



UNIVERSIDADE EDUARDO MONDLANE FACULDADE DE ENGENHARIA

**DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA
CURSO DE ENGENHARIA ELECTRÓNICA**

**Desenvolvimento de um Sistema de
Controle de Segurança com BD para
Registro de Usuários e Envio de
Notificação de Incidentes.**

Relatório do Projecto do Curso

Laércio Clementina Marcelino Macome

**Supervisor: Mestre Engº J.P. Branco
(UEM, Faculdade de Engenharia, Departamento de Engenharia Electrotécnica)**

Maputo, Novembro 2023



UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA
CURSO DE ENGENHARIA ELECTRÓNICA

**Desenvolvimento de um Sistema de Controle de
Segurança com BD para Registro de Usuários e
Envio de Notificação de Incidentes.**

Relatório do Projecto do Curso

Laércio Clementina Marcelino Macome

Supervisor: Mestre Engº J.P. Branco
(UEM, Faculdade de Engenharia, Departamento de Engenharia Electrotécnica)

Maputo, Novembro 2023

LAÉRCIO CLEMENTINA MARCELINO MACOME

Desenvolvimento de um Sistema de Controle de Segurança com BD para Registro de Usuários e Envio de Notificação de Incidentes.

Relatório apresentada ao Departamento de Engenharia Electrotécnica da Faculdade de Engenharia da Universidade Eduardo Mondlane – como requisito parcial para aprovação na disciplina projecto do curso.

Supervisor: Mestre EngºJ.P. Branco
(UEM, Faculdade de Engenharia, Departamento de Engenharia Electrotécnica)

Maputo, Novembro 2023

TERMO DE ENTREGA DO RELATÓRIO DO PC



UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

TERMO DE ENTREGA DE RELATÓRIO DO RELATÓRIO DO PROJECTO DO CURSO

Declaro que o estudante Laércio Clementina Marcelino Macome
entregou no dia _____ / _____ /20 _____ as _____ cópias do relatório do seu Relatório do
Projecto do Curso com a referência: _____
intitulado: Desenvolvimento de um Sistema de Controle de Segurança com BD para Re-
gistro de Usuários e Envio de Notificação de Incidentes.

Maputo, _____ de _____ de 20_____

O Chefe de Secretaria

DECLARAÇÃO DE HONRA

Declaro sobre palavra de honra que o trabalho apresentado neste relatório é original e foi por mim desenvolvido com base nos meus conhecimentos e com a ajuda dos recursos que ao longo do mesmo faço criteriosa referência.



UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA
FICHA-RESUMO DO RELATÓRIO DO PROJECTO DO CURSO

Referência do tema: _____

Título do tema: Desenvolvimento de um Sistema de Controle de Segurança com BD para Registro de Usuários e Envio de Notificação de Incidentes.

Nome do autor: Laércio Clementina Marcelino Macome

Supervisores: Mestre Engº J.P. Branco

RESUMO

Este foi um trabalho de pesquisa e desenvolvimento que consistiu em estudar uma estratégia de melhoramento do mecanismo usado pelo Homem para a sua proteção e seus pertences contra riscos oferecidos pela natureza e pelos seus semelhantes. No cerne do processo está uma trava eléctrica que será controlada por um programa armazenado num microcontrolador que por meio de consulta a uma base de dados, autoriza o acesso a usuários devidamente registados nela e nega aos que não constarem na base de dados. Trata-se portanto de um sistema de controle de segurança com uma base de dados que regista os usuários autorizados a aceder o referido local e também registrar os acessos realizados bem como a data e a hora de acesso. O sistema é também capaz de enviar notificações de incidentes de segurança como: três tentativas consecutivas de introdução de senha incorreta, esquecimento de fechar a porta ao sair do local. O presente relatório foi desenvolvido num ambiente LaTeX, sob template desenvolvido pela Engª I. A. J. Gune, baseado no actual regulamento de culminação de estudos da Faculdade de Engenharia para constituir corolário da aplicabilidade deste na produção uniformizada de relatórios de TL/EP na FENG.

GUIA DE AVALIAÇÃO DA APRESENTAÇÃO ORAL E DEFESA (PELO JÚRI)



UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

F2 – GUIA DE AVALIAÇÃO DA APRESENTAÇÃO ORAL E DEFESA

Nome do estudante Laércio Clementina Marcelino Macome

Referência do tema _____ Data ____/____/____

Título do tema: Desenvolvimento de um Sistema de Controle de Segurança com BD para Registro de Usuários e Envio de Notificação de Incidentes.

1. Introdução										
1.1. Apresentação dos pontos chaves na introdução (Contexto e importância do trabalho)	1	2	3	4	5	6	7	8	9	10
Secção 1 subtotal(max: 10)										

2. Organização e explanação										
2.1. Objectivos	1	2	3							
2.3. Metodologia	1	2	3	4						
2.4. Resultados, sua análise e discussão	1	2	3	4	5	6	7	8	9	10
2.5. Conclusões e aplicação dos resultados (recomendações)	1	2	3	4	5	6	7	8		
Secção 2 subtotal(max: 25)										

3. Estilo da apresentação										
3. 1. Uso efectivo do tempo	1	2	3	4	5					
3.2. Clareza, tom, vivacidade e entusiasmo	1	2	3	4	5					
3.3. Uso e qualidade dos audio-visuais	1	2	3	4	5					
Secção 3 subtotal(max: 15)										

4. Defesa										
4.1. Exactidão nas respostas	1	2	3	4	5	6	7	8	9	10
4.2. Domínio dos conceitos	1	2	3	4	5	6	7	8	9	10
4.3. Confiança e domínio do trabalho realizado	1	2	3	4	5	6	7	8	9	10
4.4. Domínio do significado e aplicação dos resultados	1	2	3	4	5	6	7	8	9	10
4.5. Segurança nas intervenções	1	2	3	4	5	6	7	8	9	10
Secção 3 subtotal(max: 50)										

Total de pontos (max: 100)	Nota (=Total*0,2)
---------------------------------------	--------------------------

GUIA DE AVALIAÇÃO DO RELATÓRIO ESCRITO



UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

F1 - GUIA DE AVALIAÇÃO DO RELATÓRIO ESCRITO

Nome do estudante Laércio Clementina Marcelino Macome

Referência do tema _____ Data ____/____/____

Titulo do tema: Desenvolvimento de um Sistema de Controle de Segurança com BD para Registro de Usuários e Envio de Notificação de Incidentes.

1. Resumo									
1.1. Apresentação dos pontos chaves no resumo (clareza, organização, correlação com o apresentado)	1	2	3	4	5				
Secção 1 subtotal (max: 5)									

2. Organização (estrutura) e explanação									
2.1. Objectivos	1	2	3	4	5				
2.2. Introdução, antecedentes e pesquisa bibliográfica	1	2	3	4	5	6	7	8	9
2.3. Metodologias	1	2	3	4	5	6	7	8	9
2.4. Resultados, sua análise e discussão	1	2	3	4	5	6	7	8	9
2.5. Conclusões e aplicação dos resultados (recomendações)	1	2	3	4	5	6	7	8	9
Secção 2 subtotal(max: 45)									

3. Argumentação									
3.1.Criatividade e originalidade	1	2	3	4	5				
3.2.Rigor	1	2	3	4	5				
3.3.Análise crítica, evidência e lógica	1	2	3	4	5	6	7	8	9
3.4.Relação objectivos/ métodos/ resultados/conclusões	1	2	3	4	5				
3.5.Relevância	1	2	3	4	5				
Secção 3 subtotal(max: 30)									

4. Apresentação e estilo da escrita									
4.1. Legibilidade e organização	1	2	3	4	5				
4.2. Ilustração e qualidade das figuras e tabelas	1	2	3	4	5				
4.3. Estilo da escrita (fluência do texto, uso da língua e gramática)	1	2	3	4	5				
4.4.Fontes bibliográficas (citação correcta, referências, etc)	1	2	3	4	5				
Secção 4 subtotal(max: 20)									

Total de pontos (max: 100)	Nota (=Total*0,2)
-----------------------------------	--------------------------

Nota: Quando exista a componente gráfica (desenhos técnicos), a nota acima é multiplicada por 0,8 cabendo os restantes 20% do peso à referida parte gráfica.

FICHA DE AVALIAÇÃO DA ATITUDE DO ESTUDANTE (PELO SUPERVISOR)



UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

FICHA DE AVALIAÇÃO DA ATITUDE DO ESTUDANTE

Nome do estudante Laércio Clementina Marcelino Macome

Referência do tema _____ Data ____/____/____

Título do tema: Desenvolvimento de um Sistema de Controle de Segurança com BD para Registro de Usuários e Envio de Notificação de Incidentes.

Indicador	Classificação				
Atitude geral (maneve uma disposição positiva e sentido de humor)	1	2	3	4	5
Dedicação e comprometimento (Deu grande prioridade ao projecto e aceitou as responsabilidades prontamente)	1	2	3	4	5
Independência (realizou as tarefas independentemente, como prometido e a tempo)	1	2	3	4	5
Iniciativa (viu o que devia ter sido feito e fê-lo sem hesitar e sem pressões do supervisor)	1	2	3	4	5
Flexibilidade (disponibilidade para se adaptar e estabelecer compromissos)	1	2	3	4	5
Sensibilidade (ouviu e tentou compreender as opiniões dos outros)	1	2	3	4	5
Criatividade (contribuiu com imaginação e novas ideias)	1	2	3	4	5
Total de pontos (max: 35)					

Valor do classificador	Cotação obtida	Significado
	1	Não aceitável (0 a 9 valores)
	2	Suficiente (10 a 13 valores)
	3	Bom (14 a 16 valores)
	4	Muito Bom (17 a 18 valores)
	5	Excelente (19 a 20 valores)

Total de pontos (max: 35)		Nota (=Total*20/35)	
----------------------------------	--	----------------------------	--

FICHA DE AVALIAÇÃO GLOBAL



UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

F3 - FICHA DE AVALIAÇÃO GLOBAL

Nome do estudante Laércio Clementina Marcelino Macome

Referência do tema _____ Data ___/___/___

Titulo do tema: Desenvolvimento de um Sistema de Controle de Segurança com BD para Registro de Usuários e Envio de Notificação de Incidentes.

AVALIADOR	NOTA OBTIDA	PESO(%)
Relatório escrito (F1)	N1=	A= 60
Apresentação e defesa do trabalho (F2)	N2=	B= 40

CLASSIFICAÇÃO FINAL =$(N1*A+N2*B)/100$	
--	--

OS MEMBROS DO JURI:

O Presidente	
O Oponente	
Os Supervisores	

TERMO DE ATRIBUIÇÃO DO TEMA DO PC



UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

TERMO DE ATRIBUIÇÃO DE TEMA DE RELATÓRIO DO PROJECTO DO CURSO

REFERÊNCIA DO TEMA:		DATA:	15/08/2023
---------------------	--	-------	------------

1. TÍTULO DO TEMA

Desenvolvimento de um Sistema de Controle de Segurança com BD para Registro de Usuários e Envio de Notificação de Incidentes.

2. DESCRIÇÃO SUMÁRIA DO TRABALHO A DESENVOLVER

2.1. Definição do Problema

A acessibilidade de locais desde tempos passados era feita por meio de fechaduras mecânicas. Tal meio não garantia máxima segurança criando assim constrangimentos na segurança, dai novas formas de acessibilidade foram sendo pesquisadas e desenvolvidas como solução, resolvendo problemas tais como:

- Incapacidade de acesso a residência ou qualquer outro local por falta da chave quer por perda ou esquecimento da mesma associado ao numero de cópias de chaves disponibilizadas por fechadura;
- Vulnerabilidade a intrusão e garantir acessibilidade a qualquer um que tivesse acesso as chaves quer de forma legal (com autorização) ou ilícita;
- Vulnerabilidade da segurança e intrusão em caso de esquecimento de trancar a porta;
- Substituir a fechadura por completo ou ter a sua chave clonada (replicada), estando vulnerável a intrusos em caso de ter a sua chave roubada ou perdida.

Assim de forma a mitigar estes constrangimentos vê-se como solução este sistema de acesso que será capaz de manter o local onde este esteja instalado, seguro. Substituindo a chave por uma senha a ser definida pelo usuário. Podendo ainda oferecer a possibilidade de adicionar múltiplos usuários cada um com a sua senha e associado a isto registrar os acessos feitos num determinado período numa base de dados, bem como enviar notificações de incidentes de segurança.

2.2. Relevância da pesquisa

A adopção deste sistema de controle de acesso irá permitir:

1) Maior segurança

No sentido de que ao eliminar o uso da chave física por uma senha/password torna difícil a intrusão por métodos conhecidos. Garantir que o usuário seja alertado em caso de esquecer-se de fechar a porta.

2) Maior comodidade

Torna a vida mais prática no sentido de que não precisa se preocupar em andar com um numero elevado de chaves e nem esconder a chave por debaixo do vaso.

3) Limitação e controle de acesso a locais reservados

A adopção deste sistema como meio de controle de acessibilidade desbloqueia inúmeras possibilidades de definir e controlar acessos especiais em locais reservados.

4) Acesso por parte de múltiplos usuários e inúmeras possibilidades de controle de acesso dos mesmos

Possibilita definir horários em que as senhas de determinados usuários estão autorizadas a ter acesso, proporcionando melhor controle de quem tem acesso ao local.

2.3. Objectivos

2.3.1. Objectivos Gerais

- a. Projectar e implementar um sistema de segurança de uma residência com controle de acesso e contra intrusão com capacidade para registrar múltiplos usuários, e dados de outros eventos de segurança e notificação de incêndios.

2.3.2. Objectivos Específicos

- a. Estudar e compreender os processos de segurança de uma residência, os pontos de interesse e as tecnologias envolvidas para o seu monitoramento;
- b. Projectar o sistema de maneira a que ele cumpra com as funcionalidades mínimas desde o cadastro de usuários até a notificação de incidentes de segurança. Programar o mecanismo de tranque e destranque;
- c. Desenvolver o mecanismo de alerta por meio de um alarme em caso de esgotamento das tentativas definidas para inserir a senha correcta;
- d. Identificar de forma generalizada os componentes necessários para o desenvolvimento e implementação do sistema em termos de Hardware e software;
- e. Desenvolver uma BD que será responsável por armazenar localmente e na nuvem os usuários e as respectivas senhas bem como armazenar os acessos efectuados pelos mesmos;
- f. Realizar simulações e apresentar os resultados.

2.4. Metodologia

De maneira a materializar a ideia de solução proposta neste projecto serão seguidas as seguintes etapas:

- Recolha de dados

A recolha de dados foi feita com base em pesquisas exploratórias. Esta tem como intuito a recolha de dados sobre situações passadas e actuais do funcionamento do controle de acesso, segurança residencial e controle de incêndio, com vista a conhecer o histórico e incidentes de intrusão do passado e actuais.

- Pesquisa documental

A pesquisa documental foi realizada através da leitura de manuais, com a finalidade de adquirir informações sobre a estrutura dos sistemas de controle de acesso e descrição generalizada dos elementos que o compõem.

- Pesquisa bibliográfica

A pesquisa bibliográfica tem como finalidade realizar pesquisas através da coleta de informações de materiais bibliográficos (manuais, vídeos na internet, fóruns e stack overflow) publicados por diversos autores, que discutem sobre o assunto relacionada com o tema em estudo.

3. LOCAL DE REALIZAÇÃO

--

4. SUPERVISORES

	Nome	Assinatura
Da UEM	Mestre Engº J.P. Branco	
Da Instituição		

5. DATAS CHAVE

Entrega do tema	09/09/2022	Previsão da conclusão	13/12/2022
-----------------	------------	-----------------------	------------

Maputo, _____ de _____ de 20_____

Chefe da Comissão Científica

Visto do chefe do departamento

Declaro que recebi o tema do Relatório do Projecto do Cursona data acima indicada

Nome: _____

Assinatura: _____

ACTA DE ENCONTROS REGULARES



UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

ACTA DE ENCONTROS

REFERÊNCIA DO TEMA:		DATA:	16/11/2023
---------------------	--	-------	------------

1. AGENDA:

- | |
|---|
| 1. Revisão e organização de alguns pontos no relatório. |
|---|

2. PRESENÇAS:

Supervisor	Mestre Engº J.P.Branco
Co-Supervisor	
Estudante	Macome, Laércio Clementina Marcelino
Outros	

3. RESUMO DO ENCONTRO:

Este foi um encontro virtual, que ocorreu pós o envio da versão atual do relatório ao supervisor, que por sua vez analisou e deixou ficar algumas recomendações.
--

4. RECOMENDAÇÕES:

O supervisor recomendou alterações de alguns pontos nas secções 1.3, 1.4.2, 4.
--

5. Observações

--

6. DATA DO PRÓXIMO ENCONTRO

--

EPÍGRAFE

Nossas dúvidas são traidoras e nos fazem perder o que, com frequência, poderíamos conquistar, senão fosse o medo de tentar.

