

**SISTEMA DE SEGURANÇA E ACIONAMENTO RESIDENCIAL
BASEADO EM GSM E BLUETOOTH**

Angelino Fernando Nhampossa

UNIVERSIDADE ZAMBEZE
FACULDADE DE CIÊNCIAS E TECNOLOGIAS
ENGENHARIA INFORMÁTICA

**SISTEMA DE SEGURANÇA E ACIONAMENTO RESIDENCIAL
BASEADO EM GSM E BLE**

ANGELINO FERNANDO NHAMPOSSA

BEIRA

2020

UNIVERSIDADE ZAMBEZE
FACULDADE DE CIÊNCIAS E TECNOLOGIAS
ENGENHARIA INFORMÁTICA

**SISTEMA DE SEGURANÇA E ACIONAMENTO RESIDENCIAL BASEADO
EM GSM E BLE**

ANGELINO FERNANDO NHAMPOSSA

Orientador: Eng^o Christian Coulon

Monografia científica apresentada à
Faculdade de Ciências e Tecnologia,
Universidade Zambeze como requisito
parcial para a obtenção de título de
Licenciado em Engenharia Informática.

BEIRA

2020

DECLARAÇÃO

Eu, Angelino Fernando Nhampossa declaro que esta monografia é resultado do meu próprio trabalho e está a ser submetida para a obtenção do grau de licenciatura na Universidade Zambeze, Beira. Ela não foi submetida antes para obtenção de nenhum grau ou para avaliação em nenhuma outra universidade.

Beira, aos__ de _____ de 2020

(Angelino Fernando Nhampossa)

DEDICATÓRIA

Dedico este Trabalho de Conclusão de Curso a minha família pelo apoio que tem-me dedicado, especialmente ao meu irmão Lucindo Fernando Nhampossa, a minha mãe Esperança André Matsimbe e ao meu pai Fernando Manuel Nhampossa (In memoriam), aos meus primos, e aos meus sobrinhos.

AGRADECIMENTOS

Em primeiríssimo lugar rendo graças ao meu Deus, que, pelo Seu infinito amor permitiu que o Seu único filho morresse na cruz maldita por um pobre pecador como eu, conforme diz a Bíblia em João 3:16.

Agradeço ao meu irmão, que de forma não recompensável tem envidado esforços para que a minha formação académica seja uma realidade.

Agradeço a minha mãe, mulher única e insubstituível que por meio dela Deus permitiu que o jovem Angelino contemplesse a luz do sol.

Agradeço a menina mais linda que já vi nesta vida, a minha futura esposa, que de forma tão romântica me apoiou em prontidão, animando-me à conceber este documento sacrificial.

Agradeço também aos meus irmãos, sobrinhos, primos, tios e amigos que de forma directa ou indirecta estavam sempre presentes nos momentos em que mais precisei.

De forma especial, agradeço ao meu orientador Eng^o Christian Franklin Coulon pela grande ajuda durante o processo da elaboração da presente monografia, pela paciência, disponibilidade, e principalmente pelas críticas que foram bastante construtivas.

Sem mais de longas o meu muito obrigado aos docentes, colegas e amigos do curso Engenharia Informática 2016. Obrigado pelo aprendizado, pelo empenho e pela dedicação.

EPÍGRAFE

“A sabedoria do homem lhe dá a paciência e a sua glória é ignorar as ofensas.”

Provérbios 19:11 – Bíblia Sagrada

ÍNDICE

RESUMO	i
ABSTRACT	ii
LISTA DE FIGURAS	iii
LISTA DE TABELAS	iii
LISTA DE SIGLAS E ABREVIATURAS	v
INTRODUÇÃO	6
CAPÍTULO I – MARCO TEÓRICO CONTEXTUAL	12
1.1. Fundamentação epistemológica	12
1.1.1. Evolução da automação residencial	12
1.1.2. Conceitos de Automação Residencial	13
1.1.3. Aplicabilidade de sistemas de automação residencial.....	13
1.1.4. Benefícios de sistemas de Automação Residencial.....	14
1.1.5. Sistemas de Segurança	15
1.1.6. Sistemas de Segurança Contra Incêndio	15
1.1.7. Sistemas de automação industrial x residencial	15
1.2. Sensores e Actuadores.....	16
1.3. Comunicação.....	17
1.3.1. Porta Serial	18
1.3.2. Porta Paralela.....	18
1.3.3. Barramento e Interfaces USB.....	19
1.4. Microcontrolador.....	20
1.5. Arduino	20
1.6. Linguagem de programação	23
1.7. Ferramentas de automação residencial.....	27
1.7.1. O Módulo GSM.....	27
1.7.2. Sensor de presença	29
1.7.3. Sensor de Gás MQ-135	30
1.7.4. LDR.....	31
1.8. Redes Domiciliares	32
1.8.1. Bluetooth	33
1.9. Sistema Operacional Android e Armazenamento	35
1.10. SQLite	36
1.11. Diagramas UML.....	37

1.12.	Fritzing	37
1.13.	Trabalhos correlatos	37
CAPÍTULO II - DESENVOLVIMENTO DO SISTEMA		40
2.1.	Descrição do sistema	40
2.2.	Arquitetura do sistema.....	40
2.3.	Etapas de desenvolvimento do sistema	41
2.4.	Especificação de requisitos	42
2.5.	Construção da Maquete.....	44
2.6.	Hardware	45
2.6.2.	Divisor de tensão.....	47
2.7.	Software	51
2.7.1.	Aplicação Android	51
2.7.2.	Fluxograma de visualização do estado dos dispositivos ou aparelhos	59
2.7.3.	Fluxograma de accionamento de dispositivos ou aparelhos.....	60
2.7.4.	Fluxograma de agendamento de tarefas	60
2.7.5.	Fluxograma do funcionamento do Subsistema de Climatização.....	61
2.7.6.	Fluxograma do funcionamento do Subsistema de iluminação	62
2.7.8.	Fluxograma do funcionamento do Subsistema de Controlo de Incêndio	64
2.8.	INTERFACE DO UTILIZADOR (APLICAÇÃO ANDROID)	65
2.9.	FIRMWARE DO SISTEMA.....	70
CAPITULO III – CONCLUSÕES E RECOMENDAÇÕES		72
3.1	Conclusão.....	72
3.2.	Recomendações.....	73
REFERENCIA BIBLIOGRÁFICAS		74
APÊNDICE		77

RESUMO

A automação está cada vez mais próxima do homem, pelo facto de permitir o desenvolvimento de sistemas que proporcionem aos seus usuários a segurança e o conforto. Neste contexto, este trabalho contempla o desenvolvimento de um sistema de automação voltado para a segurança residencial, incluindo a prevenção ou combate a incêndio. O desenvolvimento deste sistema é feito usando o microcontrolador do arduino e os módulos GSM sim800l e Bluetooth HM-10, visando a redução de custos de implementação. Com objectivo de executar medidas de segurança em uma residência como por exemplo o controle e combate a um eventual incêndio, executando o accionamento de itens de segurança como: a activação de alarmes sonoros, enviar uma mensagem de texto (SMS) ou ligação telefónica, abertura e fechamento de portas e janelas, corte de energia em determinados ambientes; É concebido também para ser contra intrusão, através da detecção de usuário dentro da residência através de sensores de presença.

Palavras-chave: Arduino, Segurança Residencial, Automação, Sensores.

ABSTRACT

Automation is increasingly closer to man, because it allows the development of systems that provide its users with safety and comfort. In this context, this work includes the development of an automation system aimed at residential security, including the prevention of fire fighting. The development of this system is done using the microcontroller and the GSM sim8001 and Bluetooth HM-10 modules, reducing the cost of execution. In order to carry out security measures in a residence, such as example or control and fight against a possible fire, use or activate security items such as: an activation of audible alarms, send a text message (SMS) or telephone call, opening and closing doors and windows, cutting power in selected environments; It is also designed to be against intrusion, by detecting users inside the residence using presence sensors.

Keywords: Arduino, Home Security, Automation, Sensors.

LISTA DE FIGURAS

Figura 1: Sensores e Actuadores	16
Figura 2: Placa Arduino.....	21
Figura 3: Arduino IDE.....	27
Figura 4: Módulo GSM Sim800L	29
Figura 5: Sensor de presença	30
Figura 6: Módulo Sensor de Gás MQ-135	31
Figura 7: Sensor de luminosidade - LDR	32
Figura 8: Tecnologias para redes domésticas	33
Figura 9: Módulo Bluetooth	35
Figura 10: Protótipo Projecto Maia	38
Figura 11: Trabalho correlato Arduino, protoboard	39
Figura 12: Arquitectura do sistema	41
Figura 13: Vista Superior da Maquete.....	45
Figura 14: Diagrama de funcionamento do relé.	46
Figura 15: Accionamento do Relé.....	47
Figura 16: Circuito de divisor de Tensão	48
Figura 17: Circuito da Leitura do LDR	49
Figura 18: Esquema do medição de tensão e recarregamento automático	50
Figura 19: Diagrama de casos de uso da Aplicação	52
Figura 20: Fluxograma de visualização do estado dos dispositivos.....	60
Figura 21: Fluxograma de agendamento de tarefas.....	61
Figura 22: Fluxograma de funcionamento do subsistema de climatização	62
Figura 23: Fluxograma de funcionamento do subsistema de Iluminação	63
Figura 24: Fluxograma de funcionamento do subsistema de Anti-intrusão	64
Figura 25: Fluxograma de funcionamento do subsistema de controlo de incêndio	65
Figura 26: Tela de Autenticação.....	66
Figura 27: Tela para seleccionar o dispositivo Bluetooth	66
Figura 28: Tela de gerenciamento	68
Figura 29: Tela de Monitoramento por Voz.....	68
Figura 30: Tela de dispositivos de iluminação	69
Figura 31: Tela de gerenciamento iluminação	69
Figura 32: Tela de cadastro de luzes	70

LISTA DE TABELAS

Tabela 1: Comparativo de Automação	10
Tabela 2: Diferença entre os sistemas de automação	16
Tabela 3: Especificações técnicas do Arduino.	23
Tabela 4: Representação das etapas do desenvolvimento do projecto	42
Tabela 5: Descrição dos Requisitos Funcionais do Sistema	43
Tabela 6: Descrição dos Requisitos Não Funcionais do Sistema.....	44
Tabela 7: Descrição do caso de uso efectuar login.....	53
Tabela 8: Descrição do caso de uso accionar lâmpadas	53
Tabela 9: Descrição do caso de uso agendar para accionar lâmpadas.....	54
Tabela 10: Descrição do caso de uso agendar tarefa para accionar ventiladores	55
Tabela 11: Descrição do caso de uso accionar o sistema de detenção de movimento ...	56
Tabela 12: Descrição do caso de uso accionar o sistema de detenção de movimento ...	57
Tabela 13: Descrição do caso de uso accionar o sistema de combate a incêndio	58
Tabela 14: Requisitos alcançados e não alcançados do Sistema.....	71

LISTA DE SIGLAS E ABREVIATURAS

HVAC	Aquecimento, ventilação e ar condicionado;
GSM	GLobal System for Mobile Communications;
SMS	Short Message Service;
PLC	Power Line Carrier;
UART	Universal Assynchronous Receiver Transmitter;
USART	Universal Synchronous Assynchronous Receiver Transmitter;
USB	Universal Serial BUS;
PIR	Passive Infrared Sensor;
SIM	Módulo de Identificação do Assinante;
ACID	Atomicity, Consistency, Isolation, Durability;
MSG	Mensagem;
PNEs	Pessoas Portadoras de Necessidades Especiais;
LDR	Light Dependent Resistor;

INTRODUÇÃO

De acordo com Muratori (2011) *Apud* Da Silva, Nardoto & Jorge (2016), a automação residencial é um conceito muito recente, sendo uma área que apresentou grande crescimento nos últimos anos. As primeiras utilizações desta tecnologia datam do final da década de 70, data a qual surgiam nos Estados Unidos os primeiros controladores inteligentes, onde os comandos eram feitos pela própria rede eléctrica da residência, no conceito de PLC.

Dentro da automação residencial¹, apresenta-se o conceito de Domótica, palavra que provém da junção do termo em latim *domus*, que significa casa e da palavra robótica, sendo seu objectivo o melhor controlo e praticidade das rotinas e tarefas residenciais. Segundo Bolzani (2004) domótica é um processo de automação em que se utilizam mecanismos automáticos de tomada de decisões seguindo técnicas de Inteligência Artificial. As ideias das residências inteligentes se aplicam em diversas ocasiões dentro de uma casa, tais como: monitoramento residencial, controle de luminosidade, diminuição dos desperdícios eléctricos e hidráulicos, controle e interrupção de aparelhos electrónicos, controle da temperatura interna da casa e/ou da piscina, entre outros.

No nosso dia-a-dia verifica-se o avanço acelerado da tecnologia, e com isso é inevitável proporcionar conforto, e principalmente segurança. A tecnologia tem avançado ao ponto onde uma casa automatizada tornou-se uma opção viável para os proprietários de residências. Imagine andar em sua casa e as luzes iluminando o seu caminho sem ter que tocar num interruptor, ser notificado através do celular sobre intrusão ou quando o ar condicionado estiver ligado enquanto ninguém estiver na casa. Além disso, tal sistema poderia permitir o utilizador agendar eventos para ocorrer em intervalos recorrentes (por exemplo, ligar o sistema de rega às 4:30h a cada terça e quinta-feira). Pensando nesses factores, a ideia de automatizar uma residência define-se em simplificar e facilitar diversas acções realizadas no dia-a-dia, tanto para pessoas comuns como também para pessoas com deficiência, que não podem se locomover para abrir uma janela, por exemplo. Para que isso ocorra são necessários alguns

¹ **Automação Residencial** é definida como sendo um conjunto de dispositivos, sistemas e subsistemas que tem a habilidade de interagir entre si ou funcionar de forma independente.

equipamentos, de preferência de baixo custo, como placas de Arduino², tecnologia Bluetooth, etc.

O ideal para arquitectar uma residência totalmente automatizada é fazer o projecto considerando, desde a sua construção, os dispositivos que serão automatizados. Isso se faz necessário devido a infra-estrutura de montagem, onde existe a indispensabilidade de um controlador que deverá estar conectado com todos os dispositivos a serem controlados (normalmente por cabos).

Problematização

Com base nos resultados auferidos da investigação científica realizada, foram estabelecidas as seguintes **manifestações fácticas**:

- As limitações encaradas pelas pessoas de menor poder económico para a aquisição de soluções de protecção do seu património;
- A não escalabilidade dos produtos de automação importados por serem de tecnologias fechadas;
- A busca de comodidade e a ineficiência dos sistemas da segurança física.

A grosso modo, o **problema** principal encontrado para justificar o desenvolvimento deste trabalho é que, apesar do grande número de opções de automação no mercado a maioria está sendo oferecida a valores altos, não permitindo que essa tecnologia seja de fácil acesso. Tais sistemas, têm o objectivo de facilitar as tarefas, ou mesmo melhorar a harmonia e a interacção das pessoas com o ambiente. No caso especial de algumas pessoas portadoras de necessidades especiais (PNEs), o sistema poderia facilitar o accionamento de lâmpadas.

Diante da base acima referenciada, estabelece-se a seguinte questão central de investigação deste trabalho: Podem os sistemas de automação residencial baseados na tecnologia Bluetooth garantir a escalabilidade e reduzir o custo de aquisição dos mesmos?

Este trabalho **delimita-se** em apresentar um protótipo de um sistema de automação residencial, controlado por meio de um smartphone³ com uma aplicação

² **Arduino** é uma plataforma de computação física de fonte aberta, com base em uma placa simples de entrada/saída (input/output, ou I/O), assim como em um ambiente de desenvolvimento que implementa a linguagem Processing (Processing é uma linguagem de programação de código aberto).

Android que faz a gestão de sistemas domésticos localmente usando a tecnologia de redes sem fio (Bluetooth) e controlado também via SMS ou ligações telefónicas por meio do módulo GSM Sim8001. Isto é, é feito usando o microcontrolador arduino e os módulos GSM Sim8001 e Bluetooth HM-10, visando a redução de custos de implementação, executando medidas de segurança em uma residência como por exemplo o controle e combate a um eventual incêndio, executando o accionamento de itens de segurança como: a activação de alarmes sonoros, enviar uma mensagem de texto (SMS) ou ligação telefónica, abertura e fechamento de portas e janelas, corte de energia em determinados ambientes.

Para responder o problema da presente investigação, tem-se a seguinte **hipótese**:

- O desenvolvimento de um sistema de automação baseado em GSM e Bluetooth permitirá que as classes económicas de menor poder aquisitivo possam ter acesso as tecnologias de conforto e protecção ao património.

Objectivos

Com vista a atender as implicações acima descritas coloca-se o seguinte **objectivo geral**:

- Elaborar uma solução para automação de accionamento e segurança residenciais, por meio de uma aplicação android e comunicação GSM.

Para que o objectivo geral seja alcançado os seguintes **objectivos específicos** são levados a cabo:

- Desenhar o circuito de interligação dos dispositivos;
- Esboçar um protótipo de casa com maquete;
- Desenvolver o firmware⁴ do sistema;
- Programar o microcontrolador para permitir a comunicação por sms ou chamadas telefónicas;
- Desenvolver uma aplicação android para accionamentos locais;
- Simular o sistema;

³ **Smartphones**: são celulares inteligentes que contam com sistemas operacionais para executar aplicativos e realizar uma série de tarefas

⁴ Em electrónica e computação, **firmware** é o conjunto de instruções operacionais programadas directamente no hardware de um equipamento electrónico.

Justificação

A escolha deste tema **justifica-se** pelo facto de, mais do que ter a comodidade, a segurança em residências constitui uma das principais preocupações da população em todo mundo, e como solução a isto muitos optam em contratar segurança privada ou em importarem o produto de “alarmes” pronto, com tecnologia fechada e pouco flexíveis, o que tem acarretado custos financeiros elevados, daí que o desenvolvimento de um sistema de segurança e accionamento residencial automatizado com Arduino baseado em GSM sim800l e Bluetooth HM-10, para além de ser eficiente por ser de tecnologias abertas, tornou-se também mais barato, assim as classes económicas de menor poder aquisitivo podem ter mais acesso as tecnologias de conforto e protecção ao património. Poderá também ser acessível às pessoas em geral e PNEs tornando o projecto adequado e interessante para uma parcela significativa da população.

Este sistema é também concebido para ser flexível e escalável⁵, de tal modo que a adição de novos componentes seja relativamente simples e modular sem necessidade de se redesenhar todo o sistema, e o facto de usar-se tecnologias abertas de software e hardware, reduz os custos de desenvolvimento e implementação.

Por outro lado, como o presente projecto busca, especificamente, atender as necessidades de automatização de residências para todas as pessoas, foi elaborado um protótipo que ilustra os conceitos abordados neste trabalho de conclusão de curso. Assim, como principais vantagens pode-se citar facilidade no manuseio e interacção com o usuário final, sendo capaz de garantir a segurança (segurança e combate a incêndio) e controlar pequenas actividades do quotidiano do homem, como iluminação, climatização, com um custo abaixo de 10.000,00 (dez mil meticais), ilustrado em **Apêndice I**, em relação a outros projectos oferecidos no mercado em 2020, segundo a Tabela 1, sem contar com o custo da mão-de-obra. Desta forma, este projecto é também justificado a partir do crescimento e emersão constante da sociedade, combinada com as novas tecnologias e recursos de software livre, bem como fontes de conhecimento distribuídas pela rede de internet e o custo de implantação e operação reduzidos em relação ao mercado.

⁵ **Escalável** - permite que a adição de novos componentes não altere a estrutura do sistema.

Sondagem no mercado internacional de automação residencial		
Sistema	Empresas	Preço mínimo
Combate a Incêndio	Car Accessories Global Dropshing Store	15.000,00 Mt
Sistemas de Alarme GSM	eMastiff SecuritAlarm Store	20.000,00 Mt
Automação de Climatização	Habitissimo	15.000,00 Mt
Automação de iluminação	Habitissimo	12.000,00 Mt
Automação de iluminação e climatização	Quero automação	18.000,00 Mt
Anti-intrusão	ACtech	35.000,00 Mt
Combate a Incêndio	ACtech	30.000,00 Mt

Tabela 1: Comparativo de Automação

Fonte: Autor (2020)

Metodologia

Para classificar a pesquisa em relação a sua natureza, com o objectivo de gerar conhecimento para uma aplicação prática, a mesma pode ser considerada aplicada e exploratória. Quanto aos procedimentos técnicos ela se utiliza de técnicas bibliográficas, experimentais e documentais (SILVA; MENEZES, 2005).

Na etapa de estudo, é realizado um levantamento bibliográfico com o objectivo de acumular informações suficientes para desenvolver as componentes de hardware e software do sistema. Este levantamento bibliográfico é baseado principalmente na documentação disponibilizada através de websites das empresas Google Inc e Arduino, livros e artigos sobre as tecnologias envolvidas.

Métodos teóricos

- Histórico – lógico: é utilizado na análise das literaturas, artigos e documentações especializadas e assuntos relativamente ao tema da pesquisa;
- Sistémico – estrutural – funcional: é utilizado para a elaboração, estruturação e a definição dos objectivos a serem alcançados;
- Análise – síntese: é utilizada no processamento de informações obtidas a partir de documentos e literatura.

Estrutura da Monografia

Para uma melhor leitura e interpretação, o presente trabalho apresenta-se estruturado em introdução e três capítulos, bibliografia, apêndices e anexos:

Na Introdução são apresentadas as noções gerais sobre o contexto da monografia assim como, os elementos chaves do trabalho de pesquisa.

Capítulo I – É feita uma abordagem teórica a qual situa o estudo, falando especificamente do conteúdo teórico relacionado ao problema de pesquisa, tecnologias e ferramentas propostas para a consubstanciação da solução em causa.

Capítulo II – É apresentado o processo de desenvolvimento da solução.

Capítulo III – É feita uma análise conclusiva relativamente ao estudo realizado, recomendações e conclusão da monografia

CAPÍTULO I – MARCO TEÓRICO CONTEXTUAL

Neste capítulo, apresenta-se o marco teórico cujo objectivo é fornecer as bases necessárias para a concepção deste trabalho e proporcionar uma melhor compreensão, explicação e interpretação do mesmo.

