

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA

**PROTÓTIPO DE AMBIENTE INTELIGENTE PERSONALIZADO VIA
WIFI UTILIZANDO RECONHECIMENTO BIOMÉTRICO**

Porto Alegre, vinte e oito de junho de 2018.

Autora: Izabela Souza de Vasconcelos

Pontifícia Universidade Católica do Rio Grande do Sul

Curso de Engenharia Elétrica

Av. Ipiranga 6681 - Prédio 30 - CEP: 90619-900 - Porto Alegre - RS - Brasil

Email: izabela.vasconcelos@acad.pucrs.br

Orientador: Prof. Me. Anderson Royes Terroso

Pontifícia Universidade Católica do Rio Grande do Sul

Av. Ipiranga 6681 - Prédio 30 - CEP: 90619-900 - Porto Alegre - RS - Brasil

Email: terroso@pucrs.br

RESUMO

Há uma crescente necessidade em melhoria de sistemas de aplicações residenciais e a *Internet of Things* (IoT) é uma tecnologia que pode ajudar a atender a essa demanda, porém está inserida em um mercado ainda restrito a um número de pessoas devido seu custo elevado. Nesse sentido, no presente trabalho desenvolveu-se um sistema de sala inteligente (*smart home*) a fim de explorar os microcontroladores atuais de baixo custo e torná-lo acessível, contendo um controle de acesso capaz de reconhecer o usuário e suas preferências. Para a realização do projeto utilizou-se a comunicação WiFi para efetivar a troca de informações entre os dispositivos e reconhecimento biométrico para obter a identificação do usuário que acessa o ambiente. Desta forma, buscou-se aplicar um sistema adaptável, utilizando a tecnologia IoT, para gerar interações que proporcionem bem-estar e conforto ao usuário.

Palavras-chave: *Internet of things. Smart Home. WiFi.* Ambientes inteligentes.

1 INTRODUÇÃO

Em 1999, Kevin Ashton utilizou a expressão *Internet of Things* (IoT) pela primeira vez para descrever a troca de informações entre dispositivos sem uma intervenção humana. Desde então, o IoT tem sido uma promessa global para a economia com previsão de que, em alguns

anos, se tenha mais dispositivos conectados do que pessoas no mundo, cerca de 30 bilhões de “coisas” conectadas à internet gerando diversas aplicações e serviços (SANTOS, 2018).

Nas construções são instalados interruptores, tomadas e termostatos por fiação enterrada com funções fixas, onde os computadores e aparelhos são vistos como intrusos. O fato de muitos dos dispositivos não terem conexão, mesmo com o grande número de aplicações nas residências e na indústria, implica de maneira significativa na economia da construção, na eficiência energética, na arquitetura e na qualidade de vida das pessoas (GERSHENFELD; KRIKORIAN; COHEN, 2004).

E, embora tenha-se tecnologias avançadas, a aplicação do IoT continua com o custo acessível a poucas pessoas, o que a mantém como promessa para a economia no futuro, pois instalações simples de automação podem custar a partir de R\$ 3 mil (CASAS INTELIGENTES, 2018). Uma cortina automatizada, por exemplo, com acionamento por controle remoto tem como preço inicial R\$1.200,00 sem a instalação. Pacotes com custos de aproximadamente R\$ 15 mil incluem luzes, ar condicionado, cortinas, sistema de som, TV e circuito interno de câmeras (AUTOMAÇÃO RESIDENCIAL, 2018).

A Samsung possui um kit chamado *SmartThings*, o pacote básico é composto por travas magnéticas, sensor de presença, tomada inteligente e sensor de vazamento de água e possui um custo em torno de U\$319,00 (SAMSUNG, 2018), sendo encontrado no mercado nacional, em sites de buscas como o Mercado Livre¹, por aproximadamente R\$999,00. A utilização dessas novas tecnologias inteligentes são aplicadas para o melhor controle, segurança e economia dos cômodos de um ambiente, visto que os sistemas funcionam comparando os dados entre si, otimizando seu funcionamento e tornando o modelo de aplicação mais eficiente e econômico.

Neste trabalho desenvolveu-se um sistema de sala inteligente contendo um controle de acesso que será capaz de reconhecer o usuário através de leitura biométrica. Desta forma, é possível configurar a refrigeração, o som ambiente e a luminosidade, conforme a preferência do usuário. Ainda, contém um sensor de temperatura e umidade e um sistema de segurança com alarme. O objetivo é explorar os microcontroladores atuais de baixo custo e a aplicar tecnologia simples de fácil acesso para tornar o sistema acessível.

1.1 Motivação

Com a crescente demanda em melhoria de sistemas industriais, houve também um crescimento nas aplicações residenciais, inicialmente, com a automação e, evoluindo para aplicações mais complexas. Nesse contexto pensou-se em desenvolver um sistema inteligente

¹ Site de compra e venda online.

aplicado à residências para que, através da utilização de IoT, torne possível interações que proporcionem ao usuário bem-estar e conforto. A realização do projeto propõe novas experiências às pessoas, com tecnologias aplicadas a ambientes do cotidiano, o que poderá melhorar a qualidade de vida dentro da própria residência.

1.2 Objetivos

O trabalho busca explorar aplicações de custo mais baixo possibilitando que um número maior de pessoas tenha acesso a casas inteligentes, pois hoje o IoT é uma tecnologia que não está acessível à maioria das pessoas devido seu alto custo. Desta forma os objetivos são:

- Aplicar um sistema inteligente que se adapta a cada usuário, utilizando as informações para controles futuros do ambiente;
- Utilizar IoT para unificar os sistemas, tornando-o um só pelos dispositivos estarem todos conectados e trocando informações em tempo real;
- Tornar o sistema acessível para abranger a utilização por um grupo maior de usuários.

1.3 Delimitações do Trabalho

O presente trabalho abrange apenas dois usuários, já cadastrados, não sendo possível novos cadastros em virtude de o foco principal do trabalho ser voltado para ambientes inteligentes que se adaptam ao usuário. Também, o trabalho aplica-se apenas a uma sala residencial, não sendo aplicado em setores comerciais ou industriais. Além disso, o trabalho não utiliza inteligência artificial, ou seja, não será mapeada todas as escolhas do usuário e sim gravada a última informação inserida.

A maneira de utilização segue duas etapas: para entrar, efetuar leitura biométrica; e para sair ou efetuar leitura biométrica ou utilizar os botões de saída. Sendo assim, o sistema não identifica comportamentos diferentes dos citados, como em situações onde o usuário libera o acesso para entrar e não realizar a entrada no ambiente, pois, desta maneira, o sistema ficará ligado com as preferências do usuário sem ele estar na residência. Ou, ainda, caso os dois usuários entrarem juntos na residência e apenas um liberar a entrada, visto que o sistema carregará apenas as preferências do mesmo, se comportando como se tivesse apenas uma pessoa na casa.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta os principais conceitos sobre Internet das Coisas (*Internet of Things*), Casa Inteligente (*Smart Home*) e reconhecimento biométrico.

2.1 *Internet of Things* (IoT)

Também conhecida pela tradução “Internet das Coisas” ou a sigla “IoT”, o conceito de *internet of things* tem como premissa a “internet” – sistema global de redes de computadores que utiliza um conjunto padrão de protocolos (TCP/IP) atendendo bilhões de usuários no mundo todo, através de redes públicas, privadas, empresariais e governamentais – e as “coisas” – dispositivos que utilizam a internet para troca de dados, informações e comunicação, através de redes eletrônicas, sem fio, ópticas, etc. Dentro desse conceito, também é possível monitorar objetos do cotidiano como coisas vivas (animais, pessoas, plantas, etc) e coisas sem vida (cortina, cadeira, mesa) ou quaisquer objetos físicos e materiais (MADAKAM; RAMASWAMY; TRIPATHI, 2015).

De acordo com o *Internet Live Stats*² (2018), no dia 28 de maio de 2018, havia cerca de 4 bilhões usuários de internet em todo o mundo. Esse número de usuários de internet representa mais da metade da população mundial (WORLDOMETERS, 2018), o que percebe-se que, aproximadamente, 52% da população mundial utiliza internet na atualidade. No Brasil os dados são mais altos, de acordo com pesquisa do IBGE³ (2016), cerca de 116 milhões de pessoas estão conectadas à internet, o que representa 64,7% da população do país.

Como citado anteriormente, a expressão *Internet of Things* (IoT) foi dita pela primeira vez por Kevin Ashton em 1999 (como um título da sua apresentação para a empresa Procter & Gamble (P&G) (ASHTON, 2009) e, mesmo após quase 20 anos, não há definição específica de IoT. Embora pesquisadores, acadêmicos, profissionais, desenvolvedores e pessoas em geral utilizarem definições diferentes, todas têm uma ideia em comum, de que entende-se IoT como uma rede aberta e abrangente de objetos inteligentes que tem a capacidade de auto-organizar, compartilhar informações, dados e recursos, reagindo e atuando em situações frente a mudanças no ambiente (BOUCOUVALAS; KOSMATOS; TSELIKAS, 2011).

A necessidade das grandes corporações de se beneficiar com a previsibilidade proporcionada pela possibilidade de acompanhar todos os dispositivos, bem como a capacidade de rastrear e codificar objetos têm impulsionado o desenvolvimento do IoT. Desta forma, a

² Site de estatísticas de uso da internet e mídias sociais.

³ Instituto Brasileiro de Geografia e Estatística.

utilização do IoT permite a redução de custos, acelera processos produtivos e os torna mais eficientes devido a facilidade na identificação de erros, possibilitando também que as empresas codifiquem e rastreiem objetos, o que melhora seus sistemas de segurança (MADAKAM; RAMASWAMY; TRIPATHI, 2015). O desenvolvimento do IoT depende de novas técnicas que abrangem importantes campos da computação, utilizando sensores sem fio, até a nanotecnologia (matéria em nanoescala, em escala atômica e molecular), fazendo com que a IoT seja a representação do futuro como uma revolução tecnológica da computação e das comunicações (MADAKAM; RAMASWAMY; TRIPATHI, 2015).

Inicialmente utilizada pelos usuários de RFID (*Radio-Frequency IDentification*) (SUN, 2012), a IoT é aplicada em diversas tecnologias de comunicação como WiFi que é uma abreviação de *Wireless Fidelity* (fidelidade sem fio, em português) é uma tecnologia de comunicação de rede que permite que computadores e outros dispositivos se comuniquem por um sinal sem fio e, geralmente, é transmitida através de frequências de rádio, infravermelhos, etc. Outra tecnologia de comunicação que aplica-se a IoT é IP address, a abreviação de *Internet Protocol address* (endereço de protocolo da internet, em português), que é o principal protocolo de rede usado na Internet, desenvolvido na década de 1970. É o responsável por endereçar e encaminhar os pacotes que trafegam pela rede mundial de computadores.

