Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

KELVIN FERRI BRANCALHÃO

MÉTODOS DE SEGURANÇA PARA COMUNICAÇÃO RFID.

Assis/SP
2017

Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

KELVIN FERRI BRANCALHÃO

MÉTODOS DE SEGURANÇA PARA COMUNICAÇÃO RFID.

Projeto de pesquisa apresentado ao curso de Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando(a): Kelvin Ferri Brancalhão
Orientador(a): Profº Me. Fábio Eder Cardoso

Assis/SP
2017

MÉTODOS DE SEGURANÇA PARA COMUNICAÇÃO RFID.

KELVIN FERRI BRANCALHÃO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

B816m BRANCALHÃO, Kelvin Ferri
Métodos de segurança para tecnologia RFID / Kelvin Ferri Brancalhão.-- Assis, 2017.
36p.

Trabalho de conclusão do curso (Ciência da Computação). – Fundação Educacional do Município de Assis-FEMA

Orientador: Ms. Fábio Eder Cardoso

1.Segurança 2.RFID 3.Arduíno

CDD 005.365

Orientador: _____
Profº Me. Fábio Éder Cardoso

Examinador: _____
Profº Me. Douglas Sanches da Cunha

Assis/SP
2017

DEDICATÓRIA

Dedico este trabalho aos meus pais Gilmar e Sueli Brancalhão, que são as velas que impulsionam o navio da minha vida.

AGRADECIMENTOS

Agradeço em primeiro lugar a minha família que sempre esteve comigo me apoiando e ajudando com o que fosse possível sem nunca me deixar desanimar nem desistir, em especial aos meus pais Gilmar e Sueli Brancalhão que são de extrema importância na minha vida.

Agradeço aos professores que ministraram aulas para a turma de Ciência da Computação, por todo o conhecimento transmitido, em especial ao professor Fábio Éder Cardoso pela ajuda na elaboração desse trabalho.

Agradeço aos meus amigos de sala pela ajuda em todos os trabalho e estudando para as provas alem de todo o tempo que passamos juntos e todo o conhecimento que me foi passado dentro e fora da aula.

E Agradeço a todos os envolvidos direta ou indiretamente na elaboração deste trabalho pela ajuda.

> “É muito melhor lançar-se em busca de conquistas grandiosas, mesmo expondo-se ao fracasso, do que alinhar-se com os pobres de espírito, que nem gozam muito nem sofrem muito, porque vivem numa penumbra cinzenta, onde não conhecem nem vitória, nem derrota.” (Theodore Roosevelt)

RESUMO

Nas ultimas décadas, a tecnologia RFID (*Radio Frequency Identification*) tem se tornado muito popular, sendo de extrema necessidade nos processos onde são usados, seja na verificação de identidade, controle de acesso ou quaisquer outros.

Um sistema RFID é composto por um sistema leitor e uma etiqueta (*tag*). As *tags* como são conhecidas podem estar dentro de celulares e aparelhos eletrônicos, placas, cartões de acesso, ou até mesmo embutidos na pele. Como o custo de implementação é extremamente baixo a tendência é que ela tenha um crescimento exponencial.

Hoje pode-se encontrar tal tecnologia em cartões de passes para ônibus, controles de acesso em empresas, rastreamento e identificação de encomendas. Apesar da ampla utilização e popularidade há vários problemas com a sua segurança e na forma de como a comunicação é feita. Neste trabalho, é proposta uma análise acerca das principais duvidas existentes sobre o sistema, assim como um levantamento de suas vulnerabilidades e as possíveis soluções.

Palavras-chave: segurança, RFID, identificação.

ABSTRACT

In recent decades, the RFID (Radio Frequency identification) technology has become very popular, being of extreme necessity in the processes where they are used, be it in identity verification, acess control or any others.

Any RFID system consists of a reader system and a tag. The tags as they are known may be inside cell phone and eletronic gadgets, boards, access cards or even embedded in the skin. As the implementation cost is extremely low, the trend is for it to grow exponentially.

Today, such technology can be found in bus passes, access controls in companies, traking and identificationor orders. Despite the wide usage and popularity there are several problems with your security and in the way of how communication is done. In this work, an analysis is proposed about the mais doubts about the system, as well as a survey os its vulnerabilities and possible solutions.

Keywords: Security, RFID, Identification.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

Atualmente os meios de comunicação e a troca de informações estão ficando cada vez mais rápidas e diversificadas, sendo usados nos mais diversos campos e para as mais amplas aplicações, como por exemplo os cartões de passes nos ônibus e metrôs.

Com o avanço dessa tecnologia chamada de RFID e a aplicação a transações financeiras, torna-se necessário um sistema seguro para assegurar os dados dos usuários, como por exemplo a implementação de criptografia, que pode ocultar e embaralhar os dados para que a leitura seja impossível para quem não possui o sistema adequado de leitura como dito por Stallings em “Criptografia e segurança de redes” (2007).

Existem vários tipos de chips que podem armazenar informações e que possuem uma frequência de leitura identificável pelos sistemas RFID, e são amplamente utilizados atualmente em meios de transporte publico para validar o valor da passagem e a identificação da pessoa, assim como em parques para facilitar o pagamento e agilizar filas.

Os sistemas de leitura e gravação podem ser realizados de várias formas, como é exemplificado por Thomsen(2015). O presente trabalho apresentará o uso da tecnologia arduíno uno com módulo de leitura e gravação RFID, linguagem Java para realizar a criptografia dos dados e para testar as dados de outros cartões e verificar o processo de proteção dos dados dos usuários nas empresas que utilizam a tecnologia RFID.

1.1. MOTIVAÇÃO.

Com o surgimento de meios de transações de informações e armazenamento por meio da tecnologia RFID, é necessário implementar métodos de segurança para preservar os dados pessoais, financeiros dos usuários. Como motivação do presente trabalho, há a necessidade do desenvolvimento de um sistema para proteger os referidos dados e verificar se estes não estão vulneráveis e passivos de exploração dessas vulnerabilidades, resultando no benefício para a sociedade, bem como, um desafio a ser enfrentado.

1.2. OBJETIVOS.

1.2.1. Objetivos Gerais.

O presente trabalho tem como objetivo apresentar, de forma detalhada o processo de criptografia de um cartão RFID desde a montagem da placa e do módulo até a parte de programação do sistema, assim como alguns teste de caso com cartões de outros lugares.

1.2.2. Objetivos específicos.

Entre os objetivos que focam na formação do aluno no referido projeto de conclusão de curso, destacam-se as técnicas de desenvolvimento em outra linguagem, os métodos que serão utilizados para formar o conhecimento do aluno em relação ao uso das diferentes ferramentas existentes em segurança, trabalhar com equipamentos computacionais, avaliar e testar a estrutura e desempenho do dispositivo, utilizando recursos de lógica, para demonstrar como funciona e se contém falhas na segurança.

1.3. HIPÓTESE.

Como esses cartões tem uma certa distancia para que eles possam operar seria possível supostamente interceptar o sinal, capturar e alterar o seu conteúdo, assim tal interceptador podia usar os dados de forma ilegal. Porem este não é o único problema relacionado a essa tecnologia, com um cartão em cartão em mãos seria possível quebrar sua chave de segurança com as ferramentas corretas e usar as informações para o próprio benefício, e ainda no caso de um cartão de acesso, entrar em locais proibidos.

1.4. JUSTIFICATIVA.

Os conceitos apresentados sobre o RFID visam apresentar como existem vários métodos de implementação e varias áreas em que se pode aplicá-lo utilizando diferentes técnicas e métodos com o intuito de aprimorar todo o conhecimento adquirido no decorrer do curso.

1.5. PERSPECTIVA DE CONTRIBUIÇÃO.

De forma geral a aplicação deste trabalho contribui diretamente para o campo da criptografia por testar os métodos que são utilizados atualmente, além de melhorar a aplicação segura de pagamentos e registros por meio da comunicação RFID (Radio Frequence Identification), que tem uma capacidade de armazenamento limitada e que precisa de um método que seja simples e que ocupe pouca memória.