1.1. Fundamentação epistemológica

Nesta secção apresenta-se a evolução histórica da *domótica*⁶ ou simplesmente automação residencial.

1.1.1. Evolução da automação residencial

A história mostra que a primeira possibilidade de se instalar uma casa inteligente foi associada ao conforto do lar, com a energia eléctrica movimentando tudo e realizando todos os afazeres. Acreditava-se que, com um simples toque em um botão, todas as tarefas rotineiras seriam realizadas automaticamente (BRUCHER,1939).

Segundo o jornalista Ed Driscoll, o nascimento da automação residencial teve como factor significativo a introdução da televisão em 1939. A televisão posteriormente popularizou o conceito de automação doméstica através de programas populares dos anos 1960 como *The Jetsons*. Uma série de desenvolvimentos tecnológicos no final dos anos 1960 e início de 1970 tornaram a domótica significativamente mais plausível. (BIONDO, 2011)

O surgimento de residências automatizadas teve início em meados do século XX, por pessoas com interesses em electricidade, como por exemplo, Jack Fletcher, em 1954 (STIMSON, 1954). Em 1970, foram desenvolvidos, por algumas empresas, equipamentos com funções limitadas de automação residencial, como controle de iluminação e de temperatura. O mesmo princípio está sendo oferecido neste projecto, porém de uma forma simplificada.

Em 1975, uma tecnologia conhecida como X-10 permitiu que aparelhos e luzes se comunicassem uns com os outros, e os primeiros dispositivos X-10 surgiram em lojas de varejo em 1978. O sistema lançou um software de automação para computador pessoal nos anos 1980 e um sistema de segurança doméstico em 1980. Ao longo da

⁶ **Domótica** termo usado na “Robótica”, definida como integração dos mecanismos automáticos de um espaço, simplificando o quotidiano das pessoas, satisfazendo necessidades de comunicação, de conforto e segurança.

década de 1990, uma série de propagandas sobre o X-10 ajudou a popularizar ainda mais esta tecnologia de automação residencial.

A indústria tem mobilizado, a vários anos, tentativas de padronizar as redes domésticas e um protocolo de troca de mensagens entre aparelhos electrónicos em uma casa. Segundo Abowd e Sterbenz (2000), ainda não existia, no ano de 2000, um acordo se essa normatização seria positiva em curto prazo. Alguns são a favor para desenvolver o mercado, outros crêem que este tipo de norma poderá atrapalhar novos projectos.

A história tem mostrado que muitas tecnologias descobertas não são desenvolvidas de imediato para o mercado, pois não têm uma aplicação viável que permita sua utilização (MELOSI, 1990). Como exemplo, no final de 1870, a lâmpada inventada por Thomas Edson alavancou todo o sector industrial na parte de iluminação e equipamento eléctricos, abrindo espaço para novas iniciativas na área.

1.1.2. Conceitos de Automação Residencial

Segundo Gerhart (1999) define-se Automação Residencial como sendo um conjunto de dispositivos, sistemas e subsistemas que tem a habilidade de interagir entre si ou funcionar de forma independente. A automação residencial permite-nos controlar e monitorar qualquer aparelho ou sistemas na casa de forma individual ou colectiva através do agendamento automático ou fazendo alterações de improviso. Pode-se controlar dispositivos e sistemas tais como: sistema de segurança, sistema de rega automático, sistema de ventilação, electrodomésticos, dentre outros. Estes sistemas são personalizados para satisfazer uma variedade de necessidades individuais incluindo a comodidade, economia de energia, protecção e segurança.

A automação é o ato de tomar a operação de um processo ou tarefa de forma automática, com pouca ou nenhuma intervenção humana, liberando homem para as tarefas mais nobres, afastando-o de actividades perigosas ou aumentando o seu tempo de lazer. (BIONDO, 2011)

1.1.3. Aplicabilidade de sistemas de automação residencial

As aplicações mais comuns de automação residencial são:

- *Aquecimento, ventilação e ar condicionado (HVAC):* Com HVAC automatizado, pode-se reduzir o calor quando uma sala está desocupada, e

aumentar ou diminuir em momentos específicos baseados em alguma programação ou ocupação;

- *Controlo de iluminação:* é possível programar tempos em que as luzes devem ser ligadas e desligadas, decidir que quartos específicos devem ser iluminados em determinados momentos, seleccionar o nível de luz que deve ser emitida;
- *Controlo de aparelhos:* pode-se gerir aparelhos domésticos a partir de um computador ou um *smartphone*;
- *Vigilância:* sistemas de segurança doméstico integrado com um sistema de automação residencial pode fornecer serviços adicionais, tais como vigilância remota de câmaras de segurança através da *internet*, fecho centralizado de todas as portas e janelas, etc. Permitindo, também aceder a informações em tempo real sobre o estado da segurança da residência.

1.1.4. Benefícios de sistemas de Automação Residencial

A automação residencial proporciona vários benefícios dependendo de onde é aplicada. Os benefícios mais comuns são:

- **Eficiência energética.** A energia é usada somente onde e quando necessário. O controlo remoto ou temporizado de ventilação, ar condicionado, iluminação e aparelhos elimina o desperdício de energia (Gerhart,1999).
- **Conforto e conveniência.** Uma casa inteligente é mais conveniente que uma casa regular. Há poucas coisas por fazer e poucas coisas por lembrar de fazer. A casa (ou os dispositivos inteligentes na casa) faz a maior parte do trabalho. A habilidade de controlar tudo na casa através do *smartphone* ou computador – ou ter tudo accionado automaticamente, baseado em parâmetros que definem-se com antecedência, é uma perspectiva atraente (Miller, 2015).
- **Segurança** - O benefício segurança pode ser abordado de vários aspectos em automação residencial. Para protecção contra roubos e assaltos, o controlo de acesso a casa que pode ser por meio de sistema de biometria ou outro sistema. O modo de segurança permite, também, que os pais monitorem os filhos, estando no escritório ou em qualquer outro lugar, através da Internet. Dessa forma, eles conseguirão saber tudo o que se passa enquanto estiverem fora de casa (Startsollution, 2015).

1.1.5. Sistemas de Segurança

Nos tempos atuais aplicações de sistemas automatizados voltados para a área de segurança se tornam cada vez mais comuns, segundo Bolzani (2004, p.52) a evolução das tecnologias e da Automação Residencial vem nos permitindo ter o controlo de diversas tarefas relacionadas a segurança, tais como:

- Circuitos fechados de TV;
- Alarmes, monitoramento;
- Prevenção de acidentes, iluminação de segurança;
- Detecção de gases, fumaça e calor;
- Alarme de incêndio;
- Sistemas de apoio a combate de incêndio;
- Simulador de presença e controle de Rondas

1.1.6. Sistemas de Segurança Contra Incêndio

A automação de itens de segurança para o combate de incêndio tem como objectivo executar medidas e procedimentos preestabelecidos para apoio ao combate. Segundo Bolzani (2004, p.82) um sistema completo de apoio e combate de incêndio deve contemplar a automação dos seguintes itens:

- Parada do sistema HVAC;
- Abertura e fechamento de portas e janelas;
- Fechamento de ductos de ar e gás;
- Avisos acústicos e ópticos;
- Envio de mensagens e alertas via rede telefónica ou rede de acesso;
- Corte de energia em determinados ambientes;
- Detecção de usuário dentro da residência através de sistemas de percepção.

1.1.7. Sistemas de automação industrial x residencial

Automação Industrial	Automação Residencial
Projecto	
Focado na automatização da manufactura, a fim de promover a maximização da produção e um aumento da qualidade do produto	Focado no cliente, a fim e atender suas necessidades que grande parte de vezes são muitos específicas

Infra-Estrutura	
Grande parte de vezes prevista e criada antes do início da obra	Na maior parte das vezes realizada depois que a obra já está pronta, o que aumenta custos.
Interface	
Complexas, já que as operações dependem de muitas variáveis de controlo	Fácil e amigável, já que o cliente busca conforto e descompilação com o sistema
Supervisão	
Monitoramento imprescindível acompanhado de relatórios de controlo e auditorias	Não necessita de supervisão, a excepção dos sistemas de segurança e alarme. O custo seria elevado se comparado aos benefícios

Tabela 2: Diferença entre os sistemas de automação

Fonte: Biondo (2011)

1.2. Sensores e Actuadores

Os sensores e actuadores são a interface da residência com o meio físico. Os sensores transformam parâmetros físicos (temperatura, humidade, etc.) em sinais eléctricos apropriados para que os sistemas de automação residencial possam analisá-los e tomar decisões. (SAMUEL, 2016).

Os actuadores são componentes electromecânicos que têm suas características alteradas conforme os impulsos eléctricos recebidos. Eles podem ser interligados directamente aos dispositivos inteligentes ou ser conectados por meio de pequenas interfaces a uma rede de dados para que possam ser utilizados por qualquer sistema doméstico.

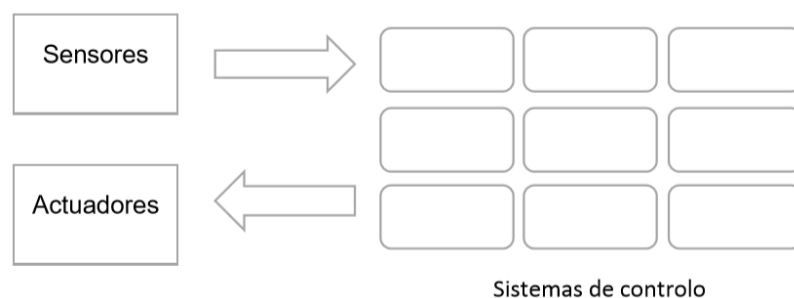


Figura 1: Sensores e Actuadores

Fonte: SAMUEL (2016)

No mercado de automação existe uma variedade muito grande de sensores que possibilitam o monitoramento de inúmeras grandezas físicas e eventos. Basicamente, são os sensores que retornam a informação de um evento permitindo que o controlador saiba se uma acção enviada foi executada com sucesso ou não. Uma bomba de água, por exemplo, só para de funcionar quando o sensor de nível retorna a informação de cheio para o controlador local que é quem efectivamente controla a bomba. Pode-se, no entanto, enviar tal informação para uma central com propósitos diversos, como um meio de informação aos moradores que há água suficiente ou não para todos tomarem banho. O mesmo acontece com um aquecedor que liga ou desliga a chama dependendo da temperatura acusada por um sensor. (SAMUEL, 2016)

Uma falha na detecção ou mesmo o mau posicionamento do sensor pode ocasionar um erro de leitura causando uma situação de risco. Os sistemas de controlo devem ser projectados sempre pensando na segurança do usuário.

1.3. Comunicação

Os periféricos utilizados actualmente como mouses, teclados e impressoras, necessitam de uma forma de comunicação e de uma sincronia de transmissão de dados entre eles e outro dispositivo, como um microcomputador, para poder funcionar e executar a sua tarefa correctamente.

A sincronia de transmissão de dados entre dois dispositivos é muito importante para que o sinal seja enviado, recebido e interpretado correctamente. Existem duas formas de sincronia de transmissão de dados, que serão analisadas a seguir, são elas: transmissão assíncrona e transmissão síncrona de dados.

Segundo Silveira (1991) e Monteiro (2001), *apud* Quinderé (2009):

A transmissão assíncrona é a forma mais simples de ser projectada e se caracteriza pela irregularidade entre o tempo de transmissão de sinais entre o emissor e o receptor que é estabelecido, através de um pulso inserido no início da informação, conhecido como start, e um pulso no final da informação, conhecido como stop, a cada informação enviada e recebida. E a transmissão síncrona tem como sua principal característica o envio de informações através de blocos de dados

de forma sequencial respeitando uma frequência determinada por dispositivos de controlo da comunicação, como exemplo podem ser citados os chips UART e USART.

Existem basicamente três interfaces de comunicação entre dois dispositivos, também chamadas de portas paralela, serial e USB.

A porta utilizada foi a paralela devido a sua simplicidade de operação que irá reduzir a complexidade do projecto e alta velocidade na transmissão de dados.

1.3.1. Porta Serial

A porta serial, uma das mais antigas e básicas interfaces de comunicação de computadores, integra a maioria dos computadores atuais e é muito utilizada para mouses e PDAs. Contudo, está perdendo o seu lugar para as portas USB (Universal Serial Bus) (TYSON, 2009).

O nome serial é devido ao fato de que os dados são serializados, ou seja, os dados são enviados e recebidos a um bit por vez, fato que determina suas principais características que foram levadas em consideração no momento de escolha da interface de comunicação. São elas, a necessidade de apenas um fio para a transmissão de dados e uma velocidade menor do que a da interface paralela que pode enviar oito bits por vez (TYSON, 2009a).

1.3.2. Porta Paralela

Esta interface, que utiliza o modo de transmissão síncrona, tem como característica principal a alta velocidade de transmissão de dados, quando comparada à interface serial, devido ao fato de transmitir vários bits paralelamente. É utilizada por diversos tipos de equipamentos, entre os mais comumente utilizados citam-se impressoras, *scanners* e controladores de jogos (MESSIAS, 2006).

Os sistemas operacionais da família Windows reconhecem como *LPTi* (Line Printer Terminal) a identificação padrão para portas paralelas, em que representa um identificador numérico com valores entre e, inclusive, desde que existam portas paralelas (*LPT1, LPT2, ..., LPTn*).

Segundo Messias (2006), as portas paralelas possuem três registos:

- (1) Data lines ou linhas de dados (registo utilizado para o envio de dados, tem por padrão o endereço 378h),
- (2) Control lines ou linhas de controle (registo utilizado para controlar os dispositivos conectados à porta paralela, tem por padrão o endereço 37Ah) e
- (3) Status lines ou linhas de status (registo utilizado para a recepção de dados, tem por padrão o endereço 379h).

Segundo Messias (2006), outra característica importante da interface paralela é que ela possui duas formas de operação. São elas:

- SPP (Standard Parallel Port): modo unidireccional com taxa de transferência de até 150 Kb/s (kilobits por segundo).
- EPP (Enhanced Parallel Port): modo bidireccional com taxa de transferência de até 2 Mb/s (megabits por segundo).

O padrão utilizado neste projecto foi o EPP, pois permite comunicação bidireccional, possibilitando a leitura e escrita simultânea dos registos da porta paralela.

1.3.3. Barramento e Interfaces USB

Presente em praticamente todos os computadores actuais, a porta USB vem ganhando cada vez mais espaço no mercado e utilidades novas como conexões com impressoras, *scanners*⁷ e teclado (TYSON, 2009b).

Devido a sua simplicidade e capacidade de resolver problemas de comunicação, a interface USB está sendo adoptada como padrão para todos os novos equipamentos desenvolvidos (TYSON, 2009b).

A inclusão desta seção dedicada ao padrão USB se deve à tendência da substituição das portas serial e paralela por interfaces USB. Em diversos computadores, notadamente nos *Notebooks*, já não são encontradas as portas seriais. Acredita-se que o mesmo venha a ocorrer brevemente com a interface paralela. Para ambos os casos, existem adaptadores USB – Serial e USB – Paralelo, que permitem a comunicação com periféricos que disponham apenas destes tipos de interface. Portanto, uma das extensões deste trabalho seria a inclusão ao suporte USB.

⁷ Um digitalizador (em inglês: **scanner**) é um periférico de entrada responsável por digitalizar imagens, fotos e textos impressos para o computador, um processo inverso ao da impressora. Ele faz varreduras na imagem física gerando impulsos eléctricos através de um captador de reflexos.

1.4. Microcontrolador

Os microcontroladores incorporam em um único encapsulamento um microprocessador, periféricos (i.e. temporizadores, comunicação serial, conversores analógico/digital e PWM, moduladores de largura de pulso) e memória de programa e dados. De acordo com Sica (2006) os microcontroladores são usados em dispositivos de controle automático e ganham vantagem, sobre os projectos que usam microprocessador, memória e dispositivos de entrada e saída separados. Estas vantagens são dadas pela redução de custo, tamanho e consumo de energia, embora apresente menor capacidade.

Com a intenção de reduzir custos em instalações eléctricas, e diminuir a quantidade de componentes empregados em seu espaço físico, surgiu a ideia de criar um dispositivo que apresente um microprocessador (CPU) juntamente com os demais periféricos (ROM, RAM, portas, A/D, D/A, entre outros), aumentando até a sua confiabilidade. Surge assim o microcontrolador (SICA, 2006) *apud* ALVAREZ (2015). Com o desenvolvimento da tecnologia dos Circuitos Integrados (CIs), foi possível inserir uma grande quantidade de transístores em um único chip. Assim foi possível a criação de microprocessadores que eram integrados a computadores e eram usados com periféricos externos como linhas de entrada e saída, temporizadores, memória, e outros. Os microcontroladores surgiram então no momento em que houve um aumento do nível de integração desses componentes, de acordo com Sica (2006) *apud* ALVAREZ (2015).

1.5. Arduino

O Arduino foi desenvolvido inicialmente na Itália, projectado como um instrumento para auxiliar alunos do ensino. No entanto, em 2005 passou a ser comercializado por Massimo Banzi e David Cuartielles, vindo a ser um produto bem-sucedido entre fabricantes e estudantes, por apresentar uma fácil utilização e boa durabilidade (MONK, 2013) *apud* Jorge (2016).

A meta dos professores foi a de desenvolver um hardware com facilidade de programação e a um custo reduzido. O projecto do Arduino foi concebido como open source, onde todo seu projecto, código fonte e hardware são de domínio público (MONK, 2013) *apud* Jorge (2016).

Por outro lado, o Arduino é uma plataforma de computação física de fonte aberta, com base em uma placa simples de entrada/saída (input/output, ou I/O), assim como em um

ambiente de desenvolvimento que implementa a linguagem Processing (Processing é uma linguagem de programação de código aberto). O Arduino pode ser utilizado para desenvolver objectos interactivos independentes, ou conectado a softwares de seu computador (BANZI, 2011).



Figura 2: Placa Arduino

Fonte: Ribeiro (2017)

O sistema Arduino apresenta dois componentes básicos: a placa Arduino, que é um elemento físico (hardware) utilizado para construir seus objectos, e a IDE (Integrated Development Environment), que é um programa de computador onde é gerado o código que será executado na placa Arduino (BANZI, 2011). Na Figura 3 é apresentado um diagrama de blocos genérico que apresenta os principais passos na programação do Arduino.

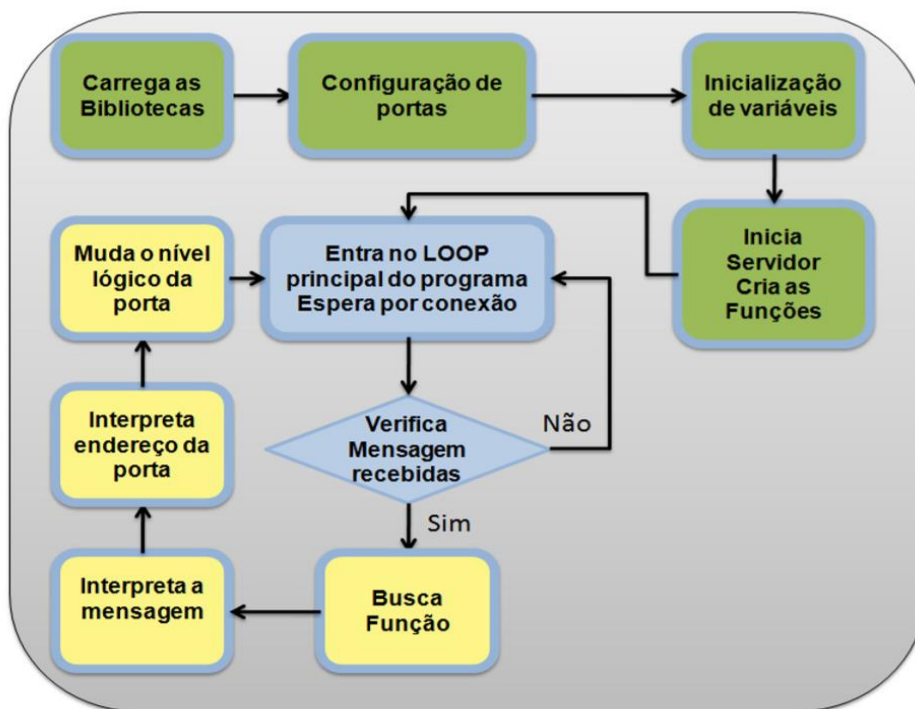


Figura 2: Diagrama de blocos da programação do Arduino.

O Arduino foi escolhido por se tratar de um ambiente multiplataforma, pois pode ser executado no Windows, Macintosh e Linux. Possui um ambiente de desenvolvimento fácil de ser utilizado, tem por base o IDE de programação Processing. Pode ser programado utilizando cabo USB, um excelente recurso tendo em vista que muitos computadores novos não possuem mais portas seriais.

A tabela a seguir contém as especificações técnicas do Arduino

Microcontrolador	ATmega328P
Tensão de operação	5V
Tensão de entrada (recomentado)	7-12V
Tensao de entrada (limite)	6-20V
Pinos digitais I/O	14 (dos quais 6 provem saída PWM)
Pinos digitais PWM I/O Pins	6
Pinos entrada analógica	6
Corrente DC por pino I/O	20 mA
Corrente DC por pino de 3.3V	50mA
Flash Memory	- 32 KB (ATmega328P) - dos quais 0.5 KB usado pelo bootloader

SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHZ
Cumprimento	68.6mm
Largura	53.4mm
Peso	25g

Tabela 3: Especificações técnicas do Arduino.

Fonte: <https://www.arduino.cc/en/Main/ArduinoBoardUno>.

1.1.1. Uso do Arduino na automação residencial

A plataforma de prototipagem Arduino foi o projecto com o objectivo de criar ferramentas que são acessíveis e também de baixo custo para ser usada de forma ágil e flexível por amadores ou artistas. Por se tratar de uma ferramenta simples de ser usada e com muitos recursos, ela permite que aficionados por tecnologia desenvolvam as mais variadas aplicações para a plataforma. Desde um projecto simples como acender ou apagar um Led, até projectos de interacção de comando de voz (DALPONTE, 2014). Projectos unitários podem não representar muito, porém quando funcionalidades começam a ser inseridas, o sistema vai se tornando mais complexo, mas em contra partida mais interessante, como é o caso: se em um sistema de interacção de comando de voz, estiver interligado com um sistema de fechaduras electrónicas, que dependendo do horário se accionadas acendem luzes espalhadas pelo ambiente, que por sua vez este ambiente possui sensores que detectam a presença de veículos e abrem e fecham portões sozinhos, teremos um ambiente domótico controlado com esta incrível ferramenta que é o Arduino.