WILLIAM SHAKESPEARE

DEDICATÓRIA

*Aos meus pais pelo apoio, incentivo e compreensão; e a todos
que deste trabalho se vão beneficiar.*

AGRADECIMENTOS

Início esta secção de agradecimentos expressando a minha mais profunda gratidão a Deus pelo dom da vida e pelo amparo e por iluminar os caminhos que possibilitaram o progresso apresentado no presente relatório.

Em sequência agradeço aos meus orientadores, Mestre Engº J.P. Branco e Doutor Engº G.J.Doho, pelas valiosas orientações e paciência ao logo do processo da elaboração do projecto. Seus conhecimentos foram fundamentais para o desenvolvimento deste trabalho. Por estas e por outras, estou profundamente grato pelas vossas orientações.

Agradeço ao meus colegas, com quem tive a oportunidade de vivenciar e dividir as peripécias desta jornada académica. As discussões, sugestões, incentivos e auxílios estarão cravadas na memória.

Encerro esta secção expressando a minha profunda gratidão ao meus pais pelo incondicional apoio, incentivo e auxílio ao longo deste processo. Em meio aos obstáculos com os quais deparei-me suas palavras de encorajamento foram fonte de energia e esperança.

Atenciosamente,

Macome, Laércio Clementina Marcelino

RESUMO

Desenvolvimento de um Sistema de Controle de Segurança com BD para Registro de Usuários e Envio de Notificação de Incidentes.

O presente relatório descreve o processo de desenvolvimento de um sistema de controle de segurança com uma base de dados para registro de usuários e envio de notificação de incidentes. Através do uso da automação foram evitados estudos que tornassem possível a automação de um mecanismo muito usado no cotidiano (a fechadura). Esta proposta surge da constatação do elevado custo de comercialização destes mecanismos em Moçambique como consequência de estes serem na sua totalidade importados. Para o desenvolvimento deste sistema, foram estudados e selecionados os hardwares que tornariam possível a materialização do mesmo. Hardware este que será descrito no decorrer do presente documento. Foi usada a linguagem de programação C/C++ para possibilitar a interação destes componentes entre si, tanto com o Homem, aliada a esta linguagem foram usadas também instruções SQL para definir e manipular os dados da base de dados que integra o sistema. O sistema consiste de um microcontrolador que funciona como o cérebro das operações, este colhe as senhas digitadas por meio de um teclado matricial. Estas senhas são comparadas com as que constam na base de dados MySQL, a integração da base de dados ao sistema é possível graças ao módulo Ethernet que permite a ligação do microcontrolador ao servidor por meio de um cabo de rede e um roteador com auxílio do protocolo TCP/IP. Consultada a base de dados, se for um usuário autorizado o microcontrolador aciona um relé de estado sólido que por sua vez aciona uma trava eléctrica destravando-a liberando o acesso e no mesmo instante o microcontrolador regista o acesso na base de dados. O conjunto de informações envolvidas no processo são apresentadas ao usuário por meio de um display LCD 16x2 que é ligado ao microcontrolador por meio de um módulo I2C por meio do protocolo TWI. Para o caso de ser introduzida uma senha incorreta três vezes o sistema envia um notificação via sms informando ao Admin sobre a eventualidade de segurança por meio de um módulo GSM integrado no sistema

Palavras Chaves: Automação, Fechadura Electrónica, Base de dados, Sistema, Microcontrolador, Arduino com MySQL

ABSTRACT

Desenvolvimento de um Sistema de Controle de Segurança com BD para Registro de Usuários e Envio de Notificação de Incidentes.

This report describes the process of developing a security control system with a database for registering users and sending incident notifications. Through the use of automation, studies were highlighted that made it possible to automate a mechanism that is widely used in everyday life (the lock). This proposal arises from the observation of the high cost of marketing these mechanisms in Mozambique as a result of them being entirely imported. To develop this system, the hardware that would make it possible to materialize was trained and selected. The hardware will be described throughout this document. The C/C++ programming language was used to enable the interaction of other components with each other, much like Man. In addition to this language, SQL instructions were also used to define and manipulate the data in the database that integrates the system. The system consists of a microcontroller that works as the brain of operations, it is collected as digitized passwords using a matrix keyboard. These passwords are compared with those contained in the MySQL database. The integration of the database into the system is possible thanks to the Ethernet module that allows the microcontroller to be connected to the server using a network cable and a router with the help of the protocol TCP/IP. After consulting the database, for an authorized user, the microcontroller activates a solid state relay which in turn activates an electrical lock, unlocking it, allowing access and at the same time the microcontroller registers the access in the database. The set of information involved in the process is presented to the user through a 16x2 LCD display that is connected to the microcontroller through an I2C module using the TWI protocol. If an incorrect password is changed three times, the system sends a notification via SMS informing the Admin about the security situation through a GSM module integrated into the system.

Keywords: Automation, Electronic Lock, Database, System, Microcontroller, Arduino with MySQL

Índice de Conteúdo

Índice	xviii
Lista de Figuras	xx
Lista de Acrónimos	xxiv
1 Introdução	1
1.1 Formulação do problema	1
1.2 Delimitação	1
1.2.1 Temporal	1
1.3 Justificativa	2
1.4 Objectivos	2
1.4.1 Objectivos Gerais	2
1.4.2 Objectivos Específicos	3
1.5 Metodologia de investigação	3
1.6 Estrutura do trabalho	4
2 Revisão Teórica	5
2.1 Sistema de segurança residencial	5
2.1.1 Tipos de sistemas de segurança residencial	5
2.1.1.1 Sistema de controle de acesso	6
2.2 Protocolos de comunicação	6
2.2.1 Protocolo de comunicação I2C	6
2.2.1.1 Características do barramento	7
2.2.1.2 Modo de funcionamento	8
2.2.2 Redes de Computadores	8
2.2.2.1 Comutação de circuitos	8
2.2.2.2 Comutação de pacotes	9

2.2.3	Arquitectura cliente/servidor	9
2.2.3.1	Tipos de servidores	9
2.2.4	Protocolo TCP/IP	11
2.2.4.1	Camada de aplicação	11
2.2.4.2	Protocolos da camada de aplicação	12
2.2.4.3	DNS(<i>Domain Name System</i>)	12
2.2.4.4	FTP (<i>File Transfer Protocol</i>)	13
2.2.4.5	HTTP (<i>Hyper Text Transfer Protocol</i>)	13
2.2.4.6	Camada de Transporte	13
2.2.4.7	Camada de rede	14
2.2.4.8	Endereçamento IP	14
2.2.4.9	IPv4	15
2.2.4.10	Mascara de rede	16
2.2.4.11	ARP (<i>Address Resolution Protocol</i>)	16
2.2.4.12	Camada Enlace-física (<i>Camada de interface com a rede</i>) .	17
2.3	Banco de dados	18
2.3.1	Linguagens e interfaces do banco de dados	19
2.3.1.1	SQL	20
2.3.2	Modelo de banco de dados relacional	20
3	Descrição do Hardware	21
3.1	Arduino Mega 2560	21
3.2	Ethernet Shield <i>W5100</i>	24
3.2.1	Especificações	25
3.3	Display LCD(<i>Liquid Crystal Display</i>) 16x2 e Módulo I2C	25
3.3.1	Princípio de funcionamento	26
3.3.2	Diagrama de pinos	26
3.4	Teclado matricial	28
3.4.1	Princípio de funcionamento	28
3.5	Modulo GSM e regulador de tensão LM2596	29
3.5.1	Especificações	30
3.6	Módulo relé	30
3.6.1	Constituição e princípio de funcionamento	31
3.6.2	Princípio de funcionamento	32

3.6.3	Módulo relé	32
3.6.4	Especificações	33
3.7	Trava eléctrica (<i>Acturador solenóide linear</i>)	33
3.7.1	Constituição da Solenóide	33
3.7.2	Princípio de funcionamento	34
3.7.3	Especificações	35
4	Desenvolvimento e descrição do protótipo	36
4.1	Visão geral	36
4.2	Requisitos do sistema	37
4.2.1	Requisitos gerais:	37
4.2.2	Requisitos específicos:	38
4.3	Descrição funcional do sistema	38
4.3.1	Para egressar da residência:	38
4.3.2	Para acessar a residência:	39
5	Ensaios e resultados	42
5.0.1	Resultados da conexão com a base de dados	43
5.1	Custos do projecto	44
6	Conclusões e recomendações	46
6.1	Conclusões	46
6.2	Recomendações	47
Referências Bibliográficas		48
Anexos		50

Lista de Figuras

2.1 Arquitectura I2C. Fonte: https://www.totalphase.com/blog/2020/12/differences-between-uart-i2c/	7
2.2 Servidor de arquivos. fonte: [Torres2001]	10
2.3 Servidor de aplicações.Fonte: [Torres2001]	10
2.4 Funcionamento da camada de aplicação. Fonte: [Torres2001]	12
2.5 funcionamento do FTP. Fonte: [Torres2001]	13
2.6 Funcionamento do ARP. Fonte [Torres2001]	17
2.7 Esquema completo de um computador operando com protocolo TCP/IP. Fonte [Torres2001]	17
2.8 diagrama esquemático de um sistema de banco de dados. Fonte [Navathe2011]	19
3.1 Diagrama de pinos do Arduino MEGA2560. Fonte: https://embarcados.com.br/arduino-mega-2560/	22
3.2 Ethernet Shield. Fonte: https://www.robotistan.com/ethernet-shield-wiznet-w5100-for-arduino	24
3.3 Hitachi-HD44780. Fonte: https://tresdprinttech.com.mx/pantallas-lcdoled/245-lcd-16x2-fondo-verde-7503040256572.html	26
3.4 Diagrama de pinos do display. Fonte: O autor	27
3.5 Módulo I2C e seu diagrama de pinos. Fonte: O autor	27
3.6 Teclado matricial e sua composição interna	28
3.7 Módulo SIM800L e seu diagrama de pinos. Fonte: https://www.mathaelectronics.com/what-is-a-sim800l-gprs-gsm-module/	29
3.8 Constituição do relé. Fonte: adaptado de: https://cromatek.com.br/reles-o-que-sao-e-para-que-servem/	31
3.9 circuito interno de um módulo relé. Fonte: https://www.robocore.net/tutoriais/introducao-ao-rele	32

3.10 constituição de um Solenóide. Fonte: [Willem2016]	33
3.11 Campo gerado por um bobina percorrida por corrente. Fonte: [Willem2016]	34
4.1 Diagrama de blocos do sistema. Fonte: O autor	37
5.1 Ilustração gráfica do hardware. Fonte: Autor	42
5.2 Mensagem de confirmação de conexão com a base de dados. Fonte: Autor	43
5.3 Mensagem de falha de conexão com a base de dados. Fonte: Autor . . .	43
5.4 Mensagem de consulta bem sucedida e registro de histórico	44
5.5 Mensagem de consulta mal sucedida. Fonte: Autor	44

Lista de Tabelas

2.1	Equivalência entre as camadas do protocolo TCP/IP e modelo OSI	11
2.2	Instruções SQL	20
3.1	Descrição resumida dos pinos	23
3.2	portas reservadas	25
3.3	Especificações da Ethernet Shield	25
3.4	Pinos do módulo SIM800L	30
4.1	Componentes que compõem o sistema e as respectivas funções	40
5.1	Levantamento dos custos para realização do projecto	45

Lista de Acrónimos

ARP Protocolo de Resolução de Endereços - *Address Resolution Protocol.* 14

DDL - *Data definition Language.* 19

DML - *Data Manipulation Language.* 19

DNS Sistema de Nomes de Domínio - *Domain Name System.* 11

FTP Protocolo de Transferência de Ficheiro - *File Transfer Protocol.* 11

HTTP Protocolo de Transferência de HiperTexto - *HyperText Transfer Protocol.* 11

I2C Circuitos Interintegrados - *Inter-Integrated Circuits.* 6

ICMP Protocolo de mensagem de controlo de *Internet* - *Internet Control Message Protocol.* 14

ICSP - *In-Circuit Serial Programming.* 21

IP Protocolo de *Internet* - *Internet Protocol.* 8

LCD Visor de cristal líquido - *Liquid Crystal Display.* 25

MAC - *Media Access Control.* 16

PWM - *Pulse Width Modulation.* 21

RARP Protocolo de Resolução de Endereços Reverso - *Reverse Address Resolution Protocol.* 14

SCA Sistema de Controle de Acesso. 6

SCL linha de relógio serial - *serial clock line.* 6

SDA linha de dados seriais - *serial data line.* 6

SGBD Sistema de gerenciamento de banco de dados - *Database Management System*.

18

SMTP Protocolo de Transferência de Correio Simples - *Simple Mail Transfer Protocol*. 11

SNMP Protocolo Simples de Gerenciamento de Rede - *Simple Network Management Protocol*. 11

SPI - *In-System Programmer*. 24

TCP Protocolo de Controle da Transmissão - *Transmission Control Protocol*. 8

TWI Interface de dois fios - *two-wired interface*. 6

UDP Protocolo de Datagramas de Utilizador - *User Datagram Protocol*. 14

USB - *Universal Serial Bus*. 21

Capítulo 1

Introdução

1.1 Formulação do problema

A acessibilidade de locais desde tempos passados era feita por meio de fechaduras mecânicas. Tal meio não garantia máxima segurança criando assim constrangimentos na segurança, dai novas formas de acessibilidade foram sendo pesquisadas e desenvolvidas como solução, resolvendo problemas tais como:

- Incapacidade de acesso a residência ou qualquer outro local por falta da chave quer por perda ou esquecimento da mesma associado ao numero de cópias de chaves disponibilizadas por fechadura;
- Vulnerabilidade a intrusão e garantir acessibilidade a qualquer um que tivesse acesso as chaves quer de forma legal (com autorização) ou ilícita;
- Vulnerabilidade da segurança e intrusão em caso de esquecimento de trancar a porta;
- Substituir a fechadura por completo ou ter a sua chave clonada (replicada), estando vulnerável a intrusos em caso de ter a sua chave roubada ou perdida.

1.2 Delimitação

1.2.1 Temporal

O presente projecto decorreu desde o dia 09 de agosto até 16 de novembro de 2023, compreendendo o período para idealização do projecto, construção do protótipo e por fim

os ensaios.

1.3 Justificativa

Segundo a pirâmide de Maslow, também conhecida como teoria das necessidades humanas, que organiza de forma hierárquica as necessidades humanas a segurança é das necessidades mais urgentes do ser humano. Foi na perspectiva de satisfazer esta necessidade que desde a pré-história o homem já vinha apercebendo-se da necessidade de proteger-se e manter seus bens pessoais seguros contra riscos oferecidos pela natureza e por seus semelhantes. Começou por colocar pedras na entrada da caverna para impedir a entrada de visitantes indesejados bem como animais selvagens, quando as suas necessidades relativas a segurança foram evoluindo, este mecanismo já não era satisfatório e dessa insatisfação a 4000 anos atrás criou a primeira fechadura, feita de madeira. Esta foi o ponto de partida para as fechaduras mecânicas usadas até hoje. Estes mecanismos até desempenham as suas funções, mas apresentam os constrangimentos previamente mencionados na formulação do problema. Assim de forma a mitigar estes constrangimentos vê-se como solução este sistema de acesso que será capaz de manter o local onde este esteja instalado, seguro. Substituindo a chave por uma senha a ser definida pelo usuário. Podendo ainda oferecer a possibilidade de adicionar múltiplos usuários cada um com a sua senha e associado a isto registrar os acessos feitos num determinado período numa base de dados, bem como enviar notificações de incidentes de segurança.

