

**UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

DIORGE RAMON ANDRADE BRITO

**CONTROLE DO ACESSO, AUTOMATIZAÇÃO DA CLIMATIZAÇÃO E DA
ILUMINAÇÃO DE UMA SALA DE AULA, UTILIZANDO O
MICROCOMPUTADOR DE BAIXO CUSTO *RASPBERRY PI*.**

MANAUS

2018

DIORGE RAMON ANDRADE BRITO

**CONTROLE DO ACESSO, AUTOMATIZAÇÃO DA CLIMATIZAÇÃO E DA
ILUMINAÇÃO DE UMA SALA DE AULA, UTILIZANDO O
MICROCOMPUTADOR DE BAIXO CUSTO *RASPBERRY PI*.**

Projeto de pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentado à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas como requisito para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Me. Walfredo Lucena da Costa Filho

MANAUS

2018

Universidade do Estado do Amazonas – UEA
Escola Superior de Tecnologia - EST

Reitor:

Cleinaldo de Almeida Costa

Vice-Reitor:

Cleto Cavalcante de Souza Leal

Diretor da Escola Superior de Tecnologia:

Roberto Higino Pereira da Silva

Coordenadora do Curso de Engenharia Elétrica:

Ingrid Sammyne Gadelha Figueiredo

Banca Avaliadora composta por:

Data da defesa: 22/11/2018.

Prof. Walfredo Lucena da Costa Filho (Orientador)

Prof. Israel Gondres Torné

Prof. Ingrid Sammyne Gadelha Figueiredo

CIP – Catalogação na Publicação

Brito, Diorge Ramon Andrade

Controle do acesso, automatização da climatização e da iluminação de uma sala de aula, utilizando o microcomputador de baixo custo *Raspberry Pi* / Diorge Ramon Andrade Brito; [orientado por] Walfredo Lucena da Costa Filho. – Manaus: 2018.

69 f. p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica). Universidade do Estado do Amazonas, 2018.

1. Raspberry Pi. 2. Arduino. 3. Módulo RFID. 4. Infravermelho. 5. Banco de Dados. 6. *Firebase*. I. Costa Filho, Walfredo Lucena da.

DIORGE RAMON ANDRADE BRITO

CONTROLE DO ACESSO, AUTOMATIZAÇÃO DA CLIMATIZAÇÃO E DA
ILUMINAÇÃO DE UMA SALA DE AULA, UTILIZANDO O MICROCOMPUTADOR DE
BAIXO CUSTO RASPBERRY PI.

Projeto de pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentado à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas como requisito para obtenção do título de Engenheiro Eletricista.

Nota obtida: _____ (_____)

Aprovada em: 22/11/2018

Área de concentração: Sistemas Microcontrolados

BANCA EXAMINADORA

Orientador: Prof. Walfredo Lucena da Costa Filho, Me.

Avaliador: Prof. Israel Gondres Torné, Dr.

Avaliador: Prof^ª. Ingrid Sammyne G. Figueiredo, Me.

Manaus, 2018.

DEDICATÓRIA

Aos meus pais Elias e Rosilene, que durante anos sonharam e trabalharam por um futuro digno para mim e minhas irmãs.

À minha amada esposa Annie e aos meus filhos Stefannie, Sofia e Filipe, razões de todo o meu esforço e dedicação.

AGRADECIMENTOS

Agradeço primeiramente a Deus e a Jesus Cristo, razões pela qual nossa existência não seria possível e que me abençoaram todos os dias até hoje, dando-me sabedoria e discernimento durante as dificuldades e barreiras impostas.

Agradeço aos meus pais Elias e Rosilene, que forjaram o meu caráter e minha disciplina, trabalhando diuturnamente para que nossa família permanecesse unida e que os filhos tivessem a melhor educação.

Agradeço à minha esposa Annie, companheira, amiga e mãe, que durante os anos da graduação, muitas vezes cuidou sozinha dos nossos filhos nos intervalos em que estive ausente, para que eu pudesse obter o sucesso durante esta empreitada na vida acadêmica.

Agradeço ao meu professor e mestre Walfredo Lucena, pela ajuda concedida durante as etapas de desenvolvimento desse projeto e por todo o conhecimento científico adquirido durante a graduação.

Agradeço aos meus pastores Wilson e Maura, juntamente com sua família, pelas orações durante todo o progresso da graduação e elaboração deste projeto.

Agradeço ao meu amigo Bernardo Alecrim, pela ajuda na implementação do código-fonte, desenvolvimento da interface web e pela enorme dedicação a este projeto diante de todas as dificuldades.

Agradeço aos meus colegas de trabalho da Petróleo Brasileiro S/A, que durante as jornadas de trabalho e no intervalo entre elas, contribuíram para que eu pudesse continuar o desenvolvimento deste projeto.

Finalmente agradeço aos meus amigos e colegas da UEA que me acompanharam durante toda a graduação e sem os quais muitas das vitórias alcançadas não seria possível.

“O temor do Senhor é o princípio do conhecimento; os loucos desprezam a sabedoria e a instrução.”

Provérbios 1:7

RESUMO

O presente trabalho propõe construir um controle automatizado de uma sala de aula, desde o acesso à sala, até o controle do sistema de refrigeração (ar-condicionado). Por trás do escopo do projeto, surgiu a idéia de facilitar o trabalho hoje realizado por apenas um funcionário da Escola Superior de Tecnologia (EST), o qual tem a responsabilidade de abrir todas as salas nos horários iniciais das aulas, operando também o funcionamento de todos os condicionadores de ar das salas, e, após o término, fechar todas as salas e desligar os equipamentos. Como “cérebro” de todo este projeto, está o microcomputador Raspberry Pi. Ele irá controlar o acesso à porta da sala através de uma trava por eletroímã o qual é liberada por um módulo relé, controlará também, o ar-condicionado através de um módulo emissor de luz infravermelha e as luzes através de outro módulo relé. O diferencial desse circuito está no tratamento dos dados. Um aplicativo construído na plataforma iOS, irá guardar e/ou cadastrar os usuários, que terão acesso às salas, em um banco de dados contido em uma nuvem virtual na plataforma de construção de aplicativos, *Firebase*. Para funcionar como “drive” para o uso dos periféricos será usada a plataforma de prototipagem eletrônica Arduino. Todo o acesso às salas será realizado de duas formas: por um módulo *Radio Frequency Identification* (RFID) e pelo aplicativo iOS. Cada funcionário terá um cartão ou tag RFID cadastrado no seu nome e também um login para o aplicativo de controle. A função do aplicativo é que, no caso da ausência do cartão, o funcionário possa ainda assim, conseguir o acesso à sala e controle dos sistemas.

Palavras-chave:

Raspberry Pi – Arduino – Módulo RFID – Infravermelho – Banco de dados – *Firebase*.

ABSTRACT

The present work proposes to construct an automated control of a classroom, from the access to the room, to the control of the refrigeration system (air conditioning). Behind the scope of the project came the idea of facilitating the work today carried out by only one employee of the Higher School of Technology (EST), who has the responsibility to open all the rooms in the opening hours of classes, also operating the all air conditioners in the rooms, and, after completion, close all rooms and turn off the equipment. As the "brain" of this whole project, there is the Raspberry Pi microcomputer. It will control access to the room door through a latch by electromagnet which is released by a relay module, will also control the air conditioning through an infrared light emitting module and lights through another relay module. The differential of this circuit is in the data processing. An application built on the iOS platform will store and / or register users who will have access to the rooms in a database contained in a virtual cloud on the application building platform, Firebase. To operate as a drive for the use of peripherals will be used the Arduino electronic prototyping platform. All access to the rooms will be done in two ways: a Radio Frequency Identification (RFID) module and the iOS application. Each employee will have a card or RFID tag registered in their name and also a login to the control application. The function of the application is that, in case of absence of the card, the employee can still gain access to the room and control the systems.

Keywords:

Raspberry Pi – Arduino – RFID Module – Infrared – Database – Firebase.

ÍNDICE DE ILUSTRAÇÕES

Figura 1 - Selo Procel	15
Figura 2 - Raspberry Pi.....	16
Figura 3 - Máquina Virtual do Raspbian dentro do Windows	17
Figura 4 - MIT Stretch.....	18
Figura 5 - Tabela de preços dos modelos produzidos.	19
Figura 6 - Tabela de especificações técnicas dos modelos lançados.....	20
Figura 7 - Kit de aprendizado para iniciantes, modelo Pi-Zero.....	20
Figura 8 - Arduino Uno	23
Figura 9 - Espectro visível.....	24
Figura 10 - LED Infravermelho super-brilho	25
Figura 11 - Módulo emissor de luz infravermelha	26
Figura 12 - Módulo Receptor Infravermelho.	27
Figura 13 - Sinal modulado e sinal modulante.....	28
Figura 14 - Módulo RFID, cartão e etiqueta	28
Figura 15 - Cartão RFID.....	29
Figura 16 - Modulo leitor biométrico USB	30
Figura 17 - Leitor biométrico para Arduino.	31
Figura 18 - Aprimorando Apps	32
Figura 19 - <i>layout</i> básico do projeto.....	34
Figura 20 - Layout do <i>RoomController</i>	38
Figura 21 - Pinagem do módulo RFID	40
Figura 22 - <i>Pinout</i> Arduino nano.....	40
Figura 23 - <i>Pinout</i> módulo relé 2 canais.	42

ÍNDICE DE TABELAS

Tabela 1 – Conexão entre o módulo RC522 e o Arduino.	40
Tabela 2 – Conexão entre o módulo receptor de luz infravermelha e o Arduino.....	41
Tabela 3 – Conexão entre o módulo emissor de luz infravermelha e o Arduino.	41
Tabela 4 - Conexão entre o módulo relé de 2 canais, a iluminação e a trava por eletroímã. ...	42
Tabela 5 - Conexões entre o módulo relé e o Raspberry Pi.	42
Tabela 6 - Custo dos itens do projeto.	43
Tabela 7 - Comparativo entre o projeto e as marcas Intelbras.	43

SUMÁRIO

INTRODUÇÃO	12
1 REFERENCIAL TEÓRICO	14
1.1 EFICIÊNCIA ENERGÉTICA.....	14
1.2 RASPBERRY PI	16
1.3 BANCOS DE DADOS (BD'S).....	21
1.4 ARDUINO.....	22
1.5 INFRAVERMELHO	24
1.5.1 Diodo Emissor de Luz (LED) infravermelha.....	25
1.5.2 Módulo Emissor de Luz Infravermelha	26
1.5.3 Módulo Receptor de Luz Infravermelha.....	27
1.6 MÓDULO <i>RADIO FREQUENCY IDENTIFICATION</i> (RFID).....	28
1.7 MÓDULO LEITOR DE IMPRESSÃO DIGITAL (BIOMÉTRICO).....	29
1.8 FIREBASE	31
2 METODOLOGIA DE DESENVOLVIMENTO	33
2.1 LAYOUT DO PROJETO.....	34
2.2 CONTRUÇÃO DO CÓDIGO-FONTE.....	35
2.2.1 Desenvolvimento da lógica do código-fonte	35
2.2.2 Construção do Aplicativo iOS	37
3 CONSTRUÇÃO DO CIRCUITO PROTÓTIPO	39
3.1 MATERIAIS UTILIZADOS.....	39
3.2 MONTAGEM DOS PERIFÉRICOS	39
3.2.1 Montagem do módulo RFID.....	39
3.2.2 Montagem dos módulos emissor e receptor de luz infravermelha.....	41
3.2.3 Montagem do módulo relé 2 canais.	42
4 LEVANTAMENTO DE CUSTOS.....	43
5 RESULTADOS.....	44
CONCLUSÃO.....	45
REFERÊNCIAS BIBLIOGRÁFICAS	46
OBRAS CONSULTADAS	50
APÊNDICE A – FLUXOGRAMA DE FUNCIONAMENTO.....	51
APÊNDICE B – CÓDIGO FONTE DO PROJETO	52
APÊNDICE C – CÓDIGO EM PYTHON PARA O RASPBERRY PI	58
ANEXO A – CÓDIGO TESTE DO MÓDULO RECEPTOR INFRAVERMELHO	60
ANEXO B – CÓDIGO TESTE DO MÓDULO EMISSOR DE INFRAVERMELHO ..	64
ANEXO C – CÓDIGO TESTE DO MÓDULO RFID.....	65

INTRODUÇÃO

A crescente demanda por energia elétrica tem levado as concessionárias a grandes investimentos voltados para infraestrutura, geração, transmissão e distribuição, afim de atender às crescentes necessidades do mercado.

Em todas as vezes, esses investimentos visam apenas ofertar uma quantidade maior de energia elétrica em prol do crescimento exponencial do consumo sem se preocupar na maioria das vezes o quanto de energia realmente está sendo utilizada e como ela está sendo utilizada. A relação entre demanda e consumo é pauta comum em empresas que buscam expandir seus negócios.

De caráter nacional, há programas que incentivam ao consumo racional de energia elétrica. Em 1985, foi criado o Programa Nacional de Conservação de Energia Elétrica, (PROCEL) (ELETROBRAS, 2017?), executado pela Eletrobras com recursos da própria empresa, da Reserva Global de Reversão (RGR) e de entidades internacionais.

Em 2016, o PROCEL contribuiu para a economia de 15,1 bilhões de quilowatts-hora (kWh), o equivalente a 3,29% de todo o consumo nacional de energia elétrica naquele ano. O resultado representa aproximadamente 7,8 milhões de residências brasileiras. Segundo o levantamento da Eletrobras, os ganhos com o programa também abrangeram os impactos ambientais, que foram reduzidos nas emissões de CO₂ da ordem de 1,238 milhão de toneladas equivalentes, o que corresponde às emissões de aproximadamente 425 mil veículos no intervalo de um ano.

Dentro do PROCEL há subprogramas voltados para diversas áreas, abrangendo desde gestão energética municipal (PROCEL GEM) até a conscientização para o uso racional e eficiente de energia elétrica (PROCEL Educação).

Diante do exposto, torna-se cada vez mais comum o surgimento de projetos que visam a eficiência energética atuando sobre o uso racional de energia elétrica, alternativas de energias (energias renováveis) e novas tecnologias para tal emprego. Tomando-se por exemplo, a Escola Superior de Tecnologia da Universidade do Estado do Amazonas, temos um problema decorrente dos altos gastos com o uso desordenado de energia elétrica, além dos inúmeros furtos realizados, de equipamentos de uso acadêmico e administrativo. Baseado nisso, esta proposta mostra a possibilidade da construção de um sistema automatizado para cada sala de aula, onde o acesso pudesse ser controlado por um sistema inteligente e *on-line*, que garantisse também, o controle dos sistemas de refrigeração e iluminação da sala de aula, diminuindo assim os gastos com energia elétrica, causados por negligência.

Sob a ótica da segurança patrimonial, este projeto também visa minimizar ou até erradicar os furtos de equipamentos das salas e dos laboratórios da Escola Superior de Tecnologia. A partir do momento em que cada funcionário possui seu próprio meio de acesso, o respaldo jurídico para o próprio funcionário e para a instituição é automático. O motivo é que, todo o acesso é registrado em um banco de dados facilitando assim, para a Universidade do Estado do Amazonas, a identificação e o acesso somente a usuários permitidos.

1 REFERENCIAL TEÓRICO

Almejar a eficiência energética é um consenso comum nas empresas que atuam nas mais diversas áreas como: indústria, transporte, construção civil e arquitetura. Esses são apenas alguns exemplos onde este conceito é bastante difundido.

