

Tiago Hártur Oliveira Feu

**DESENVOLVIMENTO DE UM SIMULADOR DE
INCÊNDIOS PARA O CORPO DE BOMBEIROS
MILITAR DO ESPÍRITO SANTO**

Parte manuscrita do Projeto de Graduação do aluno Tiago Hártur Oliveira Feu, apresentada ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de engenheiro eletricista.

Vitória – ES

2022

Tiago Hártur Oliveira Feu

DESENVOLVIMENTO DE UM SIMULADOR DE INCÊNDIOS PARA O CORPO DE BOMBEIROS MILITAR DO ESPÍRITO SANTO

Parte manuscrita do Projeto de Graduação do aluno Tiago Hártur Oliveira Feu, apresentada ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de engenheiro eletricista.

COMISSÃO EXAMINADORA:

Andre Ferreira

Universidade Federal do Espírito Santo
Orientador

Kyra White

Universidade Federal do Espírito Santo
Examinador

Devin Hill

Universidade Federal do Espírito Santo
Examinador

Vitória – ES

2022

RESUMO

XXX

ABSTRACT

xxx

LISTA DE FIGURAS

Figura 1 – Visão geral do protótipo	11
Figura 2 – Tetraedro do fogo	15
Figura 3 – Classes de incêndio	16
Figura 4 – Logo do Pygame	17
Figura 5 – Raspberry Pi 4, Modelo B.	18
Figura 6 – Espectro eletromagnético	19
Figura 7 – Modulador de produto	20
Figura 8 – Modulação OOK	21
Figura 9 – Sinal OOK recebido	22
Figura 10 – Regeneração do sinal recebido	22
Figura 11 – Diagrama geral	25
Figura 12 – Attiny85	26
Figura 13 – Módulos RF	27
Figura 14 – Antena espiral 433 MHz	27
Figura 15 – Bateria de 9 V	28
Figura 16 – Regulador de tensão L7805CV	29
Figura 17 – Circuito do botão	30
Figura 18 – Conectores	30
Figura 19 – Simulação em 3D da placa de circuito impresso	31
Figura 20 – Placa de fenolite corroída em percloroeto de ferro	32
Figura 21 – PCB com fabricação especializada	33
Figura 22 – Peças para montagem da caixa	34
Figura 23 – Extintor com circuito	35
Figura 24 – Fluxograma simplificado do <i>firmware</i>	36
Figura 25 – Protocolo de comunicação	38
Figura 26 – Tela inicial do jogo	39
Figura 27 – Fases do jogo	40
Figura 28 – Sprite de fumaça	40

LISTA DE TABELAS

Tabela 1 – Principais Rotinas	37
Tabela 2 – Custos do projeto	42

LISTA DE ABREVIATURAS E SIGLAS

ASK	<i>Amplitude Shift keying</i>
BASK	<i>Binary Amplitude Shift Keying</i>
CBMES	Corpo de Bombeiros Militar do Estado do Espírito Santo
DIP	<i>Dual In-line Package</i>
FSK	<i>Frequency Shift Keying</i>
GPIO	<i>General Purpose Input/Output</i>
HDMI	<i>High-definition Multimedia Interface</i>
JST	<i>Japan Solderless Terminal</i>
LAN	<i>Local Area Network</i>
LED	<i>Light-emitting Diode</i>
LGPL	<i>Lesser General Public License</i>
NFPA	<i>National Fire Protection Association</i>
NR	Norma Regulamentadora
OOK	<i>On-off Keying</i>
PLA	Ácido Poliláctico
PSK	<i>Phase Shift Keying</i>
RAM	<i>Random Access Memory</i>
RF	Radiofrequênciā
SD	<i>Secure Digital</i>
UFES	Universidade Federal do Espírito Santo
USB	<i>Universal Serial Bus</i>
Wi-fi	<i>Wireless Fidelity</i>

LISTA DE SÍMBOLOS

$p(t)$	Sinal de uma onda portadora no domínio do tempo
$x(t)$	Sinal modulador no domínio do tempo
E_0	Amplitude da portadora
f_0	Frequência da portadora
θ	Angulo de fase da portadora
$m(t)$	Sinal modulador multinível
N	Número de bits combinados
M	Nível de tensão
k_M	Constante de modulação adimensional
$e(t)$	Sinal modulado M-ASK no domínio do tempo
t	Instante de tempo
A	Amplitude de um sinal binário unipolar
$r(t)$	Sinal recebido no domínio do tempo
$y(t)$	Envoltória no domínio do tempo