1.4 Objectivos

1.4.1 Objectivos Gerais

- Projectar e implementar um sistema de segurança de uma residência com controle de acesso e contra intrusão com capacidade para registrar múltiplos usuários, e dados de outros eventos de segurança e notificação de incêndios.

1.4.2 Objectivos Específicos

- Estudar e compreender os processos de segurança de uma residência, os pontos de interesse e as tecnologias envolvidas para o seu monitoramento;
- Projectar o sistema de maneira a que ele cumpra com as funcionalidades mínimas desde o cadastro de usuários até a notificação de incidentes de segurança. Programar o mecanismo de tranque e destranque;
- Desenvolver o mecanismo de alerta por meio de um alarme em caso de esgotamento das tentativas definidas para inserir a senha correcta;
- Identificar de forma generalizada os componentes necessários para o desenvolvimento e implementação do sistema em termos de Hardware e software;
- Desenvolver uma BD que será responsável por armazenar localmente e na nuvem os usuários e as respectivas senhas bem como armazenar os acessos efectuados pelos mesmos;
- Realizar simulações e apresentar os resultados.

1.5 Metodologia de investigação

De maneira a materializar a ideia de solução proposta neste projecto serão seguidas as seguintes etapas:

- Recolha de dados

A recolha de dados foi feita com base em pesquisas exploratórias. Esta tem como intuito a recolha de dados sobre situações passadas e actuais do funcionamento do controle de acesso, segurança residencial e controle de incendio, com vista a conhecer o histórico e incidentes de intrusão do passado e actuais.

- Pesquisa documental

A pesquisa documental foi realizada através da leitura de manuais, com a finalidade de adquirir informações sobre a estrutura dos sistemas de controle de acesso e descrição generalizada dos elementos que o compõem.

- Pesquisa bibliográfica

A pesquisa bibliográfica tem como finalidade realizar pesquisas através da coleta

de informações de materiais bibliográficos (manuais, vídeos na internet, fóruns e stack overflow) publicados por diversos autores, que discutem sobre o assunto relacionada com o tema em estudo.

1.6 Estrutura do trabalho

O trabalho está dividido em seis capítulos, que são apresentados a seguir:

Capítulo 1 - Neste capítulo é apresentada a formulação e delimentação do problema, ou seja, a composição do anexo cinco preenchido.

Capítulo 2 - Neste capítulo é apresentada a fundamentação teórica relacionada aos conceitos abordados na pesquisa.

Capítulo 3 - De seguida é apresentada a descrição do hardware que implementa o sistema.

Capítulo 4 - Em sequência é apresentado o desenvolvimento e a descrição do protótipo.

Capítulo 5 - Neste capítulo são apresentados os ensaios e os resultados da simulação do protótipo.

Capítulo 6 - Por fim é realizada uma síntese dos resultados da pesquisa, assim como tecidas algumas considerações finais em jeito de recomendações.

Capítulo 2

Revisão Teórica

2.1 Sistema de segurança residencial

Sistema de segurança residencial, também conhecido como Sistema de Vigilância Residencial, é um conjunto de equipamentos, dispositivos e recursos de segurança, utilizados com finalidade de propiciar segurança a uma residência e seus ocupantes. Estes sistemas têm como objectivos:

- Dissuadir qualquer intensão criminosa;
- Identificar e alertar sobre tentativas de acessos não autorizados;
- Registrar acessos não autorizados e eventos indesejados para fins de investigação e responsabilização

2.1.1 Tipos de sistemas de segurança residencial

Os sistemas de segurança residencial podem ser divididos em diferentes tipos, de acordo com a sua aplicabilidade. Dentre elas podemos citar:

- Sistemas de segurança física – têm como objectivo proteger o local contra acessos físicos não autorizados. Temos como exemplo deste tipo de sistema: barreiras físicas como cercas, portões, portas, iluminação;
- Sistema de controle de acesso – Estes dizem respeito a tecnologias (hardware e software) que têm como função o gerenciamento de acesso. Como por exemplo: portas automatizadas, fechaduras electrónicas, etc.

- Sistema de alarme – Estes têm como objectivo alertar movimentos incomuns ou situações de riscos. Exemplo: sensores de movimento, sensores de abertura de porta, etc.

2.1.1.1 Sistema de controle de acesso

Como dito na secção anterior um Sistema de Controle de Acesso (SCA), diz respeito a tecnologias (hardware e software) que têm como função o gerenciamento de acessos. Estes sistemas devem obrigatoriamente (1) assegurar entrada fácil e simplificada às pessoas devidamente autorizadas, (2) impedir o acesso de pessoas não autorizadas, (3) registrar movimentações referentes aos acessos permitidos, (4) identificar, alertar e registrar tentativas de acessos não autorizados e (5) permitir consultas e emitir relatórios sobre acessos consentidos, com detalhes da sua movimentação, assim como das tentativas de acessos não autorizadas. A arquitectura básica de sistemas de controle de acesso é composta por: (1) Fonte de alimentação, (2) Microcontroladore, (3) terminais de operação, (4) bloqueios.

2.2 Protocolos de comunicação

De acordo com [Popovic2006] protocolos de comunicação são definidos de diversas formas, como por exemplo: “Um conjunto estabelecido de convenções pelas quais dois computadores ou dispositivos de comunicação validam o formato e conteúdo das mensagens trocadas;” “um conjunto de interfaces que permitem com que computadores comuniquem uns com os outros;” “métodos pelos quais dois computadores coordenam a comunicação entre eles;” “as regras que regem a troca de informação entre dispositivos em um link de dados.” Para o presente projecto interessa-nos descrever um protocolo de comunicação serial criado pela empresa Philips nos finais dos anos 90, que é muito usado em projectos de sistemas embutidos.

2.2.1 Protocolo de comunicação I2C

Circuitos Interintegrados - *Inter-Integrated Circuits* (I2C), também conhecido como Interface de dois fios - *two-wired interface* (TWI) . De acordo com o datasheet do protocolo, versão 7.0 de 2021, I2C é um protocolo em que são necessários somente dois fios, linha de dados seriais - *serial data line* (SDA) e o linha de relógio serial - *serial clock line* (SCL)

para a comunicação. Cada um dos dispositivos conectados ao barramento são endereçáveis por via de software usando um endereço exclusivo e relacionamentos simples mestre/escravo existentes o tempo todo, sendo que o mestre pode operar com mestre-transmissor ou como mestre-receptor. Conforme podemos ver na figura 2.1 em termos de hardware o barramento possui dois sinais, um para o clock e outro para dados.

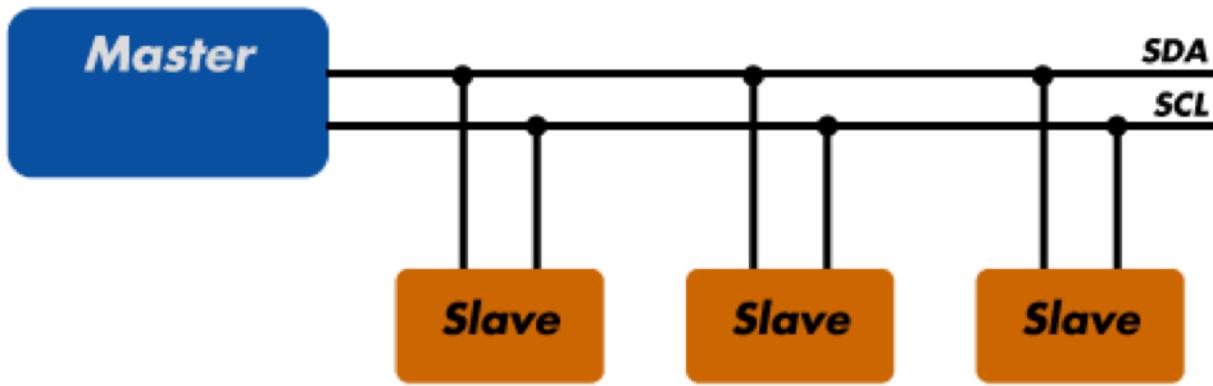


Figura 2.1: Arquitectura I2C. Fonte: <https://www.totalphase.com/blog/2020/12/differences-between-uart-i2c/>

2.2.1.1 Características do barramento

Para o barramento são estabelecidas as seguintes características gerais:

Tanto a SDA bem como a SCL, são bi-direcionais. Quando o barramento está livre, as duas linhas devem estar em nível alto.

- Condição de inicio

A condição que indica inicio é uma transição de nível alto para nível baixo na linha de dados (SDA), enquanto a linha de clock (SCL) está em nível alto.

- Condição de Fim

A condição que indica o fim é uma transição de nível baixo para nível alto na linha de dados (SDA) enquanto a linha de clock (SCL) está em nível alto.

- Condição de barramento não ocupado

A condição que indica que o barramento está livre é ambas a linhas (SDA e SCL) estarem em nível alto.

- Condição de validação de dados

O estado da linha de dados representa dados válidos quando, após a condição de

inicio de dados, durante o período do nível alto da linha do clock a linha de dados se mantiver estável.

Cada transferência de dados é iniciada com a condição de inicio e terminada com a condição de fim de dados.

O dispositivo mestre determina a quantidade de bytes a serem transferidos entre a condição de inicio e a de paragem.

- Acknowledge

Geralmente todo o receptor que tenha sido endereçado é obrigado a gerar um acknowledge após cada byte recebido.

2.2.1.2 Modo de funcionamento

De acordo com o datasheet do protocolo, tanto o dispositivo mestre assim como o dispositivo escravo podem enviar ou ler dados no barramento, isto é, qualquer um pode ser transmissor ou receptor, mas sempre é o dispositivo mestre quem controla o barramento e quem gera a condição de inicio e paragem de envio de dados no barramento.

O protocolo I2C possui endereçamento de 7 bits, o que teoricamente dá a este, capacidade para suportar 127 dispositivos em modo slave em seu barramento.

2.2.2 Redes de Computadores

Antes de falar do protocolo Protocolo de Controle da Transmissão - *Transmission Control Protocol* (TCP)/ Protocolo de Internet - *Internet Protocol* (IP) é importante falar de redes.

Segundo [Kozierok2005], redes de computadores são descritas como dispositivos conectados entre eles usando hardware e software, para permitir partilha de informação.

Esta partilha de informações é feita com base nalguns métodos. [Kozierok2005] menciona dois métodos como sendo os mais importantes, nomeadamente: comutação de pacotes e comutação de circuitos.

2.2.2.1 Comutação de circuitos

Este é formado por um conjunto de comutadores conectados por links físicos. Um exemplo clássico de comutação de circuitos é o sistema de telefones por fio. Quando realizamos uma chamada e alguém atende, é estabelecida uma conexão por meio de um circuito

e a partir desse momento podemos transmitir as informações desejadas. [Kozierok2005]

2.2.2.2 Comutação de pacotes

Diferente da comutação de circuitos, na comutação de pacotes não existe alocação fixa de recursos para um pacote. A alocação é feita sob demanda. É diferente da comutação de circuitos que ocorre na camada física, este ocorre na camada de rede. [Behrouz2010]

2.2.3 Arquitectura cliente/servidor

Se a rede que estiver sendo planejada for ter mais de 10 microprocessadores instalados ou no caso de redes pequenas onde a segurança for uma questão importante, então a escolha natural é uma rede do tipo Cliente/Servidor.

Servidor – é um microprocessador ou software especializado em um só tipo de tarefa, não sendo usado para outra finalidade. Entretanto, em redes onde o desempenho não chega a ser um problema, pode ser que se encontrem servidores sendo usados também como estações de trabalho ou sendo usado como servidor de mais de uma tarefa. [Torres2001]

Cliente – O cliente também pode ser um microprocessador ou software, normalmente acionado por um usuário. Cabe ao cliente iniciar a comunicação com o servidor, seja accionada directamente pelo usuário ou de forma automática, em resposta a um evento ou uma acção externa. Um dispositivo IOT também pode actuar como cliente, acessando servidores para buscar ou actualizar informações sobre seu funcionamento. [Jaime2022]

2.2.3.1 Tipos de servidores

Conforme mencionado anteriormente, o servidor não é necessariamente um microcomputador, pode ser um aparelho que desempenha igual função. Existem diversos tipos de servidores, mas neste projecto importa-nos falar somente de dois deles:

Servidor de Arquivos: é um servidor responsável pelo armazenamento de arquivos de dados, como arquivos de texto, planilhas e gráficos, que necessitam ser compartilhados com os usuários da rede. Note que o programa necessário para ler o arquivo (o processa-

dor de textos, por exemplo) é instalado e executado na máquina do usuário (cliente) e não no servidor. Nesse servidor não há o processamento de informações, o servidor é responsável apenas por entregar o arquivo solicitado, para então o arquivo ser processado no cliente. [Torres2001]

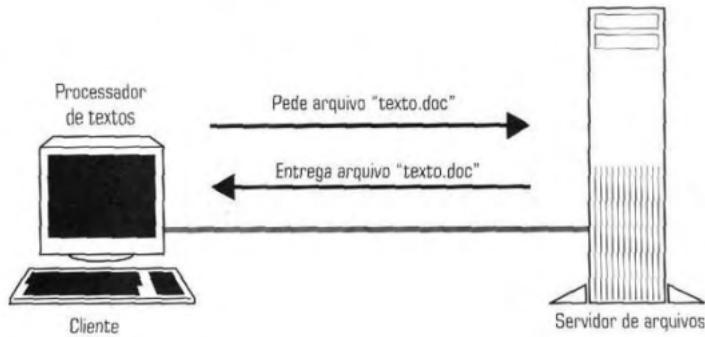


Figura 2.2: Servidor de arquivos. fonte: [Torres2001]

Servidor de Aplicações: o servidor de aplicações é responsável por executar aplicações cliente/servidor, como, por exemplo, um banco de dados. Ao contrário do servidor de arquivos, que somente armazena arquivos de dados e não os processa, o servidor de aplicações executa as aplicações e processa os arquivos de dados. Por exemplo, quando um micro cliente faz uma consulta em um banco de dados cliente/servidor, essa consulta será processada no servidor de aplicações e não no micro cliente, o micro cliente apenas mostrará o resultado enviado pelo servidor de aplicações. Com isso é possível que vários usuários accessem e manipulem ao mesmo tempo uma única aplicação, fazendo com que todos os dados fiquem sincronizados. [Torres2001]



Figura 2.3: Servidor de aplicações. Fonte: [Torres2001]

2.2.4 Protocolo TCP/IP

O protocolo de comunicação TCP/ IP agrupa dois protocolos considerados os mais importantes: TCP e o IP. Tal como o modelo OSI o protocolo TCP/IP é dividido em camadas, que permitem que dois sistemas diferentes se comuniquem independentemente de suas arquitecturas subjacentes. Apesar de o protocolo TCP/ IP ter sido criado antes do modelo OSI, existem similaridades entre TCP/ IP e o modelo OSI. O protocolo TCP/ IP é composto por 4 camadas, nomeadamente: enlace-física, rede, transporte e aplicação. [Behrouz2010]

Podemos ver na tabela 2.1 a equivalência entre as camadas do protocolo TCP/ IP e o modelo OSI.

Tabela 2.1: Equivalência entre as camadas do protocolo TCP/IP e modelo OSI

TCP/IP	Modelo OSI
Aplicação	Aplicação
	Apresentação
	Sessão
Transporte	Transporte
Rede	Rede
Enlace-física	Enlace
	Física

2.2.4.1 Camada de aplicação

Esta camada equivale às camadas 5, 6 e 7 do modelo OSI. Ela faz a comunicação entre os aplicativos e o protocolo de transporte. Os protocolos mais conhecidos que operam na camada de aplicação são: Protocolo de Transferência de HiperTexto - *HyperText Transfer Protocol* (HTTP), Protocolo de Transferência de Correio Simples - *Simple Mail Transfer Protocol* (SMTP), Protocolo de Transferência de Ficheiro - *File Transfer Protocol* (FTP), Protocolo Simples de Gerenciamento de Rede - *Simple Network Management Protocol* (SNMP), Sistema de Nomes de Domínio - *Domain Name System* (DNS).