1.6. METODOLOGIA.

Para o desenvolvimento deste trabalho experimental, diversas fontes de pesquisas e testes que foram adquiridas com o decorrer do projeto serão utilizadas e aplicadas no Arduíno com a linguagem Java auxiliando na infraestrutura com dispositivo de leitura e gravação que será montado.

2. REVISÃO BIBLIOGRÁFICA.

2.1. TECNOLOGIA RFID.

A tecnologia de Radio Frequency Identification (RFID) é aplicada muitos campos nos dias atuais, já que o seu uso é extremamente dinâmico e as variações na sua arquitetura a tornam ainda mais versátil.

O RFID surge em meados de 1980 para solucionar problemas de rastreamento e controle de acesso como conta José Pinheiro no artigo “RFID-Identificação por rádio Frequência” (2004).

Já em 1999 a Universidade de Tecnologia de Massachusetts juntamente com outros centros de pesquisa começou a estudar a ideia de alterar a arquitetura para desenvolver novos projetos e aplicações nos campos de rastreamento e identificação de encomendas, e assim nasceu o EPC (Eletronic Product Code) que posteriormente passou a se chamar RFID.

O funcionamento é bem simples os chips (figura 1), denominados etiquetas (*tags*), recebem um sinal de rádio que são captadas pela antena depois passam por uma bobina e chegam até o chip onde ele vai processar os dados e enviar de volta os dados que foram programados dentro dele, sejam dados de localização, identificação ou pagamento.

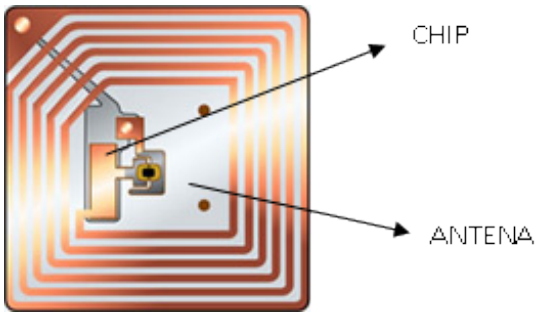

Figura 1: Etiqueta RFID.

É claro que existem diferenças entre os tipos de tags usadas atualmente e uma delas é a frequência (Tabela 1) de onda usada na comunicação, isso também

determina a distância média de leitura, e as divide para cada aplicação, por exemplo: as etiquetas de baixa frequência são mais utilizadas em cartões que normalmente precisam ser encostados nas leitoras para que possam ser analisados, como os que são usados pelas empresas de transporte de passageiros para efetuar o pagamento das passagens, seja em ônibus, metrôs ou outros do gênero. As tags de frequência alta são mais usadas em sistemas de encomendas como Correios ou transportadoras já que as caixas precisam passar por uma esteira e normalmente as etiquetas precisam estar dentro da caixa e assim ficar um pouco mais longe da leitora do que no caso da frequência baixa. As ultimas da tabela são as de frequência ultra alta e tem se tornado mais comuns a cada dia já que são utilizadas em sistemas para agilizar os pedágios nas estradas como o SemParar porem eles transmitem tanto com 915MHz como com 5.8GHz conforme dito no artigo de explicação para os clientes que está em seu web site.

Frequências RFID		
Frequência	***Banda***	***Distância Média***
Baixa	125 KHz	10cm
Alta	13.56 MHz	10cm a 1m
Ultra alta	915 MHz	12m

Tabela 1: Frequências RFID.

Por seu tempo de resposta extremamente baixo (inferior a 100 ms), tal tecnologia apresentou-se como uma grande solução para processos de produção que necessitam de uma grande velocidade já que outra vantagem é que não há necessidade de um contato visual para realizar a leitura, assim a etiqueta (*tag*) pode ficar dentro do produto, sem interferir no design ou poluir visualmente a embalagem como aponta o artigo “RFID - Vantagens e Desvantagens da Tecnologia” (2014).

É claro que tal tecnologia tem possibilidades ainda pouco exploradas como por exemplo a substituição de documentos e viabilizar métodos de pagamento mais

seguros, e isso seria possível graças a um chip colocado subcutâneo e faria a função dos vários métodos de identificação difundidos atualmente, como é mostrado na figura 2, podendo tomar o lugar do passaporte, RG, CPF e até mesmo de cartões de crédito e débito entre outras funções como abrir portas e garagens apenas pelo fato de estar portando um chip com a capacidade de armazenar dados que podem ser lidos de forma remota. Já que a sua arquitetura é bem simples como mostra a figura 3.

Figura 2: Chip Subcutâneo.

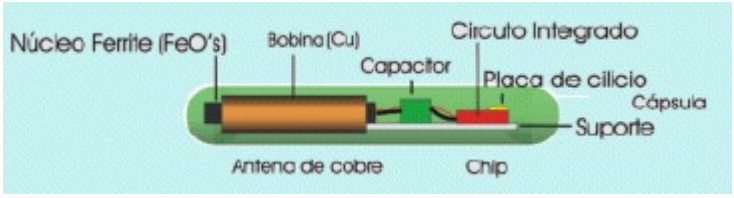

Figura 3: Chip Subcutâneo Desenho.

2.2. CARTÃO MFRC522 (MIFARE).

O cartão MIFARE Clássico é de meados de 1995, e foi criado pela NXP Semiconductors (Philips Semiconductors) atualmente já foram vendidos mais de 6 bilhões de cartões para mais de 40 tipos diferentes de aplicações.

Existe uma regulamentação de padrões que devem ser adotados por tal tecnologia e ela é mantida pela ISO/IEC 14443 que é dividida em 4 partes:

14443-1 - Características Físicas.

14443-2 - Sinal RF & Interface de Força.

14443-3 - Inicialização & Anti Colisão.

14443-4 - Protocolos de Transmissão.

Todas as informações contidas neste capítulo foram extraídas do ISO/IEC 14443 de 2016.

2.2.1. Estrutura e arquitetura.

Nos cartões é inserida uma memória EEFROM de 1kb, uma interface RF, que é responsável por fazer a comunicação entre o leitor e a unidade de controle digital, responsável por fazer o registo dos dados, a energia que alimenta o sistema e os dados são transmitidos (recebidos e emitidos) pela antena em forma de bobina que é ligada ao chip como mostra a figura 4.