1.6. Linguagem de programação

1.1.2. Conceito

Para facilitar a tarefa de programar um computador, foram criadas várias linguagens de programação. Estas linguagens são uma maneira de tentar escrever as tarefas que o computador vai realizar de maneira mais parecida com a linguagem natural. Embora ainda seja muitas vezes complexo em comparação com a linguagem natural, um programa escrito em uma linguagem de programação é muito mais fácil de ser implementado, compreendido e modificado. (Gomes, 2011)

As linguagens de programação são um meio termo entre a linguagem de máquina e a linguagem natural. Deste modo são classificadas de acordo com o nível entre a linguagem natural ou de máquina que ocupam. As linguagens muito parecidas com linguagem de máquina são chamadas de linguagens de baixo nível e suas instruções parecem-se muito com aquelas que serão executadas pelo processador. As linguagens de alto-nível são as que guardam mais semelhanças com a linguagem natural. Exemplo de linguagens de baixo nível é a linguagem de montagem (assembly). Exemplos de linguagens de alto-nível são: Pascal, C, Fortran, Java, Perl, Python, Lisp, PHP

1.1.3. Linguagem de Programação de Arduino

Toda programação tem em base uma nova linguagem que se chama processing. Para o ambiente de desenvolvimento temos várias funções que facilitam o desenvolvimento de qualquer software, pois possui bibliotecas já prontas que facilitam a comunicação com outros hardwares.

De vários tipos de linguagens de programação muito utilizadas. Mas, neste trabalho vamos utilizar a linguagem de programação em C. Pois, é uma linguagem utilizada na programação das placas Arduino. A programação em linguagem C é estruturada, já que os programas são divididos em funções, independentes entre si, para realizar determinada tarefa. O seu uso permite maior velocidade no desenvolvimento de projectos, devido à facilidade de programação e entendimento. (PEREIRA, 2003)

Essas funções básicas de referências são:

- Estruturas de controle (if, else, break);
- Operadores aritméticos e de comparação (+, -, !=, =, ==);
- Sintaxe básica (include, define);
- Operadores booleanos (||, !);
- Acesso a ponteiros (*,);
- Operadores compostos (+, -, +=);
- Operadores de bits (|, ^, :, ...);

Valores de referências:

Existem dois tipos de passagem de parâmetros para as funções:

- Valor – permite o uso dentro de uma função uma cópia do valor de uma variável, mas não permite alterar o valor da variável original (somente a cópia pode ser alterada).
- Referência – É passada para a função uma referência da variável, sendo possível alterar o conteúdo da variável original usando-se esta referência.

Com o uso de ponteiros torna possível alterar os valores das variáveis passadas como argumentos para uma função.

- Conversões (char (), byte (), int ());
- Tipos de dados (byte, array, int, char);
- Variável de escopo e de qualificação (variable scope, static, volatile);
- Utilitários (sizeof (), mostra o tamanho da variável em bytes).

O software que acompanha o Arduino possui várias funções e constantes que facilitam a programação, entre elas:

- Loop ();
- Setup ();
- Bibliotecas (Servo, Tone, Serial, etc);
- Constantes (LOW, HIGH, INPUT , OUTPUT).

Funções:

Funções são conhecidas também como sub-rotinas, muito utilizadas em programação. Uma das vantagens de usar funções é de não precisar copiar o código todas as vezes que precisar executar aquela operação, deixando assim a leitura do código mais objetiva e limpa. (FONSECA e BEPPU, 2010).

As funções básicas de referências são:

- Digital I/O – pinMode(), digitalWrite(), digitalRead();
- Analógico I/O – analogReference(), analogRead(), analogWrite(), – PWM;
- Tempo – millis(), micros(), delay(), delayMicroseconds();
- Avançado I/O – tone(), noTone(), shiftout(), pulseIn();
- Trigonométrica: Só do C/C++(sin(), cos(), tan());
- Matemática: min(), max(), abs(), constraint(), map(), sqrt();
- Números aleatórios: randomSeed(), random();

- Interrupções externas: `attachInterrupt()`, `detachInterrupt()`;
- Bits e Bytes: `lowByte()`, `highByte()`, `bitRead()`, `bitWrite()`, `bitSet()`, `bitClear()`, `bit()`;
- Interrupções: `interrupts()`, `nointerrupts()`;
- Comunicação serial.

1.1.4. Bibliotecas de arduino

As bibliotecas Arduino proporcionam funcionalidade extra para uso em códigos, por exemplo, programar outros dispositivos ou manipulação de dados. Quando usamos as bibliotecas, temos um amplo e diverso horizonte de programação quando se compara apenas o uso de estruturas, funções e valores. Para utilizar uma biblioteca, já se deve estar instalada e disponível para uso no computador. (REBESCHINI, 2012. p 32) *apud* TÓFOLI, 2014.

A seguir serão apresentadas as bibliotecas de referências:

- Ethernet – Usa-se para conectar uma rede Ethernet usando o Arduino Ethernet Shield;
- EEPROM – Faz a leitura e escrita de armazenamento permanente;
- Firmata - Usado para se comunicar com os aplicativos instalados no computador usando um protocolo Firmata;
- Servo – Usado para controlar servos motores;
- LiquidCrystal – Usado para controlar telas de cristal líquidos, os LCDs.
- SoftwareSerial – Usado para Comunicação serial em qualquer pino digital.
- Serial Peripheral Interface (SPI) – Usado para comunicar dispositivos que usam barramento serial.
- Wire – Usado para enviar e receber dados de uma rede de dispositivos e sensores.
- Stepper – Usa-se para controlar motores de passo.

1.1.5. IDE Arduino

O Arduino IDE é uma aplicação multiplataforma desenvolvida em Java, que é baseado em um ambiente de programação de código aberto. A linguagem utilizada é baseada em C e C++. (REBESCHINI, 2012. p 27) *apud* TÓFOLI, 2014.

A imagem abaixo representa a IDE de programação e upload do código para placa Arduino



Figura 3: Arduino IDE

Fonte: Autor (2020)

1.7. Ferramentas de automação residencial

1.7.1. O Módulo GSM

Existem vários módulos GSM que permitem, além de comunicação via celular, conexão de internet via GPRS e envio de mensagens SMS, ampliando mais ainda as possibilidades de controlo e monitoramento através do Arduino.

Diferentemente de outros modelos, ele já vem integrado na placa, necessitando apenas soldar os pinos (os quais acompanham). Possui integrado um slot para cartão SIM e uma antena de cobre, sem abrir mão de possibilitar conexão com outra antena.

Baseado no módulo wireless SIM800L, que é capaz de encaminhar serviços Quad-Band, ele permite fazer ligações de voz, enviar SMS e trocar dados via Internet GPRS (Serviço de Rádio de Pacote Geral) de forma a receber e realizar o envio de dados do Arduino para locais remotos junto de seu telefone celular GSM, por exemplo.

Após ampla pesquisa, foi escolhido o módulo GSM SIM800L, por ser de baixo custo e fácil implementação.

Características

Segundo Goulart (2017) as principais características do módulo GSM SIM800L são:

- Conexão junto à rede GSM;
- Formas de utilização e controle aprimoradas;
- Maior facilidade e rapidez de instalação junto ao Arduino; -
- Baseado no módulo *wireless*⁸ SIM800L;
- Capaz de receber e realizar o envio de dados de locais remotos;
- Operação semelhante a um telefone celular;
- Comunicação rápida e completa a distância;
- Suporte de cartão SIM (Verificar sua localidade e banda em MHz utilizada pela operadora);
- Pinos para conexão com microfone e agentes externos de reprodução de áudio;
- Essencial para quem quer dar um upgrade nos projectos com Arduino;
- Acompanham barras pinos;
- Pinagem especificada directamente no módulo;
- Baixo custo;

Especificações

O Módulo GSM SIM800L foi desenvolvido especialmente para integrar projectos utilizando o microcontrolador Arduino à Rede Mundial de Computadores por meio da comunicação GPRS, realizar ligações (GSM) ou ainda enviar mensagens SMS.

A seguir temos suas especificações:

- Alimentação: 3.7V a 4.2V (2A corrente de pico);
- Quadl-Band: 850/900/1800/1900MHz;
- GPRS classe multi-slot 10/8;
- Estação móvel GPRS classe B;
- Ganho da antena: 3dBi;
- Controle via comandos AT (GPP TS 27, 007, 27, 005 e SIMCOM);
- Temperatura de operação: -40 °C a +85 °C;

⁸ Rede **wireless** – também conhecida como rede sem fio – é uma forma de conexão entre dois ou mais dispositivos sem a utilização de cabos. A transmissão de dados e informações se dá por meio do uso de equipamentos de ondas de rádio ou via infravermelho.

- Dimensões (CxL): 25x23mm;

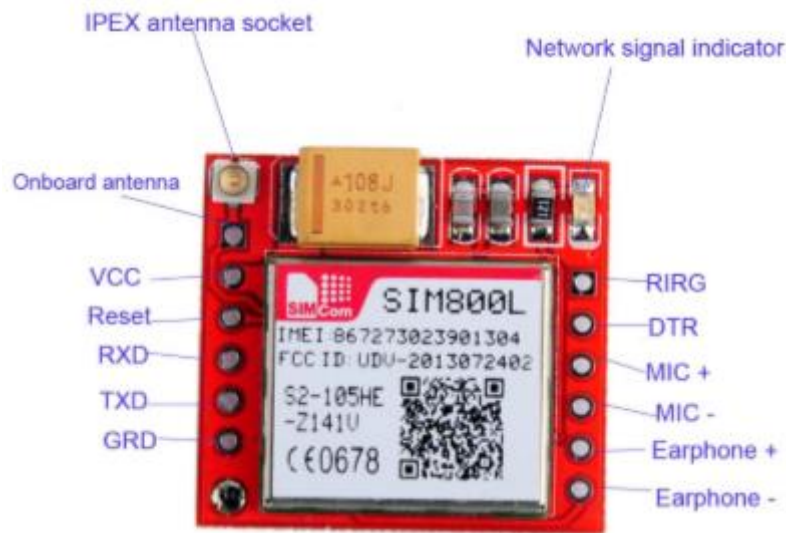


Figura 4: Módulo GSM Sim800L

Fonte: Autor (2020)

1.7.2. Sensor de presença

O Sensor de Movimento PIR DYP-ME003 consegue detectar o movimento de objectos que estejam em uma área de até 7 metros. Caso algo ou alguém se movimentar nesta área o pino de alarme é activo.

É possível ajustar a duração do tempo de espera para estabilização do PIR através do potenciômetro amarelo em baixo do sensor bem como sua sensibilidade. A estabilização pode variar entre 5-200 seg.

Especificações:

- Modelo: DYP-ME003;
- Sensor Infravermelho com controle na placa;
- Sensibilidade e tempo ajustável;
- Tensão de Operação: 4,5-20V;
- Tensão Dados: 3,3V (Alto) - 0V (Baixo);
- Distância detectável: 3-7m (Ajustável);
- Tempo de Delay: 5-200seg (Default: 5seg);
- Tempo de Bloqueio: 2,5seg (Default);
- Trigger: (L)-Não Repetível (H)-Repetível (Default: H);
- Temperatura de Trabalho: -20 ~ +80°C;

- Dimensões: 3,2 x 2,4 x 1,8cm;
- Peso: 7g.



Figura 5: Sensor de presença

Fonte: Autor (2020)

1.7.3. Sensor de Gás MQ-135

O sensor gás MQ-135 é capaz de detectar a concentração de vários gases tóxicos em um ambiente tais como: *amónia, dióxido de carbono, benzeno, óxido nítrico e também fumaça ou álcool*. Neste tutorial vamos aprender como usa o módulo **sensor de gás MQ-135** junto a um Arduino. Esse tutorial também pode ser aplicado para outros sensores de gás, com alguns ajustes, tais como: MQ-2, MQ-3, MQ-4, MQ-5, MQ-6, MQ-7, MQ-8, MQ-9.

A importância dos sistemas de protecção contra incêndio

Os processos industriais envolvem o uso e fabricação de substâncias altamente perigosas, especialmente produtos inflamáveis, tóxicos e oxidantes. Os gases fazem parte desses ambientes como participantes de reacções, catalisadores ou produtos finais. Outras vezes são resíduos ou subprodutos indesejáveis. Escapes ocasionais de gás ocorrem e criam um perigo potencial para a planta industrial, seus funcionários e pessoas que vivem nas proximidades. Incidentes e acidentes em todo o mundo, envolvendo asfixia, intoxicação, explosões e perdas de vidas é um lembrete constante desse problema.

Por isso, sistemas automáticos de detecção de gases são fundamentais para a protecção das pessoas envolvidas nesses processos. É necessário que tenhamos sensores

adequados e que sejam posicionados em lugares estratégicos, de preferência com redundância.

É claro que esse tipo de proteção não se limita a indústria. Podemos ver facilmente a aplicação de sensores de gás e fumaça em uma cozinha residencial ou em ambientes públicos. Enfim, podemos usar a automação a favor da vida, evitando assim acidentes.

O módulo MQ 135 possui 4 pinos: dois pinos para alimentação (Vcc e GND) e dois pinos para emissão dos sinais elétricos, sendo um sinal digital (D0) e o outro um sinal analógico (A0).



Figura 6: Módulo Sensor de Gás MQ-135

Fonte: Autor (2020)

1.7.4. LDR

O **LDR**, sigla em inglês de Light-Dependent Resistor, que significa resistor dependente de luz, nada mais é do que o que o próprio nome diz. Tipicamente, *quanto maior a luz incidente nesse componente, menor será sua resistência.*

O LDR é constituído de um semicondutor de alta resistência, que ao receber uma grande quantidade de fótons oriundos da luz incidente, ele absorve elétrons que melhoram sua condutibilidade, reduzindo assim sua resistência.

Especificações:

- Modelo: GL5528;

- Diâmetro: 5mm;
- Tensão máxima: 150VDC
- Potência máxima: 100mW
- Tensão de operação: -30°C a 70°C
- Espectro: 540nm
- Comprimento com terminais: 32mm
- Resistência no escuro: 1 M Ω (Lux 0)
- Resistência na luz: 10-20 K Ω (Lux 10)



Figura 7: Sensor de luminosidade - LDR

Fonte: Autor (2020)

1.8. Redes Domiciliares

Uma rede domiciliar é um sistema de comunicação que visa à interconexão de dispositivos encontrados em residências, normalmente restritos a uma distância de 30 metros, e que tem como objectivo a comunicação, o conforto, a economia de energia, a segurança, a assistência e o lazer (TEZA, 2002) *apud* ALVAREZ (2015). A Figura abaixo mostra as principais tecnologias encontradas para redes domésticas.

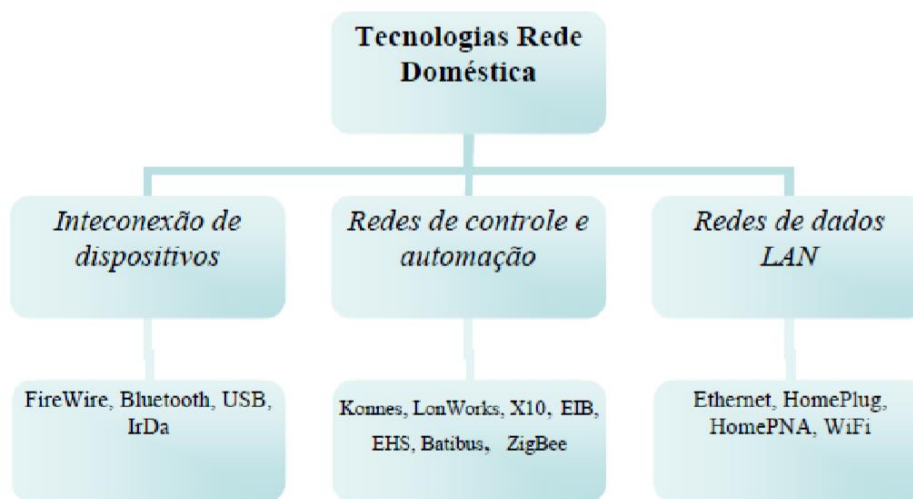


Figura 8: Tecnologias para redes domésticas

Fonte: ALVAREZ (2015)

1.8.1. Bluetooth

O Bluetooth é uma tecnologia de comunicação sem fio que permite que computadores, smartphones, tablets e afins troquem dados entre si e se conectem a mouses, teclados, fones de ouvido, impressoras e outros acessórios a partir de ondas de rádio. A ideia consiste em possibilitar que dispositivos se interligam de maneira rápida, descomplicada e sem uso de cabos, bastando que um esteja próximo do outro. Uma combinação de hardware e software é utilizada para permitir que este procedimento ocorra entre os mais variados tipos de aparelhos.

A transmissão de dados é feita por meio de radiofrequência, permitindo que um dispositivo detecte o outro independente de suas posições, sendo necessário apenas que ambos estejam dentro do limite de proximidade (a princípio, quanto mais perto um do outro, melhor). Para que seja possível atender aos mais variados tipos de dispositivos, o alcance máximo do Bluetooth foi dividido em três classes:

- Classe 1: potência máxima de 100 mW (miliwatt), alcance de até 100 metros;
- Classe 2: potência máxima de 2,5 mW, alcance de até 10 metros;
- Classe 3: potência máxima de 1 mW, alcance de até 1 metro.

Este índice sugere que um aparelho com Bluetooth classe 3 somente conseguirá se comunicar com outro se a distância entre ambos for inferior a 1 metro, por exemplo. Esta distância pode até parecer inutilizável, mas é suficiente para conectar um fone de ouvido a um telefone celular guardado no bolso de uma pessoa. É importante frisar, no

entanto, que dispositivos de classes diferentes podem se comunicar sem qualquer problema, bastando respeitar o limite daquele que possui um alcance menor.

A velocidade de transmissão de dados no Bluetooth é relativamente baixa: até a versão 1.2, a taxa pode alcançar, no máximo, 1 Mb/s (megabit por segundo). Na versão 2.0, esse valor passou para até 3 Mb/s. Embora essas taxas sejam curtas, são suficientes para uma conexão satisfatória entre a maioria dos dispositivos. Todavia, a busca por velocidades maiores é constante, como prova a versão 3.0, capaz de atingir taxas de até 24 Mb/s.

O módulo bluetooth implantado é o modelo HC-05. Oferece uma forma fácil e barata de comunicação com seu projecto Arduino. Diferente do modelo HC-06, além de ter uma fácil configuração, suporta tanto o modo mestre como escravo, ou seja, capaz de se conectar e receber conexões de outros dispositivos bluetooth. Em sua placa existe um regulador de que pode ser alimentado com 3.3 a 5v, bem como um LED que indica se o módulo está pareado com outro dispositivo. Possui alcance de até 10m.

Especificações do Módulo Bluetooth

- Protocolo Bluetooth: v2.0+EDR;
- Firmware: Linvor 1.8;
- Frequência: 2,4GHz Banda ISM;
- Modulação: GFSK;
- Emissão de energia: ≤ 4 dBm, Classe 2;
- Sensibilidade: ≤ -84 dBm com 0,1% BER;
- Velocidade Assíncrono: 2,1Mbps(Max)/160Kbps;
- Velocidade Síncrono: 1Mbps/1Mbps;
- Segurança: Autenticação e Encriptação;
- Perfil: Porta Serial Bluetooth;
- Suporta modo Escravo (Slave) e Mestre (Master);
- CSR chip: Bluetooth v2.0;
- Banda de Onda: 2,4Hz-2,8Ghz, Banda ISM;
- Tensão: 3,3v (2,7-4.2v);
- Corrente: Pareado 35mA; Conectado 8mA;
- Temperatura: -40 ~ +105°C;
- Alcance: 10m;

- BaudRate:
4800;9600;19200;38400;57600;115200;230400;460800;921600;1382400;
- Dimensões: 26,9 x 13 x 2,2mm;
- Peso: 9,6g;

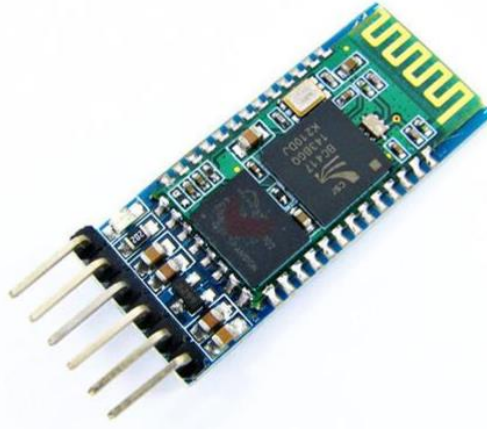


Figura 9: Módulo Bluetooth

Fonte: Autor (2020)

1.9. Sistema Operacional Android e Armazenamento

O Sistema Operacional Android é um sistema desenvolvido pela Google e outras empresas que formam o Open Handset Alliance. O desenvolvimento da plataforma é um projecto aberto onde vários fabricantes de dispositivos móveis podem modificar o sistema operacional. O Android pode ser utilizado por diversos dispositivos como, celulares, tablets relógios, televisores, geladeiras e centrais multimídia. À princípio ele foi criado para tirar proveito de telas sensíveis ao toque, com grande capacidade de processamento, boa conectividade e interação com dispositivos atuais. Com isso, é possível criar aplicativos tão complexos quanto os destinados a computadores pessoais (DE MELO EUZÉBIO, 2011).