Em pesquisa realizada pelo Conselho Americano para uma Economia de Energia Eficiente (ACEEE) que promoveu um ranking de eficiência energética entre as doze maiores economias do mundo (BARBOSA, 2016), constatou-se que o Brasil não usufrui o melhor da energia que consome. Ocupando a décima posição no ranking, o país teve avaliado sua eficiência energética que deixou muito claro que ainda há muito que alcançar.

Investimentos e incentivos governamentais obtiveram a pior nota dentre os países avaliados. A falta de política e legislações que permitam o fomento e disseminação do uso consciente de energia deixou o Brasil entre os últimos colocados no ranking avaliado.

1.1 EFICIÊNCIA ENERGÉTICA

Eficiência energética é o uso racional de energia (redução do consumo), mantendo o mesmo nível de produção de serviços e/ou produtos, ou ainda, mantendo o consumo, aumentar o nível de produção dos mesmos serviços e/ou produtos. Ou seja, fazer mais com menos energia (ABESCO, 200?).

A eficiência energética se traduz em sempre tentar melhorar o uso das fontes de energia. Em todas as atividades desenvolvidas por uma sociedade há um uso extenso e intenso de uma ou mais formas de energia (INEE, 201-?). Por esse motivo a busca por melhorias na produção de determinado serviço tem se expandido pelo mundo inteiro nos últimos anos.

Alternativas para as fontes primárias de energia tem se tornado cada vez mais comum em países cuja matéria prima fóssil é escassa. O largo investimento nessas fontes alternativas de energia primária tem “trivializado” o conceito de que o crescimento econômico de uma nação também passa pelo uso racional de sua energia.

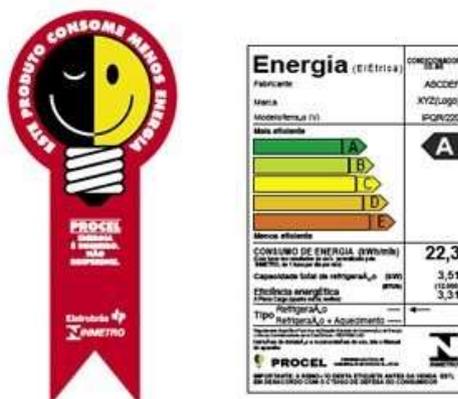
Países europeus, como a Inglaterra e a Alemanha, já possuem cerca de 20% da sua matriz energética composta por fontes alternativas de energia. Isso se refere ao fato das políticas de incentivo ao uso de fontes renováveis de energia serem abundantes e difundidas. Além disso, espera-se até 2050, uma redução de cerca de 85% das emissões de gases do efeito estufa, aumento para cerca de 30% ou mais no uso de energias renováveis e compartilhamento de energia entre os países da Europa da ordem de 15% (UNIÃO EUROPÉIA, 201-?).

O Programa Nacional de Conservação de Energia Elétrica (PROCEL), foi criado para promover o uso eficiente de energia elétrica no Brasil, combatendo o desperdício, reduzindo os custos e os investimentos no setor (ELETROBRAS, 2017?). Ele também foi o carro chefe para diversos outros programas:

- a) PROCEL GEM – Gestão Energética Municipal;
- b) PROCEL SANEAR – Eficiência Energética no Saneamento Ambiental;
- c) PROCEL EDUCAÇÃO – Informação e Cidadania;
- d) PROCEL INDÚSTRIA – Eficiência Energética Industrial;
- e) PROCEL EDIFICA – Eficiência Energética em Edificações;
- f) PROCEL EPP – Eficiência Energética em Prédios Públicos;
- g) PROCEL RELUZ – Eficiência Energética na Iluminação Pública e Sinalização Semafórica.
- h) SELO PROCEL – Eficiência Energética em Equipamentos;
- i) PROCEL INFO – Centro Brasileiro de Informação de Eficiência Energética;

O selo e a etiqueta PROCEL, ilustrados na figura 1, utilizado em equipamentos, pertence ao Programa Brasileiro de Etiquetagem (PBE), que garante ao consumidor que o equipamento utilizado possui características de eficiência energética. O uso do selo dá ao fabricante e ao consumidor garantia da qualidade do produto. Para a aquisição do selo, o fabricante deve primeiro se enquadrar nas mais diversas normas e legislações exigidas pelo Instituto Nacional de Metrologia, Qualidade e Tecnologia (INMETRO).

Figura 1 - Selo Procel



(Fonte: Consumo em Pauta, 2018?)

1.2 RASPBERRY PI

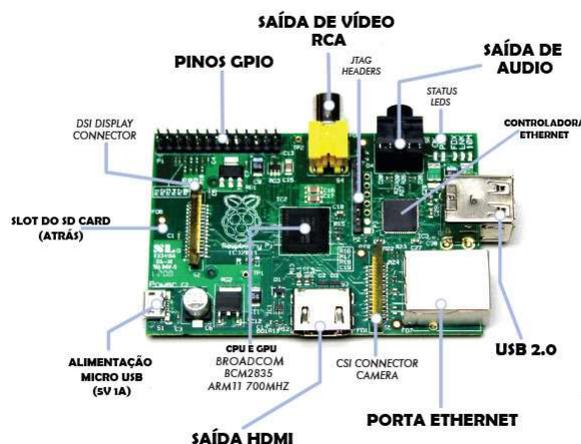
Com protótipos desenvolvidos em meados de 2006, o microcomputador *Raspberry Pi*, foi proposto pelos seus idealizadores como uma forma de incentivo ao uso da tecnologia de programação e da ciência da computação (CIRIACO, 200-?).

Criado por membros do Laboratório de Computação da Universidade de Cambridge, no Reino Unido, o microcomputador *Raspberry Pi* tinha o objetivo de interromper a crescente falta de interesse dos estudantes por ciência da computação na Inglaterra (MACHADO, 2012). Além de tentar despertar o interesse dos graduandos, o microcomputador também viria com uma proposta inclusiva, pois o seu preço proposto não ultrapassaria a faixa de US\$ 50 (cinquenta dólares).

Hoje é possível encontrar no mercado modelos que variam na faixa de US\$ 25 (vinte e cinco dólares) a US\$ 35 (trinta e cinco dólares).

Uma das características mais marcantes do *Raspberry Pi* (Figura 2) é o seu tamanho. Com dimensões que não ultrapassam as de um cartão de crédito, o *Raspberry Pi* é um computador como outro qualquer. Podemos conectar a ele, um mouse, um teclado, uma saída de áudio e até um monitor. Com tantas facilidades atreladas ao *Raspberry Pi*, ainda há nele conexões que permitem usá-lo como microcontrolador. O incentivo ao uso de programação também foi uma característica pensada pelos seus desenvolvedores, a possibilidade do uso de linguagens como o *Python*, *Scratch* e *Java*, deixou o *Raspberry Pi* cheio de opções que à primeira vista pareceria um campo complexo, porém, o tornou mais peculiar e atrativo.

Figura 2 - Raspberry Pi



(Fonte: Fundação Raspberry Pi, 2018)

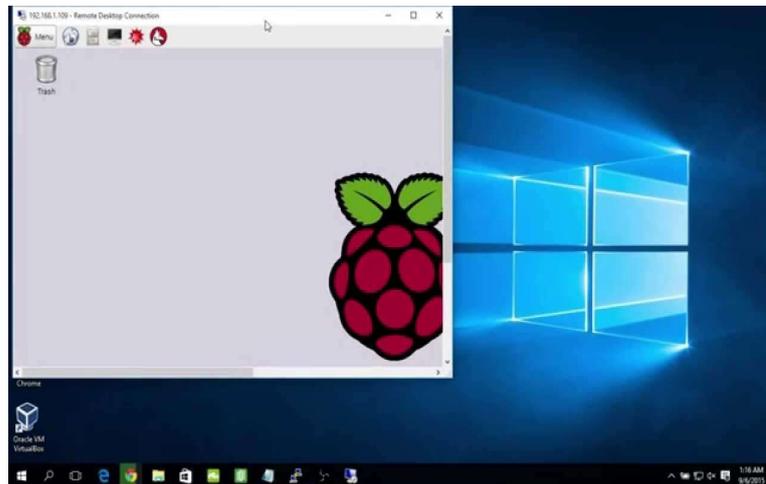
Projeto elaborado pela *Raspberry Pi Foundation*, o microcomputador conta hoje com modelos que possuem conexão *Universal Serial Bus (USB)*, *Bluetooth*, *Wireless*, entradas micro-usb e slot de cartão de memória *Secure Digital (SD)*, *MultiMedia Card (MMC)* e *Secure Digital Card Input-Output (SDIO)*, além de saídas *GPIO* e *High Definition Multimedia Interface (HDMI)*. Devido a multi-operacionalidade do *Raspberry Pi* que ele foi a escolha óbvia para a elaboração deste projeto.

A base do projeto *Raspberry Pi* é o sistema operacional *Linux*. Como a idéia era que o microcomputador fosse acessível a todos, “amarrar” o dispositivo com uma licença para uso seria um passo atrás no objetivo dos fundadores uma vez que o *Linux* é um *software* livre.

A escolha do *GNU/Linux* não foi por acaso, o *software* é um sistema operacional de código aberto, acessível a todos aqueles que quiserem contribuir com melhorias (CORSO, 200-?). Assim como o *Raspberry Pi*, o *Linux* não foi desenvolvido para fins lucrativos e por isso não é necessário comprar uma licença para utilizá-lo. Com isso, os fundadores desenvolveram o próprio sistema operacional baseado no *Linux* para o *Raspberry Pi*, o *Raspbian*.

Em todos os modelos disponíveis no mercado há a versão mais recente do *Raspbian* que pode ser usado no seu computador mesmo usando o sistema *Windows*, nesse caso, o sistema *Raspbian* será usado como uma máquina virtual dentro do *Windows* (Figura 3).

Figura 3 - Máquina Virtual do Raspbian dentro do Windows



(Fonte: Andr.oid Eric, 2015)

A *Raspberry Pi Foundation* foi criada no ano de 2008, dois anos depois dos primeiros protótipos terem sido implementados. O projeto *Raspberry Pi* foi criado sem fins lucrativos, portanto os desenvolvedores, na época estudantes, convenceram a *Broadcom* (fornecedora da unidade de processamento gráfico – *GPU*) a fornecer o sistema em um chip, *System on Chip (SoC)*, principal por um preço abaixo do mercado, cerca de US\$ 15, somando o valor da placa

aos demais componentes, que também são de baixo custo, teríamos o valor estimado de US\$ 35 para cada *Raspberry Pi* (CIRIACO, 200-?). A base dos primeiros modelos era o processador da Atmel ATmega644, que é a peça chave dos modelos *Raspberry Pi* como conhecemos hoje.

Três anos após a criação da fundação, já se dava início à produção em grandes quantidades do primeiro modelo criado, o *Raspberry Pi* Modelo A. Em fevereiro de 2012 a fundação recebeu os primeiros pedidos do modelo criado. Custando cerca de US\$ 25 (algo em torno de R\$ 80 na cotação atual), o *Raspberry Pi* Modelo A contava com um processador ARM single-core de 700 MHz e uma memória RAM de 256 MB (CIRIACO, 200-?).

Desde a comercialização do primeiro modelo produzido, já foram lançados um total de 3 (três) upgrades do *Raspberry Pi* Modelo A: O modelo A+, Modelo B, B+, Compute Module, a segunda e terceiras gerações do Modelo B. Este último conta com uma entrada *Micro Secure Digital Card High Capacity* (SDHC) para cartão de memória, um processador ARM Cortex-A7 quad-core 900 MHz e memória RAM de 1GB, além de mais 2 (duas) portas USB 2.0 em relação ao modelo A da primeira geração.

Como o objetivo inicial do projeto era proporcionar uma solução simples e barata para que os jovens possam dar os primeiros passos no mundo da programação (principalmente nos países em desenvolvimento), todo modelo do *Raspberry Pi* vem com a ferramenta MIT *Stretch* (Figura 4), que ensina o básico de programação, de forma rústica e até leviana.

Figura 4 - MIT Stretch



(Fonte: TheOpenSourcery.com, 2018?)

Como a *Raspberry Foundation* é uma instituição sem fins lucrativos, não é possível comprar ações da empresa, por esse motivo, a fundação depende de seus apoiadores que realizam doações para o desenvolvimento e crescimento da empresa. Entre os doadores, temos empresas como a Google, Microsoft, Oracle e Samsung. Com doações que vão desde a

disponibilização de recursos (como softwares e corpo técnico), até a valores financeiros que ultrapassam £500.000 (quinhentas mil libras), cerca de aproximadamente R\$ 2.393.875,00.

A fundação mantém outros projetos além do *Raspberry Pi*, para onde esses recursos também são destinados. Projetos como CODERDOJO, CODE CLUB e PICADEMY, são voltados para a educação e treinamento de professores e crianças.

Além desses projetos a fundação também mantém um fórum com mais de 200.000 participantes de todos os cantos do mundo, onde o compartilhamento de conhecimento e dúvidas sobre os projetos são difundidas (RASPBERRY PI FOUNDATION, 2018?).

Acerca da comercialização dos seus produtos, a fundação mantém “laços” com diversos países. No Brasil, o primeiro revendedor autorizado foi a FilipeFlop (THOMSEN, 2018?), o qual já comercializa todos os modelos do Projeto *Raspberry Pi* disponíveis no mercado. No site de vendas da FilipeFlop, também há um tutorial de downloads e os primeiros passos para aqueles que irão se aventurar no projeto *Raspberry Pi*. Os preços dos modelos produzidos também estão disponibilizados no site oficial da *Raspberry Pi Foundation* (figura 5), com os preços respectivos em dólar.

Figura 5 - Tabela de preços dos modelos produzidos.

Product	Price
Raspberry Pi Model A+	\$20
Raspberry Pi Model B+	\$25
Raspberry Pi 2 Model B	\$35
Raspberry Pi 3 Model B	\$35
Raspberry Pi Zero	\$5
Raspberry Pi Zero W	\$10

(Fonte: *Raspberry Pi Foundation*, 2018?)

Além dos preços praticados, a fundação também disponibiliza uma tabela de especificações técnicas (Figura 6), que permite ao usuário escolher qual modelo melhor às suas necessidades. Naturalmente que, com o upgrade dos modelos, a tendência é que os mais novos lançados se tornem cada vez mais compactos e com mais funcionalidades para o usuário.

Figura 6 - Tabela de especificações técnicas dos modelos lançados

Product	SoC	Speed	RAM	USB Ports	Ethernet	Wireless/Bluetooth
Raspberry Pi Model A+	BCM2835	700Mhz	512MB	1	No	No
Raspberry Pi Model B+	BCM2835	700Mhz	512MB	4	Yes	No
Raspberry Pi 2 Model B	BCM2836 or BCM2837	900Mhz	1GB	4	Yes	No
Raspberry Pi 3 Model B	BCM2837	1200Mhz	1GB	4	Yes	Yes
Raspberry Pi Zero	BCM2835	1000Mhz	512MB	1	No	No
Raspberry Pi Zero W	BCM2835	1000Mhz	512MB	1	No	Yes

(Fonte: Raspberry Pi Foundation, 2018?).

Em sua loja oficial, a fundação comercializa kits práticos de aprendizagem desde o modelo mais básico (Figura 7) até o último modelo lançado. Também são disponibilizados módulos compatíveis com os modelos *Raspberry Pi*, assinatura da revista e guias práticos para todos os níveis de usuários.