2.2 *Smart Home*

Smart Home, em tradução livre, significa casa inteligente. De acordo com Zhang et al (2017), pode-se entender que é um ambiente vivo que realiza o controle automático de dispositivos e sistemas. Fornece às pessoas serviços mais humanizados, como ambientes com entretenimento doméstico inteligente, comunicação, controle, vigilância remota, etc. Com a velocidade do desenvolvimento da rede de computadores, sensores embarcados e a tecnologia de informação eletrônica, a automação residencial também expande-se cada vez mais. (NUGENT et al, 2008). Essa proporciona benefícios, tais como:

- **Segurança:** Por intermédio do uso de câmeras com melhor qualidade, tanto para o controle do portão ou identificação de usuários, tudo pode ser monitorado e controlado através da internet.
- **Economia energética:** Acender as luzes somente quando necessário e na intensidade desejada, desligar aparelhos que consomem muita energia quando esses não são utilizados, fazendo a utilização dos recursos da casa de maneira eficiente tudo isso contribuindo para uma economia de energia.

- **Acompanhamento:** Um dos principais objetivos da casa inteligente é fazer com que seus moradores tenham uma vida independente. É possível lembrar o idoso dos remédios ou de ligar para os filhos, acompanhar a alimentação e as horas de sono dos moradores, dentre outras coisas. Atualmente, há um vasto campo de pesquisa para casas inteligentes para usuários infantis, jovens, adultos e idosos.
- **Conforto:** A união dos itens acima acaba gerando conforto, mas ele é bem mais abrangente: ajustar as persianas de maneira ideal na hora de assistir à um filme, preparar a casa com a iluminação que o usuário prefere, chegar em casa e a TV ligar no seu canal favorito e ainda esquentar a banheira, tudo isso automaticamente para quando o usuário chegar do trabalho. Essas são apenas algumas das funcionalidades que uma casa inteligente pode ter, o que pode melhorar a vida das pessoas.

A automação residencial sem fio é, normalmente, apresentada em duas partes: sensores embarcados sem fio e atuadores. Através dos dados coletados por estes sensores é possível gerenciar e controlar o ambiente de forma inteligente podendo tornar o espaço doméstico comum em uma experiência que melhora a qualidade de vida das pessoas. Desta forma, possibilita também diminuir o consumo de energia de eletrodomésticos, obter informações instantâneas da casa, bem como ajudar a melhorar a organização de horários das pessoas (ZHANG et al, 2017).

2.2.1 Utilização de WiFi em Smart Home

Os avanços recentes da tecnologia sem fio possibilitam observar que os sinais WiFi são sensíveis o bastante para capturar a dinâmica em um ambiente qualquer, podendo ser utilizado para a detecção de mudanças ocorridas nesse espaço. Construir casas inteligentes utilizando WiFi pode melhorar aplicações já existentes na automação residencial (que ainda utiliza meios físicos) devido a seus benefícios.

Dentre eles pode-se citar a fácil implantação, em razão de o hardware utilizado em casas inteligentes ser simples e de fácil construção, logo, é possível acrescentar vários módulos ao sistema já existente. Também, o custo benefício, pois implementações com sensores WiFi possibilitam detectar tarefas já existentes nos locais que utilizam módulos WiFi típicos presentes em ambientes internos, o que torna mais fácil a possibilidade de melhorias do sistema. Ainda, como benefício, ressalta-se a troca de informações e tarefas entre sensores WiFi,

possibilitando o sistema realizar monitoramento de mudanças no ambiente, bem como permitem que os usuários e seu comportamento sejam reconhecidos (JIANG et al, 2018).

2.3 RaspBerry Pi (RPi)

RaspBerry (RPi) é um pequeno computador com custo aproximado de R\$ 190,00⁴, criado para ensinar programação a crianças pela Fundação *Raspberry Pi*, instituição de caridade do Reino Unido que trabalha para que todas as pessoas do mundo tenham acesso a tecnologia. Porém, é possível utilizá-lo em diversas aplicações, como sistemas de domótica, monitoramento de dados, *videogames* e automações em geral. Sua compatibilidade com conexões externas através de GPIO (*General Purpose Input/Output*), que são portas programáveis de entrada e saída de dados, dá capacidades de controlar outros dispositivos que fazem parte do sistema, entre eles, sensores, câmera, vídeo, etc. O sistema operacional oficial do RPi é o Raspbian (RASPBERRYPI, 2018).

No trabalho proposto, o RPi é responsável pela troca de informações entre a interface desenvolvida no Qt Creator (ambiente de desenvolvimento integrado de plataforma, abordado no item 2.6) e o Cortex-M0 (microcontrolador Arm, abordado no item 2.5), através da conexão ao módulo WiFi (abordado no item 2.7.2). Para o desenvolvimento deste projeto, é utilizado um RPi 3 modelo B+ que já possui WiFi no seu hardware não necessitando de um adaptador. Além disso, o RPi é responsável pela Interface Homem-Máquina (IHM), pois é nele que ficará hospedada a interface Qt do sistema.

2.4 Leitor Biométrico

Com a grande necessidade de um método de autenticação seguro, devido os problemas de sistemas que tem transmissibilidade de segredos ou senhas de usuários, a autenticação biométrica tornou-se a opção mais utilizada no último século, pois ela atende questões relacionadas a autenticação e a identificação do usuário (MAGALHÃES & SANTOS, 2003). Existem alguns tipos de elementos de segurança que utilizam tecnologia biométrica em aplicações IoT, neste trabalho utilizou-se a leitura da impressão digital.

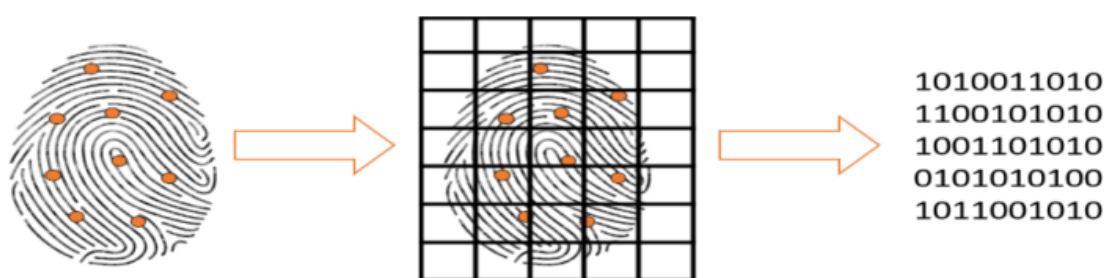
⁴ De acordo com busca no site Mercado Livre.

2.4.1 Impressão digital

A impressão digital é uma das melhores e mais utilizadas tecnologias biométricas para os sistemas de segurança e identificação (PRAKASH & VENKATRAM, 2016). É um dos métodos utilizados em investigações criminais e registros civis (MAGALHÃES & SANTOS, 2003). A tecnologia biométrica com impressão digital é aplicada, principalmente, no combate à fraudes, por exemplo, no Brasil foi aprovado em 2005 o projeto “Plano diretor de atualização cadastral e aperfeiçoamento dos sistemas de votação e identificação do eleitor” (KANASHIRO, 2011) e, desde então, tem-se efetuado cadastramentos com a meta de concluir a operação em todo país até 2022 (TSE, 2018).

Os leitores biométricos possuem diferentes tecnologias para capturar as características únicas existentes nas impressões, porém o mais comum é o leitor óptico. Um exemplo do funcionamento de leitor óptico é apresentado na **figura 1**, no qual utiliza lentes para refletir a luz em sensores complementares que convertem a luz em sinais eletrônicos. Impressões digitais, são compostas por sulcos e vales na pele, únicas para cada indivíduo. Os sulcos e os vales criam diferentes correntes elétricas em diferentes áreas do dedo. O leitor utiliza a corrente elétrica para criar uma imagem da impressão digital. Através dos vales e sulcos existentes no dedo que atuam como isoladores, o fluxo de corrente produzido pelo leitor é restringido, utilizando esses dados para gerar uma imagem individual do dedo criando um código binário para cada indivíduo.

Figura 1 - Funcionamento de leitor biométrico óptico



Fonte: Just Ask Gemalto (2018).

O leitor biométrico utilizado neste trabalho será o GT-511C3 da ADH Tech, é um pequeno módulo embarcado que consiste em um sensor óptico montado em uma pequena placa de circuito. O sensor óptico digitaliza uma impressão digital e o microcontrolador e o software fornecem a funcionalidade dos módulos, que processa automaticamente a impressão

digitalizada (DATASHEET GT-551C3, 2018). Tem como função controlar o acesso do ambiente e reconhecer o usuário para obter suas informações de preferência.

2.5 Cortex-M0

O Cortex-M0 é um processador com baixo consumo de energia, projetado para aplicações com microcontroladores e profundamente incorporados que exigem um processador otimizado para a área (DATASHEET CORTEX-M0, 2018). Neste trabalho, é o responsável pelo controle e acionamento dos sensores e controle do sistema, bem como responsável pela configuração e da comunicação que ocorre no módulo WiFi, é ele que fará a troca de dados com o RPi e através disso efetuará os comandos necessários.

2.6 Qt Creator

Qt Creator é um IDE, do inglês *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado. É uma multiplataforma desenhada para as necessidades dos desenvolvedores Qt que criam interfaces gráficas de aplicativos para computadores, celulares e etc (QTCREATOR, 2018). Neste trabalho, a plataforma do Qt é utilizada para o desenvolvimento da interface do sistema. O programa desenvolvido no Qt fica hospedado no RPi realizando a troca de dados através da comunicação WiFi com o Cortex-M0 em tempo real, recebendo e enviando os comandos exibidos na tela através da Interface Homem Máquina (IHM).

2.7 Sensores

Para o desenvolvimento de projetos IoT, são necessários sensores, atuadores, periféricos, etc. Estas são as “coisas” a serem controladas, lidas ou exibidas que, nesse trabalho, correspondem ao sensor HTS221 e módulo WiFi (SPWF01SA).

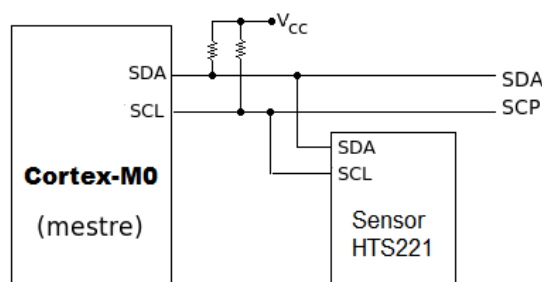
2.7.1 Sensor HTS221

O sensor HTS221 é utilizado para medição de umidade relativa do ar e temperatura em aplicações como ar condicionado, automação de *Smart Home*, automação industrial, refrigeradores, etc. Seu funcionamento é por comunicação SPI (*Serial Peripheral Interface*) ou I²C (*Inter-Integrated Circuit*) (DATASHEET HTS221, 2018).