O sistema operacional Android baseia-se no núcleo do sistema Linux para funcionar em dispositivos móveis. As aplicações desenvolvidas para esta plataforma são criadas utilizando-se a linguagem de programação Java ou Kotlin. O Kit de Desenvolvimento de Software para Android (Android SDK) inclui uma lista de ferramentas de desenvolvimento em linguagem Java, entre elas debugger, bibliotecas, emulador, documentação, códigos de exemplo e tutoriais (SMYTH, 2017a; ABLESON; KING; SEN, 2012) *apud* (BORGES, 2017). Os aplicativos gerados para o sistema Android consistem em arquivos em formato .apk e são armazenados no diretório /data/app do

sistema operacional. Esta pasta é acessível somente ao usuário root por razões de segurança. O sistema Android oferece várias opções para salvar dados de aplicativos de forma persistente, podendo-se considerar se os dados devem ser privados para a aplicação ou se devem ser acessíveis para outras aplicações (e para o usuário) e quanto espaço os dados devem consumir. As opções de armazenamento de dados são:

- **Preferências compartilhadas:** dados primitivos privados são armazenados em pares chave-valor. Cada par representa uma preferência de usuário em relação à aplicação, contendo uma chave e seu valor associado.
- **Armazenamento interno:** dados privados são mantidos na memória do dispositivo. Os arquivos salvos no armazenamento interno pertencem ao aplicativo e não podem ser acessados por outros aplicativos ou directamente pelo usuário.
- **Armazenamento externo:** dados públicos são mantidos na memória externa e podem ser vistos e compartilhados por vários usuários.
- **Bancos de dados SQLite:** possibilita o armazenamento de dados estruturados em um banco de dados relacional privado.
- **Conexão de rede:** provê o armazenamento de dados na web a partir de um servidor próprio de rede.

O Android SDK viabiliza a utilização das opções de armazenamento citadas, sendo que a escolha destas opções deve considerar as especificidades de tipo e uso das informações a serem mantidas. Para cada opção existem classes específicas dentro do Android SDK que viabilizam a implementação do tipo de armazenamento.

O programa utilizado na automação proposta neste TCC é o Android Studio, sendo compatível com Android 1.6 ou superior.

1.10. SQLite

O SQLite é composto de uma biblioteca que implementa um banco de dados SQL auto-suficiente, sem servidor ou configuração e transaccional. É um mecanismo de banco de dados SQL incorporado, visto que não possui um processo de servidor em separado. Ele é completo podendo ter várias tabelas, índices, gatilhos (triggers) e visões, e isso tudo estando contido em um único arquivo de disco, o que otimiza o gasto com espaço em memória, podendo facilmente ser embarcado em dispositivos como telefones celulares, PDAs e MP3. (BORGES, 2017).

Quando comparado com outros "formatos de arquivo de aplicativo" tem vantagens convincentes sobre os mesmos, como transacções ACID, linguagem de consulta de alto nível, actualizações incrementais e contínuas (quando somente o que foi alterado é actualizado na memória), capacidade de manipulação de grande base de dados (podendo chegar a terabytes), interface com varias linguagens como Python, Ruby, Delphi, Java, Javascript, Lua, .NET, PHP, entre outras. (BORGES, 2017).

1.11. Diagramas UML

A UML (Unified Modeling Language – Linguagem de Modelagem Unificada) tornou-se, nos últimos anos, a linguagem-padrão de modelagem adoptada internacionalmente pela indústria de engenharia de software. Em decorrência disso, existe uma demanda por profissionais que dominem essa linguagem. Entretanto UML é uma linguagem de modelagem totalmente independente, não estando vinculada a nenhum processo de desenvolvimento específico e menos ainda a qualquer linguagem de programação (Guedes, 2011).

1.12. Fritzing

Fritzing é uma iniciativa de software open-source para auxiliar no desenvolvimento de protótipos de electrónica em geral. Assim como o Arduino, Fritzing é indicado para qualquer um que tenha interesse em computação física e prototipagem.

Com ele pode-se “montar” o protótipo de forma fácil e agradável em uma protoboard virtual e depois alterar o modo de visualização para o diagrama esquemático e até mesmo o layout da placa de circuito impresso (PCB – printed circuit board).

1.13. Trabalhos correlatos

Um trabalho correlato é aquele que traz uma correlação com o trabalho principal aqui apresentado. Entre muitos trabalhos semelhantes a alguns que no ponto de vista desse autor, trouxeram excelentes exemplos e até mesmo ideias novas, como é o caso do: Accionamento remoto de portões eléctricos via celular através de micro controlador (Maia, 2012), que permite o accionamento directo de uma fechadura electrónica através de uma ligação telefónica. O usuário cadastra um número de telefone através de mensagem de texto do tipo SMS, e a partir deste cadastro, quando o sistema receber uma ligação do número previamente cadastrado, irá accionar directamente a fechadura. Segue abaixo uma das imagens do trabalho correlato.



Figura 10: Protótipo Projecto Maia

Fonte: <http://repositorio.uniceub.br/bitstream/123456789/3104/3/20855679.pdf>

Um segundo trabalho correlato muito interessante é o: Desenvolvimento de uma solução para automação residencial usando a plataforma android e arduino (Silva, 2013). Diferente do trabalho anterior o controle é feito através de uma rede sem fio do tipo WPAN, que utiliza o Bluetooth na placa Arduino. Bluetooth é um protocolo padrão de comunicação sem fio de baixíssimo custo e com baixo consumo de energia, utilizado em pequenas distâncias. A seguir uma imagem do trabalho com composto pela placa Arduino, o módulo Bluetooth, uma bateria para alimentação do relé de 12 volts, a lâmpada a ser accionada e o módulo relé, composto por um transístor, um resistor, um díodo e um relé.

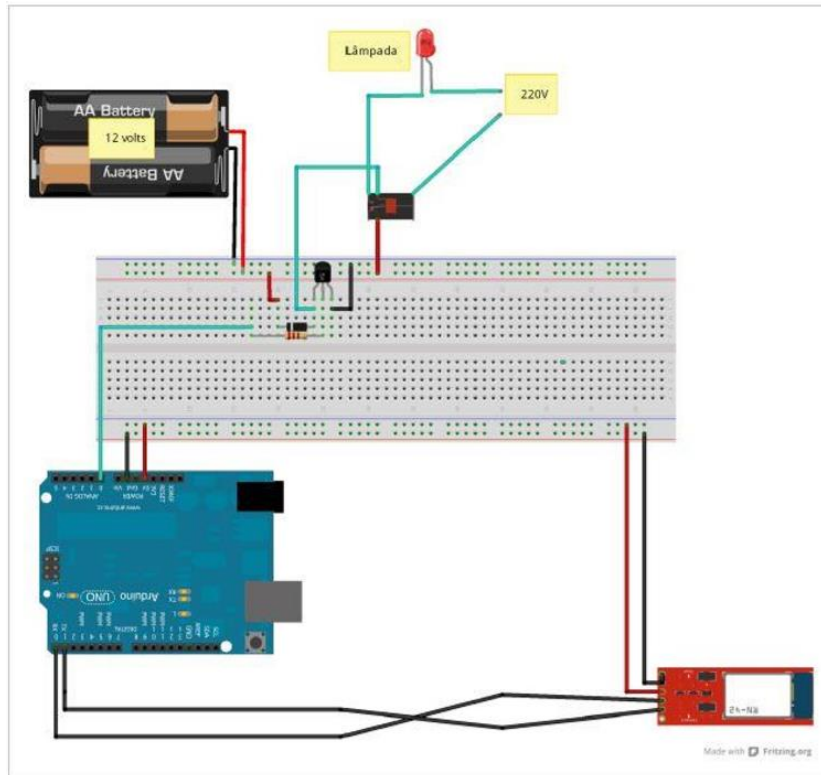


Figura 11:Trabalho correlato Arduino, protoboard

Fonte: Silva (2012)

1.14. CONCLUSÃO DO CAPÍTULO

Resumidamente, este capítulo permitiu fazer uma análise sobre a tendência da automação residencial conjugada com o avanço tecnológico acelerado. Tendo-se constatado muita falta de informação para subsidiar os desenvolvedores de sistemas inteligentes baseados em redes sensoriais, visto que é uma área ainda em exploração, para o caso concreto de Moçambique a automação residencial é ainda um conceito desconhecido.

CAPÍTULO II - DESENVOLVIMENTO DO SISTEMA

Neste capítulo serão apresentados e explicados todos os passos necessários para o desenvolvimento do sistema.

2.1. Descrição do sistema

O presente sistema de automação residencial é concebido para realizar a gestão dos seguintes sistemas domésticos: segurança, climatização e iluminação. Para interagir com o sistema o utilizador dispõe de uma aplicação Android (que deve ser instalada num smartphone), que oferece um conjunto de interfaces gráficas para a monitorização e controlo remoto e ou rede de telefonia celular.

Este sistema tem dois modos de funcionamento: o modo local e o modo remoto. No modo remoto, sempre que o utilizador precisa interagir com o sistema, recorre ao telemóvel que deve possuir um cartão SIM com acesso a rede celular, para poder enviar as instruções de controlo ou monitorização ao Arduino. A aplicação permite agendar uma hora específica para accionar os sistemas, visualizar o seu estado actual, e receber notificações sobre situações indesejadas tais como: A detecção de intrusão e uso de aparelhos em condições inadequadas (por exemplo, luzes exteriores ligadas durante o dia). O sistema anti-intrusão é baseado em sensores de movimento, que sinalizam situações de perigo na residência sob protecção. Quando a intrusão for decretada, acciona-se uma sirene e simultaneamente envia-se uma notificação ao utilizador através do GSM, o sistema permite também que em casos de incêndio accione sirene de alerta do perigo, e automaticamente desliga-se os ductos de gás. Alternativamente, pode-se usar o sistema no modo local através da aplicação Android para accionar os sistemas de iluminação e climatização, em situações em que o utilizador encontra-se na residência. Os sistemas e aparelhos a serem controlados são definidos pelo utilizador através do aplicativo.

Este sistema foi desenvolvido de modo a permitir que outros sistemas ou aparelhos (sistema de detecção de incêndio, sistema de portão automático, sistema de rega automática, entre outros) sejam integrados com facilidade de forma a estender as suas funcionalidades consoante as necessidades do utilizador.

2.2. Arquitectura do sistema

O sistema é constituído de três blocos que interagem entre si para executar tarefas, nomeadamente: O smartphone, a unidade de controlo, e a unidade de execução

(constituído de sensores, circuitos de controlo e monitorização). O smartphone actua como interface do utilizador no sistema, através do envio de instruções através aplicação ou do GSM para a unidade de controlo.

A unidade de controlo é composta do Arduino e os módulos Bluetooth e SIM 800L. Este bloco é responsável por comunicar com sensores, accionar dispositivos, e emitir notificações sobre eventos do sistema.

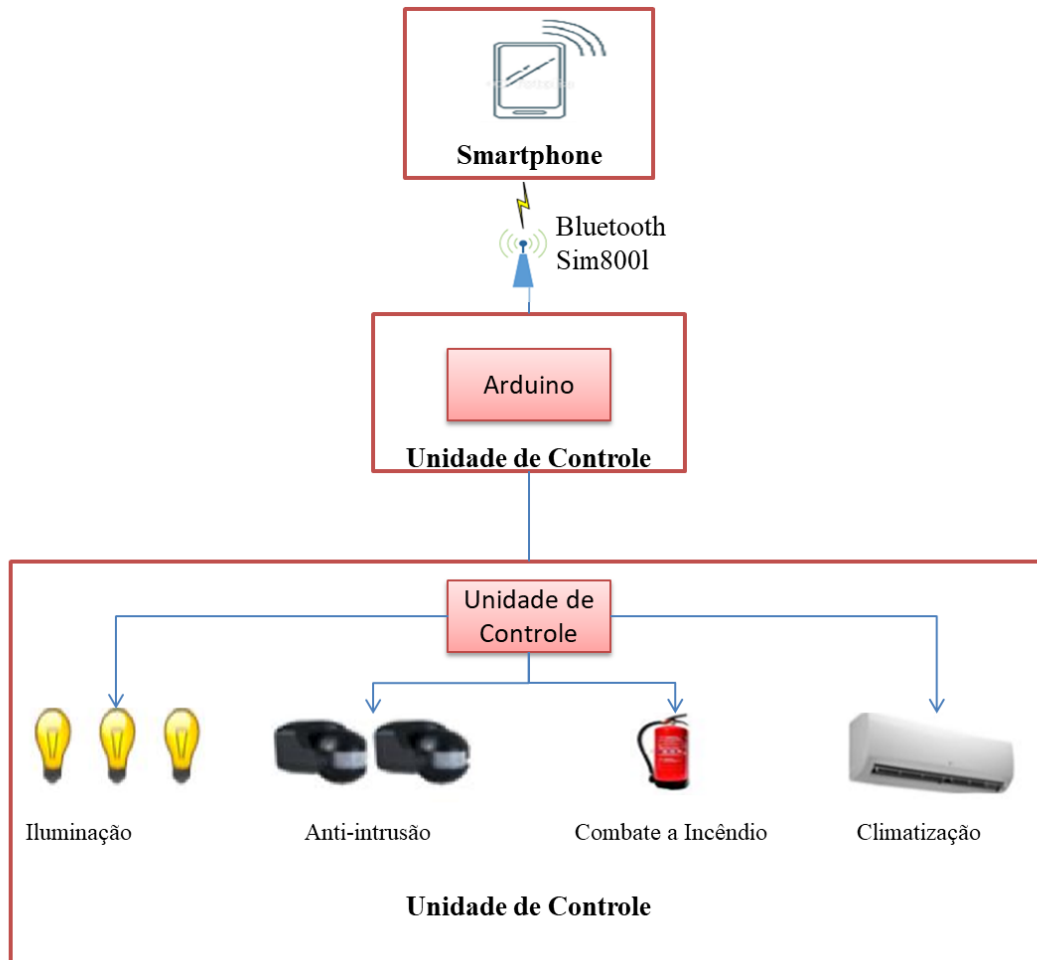


Figura 12: Arquitectura do sistema

Fonte: Autor (2020)

2.3. Etapas de desenvolvimento do sistema

A proposta deste projecto é a criação de um protótipo para a simulação de uma residência com sistema de anti-intrusão, iluminação, combate a incêndio e de climatização, ambos sendo ligados e desligados de acordo com o accionamento com um *smartphone*. Para que o protótipo fosse concebido foram levadas a cabo as seguintes etapas:

Etapa	Tarefa
1	Especificação e esquematização do sistema
2	Desenvolvimento da aplicação Android
3	Montagem do Maquete e circuitos de conexão com os dispositivos
4	Conexão dos dispositivos com ao Arduino
5	Desenvolvimento do Firmware do sistema (codificação do Arduino) e conexão com aplicação Android
6	Testes do Sistema

Tabela 4: Representação das etapas do desenvolvimento do projecto

Fonte: Autor (2020)

2.4. Especificação de requisitos

A especificação de Requisitos é uma actividade complexa que visa compreender as necessidades do utilizador e o que ele deseja que o sistema a ser desenvolvido realize. É a fase em que várias técnicas como entrevistas, questionários, prototípico, investigação de documentos, observação, são usadas com o objectivo de descobrir, articular e entender a organização e como forma de auxiliar o engenheiro de requisitos na compreensão do domínio do sistema e dos processos de negócios, buscando sempre levar em consideração a opinião dos diferentes *stakeholders*⁹ envolvidos no processo, as oportunidades e as restrições.

Os requisitos de *software* segundo SOMMERVILLE (2011) são classificados de duas maneiras:

- Requisitos funcionais;
- Requisitos não funcionais.

Os **Requisitos funcionais (RF)**, indicam as funções e serviços oferecidos pelo sistema ou o que se espera que o mesmo faça mediante o processamento a ser efectuado, quer

⁹*Stakeholders* é um termo utilizado para referir a qualquer pessoa que terá alguma influência directa ou indirecta sobre os requisitos do sistema, dentre os quais destacam-se os utilizadores finais que interagirão com o sistema e todo o pessoal, em uma organização, que venha a ser por ele afectado como os engenheiros, os gerentes de negócios, os programadores, analistas, projectistas entre outros.

seja através da entrada ou saída de informação em determinadas ocasiões, e podem variar entre requisitos gerais aos mais específicos.

Os **Requisitos não funcionais (RNF)** descrevem restrições sobre as funcionalidades oferecidas pelo sistema como um todo, não estando directamente relacionados com serviços específicos oferecidos aos utilizadores. Podem ser classificados de diversas maneiras dependendo das restrições a que o sistema estará sujeito como: portabilidade, desempenho, tempo de resposta, conformidade, uso de recursos.

Para a concepção do presente sistema foram consideradas as seguintes **Especificações Funcionais**:

Código	Descrição
RF01	Permitir autenticar o utilizador através de “ login” com nome e senhas introduzidos na aplicação;
RF02	Permitir accionar os sistemas domésticos;
RF03	Permitir agendar tarefas para accionar os sistemas domésticos;
RF04	Permitir visualizar o estado actual dos sistemas domésticos;
RF05	Enviar notificações ao utilizador sobre os eventos do sistema GSM.
RF06	Permitir o controlo manual dos sistemas domésticos;
RF07	Permitir detectar a intrusão, activar a sirene e simultaneamente notificar o utilizador através de GSM.
RF08	Permitir o recarregamento automático da fonte;

Tabela 5: Descrição dos Requisitos Funcionais do Sistema

Fonte: Autor (2020)

E para permitir que os requisitos funcionais tenham uma boa qualidade, desempenho e disponibilidade durante o uso do sistema foram considerados as seguintes **Especificações Não Funcionais**:

Código	Descrição
RNF01	Conceber o sistema de tal modo que seja escalável
RNF02	Permitir que o sistema seja flexível, isto é rápido em dar resposta quando ocorrer um accionamento
RNF03	Permitir que a data do microcontrolador esteja sempre certa, para que o usuário possa fazer agendamentos de eventos.

Tabela 6: Descrição dos Requisitos Não Funcionais do Sistema

Fonte: Autor (2020)

2.5. Construção da Maquete

Para a implementação e demonstração do projecto fez-se por necessário o uso de uma residência popular em escala (maquete). Segundo (NACCA, 2006, p. 15) “Maquete é a reprodução fiel de uma obra ou projecto em escala reduzida. A maquete tem o compromisso de reproduzir visualmente um objecto em escala reduzida.”.

Levando em consideração a visualização tridimensional, a possibilidade de demonstrar na pratica o funcionamento dos elementos aplicados na elaboração do projecto e a praticidade de sua construção, aliados ao baixo custo, justificam a escolha deste método para melhor representar a domótica aqui aplicada.

Se tratando da importância da maquete enfatiza Sarmiento, (2008 apud MOSCH, 2009, pg 123),

“A maquete é uma ferramenta educacional e um instrumento de descobertas. Gera informações em tempo semelhante ao real possibilitando a observação e a vivência de formas do espaço físico. A imagem do modelo tridimensional traz em si características essenciais do objecto de projecto e possibilita uma relação directa entre o observador leigo e o projecto auxiliando a percepção e a diferenciação da escala horizontal e vertical.

Julgou-se por necessário a elaboração de uma maquete referente a uma residência popular pois a domótica aplicada a mesma tem por objectivo principal a demonstração da acessibilidade, a popularização e a fácil disponibilidade dos itens aplicados a esta tecnologia, agregados ao baixo custo para sua implementação, antes

disponível apenas a uma fatia específica da sociedade, que comumente associava a domótica a uma tecnologia luxuosa e de grande valor agregado.

A residência miniaturizada possui cinco cômodos e uma área de jardim onde em todos eles foram aplicados algum processo de automação residencial onde facilmente podem ser aplicados em uma residência real com poucas alterações no escopo do projecto aqui demonstrado, conforme ilustrado na figura abaixo:

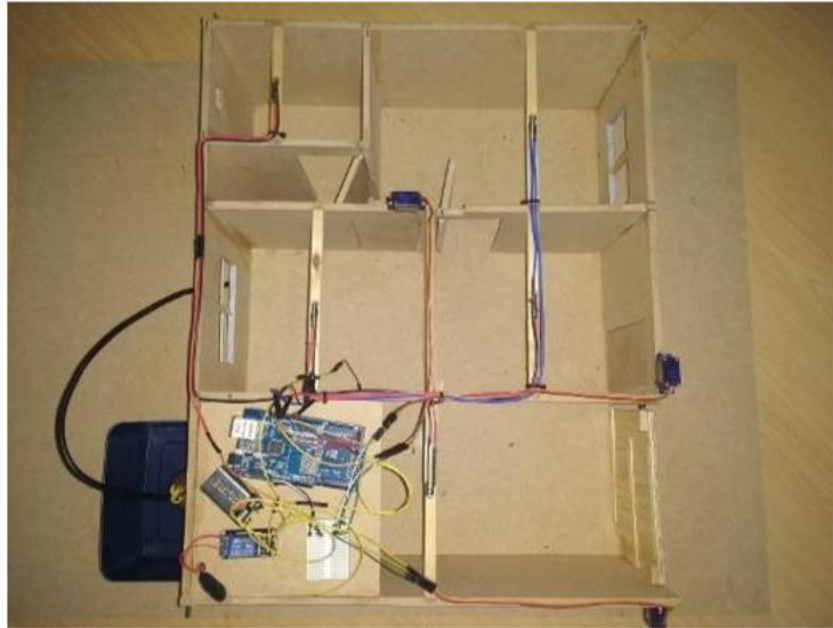


Figura 13: Vista Superior da Maquete

Fonte: Autor (Autor)

2.6. Hardware

Os Principais componentes de hardware são os seguintes:

- Arduino UNO;
- Módulos Bluetooth e Sim800L;
- Módulo RTC;
- Sensores PIR;
- Sensor de Temperatura e Humidade;
- Sensor de Gás e Fumaça;
- LDR;
- Lâmpada 220v, Sirene, Fonte de Alimentação e Relê;

Os circuitos que funcionam com tensões elevadas (o caso de sistema de iluminação que trabalha com tensões de 220V) ou altas correntes não podem ser controlados

directamente por Arduino. Para contornar isso, usa-se um sinal de controlo de baixa tensão do Arduino para controlar um relé, que é capaz de lidar e comutar circuitos de alta tensão (Writers,2016). Usou-se também uma fonte de alimentação externa (VCC) de 9V para ligar o Arduino, e os dispositivos acústicos. Para permitir que usuário possa agendar uma determinada tarefa usou-se o RTC. Para proporcionar a melhor percepção sobre a conexão dos dispositivos electrónicos faz-se uma ilustração esquemática em **Apêndice II**.

2.6.1. Instalações eléctricas e componentes

Para a instalação do comando de uma lâmpada foi utilizado um relé em paralelo com o interruptor. Caso, por algum motivo, o sistema não esteja respondendo, o usuário ficará livre para utilizar o sistema da forma convencional, pois o relé de contacto funciona em paralelo com um dos contactos normalmente abertos e outro normalmente fechado do interruptor e, quando accionado, os mesmos se invertem. O diagrama da instalação pode ser visto na Figura abaixo.