A camada de aplicação comunica-se com a camada de transporte através de uma porta. As portas são enumeradas e as aplicações padrão usam sempre uma mesma porta. Por exemplo, o protocolo SMTP utiliza sempre a porta 25, o protocolo HTTP utiliza sempre a porta 80 e o FTP as portas 20 para transmissão de dados e 21 para transmissão de

informação de controle. [Torres2001]

O uso de um número de porta permite ao protocolo de transporte (tipicamente o TCP) saber qual é o tipo de conteúdo do pacote de dados (por exemplo, saber que dado que ele está a transportar. É um e-mail) e no receptor, saber qual protocolo de aplicação ele deverá entregar o pacote destinado à porta 25, o protocolo TCP irá entregá-lo ao protocolo que estiver conectado a esta porta, tipicamente o SMTP, que por sua vez entregará o dado à aplicação que o solicitou (o programa de e-mail). [Torres2001]

Assim sendo, quando um programa cliente de e-mail quer baixar os e-mails que estão armazenados no servidor de e-mail, ele irá efectuar esse pedido para a camada de aplicação do TCP/ IP, sendo atendido pelo protocolo SMTP. Ou para o caso em que entramos em um endereço WWW em um browser para visualizar uma página na internet, o browser irá comunicar-se com a camada de aplicação do TCP/ IP, sendo atendido pelo protocolo TCP/ HTTP.

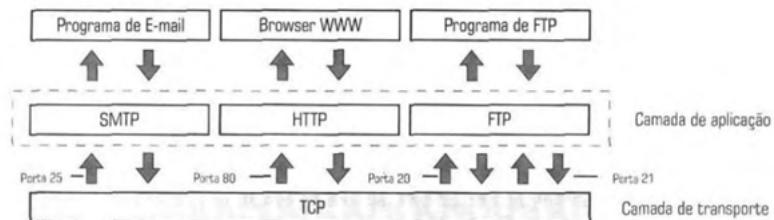


Figura 2.4: Funcionamento da camada de aplicação. Fonte: [Torres2001]

2.2.4.2 Protocolos da camada de aplicação

Nesta seção estudaremos os principais protocolos usados na comunicação das aplicações com a camada de transporte.

2.2.4.3 DNS(*Domain Name System*)

Vimos que todas às máquinas em uma rede TCP/ IP possui um endereço IP. Acontece que os endereços IP não são tão fáceis de ser recordados quanto nomes. Por isso, foi criado o sistema DNS, que permite dar nome a endereços IP, facilitando a localização de máquinas por nós, humanos. Quando você entra num endereço em um browser Internet, o browser se comunica com um servidor DNS, que é responsável por descobrir o endereço IP do nome digitado, permitindo que a conexão seja efectuada. [Torres2001]

Dessa forma, os servidores DNS possuem duas funções: converter endereços nominais em endereços IP e vice-versa.

2.2.4.4 FTP (*File Transfer Protocol*)

Como o próprio nome sugere, o FTP é um protocolo usado na transferência de arquivos. Esse protocolo utiliza duas portas para se comunicar com o TCP: 21, por onde circulam informações de controle (por exemplo, o nome do arquivo a ser transferido) e 20, por onde circulam os dados. Na figura 2.5 mostra-se um esquema simplificado do funcionamento do FTP.

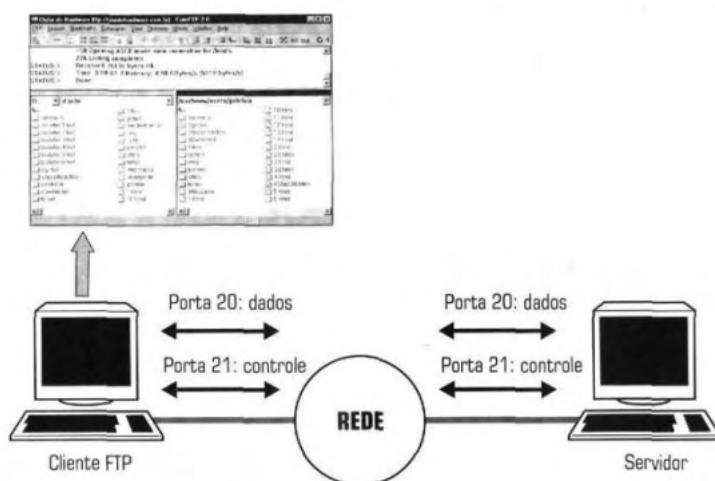


Figura 2.5: funcionamento do FTP. Fonte: [Torres2001]

2.2.4.5 HTTP (*Hyper Text Transfer Protocol*)

A transferência de documentos hipermédia é feita através do protocolo HTTP. Um servidor www hospeda o site, enquanto um cliente faz a requisição dos documentos lá contidos. Essa transferência é feita usando a porta 80 do protocolo TCP. [Torres2001]

2.2.4.6 Camada de Transporte

A camada de transporte do TCP/ IP é equivalente direto da camada de transporte (camada 4) do modelo OSI. Esta camada é responsável por pegar os dados enviados pela camada de aplicação e transformá-los em pacotes, a serem repassados para a camada de rede. [Torres2001]

No protocolo TCP/ IP a camada de transporte utiliza um esquema de multiplexação, onde

é possível transmitir “simultaneamente” dados das mais diferentes aplicações. Na verdade, ocorre o conceito de intercalamento de pacotes. Vários programas poderão estar comunicando-se com a rede de forma intercalada, não sendo preciso terminar um tipo de aplicação de rede para então começar outra. Isto é possível graças ao uso do conceito de portas, explicado no tópico 2.2.4.1, já que dentro do pacote há informação da porta de origem e de destino do dado. [Torres2001]

Em outras palavras em uma mesma sequencia de pacotes recebidos, as informações podem não ser da mesma aplicação.

Nesta camada operam dois protocolos: TCP e o Protocolo de Datagramas de Utilizador - *User Datagram Protocol* (UDP). Ao contrário do TCP, este segundo protocolo não verifica se o dado chegou ou não ao destino. Por esse motivo, o protocolo mais usado na transmissão de dados é o TCP, enquanto que o UDP é tipicamente usado na transmissão de informação de controle. [Torres2001]

2.2.4.7 Camada de rede

A camada de rede do modelo TCP/ IP é equivalente a camada 3 (Rede) do modelo OSI. Há vários protocolos que podem operar nessa camada: IP, Protocolo de mensagem de controlo de Internet - *Internet Control Message Protocol* (ICMP), Protocolo de Resolução de Endereços - *Address Resolution Protocol* (ARP) e Protocolo de Resolução de Endereços Reverso - *Reverse Address Resolution Protocol* (RARP).

Na transmissão de um dado de programa, o pacote de dados recebido da camada TCP é dividido em pacotes chamados datagramas. Os datagramas são enviados para a camada de interface com a rede, onde são transmitidos pelo cabeamento da rede através de quadros. Esta camada não verifica se os datagramas chegaram ao destino, isto é feito pelo TCP.

Esta camada é responsável pelo roteamento de pacotes, isto é, adiciona ao datagrama informações sobre o caminho que ele deverá percorrer. Para entendermos mais afundo o funcionamento desta camada e dos protocolos envolvidos, devemos estudar o esquema de endereçamento usado pelas redes baseadas no protocolo TCP/ IP.

2.2.4.8 Endereçamento IP

O protocolo TCP/ IP é roteável, isto é, foi criado pensando-se na interligação de diversas redes, onde podemos ter diversos caminhos interligando o transmissor e o receptor,

culminando na rede mundial que hoje conhecemos por Internet. Por isso, ele utiliza um esquema lógico chamado endereçamento IP. Em uma rede TCP/ IP cada dispositivo conectado em rede necessita usar pelo menos um endereço IP. Esse endereço IP permite identificar o dispositivo e a rede na qual ele pertence. [Torres2001]

2.2.4.9 IPv4

O endereço IPv4 é um número de 32 bits, representado em decimal em forma de quatro octetos, separados por um ponto, no formato a. b. c. d. Assim, o menor endereço IP possível é 0.0.0.0 e o maior é 255.255.255.255. [Torres2001]

Teoricamente uma rede TCP/ IP pode ter até 256^4 endereços IP, diz-se teoricamente porque alguns endereços são reservados e não podem ser usados.

Em redes usamos somente os endereços IP das classes A, B e C. A classificação dos endereços é feita da seguinte forma:

- **Classe A:** o primeiro octeto identifica a rede e os demais três indicam a máquina. Cada endereço classe A consegue endereçar até 16.777.216 máquinas;
- **Classe B:** os dois primeiros octetos identificam a rede e os dois demais indicam a máquina. Este tipo de endereço consegue endereçar até 65.536 máquinas;
- **Classe C:** os três primeiros octetos identificam a rede, o último número indica a máquina. Com isso, consegue endereçar até 256 máquinas.

Em princípio, se a sua rede não for estar conectada na internet, você pode definir qualquer endereço IP para os dispositivos que estejam nela conectados. O problema é que mais cedo ou mais tarde surgirá a necessidade de conectar a sua rede à Internet e o conflito de endereços será inevitável, caso você tenha montado toda a sua rede baseada em endereços IP já existentes. [Torres2001]

Existem alguns endereços que são conhecidos como “endereços mágicos”, que são endereços IPs reservados para redes privadas. Assim, você pode montar a sua rede TCP/ IP baseada nesses endereços que não gerará conflito com os endereços IP da Internet, pois os roteadores da internet reconhecem esses endereços como sendo de uma rede particular e não repassam os pedidos de pacotes que façam referência a esses endereços para o resto da Internet. [Torres2001]

Os endereços mágicos são os seguintes:

- **Classe A:** 10.0.0.0 a 10.255.255.255;
- **Classe B:** 172.16.0.0 a 172.31.255.255;
- **Classe C:** 192.168.0.0 a 192.168.255.255

2.2.4.10 Mascara de rede

A mascara de rede é formada por 32 bits no mesmo formato que o endereçamento IP e cada bit 1 da mascara informa a parte do endereço IP que é usada para o endereçamento da rede e cada bit 0 informa a parte do endereço IP que é usada para o endereçamento das máquinas. Dessa forma, as mascaras padrões são:

- **Classe A:** 255.0.0.0
- **Classe B:** 255.255.0.0
- **Classe C:** 255.255.255.0

2.2.4.11 ARP (*Address Resolution Protocol*)

O protocolo ARP é responsável por fazer a conversão entre os endereços IP e os endereços - *Media Access Control* (MAC) da rede. Em uma rede grande, os pacotes TCP/IP são encaminhados até a rede de destino através dos roteadores. Atingindo a rede de destino, o protocolo ARP entra em acção para detectar o endereço da placa de rede para a qual o pacote deve ser entregue, já que no pacote há somente o endereço IP de destino e não o endereço da placa de rede.

O ARP funciona mandando primeiramente uma mensagem de broadcast para a rede perguntando, a todas as máquinas, qual responde pelo endereço IP para o qual pretende-se transmitir um pacote. Então, a máquina que corresponde a tal endereço responde, identificando-se e informando o seu endereço MAC para que a transmissão de dados entre essas máquinas possa ser estabelecida. [Torres2001]

A figura 2.6 mostra um exemplo prático do funcionamento do protocolo ARP. O primeiro micro da rede quer enviar um pacote para o endereço 200.123.123.1. Ele manda a mensagem “Quem é 200.123.123.1?” para todos os micros da rede, em broadcast. O micro que está usando este endereço responde, informando o seu endereço MAC e iniciando a comunicação entre os dois micros.

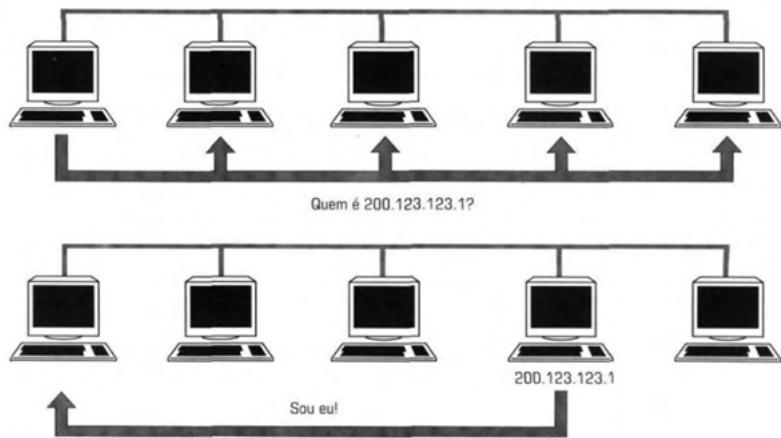


Figura 2.6: Funcionamento do ARP. Fonte [Torres2001]

2.2.4.12 Camada Enlace-física (*Camada de interface com a rede*)

Esta camada, é equivalente às camadas 1 e 2 do modelo OSI, é responsável por enviar o datagrama recebido pela camada de internet em forma de um quadro através da rede. A figura 2.7 mostra o esquema completo de um computador operando com protocolo TCP/ IP.

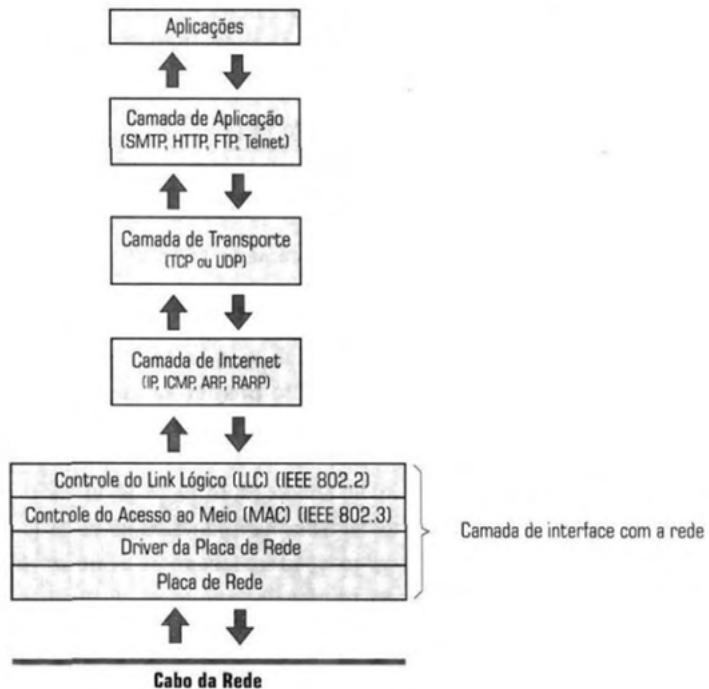


Figura 2.7: Esquema completo de um computador operando com protocolo TCP/IP. Fonte [Torres2001]

2.3 Banco de dados

Segundo [Navathe2011], banco de dados pode ser definido de diversas maneiras desde a mais genérica, como sendo, uma coleção de dados relacionados. Com dados, quer-se dizer fatos conhecidos que podem ser registrados e possuem significado implícito. E a mais restrita como representando um projecto construído e populado com dados para uma finalidade específica, possuindo um grupo específico de usuários e algumas aplicações previamente concebidas nas quais esses usuários estão interessados.

Em outras palavras, um banco de dados tem alguma fonte da qual o dado é derivado, algum grau de interação com eventos no mundo real e um público que está ativamente interessado em seu conteúdo. Um banco de dados pode ser criado e mantido manualmente ou pode ser computarizado, bem como pode ter qualquer tamanho e complexidade. [Navathe2011]

Para nós interessa-nos discutir apenas os bancos de dados computarizados.

SGBD (Database Management System) é um software de uso geral que facilita o processo de definição, construção, manipulação e compartilhamento de banco de dados entre diversos usuários e aplicações. Definir um banco de dados envolve especificar os tipos, estruturas e restrições dos dados a serem armazenados. A construção do banco de dados é o processo de armazenar os dados em algum meio controlado pelo Sistema de gerenciamento de banco de dados - *Database Management System* (SGBD). A manipulação de um banco de dados inclui funções como consulta ao banco de dados para recuperar dados específicos, actualização do banco de dados para refletir mudanças no minimundo e geração de relatórios com base nos dados. De forma a complementar as definições até aqui mencionadas [Navathe2011] chama a união do banco de dados com o software de SGBD de **sistema de banco de dados**. A figura 2.8 ilustra alguns dos conceitos discutidos até aqui.