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Apresentação	11
1.2	Justificativa	12
1.3	Objetivos	12
1.3.1	Objetivo Geral	12
1.3.2	Objetivos Específicos	12
2	EMBASAMENTO TEÓRICO	14
2.1	Combate a incêndio	14
2.1.1	Teoria do fogo	14
2.1.2	Classes de incêndio	15
2.1.3	Extintores	16
2.2	Motor de jogos	17
2.3	Plataforma	17
2.4	Comunicação via radiofrequência	18
2.4.1	Modulação ASK	20
2.4.2	Demodulação ASK(OOK)	21
2.5	Avaliação	22
3	METODOLOGIA E ETAPAS DE DESENVOLVIMENTO	24
4	IMPLEMENTAÇÃO	25
4.1	Hardware	25
4.1.1	Microcontrolador	26
4.1.2	Módulo transmissor RF e módulo receptor RF	27
4.1.3	Bateria + Regulador	28
4.1.4	LED indicador, botão e conectores	29
4.1.5	Placa de circuito impresso	31
4.1.6	Caixa para acomodar os componentes	33
4.1.7	Montagem do controle	34
4.2	Firmware	35
4.2.1	Protocolo de comunicação	38
4.3	Software	38
4.3.1	Criação e utilização dos menus	38
4.3.2	<i>Sprites</i> e animações	40
5	RESULTADOS E TRABALHOS FUTUROS	42

5.1	Custos	42
	REFERÊNCIAS	43
	APÊNDICES	44
	APÊNDICE A – CÓDIGO DO CONSOLE	45

1 INTRODUÇÃO

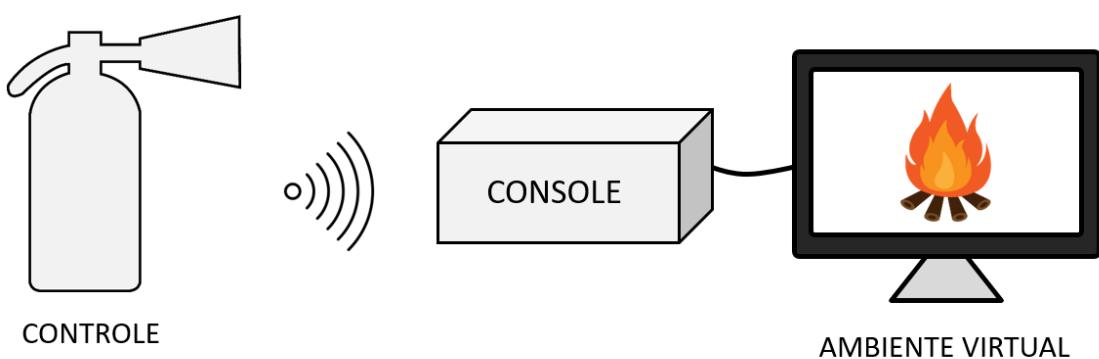
1.1 Apresentação

A motivação para o desenvolvimento desse trabalho surgiu a partir do pedido de um agente do Corpo de Bombeiros Militar do Estado do Espírito Santo - CBMES. O objetivo é criar uma ferramenta que auxilie o CBMES enquanto fornecem cursos e treinamentos para o público externo à corporação, em especial o público infantil.

É parte do cotidiano do Corpo de Bombeiros realizar demonstrações e treinamentos a respeito de como se comportar em situações de risco, e da utilização correta de ferramentas como extintores de incêndio. Existe, contudo, limitações impostas às demonstrações realizadas em ambientes fechados como auditórios e salas de aula, tornando inviável demonstrar a liberação do agente extintor em locais como esses.

Na tentativa de contornar essas limitações, a proposta é desenvolver um simulador no qual crianças possam utilizar extintores de incêndio reais que sirvam de controles (*joysticks*) para interagir com as situações de princípio de incêndio simuladas em um ambiente virtual, como mostrado na figura 1.

Figura 1 – Visão geral do protótipo



Fonte: Produção do próprio autor.

Como proposta deste projeto de graduação, deseja-se construir um kit educacional que simule situações de princípios de incêndios. Esse kit será composto dos controles remotos, que serão uma versão adaptada de extintores antigos utilizados pelo corpo de bombeiros e do console, responsável pelo ambiente virtual com os diferentes tipos de incêndio.

1.2 Justificativa

Em muitas escolas o ensino baseado em atividades práticas como estudo de caso ou aprendizagem orientada a problemas vem se tornando cada vez mais comuns, substituindo modelos mais tradicionais de ensino (LEE, 2012). Essas atividades práticas, também chamadas de atividades *hands-on*, são um fator associado ao aumento no interesse dos alunos sobre o tema exposto (HOLSTERMANN; GRUBE; BÖGEHOLZ, 2010).