O livro oficial Raspberry Pi para iniciantes traz tudo que você precisa para iniciar hoje! Nele você encontrará o modelo Raspberry Pi Zero W, o estojo oficial (com três capas), cartão SD com o NOOBS pré-instalado, sem mencionar os adaptadores para cabo USB e HDMI. Acompanhando o livro de 116 páginas vai o guia para iniciantes, para ajudar você a dominar seu novo Raspberry Pi! (RASPBERRY PI PRESS, 2018?, tradução nossa).

Figura 7 - Kit de aprendizado para iniciantes, modelo Pi-Zero



(Fonte: Fundação Raspberry Pi, 2018)

1.3 BANCOS DE DADOS (BD'S)

Desde a década de 50, a necessidade do uso de banco de dados era comum nas grandes indústrias. Manter informações de funcionários, compras, vendas, maquinário e financeiro era mais que um controle, mas sim, uma necessidade.

Os primeiros bancos de dados eram realizados na forma física, através do papel. Fichas eram preenchidas e armazenadas em pastas que por sua vez eram arquivadas em grandes armários e ficavam lá por décadas. Com a evolução da tecnologia, os bancos de dados também tiveram seu *upgrade*. A automatização da informação trouxe inúmeras melhorias ao que se trata hoje como banco de dados.

Os milhares de dados que precisavam ser digitalizados com o advento do computador, não foram feitos de uma hora para outra de forma prática e eficiente. Os primeiros bancos de dados eram lidos de forma sequencial, pois na época, os registros ficavam guardados em cartões perfurados, fitas magnéticas e etc., portanto, se quiséssemos obter os dados provenientes do décimo registro, por exemplo, teríamos que realizar a leitura de todos os registros anteriores até encontrar o solicitado.

Posteriormente, os registros tiveram um índice colocado para facilitar o acesso. Dessa maneira o acesso poderia ser feito de forma rápida e sem a necessidade de ler toda a sequência de dados para encontrar o que se queria. Essa forma de acesso ficou conhecida como Acesso Direto.

Na década de 60, o departamento de defesa dos Estados Unidos promoveu um encontro chamado CODASYL, que reunia militares, grandes empresas e universidades, com o objetivo de criar um método de armazenamento de dados de forma segura e inteligente. Desse encontro, surgiu uma linguagem que ficou conhecida em todo mundo, o Cobol. Essa linguagem não se preocupava apenas com a programação em si, mas, também com a estrutura dos dados a serem tratados (GUANABARA, 2016).

Das inúmeras discussões realizadas nesse evento, houve o surgimento de uma nova tecnologia: o banco de dados.

No modelo criado, o banco de dados era composto e, é até hoje, de quatro partes: base de dados, sistema gerenciador (SGBD – Sistema Gerenciador de Banco de Dados), linguagem de exploração e programas adicionais (gerência de usuários, otimizadores).

Após o CODASYL, grandes empresas passaram a investir na tecnologia de banco de dados e a mais notória delas foi a IBM.

Graças a IBM surgiu um novo modelo de BD, onde os dados eram ligados de maneira simplista, porém de forma hierárquica (superior e inferior). Esse modelo ficou conhecido como Modelo Hierárquico e foi utilizado por muitos anos até o surgimento da sua evolução: o Modelo em Rede.

No modelo em redes, os dados já não mais eram ligados obedecendo uma hierarquia, mas sim, numa forma de rede inteligente.

Nos modelos desenvolvidos pela IBM, os dados eram armazenados conforme seus índices e escala hierárquica, mas não permitia interação entre eles, por esse motivo surgiu o Modelo Relacional (também desenvolvido pela IBM), que é utilizado até hoje.

Para o acesso a esses bancos de dados era necessária uma linguagem de exploração que pudesse ser aceita por todos os tipos de bancos de dados, foi então que surgiu o *Structured English Query Language* (SEQUEL). Após alguns anos a linguagem passou a se chamar *Structured Query Language* (SQL).

Inicialmente, a SQL foi criada para ser uma linguagem universal, onde seria possível acessar e/ou suportar diferentes tipos de bancos de dados utilizando o mesmo comando. Como não havia padronização, cada fabricante de bancos de dados passou a implementar e criar a sua própria SQL para acessar seus bancos. Com isso houve uma intervenção promovida pelas empresas americanas *International Organization for Standardization* (ISO) e *American National Standards Institute* (ANSI) para tentar padronizar a linguagem SQL.

Após a padronização, surgiram então diversos bancos de dados, como a Oracle, DB2 (IBM), dBASE e SQL Server (*Microsoft*). Esses bancos dados de dados são de caráter empresarial, ou seja, para usufruir dessas soluções deve se dispor de recursos financeiros.

No entanto, há bancos de dados disponibilizados de forma gratuita, como: MySQL (objeto de estudo desta proposta de projeto), MariaDB, FireBird, MongoDB e o PostgreSQL.

1.4 ARDUINO

Arduino é uma plataforma de prototipagem eletrônica *open-source* criada em 2005 utilizada para a construção de projetos eletrônicos utilizando como “cérebro” do projeto o microcontrolador AVR da ATMEL (Figura 8).

O objetivo era criar um dispositivo que fosse barato, bastante funcional e fácil de programar, proporcionando com isso o acesso à estudantes e projetistas iniciantes. Além da concepção de projeto ser código aberto, o hardware também o é. O conceito de *hardware* livre

O Arduino é uma plataforma de código aberto usada para construir projetos de eletrônica. O Arduino consiste de uma placa de circuito físico programável (geralmente chamada de microcontrolador) e um software, ou IDE (Integrated Development Environment) que roda em seu computador, usado para escrever e fazer upload de código de computador para a placa física.

A plataforma Arduino tornou-se bastante popular entre as pessoas que estão começando com a eletrônica e por um bom motivo. Ao contrário da maioria das placas de circuito programáveis anteriores, o Arduino não precisa de um hardware separado (chamado de programador) para carregar um novo código na placa - você pode simplesmente usar um cabo USB. Além disso, o Arduino IDE usa uma versão simplificada do C++, facilitando o aprendizado do programa. Finalmente, o Arduino fornece um fator de forma padrão que divide as funções do microcontrolador em um pacote mais acessível (SPARKFUN, 2014, tradução nossa).

Figura 8 - Arduino Uno



(Fonte – Sparkfun.com, 2014)

De origem italiana, o projeto Arduino, no ano de lançamento, fez tanto sucesso que obteve uma menção honrosa na concorrência pelo prêmio *Prix Ars Electronica* na categoria Comunidades Digitais em 2006 e chegou a vender mais de 50.000 placas no primeiro ano (SOARES, 2013).

O projeto Arduino foi criado para qualquer pessoa que tenha interesse em criar objetos e/ou ambientes interativos. Ele pode interagir com botões, displays, *buzzers*, leds, alto-falantes, GPS, motores e outros milhares de periféricos que atendam à necessidade primordial, a sua. Os custos para se ter uma placa dessas comparados à sua funcionalidade são mínimos. Para se ter uma idéia o Arduino modelo Nano pode ser encontrado no mercado pelo preço de \$12,00 (doze dólares) aproximadamente R\$ 45,00 (quarenta e cinco reais).

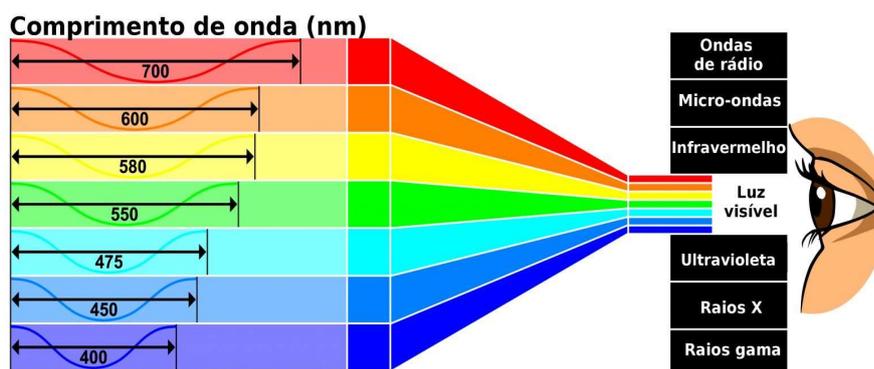
Nesta proposta de projeto, o Arduino será usado como uma interface entre o *Raspberry Pi* e os circuitos periféricos, funcionando apenas como um *drive* instalado sendo controlado e se comunicando através do protocolo I2C com o processador do *Raspberry*.

1.5 INFRAVERMELHO

“O infravermelho é uma onda eletromagnética, que possui uma frequência que não faz parte do espectro de visão humano” (JÚNIOR, 201-?).

O comprimento de onda do infravermelho fica entre $1\mu\text{m}$ ($1 \times 10^{-6}\text{m}$) e 1mm ($1 \times 10^{-3}\text{m}$) que é menor que a cor vermelha visível ao olho humano (Figura 9), daí a origem do nome.

Figura 9 - Espectro visível



(Fonte: JÚNIOR, 201-?)

“A radiação infravermelha é liberada de todos os corpos, devido à vibração das moléculas que causa oscilações nas cargas elétricas constituintes dos átomos, por esse motivo essa radiação é associada ao calor” (JÚNIOR, 201-?).

As aplicações do infravermelho vão desde a medicina até o ramo militar. Mísseis guiados por calor e o diagnóstico de doenças como a Aterosclerose (obstrução dos vasos sanguíneos), são alguns dos exemplos dessa aplicação.

Na indústria, fotos termográficas são muito comuns para o diagnóstico de sobrecarga em equipamentos e pontos quentes nos processos de destilação e craqueamento catalítico na indústria petrolífera. Após inúmeras pesquisas, o infravermelho foi ganhando ainda mais utilidades. O uso mais famoso do infravermelho é nos celulares. Informações poderiam ser transmitidas de um aparelho para outro através das ondas.

Na década de 70 o uso de controles remotos com infravermelho tomou o mercado. No controle há um diodo que emite as ondas eletromagnéticas infravermelhas e no receptor de TV há um diodo que apenas recebe essas informações e as envia para serem decodificadas pelo microprocessador contido no televisor. Esses dados, tais como: liga, desliga, aumenta volume e diminui volume apenas são uma sequência de zeros e uns transmitidos pela luz *infrared* (LAYTON, 2018?).

A radiação infravermelha são ondas de comprimento de 1 milímetro até 700 nanômetros, e, portanto, não visíveis para o olho humano. É uma radiação não ionizante, por isso, sem efeitos danosos, (sem riscos de causar males como, por exemplo, câncer). No espectro de luz, está localizado depois da luz vermelha, daí surgiu seu nome. Apesar de não poder ser vista, a radiação infravermelha pode ser notada no corpo em forma de calor: terminações nervosas, chamadas termorreceptores, conseguem captar essa radiação (CARVALHO, et.al., 2015).

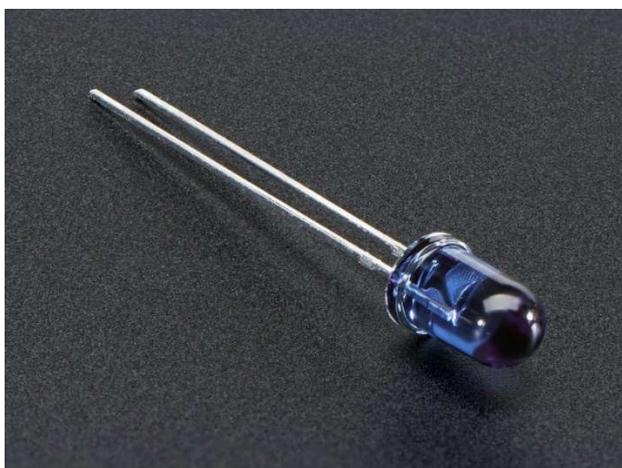
1.5.1 Diodo Emissor de Luz (LED) infravermelha

O diodo emissor de luz infravermelha é um dispositivo eletrônico que funciona como um diodo emissor de luz comum, mas que usa outros materiais para poder produzir a luz infravermelha (Figura 10).

Um LED infravermelho é, como todos os LED's, um tipo de diodo ou um semicondutor simples. Os diodos são projetados de modo que a corrente elétrica só pode fluir em uma direção. Como os fluxos de corrente, os elétrons saem de uma parte do diodo para outra parte. Para isso, os elétrons devem lançar energia na forma de fótons, que produzem luz. Embora invisíveis aos olhos humanos, muitos tipos de câmeras e outros sensores podem detectar a luz infravermelha. Isso faz com que a tecnologia de infravermelho LED se torne bem adequada para aplicações como sistemas de segurança e óculos de visão noturna. Muitas câmeras de segurança e filmadoras usam LED's infravermelhos para fornecer um modo de visão noturna. Os robôs também podem usar um LED infravermelho para detectar objetos a alguns metros de distância. Podem até mesmo ter um diodo emissor de luz infravermelha para transmitir dados para uma ferramenta de fácil leitura do medidor (BRANCO, 2017).

As aplicações do LED infravermelho se estendem até a área da medicina. Estudos recentes informam que o LED infravermelho já foi utilizado para o tratamento de doenças reumáticas crônicas como a Síndrome da Fibromialgia (SFM). Pacientes que experimentaram o tratamento com a utilização dos LED's, obtiveram melhoras significativas com a redução dos sintomas da doença (CARVALHO, et.al, 2009).

Figura 10 - LED Infravermelho super-brilho



(Fonte: Adafruit.com, 2018)

O objetivo desses diodos é transmitir a informação (trem de pulsos digitais) ao receptor infravermelho (normalmente fototransistor ou fotodiodo) que recebe esses dados e os envia para serem tratados. Cada trem de pulsos emitido é entendido pelo receptor como um código específico atrelado à um comando pré-determinado pelo programador no seu microcontrolador.

1.5.2 Módulo Emissor de Luz Infravermelha

O módulo emissor de luz infravermelha é uma aplicação do *LED Infrared* em circuitos eletrônicos que utilizam dispositivos remotos. Ele é ideal para o acionamento de motores, alarmes, relés e outros usos em automação residencial. Por se tratar de um módulo, ele se diferencia em alguns aspectos do *LED infrared* comum. Além do *LED infrared* ele possui um resistor em série para limitar a corrente (Figura 11) que o ultrapassa e ambos são instalados numa placa de circuito impresso. Na placa há também dois conectores para facilitar sua utilização numa *protoboard* (Matriz de Contato) ou diretamente nas saídas GPIO's de um microcontrolador. Normalmente utilizado em circuitos que usam microcontroladores, os módulos emissores de luz infravermelha são conectados a eles, de onde serão enviados os dados necessários para o acionamento dos dispositivos remotos através de IR.

Este Módulo LED Emissor Infravermelho é comumente utilizado em projetos com Arduino, PIC, Raspberry, NodeMCU ESP8266 e outras plataformas microcontroladas. Geralmente o LED IR é usado em projetos que envolvam clonar as teclas de um controle remoto infravermelho, para em seguida os códigos dessas teclas serem enviados de volta ao aparelho (TV, aparelho de som, DVD, BluRay), a partir do uso do LED emissor infravermelho em conjunto com um microcontrolador. Além disso, o módulo pode ser aplicado em projetos de simulador de segurança, onde um fototransistor fica apontado diretamente para o LED emissor IR do módulo e quando esse sinal é interrompido uma ação pode ser executada a partir do microcontrolador. (MASTERWALKERSHOP, 2018?).

Figura 11 - Módulo emissor de luz infravermelha



(Fonte: Multipecastec.com.br, 2018?)