No trabalho, o protocolo utilizado no sensor será o I²C, que é implementado com apenas duas linhas: *Serial Data* (SDA) e *Serial Clock* (SCL). A primeira é responsável pela comunicação bidirecional de dados e a segunda corresponde ao sinal de sincronismo (*clock*).

Nos protocolos de comunicação I²C, o dispositivo que inicia a comunicação é chamado de mestre (*master*, no caso do projeto será o Cortex-M0), sendo os demais, que são por ele controlados, chamados de escravos (*slaves*, o sensor HTS221). A troca de informações é bidirecional e efetuada de forma serial, que podem ser constituídas de dados, endereços ou mesmo comandos (LIMA E SILVA, 2012). Uma visão geral desse processo é ilustrada na **figura 2**.

Figura 2 - Protocolo I²C do sistema

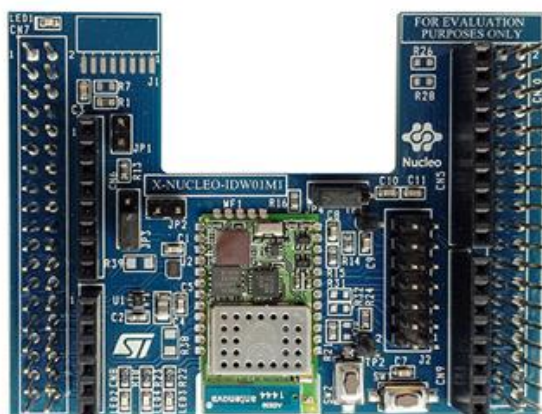


Fonte: A autora.

2.7.2 Módulo WiFi (SPWF01SA)

O módulo WiFi SPWF01SA utiliza um transceptor 802.11 b/g/n e está configurado com uma antena de banda ISM micro 2,4 GHz integrada. Com baixo consumo de energia, os módulos são ideais para aplicações sem fio fixas e móveis, bem como aplicações operadas por bateria. O módulo WiFi é configurado por comandos AT (conjunto de comandos AT é um padrão desenvolvido por Hayes para controlar modems. AT significa atenção) utilizando comunicação de porta serial UART (*Universal Asynchronous Receiver/Transmitter*). No trabalho será utilizado um núcleo da STMicroelectronics com um SPWF01SA integrado, representado na **figura 3**.

Figura 3 - X-NUCLEO-IDW01M1 da STMicroelectronics

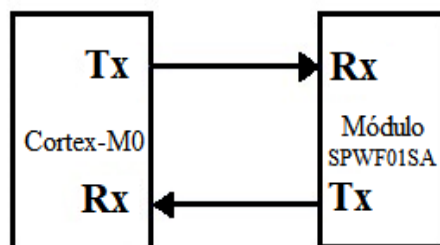


Fonte: Datasheet X-NUCLEO-IDW01M1.

O núcleo **IDW01M1** é compatível com o layout do conector do Cortex-M0, o que auxilia na aplicação deste projeto, não sendo necessário o desenvolvimento de uma PCI (placa de circuito impresso) e sim apenas encaixar o módulo no Cortex-M0. Como o núcleo IDW01M1 conecta-se fisicamente no Cortex-M0 e utiliza comunicação UART, com o RPi será através da transmissão de dados via pacotes (*sockets*), brevemente explicadas a seguir.

Transmissão de dados via UART (Receptor/Transmissor Assíncrono Universal), diferentemente dos protocolos de comunicação SPI e I²C, explicado anteriormente, transmite dados de forma assíncrona, ou seja, não há um sinal de sincronismo (*clock*). A transmissão UART adiciona *bits* (dígitos binários) no início e no fim do pacote de dados transferido, de modo que o receptor reconhece e faz a leitura dos *bits* de dados corretamente. A transferência das informações sempre ocorre do pino TX (transmissor) para o pino RX (receptor) e, quando o RX detecta um *bit* inicial, faz a leitura dos *bits* em uma frequência específica denominada como a taxa de transmissão, também usualmente conhecida como *baud rate* (CIRCUITBASICS, 2018). A taxa de transmissão é a velocidade, em *bits* por segundo (bps), que os dados são transferidos. Para o correto funcionamento da comunicação, tanto o RX quanto o TX, devem operar na mesma taxa de transmissão. Uma visão geral do protocolo é ilustrada na **figura 4**.

Figura 4 - Protocolo UART do sistema



Fonte: A autora.

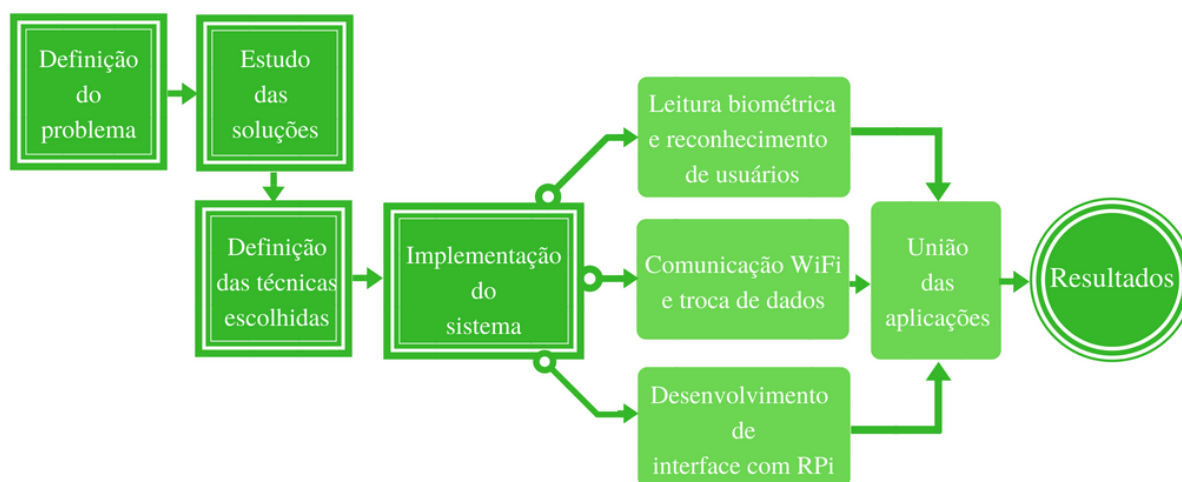
Transmissão de dados via pacotes, também é usualmente chamada de transmissão de dados via *sockets*. De acordo com Makofske et al (2004) um *socket* permite que uma aplicação conecte-se a uma rede e se comunique com outras aplicações que estão conectadas à mesma rede. A informação transmitida por uma aplicação em um *socket* pode ser recebida por outra aplicação diferente, e vice-versa. São comumente usados para interação entre cliente e servidor (IBM, 2018) e para ser estabelecida necessita que o cliente conheça o nome ou endereço IP do servidor para requisitar uma conexão a este servidor, que deve estar aguardando por conexões (ORACLE, 2018). Quando o endereço é estabelecido, o servidor espera que os clientes

solicitem um serviço. A configuração típica do sistema coloca o servidor em uma máquina, com os clientes em outras máquinas. A troca de dados de cliente para servidor ocorre quando um cliente se conecta ao servidor por meio de um *socket*. O servidor executa a solicitação do cliente e envia a resposta de volta ao cliente (IBM, 2018).

3 METODOLOGIA

Neste trabalho desenvolveu-se um sistema para controle inteligente do ambiente, o código principal implementado encontra-se no apêndice A devidamente comentado. Utiliza-se *IoT* que, como mencionado no **item 2.1**, são os dispositivos que utilizam a internet para troca de dados, informações e comunicação, através de redes eletrônicas, sem fio, ópticas (MADAKAM; RAMASWAMY; TRIPATHI, 2015), o que torna o sistema conectado entre si. Na **figura 5** pode-se observar o fluxograma das etapas deste trabalho, que inicialmente traz a definição do problema para elaboração do tema, após foi feito o estudo das possíveis soluções e em seguida a definição das técnicas a serem utilizadas. A implementação do projeto foi dividida em etapas para facilitar sua resolução e por fim estas etapas foram unificadas tornando o sistema um só.

Figura 5 - Fluxograma do projeto do protótipo *Smart Home*



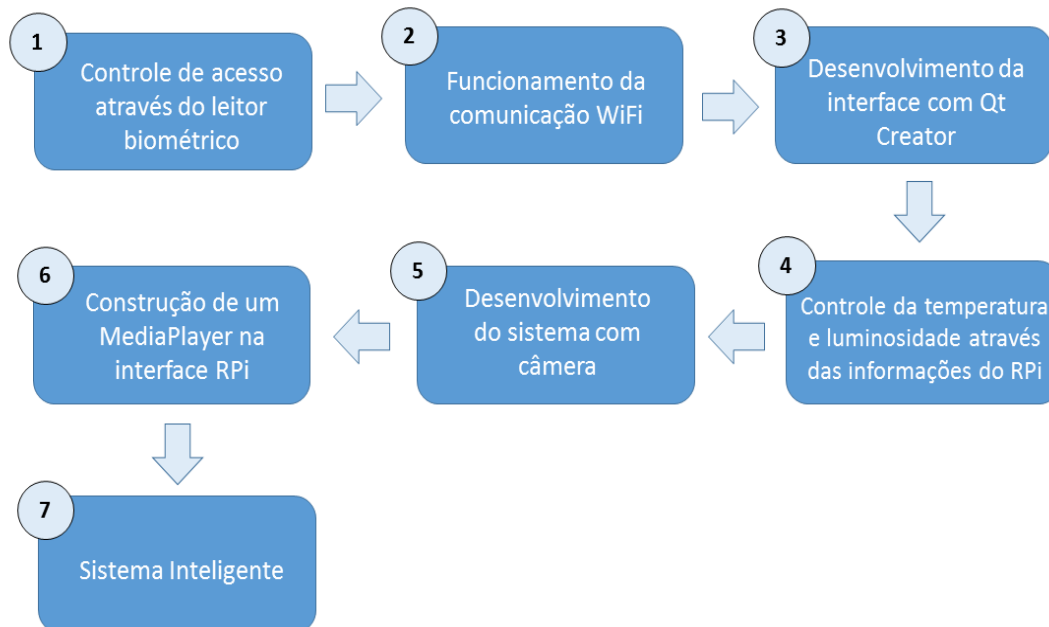
Fonte: A autora.

3.1 Estrutura do sistema

O sistema principal está embarcado no microcontrolador Cortex-M0 em linguagem C, que recebe informações do RPi + LCD, através da IHM, utilizando comunicação WiFi para transmitir e receber esses dados e efetuar todo controle do sistema com base nas informações

inseridas pelo usuário. O sistema operacional utilizado no RPi será o Raspbian. A ordem do desenvolvimento do sistema visualiza-se na **figura 6**.

