, R. Krikorian e D. Cohen, “The Internet of Things,” *Scientific American*, pp. 76-81, 10 2004.
- [24] P. Xiao, *Designing Embedded Systems and the Internet of Things (IoT) with the ARM® Mbed™*, 1st ed., Hoboken, New Jersey: John Wiley & Sons, Inc., 2018.
- [25] Cirani, Simone, G. Ferrari, M. Picone e L. Veltri, *Internet of Things: Architectures, protocols and standards*, Hoboken, New Jersey: John Wiley & Sons, Inc., 2019.
- [26] R. Singh, A. Gehlot, L. R. Gupta, B. Singh e M. Swain, *Internet of Things with Raspberry Pi and Arduino*, Boca Raton: CRC Press, 2020.
- [27] J. F. Kurose e K. W. Ross, *Computer Networking – A Top-Down Approach*, 8th ed., Harlow: Pearson Education, 2021, p. 797.
- [28] H. F. W. Puhlmann, “Introdução à tecnologia de identificação RFID,” *Embarcados*, 31 03 2015. [Online]. Available: <https://embarcados.com.br/introducao-a-rfid/>. [Acesso em 17 11 2022].
- [29] EQUIPE TOTVS, “O que é RFID, como funciona, importância, tipos, como usar e mais!,” 08 07 2022. [Online]. Available: <https://www.totvs.com/blog/gestao-industrial/rfid/>. [Acesso em 17 11 2022].
- [30] I. Sommerville, *Engenharia de Software*, 9ª ed., São Paulo: Parson Prentice Hall, 2011, p. 529.
- [31] SpecsEra, “Samsung Galaxy A24 Specs, Release Date, Price & Deals,” *SpecsEra*, 1 6 2023. [Online]. Available: <https://www.specsera.com/mobile/samsung-galaxy-a24/>. [Acesso em 25 07 2023].
- [32] Microsoft, “Banco de Dados SQL do Azure,” 2023. [Online]. Available: <https://azure.microsoft.com/pt-br/products/azure-sql/database/>. [Acesso em 15 11 2023].
- [33] T. Noergaard, *Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers*, 2 ed., Waltham, Massachusetts: Elsevier, 2013.

APÊNDICE A – LISTAGEM COMPLETA DOS REGISTROS DE TESTE

```
PS E:\Logs\2023-07-10\B> dir | Sort-Object Name
```

Diretório: E:\Logs\2023-07-10\B

Mode	LastWriteTime	Length	Name
-a----	10/07/2023 15:05	131	02-C1 .001.12-05-33.CHECKIN.txt
-a----	10/07/2023 15:08	132	02-C1 .001.12-08-49.CHECKOUT.txt
-a----	10/07/2023 12:18	131	10-93 .001.09-18-21.CHECKIN.txt
-a----	10/07/2023 12:18	131	13-4F .001.09-18-15.CHECKIN.txt
-a----	10/07/2023 12:50	132	13-4F .001.09-50-00.CHECKOUT.txt
-a----	10/07/2023 12:51	131	13-4F .002.09-51-01.CHECKIN.txt
-a----	10/07/2023 14:15	132	13-4F .002.11-15-50.CHECKOUT.txt
-a----	10/07/2023 15:09	131	13-4F .003.12-09-38.CHECKIN.txt
-a----	10/07/2023 16:25	132	13-4F .003.13-25-24.CHECKOUT.txt
-a----	10/07/2023 16:26	131	13-4F .001.13-26-41.CHECKIN.txt
-a----	10/07/2023 16:29	132	13-4F .001.13-29-15.CHECKOUT.txt
-a----	10/07/2023 14:25	131	13-51 .001.11-25-10.CHECKIN.txt
-a----	10/07/2023 15:04	132	13-51 .001.12-04-42.CHECKOUT.txt
-a----	10/07/2023 12:58	131	4B-75 .001.09-58-46.CHECKIN.txt
-a----	10/07/2023 13:12	132	4B-75 .001.10-12-25.CHECKOUT.txt
-a----	10/07/2023 12:42	131	4B-7C .001.09-42-41.CHECKIN.txt
-a----	10/07/2023 12:44	132	4B-7C .001.09-44-52.CHECKOUT.txt
-a----	10/07/2023 12:18	131	53-15 .001.09-18-07.CHECKIN.txt
-a----	10/07/2023 12:19	132	53-15 .001.09-19-22.CHECKOUT.txt
-a----	10/07/2023 15:05	131	53-15 .002.12-05-51.CHECKIN.txt
-a----	10/07/2023 15:08	132	53-15 .002.12-08-31.CHECKOUT.txt
-a----	10/07/2023 15:00	131	6F-52 .001.12-00-03.CHECKIN.txt
-a----	10/07/2023 15:09	132	6F-52 .001.12-09-28.CHECKOUT.txt
-a----	10/07/2023 12:18	131	9A-4F .001.09-18-36.CHECKIN.txt
-a----	10/07/2023 17:22	131	9C-F4 .001.14-22-44.CHECKIN.txt
-a----	10/07/2023 17:24	132	9C-F4 .001.14-24-26.CHECKOUT.txt
-a----	10/07/2023 17:23	131	A6-C7 .001.14-23-00.CHECKIN.txt
-a----	10/07/2023 17:24	132	A6-C7 .001.14-24-34.CHECKOUT.txt
-a----	10/07/2023 12:38	131	BA-17 .001.09-38-49.CHECKIN.txt
-a----	10/07/2023 12:46	131	BA-CE .001.09-46-15.CHECKIN.txt
-a----	10/07/2023 12:49	132	BA-CE .001.09-49-33.CHECKOUT.txt
-a----	10/07/2023 12:51	131	BA-CE .002.09-51-20.CHECKIN.txt
-a----	10/07/2023 15:05	132	BA-CE .002.12-05-27.CHECKOUT.txt
-a----	10/07/2023 15:08	131	BA-CE .003.12-08-42.CHECKIN.txt
-a----	10/07/2023 16:26	132	BA-CE .003.13-26-45.CHECKOUT.txt
-a----	10/07/2023 12:22	131	BA-DE .001.09-22-26.CHECKIN.txt
-a----	10/07/2023 12:26	132	BA-DE .001.09-26-00.CHECKOUT.txt
-a----	10/07/2023 12:46	131	D3-ED .001.09-46-23.CHECKIN.txt
-a----	10/07/2023 13:11	131	DF-64 .001.10-11-59.CHECKIN.txt
-a----	10/07/2023 12:31	131	F5-4F .001.09-31-18.CHECKIN.txt
-a----	10/07/2023 12:34	132	F5-4F .001.09-34-40.CHECKOUT.txt