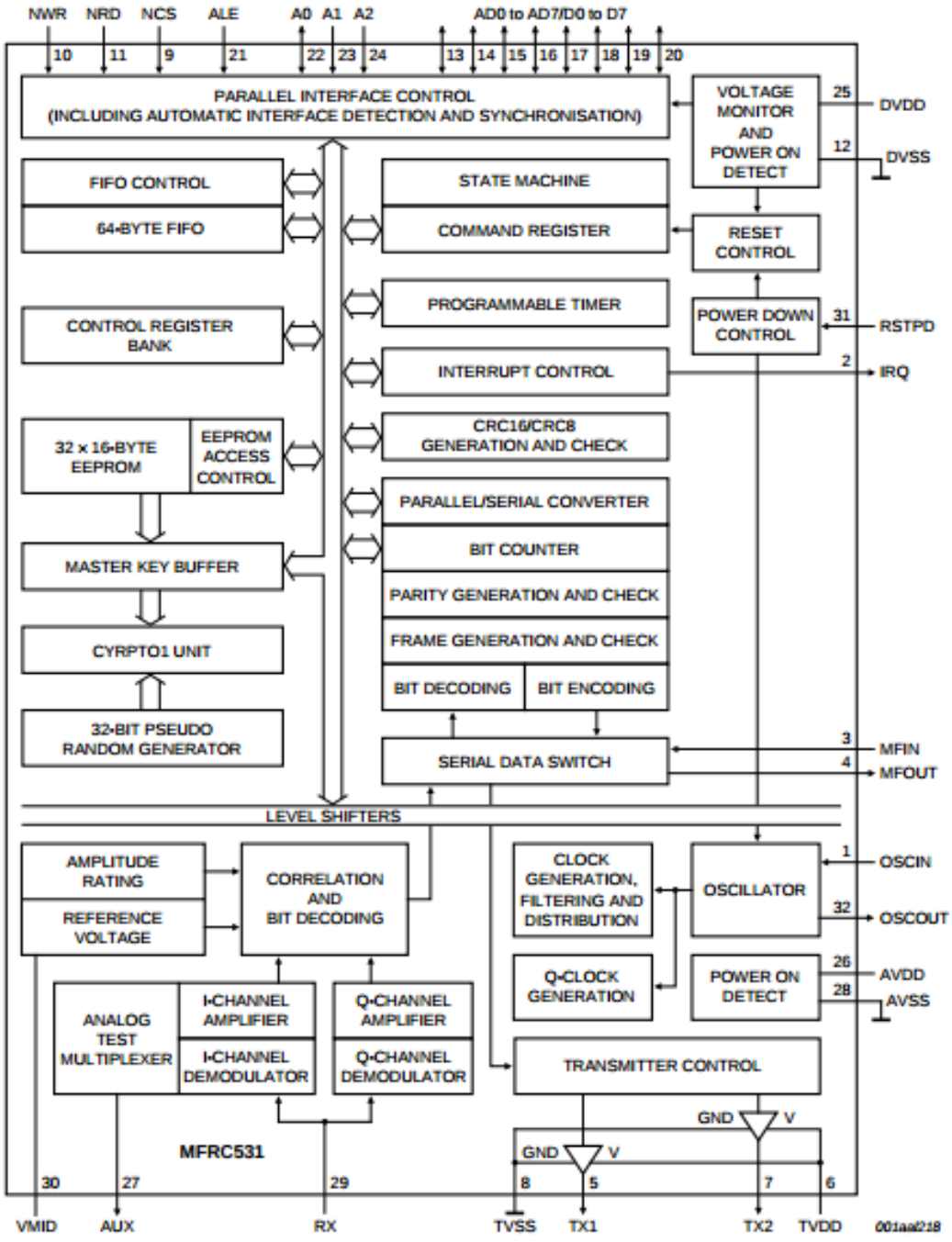

Figura 4: Diagrama de Blocos.

2.2.2. Arquitetura de armazenamento de dados.

De forma geral existe 5 blocos de armazenamento com denominações diferentes, de acordo com a ISO 14443 *Identification Cards – Contactless Integrated Circuit(s) cards,* de 2007 :

O “Sector Trailer”, é como é chamado o ultimo bloco de cada setor que contem as chaves A e B, ambas de 6 bytes, com mais 4 bytes para configuração das condições de acesso daquele setor, e é quem valida, então caso uma informação seja inserida errada ou incompleta o setor inteiro fica inacessível.

O bloco do fabricante é composto pelos 4 primeiros bytes do cartão e não é possível gravar neles graças a uma proteção contra escrita que é colocado por segurança, já que neles é gravada as informações de fábrica, tais como, numero de serie e outras identificações que são particulares de cada cartão.

Blocos de dados são formados de 3 blocos de 16 bytes (exceto o setor 0) e podem ser divididos entre blocos de leitura, gravação e blocos de valor.

Os blocos de valor são um tipo especial de Blocos de dados, já que neles é possível operações semelhantes a uma “carteira eletrônica” como: escrita, leitura, acréscimo, decremento, restauração e transferência.

E por último o bloco de leitura/escrita que são os mais comuns tem 16 bytes de dados para serem gravados e lidos apenas.

Dentro de tais blocos são possíveis 6 operações: Leitura, Escrita, Incremento, Decremento, Transferência e Restauração.

2.2.2.1. Arquitetura de Acesso aos Blocos.

As operações nos blocos são autorizadas pelo cartão se suas condições de acesso estiverem compatíveis. As Condições são representadas por 3 bits que fica invertidos ou não-invertidos dependendo de cada “Sector Trailer”.

2.2.3. Protocolos de Comunicação.

O protocolo ISO-14443-A-3 é o padrão que permite que sejam fabricados no mundo todos vários componentes por diferentes pessoas e ainda sim os sistemas ainda consiga comunicar tranquilamente.

2.2.3.1. Requisição padrão.

Os cartões, quando não estão no campo de alcance de nenhum leitor, iniciam no estado de Power-on-line. Para que ele seja ativado o leitor deve emitir continuamente o sinal de requisição (WUPA - Weak Up Command - type A) Caso haja qualquer cartão no campo magnético no leitor, ele deve responder com o sinal de resposta ATQA (Answer to Request - Type A), assim o dispositivo de leitura sabe que tem ao menos um cartão no seu alcance, e a partir daí começa o processo de loop de anti-colisão.

2.2.3.2. Loop de anti-colisão.

Um caso que pode acontecer é de ter mais de dois chips no campo magnético do leitor e é exatamente por isso que existe o loop anti-colisão, e seu identificador é obtido através do comando AC (Anti-Collision). Neste caso os cartões são distinguidos um do outro e o que não foi selecionado volta para o estado de IDLE.

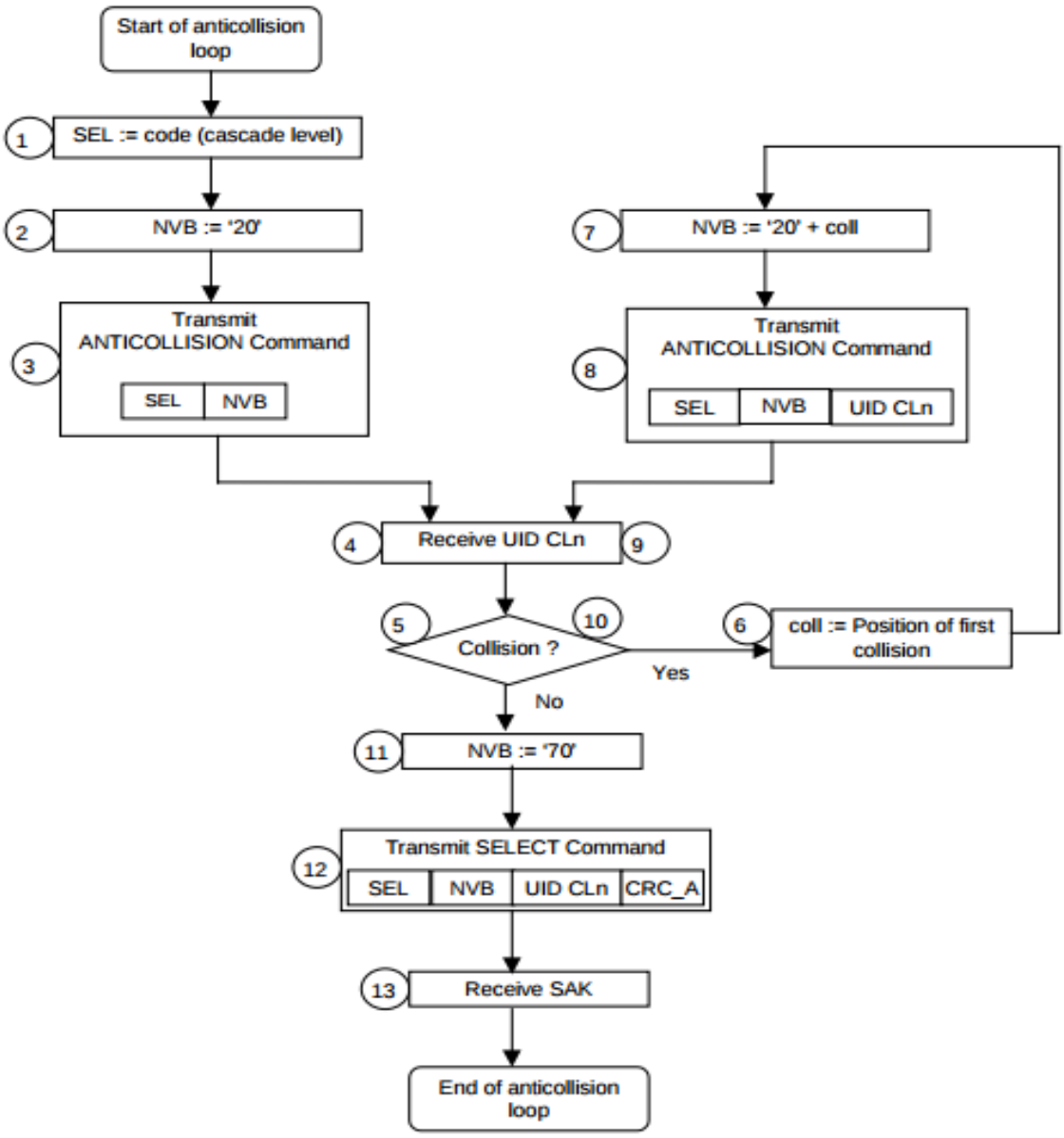

Figura 5: Diagrama de Anti Colisão.