Figura 6 - Ordem de desenvolvimento do sistema



Fonte: A autora.

3.1.1 Controle de acesso através do leitor biométrico

Conforme abordado no item 2.4.1, a impressão digital é uma das maneiras mais utilizadas e seguras para identificação de usuários. Neste trabalho foi utilizada a impressão digital para realizar a identificação do usuário e para o controle de acesso, fazendo com que o leitor biométrico seja o responsável pela inicialização do sistema, além disso, através dele é possível saber qual usuário está acessando a residência e assim é possível carregar as preferências desse usuário.

O módulo de impressão digital utilizado no projeto é o GT-511C3 da ADH-Tech. O próprio módulo faz todo o trabalho pesado por trás da leitura e identificação das impressões digitais com um sensor ótico integrado e processador ARM Cortex M3 de 32 bits. Este módulo pode armazenar até 200 impressões digitais diferentes (DATASHEET GT-511C3, 2018).

3.1.2 Funcionamento da comunicação WiFi

A troca de informações do sistema é através de um módulo WiFi, como visto no **item 2.1**, é uma das comunicações mais utilizadas em IoT. O módulo WiFi envia e recebe informações para o RPi e para o Cortex-M0, sendo o responsável por toda troca de dados do

sistema. Mas para o funcionamento da comunicação entre o módulo WiFi e o RPi é necessário um roteador. O envio e recebimento de dados entre o módulo WiFi e o Cortex-M0 é através de comunicação UART.

3.1.3 Desenvolvimento da interface com Qt Creator

Para a construção da interface foi utilizado a multiplataforma Qt Creator, que é um Ambiente de Desenvolvimento Integrado, e será responsável por exibir as informações do sistema, bem como receber os comandos do usuário (IHM). A interface ficará hospedada no RPi onde é possível o uso de monitores diversos, porém neste trabalho foi utilizada uma tela LCD de 7" para a exibição. Nessa tela são exibidas as informações:

Temperatura e Umidade: Sensores a serem monitorados a cada 3 segundos, exibindo as informações na tela. A sua função no Qt é apenas informativa. No cortex M0 essa informação é utilizada para fazer o controle da temperatura.

Controle de temperatura: A temperatura é inserida pelo usuário neste campo do Qt e será enviada para o Cortex-M0 através da comunicação WiFi, utilizando desta informação é feito o controle de temperatura para aquecer ou resfriar o ambiente, dependendo da temperatura atual e a temperatura desejada pelo usuário.

Controle de luminosidade: Utilizando a mesma lógica do controle de temperatura, o controle de luminosidade é inserido pelo usuário através de uma barra progressiva que o usuário poderá dimerizar a luminosidade, do qual é possível escolher até 8 níveis diferentes de luminosidade.

Alarme e travas: É possível armar e desarmar o sistema de segurança, onde cada um é independente. O alarme também é disparado quando o leitor biométrico recebe 3 tentativas de acesso inválidas.

Alerta de invasão: Quando o leitor biométrico recebe 3 tentativas de acesso inválidas ou a casa sofre alguma tentativa de invasão, uma câmera é ativada e captura a imagem do invasor, onde é exibido na tela um alerta de invasão do qual pode ser acessado pelo usuário.

Som ambiente: O usuário pode fazer o controle de som ambiente através da interface da tela.

3.1.4 Controle da temperatura e luminosidade através das informações incluídas no RPi

A partir da interface Qt é possível alterar e fazer o controle da temperatura e da luminosidade do sistema. Essas informações quando alteradas são enviadas ao Cortex-M0 e controladas por ele.

3.1.5 Desenvolvimento do sistema com câmera

Para o controle do alerta de invasão foi utilizada uma câmera Pi que é um equipamento específico para uso no RPi.

3.1.6 Construção de um Media Player na interface RPi

Um Media Player simples foi desenvolvido para som ambiente, utilizando as músicas de preferências de cada usuário.

3.1.7 Sistema Inteligente

Através deste sistema, busca-se tornar o ambiente adaptável a cada cliente, utilizando as preferências do usuário quando acessar a residência.

4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

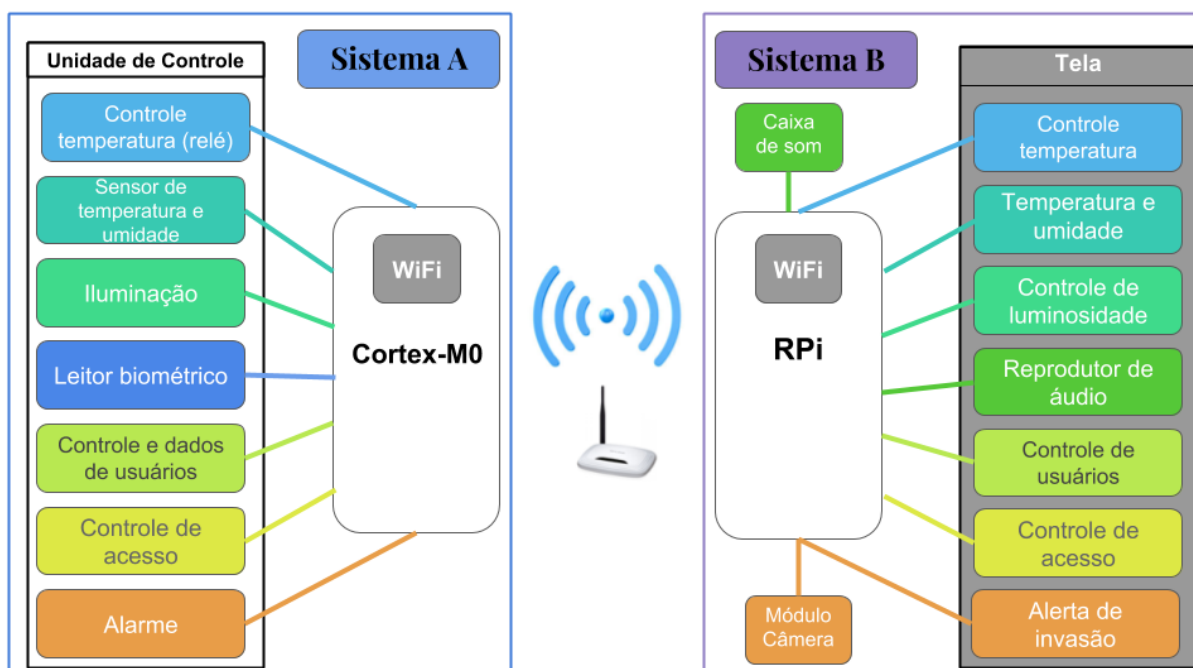
Neste capítulo são apresentados os resultados obtidos através da aplicação da metodologia proposta no capítulo anterior.

4.1 Estrutura do sistema

Para que a comunicação entre o Cortex-M0 e o RPi seja efetiva, é necessário um roteador WiFi externo, visto que o módulo WiFi e o RPi são clientes e o roteador será o servidor. Para melhor entendimento do projeto, o sistema foi separado em sistema A e sistema B, e explicou-se a união dos dois, tornando-o um só.

Um diagrama de blocos geral do sistema pode ser visto na **figura 7**. Representando a maneira que ocorre a troca de informações entre os sistemas através da comunicação WiFi com a utilização de um roteador local. O **sistema A** é a unidade de controle e é composta por: Cortex-M0, leitor biométrico, sensor de temperatura e umidade, lâmpada dimerizável, *buzzer*, módulo WiFi e módulo relé. O **sistema B** é a IHM e é composto por: RPi 3 Model B+, tela LCD 7", caixa de som, câmera Pi e interface Qt.

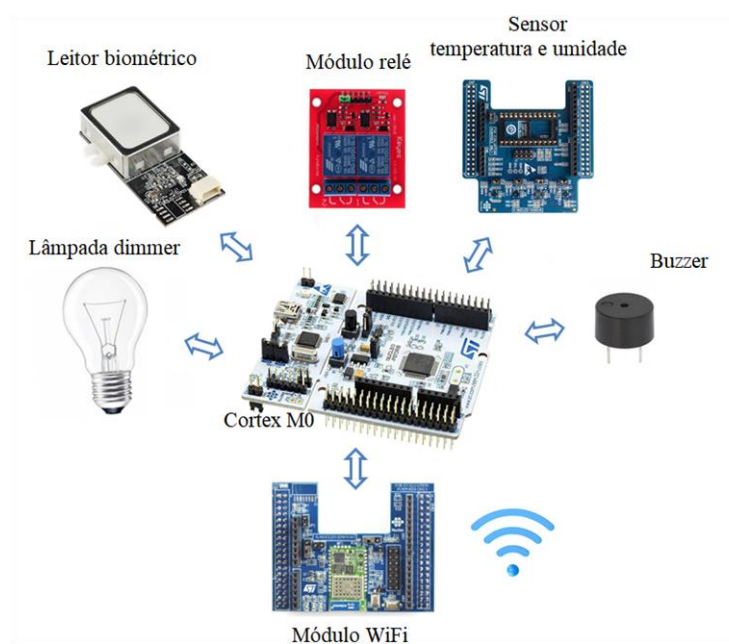
Figura 7 - Diagrama geral do sistema



Fonte: A autora.

Primeiramente, será apresentado o desenvolvimento do sistema A, e logo após o sistema B. A **figura 8** apresenta o sistema A que, como dito anteriormente, é composto por: Cortex-M0, leitor biométrico, sensor de temperatura e umidade, lâmpada dimerizável, *buzzer*, módulo WiFi e módulo relé. Neste sistema, foi desenvolvido um código em linguagem C implementado no Cortex-M0.

Figura 8 - Diagrama do sistema A



Fonte: A autora.

4.1.1 Controle de acesso através do leitor biométrico

Como citado anteriormente, o leitor biométrico é o responsável pela inicialização do sistema, além disso, através dele é possível saber qual usuário está acessando a residência e assim é possível carregar as preferências deste usuário. O funcionamento do leitor biométrico é através de comunicação UART e segue o protocolo apresentado na **figura 9**, que mostra a estrutura do protocolo de pacotes de comandos utilizados pelo leitor biométrico. Todos os comandos são através desta estrutura, alterando apenas o *offset* 4 e 8 (parâmetro e comando) para sua utilização.

Figura 9 - Protocolo de pacotes de comando

OFFSET	ITEM	TIPO	DESCRIÇÃO
0	0x55	BYTE	Comando inicial 1
1	0xAA	BYTE	Comando inicial 2
2	<i>Device ID</i>	WORD	ID do dispositivo é padrão 0x0001
4	<i>Parameter</i>	DWORD	Parâmetro de entrada
8	<i>Command</i>	WORD	Código de comando
10	<i>Check Sum</i>	WORD	<i>Check Sum</i> é o somatório dos itens anteriores OFFSET[0]+...+OFFSET[9]= <i>Check Sum</i>

Fonte: Datasheet GT-511C3⁵.