1.5.3 Módulo Receptor de Luz Infravermelha.

A recepção infravermelha é utilizada para receber o código transmitido pelo emissor de luz infravermelha e enviá-lo ao módulo de processamento para ser decodificado e posteriormente executar a ação solicitada pelo usuário. O componente responsável para tal recepção é o módulo receptor de luz infravermelha que é composto por um fototransistor ou fotodiodo o qual converte a luz recebida em sinal elétrico para posteriormente ser tratado (Figura 12).

Figura 12 - Módulo Receptor Infravermelho.



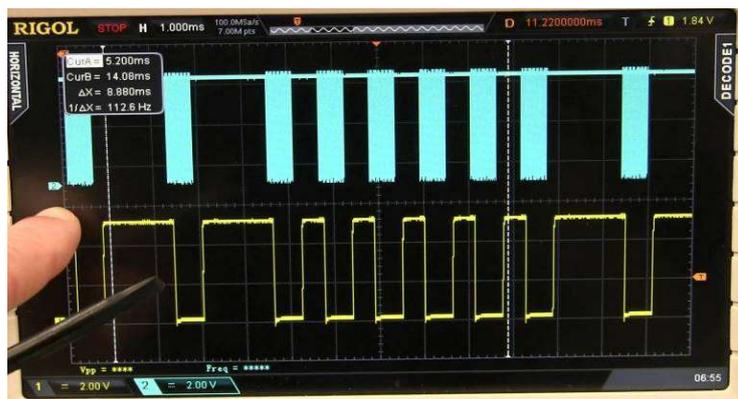
(Fonte: vidadesilicio.com.br, 2018)

O que é transmitido pelo controle remoto de um TV, por exemplo, nada mais é que um trem de pulsos que identifica cada “botão” acionado no controle. O código de cada botão já está previamente gravado no receptor, assim, não há a possibilidade de existir erro entre informações distintas, uma vez o que o processador do receptor decodificará exatamente aquela ação selecionada.

Em alguns casos, para melhorar a segurança do que se é transmitido pelo controle, usa-se circuitos moduladores de sinais, para impedir que outras frequências na faixa do infravermelho que estejam no ambiente interfiram no código do sinal transmitido (Figura 13).

O receptor infravermelho (IR) disposto num dado aparelho recebe os pulsos de luz, transformando-os em sinais elétricos que são tratados como dados binários. Este sinal é recebido e processado pelo sistema receptor. Portanto, neste processo de comunicação, é necessário existir uma espécie de “tradutor” que deve receber estes pulsos armazená-los em uma frase (que contem de 32 bits) associada um dado comando (CAVALCANTE, et. al., 2013).

Figura 13 - Sinal modulado e sinal modulante.



(Fonte: vidadesilicio.com.br, 2018)

1.6 MÓDULO *RADIO FREQUENCY IDENTIFICATION* (RFID)

Identificação por radiofrequência ou RFID é um método utilizado para detecção e reconhecimento automático através de ondas de rádio, gravando e/ou recuperando os dados em dispositivos denominados etiquetas RFID (SILVEIRA, et. al., 201-?).

A tecnologia RFID veio em substituição aos códigos de barras convencionais (INSOFT4, 201-?). Ela utiliza uma comunicação de curto alcance onde esses dados podem ser lidos por módulos sensores automaticamente, apenas pela passagem do objeto portador da etiqueta RFID (Figura 14).

A identificação por rádio frequência (RFID) surgiu em 1960 com a finalidade de resolver os problemas de acompanhamento e acesso em sistemas que não permitiam o contato direto e em ambientes hostis onde o código de barra não podia ser aplicado. A idéia básica da tecnologia de identificação por rádio frequência consiste em se utilizar um microchip ligado a uma antena, operando tanto em baixas como altas frequências. Esse microchip consiste num transponder que não necessita de fonte de alimentação, pois o sinal que o excita vem diretamente de um circuito de leitura/gravação que o excita (BRAGA, 201?).

Figura 14 - Módulo RFID, cartão e etiqueta

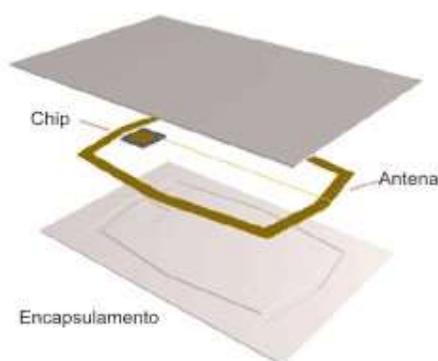


(Fonte: Eletrogate.com, 2018?)

O cartão RFID (Figura 15) é composto basicamente de três camadas: o chip, a antena e o invólucro (encapsulamento). O cartão não possui bateria, por esse motivo, a bobina que há no seu interno (antena) precisa ser excitada pelo módulo de leitura/gravação. O módulo de leitura/gravação irá emitir uma onda de frequência constante e baixa potência, daí a necessidade da aproximação do cartão junto ao módulo. Essa distância pode variar de 5cm a 90cm dependendo da potência do módulo leitor/gravador. Através do campo magnético gerado na bobina do cartão, será então induzida uma corrente que alimentará o circuito do cartão RFID. Dessa forma, o chip poderá então transmitir de volta o número gravado para o leitor/gravador, de forma a identificá-lo.

“Na prática isto ocorre durante o processo de aproximar o cartão do leitor, tudo muito rápido e imperceptível ao usuário. Este número é único e já vem gravado de fábrica, evitando sua duplicação para tentativa de fraudar o processo de acesso” (MSECCO, 2009).

Figura 15 - Cartão RFID



(Fonte: Dias, 2015).

1.7 MÓDULO LEITOR DE IMPRESSÃO DIGITAL (BIOMÉTRICO)

“A biometria consiste na tecnologia capaz de reconhecer as pessoas através de suas características físicas, como a face, a íris dos olhos, pelos vasos sanguíneos das mãos ou pelo método mais comum que é a impressão digital” (VANI, 2012).

As aplicações mais comuns da biometria são o controle de acesso, identificação criminal e cadastro em bancos de dados.

O termo biometria significa medição biológica, ou seja, é o estudo das características físicas e comportamentais de cada pessoa. O princípio básico desta técnica para identificação é: seu corpo, sua senha. Embora tenham se tornado famosas há pouco tempo, as técnicas de reconhecimento por meio das características das pessoas já eram utilizadas na China no ano 800 d.C., quando comerciantes confirmavam a identidade de seus clientes por meio da impressão de suas digitais em tábuas de barro. Já deu para notar que, mesmo a ideia já sendo usada há muito tempo,

até recentemente os scanners biométricos eram apenas coisa de televisão. A maior limitação para o avanço na área se deve aos aparelhos utilizados, afinal não é assim tão simples criar um scanner de retina ou palmar que extraia apenas as informações necessárias. (MARTINS, 2009).

Os sistemas biométricos para realizarem a identificação, necessitam de dados característicos físicos de uma pessoa, como os olhos e/ou as digitais das mãos por exemplo.

A captura, a extração e a comparação são os principais componentes de um sistema biométrico. Na captura é possível realizar a aquisição de uma amostra biométrica para realizar a identificação de um indivíduo. Isso pode ser feito por meio de digitais ou por meio da leitura da íris do olho, por exemplo. Já na extração é realizada a remoção de uma amostra de informação biológica única do indivíduo, para que, com base nela, possa ser feita a identificação. O resultado dessa análise é chamado de template. Por último, a comparação utiliza o template armazenado para compará-lo com outra pessoa. (CANALTECH, 2018?).

Para esta proposta de projeto será utilizado o leitor biométrico ou leitor de impressão digitais (Figura16) que irá comparar as imagens ‘*scaneadas*’ com o banco de dados pré-existente.

Neste banco de dados poderão ser cadastradas as digitais de todos os funcionários da escola ou universidade, que terão acesso às salas de aulas. Pelo sistema implantado será possível permitir, registrar e/ou bloquear o acesso às salas possibilitando assim mais praticidade à instituição e aos usuários.

Figura 16 - Modulo leitor biométrico USB



(Fonte: teknew.com.br)

O módulo leitor biométrico retira uma fotografia da impressão digital dos dedos das mãos, para cadastro e/ou comparação com banco de dados existentes. Há muitos modelos disponíveis no mercado, com utilizações para os diversos tipos de sistemas.

A facilidade do leitor biométrico tornou alguns serviços mais práticos e eficientes. Segurança eletrônica e reconhecimento para a justiça eleitoral, são alguns dos exemplos de

serviços facilitados pelo leitor. Grandes empresas também utilizam esse tipo de dispositivo para o acesso e controle de ponto eletrônico ou para a utilização de convênios médicos.

Para a aplicação deste projeto, o módulo leitor de impressões digitais (biométrico) será o modelo para Arduino (Figura 17), utilizaremos apenas como alternativa para o módulo de entrada RFID uma vez que a implementação do módulo juntamente ao Raspberry é muito mais complexa e depende de bibliotecas pouco difundidas no meio. Dessa forma, a montagem do leitor biométrico para o circuito final deste projeto, não será realizada.

Figura 17 - Leitor biométrico para Arduino.



(Fonte: smartsprojectsbrasil.com.br, 2018?)

1.8 FIREBASE

O *Firebase* é uma plataforma de criação de aplicativos (Figura 18) desenvolvida pela empresa de mesmo nome fundada em 2011 por Andrew Lee e James Tamplin, que foi adquirida pela Google em 2014 (CLEVEROAD.COM, 2016).

O *Firebase* é um *Backend as a Service* (BaaS), que nada mais é que uma plataforma de soluções hospedadas em nuvem que possui o propósito *Application Programming Interface* (API) e *Software Development Kit* (SDK) para que essas aplicações sejam utilizadas por outras aplicações (AMORIM, 2017).

Batschinski (2016) diz, *Backend-as-a-service* (BaaS) é um serviço de computação em nuvem que serve como *middleware*. O mesmo fornece aos desenvolvedores uma forma para conectar suas aplicações mobile e web a serviços na nuvem a partir de APIs e SDKs.

API (Application Programming Interface) é simplesmente uma interface que permite que software interaja com outro software. Esse é o core de sua funcionalidade. Pense na API como a Pedra Roseta, uma tabuleta na qual duas línguas completamente diferentes com dois conjuntos de instruções diferentes podem ser traduzidas e transferidas para entendimento mútuo[...].

SDK significa Software Development Kit, o que é uma ótima forma de enxergá-lo: como um kit. Imagine montar um carro ou um avião.

Um SDK ou devkit funciona de forma muito similar, fornecendo um conjunto de ferramentas, exemplos de código, documentação relevante e processos ou guias para ajudar desenvolvedores criarem aplicações de software em uma plataforma específica. Se uma API é um conjunto de blocos de construção que permitem a criação de algo, um SDK é um workshop completo, facilitando a criação longe do escopo que uma API permitiria por conta própria.

SDKs são a fonte originária de praticamente todo programa que um usuário moderno iria interagir com. Desde o browser que você usa no trabalho até os games que joga no final do dia, muito foi construído com um SDK mesmo antes de uma API ser usada para comunicar com outras aplicações (NORDIC APIS, 2016).

A plataforma *Firebase* fornece ao desenvolvedor todas as ferramentas necessárias para a praticidade e qualidade dos aplicativos criados, com um banco de dados em tempo real e suporte para os diversos sistemas operacionais, como Android e iOS, por exemplo.

Ferramentas como Autenticação, *Storage*, Monitoramento de Desempenho, Previsões, Resultados Analíticos, Indexação de *Apps* e outros, permite ao usuário do *Firebase* criar, aprimorar ou melhorar a qualidade dos seus aplicativos ou ainda expandir seus negócios (FIREBASE, 2018?).

Figura 18 - Aprimorando Apps



(Fonte: [Firebase.google.com](https://firebase.google.com))

2 METODOLOGIA DE DESENVOLVIMENTO

O trabalho apresentado será uma pesquisa aplicada, e terá como objetivo a realização de pesquisa exploratória e experimental, descritiva e explicativa sobre o material bibliográfico adquirido sobre o assunto. Será utilizado o procedimento técnico de pesquisa bibliográfica. O método de abordagem será o hipotético-dedutivo e o método de procedimento de elaboração será o monográfico. A coleta de dados será realizada através da documentação direta e indireta, sendo estes dados qualitativos e interpretados de forma global.

Foram realizadas pesquisas bibliográficas na área de sistemas microprocessados e microcontrolados, focados no microcomputador *Raspberry Pi*, plataforma de desenvolvimento de aplicativos *mobile* e *web*, enfoque na plataforma *Firebase*, automação residencial e eficiência energética.

Inicialmente será feita a aquisição dos módulos leitor biométrico, infravermelho e do microcomputador *Raspberry Pi* Modelo 3.

O projeto será desenvolvido baseado nas seguintes etapas:

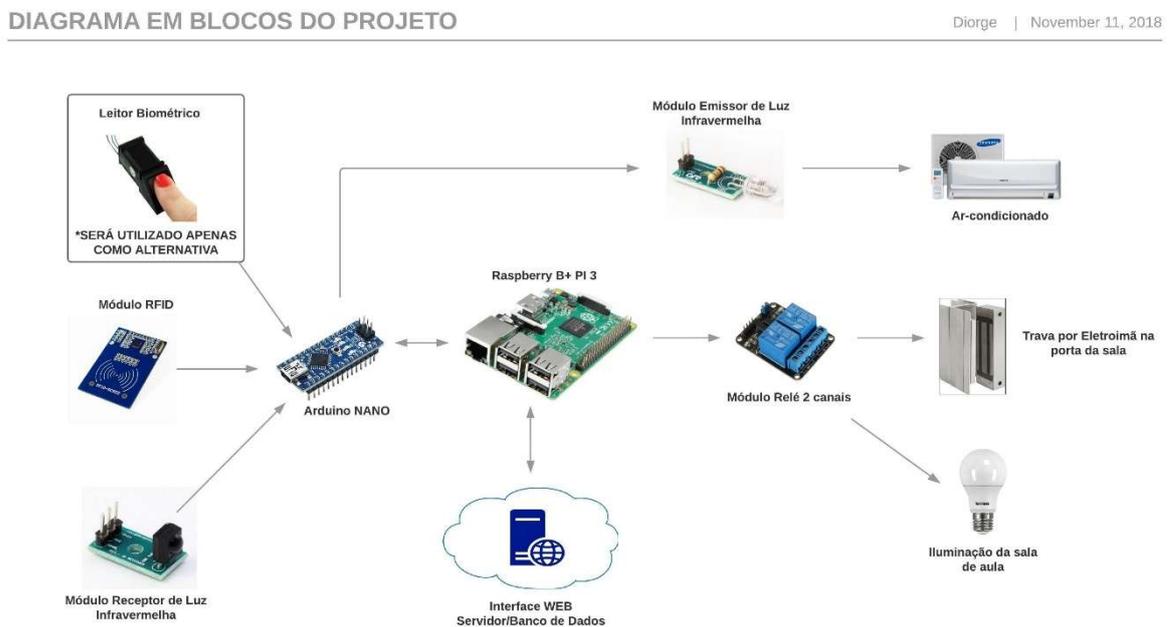
- a) Instalação do *Virtual Network Computing* (VNC), para acesso ao *Raspberry Pi* por meio do endereço de IP;
- b) Instalação dos módulos Infravermelho e RFID para utilização com *Raspberry Pi*;
- c) Criação de uma interface *Web* para acesso e cadastro ao banco de dados do *Firebase*;
- d) Instalação de um módulo relé para acionamento do sistema de iluminação através do *Raspberry Pi*;
- e) Instalação de uma trava eletromagnética na porta da sala de aula que será liberada pelo microcomputador *Raspberry Pi*;
- f) Gravação das TAG's (etiquetas RFID) para cada funcionário e professor com acesso permitido;
- g) Criação de uma rede de dados por onde irá ser realizada a comunicação entre o computador gerenciador/servidor que será instalado em uma sala remota (também *Raspberry Pi*) e o microcomputador que ficará na porta da sala de aula;
- h) Criação de um programa em PHP, dentro do *Raspberry Pi*, que realize todas as ações necessárias à execução das propostas do projeto;
- i) Por fim será realizada a montagem e instalação do circuito na sala de aula escolhida para demonstração do projeto.