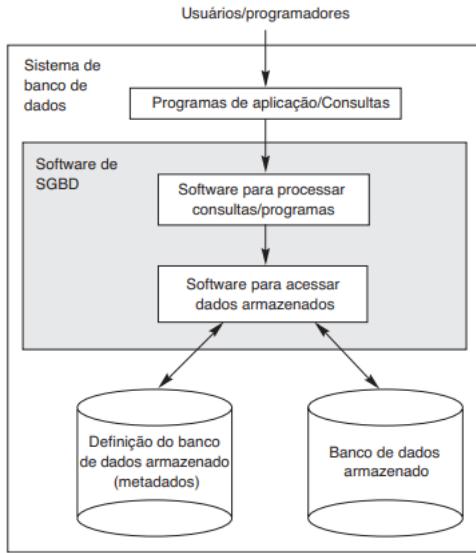


Figura 2.8: diagrama esquemático de um sistema de banco de dados.

Fonte [Navathe2011]

2.3.1 Linguagens e interfaces do banco de dados

Na secção 2.3 falamos dos SGBDs. O sistema precisa oferecer linguagens e interfaces apropriadas para os usuários. Nesta secção falaremos dos tipos de linguagens e interfaces oferecidas por um SGBD.

a. DDL (*Data Definition Language*)

É usada para definir os dois esquemas. O SGBD terá um compilador da - *Data definition Language* (DDL) cuja função é processar instruções da DDL a fim de identificar as descrições dos construtores de esquema e armazenar a descrição de esquema no catálogo do SGBD. [Navathe2011]

b. DML (*Data Manipulation Language*)

Após compilados os esquemas e o banco de dados populado, os usuários precisam de alguma forma de manipulá-lo. As manipulações típicas incluem recuperação, inserção, exclusão e modificação dos dados. E este conjunto de operações é chamada de - *Data Manipulation Language* (DML). [Navathe2011]

2.3.1.1 SQL

Na tabela 2.2, podemos ver algumas das instruções que podem ser usadas para definir e manipular as BDs.

Tabela 2.2: Instruções SQL

Palavra Chave do SQL	Descrição
DDL	
CREATE	Cria uma nova tabela
ALTER	Altera as características de uma tabela existente e de suas colunas
DML	
INSERT	Insere/acrescenta dados na tabela
SELECT	Recupera informações de uma BD
FROM	Especifica a tabela da qual obter as informações
WHERE	Condiciona/especifica/identifica as tuplas a serem recuperadas pela consulta
DELETE	Remove dados de uma tabela
UPDATE	Actualiza dados de uma tabela

2.3.2 Modelo de banco de dados relacional

O modelo relacional representa o banco de dados como uma coleção de relações. Informalmente, cada relação é semelhante a uma tabela de valores ou, até certo ponto, a um arquivo plano de registros. Ele é chamado de arquivo plano porque cada registro tem uma simples estrutura linear ou plana.

Quando uma relação é considerada uma tabela de valores, cada linha na tabela representa uma coleção de valores de dados relacionados. Uma linha representa um fato que normalmente corresponde a uma entidade ou relacionamento do mundo real. Os nomes da tabela e de coluna são usados para ajudar a interpretar o significado dos valores em cada linha. Na terminologia formal do modelo relacional, uma linha é chamada de tupla, um cabeçalho da coluna é chamado de atributo e a tabela é chamada de relação. O tipo de dado que descreve os tipos de valores que podem aparecer em cada coluna é representado por um domínio de valores possíveis. [Navathe2011]

Capítulo 3

Descrição do Hardware

Este capítulo será dedicado ao fornecimento das especificações técnicas dos materiais utilizados para o desenvolvimento do projecto, bem como o princípio de funcionamento dos mesmos.

3.1 Arduino Mega 2560

Arduino MEGA 2560 é uma placa baseada no microcontrolador ATmega2560. Esta placa possui 54 pinos de I/O digitais, das quais 15 podem ser usadas como saídas - *Pulse Width Modulation* (PWM). Possui também 16 entradas analógicas, 4 portas de comunicação serial, uma conexão - *Universal Serial Bus* (USB), um conector para alimentação externa que suporta teoricamente tensões de alimentação de entre 6V a 20V, um conector - *In-Circuit Serial Programming* (ICSP) e um botão de reset. Os 54 pinos digitais podem ser usados como entrada ou saídas usando as funções `pinMode()`, `digitalWrite()` e `digitalRead()`. Eles operam com 5V. Sendo que cada pino é capaz de fornecer ou drenar um máximo de 40 mA.

Podemos ver estas e mais descrições de pinos acerca do Arduino MEGA 2560, na figura 3.1.

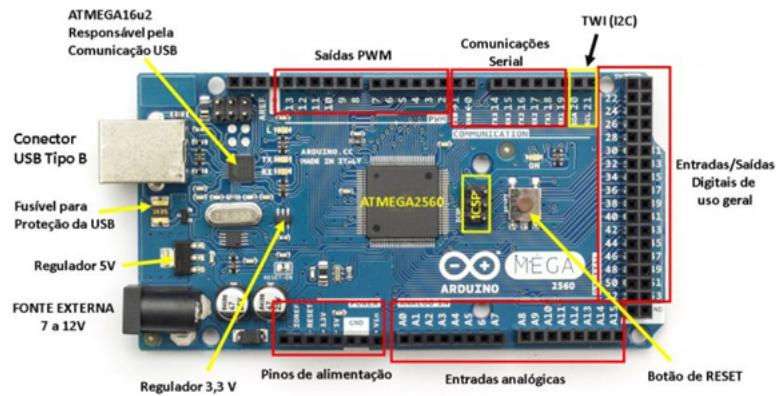


Figura 3.1: Diagrama de pinos do Arduino MEGA2560.

Fonte: <https://embarcados.com.br/arduino-mega-2560/>

A placa pode ser alimentada pela conexão USB ou por qualquer fonte de alimentação externa, conforme mencionado anteriormente. A fonte de alimentação externa pode ser de 6 a 20V e 2A no máximo. Porém se alimentada com uma tensão inferior a 7V o pino de 5V pode fornecer menos de 5V, ou seja, ficará instável e quando alimentado com tensões acima de 12V há possibilidade de danificar o regulador de tensão.

Na tabela 3.1 podemos ver uma descrição resumida dos pinos do arduino, bem como suas funções e algumas especificações.

Tabela 3.1: Descrição resumida dos pinos

Categoria	Nome	Função
alimentação	Vin, 3.3, 5V, GND	Vin: pino para alimentar dispositivos, através de uma fonte externa. 3V3: fornece tensão de 3.3 V para alimentação de dispositivos. 5V: fornece tensão de 5V para alimentação de dispositivos. GND: pino de referencia de terra
Reset	Reset	Reinicia o microcontrolador
Pinos digitais	D0 – D54	Podem ser usados como pinos de I/O, sendo que 15 são PWM
Pinos analogicos	A0 – A16	Podem ser usados como pinos de I/O analógicas na gama de 0 a 5V
SPI	50 (MISO), 51(MOSI), 52 (SCK) e 53 (SS)	São usados para comunicação SPI
TWI	20 (SDA) e 21 (SCL)	Usado para comunicação TWI
Serial	0 (RX) e 1 (TX), 19 (RX) e 18 (TX), 17 (RX) e 16 (TX), 15 (RX) e 14 (TX)	Usados para receber e transmitir dados de forma serial.
Flash Memory		256 KB
SRAM		8 KB
EEPROM		4096 bytes
Clock Speed		16 MHz

3.2 Ethernet Shield W5100

A Ethernet Shield utiliza o chip WIZnet W5100 Ethernet, este chip fornece a pilha de rede TCP/ IP, que habilita o dispositivo a se comunicar através de uma rede comum, tanto local quanto via Internet. Para o caso específico deste projecto esta placa irá possibilitar a comunicação na rede local, entre o Arduino e o servidor (Banco de dados). Basicamente ela irá desempenhar o mesmo papel que uma placa de rede executa em um computador comum.

A shield possui um conector para cabo de rede RJ45 e tem uma taxa de transferência de 10/100 Mbps. A shield incorpora ainda um slot para cartão microSD, que possibilita ler e armazenar dados na shield. A figura 3.2 é uma ilustração gráfica de uma Ethernet Shield.



Figura 3.2: Ethernet Shield.

Fonte:<https://www.robotistan.com/ethernet-shield-wiznet-w5100-for-arduino>

A Ethernet shield é compatível com as versões do Arduino mais tradicionais, tais como Duemilanove, Arduino UNO, MEGA, etc.

O Arduino comunica-se com o W5100 e o cartão microSD usando o bus - *In-System Programmer* (SPI) (através do ICSP), as portas digitais 11, 12 e 13 no Arduino UNO e 50, 51 e 52 no MEGA. Nas duas placas o pino 10 é usado para selecionar o W5100 e o pino 4 para selecionar o cartão microSD. Deste modo, todos os pinos previamente mencionados não podem ser usados como I/Os gerais. Na tabela 3.3 podemos ver as portas reservadas e as respectivas funções.

Tabela 3.2: portas reservadas

Arduino MEGA 2560	Arduino UNO	
Pino	Pino	Conexão
D52	D13	SCK
D50	D12	MISO
D51	D11	MOSI
D10	D10	W5100
D4	D4	Cartão microSD

3.2.1 Especificações

Tabela 3.3: Especificações da Ethernet Shield

Controlador	W5100
Tensão de operação	3.3 - 5 VDC
Velocidade de conexão	10/100 Mbps
Protocolos suportados	TCP/IP, UDP, ICMP, ARP, IPv4, IGMP, PPPoE, Ethernet
Memória interna	16 KB para buffer de Tx/Rx

3.3 Display LCD(*Liquid Crystal Display*) 16x2 e Módulo I2C

Este é um módulo muito utilizado como método de exibição de textos e símbolos. Ele está presente em inúmeros sistemas electrónicos como o principal elemento na sustentação da interação visual homem – máquina. O mercado conta com um acervo de Visor de cristal líquido - *Liquid Crystal Display* (LCD)s, com diversas funções, custos variados e complexidades. O modelo a ser usado no presente projecto pode ser visto na imagem 3.3.

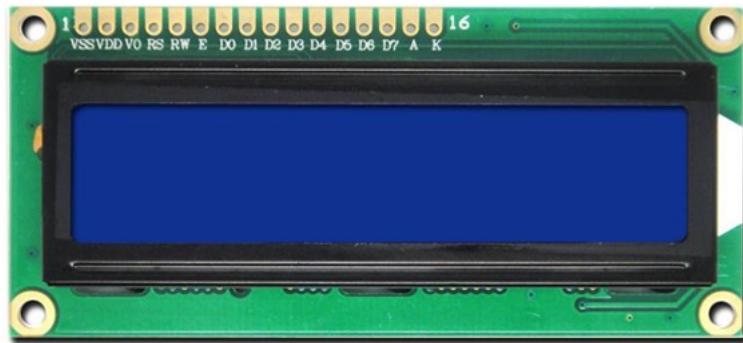


Figura 3.3: Hitachi-HD44780.

Fonte:<https://tresdprinttech.com.mx/pantallas-lcdoled/245-lcd-16x2-fondo-verde-7503040256572.html>

3.3.1 Principio de funcionamento

O mostrador é formado de duas placas acrílicas transparentes e entre elas está o cristal líquido. O cristal líquido altera o seu comportamento cristalino, dependendo da tensão aplicada entre ele. Os displays são formados de vários pontinhos designados pixéis, cada pixel pode ficar claro ou escuro dependendo da polarização de cada um. Sob as placas transparentes, existe uma matriz de conexões que controla todos os pixéis. E o responsável por essa função é o chip controlador que fica por trás do display, o HD44780 desenvolvido pela Hitachi há muito tempo atrás.[**Eletrogate**]

A integração deste com o arduino é possível e simplificada, graças ao conjunto de bibliotecas de códigos para LCDs "*LiquidCrystal.h*".

3.3.2 Diagrama de pinos

O display LCD possui 16 pinos que permitem controlar o display desde a alimentação às ações de escrita e leitura. Podemos ver na figura 3.4, os pinos que compõem este display.

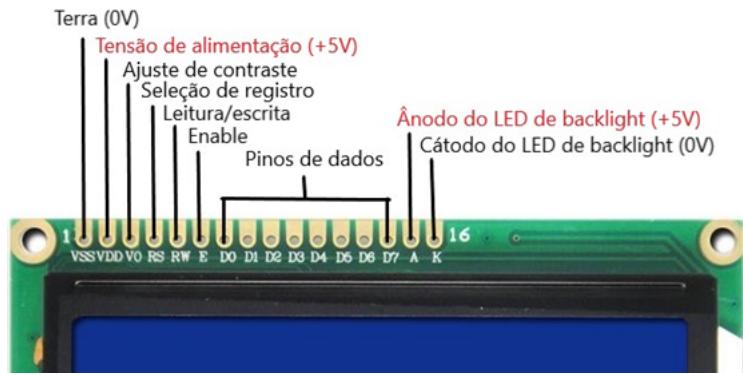


Figura 3.4: Diagrama de pinos do display. Fonte: O autor

Como podemos ver na imagem, este módulo necessita de uma quantidade relativamente grande de pinos para funcionar. De forma simplificar e compactar o projecto foi incorporado o módulo I2C ao display LCD e associada a biblioteca “*LiquidCrystal-I2C*”, que no lugar da alimentação comum do display que exigiria 16 pinos este possibilita controlá-lo com apenas 4 pinos . Na figura 3.5 podemos ver uma ilustração gráfica do módulo I2C e seu diagrama de pinos.



Figura 3.5: Módulo I2C e seu diagrama de pinos. Fonte: O autor

Da figura 3.5 podemos ver, na parte superior do módulo que ele conta com 16 pinos que irão servir para conexão do display LCD, na parte lateral direita têm-se 4 pinos, sendo que dois são para a alimentação do display (VCC e GND) e os outros dois são da interface I2C (SDA e SCL) que permitirão controlar o LCD. O jumper de Backlight na lateral esquerda permite controlar a luz de fundo pelo programa e o potenciômetro da placa é responsável por ajustar o contraste do display.

3.4 Teclado matricial

O teclado matricial é outro componente que foi usado neste projecto para sustentar a interação Homem-máquina, foi programada com o auxilio da biblioteca keypad.h para simplificar o uso do mesmo, no código. Este é um componente de entrada de dados que possui 16 teclas dispostas em 4 linhas por 4 colunas, consequentemente 8 pinos de ligação.

3.4.1 Princípio de funcionamento

Como referido anteriormente, as teclas estão dispostas em 4 linhas por 4 colunas e possui 8 pinos para ligações. Embaixo de cada tecla há um interruptor. Cada interruptor em uma linha é conectado aos interruptores da mesma linha por um condutor sob o bloco e da mesma forma são conectadas às colunas, como mostra a figura 3.6. por isso que é chamado de teclado matricial.

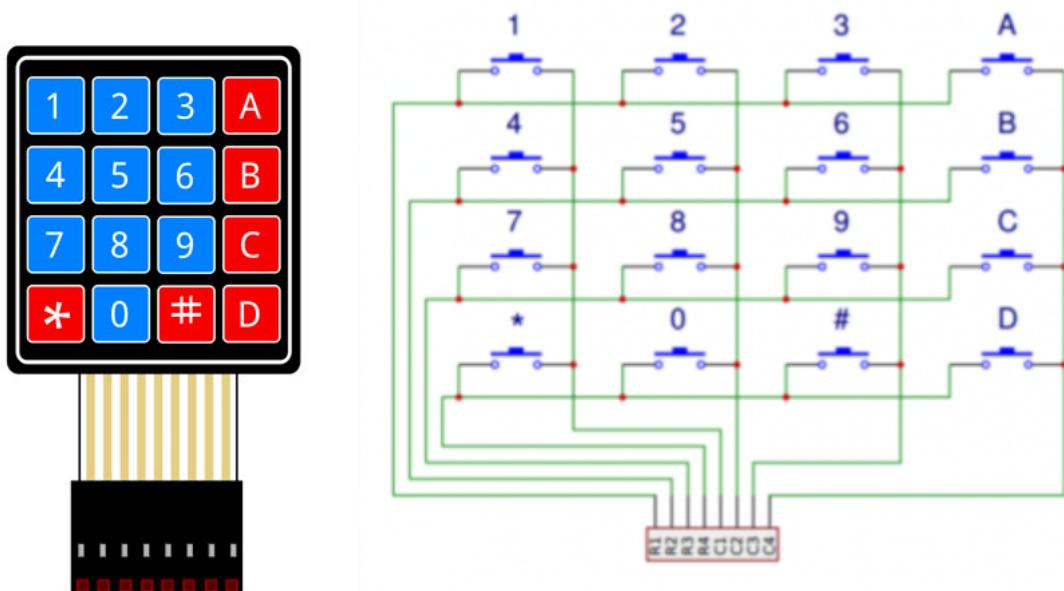


Figura 3.6: Teclado matricial e sua composição interna

Para identificar qual o botão pressionado, o microcontrolador executa quatro passos:

1. Configurar todas as colunas da matriz em nível alto (resistor de pull-up interno ativado) e todas as linhas como saídas em nível lógico baixo;
2. Caso uma tecla seja pressionada, a coluna a qual essa tecla pertence passará para nível lógico baixo;

3. De seguida o microcontrolador identifica a linha correspondente a tecla pressionada e inverte o nível lógico do passo (1);
4. Por fim o microcontrolador identificará que botão foi pressionado na linha x, coluna y, completando o processo.