Com base neste protocolo foi criada uma função de comando *Cmd_Resp_ID* (apêndice A), que será utilizada sempre que for necessário enviar ou receber comandos ou informações do leitor, pois como resposta retorna uma estrutura com a informação solicitada.

A **figura 10** apresenta alguns dos comandos disponíveis para o funcionamento do GT-511C3, o comando *Open* é responsável pela inicialização do leitor, que é necessária para o funcionamento correto do leitor, no trabalho é efetuada através de uma função *init_ID* (apêndice A). Além disso, também são utilizados os comandos *CmosLed* para controle do LED do sensor, *IsPressFinger* para verificar se o dedo está pressionado, *CaptureFinger* para fazer a captura da digital e por fim *Identify* que identifica a impressão capturada no sensor.

⁵ Traduzido pela autora.

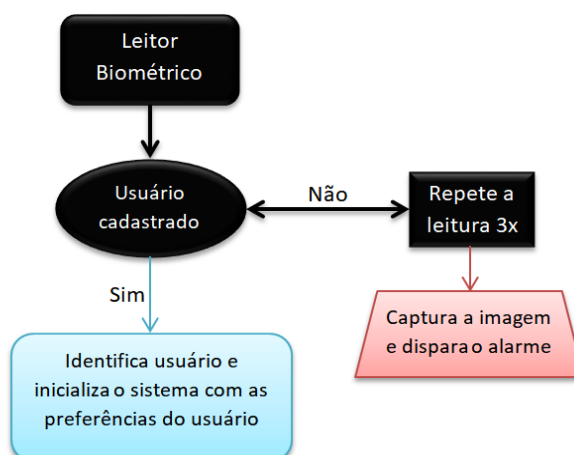
Figura 10 - Pacote de comando do GT-511C3

NÚMERO (HEX)	COMANDO	DESCRIÇÃO
01	<i>Open</i>	Inicializar
02	<i>Close</i>	Terminar
03	<i>UsbInternalCheck</i>	Verifica se o USB que está conectado é válido
04	<i>ChangeBaudrate</i>	Altera o <i>baud rate</i> da UART
12	<i>CmosLed</i>	Liga e desliga o LED do leitor
20	<i>GetEnrollCount</i>	Obtém o número de impressões cadastradas
21	<i>CheckEnrolled</i>	Verifica se a impressão está registrada
22	<i>EnrollStart</i>	Inicializa um registro
23	<i>Enroll1</i>	Faz a primeira leitura para registro
24	<i>Enroll2</i>	Faz a segunda leitura para registro
25	<i>Enroll3</i>	Faz a terceira leitura para registro e salva
26	<i>IsPressFinger</i>	Verifica se o dedo está pressionado no leitor
41	<i>DeleteAll</i>	Deleta todas impressões armazenadas
50	<i>Verify</i>	Verifica a ID específica de uma impressão
51	<i>Identify</i>	Identifica a impressão capturada no sensor
60	<i>CaptureFinger</i>	Captura a imagem (256x256) da impressão pressionada no sensor

Fonte: Datasheet GT-511C3⁶.

A **figura 11** representa a liberação de acesso através do leitor biométrico. Quando o usuário está cadastrado, o sistema identifica quem está acessando a residência, carrega suas configurações e após a identificação é realizada a checagem para saber se há outro usuário na residência. Caso haja, gera-se uma terceira identificação que será com preferências intermediárias entre os dois usuários existentes. Consequentemente, o ambiente é ajustado, através dessas informações, e envia para o RPi. Sendo assim, caso não haja um segundo usuário, são apenas carregadas as preferências do usuário que acessou a residência e, da mesma forma, as informações são enviadas para o RPi.

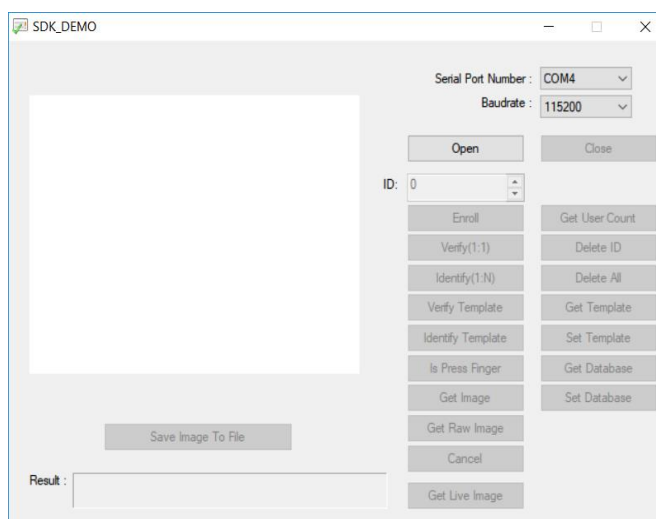
⁶ Traduzido pela autora.

Figura 11 - Liberação do leitor biométrico

Fonte: A autora.

Após a liberação do sistema as informações a respeito do usuário, dos sensores e da iluminação são enviadas através da comunicação WiFi para o RPi, que irá exibir na tela em tempo real todas as informações. Caso o usuário não seja reconhecido ou ocorra erro no posicionamento do dedo no leitor, este solicita novamente a identificação, repetindo a leitura até 3 vezes. Na terceira tentativa inválida o sistema dispara um alarme e gera um alerta de invasão capturando a imagem do indivíduo.

Para efetuar o cadastro das duas digitais usadas neste trabalho, foi utilizado o software de demonstração disponibilizado pela ADH-Tech **figura 12**. O software demo SDK para o módulo de scanner de impressão digital foi utilizado por meio de um adaptador USB para TTL. Com sua utilização foi possível, cadastrar, verificar e deletar as impressões utilizadas no trabalho.

Figura 12 - SDK Demo para utilização do leitor biométrico

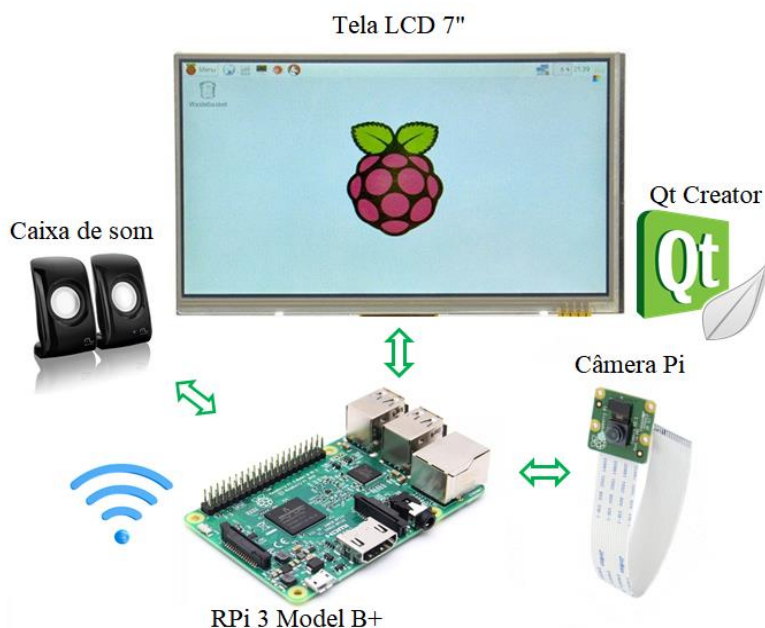
Fonte: Software ADH-Tech.

4.1.2 Funcionamento da comunicação WiFi

Como citado anteriormente, o módulo WiFi funciona através da configuração por comandos AT e essa inicialização se faz por comunicação UART. A configuração do módulo WiFi pode ser vista no apêndice A. O mesmo é configurado para utilizar uma rede WiFi local, ou seja, o módulo é conectado a um roteador, através dessa conexão é que se obtém a troca de informações do Cortex-M0 com o RPi. Tanto o módulo WiFi quanto o RPi se conectam a rede do roteador, o que os tornam clientes deste servidor local.

A **figura 13** apresenta o diagrama do sistema B e pode-se visualizar que é composto de: RPi 3 Model B+, tela LCD 7", caixa de som, câmera Pi e interface Qt, como citado anteriormente (item 4.1, **figura 7**).

Figura 13 - Diagrama do sistema B



Fonte: A autora.

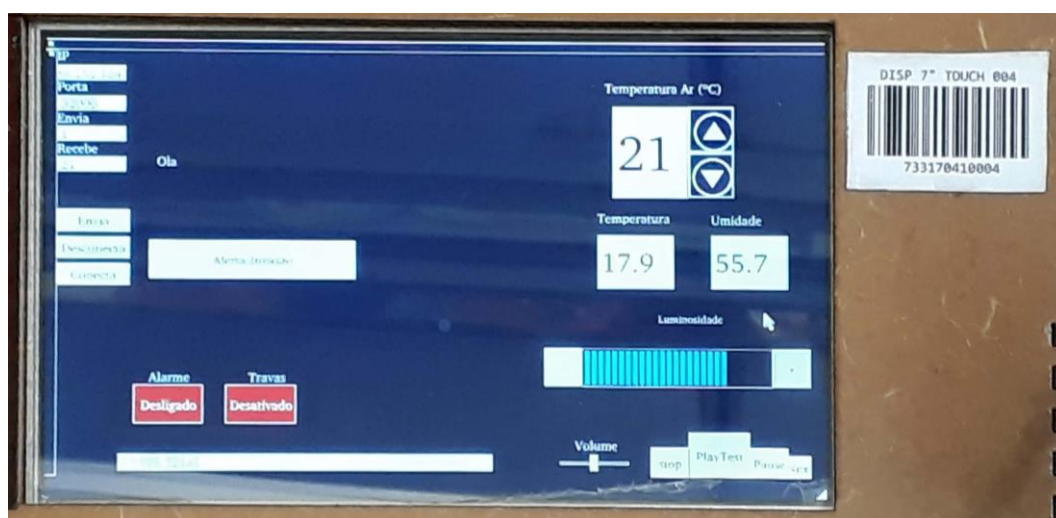
É na interface Qt embarcada no RPi que o usuário tem possibilidade de ajustar a iluminação, ligar e desligar o alarme, travar e destravar a casa, ver alertas de invasão (que, como explicado anteriormente, no caso de ocorrência irá tirar uma foto do indivíduo que tentou invadir a residência e exibe na tela como “alerta de invasão”), ajuste do som ambiente e, por fim, o ajuste da temperatura desejada. Toda essas informações são enviadas via comunicação WiFi, pois o RPi, que faz parte do sistema B, é o responsável pela IHM e o sistema A é a unidade de controle dos dados inseridos no RPi, pois todos sensores se encontram nele.