2.1 LAYOUT DO PROJETO

Utilizando um software *on-line* para construção de diagramas, foi montado o *layout* mostrado na figura 19.

Neste *layout* pode ser visto, de forma intuitiva, como os dados e os comandos fluem através dos dispositivos de controle, neste caso temos o Arduino Nano e o *Raspberry Pi*, lembrando que o Arduino serve apenas como interface entre os periféricos e o microcomputador *Raspberry Pi*. A comunicação entre o *Raspberry Pi* e o Arduino será realizada de forma serial.

Figura 19 - *layout* básico do projeto.



(Fonte: Elaborado pelo autor – lucidchart.com)

Conforme previsto na metodologia de desenvolvimento do projeto, as etapas para elaboração e construção do código-fonte, circuito protótipo e banco de dados hospedados em nuvem, seguiram os seguintes passos:

- a) Aquisição de materiais;
- b) Construção do código-fonte;
- c) Construção do circuito protótipo;
- d) Desenvolvimento do aplicativo *Web* na plataforma *Firebase*;
- e) Desenvolvimento do banco de dados na plataforma *Firebase*;
- f) Desenvolvimento da interface entre o aplicativo *Web* e a plataforma *Firebase*.

2.2 CONTRUÇÃO DO CÓDIGO-FONTE

Para início do desenvolvimento do código-fonte, foi montado um fluxograma de como irá funcionar o projeto, este fluxograma se encontra no apêndice A, em seguida foram montados todos os periféricos de forma individual para testes.

Na plataforma do Arduino utilizamos apenas os códigos exemplos que o software já disponibiliza, para podermos realizar o teste de cada periférico. Nos anexos A, B e C, são mostrados os códigos exemplos que foram utilizados para teste dos módulos RFID, receptor IR e emissor IR.

2.2.1 Desenvolvimento da lógica do código-fonte

Para funcionamento do circuito, a lógica prevista no fluxograma de funcionamento encontrada no apêndice A desta proposta de projeto deve ser seguido.

Primeiramente deverão ser definidas as entradas de dados para o sistema. O acesso será realizado por meio de um cartão RFID previamente cadastrado para cada funcionário da Escola Superior de Tecnologia. Esses dados serão registrados via aplicativo iOS para o banco de dados modelado e construído na plataforma *Firestore*. Os dados incluem: nome do funcionário, matrícula e cargo.

O circuito instalado na porta da sala de aula servirá apenas de controle do acesso e dos sistemas contidos na sala, não sendo possível realizar cadastro de novos cartões. Essa função será realizada por outro circuito previamente instalado na sala administrativa da segurança da UEA. Esse processo impedirá que novos usuários obtenham acesso ou disponibilizem acesso à terceiros sem autorização prévia da instituição.

Conforme disponibilizado no *layout* da figura 19, há uma alternativa ao uso do cartão RFID, que seria o leitor de impressões digitais (biométrico). Essa alternativa prevê os mesmos campos cadastrados para o cartão RFID, somente com o adendo de uma fotografia da digital do funcionário para cadastro. O uso do leitor biométrico traria mais comodidade e segurança ao usuário do sistema, uma vez que na ausência do cartão RFID a utilização da digital permitiria o acesso à sala de aula. Conforme informado no tópico 1.7 deste trabalho, o módulo leitor de impressões digitais (biométrico), não será implementado no circuito final deste projeto.

Toda a comunicação dos periféricos de entrada será realizada com o Arduino Nano conforme código-fonte previsto no apêndice B deste trabalho. Após os dados terem sido coletados pelo Arduino Nano, eles serão transmitidos de forma serial para o Raspberry Pi, que

estará *on-line* com o banco de dados na base virtual do *Firebase* para liberação do acesso ou cadastro de novo usuário. Inicialmente, a comunicação entre o *Raspberry* e o *Arduino* seria realizada por meio do protocolo I2C, porém, a velocidade de transmissão dos dados estaria comprometida. Como exemplo, realizamos testes que demoraram cerca de três minutos para a leitura e transmissão dos dados de um único cartão RFID. Dessa forma, a comunicação serial foi obviamente a escolha trivial. Lembrando que o aplicativo construído na plataforma iOS para controle dos dados terá dupla função: 1) Cadastrar os dados de um novo cartão para determinado funcionário; 2) Controlar via nuvem virtual os sistemas da sala de aula de forma manual. Essa última função foi implementada devido à impossibilidade da montagem do leitor biométrico.

No microcomputador *Raspberry Pi*, há um código escrito na linguagem *Python* que possibilita a comunicação com a base de dados na nuvem virtual da plataforma *Firebase*, esse código se encontra no apêndice C deste trabalho.

Após a confirmação dos dados, pelo microcomputador *Raspberry Pi*, o acesso será liberado com um comando disparado em duas saídas GPIO do próprio *Raspberry Pi*, que será enviado ao módulo Relé de 2 canais, para acionar tanto a trava por eletroímã contida na porta da sala de aula, liberando a entrada, assim como o acionamento das luzes da sala. O acionamento do ar-condicionado será realizado também pelo *Raspberry Pi*, porém, por meio do *Arduino Nano*. Conforme descrito no código-fonte do *Raspberry Pi*, após a confirmação dos dados de usuário, será enviado um código para o *Arduino Nano*, via barramento serial para o acionamento do módulo emissor de luz infravermelha, concluindo assim, o acionamento do ar-condicionado.

Liberando o acesso ao usuário, o microcomputador *Raspberry Pi* irá disparar 2 cronômetros em contagem regressiva: o primeiro contabilizará o tempo de 1h e 20min; o segundo contabilizará o tempo de 1h e 40min.

A função de cada timer é simples: o primeiro informará ao *Raspberry Pi* a proximidade do horário do final da aula, o que fará com o que o *Raspberry Pi* envie ao *Arduino Nano* um comando que posteriormente será repassado ao módulo emissor de luz infravermelha, para que seja realizado o desligamento do ar-condicionado da sala de aula. Essa ação informará ao usuário, ainda que bruscamente, que o horário de término da aula se aproxima; o segundo timer tem a função parecida com o primeiro, somente com a diferença que ao invés de desligar o ar-condicionado ele informará ao *Raspberry Pi* para desligar as luzes da sala. Lembrando que o usuário poderá, de forma manual, religar todos os sistemas da sala de aula por meio do aplicativo para *smartphone* na plataforma iOS.

A utilização do módulo receptor de luz infravermelha, se dá somente durante a instalação de todo o sistema de controle do acesso em uma sala de aula. Devido aos vários modelos de condicionadores de ar existentes na Escola Superior de Tecnologia, os códigos de cada controle remoto devem ser previamente cadastrados, para que a emissão dos sinais pelo emissor de IR seja a correta. Após realizada a leitura do código binário proveniente de cada controle remoto e esse valor for registrado no código-fonte, não se faz mais necessário o uso do módulo receptor de luz infravermelha.

No entanto, durante a construção do circuito protótipo, descobrimos que o código enviado pelos controles remotos dos condicionadores de ar mais modernos, não contém somente um código binário simples com a função desligar, ou a função ligar, mas sim, todo o modo de operação do ar-condicionado, dificultando assim a leitura pelo Arduino Nano. Mas, como foi visto em tópicos anteriores esse problema foi corrigido, aumentando o tamanho da variável de armazenamento desse código binário no código-fonte do projeto.

2.2.2 Construção do Aplicativo iOS

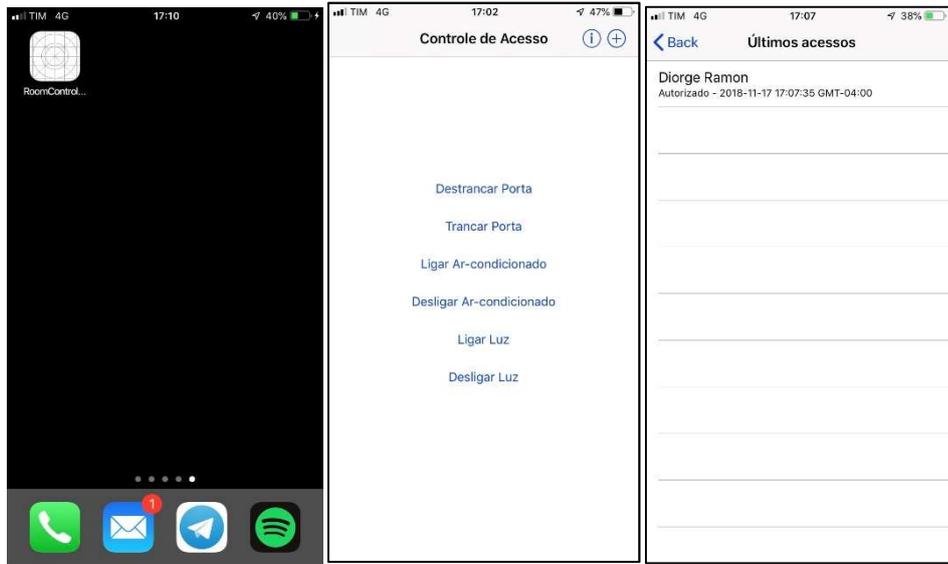
Para controle e cadastro dos usuários no banco de dados, foi implementado um aplicativo na plataforma iOS conforme figura 20. Esse aplicativo é simples, porém de muita funcionalidade e objetividade.

Foi dado o nome de *RoomController* e suas funções são as seguintes:

- 1) Destancar Porta;
- 2) Ligar Ar-Condicionado;
- 3) Desligar Ar-Condicionado;
- 4) Histórico de acesso;
- 5) Adicionar novo cartão.

Todas as funções acima definidas, têm por objetivo o controle de forma manual dos sistemas da sala de aula e o cadastro dos usuários, além de proporcionar a cada usuário a praticidade do acesso via internet no caso da ausência do cartão RFID.

O aplicativo receberá e enviará ordens ao banco de dados contido na plataforma *Firebase*, para controlar o circuito montado na sala de aula, uma vez realizado o acesso, será feito um registro que posteriormente poderá ser consultado, através do próprio aplicativo para referências e pesquisas futuras.

Figura 20 - Layout do *RoomController*.

(Fonte: Elaborado pelo autor.)

3 CONSTRUÇÃO DO CIRCUITO PROTÓTIPO

Este tópico apresenta a construção do circuito do projeto em questão. Nele serão abordados os materiais utilizados, montagem e teste dos periféricos juntamente com o microcontrolador Arduino Nano e o microcomputador *Raspberry Pi*.

3.1 MATERIAIS UTILIZADOS

Para a montagem do circuito foram utilizados os seguintes materiais:

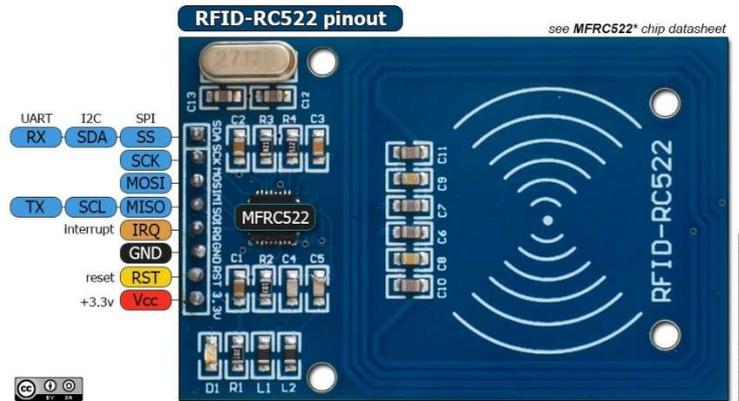
- 1 microcomputador *Raspberry Pi B* – modelo 3;
- 1 microcontrolador Arduino Nano;
- 1 módulo receptor RFID;
- 1 módulo receptor de luz infravermelha;
- 1 módulo emissor de luz infravermelha;
- 1 módulo relé eletromecânico de 2 canas;
- 1 trava por eletroímã 12V;
- 1 matriz de contatos (*Protoboard*) de 2200 furos;
- 1 *breakout board* para GPIO do Raspberry;
- 1 cabo micro-usb para Arduino;
- 1 LED Red Green;
- 1 fonte de alimentação de 2A para o *Raspberry Pi*;
- Jumpers para conexão M-M, M-F, F-F.

3.2 MONTAGEM DOS PERIFÉRICOS

3.2.1 Montagem do módulo RFID

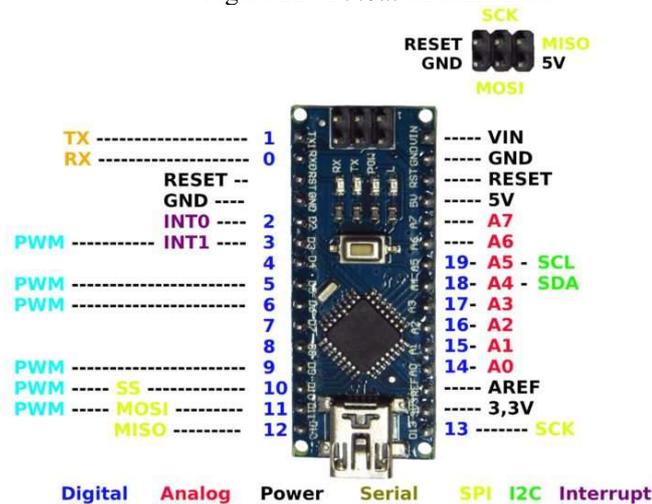
O módulo RFID foi montado conforme pinagem específica indicada na figura 21. No microcontrolador Arduino, as conexões também foram estabelecidas conforme pinagem pré-estabelecida (Figura 22). As conexões entre o módulo RFID e o Arduino estão especificadas conforme tabela 1.

Figura 21 - Pinagem do módulo RFID



(Fonte: Koyanagi, 2018)

Figura 22 - Pinout Arduino nano.



(Fonte: Filipeflop.com, 201-?)

Tabela 1 – Conexão entre o módulo RC522 e o Arduino.

RC-522 (RFID)	ARDUINO NANO
SDA	D10
SCK	D13
MOSI	D11
MISO	D12
IRQ	NÃO CONECTADO
GND	GND
RST	D9
VCC	+3.3V

(Fonte: Elaborada pelo autor)

3.2.2 Montagem dos módulos emissor e receptor de luz infravermelha

Conforme descrito no fluxograma, a importância do módulo receptor de luz infravermelha para o projeto, nada mais é que captar inicialmente o código hexadecimal proveniente de um controle remoto de um ar-condicionado qualquer e gravar esse código para posterior utilização no funcionamento do projeto.

Face a isto, deparamo-nos com um problema muito comum para os equipamentos mais modernos. Os controles de ar-condicionado não apenas armazenam mais um único código binário para cada botão pressionado, mas sim, toda a informação de funcionamento do mesmo, tais como modo (resfriar, ventilar, retira umidade), temperatura atual, velocidade do ventilador, mas sim um código para todo o modo de operação.

Como visto no tópico 2.2.1, este problema foi corrigido na construção do código-fonte aumentando o tamanho da variável que recebe o código do botão pressionado.