3.5 Modulo GSM e regulador de tensão LM2596

O presente projecto inclui como uma de suas funcionalidades o envio de notificações de incidentes de segurança. Para tal é necessário que haja um dispositivo capaz de garantir a comunicação entre o sistema e a rede celular, um destes dispositivos é o módulo SIM800L.

O módulo SIM800L é desenvolvido pela empresa SIMcom, este módulo possibilita aos microcontroladores comunicações na rede celular para envio de mensagens de texto, dados e ligações. Na figura 3.7, podemos ver a ilustração gráfica do módulo.

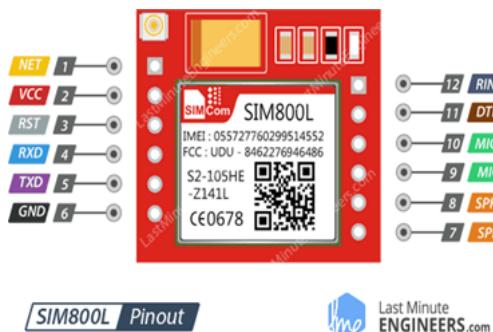


Figura 3.7: Módulo SIM800L e seu diagrama de pinos.

Fonte:<https://www.mathaelectronics.com/what-is-a-sim800l-gprs-gsm-module/>

Diagrama de pinos

Tabela 3.4: Pinos do módulo SIM800L

Pino	função
NET	Pino de conexão da antena
VCC	Pino de alimentação
RST	Pino de reset
RXD	Pino para envio de comandos ao módulo
TXD	Pino para transmitir comandos para o microcontrolador
GND	Pino terra
RING	Pino de indicação de entrada de chamada
DTR	Pino de modo de suspensão
MIC +/-	Pinos para conexão do microfone de eletreto
SPK +/-	Pinos para conexão de alto-falante

3.5.1 Especificações

- **Tecnologia:** GSM/GPRS;
- **Bandas de frequência:** quad-band 850/900/1800/1900 MHz;
- **Tensão de alimentação:** de 3.4 a 4.4 V;
- **Consumo de corrente:** em torno de 1 A durante transmissão.

Como as portas digitais do arduino funcionam a uma tensão de 5 V a alimentação deste diretamente as portas digitais pode danificá-lo. Deste modo uma alternativa para uma alimentação segura é o regulador de tensão DC-DC LM2596. Este irá permitir ajustar a tensão fornecida por uma fonte externa de 12V para 4.1V que está dentro do intervalo de tensão recomendada para a alimentação do SIM800L.

3.6 Módulo relé

Para falar do módulo relé é necessário conhecer o relé em si, daí que iniciaremos esta secção falando do relé. Neste projecto o módulo relé é usado como elemento de ligação entre o microcontrolador e a trava eléctrica.

Um relé é um dispositivo electromecânico que opera como um interruptor. Ele é usado em aplicações eléctricas/electrónicas, quando se pretende isolar dois circuitos, um de comando e outro de potência. A razão por de trás disto é a necessidade de controlar um circuito de potência com um circuito de comando que opera em tensões e correntes baixas, enquanto que os circuitos de potência são alimentados por tensões elevadas e percorridos por correntes também elevadas.

Pensando na segurança das pessoas, tanto na dos equipamentos e bem como na questão de operacionalização os dois circuitos devem estar isolados electricamente e o dispositivo responsável por isso é o relé.

3.6.1 Constituição e princípio de funcionamento

O relé é constituído pelas seguintes partes: Bobina, armadura fixa, armadura móvel, contatos (NF e NA) e uma mola de rearme. Como mostra a figura 3.8.

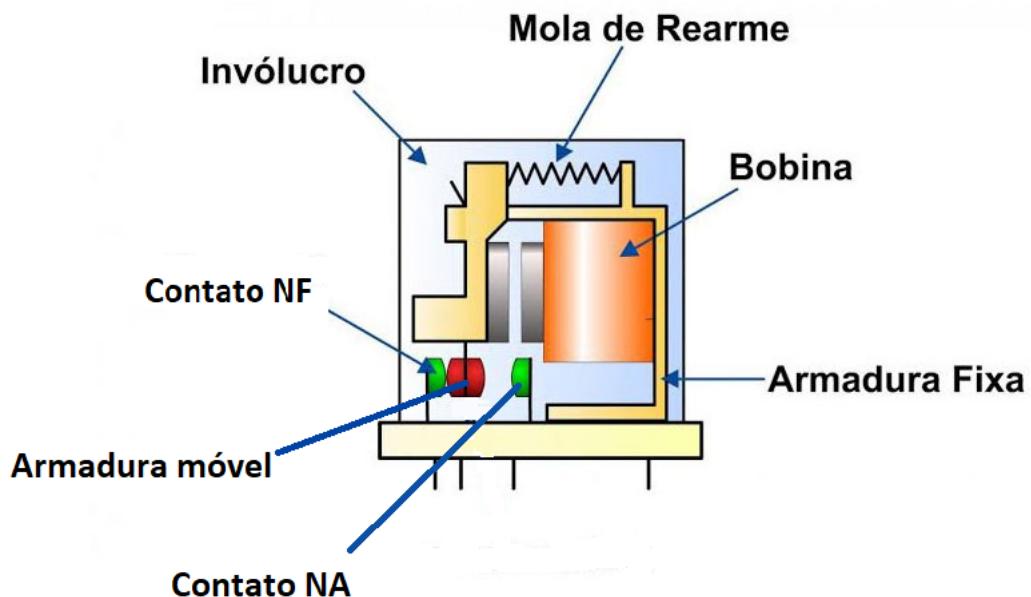


Figura 3.8: Constituição do relé. Fonte: adaptado de:<https://cromatek.com.br/reles-o-que-sao-e-para-que-servem/>

3.6.2 Princípio de funcionamento

Quando o relé é alimentado com baixa tensão é criado um campo magnético que vai exercer uma força sobre a peça móvel que dependendo da situação irá fechar, alimentando o circuito de potência com uma tensão de alimentação alta ou abrir o circuito de potência interrompendo o campo. Isto, sem haver qualquer contacto entre o circuito de comando e o de potência. Os contactos podem ser normalmente abertos ou normalmente abertos ou normalmente fechados. O contato normalmente aberto é aquele que está aberto em seu estado de repouso (sem a influência de corrente eléctrica), deste modo, quando energizado o relé o contato normalmente aberto, fecha deixando passar corrente para o circuito de potência. E a lógica inversa ocorre para o contato normalmente fechado.

3.6.3 Módulo relé

O módulo relé, segue o mesmo funcionamento que o relé convencional. A diferença entre eles é que o módulo relé inclui componentes que tornam fácil a integração do mesmo com os microcontroladores, bem como protege-lo contra sobretensões e picos de corrente. Na figura 3.9. podemos ver o circuito de um módulo relé.

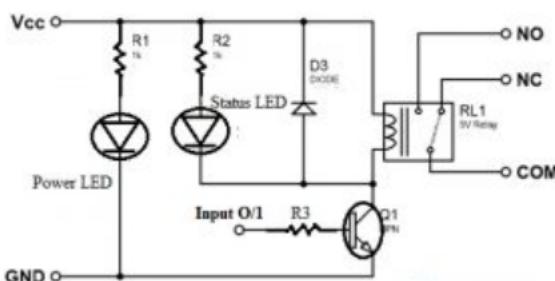


Figura 3.9: circuito interno de um módulo relé. Fonte:<https://www.robocore.net/tutoriais/introducao-ao-rele>

No circuito da figura 3.9 podemos ver que o módulo relé é activado e desactivado por um sinal digital aplicado na base do transístor Q1. Os LEDs (Power e Status) indicam que o módulo está alimentado e se algum sinal foi aplicado na entrada do módulo, respectivamente. O diodo D3 é um diodo de flyback, ele protege o circuito de comando de picos de tensão. O funcionamento do módulo resume-se basicamente na polarização do transístor para que a bobina do relé seja alimentada.

3.6.4 Especificações

- Tensão de alimentação 5VDC;
- Tensão máxima de carga 240 VAC;
- Corrente máxima de carga 2 A;
- Comutação máxima 300 operações por minuto.

3.7 Trava eléctrica (*Actuatorador solenóide linear*)

Uma solenoíde linear é um dispositivo electromagnético que converte energia eléctrica em uma força ou movimento mecânico de empurrar ou puxar. Neste projecto como o próprio nome do dispositivo sugere, ele desempenhara o papel do mecanismo que permitira travar e destravar a porta.

3.7.1 Constituição da Solenoíde

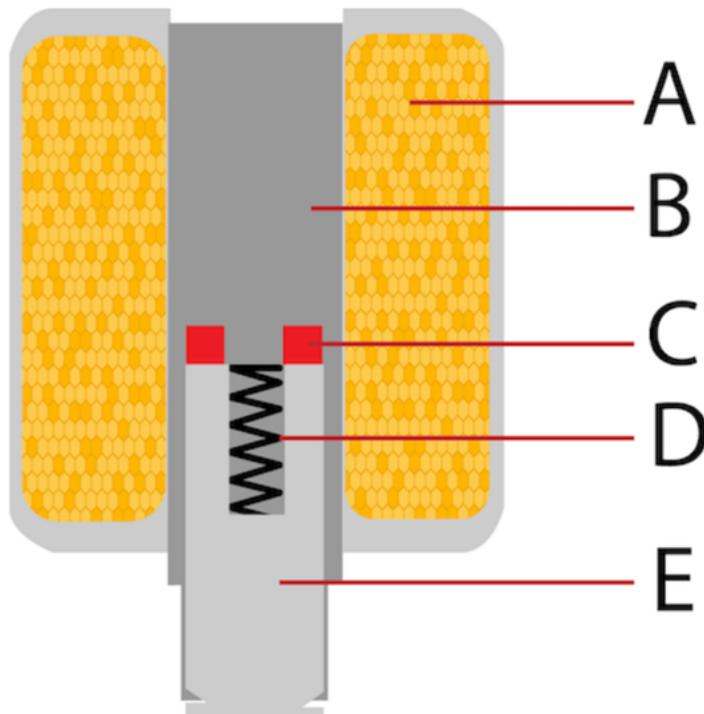


Figura 3.10: constituição de um Solenoíde. Fonte: [Willem2016]

Conforme ilustra a figura 3.10, selenoide é constituída pelas seguintes partes:

- **Bobina (A)**: é um fio de cobre enrolado firmemente em torno do núcleo estacionário;
- **Núcleo estacionário (B)**: trata-se de um cilindro ferromagnético;
- **Anel de sombreamento (C)**: também conhecida como bobina de sombreamento, é uma única volta ou algumas voltas de um condutor eléctrico (Cobre ou alumínio);
- **Mola (D)**: É uma mola de aço inoxidável que retorna a armadura a sua posição normal quando a bobina é desenergizada;
- **Armadura (E)**: Parte do núcleo estacionário que se move quando a bobina é energizada, também chamada de êmbolo selonóide.

3.7.2 Princípio de funcionamento

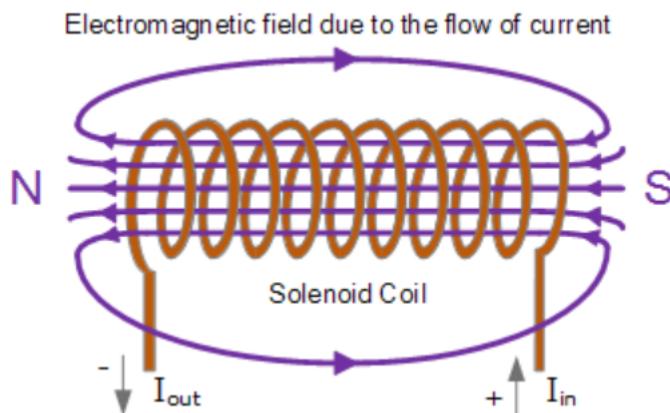


Figura 3.11: Campo gerado por um bobina percorrida por corrente. Fonte: [Willem2016]

Um campo magnético se forma quando a corrente em um selenoide flui através da bobina. A intensidade do campo é diretamente proporcional a corrente, ao número de enrolamentos e a permeabilidade do material ferromagnético do núcleo estacionário. O núcleo atua como um caminho fechado que confina o campo magnético. [Willem2016]

O campo magnético gerado pela bobina é dado pela equação 3.1

$$B = \frac{\mu I N}{L} \quad (3.1)$$

Onde:

μ é permeabilidade magnética do material;

I é a corrente que flui pela bobina;

N é o número de enrolamentos;

L é o comprimento do fio;

o campo magnético induz uma força na armadura que puxa para cima ou a empurra para baixo. A norma é puxar a armadura para cima, mas estender a lateral da armadura perto do núcleo estacionário com uma haste resulta no campo empurrando a armadura para baixo. Em qualquer cenário, uma mola é comprimida. A armadura permanece na posição enquanto o campo permanecer. Quando o campo se dissipar, a mola retorna a armadura a sua posição original.

E por fim, o anel de sombreamento fornece um caminho de baixa impedância para um pico de alta tensão gerado quando o campo magnético se dissipar. Isso diminui a magnitude e a duração do pico de tensão, o que protege o circuito. [Willem2016]

3.7.3 Especificações

- **Tensão de alimentação:** 12 VDC;
- **Corrente:** 650 mA;
- **tempo de ativação:** 1 - 10 segundos.

Capítulo 4

Desenvolvimento e descrição do protótipo

Este capítulo é dedicado a apresentação da descrição técnica do sistema, bem como a construção e as etapas empregues para a concepção do sistema.

4.1 Visão geral

O diagrama de blocos da figura 4.1 podemos ter uma visão geral por meio de um diagrama de blocos do sistema de controle de acesso. Este diagrama ilustra as comunicações, bem como as interfaces de comunicação utilizadas para que os demais dispositivos se comuniquem com o microcontrolador (Arduino MEGA 2560). Deste diagrama pode-se observar:

A Ethernet Shield utiliza o barramento SPI para se comunicar com o Arduino MEGA 2560.

O display LCD 16x2 utiliza a interface I2C através do módulo I2C para se comunicar com o Arduino MEGA 2560.

O SIM800L utiliza a interface de comunicação serial para se comunicar com o Arduino MEGA 2560.