4.1.3 Desenvolvimento da interface com Qt Creator

Após a configuração da WiFi, foi desenvolvido uma interface utilizando o Qt Creator em linguagem C/C++ (apêndice B), que fica hospedado no RPi e é exibido na tela. Essa interface recebe todos comandos do usuário, além de mostrar as informações para o mesmo. O desenvolvimento teve suporte da documentação do próprio Qt⁷.

A **figura 14** apresenta a primeira versão da interface desenvolvida no Qt. Concluído o layout da interface Qt, foi realizada a comunicação entre o RPi e o Cortex-M0 através do módulo WiFi. Para essa comunicação utilizou-se no Qt a classe chamada *QTcpSocket*⁸, essa classe permite criar um ponteiro que conecta-se ao módulo WiFi através do IP e da porta, possibilitando escrever e ler informações do mesmo. Quando o módulo WiFi é configurado (**item 4.1.2**), é retornado o IP do módulo, e através dessa informação que a interface do RPi poderá se comunicar com o Cortex-M0. E a partir disto inicia-se a comunicação entre o Cortex-M0 e o RPi (**Figura 15**).

Figura 14 - Interface desenvolvida no Qt - versão A



Fonte: A autora.

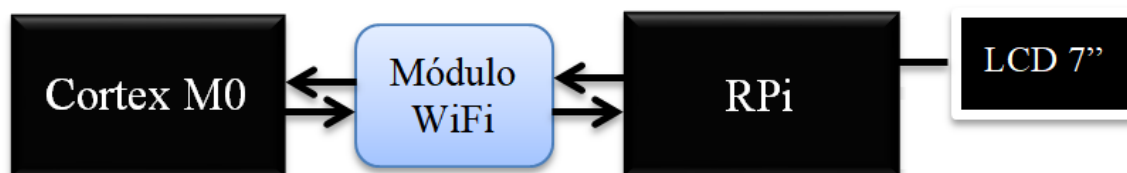
Como apresentado na **figura 15**, a comunicação entre o RPi e o Cortex-M0 funciona da seguinte maneira: o Cortex-M0 envia um vetor de dados para o módulo WiFi utilizando a comunicação UART; a partir disso essa informação é recebida pelo RPi, através do *Socket* implementado no Qt, utilizando o IP do módulo; como cada posição do vetor é um dado, o mesmo é separado e apresentado na tela ou manipulado no Qt. No caso do RPi, utiliza uma estrutura para enviar as informações para o Cortex-M0, através do *Socket* implementado no Qt,

⁷ Qt Documentation.

⁸ Qt 5.10.1 Reference Documentation.

as informações são enviadas para o WiFi, que no Cortex-M0, por interrupção lê as informações em uma estrutura, o motivo de um utilizar vetor e o outro utilizar estrutura é abordado no final deste capítulo.

Figura 15: Diagrama da comunicação entre o Cortex-M0 e o RPi



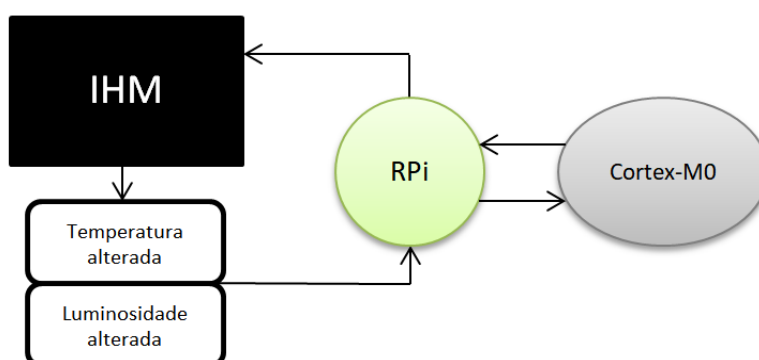
Fonte: A autora.

4.1.4 Controle da temperatura e luminosidade através das informações incluídas no RPi

A temperatura varia de 18°C a 28°C e sempre que o usuário alterar a temperatura do ambiente o RPi envia para o Cortex-M0 essa informação, pois o controle é feito no cortex M0, apenas a informação é inserida no RPi. O controle é feito comparando a temperatura ambiente, obtida do sensor utilizando comunicação I²C, com a temperatura inserida pelo usuário. Caso a informação seja menor que a temperatura ambiente, é acionado o relé 1 para simular o aquecimento, caso contrário, é acionado o relé 2 para simular o resfriamento.

A dimerização da luminosidade utiliza a lógica semelhante ao controle de temperatura. O usuário insere na tela a intensidade desejada, que é controlada através de uma barra progressiva que disponibiliza 8 níveis de iluminação. A partir desta seleção a informação é enviada ao Cortex-M0 que faz este controle. Tanto a temperatura como a luminosidade quando alteradas, são enviadas para unidade de controle e só serão atualizadas na tela depois que o Cortex-M0 realizar a modificação do sistema e retornar uma confirmação da alteração ao RPi, como mostra na **figura 16**.

Figura 16: Manipulação dos dados entre o Cortex M0 e o RPi



Fonte: A autora.

4.1.5 Desenvolvimento do sistema com câmera

Como citado anteriormente, foi utilizada uma câmera Pi no sistema. O comando *raspistill* é utilizado para capturar imagens com o módulo da câmera Pi, através da programação em linha de comando (RASPISTILL, 2018). Quando o leitor biométrico recebe 3 tentativas inválidas de acesso, é gerado um processo apresentado na **figura 17**, que captura a imagem e salva como “*tccAgoraVai.jpg*”(-o significa *output*, é necessário para salvar o arquivo em *.jpg*). No final do comando é preciso um & para que o processo seja gerado em segundo plano, caso contrário, a interface Qt é congelada até que este termine.

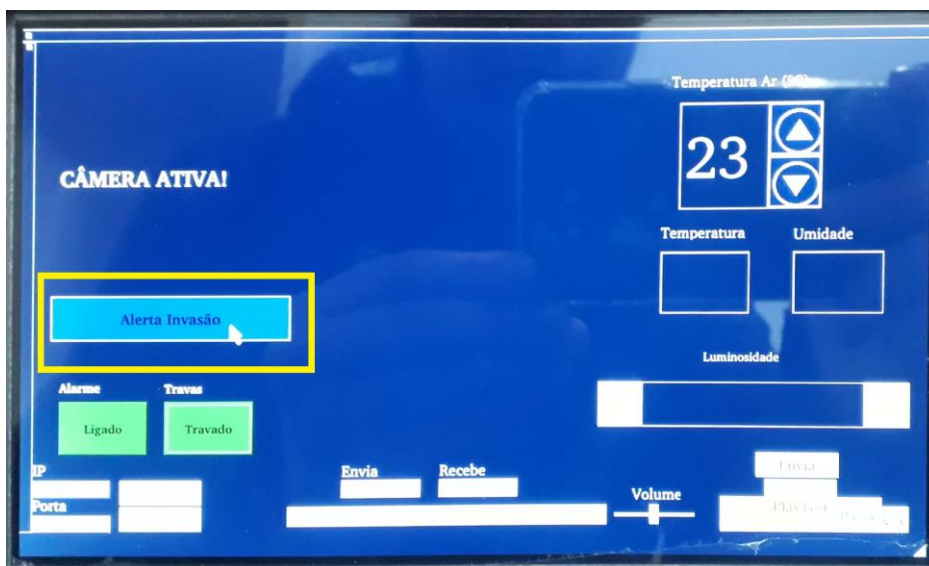
Figura 17 - Comando raspistill.

```
system("raspistill -o tccAgoraVai.jpg &");
```

Fonte: A autora.

Abaixo, visualiza-se o teste da câmera utilizando a versão A da interface. A **figura 18** apresenta um alerta de invasão na tela após realizar 3 tentativas de acesso inválidas no leitor biométrico. Além disso, o alarme também está tocando nesse momento e apenas os usuários cadastrados podem desligar o alarme.

Figura 18 - Teste câmera: alerta na tela



Fonte: A autora.

A **figura 19** apresenta o resultado da captura com a foto do indivíduo sem cadastro que tentou acessar o sistema.

Figura 19 - Captura do indivíduo após tentativa inválida de acesso



Fonte: A autora.

4.1.6 Construção de um Media Player na interface RPi

Na interface Qt também foi desenvolvido um reprodutor de áudio para som ambiente do sistema composto por quatro playlists sendo: Rock, Funk, Top Hits e Sertanejo. Para o seu desenvolvimento, foi utilizado o programa disponível no RPi chamado *omxplayer*, que é um reprodutor de áudio de linha de comando instalado no Raspbian (sistema operacional do RPi, semelhante ao Linux, que é um sistema operacional de código aberto).

No Qt são criados processos em segundo plano que controlam o reprodutor de áudio através de linha comando. É possível alterar a *playlist*, pausar, parar e passar a música e quando o usuário entra na casa é carregada a *playlist* de sua preferência, caso tenha-se dois usuários no ambiente é carregada uma *playlist* intermediária.

4.1.7 Sistema Inteligente

O diferencial do sistema é que ele se adapta a cada pessoa, ou seja, ele grava as últimas preferências do usuário e quando o mesmo acessa o sistema novamente é carregado essas informações ajustando tudo conforme salvo anteriormente. Para melhor compreensão, foram realizados testes a seguir, apresentando como o sistema se adapta.

A **figura 20** apresenta a versão final da interface desenvolvida no Qt e apenas o usuário 1 encontra-se na residência. Além disso, foi realizado uma demonstração de como o sistema se comporta quando se tem dois usuários utilizando o ambiente. A temperatura selecionada pelo usuário 1 foi 22°C e sem luminosidade, ou seja, desligada.

Figura 20 - Versão final da interface com usuário 1 na residência



Fonte: A autora.

Após a configuração o usuário 1 foi retirado do sistema e o usuário 2, como mostra na **figura 21**, foi inserido para selecionar suas preferências. A temperatura selecionada pelo usuário 2 foi 26°C e a luminosidade no máximo.

Figura 21 - Versão final da interface com usuário 2 na residência



Fonte: A autora.

Realizado as configurações, o usuário 1 foi inserido novamente no sistema e o resultado pode ser visto na **figura 22**. A partir das informações, o sistema fez a configuração de maneira intermediária para satisfazer ambos usuários.

Figura 22 - Versão final da interface com os dois usuário na residência



Fonte: A autora.

4.2 Dificuldades encontradas

Neste tópico apresenta-se, brevemente, as duas principais dificuldades encontradas no trabalho, bem como suas resoluções.

4.2.1 Perda de dados na comunicação

Um dos principais problemas encontrados no trabalho foi a perda de dados enviados pelo RPi para o Cortex-M0. Inicialmente, utilizou-se um vetor para enviar os dados do RPi para o Cortex-M0, mas em um determinado momento, o vetor deslocou uma posição, o que resultou no envio de lixo, ocasionando perda de dados. Além disso, depois da ocorrência deste problema, verificou-se uma falha no envio de dados do RPi para o Cortex-M0. Para a solução eficaz do problema utilizou-se uma estrutura de dados, assim, o RPi faz o envio de uma estrutura conhecida e o Cortex-M0 recebe a estrutura já prevendo o tamanho desse dado, evitando-se a perda de informações. No caso do envio de dados do Cortex-M0 para o RPi manteve-se a utilização de um vetor, pois o mesmo não apresentou nenhum problema nessa troca de informações.