A tabela 2 mostra como foi realizada a montagem do módulo receptor de luz infravermelha.

Tabela 2 – Conexão entre o módulo receptor de luz infravermelha e o Arduino

RECEPTOR IR GBK	ARDUINO NANO
VCC	+5V
GND	GND
SINAL	D2

(Fonte: Elaborada pelo autor)

Na emissão do sinal captado de volta para o ar-condicionado, utilizamos o módulo emissor de luz infravermelha na porta digital D3 previamente *setada* como saída no Arduino nano. O outro pino do módulo é o GND ligado diretamente à conexão terra do Arduino, ambas conexões são listadas na tabela 3.

Tabela 3 – Conexão entre o módulo emissor de luz infravermelha e o Arduino.

EMISSOR IR GBK	ARDUINO NANO
SINAL	D3
GND	GND

(Fonte: Elaborada pelo autor)

3.2.3 Montagem do módulo relé 2 canais.

Para controle da iluminação e da trava por eletroímã da porta, foi montado um módulo relé de 2 canais 5V (Figura 23), sendo um canal para controle da iluminação e o outro canal para controle da trava por eletroímã. O sinal será emitido pelo próprio microcomputador Raspberry Pi, por meio do barramento GPIO. As conexões para estas conexões estão descritas na tabela 4.

Tabela 4 - Conexão entre o módulo relé de 2 canais, a iluminação e a trava por eletroímã.

MÓDULO RELÉ 2 CANAIS			
CANAL 1		CANAL 2	
NC	(não usado)	NC	Terminal GND do imã ³
COMUM	Lâmpada ¹	COMUM	Fonte 12 V
NO	Neutro ²	NO	(não usado)

(Fonte: Elaborada pelo autor)

Figura 23 - Pinout módulo relé 2 canais.



(Fonte: Mirella, 2015)

Tabela 5 - Conexões entre o módulo relé e o Raspberry Pi.

MÓDULO RELÉ 2 CANAIS	RASPBERRY PI
JD-VCC (interligado ao pino VCC)	-
VCC (interligado ao pino JD-VCC)	-
GND (não usado)	-
GND	GND
IN1	A6
IN2	A7
VCC	+5V

(Fonte: Elaborada pelo autor)

4 LEVANTAMENTO DE CUSTOS

Para este projeto, os gastos realizados na sua construção serão listados na tabela 6. Também será mostrado o comparativo com um projeto similar existente no mercado, mas que não possui todas as funções propostas e implementadas.

Tabela 6 - Custo dos itens do projeto.

ITENS	VALOR UNITÁRIO (R\$)
Raspberry Pi	129,90
Módulo RFID com TAG's	5,21
Módulo relé 2 canais	2,17
Módulo Emissor IR	3,41
Módulo Receptor IR	3,71
Arduino Nano	45,00
Trava por eletroímã	79,89
TOTAL GASTO	269,29

(Fonte: Elaborada pelo autor)

Para realizar um comparativo, pesquisamos a marca mais utilizada no mercado de travas eletromagnéticas interligadas a um sistema de acesso (fechaduras inteligentes), a marca comparada é a Intelbras. Observe na tabela 7, os valores encontrados.

Tabela 7 - Comparativo entre o projeto e as marcas Intelbras.

Modelo	Valor Unitário (R\$)	Conexão Internet	RFID	Qtd. Senhas cadastradas	Qtd. cartões RFID
Projeto TCC	269,290	Sim	Sim	Nenhuma	Ilimitado
FR 101	398,91	Não	Não	4	0
FR 200	495,00	Não	Sim	4	30
FR 220	1.169,21	Não	Não	4	0
FR 400	849,99	Não	Sim	4	100

(Fonte: Elaborada pelo autor)

5 RESULTADOS

De forma objetiva, a construção deste projeto alcançou os resultados esperados e propostos pelo tema inicial escolhido.

O uso dos módulos emissor e receptor de luz infravermelha não foi possível como indicado no layout inicial do projeto. O primeiro motivo foi a falta de uma biblioteca adequada e difundida desses módulos para o seu uso com as saídas GPIO do *Raspberry Pi*, isso causou uma dificuldade no momento da criação do código em Python. Dessa forma foi escolhido o Arduino nano para realizar a interface entre esses periféricos e o *Raspberry*. Baseado nisso, o Arduino passou a ser uma espécie de *driver* que serviu não somente para os periféricos citados, mas também, para os outros módulos. O segundo motivo foi a falta de memória do Arduino nano. Durante a construção do código, constatamos a dificuldade de manter as bibliotecas MRFC522 (módulo RFID) e IRremote (módulos IR) funcionando ao mesmo tempo no Arduino. Isso se deve ao fato de ambas as bibliotecas ocuparem muito espaço nas memórias dinâmica e estática do Arduino nano.

Com esse fato, desabilitamos a biblioteca IRremote e passamos a não utilizar mais os módulos IR. Para substituir esses módulos, a alternativa foi utilizar um dos canais do módulo relé para controlar a alimentação do ar-condicionado da sala de aula. Desse modo o módulo relé de 2 canais não foi mais viável. Para controle da trava da porta, da iluminação e do ar-condicionado, precisaríamos de um módulo relé de pelo menos 4 canais, o qual foi testado com sucesso, mas não foi implementado durante a demonstração.

A implementação do banco de dados para o cadastro e o registro dos usuários na plataforma *Firebase* foi realizada e testada com sucesso. Todos os cartões utilizados na demonstração, foram previamente cadastrados no banco de dados, permitindo o acesso dos usuários fictícios e registrando a data e o horário de acesso perfeitamente.

Para finalizar, a implementação que foi o diferencial deste projeto: a construção do aplicativo *RoomController*. Com ele foi possível, além de obter na palma da sua mão o registro dos acessos, o controle de todos os sistemas da sala de aula, utilizando para isso a internet. Com essa praticidade os usuários poderiam inclusive controlar a entrada das salas apenas utilizando o seu *smartphone*.

Diante de todas as dificuldades aqui apresentadas, o objetivo proposto pelo projeto foi alcançado. Este projeto será de grande relevância para instituições e para os alunos que desejarem realizar melhorias e expandir o seu uso.

CONCLUSÃO

A elaboração deste projeto visou construir um controle automático dos sistemas de iluminação e refrigeração para as salas de aula da Escola Superior de Tecnologia além de também controlar o acesso às mesmas. Foi aplicado todo o conhecimento adquirido durante a graduação e em inúmeros momentos conhecimentos além.

Foram apresentados breves conceitos e explicações sobre eficiência energética, sistemas microcontrolados, o microcomputador Raspberry Pi, a luz infravermelha e suas utilizações, o módulo RFID, a placa de prototipagem Arduino, o módulo leitor de impressões digitais e a plataforma de construção de aplicativos *web* e *mobile*, *Firestore*. A partir disto, o sistema foi construído utilizando diversas linguagens de programação, como *Python*, *Swift* e *C++*.

A idéia do projeto foi construir um sistema de acesso, que não existe nas salas de aulas da Escola Superior de Tecnologia, que fosse simples, confiável e de baixo custo, a fim de proporcionar à instituição uma nova metodologia de segurança e praticidade aos usuários das inúmeras salas de aula. Baseado nisso, temos um circuito protótipo ao qual os custos estipulados giram em torno de R\$ 400,00 (quatrocentos reais), com um custo de instalação que pode variar de R\$ 80,00 (oitenta reais) à R\$ 100,00 (cem reais) dependendo do fornecedor.

Se formos ao mercado de segurança eletrônica, podemos encontrar módulos acoplados à trava por eletroímã ou até mesmo fechadura eletrônica, ambas com acesso realizado por módulo RFID e/ou leitor biométrico, por volta de R\$ 2000,00 (dois mil reais) à R\$ 4000,00 (quatro mil reais) dependendo do modelo e dos adicionais. Porém, em nenhum desses modelos, há o automatismo proposto por este projeto e um *link* com uma rede de dados hospedada em nuvem.

Para trabalhos futuros, a necessidade da correção da capacidade de memória ainda continuará sendo o principal objetivo. As plataformas de desenvolvimento ainda perecem nesse campo e no momento em que os códigos exigem maior quantidade do uso de memória, os periféricos interrompem seu funcionamento.

Portanto, explicitada aqui as inúmeras vantagens deste projeto, concluo sua elaboração, atingindo o objetivo proposto inicialmente: construir um projeto de controle do acesso, refrigeração e iluminação de uma sala de aula, que fosse prático e de baixo custo.

REFERÊNCIAS BIBLIOGRÁFICAS

ABESCO. **O que é Eficiência Energética? (EE)**. São Paulo, 200?. Disponível em: <<http://www.abesco.com.br/pt/o-que-e-eficiencia-energetica-ee/>>. Acesso em: 12 de abril de 2018.

ADAFRUIT. **Super-bright 5mm IR LED – 940nm**. 2018?. Disponível em: <<https://www.adafruit.com/product/387>>. Acesso em: 12 de abril de 2018.

AMORIM, Robson Soares. **O que é um BasS? – Backend as a Service?** 2017. Disponível em: <<https://rsamorim.azurewebsites.net/2017/12/05/o-que-e-um-baas-backend-as-a-service/>>. Acesso em: 25 de outubro de 2018.

ANDR.OID ERIC. **Remote Desktop Connection from Windows 10 to Raspberry Pi**. 2015. Disponível em: <https://www.youtube.com/watch?v=FDdkxPv_Cg>. Acesso em: 09 de maio de 2018.

BARBOSA, Vanessa. **Os cinco países líderes em eficiência energética**. 2016. Disponível em: <<https://exame.abril.com.br/economia/os-5-paises-lideres-em-eficiencia-energetica/>>. Acesso em: 01 de maio de 2018.

BATSCHINSKI, George. **Backend as a Service: Prós e Contras**. 2016. Disponível em: <<https://www.infoq.com/br/news/2016/07/backend-pros-e-contras>>. Acesso em: 25 de outubro de 2018.

BRAGA, Newton. **Como funciona o RFID – Identificação por rádio frequência (ART1088)**. 201?. Disponível em: <<http://www.newtonbraga.com.br/index.php/como-funciona/6372-art1088>>. Acesso em: 28 de maio de 2018.

BRANCO, Renata. **O que é um diodo emissor de luz infravermelha?** 201?. Disponível em: <<http://www.manutencaoesuprimentos.com.br/conteudo/3902-o-que-e-um-diodo-emissor-de-luz-infravermelha/>>. Acesso em 28 de maio de 2018.

CANALTECH. **O que é biometria?** 2018?. Disponível em: <<https://canaltech.com.br/seguranca/O-que-e-biometria/>>. Acesso em: 08 de maio de 2018.

CARVALHO, Guilherme Melo. DA SILVA, Stephanie Cristhyne Araújo. **Infravermelho**. 2015. Disponível em: <<http://infravermelho.info/>>. Acesso em: 09 de maio de 2018.

CARVALHO, Juliana Mota. FLORES, Carlos Augusto Toledo, et.al. **A atuação do LED infravermelho no tratamento da dor fibromiálgica: Experiência do Enfermeiro. Encontro Latino Americano de Iniciação Científica e Pós-Graduação XIII INIC e IX EPG**. 2009, São José dos Campos, São Paulo. Disponível em: <http://www.inicepg.univap.br/cd/INIC_2009/anais/trabalhos_saude.html>. Acesso em: 12 de outubro de 2018.

CAVALCANTE, Marisa A. RODRIGUES, Thais T. Tavares. BUENO, Darlene A. **Controle Remoto: Princípio de funcionamento (parte 1 de 2)**. 2013. Disponível em: <<https://dialnet.unirioja.es/descarga/articulo/5165544.pdf>>. Acesso em: 14 de outubro de 2018.

CIRIACO, Douglas. **O que é Raspberry Pi?** 200-?. Disponível em: <<https://canaltech.com.br/hardware/o-que-e-raspberry-pi/>>. Acesso em: 01 de maio de 2018.

CLEVEROAD.COM. **A story of Firebase or your next favourite cloud-based service**. 2016. Disponível em: <<https://www.cleveroad.com/blog/a-story-of-firebase-or-your-next-favourite-cloud-based-service>>. Acesso em: 25 de outubro de 2018.

CONSUMO EM PAUTA. **Para que serve o selo Procel**. 2018?. Disponível em: <<http://www.consumoempauta.com.br/para-que-serve-o-selo-procel-2/>>. Acesso em: 01 de maio de 2018.

CORSO, Jorge Alberto. **O que é GNU/Linux**. 200-?. Disponível em: <<https://www.vivaolinux.com.br/linux/>>. Acesso em: 08 de maio de 2018.

DIAS, Henrique. **Como utilizar corretamente meu cartão RFID proximidade?** 2015. Disponível em: <<http://www.falex.com.br/blog/25-como-utilizar-corretamente-meu-cartao-rfid-proximidade>>. Acesso em: 13 de agosto de 2018.

ELETROBRAS. **Procel**. 2017?. Disponível em: <<http://eletrobras.com/pt/Paginas/Procel.aspx>>. Acesso em: 01 de maio de 2018.

ELETROGATE. **KIT Módulo RFID MRFC522 13,56 MHz**. 2018?. Disponível em: <https://www.eletrogate.com/kit-modulo-rfid-mfrc522-1356-mhz?utm_source=Site&utm_medium=GoogleMerchant&utm_campaign=GoogleMerchant&gclid=CjwKCAjw2MTbBRASEiwAdYIpsarHf4fC93pZhErbXLWXFsUBOCi3fbR64tol-58d9HoQkDXAMpd2qRoC_CkQAvD_BwE>. Acesso em: 10 de maio de 2018.

FIREBASE. **Documentação**. 2018?. Disponível em: <<https://firebase.google.com/docs/?hl=pt-br>>. Acesso em: 25 de outubro de 2018.

FILIFELOP.COM. **Pinagem Arduino**. 201-?. Disponível em: <<https://www.filipeflop.com/blog/ethernet-shield-enc28j60-arduino-nano/pinagem-arduino/>>. Acesso em: 04 de novembro de 2018.

GUANABARA, Gustavo. **O que é um banco de dados?** 2016. Disponível em: <https://www.youtube.com/watch?v=Ofktsne-utM&list=PLHz_AreHm4dkBs-795Dsgvau_ekxg8g1r>. Acesso em: 09 de maio de 2018.

INSOFT4. **O que é RFID e para que serve?** 201-?. Disponível em: <<http://insoft4aps.com.br/noticia/O-que-e-RFID-e-para-que-serve>>. Acesso em: 09 de maio de 2018.

INSTITUTO NACIONAL DE EFICIÊNCIA ENERGÉTICA – INEE. **O que é eficiência energética?** 201-?. Disponível em: <www.inee.org.br/eficiencia_o_que_eh.asp?Cat=eficiencia>. Acesso em: 01 de maio de 2018.

JÚNIOR, Joab Silas Da Silva. **O que é infravermelho?** *Brasil Escola*. 201-?. Disponível em <<https://brasilecola.uol.com.br/o-que-e/fisica/o-que-e-infravermelho.htm>>. Acesso em 20 de novembro de 2018

KOYANAGI, Fernando. **ESP32 com RFID: Controle de Acesso**. 2018. Disponível em: <<https://www.fernandok.com/2018/02/esp32-com-rfid-controle-de-acesso.html>>. Acesso em: 03 de novembro de 2018.

LAYTON, Julia. **Você sabe como funciona o controle remoto da sua TV?** 2018?. Disponível em: <<http://www.crcontroleremoto.com.br/como-funciona-um-controle-remoto.html>>. Acesso em: 09 de maio de 2018.