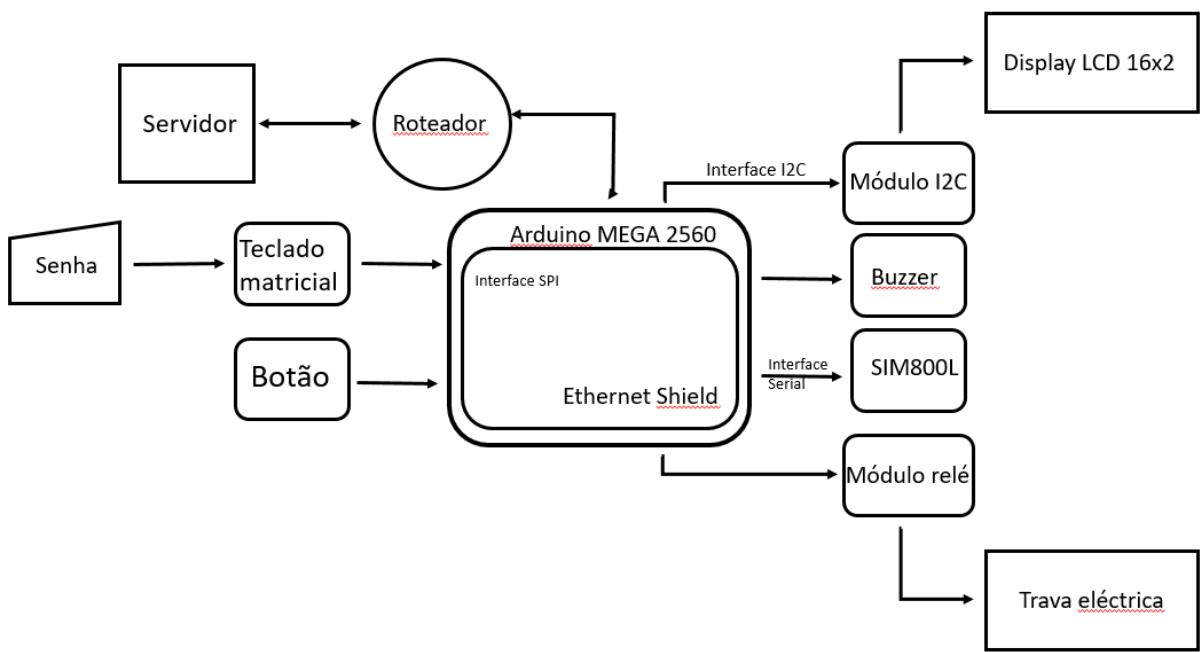


Figura 4.1: Diagrama de blocos do sistema. Fonte: O autor

O desenvolvimento e implementação do sistema envolveu 3 etapas, nomeadamente:

1. Aquisição e montagem do hardware;
2. Desenvolvimento do software de operacionalização do hardware;
3. Definição e integração da base de dados ao sistema.

4.2 Requisitos do sistema

Este sistema foi concebido com a finalidade de acima de tudo prover segurança ao local onde estiver instalado, deste modo podemos especificar os requisitos do sistema tendo em vista a categorização dos mesmos como gerais e específicos. Os requisitos do sistema são nomeados abaixo:

4.2.1 Requisitos gerais:

- Travar a porta, até que se exija que o sistema destrave a porta;
- Destrarvar a porta, somente em situações de devida verificação de integridade (pressionar o botão para sair, entrada com senha devidamente autorizada);

- Possuir conectividade com uma base de dados que permita registro dos usuários e as respectivas senhas, bem como registro dos acessos.

4.2.2 Requisitos específicos:

- Emitir bips que sinalizem a inserção da senha através de um buzzer;
- Emitir bips que sinalizem a verificação da senha (se é uma senha que consta na base de dados ou uma senha incorreta);
- Emitir mensagens que tornem comoda a interação Homem-máquina através de um display LCD;
- Enviar mensagens de notificação de incidentes de segurança.

4.3 Descrição funcional do sistema

O sistema de controle de acesso, é uma solução proposta para resolver os problemas colocados em 1.1 e consequentemente atender aos requisitos definidos em 4.2. Ele é um sistema que visa oferecer segurança nos ambientes onde for instalado, porém aliado a isto com algumas actualizações de funcionalidades é capaz de prover outros mais benefícios como: relatórios de acessos (diários, semanais, etc), controle remoto, e outros. O protótipo foi construído e testado com foco em um ambiente residencial e com auxilio uma base de dados, o sistema opera da seguinte maneira:

4.3.1 Para egressar da residência:

1. O usuário por meio de um botão, pressiona-o requisitando que a trava seja destrancada;
2. O microcontrolador recebe a requisição e destrava a tranca;
3. Destravada a tranca o microcontrolador informa ao usuário por meio do display que deve fechar a porta e pressionar a tecla “C” no teclado matricial para confirmar que fechou a porta;
4. Volvida a etapa 3 o microcontrolador inicia um temporizador de 30s, se este temporizador terminar a contagem sem que a porta seja fechada, o microcontrolador

aciona um buzzer para alertar ao usuário que se esqueceu de fechar a porta. O buzzer só cessará o alarme quando a porta for fechada.

4.3.2 Para acessar a residência:

1. O usuário deve por meio do teclado matricial pressionar a tecla “A” para que seja autorizado a digitar a sua senha de entrada;
2. Ao pressionar a tecla “A”, sistema entende que encontra-se um usuário que pretende ser autorizado a acessar a residência e inicia a rotina de autorização;
3. Esta rotina inicia com o microcontrolador informando ao usuário para que digite a sua senha por meio do display LCD, emitindo a mensagem: “Palavra-passe:”;
4. O usuário digita a senha por meio do teclado matricial e confirma-a ao microcontrolador pressionando a tecla “cardinal”;
5. Pressionada a tecla “cardinal” o sistema, consulta a base de dados para verificar se existe alguma senha igual a senha digitada. Caso a senha confira com alguma registada na BD:
 - a) O sistema, destrava a tranca;
 - b) Emite no display a mensagem: “Bem-vindo, nome do usuário associado a senha digitada”;
 - c) E por fim regista o acesso feito, na base de dados com nome Histórico. Isto é feito registando o nome associado a senha digitada e a data e a hora de acesso;
6. Caso a senha não confira com alguma registada na BD:
 - a) descriptionO sistema emite no display a mensagem: “Senha incorreta”;
 - b) Se esta situação ocorrer 3 vezes seguidas o sistema envia uma notificação via sms ao ADMIN informando sobre o incidente de segurança.

Podemos ver na tabela x os componentes que compõem o sistema e suas respectivas funções no sistema.

Tabela 4.1: Componentes que compõem o sistema e as respectivas funções

Componente	Função no sistema
Arduino MEGA 2560	Elemento responsável por: Actuar como controlador do sistema; Fazer a conexão entre os demais componentes, tornando-os um sistema.
Ethernet Shield W5100	É responsável por permitir que o microcontrolador se conecte a um rede local, desse modo garantindo o acesso a base de dados hospedada em um servidor na mesma rede
Display LCD 16x2	Garante a interface visual Homem-Máquina
Teclado matricial 4x4	Elemento que proporciona interface Homem-máquina permitindo: Introdução da senha; Fechar a porta do lado de fora.
LED azul	Proporciona interface de áudio Homem-máquina responsável por: Emitir bips de sinalização, sinalizando alguma tecla pressionada; Emitir bips de sinalização, sinalizando abertura da trava; Emitir bips de sinalização, sinalizando senha incorreta; Emitir bips de sinalização, sinalizando esquecimento de fecho da porta.
Módulo relé 5VDC	Elemento responsável por: Garantir a conexão segura entre o microcontrolador e a trava eléctrica; Isolar o sistema de comando (Arduino) do sistema de potência (Trava eléctrica), evitando que ocasionais picos de tensão danifiquem o circuito de comando.
Trava eléctrica	É responsável por travar e destravar a porta.

Módulo GSM SIM800L	Módulo responsável conectar o sistema a rede móvel permitindo que este envie notificações de incidentes de segurança via sms.
LM2596 HVS DC-DC	Módulo regulador de tensão ajustável que permite alimentar o módulo SIM800L com uma tensão de 4.1V através de uma fonte de tensão 12 VDC
Roteador Tenda 300 Mbps Wireless N ADSL 2	Elemento responsável por implementar a LAN para a comunicação entre o sistema e o servidor
Conektor RJ45	Conecta a Ethernet shield e o hospedeiro do servidor ao roteador
Jumpers	Fios condutores que ligam diferentes pontos do circuito entre si

Capítulo 5

Ensaios e resultados

Afim de comprovar o funcionamento do sistema, o protótipo foi testado em ambiente de bancada. Foram efectuadas a ligações necessárias e a ilustração gráfica do protótipo ligado pode ser vista na figura 5.1.

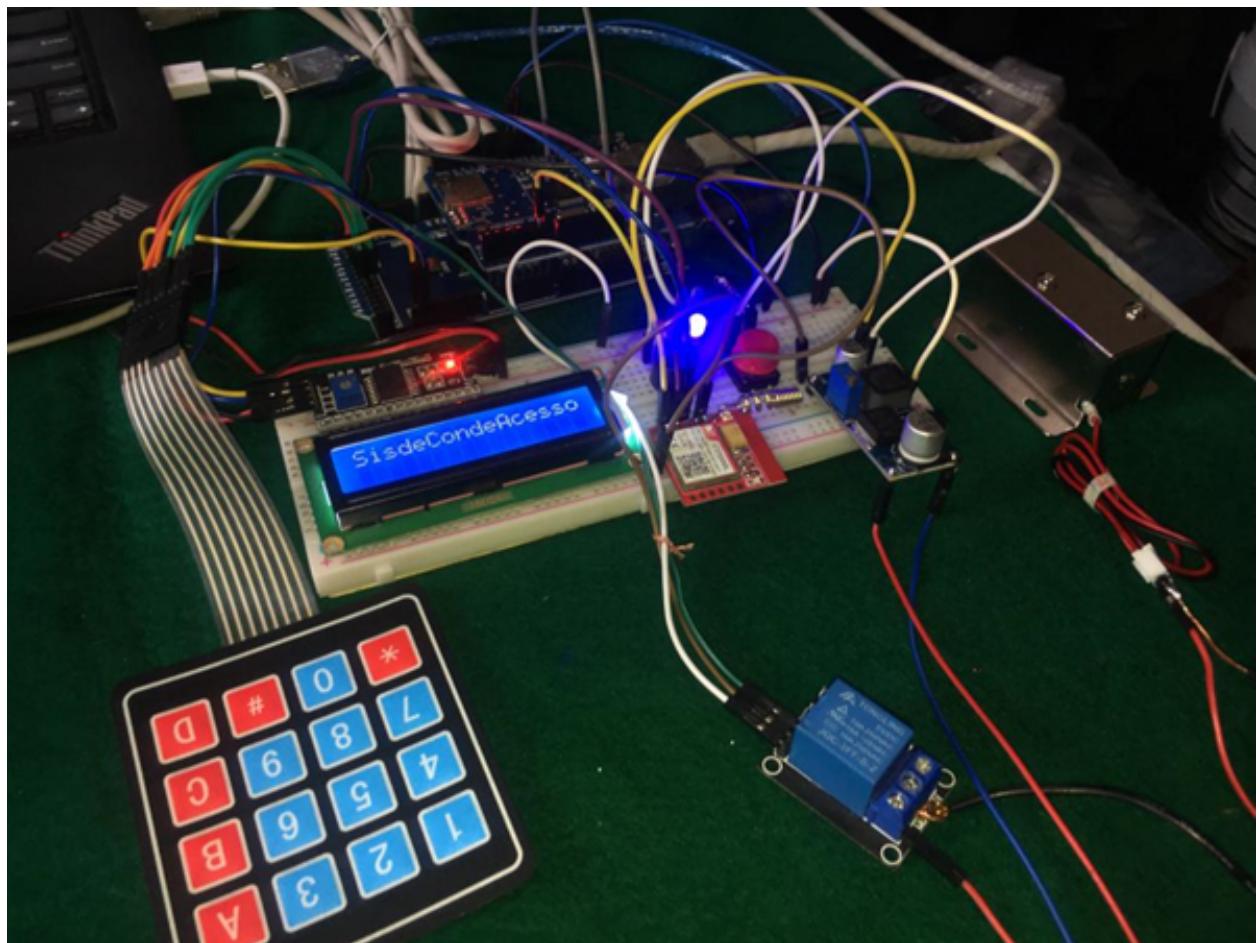


Figura 5.1: Ilustração gráfica do hardware. Fonte: Autor

Por meio do LED azul e mensagens enviadas ao serial pelo Arduino verifica-se se a conexão entre o sistema e a BD foi estabelecida. E o resultado desta consulta pode ser vista na seção 5.0.1.

5.0.1 Resultados da conexão com a base de dados

Em caso de a conexão ser estabelecida com sucesso, a seguinte informação é impressa no monitor serial:

```
Output Serial Monitor X
Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM4')
11:30:35.224 -> ...trying...
11:30:35.742 -> Connected to server version 8.0.26
```

Figura 5.2: Mensagem de confirmação de conexão com a base de dados. Fonte: Autor

E a mensagem vista na figura 5.3, é impressa caso contrário

```
Output Serial Monitor X
Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM4')
11:34:41.931 -> ...trying...
11:34:42.929 -> ...got: 0 retrying...
11:34:43.431 -> ...trying...
11:34:44.440 -> ...got: 0 retrying...
11:34:44.930 -> ...trying...
11:34:45.939 -> ...got: 0 retrying...
11:34:46.501 -> Erro: Falha na conexão com o banco de dados
```

Figura 5.3: Mensagem de falha de conexão com a base de dados. Fonte: Autor

Digitada a senha, a query (1) de pesquisa é impressa no mesmo instante em que a consulta a base de dados é efectuada, caso a senha confira com alguma senha que conste na base de dados o usuário é autorizado e por fim é registada na base de dados o acesso por meio da query (2) da figura 5.4. o resultado da inserção pode ser visto na figura abaixo, marcado por uma seta.

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM4')

```

11:38:25.340 -> ...trying...
11:38:25.856 -> Connected to server version 8.0.26
11:38:35.992 ->
11:38:35.992 -> SELECT NOME, PASSWORD FROM ARDUINO.USUARIOS WHERE PASSWORD = '20002'; (1)
11:38:36.037 ->
11:38:36.037 ->
11:38:36.568 -> INSERT INTO ARDUINO.HISTORICO (nome, datahora) VALUES ('Macome', now()); (2)

```

	nome	datahora
	Macome	2023-11-03 17:16:17
	Macome	2023-11-03 17:18:34
	Macome	2023-11-03 17:20:11
	Macome	2023-11-03 17:21:04
	Cleuma	2023-11-07 10:57:22
	Cleuma	2023-11-07 11:01:38
	Cleuma	2023-11-07 11:38:16
	Macome	2023-11-07 11:38:36

Figura 5.4: Mensagem de consulta bem sucedida e registro de histórico

Caso a senha não confira com nenhuma senha que conste na base de dados a mensagens da figura 5.5 é impressa.

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM4')

```

11:46:22.422 -> ...trying...
11:46:22.958 -> Connected to server version 8.0.26
11:46:32.304 -> 20008
11:46:32.304 -> SELECT NOME, PASSWORD FROM ARDUINO.USUARIOS WHERE PASSWORD = '20008';
11:46:32.304 -> Erro: Falha ao consultar o banco de dados

```

Figura 5.5: Mensagem de consulta mal sucedida. Fonte: Autor

5.1 Custos do projecto

Nesta secção apresentamos o levantamento dos custos para a realização do projecto e a estimativa do valor final do dispositivo. Na tabela 5.1 encontram-se os custos para a realização do projecto.

Tabela 5.1: Levantamento dos custos para realização do projecto

Item	Qtd	Preço por unidade(MZN)
Arduino Mega 2560	1	1 250.00
Ethernet Shield	1	1 500.00
Display LCD 16x2	1	400.00
Módulo I2C	1	250.00
Trava eléctrica	1	1 400.00
Módulo relé	1	250.00
Buzzer activo	1	60.00
Push buttons	1	24.00
LED	1	10.00
Resistências 10k	2	10.00
Módulo SIM800L	1	1 200.00
Módulo LM2596	1	350.00
Conector RJ45	2	291.40
Jumpers	3	100.00
Fonte de alimentação 12VDC	3	500.00
Sub-total		10 346.80
Mão de obra		11 381.48
Total		21 728.28

O custo total da construção do sistema, sem incluir a mão de obra é estimado em dez mil trezentos e cinquenta meticais (10350). Este valor pode reduzir a metade ou mais do que a metada, pelas seguintes razões:

Os dispositivos são comprados a atacado e nestas condições eles acabam tendo o seu valor elevado em relação a compra em quantidades.

Alguns dispositivos foram adquiridos em formato de plug-and-go o que também agrava o seu preço diferente de os adquirir na forma de microprocessadores simples. O caso do Arduino MEGA 2560 (pode-se adquirir somente o microprocessador ATmega 2560) ou a Ethernet Shield (o caso do Ethernet chip W5100 da Wiznet).