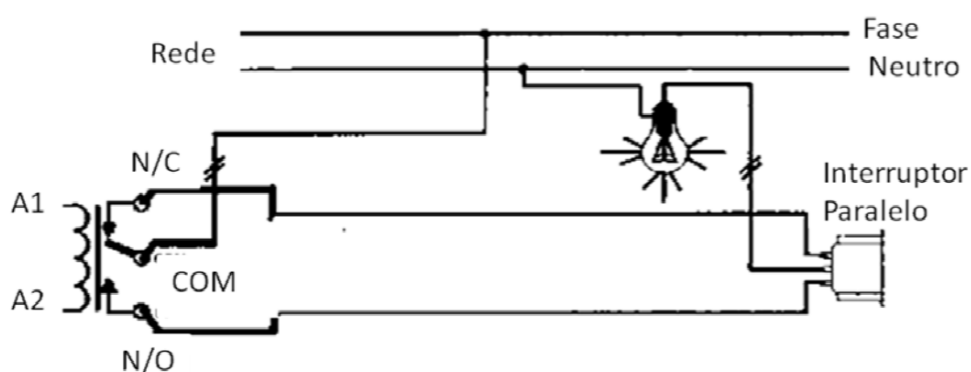


Figura 14: Diagrama de funcionamento do relé.

Fonte: Jorge (2016)

Para accionar o relé foi projectado um circuito eléctrico de acordo com o modelo da Figura abaixo. A Equação utilizada para dimensionar o resistor foi para um transístor BC547 que tem sua base excitada com corrente 30 mA, conforme especificado pelo manual do fabricante.

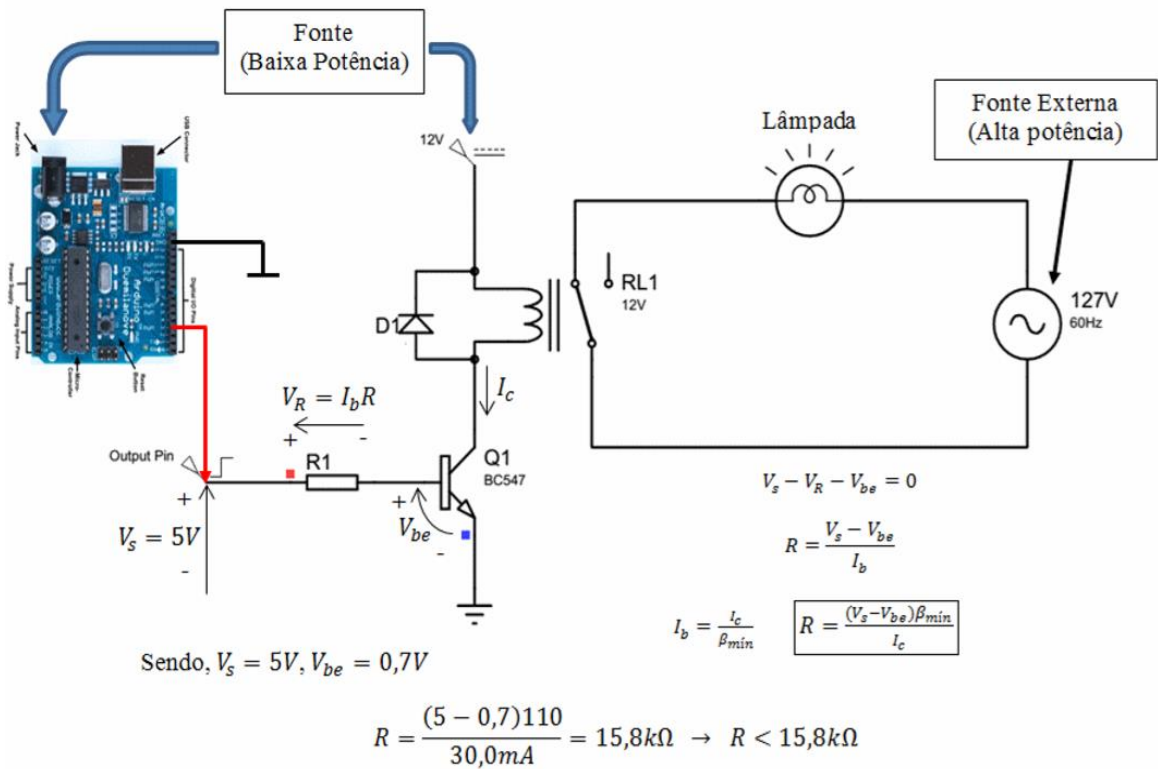


Figura 15: Accionamento do Relé

Fonte: Guilherme (2013)

2.6.2. Divisor de tensão

Um divisor de tensão¹⁰ (também conhecido como divisor de potencial) é um circuito linear passivo que produz uma tensão de saída (V_{out}) que é uma fração de sua tensão de entrada (V_{in}). A divisão de tensão é o resultado da distribuição da tensão de entrada entre os componentes do divisor. Um exemplo simples de um divisor de tensão são dois resistores conectados em série, com a tensão de entrada aplicada no par de resistores e a tensão de saída emergindo da conexão entre eles.

Os divisores de tensão do resistor são comumente usados para criar tensões de referência ou reduzir a magnitude de uma tensão para que possa ser medida e também pode ser usada como atenuador de sinal em baixas frequências. Para corrente directa e frequências relativamente baixas, um divisor de tensão pode ser suficientemente preciso se for feito apenas de resistores; onde é necessária uma resposta de frequência em uma ampla faixa (como em uma sonda de osciloscópio), um divisor de tensão pode ter elementos capacitivos adicionados para compensar a capacitância da carga. Na

¹⁰ https://en.wikipedia.org/wiki/Voltage_divider

transmissão de energia eléctrica, um divisor de tensão capacitivo é usado para medição de alta tensão.

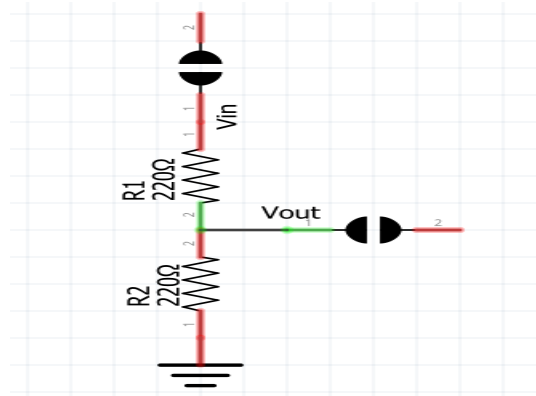


Figura 16: Circuito de divisor de Tensão

Fonte: Autor (2020)

Uma vez que temos duas resistências associadas em série temos a seguinte situação:

$$V_{in} = \frac{R_1}{R_2} \times I, \text{ Se a tensão é fixa } (V_{in} = 5V), \text{ temos: } I = \frac{5V}{(R_1 + R_2)} \text{ Onde: } I - \text{ corrente}$$

Neste caso, a corrente é o resultado da divisão da tensão 5V pela soma das resistências.

Nesses circuitos temos as mesmas correntes para os dois resistores mas temos uma tensão diferente em cada um. Esse circuito é chamado de divisor de tensão. Como a corrente do circuito é calculada pela divisão da tensão total pela soma dos resistores, teremos a tensão em cima de um resistor igual a resistência desse resistor vezes a tensão total dividida pela soma dos resistores. Este conceito é aplicável para o LDR.

2.6.3. Particularidades do LDR

Uma vez que o LDR, é uma resistência foto-variável, usou-se da propriedade do divisor de tensão para medir a variação da queda de tensão em cima do mesmo. Sabemos que a tensão total e a resistência total são fixas. Dessa forma, o divisor de tensão vai variar com a resistência entre A0 e GND.

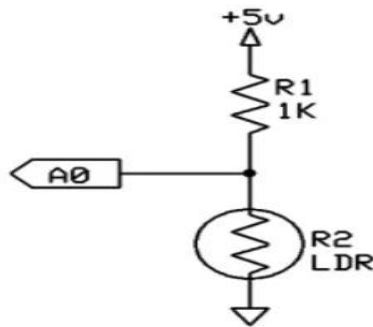


Figura 17: Circuito da Leitura do LDR

Fonte: <https://portal.vidadesilicio.com.br/sensor-de-luz-com-ldr/>

Levando em conta que quanto menos luz incidir sobre o LDR maior será sua resistência, teremos a tensão sobre o **LDR** e por conseguinte o valor de **A0** maior com um índice de luminosidade incidente menor, isto é, num local mais escuro.

$$V_{A0} = \frac{5V}{1K + LDR} \times LDR$$

- Quanto menor luminosidade maior a resistência do LDR;
- Quanto maior a resistência do LDR, maior a tensão em A0;
- Então: quanto menor a luminosidade maior é a tensão em A0.

2.6.4. Fonte de alimentação e recarregamento automático

Foi utilizada uma alimentação externa (não USB) oriunda de uma bateria. A alimentação feita através dos pinos GND e VIN do conector de alimentação.

A placa pode operar com uma fonte externa de 6 a 20 volts. Se a fonte externa tiver voltagem menor que 7V, no entanto, o pino de 5V pode fornecer menos de cinco volts e a placa poderá ficar instável. Se usar mais do que 12V, o regulador de voltagem pode superaquecer e danificar a placa. A faixa recomendada para alimentação externa é de 7 a 12 volts.

Os pinos de alimentação da placa Arduino são os seguintes:

- **V_{IN}**: A entrada de tensão da placa Arduino, quando ela está usando uma fonte externa de energia (ao contrário de 5 volts a partir da conexão USB ou fonte de alimentação regulada).
- **5V**: A fonte de alimentação regulada usada para alimentar o microcontrolador e outros componentes na placa. Este pode vir do pino V_{IN} através de um regulador na placa, ou ser fornecido pelo USB ou outra fonte de regulada de 5V.

- **3,3V:** Uma fonte de 3,3 volts gerado pelo regulador da placa. A corrente máxima é 50 mA.
- **GND:** Pinos terra.

Para evitar que a bateria se esgote e paralise o funcionamento normal da aplicação, foi desenvolvido um **medidor de tensão**, conforme ilustrado no esquema na figura abaixo (Z=R - Resistência).

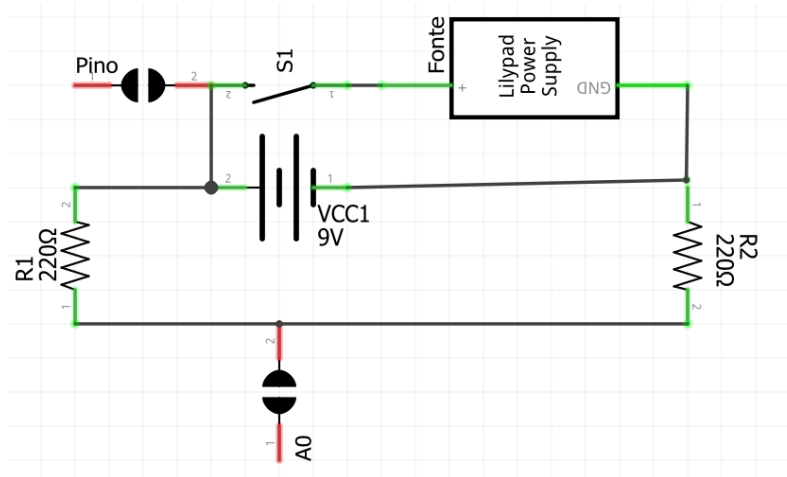


Figura 18: Esquema do medição de tensão e recarregamento automático

Fonte: Autor (2020)

Uma vez conectada a bateria no pino **A0** do Arduino, acontece o seguinte:

- Se o V_{in} lido da bateria for menor que o mínimo recomendado (7V), automaticamente acionará o interruptor para o recarregamento do mesmo;
- Se o V_{in} variar de 7V a 14V que é a tensão recomendada, o sistema estará em funcionamento normal;
- Se o V_{in} lido for igual a 14V, automaticamente será desligado o interruptor de recarregamento.

Para o efeito, usou-se uma bateria de **12V**, e com base no conceito de divisor de tensão, calculou-se a proporção dos resistores, usando a seguinte fórmula:

$$\frac{V_{out}}{V_{in}} = \frac{R2}{R1 + R2}$$

Primeiro, precisamos converter nosso valor analógico em tensão.

$$V_{out} = \text{analogRead}(A0) * (5.0/1023.0) // \text{reconhecido em programação do Arduino}$$

Depois de ter o valor do V_{out} é possível ter o V_{in} levando em conta os valores das resistências.

$$V_{in} = \frac{R_1 + R_2}{R_2} * V_{out}$$

Onde:

V_{in} – Tensão de Entrada;

V_{out} – Tensão de Saída;

R_1 e R_2 – Resistências.

2.7. Software

O sistema é constituído de dois softwares: A aplicação Android que será executada no smartphone e o firmware que será executado no Arduino. A aplicação contém um conjunto de interfaces gráficas que permitem o utilizador fazer a gestão dos sistemas domésticos, e não necessita de estar conectada constantemente ao Arduino, apenas quando for enviar alguma instrução. Foi desenvolvida usando o IDE Android Studio 2.0. O firmware, é composto de código fonte para executar as tarefas do sistema a nível do físico.

2.7.1. Aplicação Android

O desenho da aplicação baseou-se na linguagem de modelação UML. Os artefactos produzidos na fase de análise e desenho são apresentados abaixo.

2.7.1.1. Diagrama de caso de uso da aplicação

Neste ponto será apresentado o diagrama de caso de uso da aplicação e as descrições textuais dos casos de usos com o objectivo de detalhar como o utilizador interage com a aplicação, descrevendo as suas condições e fluxos.

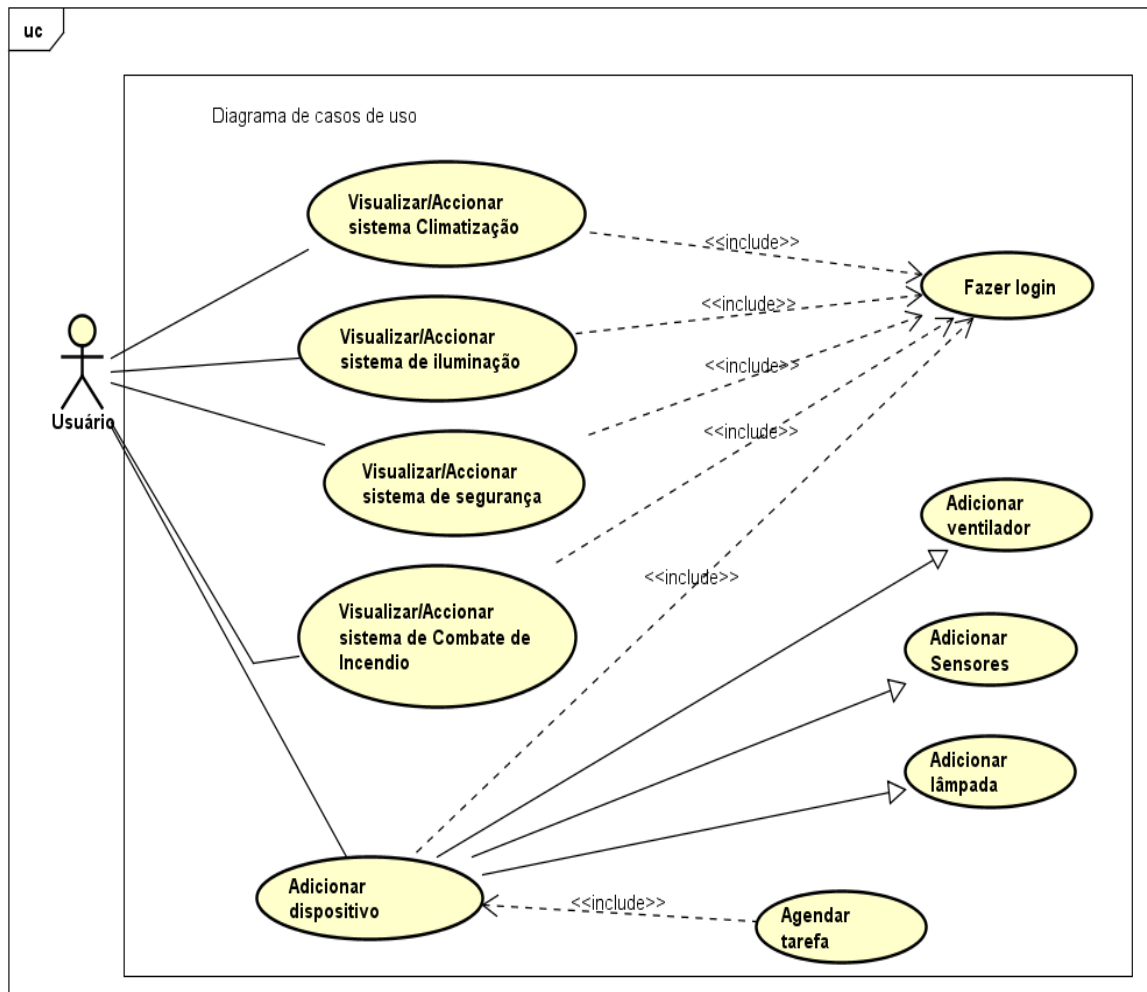


Figura 19: Diagrama de casos de uso da Aplicação

Fonte: Autor (2020)

Será especificado cada caso de uso, descrevendo a sequência de ações com o objetivo de demonstrar o comportamento da aplicação através de interações com os atores, apresentando também cenários alternativos ao caso de uso.

Casos de Uso: Efectuar Login					
Actores:	Utilizador				
Pré-condição:	Nenhuma				
Descrição:	Processo através do qual o utilizador fornece as suas credenciais de modo a usar as funcionalidades do sistema.				
	<table border="1"> <thead> <tr> <th>Ação do actor</th> <th>Resposta do sistema</th> </tr> </thead> <tbody> <tr> <td>1. Este caso de uso começa quando o utilizador executa a</td> <td>2. O sistema autentica o utilizador.</td> </tr> </tbody> </table>	Ação do actor	Resposta do sistema	1. Este caso de uso começa quando o utilizador executa a	2. O sistema autentica o utilizador.
Ação do actor	Resposta do sistema				
1. Este caso de uso começa quando o utilizador executa a	2. O sistema autentica o utilizador.				

Sequência de eventos	aplicação e introduz seu nome de utilizador e senha.	3. A aplicação exhibe a interface inicial da aplicação.
Sequência alternativa	O utilizador introduz credenciais erradas. Indicar erro	

Tabela 7: Descrição do caso de uso efectuar login

Fonte: Autor (2020)

Casos de Uso: Accionar as lâmpadas		
Actores:	Utilizador	
Pré-condição:	Efectuar login	
Descrição:	O utilizador usa a aplicação para ligar ou desligar lâmpadas.	
Sequência de eventos	Acção do actor	Resposta do sistema
	1. O utilizador acede o menu “sistema de iluminação”. 3. O utilizador selecciona a opção ligar ou desligar o sistema de iluminação consoante o estado actual apresentado.	2. A aplicação apresenta uma interface sobre o estado actual do sistema de iluminação (ligado ou desligado). 4. A aplicação envia instruções de ligar/desligar para o microcontrolador. 5. O Microcontrolador altera o estado do sistema de iluminação e envia uma confirmação para a aplicação, que por sua vez exhibe uma mensagem.
Sequência alternativa		A aplicação por alguma razão não comunica-se com o Microcontrolador. Indicar erro.

Tabela 8: Descrição do caso de uso accionar lâmpadas

Fonte: Autor (2020)

Casos de Uso: Agendar tarefa para accionar lâmpadas.		
Actores:	Utilizador	
Pré-condição:	Efectuar login	
Descrição:	O utilizador agenda uma hora específica para ligar ou desligar lâmpadas através da aplicação.	
Sequência de eventos	Acção do actor	Resposta do sistema
	1. O utilizador acede o menu “sistema de iluminação”. 3. O utilizador clica no menu geral e selecciona a opção agendar. 5. O utilizador selecciona o tipo de acção, e especifica a hora.	2. A aplicação apresenta uma interface gráfica sobre o estado actual do sistema de iluminação (ligado ou desligado). 4. A aplicação exhibe a interface gráfica com os campos necessários para agendar o tipo de acção e a hora. 6. A aplicação salva a informação e envia para o microcontrolador. 7. A aplicação exhibe uma mensagem.
Sequência alternativa		<ul style="list-style-type: none"> ➤ A aplicação por alguma razão não comunica-se com o Microcontrolador. Indicar erro. ➤ Quando não for possível enviar a informação, a aplicação aguarda 1h e tenta enviar novamente. ➤ Se houver alguma actividade agendada previamente, exhibir a informação na interface gráfica.

Tabela 9: Descrição do caso de uso agendar para accionar lâmpadas

Fonte: Autor (2020)

Casos de Uso: Agendar tarefa: accionar ventiladores.		
Actores:	Utilizador	
Pré-condição:	Efectuar login	
Descrição:	O utilizador agenda uma hora específica para ligar ou desligar ventiladores através da aplicação.	
Sequência de eventos	Ação do actor	Resposta do sistema
	1. O utilizador acede o menu “sistema de climatização”. 3. O utilizador clica no menu geral e selecciona a opção agendar. 5. O utilizador selecciona o tipo de acção, e especifica a hora.	2. A aplicação apresenta uma interface gráfica sobre o estado actual do sistema de climatização (ligado ou desligado). 4. A aplicação exhibe a interface gráfica com os campos necessários para agendar o tipo de acção e a hora. 6. A aplicação salva a informação e envia para o microcontrolador. 7. A aplicação exhibe uma mensagem.
Sequência alternativa		<ul style="list-style-type: none"> ➤ A aplicação por alguma razão não comunica-se com o Microcontrolador. Indicar erro. ➤ Quando não for possível enviar a informação, a aplicação aguarda 1h e tenta enviar novamente. ➤ Se houver alguma actividade agendada previamente, exhibir a informação na interface gráfica.