O leitor envia o sinal SEL (Select Card) Ao cartão escolhido na fase anterior, para que a confirmação e autenticação aconteça e depois as operações de memória, o cartão retorna o Sinal SAK (Select Acknowledge). O sinal SAK e ATQ determinam o tipo de cartão, assim como seu fabricante, como é mostrado no diagrama da figura 5.

2.2.3.3. FIFO Buffer.

Arquitetura FIFO é aquele que executa os procedimentos em fila (First In First Out).Um buffer FIFO de 8 por 64 bits é usado no MFRC531 para atuar como um conversor paralelo a paralelo. Ele armazena os fluxos de dados de entrada e saída entre o microprocessador e os circuitos internos do cartão. Isso permite gerenciar fluxos de dados até 64 bytes de comprimento sem precisar levar em consideração as restrições de tempo.

O barramento de dados de entrada e saída do buffer FIFO é conectado ao registrador de dados, escrevendo para este registro um byte no buffer FIFO e incrementando a gravação do buffer Ponteiro. A leitura deste registro mostra o conteúdo do FIFO armazenado no Ponteiro de leitura dele e incrementa o ponteiro de leitura do buffer FIFO. A distância entre o escrever e ler o ponteiro pode ser obtido lendo o registro de comprimento do FIFO.

2.2.4. Chaves Crypto1.

A segurança MIFARE requer chaves criptográficas específicas para criptografar o fluxo de dados na comunicação na interface sem contato. Essas chaves são chamadas chaves Crypto1.

Elas são armazenadas na EEPROM e escritas em um formato específico. Cada byte-chave deve ser dividido em quatro bits inferiores a k0 e k3 e os quatro bits maiores k4 e k7. Cada nibble é armazenado duas vezes em um byte e um dos dois nibbles é invertido bit-wise, este formato é uma condição prévia para a execução bem-sucedida do carregamento da chave E2, como mostrado na figura 6.

Master key byte	0 (LSB)		1			5 (MSB)	
Master key bits	$\overline{k7}$ $\overline{k6}$ $\overline{k5}$ $\overline{k4}$ k7 k6 k5 k4	$\overline{k3}$ $\overline{k2}$ $\overline{k1}$ $\overline{k0}$ k3 k2 k1 k0	$\overline{k7}$ $\overline{k6}$ $\overline{k5}$ $\overline{k4}$ k7 k6 k5 k4	$\overline{k3}$ $\overline{k2}$ $\overline{k1}$ $\overline{k0}$ k3 k2 k1 k0		$\overline{k7}$ $\overline{k6}$ $\overline{k5}$ $\overline{k4}$ k7 k6 k5 k4	$\overline{k3}$ $\overline{k2}$ $\overline{k1}$ $\overline{k0}$ k3 k2 k1 k0
EEPROM byte address	n	n + 1	n + 2	n + 3		n + 10	n + 11
Example	5Ah	F0h	5Ah	E1h		5Ah	A5h

001aak640

Figura 6: Formato de armazenamento das chaves.

2.3. ARDUINO.

O arduino é um hardware OpenSource focado para o aprendizado e desenvolvimento de protótipos criado em 2005 por 5 pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis.

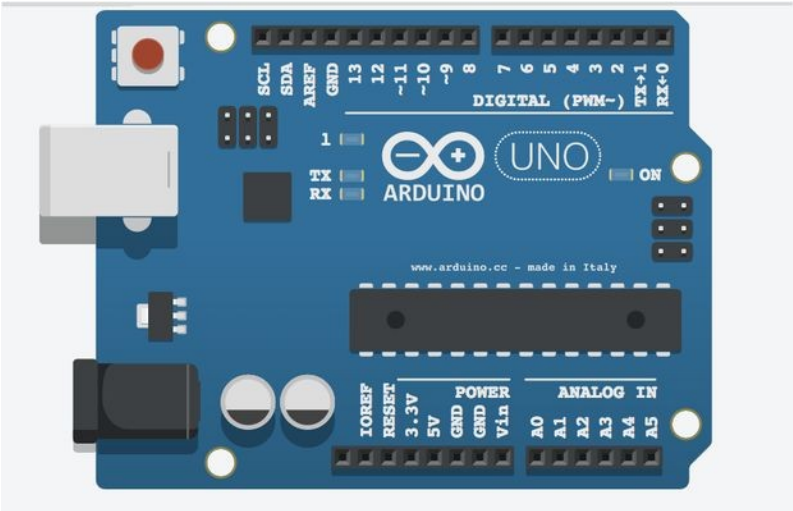

Figura 7: Arduino UNO R3

2.3.1. Arquitetura.

O arduino é composto por um micro-controlador Atmega e interfaces de simples conexão por cabos que são plugados as portas, a divisão das portas é feita entre: portas de força, Digitais e portas Analógicas utilizando uma linguagem baseada em C/C++, sendo conectado ao computador somente com o cabo USB, e após a programação da placa e o código ser armazenado o Hardware pode ser utilizado com uma simples fonte que seja compatível com a interface de alimentação variando entre 7 e 30 volts de energia.