4.2.2 Reprodução de música

Foram criadas duas interfaces de Media Player que funcionaram no Qt embarcado em Windows e quando carregadas no RPi não rodaram por incompatibilidade do sistema o que dificultou o seu desenvolvimento, pois apesar do Qt disponibilizar bibliotecas prontas de Media Player, algumas não funcionam no *Raspbian* se as configurações necessárias não estiverem atualizadas. Para a resolução do problema utilizou-se a função *system()*, que executa comandos do sistema, nesse contexto realizou-se em segundo plano um reprodutor de música do sistema operacional (item 4.1.6). Desta forma, quando, por exemplo, pressiona-se o botão *Play* gera-se um processo executado em linha de comando e, quando se deseja pausar a música um novo comando busca o processo gerado anteriormente e suspende a execução até que realize-se um outro comando, novamente, para continuar.

5 CONCLUSÃO

A tecnologia já faz parte do cotidiano da sociedade, ao realizar o desenvolvimento de um ambiente inteligente busca-se melhorar a qualidade de vida das pessoas, proporcionando bem-estar e novas experiências diárias dentro de suas residência. Além disso, com os avanços da automação residencial, é possível construir aplicações inteligentes com um custo mais acessível abrangendo um maior grupo de pessoas. Nesse contexto, o presente trabalho propôs explorar as aplicações com IoT para implementar um sistema inteligente possibilitando que um número maior de pessoas tenha acesso a essa tecnologia. Inicialmente, foram apresentados os conceitos base para entendimento do tema, por meio do referencial teórico conhecido, após, abordou-se a metodologia utilizada e as aplicações realizadas.

O estudo limitou-se a abranger apenas dois usuários, já cadastrados, e a aplicação somente à residências. Também, não utilizou-se de inteligência artificial para compor as escolhas do usuário, ou seja, não aplica-se o controle constante de preferências em um banco de dados, e sim, a gravação da última opção utilizada. Desta forma, o desenvolvimento do ambiente inteligente, através da tecnologia IoT, diferencia-se da automação residencial comum, pois é um sistema adaptável a cada usuário com a possibilidade de memorizar a última informação inserida. Sendo assim, quando acessado o ambiente novamente, a temperatura, iluminação e sonorização adequam-se às suas opções de conforto, que foram pré-definidas. Sendo assim, compreende-se que as tecnologias empregadas também possuem características relacionadas ao bem-estar do indivíduo.

Para responder ao primeiro objetivo, que buscava aplicar um sistema inteligente adaptável ao usuário, e ao segundo objetivo, de utilizar IoT para unificar os sistemas, através da troca de informações em tempo real via WiFi, utilizou-se de dois sistemas onde o denominado Sistema A possui o Cortex-M0 como unidade de controle, que a partir do código implementado em linguagem C, controla o módulo WiFi, luminosidade, alarme, temperatura, e o leitor biométrico que reconhece o usuário. E, o Sistema B, que utilizou o RPi para realizar a Interface Homem-Máquina, onde exibe-se as informações gravadas para o usuário através da tela de 7” conectada ao RPi. Também, para obter as informações de preferência do usuário, desenvolveu-se outro código criado em linguagem C/C++ para a criação da interface utilizando a IDE Qt, para a troca de informações entre Sistema A e Sistema B (do Cortex-M0 para o RPi, e vice-versa), do qual é realizada através da comunicação WiFi. Conforme abordado no referencial teórico, compreende-se que é nesse contexto que ocorre a IoT, visto que os sistemas estão conectados trocando informações em tempo real.

Para responder ao terceiro objetivo, que abordava transformar o sistema acessível para contemplar sua utilização por um grupo maior de pessoas, foi parcialmente alcançado, uma vez que não foi elaborada uma planilha de análise de custos, assim como, não se utilizou um estudo econômico visando o material de menor valor no mercado, e sim, os componentes disponíveis para a autora. Sendo assim, é sugerido para trabalhos futuros um estudo crítico em relação aos componentes similares aos utilizados no projeto, com o objetivo de melhorar a aplicação do sistema buscando seu menor custo.

Na aplicação da metodologia proposta, considerou-se o armazenamento da última preferência do usuário, apresentando os componentes e ferramentas utilizadas para efetuar a implementação dos softwares e a montagem do sistema. Desta forma, possibilita a utilização deste trabalho como ponto de partida para estudos futuros, abrindo possibilidades de aplicações como, por exemplo, o controle do ambiente por meio de análise do humor do usuário através de uma câmera que faz o reconhecimento facial das expressões. Outra indicação de potencial pesquisa é a de aplicação na área de inteligência artificial, desenvolvendo um sistema que percebe alterações e reage com base nas informações já integradas na memória, abordando ambientes inteligentes, utilizando sua aplicação em meios não residenciais (indústria, comércio, etc), tema que não foi tratado no projeto.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ASHTON, K. That “Internet of Things” Thing: In the Real World Things Matter More than Ideas. *RFID Journal*. Jun. 2009. Disponível em:
<<http://www.rfidjournal.com/articles/view?4986>> Acesso em: 20 mai 2018.

AUTOMAÇÃO RESIDENCIAL. **Tudo o que você precisa saber a respeito**. Disponível em:
<<https://www.tuacasa.com.br/automacao-residencial/>> Acesso em: 16 jun. 2018.

BOUCOUVALAS, A. C. KOSMATOS, E. A. TSELIKAS, N. D. Integrating RFIDs and Smart Objects into a Unified Internet of Things Architecture, *Advances in Internet of Things*, vol. 1, pp. 5-12, Apr. 2011.

CIRCUITBASICS. Basics of UART Communication. Disponível em:
<<http://www.circuitbasics.com/basics-uart-communication/>> Acesso em: 22 jun. 2018.

DATASHEET CORTEX-M0. **STMicroelectronics**: Disponível em:
<<http://www.st.com/en/microcontrollers/stm32f072rb.html>> Acesso em: 13 jun. 2018.

DATASHEET GT-551C3. **ADH-Tech** Documentation: Disponível em
<https://www.sparkfun.com/products/retired/11792?_ga=2.210759125.1304154238.1530190237-1187703193.1528992646> Acesso em: 18 jun. 2018.

DATASHEET HTS221. **STMicroelectronics**: Disponível em:
<<http://www.alldatasheet.com/view.jsp?Searchword=Hts221>> Acesso em: 13 jun. 2018.

GERSHENFELD, N.; KRIKORIAN, R.; COHEN, D. **The Internet Of Things**. New York: Scienti_c American, 2004.

IBGE. Nove entre dez usuários de Internet no país utilizam aplicativos de mensagens. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/20077-nove-entre-dez-usuarios-de-internet-no-pais-utilizam-aplicativos-de-mensagens.html>> Acesso em: 28 mai 2018.

IBM. How Do Sockets? Disponível em:
<https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_71/rzab6/howdosockets.htm> Acesso em: 18 jun. 2018.

INTERNET LIVE STATS. **Internet users in the world**. Disponível em
<<http://www.internetlivestats.com/>> Acesso em: 28 mai 2018.

JIANG, H; CAI, C.; MA, X.; YANG, Y.; LIU, J. **Smart home based on WiFi sensing: a survey**. *IEEE Access*, vol. 6, pp. 13 317–13 325, 2018.

JUST ASK GEMALTO. Disponível em <<https://www.justaskgemalto.com/br/como-funciona-um-leitor-de-impressoes-digitais/>> Acesso em: 27 mai. 2018.

KANASHIRO, Marta; UNIVERSIDADE DE SÃO PAULO. Biometria no Brasil e o Registro de Identidade Civil: novos rumos para identificação. Tese de doutorado, 2011.

LIMA E SILVA, Itallo Santos; KASCHNY, JR. **Aplicações do protocolo I2C em sistemas microcontrolados**. In: VII CONNEPI-Congresso Norte Nordeste de Pesquisa e Inovação. 2012.

MADAKAM, S.; RAMASWAMY, R.; TRIPATHI, S. **Internet of Things (IoT): A literature review**. J. Comput. Commun., vol. 3, no. 5, pp. 164-173, 2015.

MAGALHÃES, P. S., & SANTOS, H. D. **Biometria e Autenticação**. 4ª Conferência da Associação Portuguesa de Sistemas de Informação. Porto. Portugal, 2003.

MAKOFESKE, D. B., DONAHOO, M. J., CALVERT, K. L. **TCP/IP Sockets in C#: Practical Guide for Programmers**. 1ª edição. San Francisco: Morgan Kaufmann, 2004.

MERCADO LIVRE. **Kit SmartThings da Samsung**. Disponível em: <https://produto.mercadolivre.com.br/MLB-1049938914-samsung-smart-things-home-monitoring-kit-novo-importado-usa-_JM> Acesso em: 18 jun. 2018.

MERCADO LIVRE. **Raspberry Pi**. Disponível em <https://produto.mercadolivre.com.br/MLB-987920468-raspberry-pi3-pi-3-model-b-quadcore-12ghz-pronta-entrega-_JM> Acesso em: 18 jun. 2018.

NUGENT, C. D.; FINLAY, D. D; FIORINI. P.; TSUMAKI, Y.; PRASSLER E. **Home automation as a means of independent living**. IEEE Trans. Autom. Sci. Eng., vol. 5, no. 1, pp. 1-8, Jan. 2008.

ORACLE. JAVA Documentation: What Is a Socket? Disponível em: <<https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html>> Acesso em: 18 jun. 2018.

PRAKASH, S.; VENKATRAM, N. **Establishing efficient security scheme in home IOT devices through biometric fingerprint technique**. Indian Journal of Science and Technology. 2016.

QTCREATOR. **Qt 5.10.1** Reference Qt Documentation: Disponível em <<https://doc.qt.io/archives/qt-5.10/index.html>> Acesso em: 13 jun. 2018.

RASPBERRY PI. **Fundação RaspBerry Pi**. Disponível em <<https://www.raspberrypi.org/>> aAcesso em: 27 mar. 2018.

RASPISTILL. **Documentation Raspberry Pi Camera**. Disponível em: <<https://www.raspberrypi.org/documentation/raspbian/applications/camera.md>> Acesso em: 17 jun. 2018.

SAMSUNG. **Kit Smart things**. Disponível em: <<https://www.samsung.com/us/smart-home/smartthings/kits/home-monitoring-system---special-f-cen-hms-1/>> Acesso em: 18 jun. 2018.

SANTOS, Sandro. **Introdução a Iot: Desvendando a Internet Das Coisas** 1. ed. SS Trader, 2018.