Tabela 10: Descrição do caso de uso agendar tarefa para accionar ventiladores

Fonte: Autor (2020)

Casos de Uso: Accionar o sensor de detenção de movimento		
Actores:	Utilizador	
Pré-condição:	Efectuar login	
Descrição:	O utilizador através da aplicação arma ou desarma o sistema de detenção de movimento não autorizado.	
Sequência de eventos	Ação do actor	Resposta do sistema
	1. O utilizador acede o menu “sistema de segurança”. 3. O utilizador selecciona a opção armar ou desamar consoante o estado actual apresentado.	2. A aplicação apresenta uma interface sobre o estado actual do sistema de detenção de movimento (armado ou desarmado). 4. A aplicação envia instruções de ligar/desligar para o Microcontrolador. 5. O Microcontrolador altera o estado do sistema de sistema de detenção de movimento e envia uma confirmação para a aplicação, que por sua vez exhibe uma mensagem.
Sequência alternativa		A aplicação por alguma razão não comunica-se com o Microcontrolador. Indicar erro.

Tabela 11: Descrição do caso de uso accionar o sistema de detenção de movimento

Fonte: Autor (2020)

Casos de Uso: Agendar tarefa para accionar o sistema de segurança.		
Actores:	Utilizador	
Pré-condição:	Efectuar login	
Descrição:	Agendar tarefa para accionar o sistema de segurança.	
	Ação do actor	Resposta do sistema
	1. O utilizador acede o	2. A aplicação apresenta uma

<p>Sequência de eventos</p>	<p>menu “sistema de segurança”.</p> <p>3. O utilizador clica no menu geral e selecciona a opção agendar.</p> <p>5. O utilizador selecciona a opção, o tipo de acção, específica a hora e clica em salvar.</p>	<p>interface sobre o estado actual do sistema de segurança (detecção de intrusão e detecção de movimento).</p> <p>4. A aplicação exhibe a interface com os campos necessários para agendar o tipo de acção e a hora. A interface contém informação sobre sistema de detecção de intrusão e detecção de movimento.</p> <p>6. A aplicação salva a informação e envia para o microcontrolador.</p> <p>7. A aplicação exhibe uma mensagem.</p>
<p>Sequência alternativa</p>		<ul style="list-style-type: none"> ➤ A aplicação por alguma razão não comunica-se com o Microcontrolador. Indicar erro. ➤ Quando não for possível enviar a informação, a aplicação aguardar 1h e tenta enviar novamente. ➤ Se houver alguma actividade agendada previamente, exhibir a informação na interface gráfica.

Tabela 12: Descrição do caso de uso accionar o sistema de detecção de movimento

Fonte: Autor (2020)

Casos de Uso: Accionar o Sistema de Combate de incêndio		
Actores:	Utilizador	
Pré-condição:	Efectuar login	
Descrição:	O utilizador através da aplicação liga ou desliga o sistema de controlo ou combate a um eventual incêndio.	
Sequência de eventos	Ação do actor	Resposta do sistema
	1. O utilizador acede o menu “Combate a Incêndio” 3. O utilizador selecciona a opção ligar ou desligar consoante o estado actual apresentado.	2. A aplicação apresenta uma interface sobre o estado actual do sistema de controlo ou combate a incêndio (ligar ou desligar). 4. A aplicação envia instruções de ligar/desligar para o Microcontrolador. 5. O Microcontrolador altera o estado do sistema de sistema de controlo de incêndio e envia uma confirmação para a aplicação, que por sua vez exhibe uma mensagem.
Sequência alternativa		A aplicação por alguma razão não comunica-se com o Microcontrolador. Indicar erro.

Tabela 13: Descrição do caso de uso accionar o sistema de combate a incêndio

Fonte: Autor (2020)

2.7.1.2. Diagramas de Sequência

O diagrama de sequência representa de maneira simples e lógica a sequência de mensagens que transitam entre os objectos no sistema. Em outras palavras, descreve a maneira como os grupos de objectos se comportam ao longo do tempo na aplicação, conforme se apresenta no **Apêndice III**. Ferramenta usada: Astah Professional.

Um aspecto muito importante sobre o diagrama de sequência é o facto de ele identificar o evento gerador do processo modelado, como também o actor responsável

por disparar esse evento. A sua construção é dependente dos diagramas de casos de uso e classes, por estes serem compostos por atores, objectos e mensagens.

2.7.1.3. Diagrama de actividades

Um diagrama de actividades é uma extensão de um diagrama de estados, que tem por objectivo demonstrar o fluxo de controlo entre actividades de um único processo, podendo ainda ser utilizado na descrição de um fluxo de actividades mais alargado. Nele são identificados os objectos responsáveis pela realização de cada uma das actividades. É caracterizado por conter estados de acção, estados de actividade, transições e objectos para a sua criação, tendo a capacidade de modelar processos em diversos níveis conceituais, dando uma “visão simplificada do fluxo de controlo de uma operação ou de um processo de negócio”

Este diagrama é do tipo comportamental, sendo ele um dos diagramas que fazem parte da documentação da aplicação móvel, tendo auxiliado na observação de quais actividades serão executadas no decorrer do seu funcionamento.

O diagrama de actividades em **Apêndice V**, mostra a sequência das actividades entre o Utilizador, a Aplicação Android e o Microcontrolador, após efectuar o login.

2.7.2. Fluxograma de visualização do estado dos dispositivos ou aparelhos

No sistema, todos dispositivos e aparelhos que pretende-se monitorar tem estados: Ligado ou desligado para o caso das lâmpadas, AC, ventiladores, e sistema de combate a incêndio. Armado, desarmado, ou alarme para os sensores. Quando aberta, a aplicação mostra o último estado obtido, no entanto, o utilizador pode actualizar o estado dos dispositivos ou equipamentos, se estiver ligado a rede. A figura abaixo ilustra os passos necessários para a visualização do estado dos dispositivos.

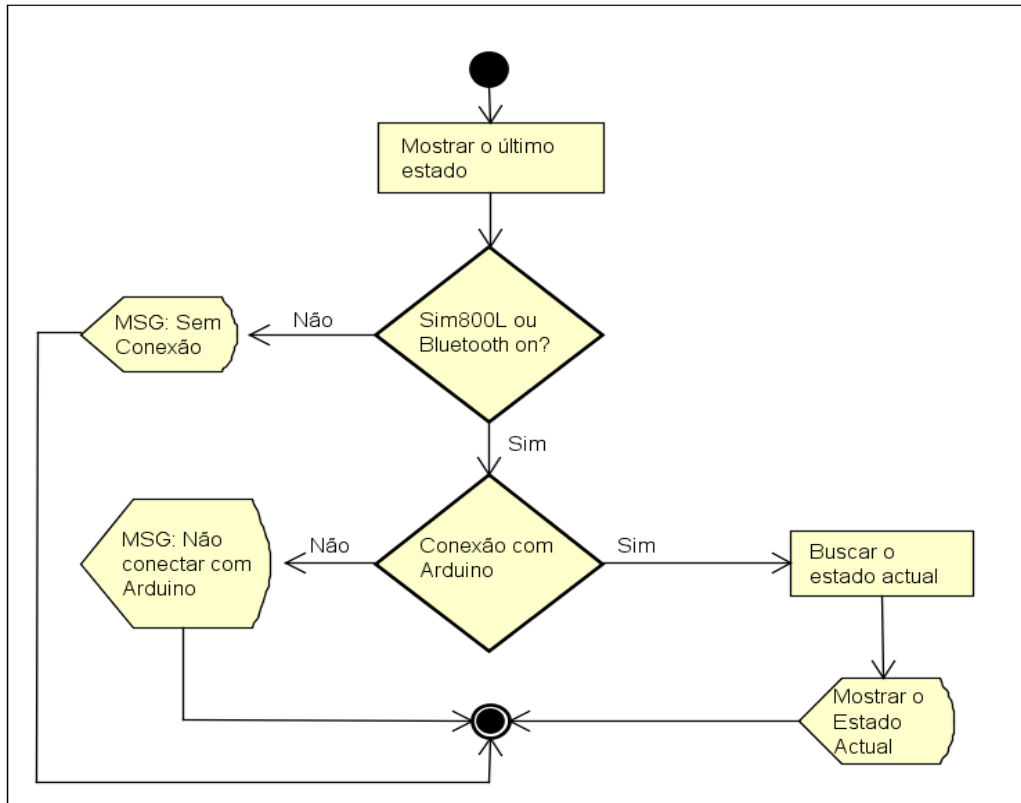


Figura 20: Fluxograma de visualização do estado dos dispositivos

Fonte: Autor (2020)

2.7.3. Fluxograma de accionamento de dispositivos ou aparelhos

No contexto deste sistema, accionar um dispositivo significa liga-lo ou desliga-lo para o caso das lâmpadas, AC, ventiladores, sistema de combate a incêndio e armar ou desarmar para o caso dos sensores. Antes de accionar um dispositivo o sistema verifica o seu estado actual, para poder permitir a acção contrária, isto é, se o dispositivo estiver ligado ou armado então só podemos desliga-lo ou desarma-lo e vice-versa. O accionamento através da aplicação só é possível se a *smartphone* estiver ligado ao Bluetooth. A **Apêndice IV** ilustra os passos para accionar um dispositivo ou aparelho.

2.7.4. Fluxograma de agendamento de tarefas

Em certas situações o utilizador pode desejar agendar uma tarefa, por exemplo, ligar ou desligar lâmpadas numa determina data e hora. Quando o *smartphone* estiver ligado ao Bluetooth, a aplicação envia a informação sobre o agendamento da tarefa para o Arduino, e este encarrega-se de executa-la na data e hora programada. A figura abaixo ilustra o procedimento para agendamento de tarefas.

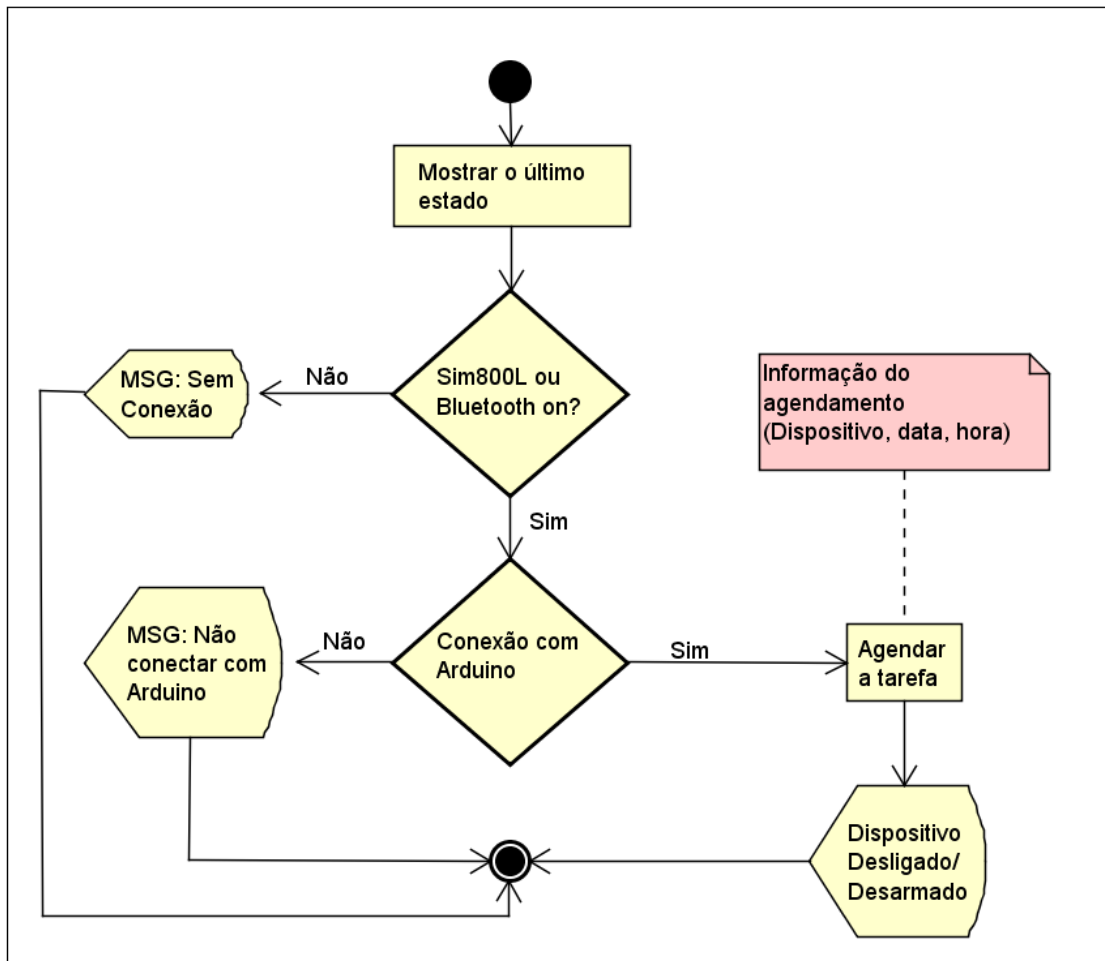


Figura 21: Fluxograma de agendamento de tarefas

Fonte: Autor (2020)

2.7.5. Fluxograma do funcionamento do Subsistema de Climatização

Inicialmente o usuário liga o subsistema de climatização através do aplicativo Android, assim que estiver ligado o Arduino fica a gerenciar os sensores de climatização, quando este estiver a ler uma temperatura acima da temperatura ambiente pode de forma automática ligar os dispositivos de climatização (ventilador ou AC), sem a necessidade de o usuário fazer nenhum clique. A seguir é ilustrado o procedimento de funcionamento do subsistema de climatização.

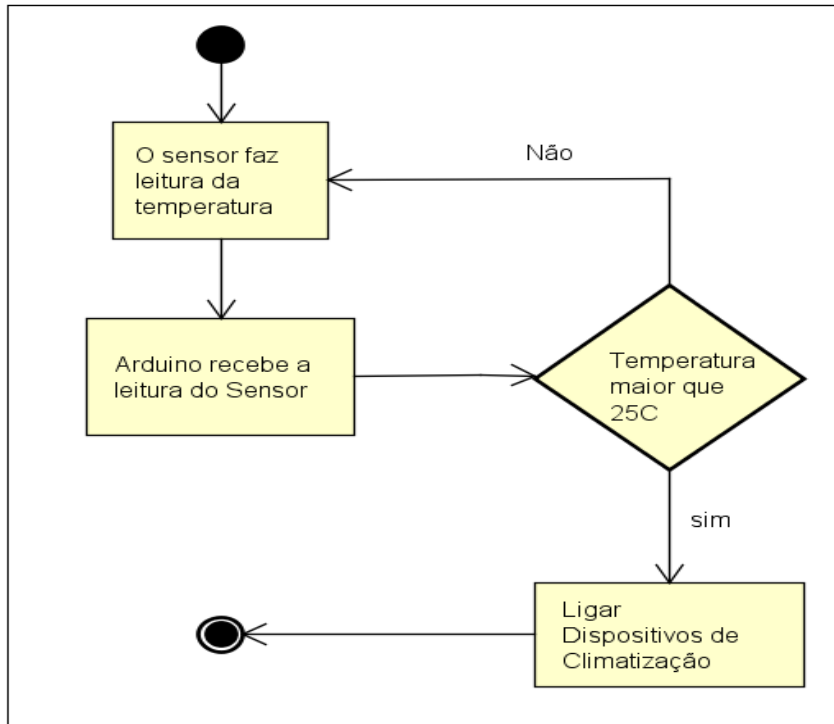


Figura 22: Fluxograma de funcionamento do subsistema de climatização

Fonte: Autor (2020)

2.7.6. Fluxograma do funcionamento do Subsistema de iluminação

Inicialmente o usuário liga o subsistema de Iluminação através do aplicativo Android, assim que estiver ligado o Arduino fica a gerenciar os sensores de luminosidade, quando este estiver a ler valores acima do valor previamente determinado (nível de luminosidade) automaticamente liga as lâmpadas, sem a necessidade de o usuário fazer nenhum clique. A seguir é ilustrado o procedimento de funcionamento do subsistema de Iluminação.

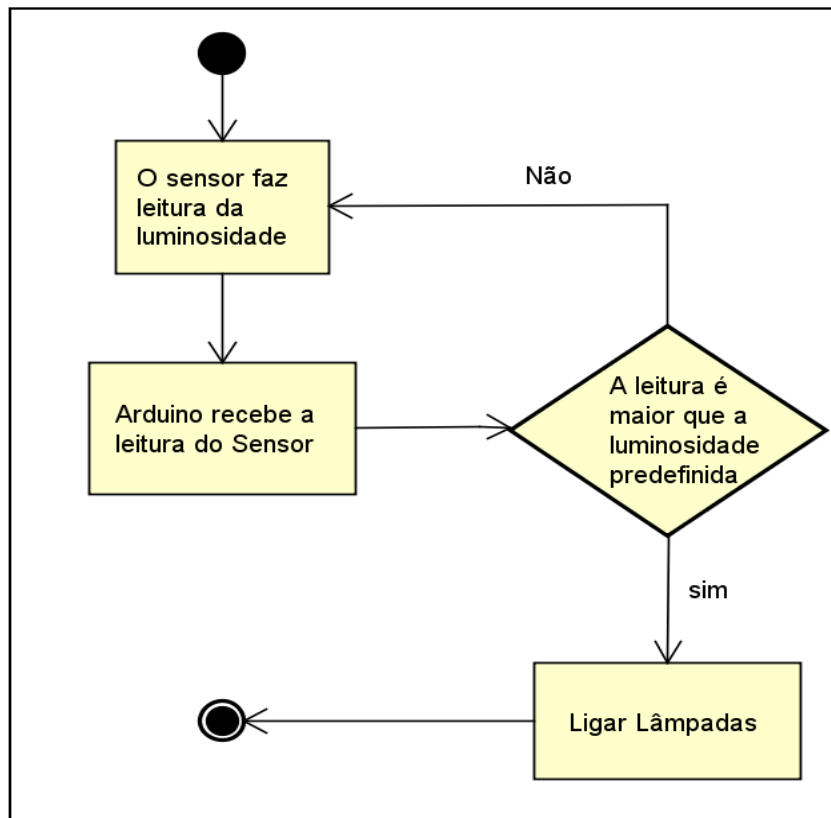


Figura 23: Fluxograma de funcionamento do subsistema de Iluminação

Fonte: Autor (2020)

2.7.7. Fluxograma do funcionamento do Subsistema de Anti-intrusão

Inicialmente o usuário liga o subsistema de **Anti-intrusão** através do aplicativo Android, assim que estiver ligado o Arduino fica a gerenciar os sensores de detecção de presença (PIR), quando este estiver a ler valor lógico (1) ou HIGH automaticamente liga as Sirenes e liga para um numero previamente configurado a nível do firmwere, sem a necessidade de o usuário fazer nenhum clique. A seguir é ilustrado o procedimento de funcionamento do subsistema de Anti-intrusão.

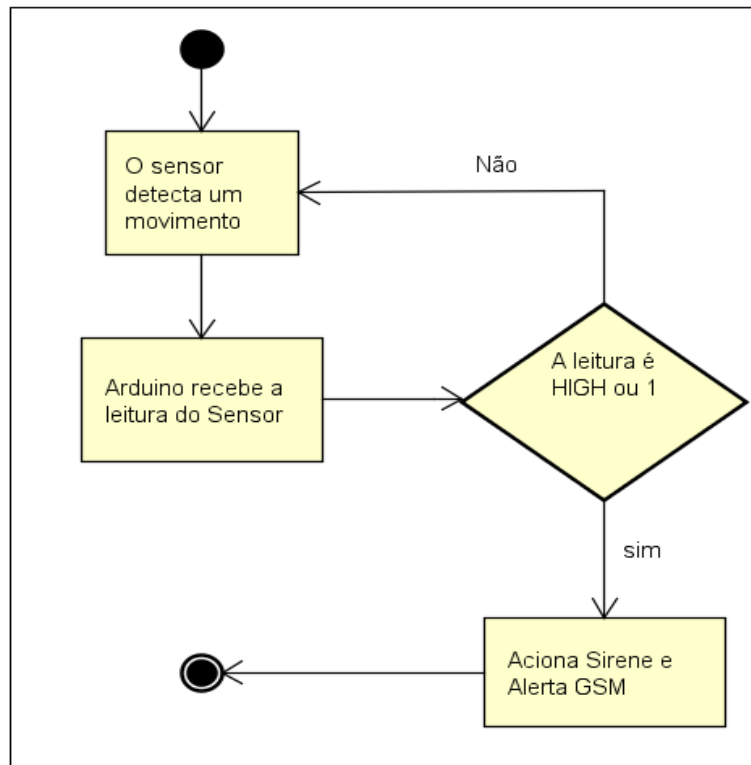


Figura 24: Fluxograma de funcionamento do subsistema de Anti-intrusão

Fonte: Autor (2020)

2.7.8. Fluxograma do funcionamento do Subsistema de Controlo de Incêndio

Inicialmente o usuário liga o subsistema de controlo de incêndio através do aplicativo Android, assim que estiver ligado o Arduino fica a gerenciar os sensores de deteção de fumaça, quando este estiver a ler valor lógico (1) ou HIGH automaticamente executa as seguintes operações;

- Abertura de portas e janelas;
- Fechamento de ductos de ar e gás;
- Avisos acústicos;
- Envio de mensagens e alertas via rede telefónica;

A seguir é ilustrado o procedimento de funcionamento do subsistema de controlo ou combate a incêndio.