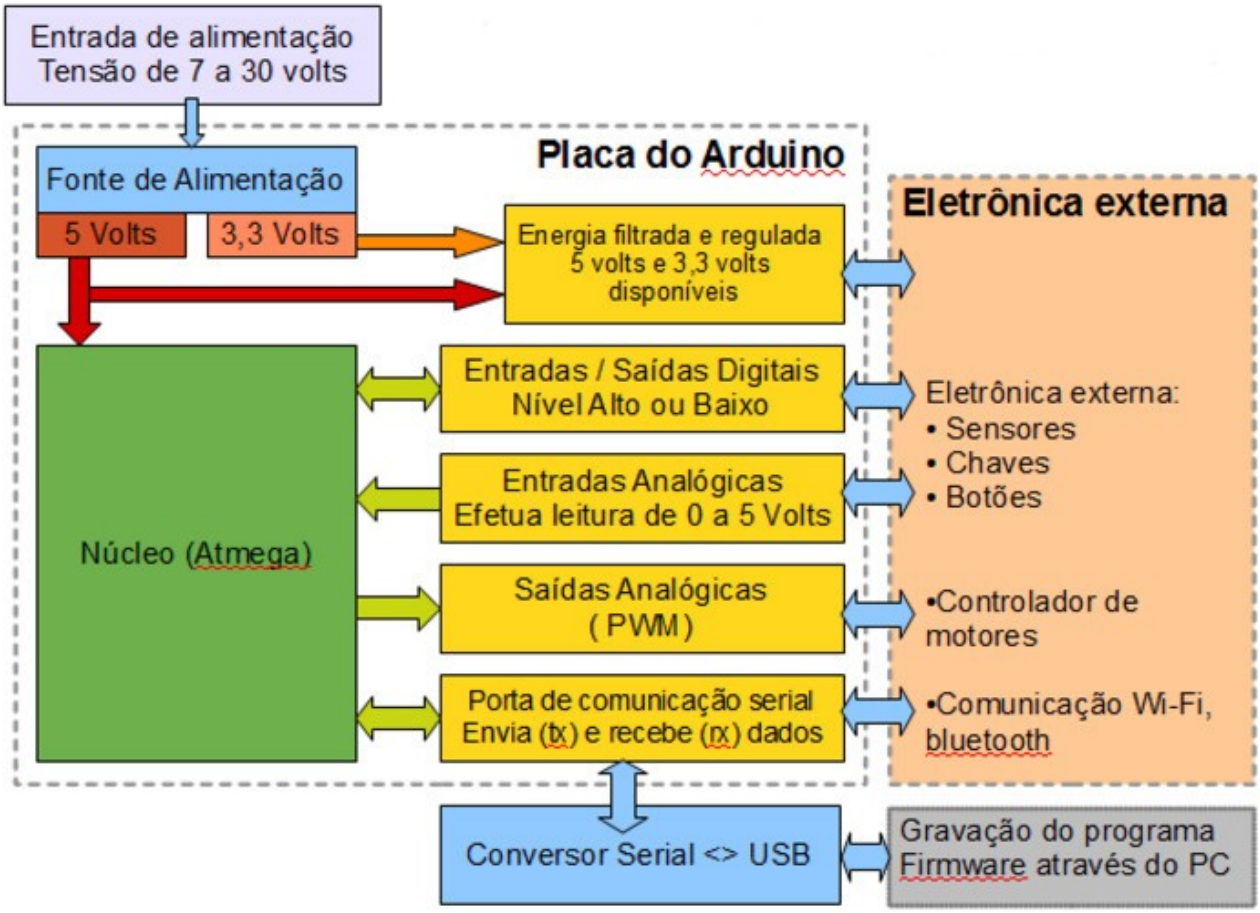

Figura 8: Arquitetura da placa Arduino.

2.3.2. Entrada e saída.

O arduino Uno R3 (Figura 7 tem 14 pinos digitais que podem servir com entrada ou saída dependo do uso feito pelos comandos pinMode(), digitalWrite() e digitalRead(). Cada pino opera a 5v e pode fornecer no máximo 40mA.

Além deles existem alguns com funções especiais pré determinadas como: pino 0 (RX) e pino 1 (TX) utilizados para receber e transmitir dados seriais respectivamente, são conectados ao pinos correspondentes no USB-TTL serial chip ATMEGA8U2.

2.3.3. Fonte de Alimentação.

O bloco de alimentação é responsável por receber a energia externa que precisa ser de no mínimo 7v e menos do que 35v com uma corrente mínima de 300mA, a fonte filtra e depois regula a tensão de entrada para duas saídas, a de 5v e a de 3,3V. O arduino não necessita de uma conexão direta a tomada quando está com o cabo USB conectado pois ele já fornece uma voltagem de 12v que é o suficiente o sistema.

2.3.4. Processamento e micro-controlador.

O núcleo de processamento de uma placa de arduino é a CPU que é praticamente um computador completo com memória RAM e memória ROM, uma unidade de processamento aritmética e interfaces de entrada e saída.

Os inventores do arduino optaram por um processador ATMega e são usados diferentes modelos dependendo de qual arduino estão inseridos, os mais comuns são ATMega8, ATMega162 e ATMega328p e entre eles as diferenças estão na memória e na configuração dos módulos.

2.3.5. Programação.

O Arduino é programado via cabo USB que conectado ao computador pode transmitir os códigos contendo o programa que será executado na placa. A verificação desse código (para verificar se não contém nenhum erro de programação) é feito pela sua IDE, que lê o programa escrito pelo usuário e envia para o hardware, onde é armazenado e compilado.

2.3.5.1. IDE.

O ambiente de desenvolvimento do arduino é bem simples e está disponível para Windows, Linux e Mac, e como o foco de todo o projeto é ser OpenSouce seu código está disponível no GitHub onde pode ser alterado e cada pessoa pode construir a sua versão da ferramenta.

2.3.5.1 Código.

Os programas escritos para arduino tem sempre que seguir uma base comum que é ter um método chamado Setup e outro chamado Loop, como mostra a figura 9.

```
1  //Programa : Pisca Led Arduino
2
3
4  void setup()
5  {
6    //Define a porta do led como saida
7    pinMode(13, OUTPUT);
8  }
9
10 void loop()
11 {
12   //Acende o led
13   digitalWrite(13, HIGH);
14
15   //Aguarda o intervalo especificado
16   delay(1000);
17
18   //Apaga o led
19   digitalWrite(13, LOW);
20
21   //Aguarda o intervalo especificado
22   delay(1000);
23 }
```

Figura 9: Programa Pisca LED.

Neste caso o pino 13 é configurado como saída, e no método *Loop* o primeiro comando *digitalWrite* acende o LED então é dado um *delay,* que é um comando para esperar um determinado tempo, e então a luz se apaga, mais um tempo é esperado e começa tudo novamente, muito semelhante ao que acontece no “Enquanto” na programação.

2.4. CRIPTOGRAFIA.

A palavra criptografia deriva do grego “Kryptós” que significa oculto e “gráphein” que significa escrita. Pode considerar-se que é uma técnica amplamente difundida para codificar uma informação de uma forma que somente o emissor e o receptor designado consiga ler o conteúdo da mensagem.

A evolução dos métodos criptográficos acompanham a evolução humana uma vez que ao passar do tempo ela tem mudado e se tornando cada vez mais segura.

A maioria dos métodos de criptografia se baseia em sistema de substituição simples, que acontece quando uma letra é substituída por outra. Um dos primeiros métodos documentados é a Cítala, um método grego que consiste em uma tira de couro com diversas letras escritas que eram enroladas ao redor de um peça de forma retangular ou hexagonal e só que tinha o tamanho correto da peça podia saber o significado da mensagem. E muitos anos depois quando a criptografia foi utilizado para transmitir mensagem na primeira e segunda guerra mundial onde a famosa e até então indecifrável máquina Enigma foi utilizada pelos alemães até ser “quebrada” por Alan Turing, Matemático genial e considerado por muitos o pai da computação.

2.4.1. Cifra de Viginére.

A cifra de Viginère é uma técnica de substituição polialfabética que utiliza uma forma de Matriz de Leitura com vários alfabetos para que em um mesmo texto tenha várias representações da mesma letra, dificultando assim a quebra da cifra, que só pode ser revelada se você tiver a chave com a qual a mudança foi feita.

Ela foi criada em 1553 por Blaise de Viginére, e só foi quebrada em 1853 pelo criptógrafo alemão Friedrich Kasiski e durante esse período ficou conhecida como “le chiffre indéciffravel” (em francês cujo tradução é “a cifra indecifrável”).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figura 10: Matriz de Viginére.

Supondo que a frase seja proposta: “Infantaria um posicionada” e a chave de Criptografia seja: ”Exercito”, o resultado seria: “MKJAPBOVFR GU JCAWGFTECLT”.

Porem como podemos ver as palavras ainda ficam separadas e assim facilita para ser decifrada, para que ela possa ser melhorada é necessário que o espaço seja inserido na tabela assim como os números de 0 a 9.

Se for colocado na matriz todos os caracteres da tabela ASCII e se sua sequencia for embaralhada é possível aumentar ainda mais o “força” do método e tornar ainda complicada a substituição.

3. IMPLEMENTAÇÃO.

Foi desenvolvido uma placa para ser encaixada em cima do arduino, modelo UNO R3, tal equipamento é chamado de “shield” . A função do equipamento é ler e gravar informações fornecidas pelo computador.

3.1. FERRAMENTAS.

3.1.1. Netbeans.

O Netbeans foi escolhido como IDE para desenvolvimento do código por fornecer ao usuário uma facilidade excelente quando foi necessário baixar e instalar os pacotes de conexão do Arduino.

O software é distribuído atualmente (2017) de forma gratuita pela Oracle e é a plataforma oficial de desenvolvimento de projetos e códigos Java. Encontra-se versões para Windows, Mac OS X e Linux em 32 e 64 bits no próprio site da empresa.

3.1.2. RXTX.

A RXTX é a API de comunicação serial que permite a transmissão do código para arduino por uma porta USB, ela foi baseada na biblioteca Javacomm que era distribuída anteriormente pela empresa Sun Microsystems. É possível encontra-la em alguns site na Internet e eu código está disponível no GitHUB.