MACHADO, Jonathan. **Raspberry Pi: computador de 50 reais pode revolucionar a informática**. 2012. Disponível em: <<https://m.tecmundo.com.br/hardware/23175-raspberry-pi-como-um-computador-de-50-reais-pode-revolucionar-a-informatica.htm>>. Acesso em: 01 de maio de 2018.

MARTINS, Elaine. **O que é biometria?** 2009. Disponível em: <<https://www.tecmundo.com.br/o-que-e/3121-o-que-e-biometria-.htm>>. Acesso em: 03 de agosto de 2018.

MASTERWALK SHOP. **Módulo LED Emissor Infravermelho (IR) KY-005**. 2018?. Disponível em: <<https://www.masterwalkershop.com.br/modulo-led-emissor-infravermelho-ir-ky-005>>. Acesso em: 28 de maio de 2018.

MIRELLA, Andressa. **Controlando lâmpadas com módulo relé Arduino**. 2015. Disponível em: <<https://ardudino.wordpress.com/2015/12/12/controlando-lampadas-com-modulo-rele-arduino/>>. Acesso em 06 de novembro de 2018.

MSECCO. **Cartão de proximidade passivo**. 2009. Disponível em: <<http://acessoeponto.mixlog.com.br/artigo/cartao-de-proximidade-passivo/>>. Acesso em: 28 de maio de 2018.

MULTIPECASTEC. **Módulo Emissor de IR P18 para Arduino**. 2018?. Disponível em: <<http://www.multipecastec.com.br/loja/produto/modulo-emissor-de-ir-p18-para-arduino>>. Acesso em: 29 de maio de 2018.

NORDIC APIS. **Você sabe as diferenças entre API e SDK?** 2016. Disponível em: <<https://mundoapi.com.br/materias/voce-sabe-as-diferencas-entre-api-e-sdk/>>. Acesso em: 25 de outubro de 2018.

RASPBERRY PI FOUNDATION. **About us.** 2018?. Disponível em: <<https://www.raspberrypi.org/about/>>. Acesso em: 07 de maio de 2018.

RASPBERRY PI PRESS. **The official Raspberry Pi Beginner's Book – Volume 1 (2018).** 2018?. Disponível em: <<https://store.rpipress.cc/products/the-official-raspberry-pi-beginners-book-volume-1-2018>>. Acesso em: 08 de maio de 2018.

SILVEIRA, Cristiano B.. LOPES, Guilherme C.. **O que é RFID?** 201-?. Disponível em: <<https://www.citisystems.com.br/rfid/>>. Acesso em: 10 de maio de 2018.

SMARTS PROJECTS BRASIL. **Sensor leitor biométrico de impressão digital.** 2018?. Disponível em: <<https://www.smartprojectsbrasil.com.br/product-page/sensor-leitor-biometrico-de-impressao-digital>>. Acesso em: 25 de outubro de 2018.

SOARES, Karla. **O que é um Arduino e o que pode ser feito com ele?** 2013. Disponível em: < <https://www.techtudo.com.br/noticias/noticia/2013/10/o-que-e-um-arduino-e-o-que-pode-ser-feito-com-ele.html>>. Acesso em: 12 de outubro de 2018.

SPARKFUN. **What is an Arduino?** 2014. Disponível em: < <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>>. Acesso em: 12 outubro de 2018.

TEKNEW. **Leitor biométrico digital persona 4000.** Disponível em: <<https://www.teknew.com.br/informatica/leitor-biometrico-digital-persona-4000/>>. Acesso em: 13 de maio de 2018.

THE OPEN SOURCERY. **Google Blockly, MIT Scratch, and Google to MIT App Inventor.** 2018?. Disponível em: <<https://www.theopensourcery.com/keepopen/2012/google-blockly-mit-scratch-and-google-to-mit-app-inventor/>>. Acesso em: 01 de maio de 2018.

THOMSEN, Adilson. **Primeiros passos com Raspberry Pi e Linux.** 2018?. Disponível em: <<https://www.filipeflop.com/blog/primeiros-passos-raspberry-pi-e-linux/>>. Acesso em: 01 de maio de 2018.

UNIÃO EUROPÉIA. **Energia.** 201-?. Disponível em: <https://europa.eu/european-union/topics/energy_pt>. Acesso em: 01 de maio de 2018.

VANI, Anderson. **O que é leitor biométrico?** 2012. Disponível em: <<http://forum.techtudo.com.br/perguntas/26743/o-que-e-leitor-biometrico>>. Acesso em 21 de maio de 2018.

VIDA DE SILÍCIO. **Módulo Receptor Infravermelho – GBK.** 2018?. Disponível em: <<https://www.vidadesilicio.com.br/receptor-infravermelho-gbk>>. Acesso em: 14 de outubro de 2018.

OBRAS CONSULTADAS

ALVES, Josemar. **RFID – Cartões Mifare.** Disponível em: <<https://www.embarcados.com.br/rfid-cartoes-mifare/>>. Acesso em: 28 de maio de 2018.

CIRIACO, Douglas. **Como funciona a RFID?** 2018?. Disponível em: <<https://www.tecmundo.com.br/tendencias/2601-como-funciona-a-rfid-.htm>>. Acesso em: 28 de maio de 2018.

CIPOLI, Pedro. **O que é um SoC?** Disponível em: <<https://canaltech.com.br/hardware/O-que-e-um-SoC/>>. Acesso em: 09 de maio de 2018.

COMO FAZER AS COISAS. **Raspberry Pi o que é? Para que serve? Conheça em detalhes.** Disponível em: <<http://www.comofazerascosas.com.br/raspberry-pi-o-que-e-para-que-serve-conheca-em-detalhes.html>>. Acesso em: 01 de maio de 2018.

FIREBIRD. **Downloads: Server Packages.** Disponível em: <<https://www.firebirdsql.org/en/server-packages/>>. Acesso em: 09 de maio de 2018.

GARRETT, Filipe. **Como funciona o Raspberry Pi? Entenda a tecnologia e sua aplicabilidade.** Disponível em <<http://www.techtudo.com.br/noticias/noticia/2014/11/como-funciona-o-raspberry-pi-entenda-tecnologia-e-sua-aplicabilidade.html>>. Acesso em: 01 de maio de 2018.

GUERREIRO, Yan. **O que é a tecnologia RFID e como ela pode ajudar sua empresa?** Disponível em: <<https://rfidbrasil.com/blog/o-que-e-a-tecnologia-rfid-e-como-ela-pode-ajudar-sua-empresa/>>. Acesso em: 10 de maio de 2018.

GUILHERME, Paulo. **Finalmente temos um Raspberry Pi com WiFi de 5 GHz e Bluetooth 4.2.** Disponível em: <<https://m.tecmundo.com.br/produto/128237-finalmente-temos-raspberry-pi-wifi-5-ghz-bluetooth-4-2.htm>>. Acesso em: 01 de maio de 2018.

ISO. **ISO/IEC 13249-6:2006.** Disponível em: <<https://www.iso.org/standard/38648.html>>. Acesso em: 09 de maio de 2018.

MACHADO, Jonathan. **O que é GPU?** Disponível em: <<https://m.tecmundo.com.br/hardware/1127-o-que-e-gpu-.htm>>. Acesso em: 08 de maio de 2018.

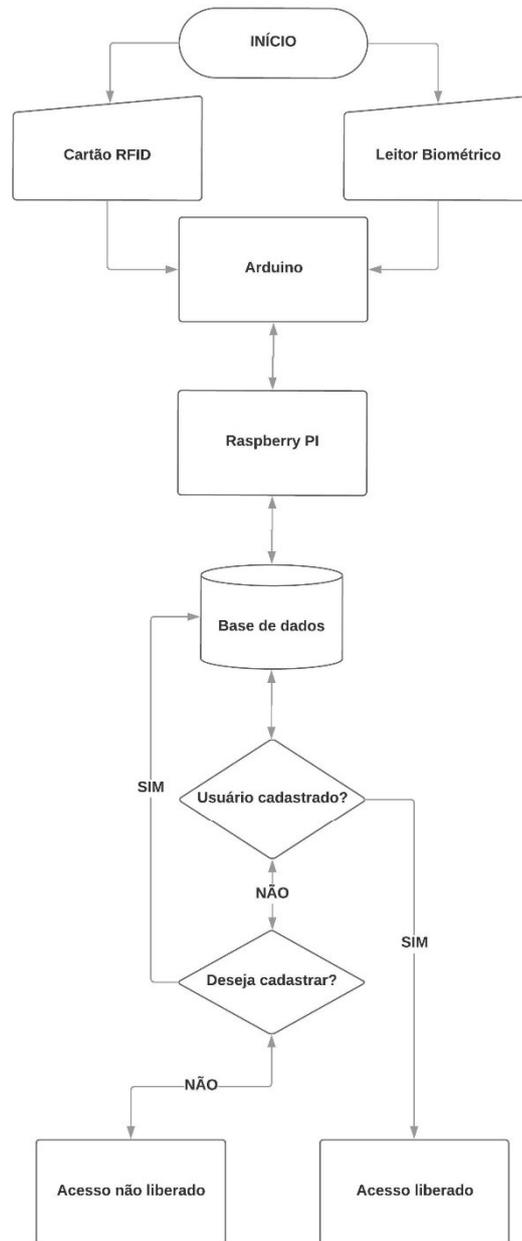
MYSQL. **MySQL Documentation.** Disponível em: <<https://dev.mysql.com/doc/>>. Acesso em: 09 de maio de 2018.

POSTGRESQL. **Documentation.** Disponível em: <<https://www.postgresql.org/docs/>>. Acesso em: 09 de maio de 2018.

APÊNDICE A – FLUXOGRAMA DE FUNCIONAMENTO

FLUXOGRAMA DE FUNCIONAMENTO

Diorge | November 3, 2018



APÊNDICE B – CÓDIGO FONTE DO PROJETO

```
#include <ArduinoJson.h>
#include <SPI.h>
#include <MFRC522.h>
#include <base64.hpp>
//#include <IRremote.h>

// RFID Pins
#define RST_PIN 9
#define SS_PIN 10

// LED Pins
#define BLUE_LED A4
#define YELLOW_LED A2
#define RED_LED A3
#define IR_LED 3
#define IR_RECV 2

// Relay Channels
#define CHNL_1 5
#define CHNL_2 4

// Timeouts in milliseconds
#define LEARNING_TIMEOUT 30000

//unsigned int AC_ON_COMMAND[256];
//int AC_ON_LEN = 0;
//unsigned int AC_OFF_COMMAND[256];
//int AC_OFF_LEN = 0;

MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance
//IRrecv irrecv(IR_RECV);
```

```
//IRsend irsend;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  SPI.begin(); // Init SPI bus
  mfrc522.PCD_Init(); // Init MFRC522

  pinMode(BLUE_LED, OUTPUT);
  pinMode(YELLOW_LED, OUTPUT);
  pinMode(RED_LED, OUTPUT);

  pinMode(CHNL_1, OUTPUT);
  pinMode(CHNL_2, OUTPUT);

  // irrecv.enableIRIn();

  digitalWrite(CHNL_1, HIGH);
  digitalWrite(CHNL_2, HIGH);

  openDoor();
}

void loop() {

  /// Section 1 - Command parsing from RPi

  String piCommand = "";

  while (Serial.available() > 0) {
    char received = Serial.read();
    piCommand += received;
  }
}
```

```

if (piCommand == "learn_ac_on\n") {
    // learnAcOn();
} else if (piCommand == "learn_ac_off\n") {
    // learnAcOff();
} else if (piCommand == "turn_ac_on\n") {
    // sendAcOn();
} else if (piCommand == "turn_ac_off\n") {
    // sendAcOff();
} else if (piCommand == "unlock\n") {
    openDoor();
} else if (piCommand == "lock\n") {
    closeDoor();
} else if (piCommand == "lights_on\n") {
    lightsOn();
} else if (piCommand == "lights_off\n") {
    lightsOff();
}

```

```
Serial.println(piCommand);
```

```
/// Section 2 - RFID detection code
```

```

StaticJsonDocument<128> doc;
JsonObject root = doc.to<JsonObject>();

// Look for new cards, and select one if present
if ( ! mfr522.PICC_IsNewCardPresent() ) {
    root["task"] = "none";
    serializeJson(root, Serial);
    delay(500);
    return;
}

```

```

if (! mfr522.PICC_ReadCardSerial()) {
    return;
}
// If a card's been detected, encode its UID and send it over via serial
char data[256];
int b64len = encode_base64(mfr522.uid.uidByte, mfr522.uid.size, data);

root["task"] = "card_detected";
root["uid"] = String(data);
serializeJson(root, Serial);

delay(500);

}

//void learnAcOn(){
// long startTime = millis();
// bool learned = false;
// digitalWrite(BLUE_LED, HIGH);
//
// while(millis() - startTime < LEARNING_TIMEOUT && !learned){
//
// decode_results results;
//
// if(irrecv.decode(&results)){
//     for(int i = 0; i < results.rawlen; i++){
//         AC_ON_COMMAND[i] = results.rawbuf[i] * USECPERTICK;
//     }
//
//     AC_ON_LEN = results.rawlen;
//     learned = true;
//     irrecv.resume();
// }
//

```

```
// }  
//  
// digitalWrite(BLUE_LED, LOW);  
//  
//}  
//  
//void learnAcOff(){  
// long startTime = millis();  
// bool learned = false;  
// digitalWrite(BLUE_LED, HIGH);  
//  
// while(millis() - startTime < LEARNING_TIMEOUT && !learned){  
//  
//   decode_results results;  
//  
//   if(irrecv.decode(&results)){  
//     for(int i = 0; i < results.rawlen; i++){  
//       AC_OFF_COMMAND[i] = results.rawbuf[i] * USECPERTICK;  
//     }  
//  
//     AC_OFF_LEN = results.rawlen;  
//     learned = true;  
//     irrecv.resume();  
//   }  
//  
// }  
// }  
//  
// digitalWrite(BLUE_LED, LOW);  
//  
//}  
//  
//void sendAcOn(){  
// digitalWrite(RED_LED, HIGH);  
//
```

```
// int khz = 38;
// irsend.sendRaw(AC_ON_COMMAND, AC_ON_LEN, khz);
//
// delay(500);
//
// digitalWrite(RED_LED, LOW);
//}
//
//void sendAcOff(){
//
// digitalWrite(RED_LED, HIGH);
//
// int khz = 32;
// irsend.sendRaw(AC_OFF_COMMAND, AC_OFF_LEN, khz);
//
// delay(500);
//
// digitalWrite(RED_LED, LOW);
//
//}
void openDoor() {
    digitalWrite(CHNL_1, LOW);
}
void closeDoor() {
    digitalWrite(CHNL_1, HIGH);
}
void lightsOn() {
    digitalWrite(CHNL_2, LOW);
}
void lightsOff() {
    digitalWrite(CHNL_2, HIGH);
}
```

APÊNDICE C – CÓDIGO EM PYTHON PARA O RASPBERRY PI

```

import json
import serial
import firebase_admin

from firebase_admin import credentials
from firebase_admin import db
from time import sleep

## Serial port configuration

port = "/dev/tty.usbserial-1440"
#port = "/dev/ttyACM0"

serialport = serial.Serial()
serialport.baudrate = 115200
serialport.port = port
serialport.open()

## Cloud services configuration

cred = credentials.Certificate("firebase_creds.json")
firebase_admin.initialize_app(cred)

dbRoot = db.reference()
dbRoot.child('task').listen(onUpdate)

def onUpdate(obj):
    print(obj)

def dispatch(blob):
    obj = json.loads(blob)
    task = obj["task"]

    # This is the only task that should come from
    # the arduino device driver itself
    if task == "card_detected":
        card_detected(obj["card_data"])
        pass
    if task == "none":
        print("none")
        pass

    # These are the tasks that can come from the Firebase DB
    if task == "learn_card":
        pass
    if task == "learn_ac_on":
        pass
    if task == "learn_ac_off":
        pass
    if task == "turn_ac_on":
        pass
    if task == "turn_ac_off":
        pass
    if task == "unlock":
        pass
    if task == "lock":
        pass
    if task == "lights_on":