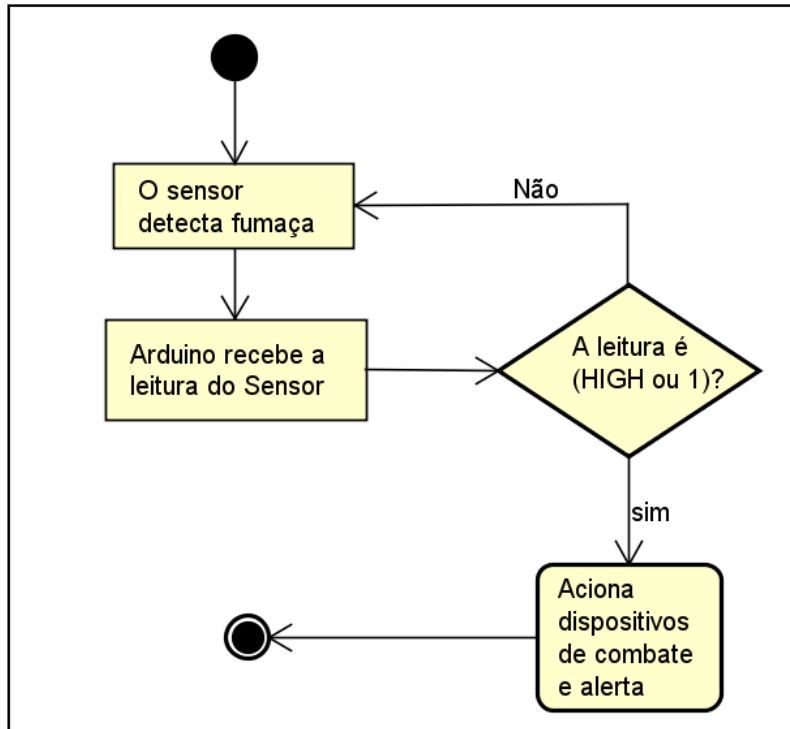


Figura 25: Fluxograma de funcionamento do subsistema de controlo de incêndio

Fonte: Autor (2020)

2.8. INTERFACE DO UTILIZADOR (APLICAÇÃO ANDROID)

Este ponto mostra as telas de interação do utilizador e o sistema de accionamento residencial a partir da aplicação android, neste caso a aplicação será útil para configurar o sistema, fazer agendamento de tarefas e consultar estado. O funcionamento do sistema independe do aplicativo android, uma vez que o sistema proposto trata de automação residencial controlado a partir do Arduino Uno.

Tela de Autenticação

Ao Executar a Aplicação, lhe será aberta a primeira tela da aplicação referente a autenticação do usuário. Após a habilitação da tela o usuário entrará com dados de acesso previamente cadastrados correspondentes ao nome do usuário e a senha do mesmo, conforme descrito na figura abaixo.

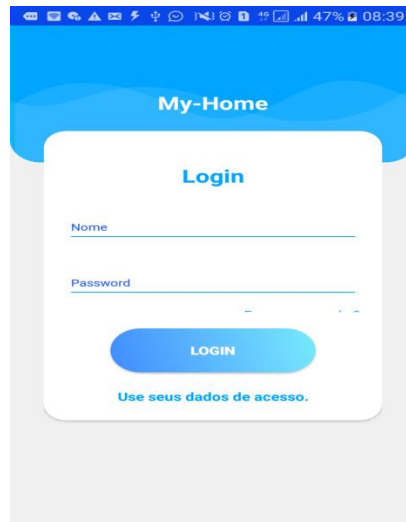


Figura 26: Tela de Autenticação

Fonte: Autor (2020)

Ao primar-se o botão login, abre-se uma nova tela onde são encontrados todos dispositivos Bluetooth emparelhados, incluindo o próprio dispositivo em uso que é o módulo HC-05, com capacidade de funcionar tanto escravo quanto mestre.

Quando o utilizador inserir credenciais incorrectas, a aplicação exhibi uma mensagem de erro informando que os credenciais de acesso estão incorrectos, fazendo com que o utilizador permaneça na tela de login até inserir credenciais certos.

A tela que segue a tela de login serve simplesmente para seleccionar o dispositivo Bluetooth que comunica com o arduino, conforme exibido abaixo.



Figura 27: Tela para seleccionar o dispositivo Bluetooth

Fonte: Autor (2020)

Quando o usuário clica em HC-05 (que é o módulo Bluetooth utilizado para a comunicação do arduino e a aplicação android para o accionamento dos dispositivo) a aplicação automaticamente abre a tela principal ou tela de gerenciamento. A partir desta tela o usuário da aplicação pode:

- Escolher accionar os dispositivos apenas com comando de voz, por exemplo (ligar luz do quarto);
- Pode gerenciar o subsistema de controle de iluminação, isto é, pode agendar ligar luzes numa determinada hora, ou poder ligar ou desligar uma determinada luz.
- Pode gerenciar o sistema de climatização, que consiste simplesmente em ligar ou desligar o sistema de climatização, ou um determinado dispositivo climatização.
- Pode ligar ou desligar o sistema de combate a incêndio. Que consiste num sensor de fumaça ligado que quando detecta uma determinada fumaça o arduino de forma automática accionará uma sirene para chamada de atenção e envia uma mensagem de alerta ao número de contacto do usuário do sistema, previamente cadastrado.
- Pode ligar ou desligar o sistema de Intrusão ou por outra pode armar ou desarmar o sistema de intrusão, que consiste em sensores de detenção de presença instalados em lugares estratégicos que quando detectam um movimento de forma automática o microcontrolador Arduino toca sirene para a chamada de atenção e sinaliza pelo GSM o numero de contacto do usuário.

A figura abaixo mostra o aspecto da tela principal ou a tela de gerenciamento.

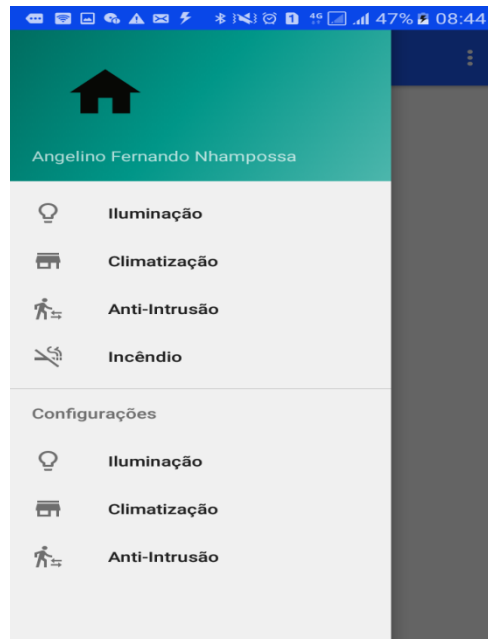


Figura 28: Tela de gerenciamento

Fonte: Autor (2020)

Para a figura 29, se o utilizador escolher a opção de accionamento por comando de voz, o usuário terá de informar palavras ou frases previamente cadastrados e a aplicação processará as palavras, depois de descodificá-las enviá-los-á em forma de bytes ao arduino e este será responsável de executar a acção pela qual foram dirigidas.

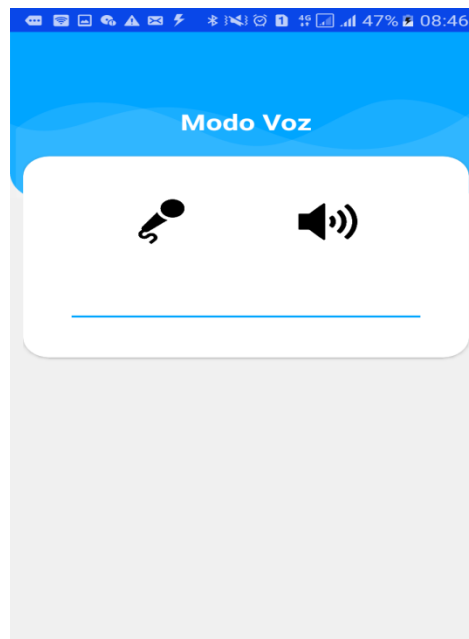


Figura 29: Tela de Monitoramento por Voz

Fonte: Autor (2020)

Se, por exemplo, o usuário escolher a opção de Sistema de Iluminação, lhe será aberto o formulário de eliminação para escolher a luz, depois de efectuar um clique seguirá a tela de acções (Ligar, Desligar ou Agendar), conforme ilustrado abaixo.

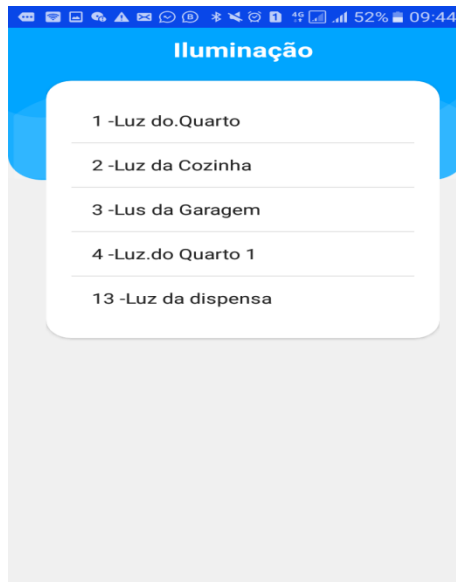


Figura 30: Tela de dispositivos de iluminação

Fonte: Autor (2020)

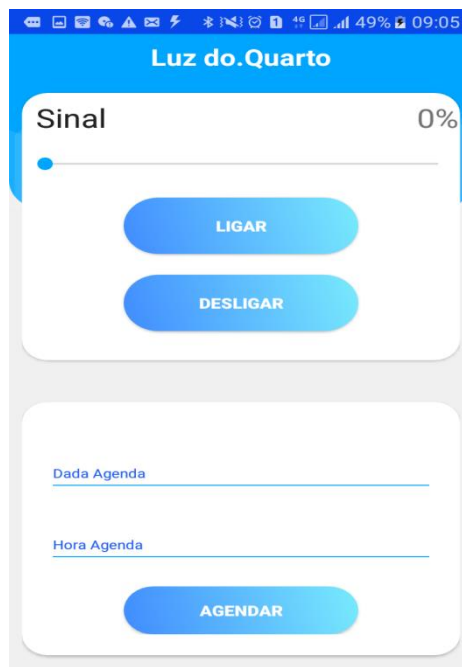


Figura 31: Tela de gerenciamento iluminação

Fonte: Autor (2020)

Com vista a garantir a escalabilidade, a aplicação permite o cadastro dos eventos da aplicação em uma Base de dados. A tela abaixo mostra especificamente o cadastro de luzes, onde, a “descrição” indica o nome da luz, “ON” é a acção que será enviada ao arduino para ligar enquanto que “OFF” representa a acção desligar.



Figura 32: Tela de cadastro de luzes

Fonte: Autor (2020)

2.9. FIRMWARE DO SISTEMA

O conceito do firmware consiste em um conjunto de sub-rotinas para executar as tarefas a nível do Arduino ou hardware. Para cada tarefa há uma sub-rotina responsável pela sua execução. No Arduino após a função denominada setup(), que inicializa e declara os valores iniciais, a função loop() repete-se continuamente permitindo que o programa funcione dinamicamente ficando à escuta de sinais gerados pelos dispositivos ou instruções enviadas pela aplicação.

Os módulos de testes e sub-rotinas que compõem o firmware estão representados em **APÊNDICE VI**.

2.10. CONCLUSÃO DO CAPÍTULO

De forma sintética, o presente capítulo mostra o processo que marca o desenvolvimento do “*Sistema de segurança e accionamento residenciais usando GSM e Bluetooth*”, neste caso, para permitir a comunicação do Hardware e Software, que são as principais partes constituintes do presente sistema, foi desenvolvido um firmware na linguagem de

programação C e embutido no microcontrolador (placa arduino). A aplicação Android desenvolvida é útil para facilitar o utilizador na configuração e accionamento do sistema mas não dispensa o accionamento manual dos dispositivos.

Requisitos alcançados e não alcançados no desenvolvimento do Sistema	
Requisito Funcional	Estado
[RF01] Efectuar Login	A
[RF02] Permitir accionar os sistemas domésticos;	A
Permitir agendar tarefas para accionar os sistemas domésticos;	A
Permitir visualizar o estado actual dos sistemas domésticos;	ED
Enviar notificações ao utilizador sobre os eventos do sistema GSM.	ED
Permitir detectar a intrusão, activar a sirene e simultaneamente notificar o utilizador através de GSM.	ED
Permitir o recarregamento automático da fonte;	A
Requisito não Funcional	Estado
Conceber o sistema de tal modo que seja escalável	A
Permitir que o sistema seja flexível, isto é rápido em dar resposta quando ocorrer um accionamento	NA
Permitir que a data do microcontrolador esteja sempre certa, para que o usuário possa fazer agendamentos de eventos.	A

Tabela 14: Requisitos alcançados e não alcançados do Sistema

A – Alcançado

ED – Em Desenvolvimento

NA – Não Alcançado (próxima versão do desenvolvimento)

CAPITULO III – CONCLUSÕES E RECOMENDAÇÕES

3.1 Conclusão

A automação residencial é uma derivação da automação predial com foco em atender necessidades domésticas, com projectos de controle em equipamentos, como portas, janelas, televisores, sistemas de climatização e irrigação e outros. Grande parte dos comandos são realizados por meio de relés de contacto que são activados através de uma central que recebe o comando do usuário e repassa para seus periféricos actuarem. A central é programada para realizar acções pré-definidas ou mesmo através de comando externos realizados pelo usuário ou mesmo por sensores no meio.

Este trabalho apresentou o projecto e a execução de um produto voltado para automação residencial, aplicável em um ambiente real. Foram utilizadas padrões abertos de hardware e software facilitando a escalabilidade e integração com outros sistemas e produtos de automação residencial.

A interface Arduino se mostrou muito simples em sua aplicação na automação residencial, e por ser modular, faz com que o sistema possa ser adaptado facilmente para a necessidade do usuário.

Através das informações obtidas das empresas que fornecem o serviço de automação residencial, foi possível comparar preços disponíveis no mercado com o baixo custo do presente projecto, podendo concluir que, mesmo comparado com outros tipos de automação residencial já se mostra com o valor muito menor, viabilizando assim a sua aplicação.

Por se tratar de um sistema totalmente modular e escalável, será possível adicionar ao sistema outros componentes, sem necessariamente a alteração de todo projecto.

Desafios foram obtidos no decorrer deste trabalho, juntamente com eles soluções e ideias foram buscadas para que um trabalho bem feito e coerente fosse feito.

Analisando o resultado final, percebe-se que muitas melhorias e implementações poderão ser feitas. Quando se fala em Arduino uma gama enorme aparece e deve-se analisar todos os casos e possibilidades possíveis a serem empregadas.

3.2. Recomendações

O presente projecto foi influenciado negativamente por factores externos no desenvolvimento do hardware, pelo facto de existirem poucos fornecedores de placas de circuito impresso e componentes electrónicos. Alguns componentes utilizados no sistema não são comercializados no mercado Moçambicano, necessitando que os mesmos fossem importados. Isto teve impacto significativo no custo e no prazo de desenvolvimento do projecto.

O Arduino UNO apresenta pouca memória de armazenamento interno, portanto, nem toda informação sobre os eventos (data, hora, e dispositivo que gera um alarme) é registada. Durante a simulação verificou-se que o Arduino congela quando a memória RAM se esgota. Para projectos maiores, que incluem mais dispositivos e equipamentos será necessário identificar uma placa Arduino com mais capacidade de armazenamento.

A aplicação Android é customizável de tal forma que permite adicionar mais dispositivos sem necessidade de programa-lo. As principais funcionalidades da aplicação foram implementadas.

Outro ponto importante a considerar é o custo de implantação do projecto, dado que a maioria das residências em Moçambique não são projectadas para acomodar sistemas de automação residencial, tornando complicado a integração deste sistema com sistemas domésticos presentes na residência. No entanto, a utilização de tecnologias de comunicação sem fio pode reduzir o custo de implantação do sistema, dado que, reduz a necessidade de instalação de cabos eléctricos pelas diferentes partes da residência.

Finalizando, os dispositivos comportaram-se bem nos testes realizados. Algumas melhorias podem ser implementadas na construção dos circuitos de controlo e monitorização, porém as funcionalidades principais do sistema estão bem atendidas.

No futuro, pode ser possível a modificação do projecto para se tornar um sistema com inteligência artificial (IA) aplicada na automação, utilizando, por exemplo, sensores de captura que visão para atender as necessidades do usuário e manipular relés e sensores para modificar o ambiente.

REFERENCIA BIBLIOGRÁFICAS

ARDUINOCC de. "Arduino Uno Rev3"
<<https://www.arduino.cc/en/Main/ArduinoBoardUno>>. Acessado em 30 de Dezembro de 2019;

DA SILVA, Cleber Benedito Savi . NARDOTO, Daniel Ribeiro. JORGE, Marcello de Melo. *Desenvolvimento de um sistema modular microcontrolado de automação residencial via wi-fi*, 2016. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/11993/1/CT_COELE_2016_2_17.pdf>. Acessado em: 20 de Fevereiro 2020.

BIONDO, Rodrigo Mingolelli. *Sistemas e Aplicabilidade*, 2011. Disponível em: <https://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-19102011-122815/publico/Biondo_Rodrigo_Mingolelli.pdf>. Acessado em: _____.

QUINDERÉ, Patrick Romero Frota. *Casa Inteligente – Um Protótipo de Sistema de Automação Residencial de Baixo Custo*; 2009. Disponível em: <<https://cepein.femanet.com.br/BDigital/arqTccs/1211320586.pdf>>. Acessado em: 20 de Janeiro de 2020.

GOULART, Ederson Lopes. *Automação de Interfone com Arduino*; 2017. Disponível em: < <https://repositorio.uniceub.br/jspui/bitstream/235/5461/1/>> Acessado em: 29 de Dezembro de 2019.

ABOWD, Gregory D.; STERBENZ, James PG. Final report on the inter-agency workshop on research issues for smart environments. IEEE Personal Communications, v. 7, n. 5, p. 36-40, 2000.

BOLZANI, Caio AM. *Residências Inteligentes: Domótica; Redes Domésticas; Automação Residencial*. São Paulo: Editora Livraria da Física, 2004.

SILVA, Domiciano Correa Marques "*Microfone*"; Brasil Escola. Disponível em <<http://brasilecola.uol.com.br/fisica/microfone.htm>>. Acesso em 30 de Dezembro de 2019.

BRUCHER, G. The electric home of the future. Popular Mechanics, v. 72, n.2, 1939.

- STIMSON, J. T. A house of magic. Popular Mechanics, 1954.
- MELOSI, M. Thomas A. Edison end the Modernization of America. [S.I]: Longman, 1990.
- GERHART, James. 1999. *Home Automation & Wiring. s.l. : McGraw-Hill Education - Europe*, 1999
- MILLER, Michael. 2015. The Internet of Things. US : Que Publishing, 2015.
- TYSON, J. *How Serial Ports Work*. 2009a Disponível em:<<http://computer.howstuffworks.com/serial-port3.htm>> Acessado em: 08 de Maio de 2020.
- TYSON, J. *How USB Ports Work*. 2009b Disponível em:<<http://computer.howstuffworks.com/usb.htm>> Acessado em: 18 de Abril de 2020
- RIBEIRO, Eduardo de Medeiros. *Controle de portão eletrônico através de módulo gsm/gprs shield com arduino*; Lages,2017.
- BORGES, Bruno Rodrigues. *Desenvolvimento de Aplicação Mobile Utilizando Metodologia Ágil SCRUM*; 2017. Disponível em:<<https://repositorio.ufu.br/bitstream/123456789/20098/1/DesenvolvimentoAplicacaoMobile.pdf>> Acessado em: 20 de Maio de 2020.
- GOMES, Hugo Gabriel Lins. *Automação Residencial usando tecnologia IEEE 802.15.4 Zigbee*. 2011. Disponível <<https://repositorio.uniceub.br/jspui/bitstream/235/3604/3/Monografia%20Hugo%20Gabriel%20Lins%20Gomes.pdf>>. Acessado em: 25 de Abril de 2020.
- Guedes, Gilleanes T. A. UML 2 : uma abordagem prática / Gilleanes T. A. Guedes. --2. ed. -- São Paulo : Novatec Editora, 2011.
- Centro Universitário de Brasília. UniCEUB, Maia. Gustavo Moura Fé. Disponível em:<<http://repositorio.uniceub.br/bitstream/123456789/3104/3/20855679.pdf>>.
- SILVA, Domiciano Correa Marques da. "Microfone"; Brasil Escola. Disponível em <<http://brasilecola.uol.com.br/fisica/microfone.htm>>.

ALVAREZ, Daniel Fernandes De Souza; ANTUNES, Felipe Ihlenfeldt. *Automação Residencial Utilizando Bluetooth, Ethernet E Smartphone*. 2015. Disponível <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/3977/1/CT_COMET_2014_2_05.pdf>. Acessado em: 12 de Abril de 2020.

TÓFOLI, Ricardo José. *Casa Inteligente – Sistema De Automação Residencial*. 2014. Disponível < <https://cepein.femanet.com.br/BDigital/arqTccs/1211320586.pdf> >. Acessado em: 06 de Junho de 2020.

FONSECA, E.G.P. e BEPPU, M.M., Apostila Arduino, Universidade Federal Fluminense, Rio de Janeiro, 2010.

SAMUEL, Agnaldo Hilário. *Desenvolvimento De Um Sistema De Automação Residencial Usando Uma Aplicação Android*. 2016.

SOMMERVILLE, I. Engenharia de Software. Pearson Prentice Hall, 9ª edição. São Paulo, 2011.

NACCA, R. M.. Maquetes e Miniaturas. São Paulo: Giz Editorial, 2006.

MÖSCH, M. E. O processo projetivo na arquitetura: o ensino do projeto de escolas. 2009 251f. Tese (Doutorado) - Faculdade De Engenharia Civil, Arquitetura e Urbanismo, UNICAMP, Campinas, 2009.

Writers, Community. Use Relays to Control High-Voltage Circuits with an Arduino. Disponível em: <http://www.allaboutcircuits.com/projects/userelays-to-control-high-voltage-circuitswith-an-arduino/>. Acessado em: 01/06/2020

JORGE, Marcello De Melo. *Desenvolvimento de um sistema modular microcontrolado de automação residencial via wi-fi*. 2016. Disponível <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/11993/1/CT_COELE_2016_2_17.pdf> Acessado em: 16 de Maio de 2020.