Capítulo 6

Conclusões e recomendações

6.1 Conclusões

Concluindo, constata-se que o sistema cumpre os requisitos de funcionalidades pré-definidas, e é capaz de travar e destravar a trava eléctrica e ainda em situação de falta de energia eléctrica em momento algum a trava será acionada, garantindo a segurança do local em que for instalado.

Verificou-se ainda, que interface entre os elementos da rede por meio de rede Ethernet mostra-se ser bem eficaz, apesar de o avanço das tecnologias estar cada vez mais virada para interfaces sem fios.

Foi também possível observar a quando da introdução do módulo GSM ao projecto, que este é na mesma proporção um elemento que abre um horizonte de várias possibilidades de melhoria ao sistema, mas também é um catalisador de inúmeros problemas, pelas seguintes razões verificadas ao longo do trabalho com o mesmo: O SIM800L é um módulo que é importado e vendido nacionalmente por entusiastas da área de tecnologias, estes raramente têm como submeter estes dispositivos a testes de qualidade, fazendo com que alguns deles sejam vendidos com defeitos, comprometendo o bom funcionamento do mesmo. O SIM800L é um módulo muito compacto e isto torna-o complexo ou quase que impossível de reparar o que mais uma vez torna a experiência de uso do dispositivo. Ensuma, o módulo SIM800L é eficaz, porém com uma experiência de utilização relativamente má.

Encerra-se a etapa conclusiva afirmando que os objectivos essenciais do trabalho foram integralmente cumpridos e as comprovações factuais do sucesso da realização dos mesmos, bem como os procedimentos levados acabo para tal encontram-se documentados

no presente relatório.

6.2 Recomendações

As limitações de tempo e custos impediram que o autor, incorpora-se no projecto as funcionalidades que permitissem o controle remoto do dispositivo para: adicionar usuários remotamente, definir horários de entrada e saída para determinados usuários. Assim recomenda-se que futuros trabalhos sejam empregues de modo a realizar estas melhorias que tornarão ainda mais comodo o uso deste dispositivo.

O Latex mostrou ser uma ferramenta de edição de texto muito potente, porém exigente no que diz respeito a curva de aprendizagem. Assim sendo recomenda-se que o contato com a linguagem e o software seja feita mais cedo na jornada académica, de maneira a familiarizar o estudante para posterior uso na culminação dos estudos

Bibliografia

- [1] Popovic, Miroslav. (2006). *Communication protocol engineering*. (1^a ed). Vol 1. CRC Press. ISBN: 0849398142
- [2] Kozierok, M, Charles. (2005). *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*. 1^a ed. No Starch press. ISBN: 159327047X
- [3] Behrouz, A, Forouzan. (2010). *Comunicação de dados e redes de computadores*. 4^a ed. AMGH Editora Ltda. ISBN: 978-85-63308-47-4
- [4] De Oliveira, Sérgio. (2017). *Internet das coisas com ESP8266, arduino e Raspberry Pi*. 1^a ed. Novatec. ISBN: 978-85-7522-582-0
- [5] Torres, Gabriel. (2001). *Redes de Computadores Curso Completo*. Axcel Books. ISBN: 85-7323-144-0
- [6] Jaime, C, Gerson. (2022). *Supervisão e Gestão Remota de uma Rede de Geradores de Energia Elétrica com recurso a IOT (Caso de Estudo: Gerador Diesel Himoinsa CEA7)*. Universidade Eduardo Mondlane.
- [7] Navathe, Elmasri. (2011). *Sistemas de banco de dados. Traduzido de (Fundamentals of Database Systems)* 6^a ed. Person Education. ISBN:978-85-4301-381-7
- [8] Pustjens, J, Willem. (2016). *Compreendendo o design e a função do solenoide*. TAMESON. Disponível em: <https://tameson.com/pages/solenoid>. Acesso em: 27/10/2023
- [9] Dos Reis, Fábio. (2018). *Como usar um Shield Ethernet no Arduino*. <http://www.bosontreinamentos.com.br/electronica/arduino/como-usar-um-shield-ethernet-no-arduino>. Acesso em: 27/10/2023
- [10] Arduino Stack Exchange. (2021). *Arduino Ethernet Usage*. Disponível em: 48

<https://arduino.stackexchange.com/questions/33019/arduino-ethernet-shield-on-arduino-mega-pin-usage>. Acesso em: 29/10/2023

- [11] Eletrogate. (2023). *Guia Completo do Display LCD – Arduino*. Disponível em: <https://blog.eletrogate.com/guia-completo-do-display-lcd-arduino/> . Acesso em: 29/10/2023

Anexos


```

//-----
//Banco de dados
#include <Ethernet.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>

byte mac_addr[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 137, 2); // IP da Ethernet Shield

IPAddress server_addr(192, 168, 137, 1); // IP do MySQL
char user[] = "macome"; // username de login no MySQL
char password[] = "Parmalate1"; // password do usuário no MySQL

EthernetClient client;
MySQL_Connection conn((Client*)&client);
MySQL_Cursor* cur_mem = new MySQL_Cursor(&conn);
//-----
//Querys(instrucoes) para manipular a BD
char query_1[100] = "SELECT NOME, PASSWORD FROM ARDUINO.USUARIOS WHERE
PASSWORD = '%s';";
//char query_2 [180] = "INSERT INTO ARDUINO.HISTORICO (nome, datahora) VALUES
('%s', now());";
char query[200] = "";
void (*softreset)() = 0;
//-----
//Definindo os pinos para a trava, o buzzer e o LED
#define trava 38
#define buzzer 6
#define LED 7
#define botaoD 39

//-----
int estadoporta = 1; //zero = aberta e um = fechada;
int estadobotaoD;
int contachances = 0;
unsigned long tempoInicial;
unsigned long tempoAnterior = 0;
unsigned long intervalo = 1000;
unsigned long tempoDecorrido;

bool temSMS = false;
String telefoneSMS;
String dataHoraSMS;
String mensagemSMS;
String comandoGSM = "";

void setup() {

```

```

Serial.begin(115200);
Serial1.begin(115200);
pinMode(trava, OUTPUT);
pinMode(buzzer, OUTPUT);
pinMode(LED, OUTPUT);
pinMode(botaoD, INPUT);
tempoInicial = millis();
lcd.init();
lcd.backlight();
lcd.blink();
conectar();
enviarcomando("AT+CMGF=1");
enviarcomando("AT+CNMI=1,2,0,0,0");

lcd.print("SisdeCondeAcesso");
}

void loop() {

if (portaFechada()) {
    estadobotaoD = botaoApertado();
    if (estadobotaoD == 1) {
        estadolaporta = 0;
        sair();
    }
    char tecla_pressionada = teclado.getKey();
    if (tecla_pressionada == 'A') {
        modoAbrir();
        portaFechada();
    }
}
}

leGSM();

if (temSMS) {

//Serial.println("Chegou Mensagem!!!");
//Serial.println();

//Serial.print("Remetente: ");
//Serial.println(telefoneSMS);
//Serial.println();

//Serial.print("Data/Hora: ");
//Serial.println(dataHoraSMS);
//Serial.println();

//Serial.println("Mensagem:");
}

```

```

//Serial.println(mensagemSMS);
//Serial.println();

mensagemSMS.trim();
if (mensagemSMS == "estadoporta") {
    Serial.println("Enviando estado da porta.");
    if (estadoporta == 1) {
        enviaAlerta("+258824096696", "A porta esta fechada");
    }
    if (estadoporta == 0) {
        enviaAlerta("+258824096696", "A porta esta Aberta");
    }
}
temSMS = false;
}

//-----
//funcoes que auxiliam o programa
void sair() {
    digitalWrite(trava, HIGH);
    delay(3000);
    digitalWrite(trava, LOW);

    mensagemaosair();
}

void mensagemaosair() {
    lcd.clear();
    lcd.print("Porta Aberta");
    lcd.setCursor(0, 1);
    lcd.print("Feche a porta");
    fecharaosair();
}

void fecharaosair() {
    char tecla1;
    int x = 0;
    while (tecla1 != 'C') {
        //tempoInicial = millis();
        tempoDecorrido = millis() - tempoInicial;
        //Serial.println(tempoDecorrido);
        if (tempoDecorrido - tempoAnterior >= intervalo) {
            tempoAnterior = tempoDecorrido;
            digitalWrite(LED, !digitalRead(LED));
        }
        tecla1 = teclado.getKey();
        if (tecla1 == 'C') {
}
}
}

```

```

        digitalWrite(buzzer, LOW);
        tempoInicial = millis();
        tempoAnterior = 0;
        estadoporta = 1;
        infoFechada();
        portaFechada();
        return;
    }

    if (tempoDecorrido >= 30000 && !portaFechada()) {
        digitalWrite(buzzer, HIGH);
        delay(500);
        digitalWrite(buzzer, LOW);
    }
}

bool portaFechada() {
    if (estadoporta == 1) {
        //infoFechada();
        return true;
    } else {

        return false;
    }
}

void infoFechada() {
    lcd.clear();
    lcd.print("PortaFechada");
    delay(2000);
    lcd.clear();
    lcd.print("SisdeCondeAcesso");
    return;
}
}

byte botaoApertado() {
    byte botaoApertado;
    byte estado = 0;
    botaoApertado = digitalRead(botaoD);
    if (botaoApertado == HIGH) {
        estado = 1;
    }
    return estado;
}
//-----

```

```

void modoAbrir() {
    lcd.clear();
    introSenha();
}

void introSenha() {
    char tecla;
    int cont = 0;
    lcd.print("Palavra-passe:");
    lcd.setCursor(0, 1);
    //memset(senhaDigitada, 0, sizeof(senhaDigitada));

    while (cont < 6) {
        tecla = teclado.getKey();
        if (tecla != NO_KEY && tecla != '#') {
            lcd.print('*');
            piscaledebuzzer(1, 100);
            senhaDigitada[cont] = tecla;
            cont++;
        } else if (tecla == '#') {
            //Serial.println(get_free_memory());
            //Serial.println(senhaDigitada);
            break;
        }
    }

    senhaDigitada[cont] = '\0';
    if (consultadb()) {
        comparaSenha();

    } else {
        //Serial.println("Senha incorreta");
        lcd.clear();
        lcd.print("Senha incorreta");
        piscaledebuzzer(5, 20);
        delay(2000);
        lcd.clear();
        lcd.print("SisdeCondeAcesso");
        contachances++;
        //Serial.println(contachances);
        if (contachances == 3) {
            enviarcomando("AT+CMGF=1");
            enviaAlerta("+258824096696", "SisdeCondeAcesso: Ação suspeita, alguém
excedeu o limite de tentativas.");
            delay(5000);
            contachances = 0;
        }
    }
}

```

```

    }
    //delete [] senhaDigitada;
    // free(query_1);
    free(senhaDigitada[cont]);
}

bool consultadb() {

    sprintf(query, query_1, senhaDigitada);
    //Serial.println(query);
    if (cur_mem->execute(query)) {
        column_names* columns = cur_mem->get_columns();
        if (columns) {
            row_values* row = cur_mem->get_next_row();
            if (row) {
                strcpy(nome, row->values[0]);
                strcpy(senhaBD, row->values[1]);
                //free(query_1);
                //free(query);
                //free(cur_mem);
                //Serial.println(nome);
                //Serial.println(senhaBD);
                //free(row);
                //Serial.println(get_free_memory());
                return true;
            }
        }
        //delete cur_mem;
        //free(columns);
    }

    //free(query_1);
    //free(cur_mem);
    return false;
}

void comparaSenha() {
    if (strcasecmp(senhaBD, senhaDigitada) == 0) {
        contachances = 0;
        lcd.clear();
        acessopermitido();
        historico();
    } //else {
    //lcd.clear();
    //lcd.print("Senha incorreta");
    //piscaledebuzz(5, 20);
    //Serial.println("Erro: Senha incorreta");
    //contachances++;
}

```

```

//Serial.println(contachances);
//if(contachances == 3){
//enviarcomando("AT+CMGF=1");
//enviaAlerta("+258824096696", "Fulano esqueceu-se de fechar a porta");
//delay(5000);
//contachances = 0;
//}
//}
memset(senhaDigitada, 0, sizeof(senhaDigitada));
//memset(nome, 0, sizeof(nome));
//memset(senhaBD, 0, sizeof(senhaBD));
estadoporta = 1;
//delete cur_mem;
}

bool conectar() {
Ethernet.begin(mac_addr, ip);
if (conn.connect(server_addr, 3306, user, password)) {
  delay(1000);
  lcd.clear();
  lcd.print("Conectado.");
  digitalWrite(LED, HIGH);
  delay(100);
  lcd.clear();
  return true;
} else {
  lcd.print("Erro: Falha na conexão.");
  //Serial.println("Erro: Falha na conexão com o banco de dados");
  return false;
}
}

void historico() {
MySQL_Cursor* cur = new MySQL_Cursor(&conn);
char query_2[180] = "INSERT INTO ARDUINO.HISTORICO (nome, datahora) VALUES
('%s', now());\n";
char query_3[180] = "";
sprintf(query_3, query_2, nome);
//Serial.print(query_3);
cur->execute(query_3);
//free(query_1);
//free(query);
//free(cur_mem);
//delete cur;
delete[] nome;
delete[] senhaBD;
delete[] senhaDigitada;
delay(500);
}

```

```

    softreset();
}

void acessopermitido() {
    digitalWrite(buzzer, HIGH);
    lcd.print("Bem-vindo");
    lcd.setCursor(0, 1);
    lcd.print(nome);
    digitalWrite(trava, HIGH);
    delay(500);
    digitalWrite(buzzer, LOW);
    digitalWrite(trava, LOW);
}

void piscaledebuzzer(int pisca, int tempo) {
    for (int n = 1; n <= pisca; n++) {
        digitalWrite(LED, LOW);
        digitalWrite(buzzer, HIGH);
        delay(tempo);
        digitalWrite(LED, HIGH);
        digitalWrite(buzzer, LOW);
        delay(tempo * 2);
    }
}
//-----
//envio de notificacoes

void enviarcomando(const char* comando) {
    Serial.print("Comando enviado: ");
    Serial1.println(comando);
    delay(500);

    Serial.println("Resposta do SIM800L: ");
    while (Serial1.available()) {
        Serial.write(Serial1.read());
    }
}

void enviaAlerta(const char* MSISDN, const char* mensagem) {
    Serial.println("Enviando alerta...");

    String comando = "AT+CMGS=\"" + String(MSISDN) + "\"";
    enviarcomando(comando.c_str());
    delay(1000);
    Serial1.print(mensagem);
}

```

```

delay(1000);
Serial1.write(26);
Serial.println();
delay(5000);
}

void leGSM() {
    static String textoRec = "";
    static unsigned long delay1 = 0;
    static int count = 0;
    static unsigned char buffer[64];

    if (Serial1.available()) {

        while (Serial1.available()) {

            buffer[count++] = Serial1.read();
            if (count == 64) break;
        }

        textoRec += (char*)buffer;
        delay1 = millis();

        for (int i = 0; i < count; i++) {
            buffer[i] = NULL;
        }
        count = 0;
    }

    if (((millis() - delay1) > 100) && textoRec != "") {

        if (textoRec.substring(2, 7) == "+CMT:") {
            temSMS = true;
        }

        if (temSMS) {

            telefoneSMS = "";
            dataHoraSMS = "";
            mensagemSMS = "";

            byte linha = 0;
            byte aspas = 0;
            for (int nL = 1; nL < textoRec.length(); nL++) {

                if (textoRec.charAt(nL) == "'") {
                    aspas++;
                    continue;
                }
            }
        }
    }
}

```

```

    }

    if ((linha == 1) && (aspas == 1)) {
        telefoneSMS += textoRec.charAt(nL);
    }

    if ((linha == 1) && (aspas == 5)) {
        dataHoraSMS += textoRec.charAt(nL);
    }

    if (linha == 2) {
        mensagemSMS += textoRec.charAt(nL);
    }

    if (textoRec.substring(nL - 1, nL + 1) == "\r\n") {
        linha++;
    }
} else {
    comandoGSM = textoRec;
}

textoRec = "";
}
}

```