```

```
        pass
    if task == "lights_off":
        pass

# This function is responsible for checking
# if a card is present on the database and
# unlocking the door if user is allowed.
# Should take in a string with the card uid.
def card_detected(card_data):
    return

def sendObject(obj):
    serialized = json.dumps(obj)

    if serialport.is_open == True:
        serialport.write(serialized)
        serialport.flush()

def sendString(obj):

    if serialport.is_open == True:
        serialport.write(obj)
        serialport.flush()

while True:

    # Grabbing data from driver Arduino
    string = serialport.readline()

    # If there's anything in the buffer,
    if len(string) != 0 and string != "\n":
        dispatch(string)

    # Sleep for 50ms to avoid excessive CPU usage
    sleep(0.05)
```

ANEXO A – CÓDIGO TESTE DO MÓDULO RECEPTOR INFRAVERMELHO

```

//-----
//-----
// Include the IRremote library header
//
#include <IRremote.h>

//-----
//-----
// Tell IRremote which Arduino pin is connected to the IR
Receiver (TSOP4838)
//
int recvPin = 11;
IRrecv irrecv(recvPin);

//+=====
=====
// Configure the Arduino
//
void setup ( )
{
  Serial.begin(9600); // Status message will be sent to PC at
9600 baud
  irrecv.enableIRIn(); // Start the receiver
}

//+=====
=====
// Display IR code
//
void ircode (decode_results *results)
{
  // Panasonic has an Address
  if (results->decode_type == PANASONIC) {
    Serial.print(results->address, HEX);
    Serial.print(":");
  }

  // Print Code
  Serial.print(results->value, HEX);
}

//+=====
=====
// Display encoding type
//
void encoding (decode_results *results)
{
  switch (results->decode_type) {
    default:
    case UNKNOWN:      Serial.print("UNKNOWN");      break ;
  }
}

```

```

    case NEC:          Serial.print("NEC");          break ;
    case SONY:         Serial.print("SONY");         break ;
    case RC5:          Serial.print("RC5");          break ;
    case RC6:          Serial.print("RC6");          break ;
    case DISH:         Serial.print("DISH");         break ;
    case SHARP:        Serial.print("SHARP");        break ;
    case JVC:          Serial.print("JVC");          break ;
    case SANYO:        Serial.print("SANYO");        break ;
    case MITSUBISHI:   Serial.print("MITSUBISHI");   break ;
    case SAMSUNG:      Serial.print("SAMSUNG");      break ;
    case LG:           Serial.print("LG");           break ;
    case WHYNTER:     Serial.print("WHYNTER");      break ;
    case AIWA_RC_T501: Serial.print("AIWA_RC_T501"); break ;
    case PANASONIC:    Serial.print("PANASONIC");    break ;
    case DENON:        Serial.print("Denon");        break ;
}
}

//+=====
=====
// Dump out the decode_results structure.
//
void dumpInfo (decode_results *results)
{
    // Check if the buffer overflowed
    if (results->overflow) {
        Serial.println("IR code too long. Edit IRremoteInt.h and
increase RAWBUF");
        return;
    }

    // Show Encoding standard
    Serial.print("Encoding : ");
    encoding(results);
    Serial.println("");

    // Show Code & length
    Serial.print("Code : ");
    ircode(results);
    Serial.print(" (");
    Serial.print(results->bits, DEC);
    Serial.println(" bits)");
}

//+=====
=====
// Dump out the decode_results structure.
//
void dumpRaw (decode_results *results)
{
    // Print Raw data

```

```

Serial.print("Timing[");
Serial.print(results->rawlen-1, DEC);
Serial.println("]: ");

for (int i = 1; i < results->rawlen; i++) {
    unsigned long x = results->rawbuf[i] * USECPERTICK;
    if (!(i & 1)) { // even
        Serial.print("-");
        if (x < 1000) Serial.print(" ");
        if (x < 100) Serial.print(" ");
        Serial.print(x, DEC);
    } else { // odd
        Serial.print(" ");
        Serial.print("+");
        if (x < 1000) Serial.print(" ");
        if (x < 100) Serial.print(" ");
        Serial.print(x, DEC);
        if (i < results->rawlen-1) Serial.print(", "); // ',' not
needed for last one
    }
    if (!(i % 8)) Serial.println("");
}
Serial.println(""); // Newline
}

//+=====
=====
// Dump out the decode_results structure.
//
void dumpCode (decode_results *results)
{
    // Start declaration
    Serial.print("unsigned int "); // variable type
    Serial.print("rawData["); // array name
    Serial.print(results->rawlen - 1, DEC); // array size
    Serial.print("] = {"); // Start declaration

    // Dump data
    for (int i = 1; i < results->rawlen; i++) {
        Serial.print(results->rawbuf[i] * USECPERTICK, DEC);
        if (i < results->rawlen-1) Serial.print(","); // ',' not
needed on last one
        if (!(i & 1)) Serial.print(" ");
    }

    // End declaration
    Serial.print("};"); //

    // Comment
    Serial.print(" // ");
    encoding(results);
    Serial.print(" ");
}

```

```

ircode(results);

// Newline
Serial.println("");

// Now dump "known" codes
if (results->decode_type != UNKNOWN) {

    // Some protocols have an address
    if (results->decode_type == PANASONIC) {
        Serial.print("unsigned int  addr = 0x");
        Serial.print(results->address, HEX);
        Serial.println(";");
    }

    // All protocols have data
    Serial.print("unsigned int  data = 0x");
    Serial.print(results->value, HEX);
    Serial.println(";");
}
}

//+=====
=====
// The repeating section of the code
//
void loop ( )
{
    decode_results  results;           // Somewhere to store the
results

    if (irrecv.decode(&results)) { // Grab an IR code
        dumpInfo(&results);         // Output the results
        dumpRaw(&results);          // Output the results in RAW
format
        dumpCode(&results);         // Output the results as source
code
        Serial.println("");         // Blank line between entries
        irrecv.resume();            // Prepare for the next value
    }
}

```

ANEXO B – CÓDIGO TESTE DO MÓDULO EMISSOR DE INFRAVERMELHO

```

/*
 * IRremote: IRsendRawDemo - demonstrates sending IR codes with
sendRaw
 * An IR LED must be connected to Arduino PWM pin 3.
 * Version 0.1 July, 2009
 * Copyright 2009 Ken Shirriff
 * http://arcfn.com
 *
 * IRsendRawDemo - added by AnalysIR (via www.AnalysIR.com), 24
August 2015
 *
 * This example shows how to send a RAW signal using the IRremote
library.
 * The example signal is actually a 32 bit NEC signal.
 * Remote Control button: LGTV Power On/Off.
 * Hex Value: 0x20DF10EF, 32 bits
 *
 * It is more efficient to use the sendNEC function to send NEC
signals.
 * Use of sendRaw here, serves only as an example of using the
function.
 */
#include <IRremote.h>

IRsend irsend;

void setup()
{

}

void loop() {
  int khz = 38; // 38kHz carrier frequency for the NEC protocol
  unsigned int irSignal[] = {9000, 4500, 560, 560, 560, 560, 560,
1690, 560, 560, 560, 560, 560, 560, 560, 560, 560, 560, 560, 1690,
560, 1690, 560, 560, 560, 1690, 560, 1690, 560, 1690, 560, 1690,
560, 1690, 560, 560, 560, 560, 560, 560, 560, 1690, 560, 560, 560,
560, 560, 560, 560, 560, 560, 1690, 560, 1690, 560, 1690, 560,
560, 560, 1690, 560, 1690, 560, 1690, 560, 1690, 560, 39416, 9000,
2210, 560}; //AnalysIR Batch Export (IRremote) - RAW

  irsend.sendRaw(irSignal, sizeof(irSignal) / sizeof(irSignal[0]),
khz); //Note the approach used to automatically calculate the
size of the array.

  delay(5000); //In this example, the signal will be repeated
every 5 seconds, approximately.
}

```

ANEXO C – CÓDIGO TESTE DO MÓDULO RFID

```

/**
 * -----
 * This is a MFRC522 library example; see
 https://github.com/miguelbalboa/rfid
 * for further details and other examples.
 *
 * NOTE: The library file MFRC522.h has a lot of useful info.
 Please read it.
 *
 * Released into the public domain.
 * -----
 * This sample shows how to read and write data blocks on a
 MIFARE Classic PICC
 * (= card/tag).
 *
 * BEWARE: Data will be written to the PICC, in sector #1 (blocks
 #4 to #7).
 *
 * Typical pin layout used:
 * -----
 *
 * MFRC522      Arduino      Arduino      Arduino
 Arduino      Reader/PCD  Uno/101      Mega         Nano v3
 Leonardo/Micro Pro Micro
 * Signal      Pin          Pin          Pin          Pin
 Pin          Pin
 * -----
 * RST/Reset    RST           9            5            D9
 RESET/ICSP-5  RST
 * SPI SS       SDA(SS)      10           53           D10          10
 10
 * SPI MOSI     MOSI         11 / ICSP-4  51           D11
 ICSP-4        16
 * SPI MISO     MISO         12 / ICSP-1  50           D12
 ICSP-1        14
 * SPI SCK      SCK          13 / ICSP-3  52           D13
 ICSP-3        15
 *
 */

#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN          9           // Configurable, see typical
pin layout above

```

```

#define SS_PIN          10          // Configurable, see typical
pin layout above

MFRC522 mfr522(SS_PIN, RST_PIN); // Create MFRC522 instance.

MFRC522::MIFARE_Key key;

/**
 * Initialize.
 */
void setup() {
  Serial.begin(9600); // Initialize serial communications with
the PC
  while (!Serial); // Do nothing if no serial port is opened
(added for Arduinos based on ATMEGA32U4)
  SPI.begin(); // Init SPI bus
  mfr522.PCD_Init(); // Init MFRC522 card

  // Prepare the key (used both as key A and as key B)
  // using FFFFFFFFh which is the default at chip delivery
from the factory
  for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
  }

  Serial.println(F("Scan a MIFARE Classic PICC to demonstrate
read and write.));
  Serial.print(F("Using key (for A and B):"));
  dump_byte_array(key.keyByte, MFRC522::MF_KEY_SIZE);
  Serial.println();

  Serial.println(F("BEWARE: Data will be written to the PICC,
in sector #1"));
}

/**
 * Main loop.
 */
void loop() {
  // Look for new cards
  if ( ! mfr522.PICC_IsNewCardPresent())
    return;

  // Select one of the cards
  if ( ! mfr522.PICC_ReadCardSerial())
    return;

  // Show some details of the PICC (that is: the tag/card)
  Serial.print(F("Card UID:"));
  dump_byte_array(mfr522.uid.uidByte, mfr522.uid.size);
  Serial.println();
  Serial.print(F("PICC type: "));

```

```

MFRC522::PICC_Type piccType =
mfrc522.PICC_GetType(mfrc522.uid.sak);
Serial.println(mfrc522.PICC_GetTypeName(piccType));

// Check for compatibility
if ( piccType != MFRC522::PICC_TYPE_MIFARE_MINI
    && piccType != MFRC522::PICC_TYPE_MIFARE_1K
    && piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
    Serial.println(F("This sample only works with MIFARE
Classic cards.));
    return;
}

// In this sample we use the second sector,
// that is: sector #1, covering block #4 up to and including
block #7
byte sector          = 1;
byte blockAddr       = 4;
byte dataBlock[]     = {
    0x01, 0x02, 0x03, 0x04, // 1, 2, 3, 4,
    0x05, 0x06, 0x07, 0x08, // 5, 6, 7, 8,
    0x09, 0x0a, 0xff, 0x0b, // 9, 10, 255, 11,
    0x0c, 0x0d, 0x0e, 0x0f // 12, 13, 14, 15
};
byte trailerBlock    = 7;
MFRC522::StatusCode status;
byte buffer[18];
byte size = sizeof(buffer);

// Authenticate using key A
Serial.println(F("Authenticating using key A..."));
status = (MFRC522::StatusCode)
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
trailerBlock, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("PCD_Authenticate() failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}

// Show the whole sector as it currently is
Serial.println(F("Current data in sector:"));
mfrc522.PICC_DumpMifareClassicSectorToSerial(&(mfrc522.uid),
&key, sector);
Serial.println();

// Read data from the block
Serial.print(F("Reading data from block "));
Serial.print(blockAddr);
Serial.println(F(" ..."));
status = (MFRC522::StatusCode) mfrc522.MIFARE_Read(blockAddr,
buffer, &size);

```

```

if (status != MFRC522::STATUS_OK) {
    Serial.print(F("MIFARE_Read() failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
}
Serial.print(F("Data in block ")); Serial.print(blockAddr);
Serial.println(F(":"));
dump_byte_array(buffer, 16); Serial.println();
Serial.println();

// Authenticate using key B
Serial.println(F("Authenticating again using key B..."));
status = (MFRC522::StatusCode)
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_B,
trailerBlock, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("PCD_Authenticate() failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}

// Write data to the block
Serial.print(F("Writing data into block "));
Serial.print(blockAddr);
Serial.println(F(" ..."));
dump_byte_array(dataBlock, 16); Serial.println();
status = (MFRC522::StatusCode)
mfrc522.MIFARE_Write(blockAddr, dataBlock, 16);
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("MIFARE_Write() failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
}
Serial.println();

// Read data from the block (again, should now be what we
have written)
Serial.print(F("Reading data from block "));
Serial.print(blockAddr);
Serial.println(F(" ..."));
status = (MFRC522::StatusCode) mfrc522.MIFARE_Read(blockAddr,
buffer, &size);
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("MIFARE_Read() failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
}
Serial.print(F("Data in block ")); Serial.print(blockAddr);
Serial.println(F(":"));
dump_byte_array(buffer, 16); Serial.println();

// Check that data in block is what we have written
// by counting the number of bytes that are equal
Serial.println(F("Checking result..."));
byte count = 0;

```

```

    for (byte i = 0; i < 16; i++) {
        // Compare buffer (= what we've read) with dataBlock (=
        // what we've written)
        if (buffer[i] == dataBlock[i])
            count++;
    }
    Serial.print(F("Number of bytes that match = "));
    Serial.println(count);
    if (count == 16) {
        Serial.println(F("Success :-"));
    } else {
        Serial.println(F("Failure, no match :-("));
        Serial.println(F(" perhaps the write didn't work
properly..."));
    }
    Serial.println();

    // Dump the sector data
    Serial.println(F("Current data in sector:"));
    mfrc522.PICC_DumpMifareClassicSectorToSerial (&(mfrc522.uid),
&key, sector);
    Serial.println();

    // Halt PICC
    mfrc522.PICC_HaltA();
    // Stop encryption on PCD
    mfrc522.PCD_StopCryptol();
}

/**
 * Helper routine to dump a byte array as hex values to Serial.
 */
void dump_byte_array(byte *buffer, byte bufferSize) {
    for (byte i = 0; i < bufferSize; i++) {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");
        Serial.print(buffer[i], HEX);
    }
}

```