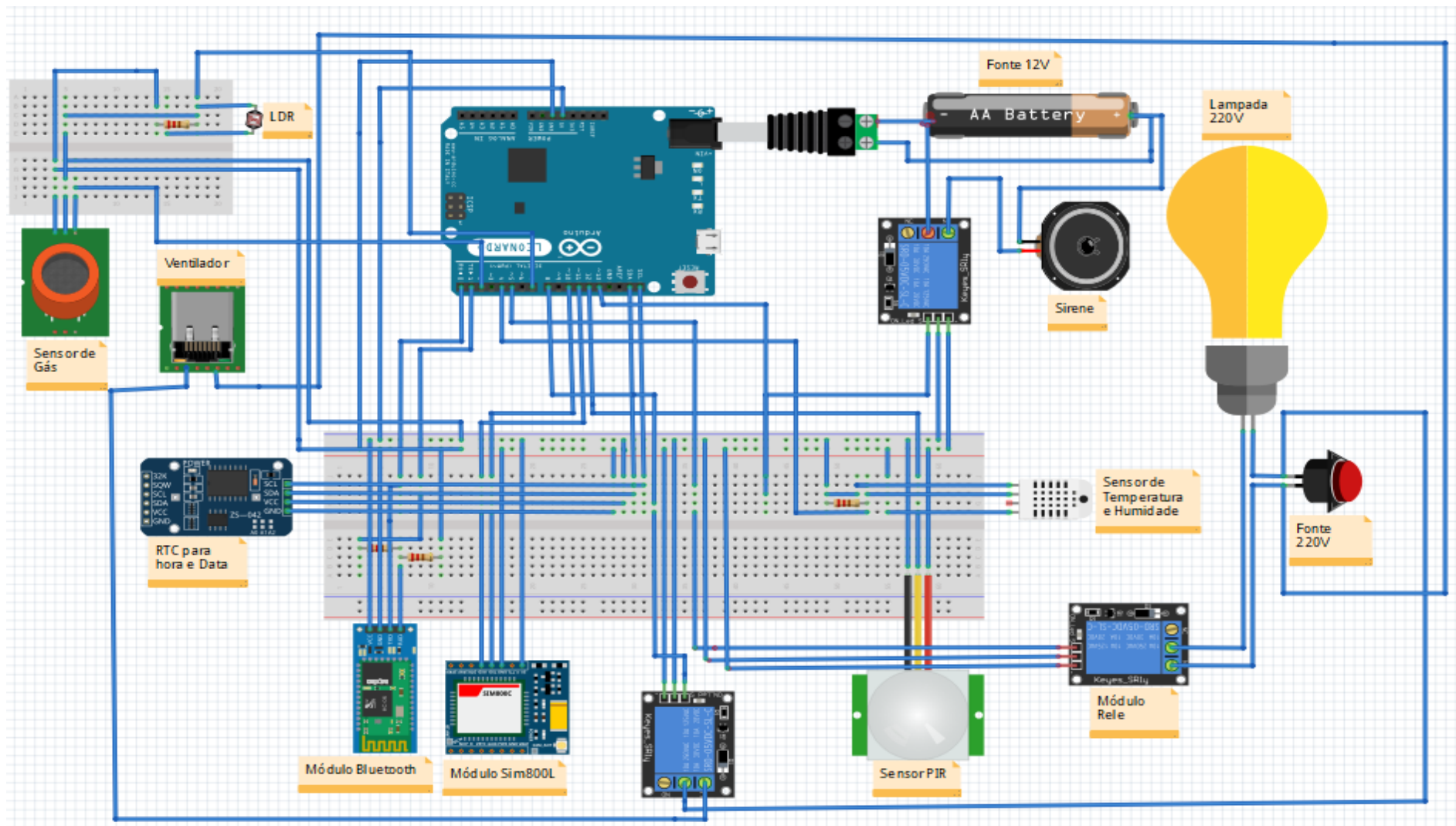
GUILHERME, Vianna Santos. 2013. Como accionar um relé com arduino ou pic projecto e funcionamento. Disponível em <<http://eletronicaemcasa.blogspot.com.br/2013/12/como-acionar-um-rele-comarduino-ou-pic.html>>. Acessado em: 01 de Junho de 2020.

APÊNDICE

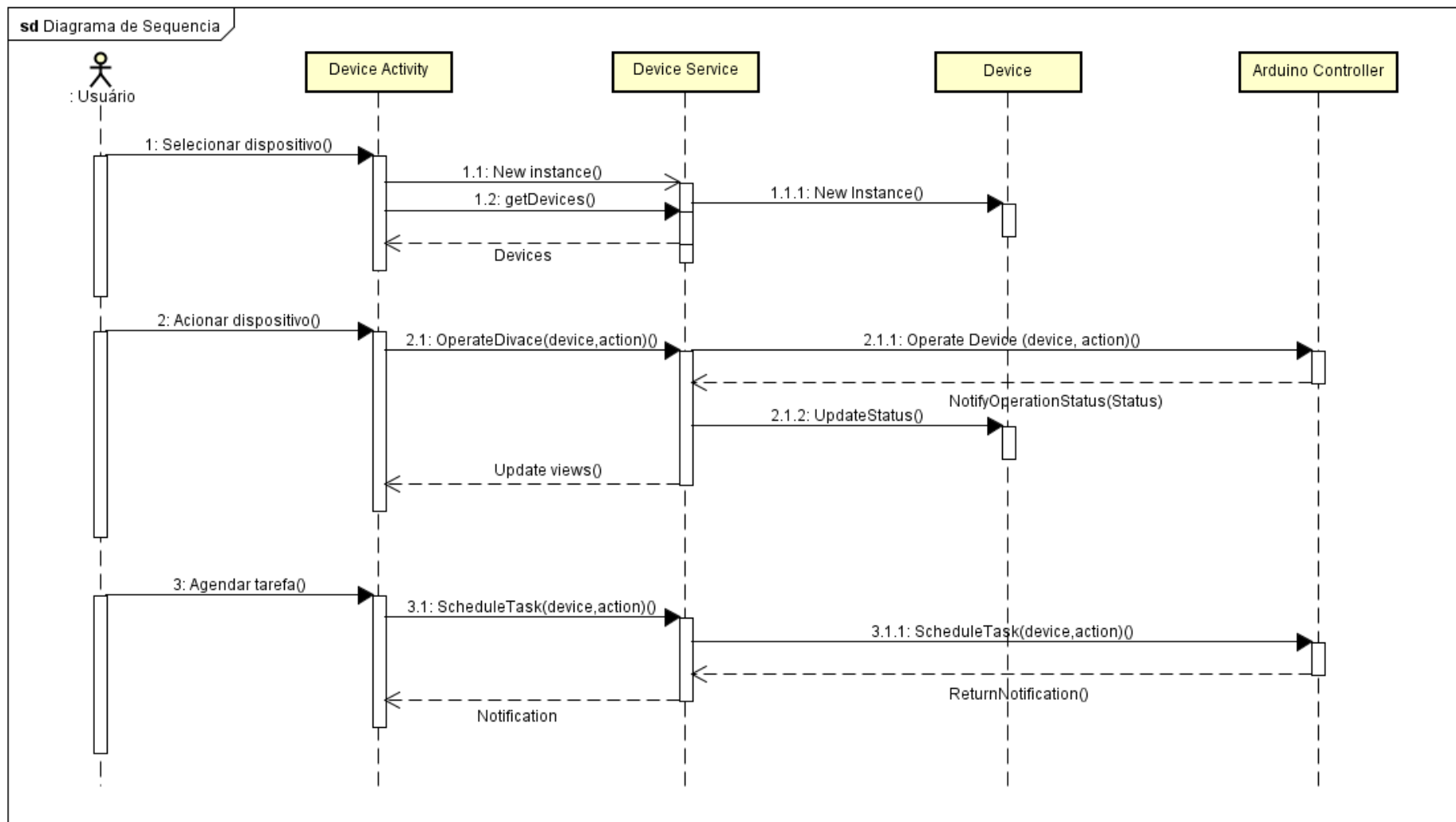
APÊNDICE I: CUSTOS FINANCEIROS

Item	Material	Quantidade	Preço Unitário	Preço Total (MZN)
1	Microcontrolador	1	1000,00	1.000,00
2	Módulo de reles	4	125,00	500,00
3	Módulo Bluetooth	1	200,00	200,00
4	Módulo GSM Sim800L	1	650,00	650,00
5	Protobord	1	100,00	100,00
6	Jumper	30	5,00	150,00
7	Fonte 12V	1	500,00	500,00
8	Sensor de Movimento	1	200,00	200,00
9	Sensor de Fumaça	1	200,00	200,00
10	Sensor de Temperatura	1	200,00	200,00
11	RTC	1	350,00	350,00
12	Complementares		250,00	250,00
Total				4.500,00

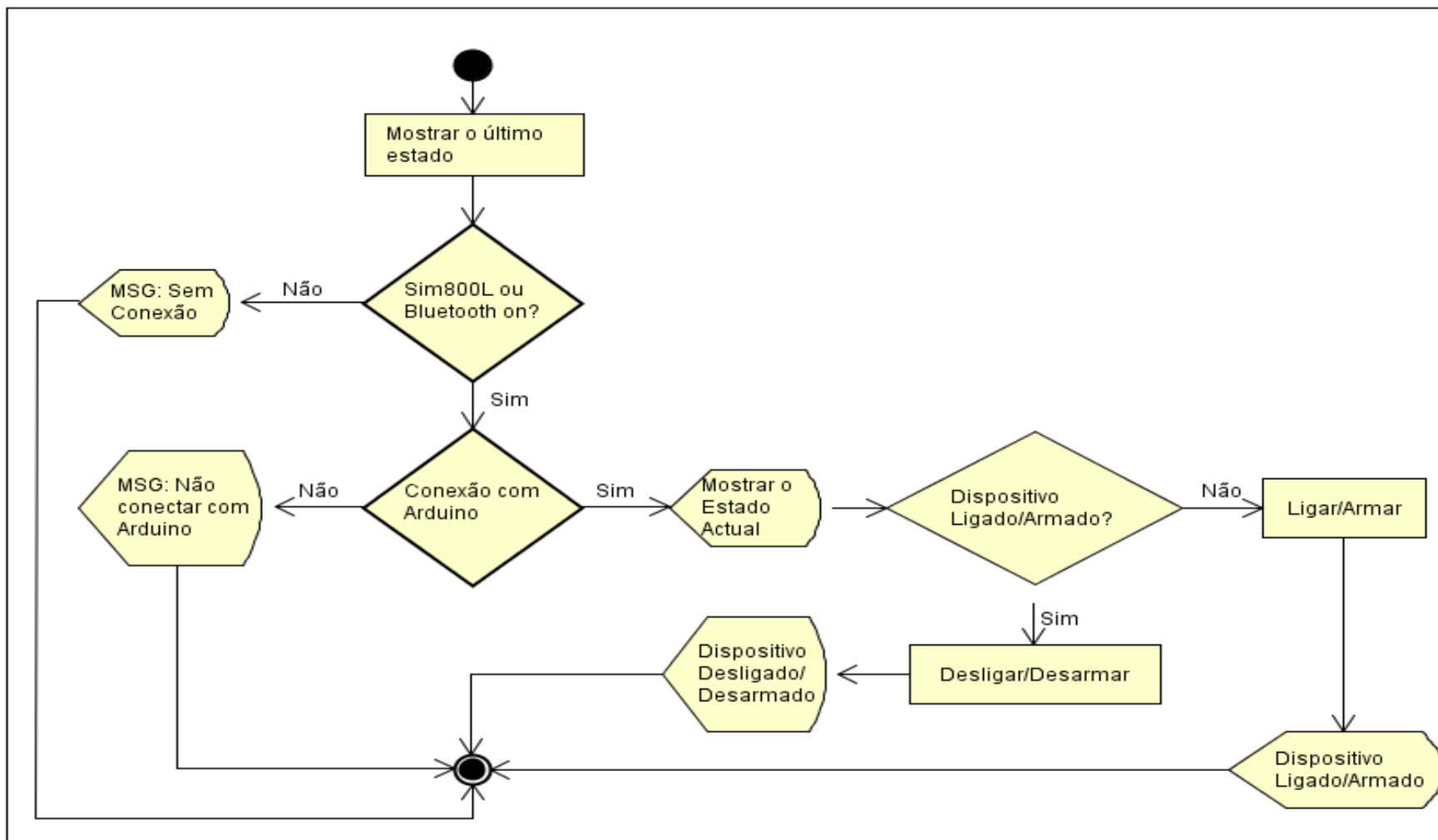
APÊNDICE II: ESQUEMA GERAL DE INTERLIGAÇÃO DOS DISPOSITIVOS



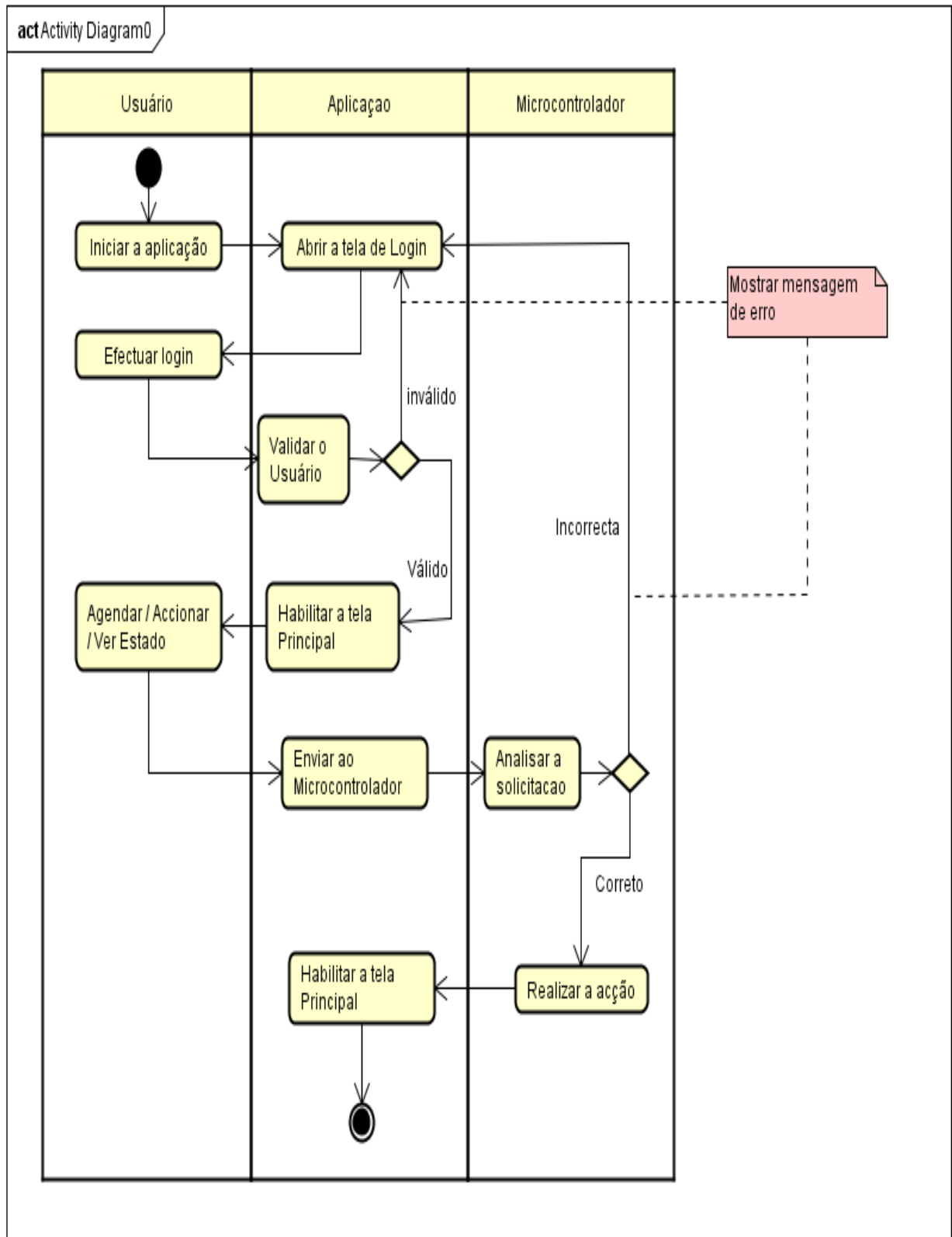
APÊNDICE III: DIAGRAMA DE SEQUÊNCIA



APÊNDICE IV: FLUXOGRAMA DE ACCIONAMENTO DE DISPOSITIVOS OU EQUIPAMENTOS



APÊNDICE V: DIAGRAMA DE ACTIVIDADES



APÊNDICE VI

CÓDIGO DE TESTE DE SENSOR DE DETEÇÃO DE PRESENÇA



```
int pir1 = 6;
int pir2 = 7;

void setup() {
  Serial.begin(9600);

  pinMode(pir1, INPUT);
  pinMode(pir2, INPUT);
}

void loop() {

  if (movimentol == HIGH) {
    Serial.println("Detectou.");
  } else if(movimentol == LOW) {
    Serial.println("Sem Movimento.");
  }
}

boolean movimentol() {
  return digitalRead(pir1);
}
boolean movimento2() {
  return digitalRead(pir2);
}
```

Guardado com Sucesso.

CÓDIGO DE TESTE DO MÓDULO RTC PARA AGENDAMENTO

```
RTC_teste $

#include <Wire.h>
#include "RTClib.h"

RTC_DS3231 rtc;
char daysOfTheWeek[7][12] = {"Domingo", "Segunda", "Terça", "Quarta", "Quinta", "Sexta", "Sábado"};

void setup() {
  Serial.begin(9600);
  if(! rtc.begin()) {
    Serial.println("DS3231 não encontrado");
    while(1);
  }
  if(rtc.lostPower()){
    Serial.println("DS3231 OK!");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }
  delay(100); //INTERVALO DE 100 MILISSEGUNDOS
}

void sensor(){
}

void loop () {
  DateTime now = rtc.now(); //CHAMADA DE FUNÇÃO
  Serial.print("Data: "); //IMPRIME O TEXTO NO MONITOR SERIAL
  Serial.print(now.day(), DEC); //IMPRIME NO MONITOR SERIAL O DIA
  Serial.print('/'); //IMPRIME O CARACTERE NO MONITOR SERIAL
  Serial.print(now.month(), DEC); //IMPRIME NO MONITOR SERIAL O MÊS
  Serial.print('/'); //IMPRIME O CARACTERE NO MONITOR SERIAL
  Serial.print(now.year(), DEC); //IMPRIME NO MONITOR SERIAL O ANO
  Serial.print(" / Dia: "); //IMPRIME O TEXTO NA SERIAL
  Serial.print(daysOfTheWeek[now.dayOfTheWeek()]); //IMPRIME NO MONITOR SERIAL O DIA
  Serial.print(" / Horas: "); //IMPRIME O TEXTO NA SERIAL
  Serial.print(now.hour(), DEC); //IMPRIME NO MONITOR SERIAL A HORA
  Serial.print(':'); //IMPRIME O CARACTERE NO MONITOR SERIAL
  Serial.print(now.minute(), DEC); //IMPRIME NO MONITOR SERIAL OS MINUTOS
  Serial.print(':'); //IMPRIME O CARACTERE NO MONITOR SERIAL
  Serial.print(now.second(), DEC); //IMPRIME NO MONITOR SERIAL OS SEGUNDOS
  Serial.println(); //QUEBRA DE LINHA NA SERIAL
  delay(1000); //INTERVALO DE 1 SEGUNDO
}
```

CÓDIGO DE TESTE DO SENSOR DE TEMPERATURA HUMIDADE

```

✓ → 📄 ⬆️ ⬇️ Verificar 🔍
temperatura §
#include "DHT.h"

#define DHTPIN 2
#define DHTTYPE DHT22

void setup() {
  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));

  dht.begin();
}

void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);

  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
  float hif = dht.computeHeatIndex(f, h);
  float hic = dht.computeHeatIndex(t, h, false);

  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.print(f);
  Serial.print(F("°F  Heat index: "));
  Serial.print(hic);
  Serial.print(F("°C "));
  Serial.print(hif);
  Serial.println(F("°F"));
}

```

CÓDIGO DE TESTE DO SENSOR DE GÁS



```
teste_fumac

#define MQ_analog A2
#define MQ_dig 7
int valor_analog;
int valor_dig;
void setup() {
  Serial.begin(9600);
  pinMode(MQ_analog, INPUT);
  pinMode(MQ_dig, INPUT);
}
void loop() {
  valor_analog = analogRead(MQ_analog);
  valor_dig = digitalRead(MQ_dig);
  Serial.print(valor_analog);
  Serial.print(" || ");
  if(valor_dig == 0)
    Serial.println("GAS DETECTADO !!!");
  else
    Serial.println("GAS AUSENTE !!!");
  delay(500);
}

Guardado com Sucesso.
```

CÓDIGO DE TESTE DO MÓDULO BLUETOOTH



```
Teste_hm10 $

byte LED = 2;
char BT_input=' ';

void setup()
{
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  while (!Serial)
  {
  }
}

void loop()
{
  if (Serial.available())
  {
    BT_input = Serial.read();
    if ( BT_input== 48)
    {
      digitalWrite(LED, LOW);
      Serial.println(BT_input);
      Serial.println("LED is OFF");
    }
    if ( BT_input== 49)
    {
      digitalWrite(LED, HIGH);
      Serial.println(BT_input);
      Serial.println("LED is ON");
    }
  }
}

Guardado com Sucesso.
```

CÓDIGO DE RECARREGAMENTO AUTOMÁTICO DA FONTE DE ALIMENTAÇÃO

```
TesteFonte
1 #define R1 660.0
2 #define R2 330.0
3
4 void setup() {
5     pinMode(2, OUTPUT);
6     pinMode(3, OUTPUT);
7 }
8
9 void loop() {
10    float Vout = analogRead(A0) * (5.0/1023.0);
11    float Vin = (R1 + R2)/(R2) * Vout;
12    if((Vin >= 7)&&(Vin <= 13))
13        //Funcionamento Normal
14        digitalWrite(2, HIGH);
15    else if(Vin <7)
16        //liga o interruptor para carregar a bateria
17        digitalWrite(3, HIGH);
18    else if(Vin > 14)
19        //desliga o interruptor para não carregar a bateria
20        digitalWrite(3, LOW);
21
22 }
```

AVALIAÇÃO FINAL - PARECER DO ORIENTADOR

Eng° Christian Coulon, docente na Faculdade de Ciências e Tecnologias da Universidade Zambeze, vem na qualidade de orientador da Monografia de Licenciatura submetida por Angelino Fernando Nhampossa, com o tema “**SISTEMA DE SEGURANÇA E ACIONAMENTO RESIDENCIAL USANDO GSM E BLE**”, declarar que o trabalho apresentado pelo aspirante reúne qualidade científica e condições necessárias para que o estudante seja submetido à prova de defesa para obter o grau de Licenciatura em Engenharia Informática.

Beira, aos ____ de _____ 2020

(Eng° Christian Coulon)