3.1.3. Arduino.

Neste trabalho foi utilizado um Arduino UNO R3, fornecido pela empresa Robocore, e de maneira prática ele possui as mesmas funções do modelo original italiano porém essa consegu e alimentar mais equipamentos em sua porta de 5V. Ela pode ser comprada no site por aproximadamente R$ 85,00.

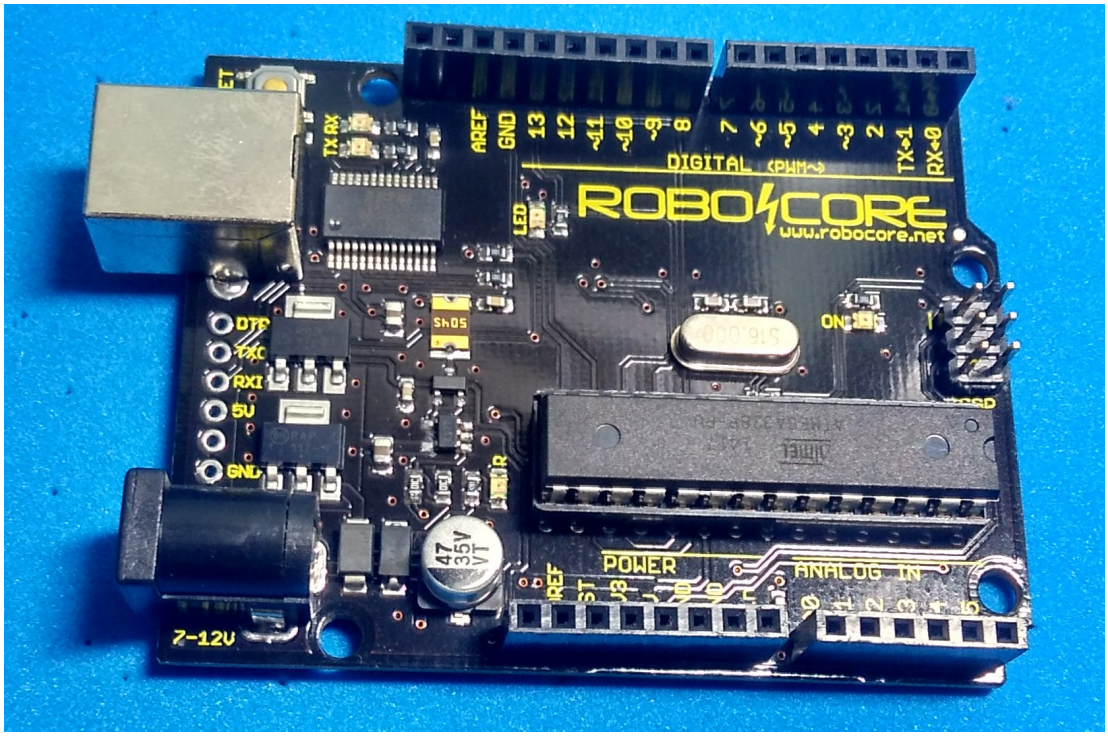

Figura 11: BlackBoard

3.1.4. Módulo de Leitura.

O módulo de leitura utilizado foi o RFID-RC522 que possui 8 pinos sendo eles respectivamente:(da esquerda para a direita) SDA, SCK MOSI,MISO, IRQ, GND, RST, e o pino de 3,3V.

Figura 12: Módulo RFID

3.1.5. Protoshield.

O protoshield é uma placa pré montada que vai posicionada em cima do arduino encaixando nas portas de comunicação e diminuindo a necessidade de fios e cabos de contato alem de ser uma solução mais prática e rápida para a produção de um protótipo. Existem muitos modelos e preços diferentes porém a utilizada neste projeto foi encontrada por R$20,00.

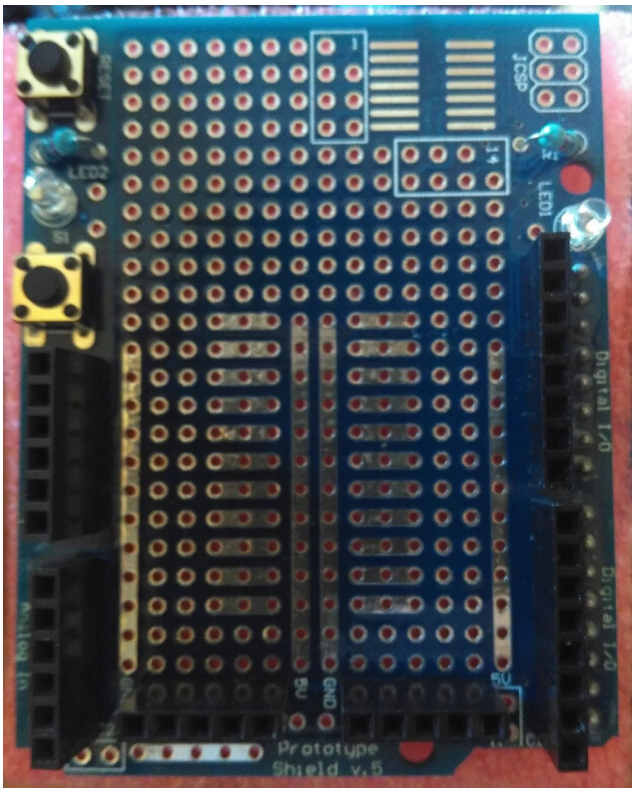

Figura 13: Protoshield.

3.1.6. Cartões RFID.

Para que fosse possível ser feito a gravação e a leitura das informações foi utilizado cartões MIFARE522-1K em branco. O custo estimado de cada unidade é de R$1,00.

3.2. CONSTRUÇÃO.

Além de todas as peças já listadas foram utilizadas 4 luzes de LED sendo duas vermelhas e duas verdes para mostrar o estado dos operadores e 11 pedaços de fios coloridos que foram usados nas conexões.

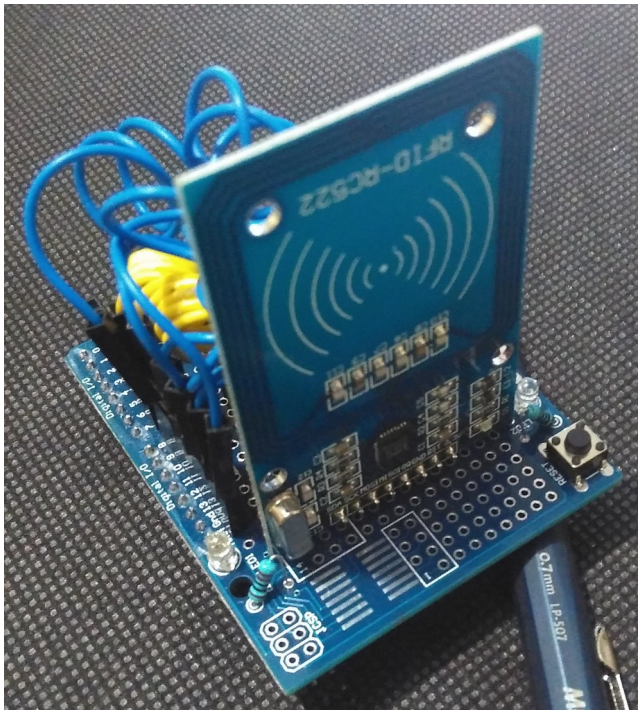

Figura 14: Protótipo atrás.