SUN, C. **Application of RFID Technology for Logistics on Internet of Things**. AASRI Procedia, vol. 1, pp. 106-111, 2012.

TSE. **Tribunal de Justiça Eleitoral**. Disponível em <<http://www.tse.jus.br/eleitor/biometria>> Acesso em: 13 jun. 2018.

WORLDOMETERS. Current World Population: Disponível em: <<http://www.worldometers.info/world-population/>> Acesso em: 28 mai. 2018.

ZHANG, Li; YAN, Ruqiang; SUN, Hanghang; QIAN, Yuning. **A Wireless General Air-Conditioner Remote-Controller for Smart Homes**. Sensing Technology (ICST), 2017 Eleventh International Conference on 4-6 Dec. 2017.

APÊNDICE

A - CÓDIGO IMPLEMENTADO PARA O CORTEX-M0

- Principais Funções utilizadas no trabalho

```
//Controle do relé baseado na temperatura
void Controle_Temp()
{
    if (tempWant>temperatura)
    {
        HAL_GPIO_WritePin(GPIOC, (1<<9),1);//LIGA RELÉ 1 -> PORTA PC_9 //AQUECENDO
        HAL_GPIO_WritePin(GPIOC, (1<<8),0);//DESLIGA RELÉ 2 -> PORTA PC_8 //RESFRIA
    }
    else
    {
        HAL_GPIO_WritePin(GPIOC, (1<<9),0);//DESLIGA RELÉ 1 -> PORTA PC_9 //AQUECENDO
        HAL_GPIO_WritePin(GPIOC, (1<<8),1);//LIGA RELÉ 2 -> PORTA PC_8 //RESFRIA
    }
    // Fim do controle
}

// Led dimmer
void HAL_SYSTICK_Callback(void)
{
    if(perc==0) HAL_GPIO_WritePin(GPIOC, (1<<2),0);
    else if(perc==8) HAL_GPIO_WritePin(GPIOC, (1<<2),1);
    else if(cont<perc)
    {
        HAL_GPIO_WritePin(GPIOC, (1<<2),1);//led=1
        cont++;
    }
    else
    {
        HAL_GPIO_WritePin(GPIOC, (1<<2),0);//led=1
        cont++;
    }
    if (cont==8) cont=0;
}

// CONFIGURAÇÃO Wifi
void init_Wifi()
{
    HAL_UART_Transmit(&huart1, "AT\r", 3, 100);
    HAL_UART_Receive(&huart1, vetor_at, 11, 3000);
    HAL_UART_Transmit(&huart2, vetor_at, 11, 300);
    HAL_Delay(100);

    HAL_UART_Transmit(&huart1, "AT+SSIDTXT=embarcados\r", 24, 500);
    HAL_UART_Receive(&huart1, vetor_at2, 500, 1000);
    HAL_UART_Transmit(&huart2, vetor_at2, 500, 1000);
    HAL_Delay(100);

    HAL_UART_Transmit(&huart1, "AT+SCFG=wifi_wpa_psk_text,embarcados\r\n", 39,
500);
    HAL_UART_Receive(&huart1, vetor_at3, 500, 1000);
    HAL_UART_Transmit(&huart2, vetor_at3, 500, 1000);
    HAL_Delay(100);

    HAL_UART_Transmit(&huart1, "AT+SCFG=wifi_priv_mode,2\r", 27, 700);
```



```

HAL_UART_Receive(&huart1, vetor_at4, 1000, 10000);
HAL_UART_Transmit(&huart2, vetor_at4, 1000, 10000);
HAL_Delay(500);
HAL_UART_Transmit(&huart1, "AT+S.SCFG=wifi_mode,1\r", 22, 500);
HAL_UART_Receive(&huart1, vetor_at3, 500, 1000);
HAL_UART_Transmit(&huart2, vetor_at3, 500, 1000);
HAL_Delay(100);

HAL_UART_Transmit(&huart1, "AT&W\r", 5, 700);
HAL_UART_Receive(&huart1, vetor_at4, 1000, 10000);
HAL_UART_Transmit(&huart2, vetor_at4, 1000, 10000);
HAL_Delay(100);

HAL_UART_Transmit(&huart1, "AT+CFUN=1\r", 10, 700);
HAL_UART_Receive(&huart1, vetor_at4, 1000, 5000);
HAL_UART_Transmit(&huart2, vetor_at4, 1000, 5000);
HAL_Delay(20000);

HAL_UART_Transmit(&huart1, "AT+S.STS=ip_ipaddr\r", 19, 700);
HAL_UART_Receive(&huart1, vetor_at4, 500, 5000);
HAL_UART_Transmit(&huart2, vetor_at4, 500, 5000);
HAL_Delay(500);

HAL_UART_Transmit(&huart1, "AT+S.SOCKD=32000\r", 17, 700);
HAL_UART_Receive(&huart1, vetor_at4, 1000, 5000);
HAL_UART_Transmit(&huart2, vetor_at4, 1000, 5000);
HAL_Delay(100);
}

```

//FUNÇÕES DO LEITOR

void init_ID(void)

```

{
    Cmd_Resp_ID(0, Open); //Open
    HAL_Delay(300);
    sprintf(vaux, "\n Init okay \n");
    HAL_UART_Transmit(&huart2, vaux, strlen(vaux), 100);
    Cmd_Resp_ID(1, CmosLed); //LED On
    HAL_Delay(100);
}

```

char Cmd_Resp_ID(uint32_t Param, uint16_t Cmd)

```

{
    char Resp = 0;
    unsigned long ChkSum=0;

    //COMMAND PACKET
    //Command start code1
    ID_cmd.b1 = 0x55;
    ChkSum += ID_cmd.b1;
    //Command start code1
    ID_cmd.b2 = 0xAA;
    ChkSum += ID_cmd.b2;
    //Device ID: default is 0x0001, always fixed
    ID_cmd.devId = 0x0001;
    ChkSum += ID_cmd.devId;
    //Input Parameter

```

```

ID_cmd.parameterID =Param;
ChkSum += ID_cmd.parameterID;
//Command Code
ID_cmd.wCmd_Resp = Cmd;
ChkSum += ID_cmd.wCmd_Resp;
//Checksum (byte addition)
// O check sum é o somatório de todos os OFFSETS
ID_cmd.wChkSum = ChkSum;
//Envia a Struct da comando completa
HAL_UART_Transmit(&huart4, (uint8_t *)&ID_cmd, sizeof(ID_struct), 1000);
//Response Packet (Acknowledge)
//A resposta vem como struct tbm
HAL_UART_Receive(&huart4, (uint8_t *)&ID_resp, sizeof(ID_struct), 1000);
//HAL_UART_Receive(&huart4, vaux, strlen(vaux), 2000);
if(ID_resp.wCmd_Resp == ACK) // OUTPUT PARAMETER
{
    Resp = 1;
}
else if(ID_resp.wCmd_Resp == NACK) //ERROR CODE
{
    Resp = 0;
}
return Resp;
}

```

```

void check_ID(void)

```

```

{

    Cmd_Resp_ID(1, CmosLed); //LED On

    if(Cmd_Resp_ID(0, IsPressFinger) && ID_resp.parameterID == 0)
    {
        Cmd_Resp_ID(0, CaptureFinger); //Fast Capture Fingerprint
        if(Cmd_Resp_ID(0, Identify))
        {
            enroll=1;
            if(ID_resp.parameterID==1)
            {
                sprintf(vaux, "\nuser 1\n");
                HAL_UART_Transmit(&huart2, vaux, strlen(vaux), 100);
                HAL_GPIO_WritePin(GPIOC, (1<<5), 0);
                buzzer=0;
                if(usuario==0) {
                    usuario=1;
                    tempWant=tUser1;perc=lumUser1;}

                else if(usuario==2)
                {usuario=3;
                    tempWant=(tUser1+tUser2)/2;
                    perc=(lumUser1+lumUser2)/2;}

                else if(usuario==1)
                {usuario=0;tempWant=23;perc=0;}
            }
        }
    }
}

```

```

        else if(usuario==3)
        {usuario=2;tempWant=tUser2;perc=lumUser2;}
        HAL_Delay(1000);
        Cmd_Resp_ID(0,CmosLed);
        HAL_Delay(1000);
    }

    else if(ID_resp.parameterID==0)
    {
        HAL_GPIO_WritePin(GPIOC, (1<<5),0);
        buzzer=0;
        sprintf(vaux,"Teste 0! ID:%u",ID_resp.parameterID);
        if(usuario==0)
        {usuario=2;tempWant=tUser2;perc=lumUser2;}
        else if(usuario==1)
        {usuario=3; tempWant=(tUser1+tUser2)/2;
        perc=(lumUser1+lumUser2)/2;}

        else if(usuario==2)
        {usuario=0;tempWant=23;perc=0;}
        else if(usuario==3)
        {usuario=1;tempWant=tUser1; perc=lumUser1;}
        HAL_Delay(1000);
        Cmd_Resp_ID(0,CmosLed);
        HAL_Delay(1000);
    }
}
else
{
    buzzer++;
    if(alarme==1)
    {
        if (buzzer==3)
        {
            if(pix==1)pix=0;
            else pix=1;
            HAL_GPIO_WritePin(GPIOC, (1<<5),1);
        }
    }
}
} // fim if
}

```

B - CÓDIGO IMPLEMENTADO PARA O RPi (Qt)

```

//WIFI CONECTA
void MainWindow::on_botaoConecta_clicked()
{
    QString ip;
    int port;
    ip=ui->ip->text();
    port=ui->porta->text().toInt();
    wifi->connectToHost(ip, port);
    if(wifi->waitForConnected(3000))
    {
        qDebug() << "Conectado!";
        wifi->write("Oi servidor!\r\n");
        wifi->waitForBytesWritten(1000);
    }
    else
    {
        qDebug() << "Not connected!";
    }
}

//RECEBE INFOS DO CORTEX
void MainWindow::teste()
{
    QByteArray dadoRX, temp, umid, teste, usuario,lumi,alarme;
    int user, lum, pic;
    dadoRX = wifi->readAll();
    //Teste
    ui->recebeTeste->del();
    ui->recebeTeste->setText(dadoRX);
    //Mostra Tremperatura
    temp = dadoRX.left(4);
    ui->recebeTemp->del();
    ui->recebeTemp->setText(temp);
    //Mostra Umidade
    umid = dadoRX.mid(4,4);
    ui->recebeUmid->del();
    ui->recebeUmid->setText(umid);
    //Testando
    teste = dadoRX.mid(8,2);
    graus = teste.toInt();
}

```

```
ui->recebeTexto->del();  
ui->tempControll->setText(teste);  
ui->recebeTexto->setText(teste);  
//Luminosidade  
lumi = dadoRX.mid(10,1);  
lum = lumi.toInt();  
lumAux=lum*12;  
ui->progressBar->setValue(lumAux);  
}
```