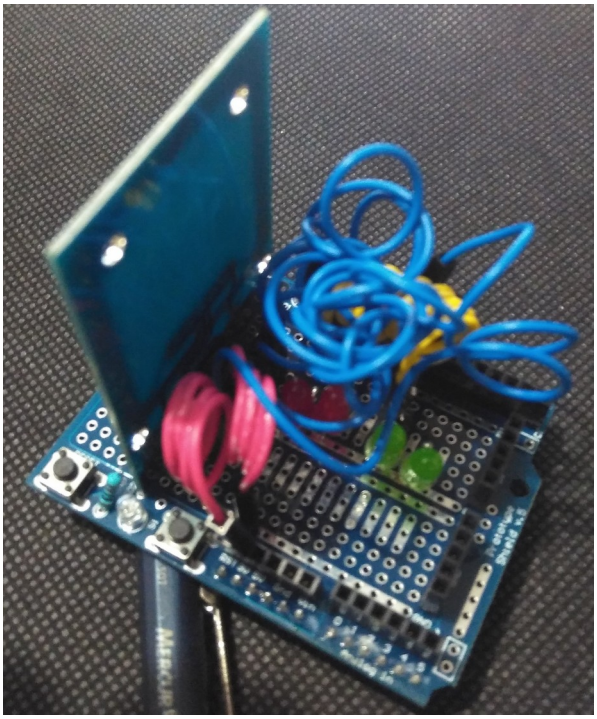

Figura 15: Protótipo frente.

3.2.1. Código.

O código desenvolvido para a comunicação foi feito em C e Java, porém o método de criptografia foi desenvolvido exclusivamente em Java separado, assim como a interface com o usuário e classe de teste de conexão.

Todo o código do projeto e de todos os arquivos estão disponíveis no Anexo 1 - “Programa RFID”.

3.2.2. Comunicação.

Para fazer de forma separada foram realizadas 2 código distintos, um método de criptografia próprio baseado na cifra de viginére feito em Java e que recebe 3 valores como parâmetros, a mensagem, o tipo do precedimento se vai ser criptografia(1) ou decriptografia(2) e a senha escolhida para a mensagem, e um software de comunicação gravação e leitura de tag's RFID baseada no código em C++ disponível no site

4. CONCLUSÃO

Sendo assim é possível analisar que a tecnologia RFID possui um campo de aplicação imensamente vasto, que possui um preço de implementação relativamente baixo contando que dentro de sua arquitetura nativa mais as aplicações que são possíveis com o espaço de armazenamento pequeno que podem aumentar a segurança dos dados, os tornando menos suscetíveis a alterações não autorizadas e até mesmo o corrompimento dos dados. Quando a implantação é feita com um protótipo construído a compreensão fica muito mais visível de modo que as particularidades da implementação e construção tanto quando ao que se refere ao arduino, ficam muito mais claras, já que é possível lidar com uma camada mais “baixa” da tecnologia.

Após o término da finalização do estágio de construção relatado neste trabalho, ficou claro o quanto as implementações realizadas podem ser melhoradas é claro, já que o conceito central de toda tecnologia é que ele pode ser melhorada sempre, e é certo que seria possível uma matriz tridimensional reforçando ainda mais as frases que estão criptografadas.

Concluindo como dito anteriormente a prataforma e métodos apresentados apesar de serem simples e de baixo custo, quando comparada a outras sistemáticas já disponíveis prontas, podem ser muito escaláveis e passíveis de melhoria sempre e podem sim apresentar soluções profissionais e proporcionar uma “camada” a mais de segurança as pessoas que estiverem armazenando dados com a tecnologia RFID.

5. REFERÊNCIAS

STALLINGS, W. **Criptografia e Segurança de Redes.** Editora Prentice-Hall , 2007.

OXER, J e BLEMINGS, H. **Pratical Arduino Cool Projects for Open Source Hardware.** Editora Technology in Action, 2009.

MARGOLIS, M. **Arduino Cookbook.** Editora O'Reilly, 2011.

THOMSEN, A. **Como gravar dados no cartão RFID.** Disponível em:http://blog.filipeflop.com/wireless/como-gravar-dados-no-cartao-rfid.html.

PINHEIRO, J. **RFID Identificação por Rádio Frequencia.** Disponível em:http://www.projetoderedes.com.br/artigos/artigo_identificacao_por_radiofrequencia.php.

RFID - Vantagens e desvantagens da tecnologia (2014). Disponível em:http://www.afixgraf.com.br/rfid-vantagens-desvantagens/

Sobre a tecnologia. Disponível em:.

Informações gerais sobre a plataforma arduíno. Disponível em: .

MFRC531 Standard ISO/IEC 14443 A/B reader solution. (Documentação sobre a ISO do RFID) Disponível em: . Jun 2015

International Standart ISO/IEC 14443-3 first edition. Disponível em:

International Standart ISO/IEC 14443-4 first edition. Disponível em: .

FILHO, Daniel O. Basconcello. **Aula 2 – O Hardware do Arduino.** Disponível em: http://www.robotizando.com.br/curso_arduino_hardware_pg1.php.

THOMSEN, Adilson. **O que é Arduino.** Disponível em: https://www.filipeflop.com/blog/o-que-e-arduino/.

6. ANEXOS

ANEXO I – Código de Criptografia.

ANEXO II – Código de Comunicação e Gravação.

ANEXO III – Configuração de ligação do ProtoShield.

ANEXO I – CÓDIGO DE CRIPTOGRAFIA.

```
public class Criptografia2d {
    public String cripto (String mensagem, int metodo, String chave){
        char[] encriptado = null;
        char[] mensagemChar = mensagem.toCharArray();
        char[] decriptado = null;
        char[] chaveChar = chave.toCharArray();
        int[]  Colunas;
        int tamanhoMensagem = 0, contColum = 0, tamanhoChave,contCom =0;
        String resultado = "" ;
        tamanhoMensagem= mensagemChar.length + chaveChar.length;
        tamanhoChave = chaveChar.length;
        encriptado = new char[tamanhoMensagem];
        decriptado = new char[tamanhoMensagem];
        int coluna=0, linha =0,xcolum = 0, xlinha =0;
        int nExiste = 0 ;
        int xc = 0 , y = 0 ;
     //variáveis de checagem de chaves
        int w=0;
        boolean valeChave = true;


          // O método de Viginére é um metodo de chave privada aplicada a tecnologia de criptografia de substituição, e quanto maior a chave mais forte é a criptografia
        char viginere[][] = {
```

{'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' '},

{'b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a'},

{'c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b'},

{'d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c'},

{'e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d'},

{'f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e'},

{'g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f'},

{'h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g'},

{'i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h'},

{'j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i'},

{'k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j'},

{'l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k'},

{'m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l'},

{'n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m'},

{'o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n'},

{'p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o'},

{'q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p'},

{'r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q'},

{'s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r'},

{'t','u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s'},

{'u','v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t'},

{'v','w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u'},

{'w','x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v'},

{'x','y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w'},

{'y','z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x'},

{'z','1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y'},

{'1','2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'},

{'2','3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1'},

{'3','4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2'},

{'4','5','6','7','8','9','0',' ','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3'},

```
{'5','6','7','8','9','0','
','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4'},
{'6','7','8','9','0','
','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5'},
{'7','8','9','0','
','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6'},
{'8','9','0','
','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7'},
{'9','0','
','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8'}
,
{'0','
','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8',
'9'},
{'
','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','1','2','3','4','5','6','7','8',
'9','0'}


        };
    if(metodo == 1){ //Criptografar é o método 1      // a variavel método é a que determina o processo a ser feito, criptografar ou decriptografar.


    for(int i= 0; i < tamanhoMensagem; i++){


     if (xc < tamanhoChave){
                // dentro desse if a chave é criptografada e adicionada no inicio da mensagem.
        while(chaveChar[xc] != viginere[0][xcolum]){
           xcolum++;
```

```
            }
            while(chaveChar[xc] != viginere[xlinha][0]){
                xlinha++;
            }
            encriptado[i]= viginere[xlinha][xcolum];
            xcolum = 0;
            xlinha = 0;
            xc++;
          }else{  // depois de adicionar a chave
        if (contColum == chaveChar.length){
            contColum = 0;

        }  //encontra posição da chave
        while(chaveChar[contColum] != viginere[0][coluna]){
            coluna++;
        }

        while(mensagemChar[y] != viginere[linha][0]){ // erro está aki o i tem 3 posições e a mensagem so tem 2
            linha ++;
              if (linha >= 29){ // verifica se a letra nao existe
               nExiste = 1;
                break;
            }
        }
      if(nExiste == 0){
            encriptado[i] = viginere[linha][coluna];
```

```
  }else if(nExiste == 1) {
        encriptado[i] = mensagemChar[y];
    }
  coluna = 0 ;
  linha = 0 ;
  contColum++;
  y++;
  nExiste = 0;
  }

}
 }else if (metodo == 2){
    while(w < tamanhoChave){
        if (mensagemChar[w] != chaveChar[w]){
            valeChave = false;
        }else{
            valeChave = true;
        }
        w++;
    }
    if (valeChave == false){
    return ""
    }else{
    y = chaveChar.length;
    tamanhoMensagem = mensagemChar.length - chaveChar.length;
     for(int i = 0; i < tamanhoMensagem; i++){
```

```
    if (contColum == chaveChar.length){
        contColum = 0;
    }  //encontra posição da chave

    while(chaveChar[contColum] != viginere[coluna][0]){
        coluna++;
    }
    while(mensagemChar[y] != viginere[coluna][linha]){
        linha ++;
          if (linha >= 29){ // verifica se a letra nao existe
           nExiste = 1;
            break;
          }
    }
   if(nExiste == 0){
        decriptado[i] = viginere[linha][0];
  }else if(nExiste == 1) {
        decriptado[i] = mensagemChar[y];
    }
  coluna = 0 ;
  linha = 0 ;
  contColum++;
  y++;
  }
}
  }
 //verifica qual vai ser passado por resultado
```

```
        if (metodo == 1 ){
            resultado = new String (encriptado);
        }else if (metodo == 2){
            resultado = new String (decriptado);
        }
      return resultado;
   }



}

// para chamar o método em outra classe é so fazer
Criptografia2d cptd2d = new Criptografia2d();
cptd2d.cripto(mensagem, 1 , senha); // onde mensagem e senha são strings e o numero
// pode ser 1 ou 2
```

ANEXO II – CÓDIGO DE COMUNICAÇÃO E GRAVAÇÃO.

```
/*
Nome: Leitor e gravador de cartoes RFID
Data: 20/12/16
Autor: Kelvin Ferri Brancalhao
*/
#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal.h>
#define SS_PIN 10
#define RST_PIN 9
#define LEDVermelho 8
#define LEDVerde 7
MFRC522 mfrc522(SS_PIN, RST_PIN);

#define pino_botao_le A2
#define pino_botao_gr A3

MFRC522::MIFARE_Key key;

void setup()
{
  pinMode(pino_botao_le, INPUT);
  pinMode(pino_botao_gr, INPUT);
  pinMode(LEDVermelho, OUTPUT);
```

```
  pinMode(LEDVerde, OUTPUT);
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();

  mensageminicial();


  for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
}

void loop()
{

  int modo_le = digitalRead(pino_botao_le);
  if (modo_le != 0)
  {
    Serial.println("Modo leitura selecionado");
    while (digitalRead(pino_botao_le) == 1) {}
    delay(3000);
    modo_leitura();
  }

  int modo_gr = digitalRead(pino_botao_gr);
  if (modo_gr != 0)
  {
    Serial.println("Modo gravacao selecionado");
```

```
    while (digitalRead(pino_botao_gr) == 1) {}
    delay(3000);
    modo_gravacao();
  }
}
void mensageminicial()
{
  Serial.println("\nSelecione o modo leitura ou gravacao...");
  Serial.println();
}

void mensagem_inicial_cartao()
{
  Serial.println("Aproxime o seu cartao do leitor...");
}

void modo_leitura()
{
  mensagem_inicial_cartao();

  while ( ! mfrc522.PICC_IsNewCardPresent())
  {
    delay(100);
  }
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
```

```
}

Serial.print("UID da tag : ");
String conteudo = "";
byte letra;
for (byte i = 0; i < mfrc522.uid.size; i++)
{
  Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
  Serial.print(mfrc522.uid.uidByte[i], HEX);
  conteudo.concat(String(mfrc522.uid.uidByte[i]<0x10 ? " 0" : " "));
  conteudo.concat(String(mfrc522.uid.uidByte[i], HEX));
}
Serial.println();


byte sector         = 1;
byte blockAddr      = 4;
byte trailerBlock   = 7;
byte status;
byte buffer[18];
byte size = sizeof(buffer);


status=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;
}
status = mfrc522.MIFARE_Read(blockAddr, buffer, &size);
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("MIFARE_Read() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
}

for (byte i = 1; i < 16; i++)
{
  Serial.print(char(buffer[i]));
}
Serial.println();


sector         = 0;
blockAddr      = 1;
trailerBlock   = 3;


status=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;
  }
  status = mfrc522.MIFARE_Read(blockAddr, buffer, &size);
  if (status != MFRC522::STATUS_OK)
  {
    Serial.print(F("MIFARE_Read() failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
  }

  lcd.setCursor(0, 1);
  for (byte i = 0; i < 16; i++)
  {
    Serial.print(char(buffer[i]));
  }
  Serial.println();

  mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();
  delay(3000);
  mensageminicial();
}

void modo_gravacao()
{
  mensagem_inicial_cartao();
  while ( ! mfrc522.PICC_IsNewCardPresent()) {
    delay(100);
```

```
}
if ( ! mfrc522.PICC_ReadCardSerial())    return;

Serial.print(F("UID do Cartao: "));
for (byte i = 0; i < mfrc522.uid.size; i++)
{
  Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
  Serial.print(mfrc522.uid.uidByte[i], HEX);
}
Serial.print(F("\nTipo do PICC: "));
byte piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
Serial.println(mfrc522.PICC_GetTypeName(piccType));

byte buffer[34];
byte block;
byte status, len;

Serial.setTimeout(20000L) ;
Serial.println(F("Insira seus dados #"));
len = Serial.readBytesUntil('#', (char *) buffer, 30) ;
for (byte i = len; i < 30; i++) buffer[i] = ' ';

block = 1;

status=mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
                                block, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
```

```
  Serial.print(F("PCD_Authenticate() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}


status = mfrc522.MIFARE_Write(block, buffer, 16);
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("MIFARE_Write() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}

block = 2;

status=mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
                                block, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("PCD_Authenticate() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}


status = mfrc522.MIFARE_Write(block, &buffer[16], 16);
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("MIFARE_Write() failed: "));
```

```
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}

Serial.println(F("Insira seus dados: #"));
len = Serial.readBytesUntil('#', (char *) buffer, 20) ;
for (byte i = len; i < 20; i++) buffer[i] = ' ';

block = 4;

status=mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
                                block, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("PCD_Authenticate() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}


status = mfrc522.MIFARE_Write(block, buffer, 16);
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("MIFARE_Write() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}

block = 5;
```

```
status=mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
                                block, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("PCD_Authenticate() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}


status = mfrc522.MIFARE_Write(block, &buffer[16], 16);
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("MIFARE_Write() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  digitalWrite(LEDVermelho, HIGH);
  delay(500);

}
else
{
  Serial.println(F("Dados gravados com sucesso!"));
  digitalWrite(LEDVerde, HIGH);
  delay(500);

}

mfrc522.PICC_HaltA();
```

```
  mfrc522.PCD_StopCrypto1();
  delay(5000);
  mensageminicial();
}
```

ANEXO III–CONFIGURAÇÃO DE LIGAÇÃO DO PROTOSHIELD.

Esquema de conexão das peças com as portas do arduíno UNO R3, é necessário o uso de jumpers (Fios de comunicação).

PLACA DE COMUNICAÇÂO RFID.

RST – Porta Digital 9

MISO – Porta Digital 12

MOSI – Porta Digital 11

SCK – Porta Digital 13

SDA – Porta Digital 10

GND – Porta GND

3,3V – Porta 3,3V

BOTAO GRAVAR – Porta Analógica A3.

BOTAO LER – Porta Analógica A2.

LED VERMELHO – Porta Digital 8.

LED VERDE – Porta Digital 7.