Ler e escrever, por muito tempo, têm sido a principal maneira de alunos aprenderem. Entretanto, estudos mostram que estudantes aprendem e acumulam informações melhor quando são apresentados a múltiplos estímulos. Acredita-se que ao utilizar-se de múltiplos sistemas sensoriais em atividades de aprendizagem, o cérebro consegue armazenar as informações apresentadas de forma mais fácil e rápida. Nessa linha de pensamento, alunos de várias idades podem se beneficiar de atividades *hands-on*, como jogos educativos (COX, 2014).

O interesse em desenvolver um simulador virtual controlado por extintores reais que, anteriormente foram utilizados por bombeiros no seu dia-a-dia, surgiu dessa linha de raciocínio. A intenção é tornar o aluno um participante ativo da construção de conhecimento, em oposição à apenas receber as informações a eles apresentadas como ouvinte.

1.3 Objetivos

1.3.1 Objetivo Geral

Construir um protótipo de um simulador de incêndios que auxilie o Corpo de Bombeiros a desenvolver suas atividades de ensino à comunidade, em especial para o público infantil, tornando o processo de ensino mais eficiente.

1.3.2 Objetivos Específicos

- Desenvolver o ambiente virtual adequado às necessidades do Corpo de Bombeiros, simulando situações de princípio de incêndio.
- Desenvolver a interface física com a qual os usuários irão interagir para controlar o ambiente virtual.

- Desenvolver o protocolo de comunicação entre os controles e o console que executa o ambiente virtual.
- Validar a ferramenta de ensino através de uma demonstração prática junto ao CBMES com aplicação de questionários.

2 EMBASAMENTO TEÓRICO

2.1 Combate a incêndio

Um dos grandes marcos da história humana foi o domínio sobre o fogo. Esse domínio tornou possível coisas como o cozimento de alimentos, desenvolvimento de ferramentas e máquinas. Contudo, junto dessas possibilidades novas, surgiu também o risco de incêndios.

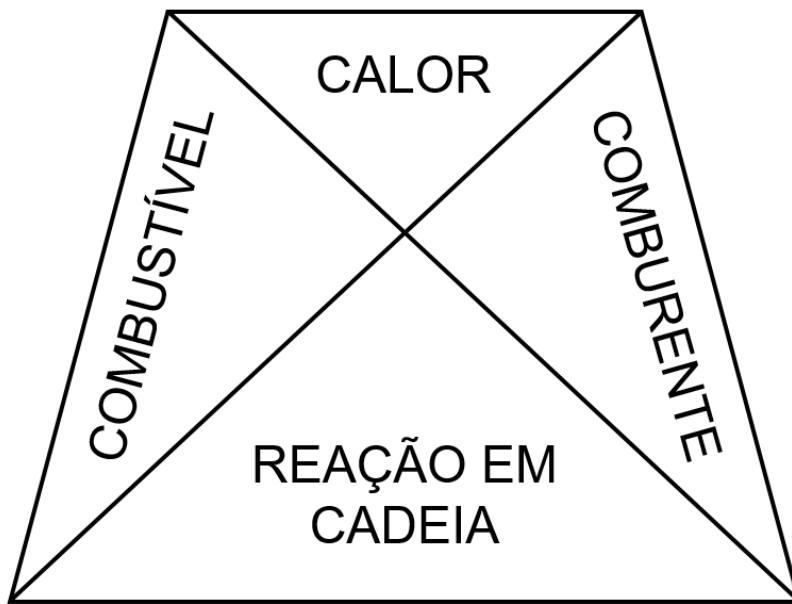
Por conta disso, de acordo com a Norma Regulamentadora 23 do Ministério do Trabalho e Previdência (NR 23), as organizações devem possuir em seus quadros pessoas capacitadas para utilizar equipamentos de proteção contra incêndios. O objetivo dessas pessoas é agir rapidamente no combate ao princípio de incêndio e proteger a integridade de pessoas e equipamentos.

2.1.1 Teoria do fogo

O fogo é um processo químico de transformação no qual um material combustível combina-se com o comburente (normalmente oxigênio atmosférico) e produz luz e calor. Esse calor proveniente da reação de combustão estimula novas reações, iniciando um processo de reação em cadeia (LIMA; ENCARNAÇÃO, 2016). As bases dessa reação química formam o tetraedro do fogo representado na figura 2, que é uma forma didática de representar os requisitos da combustão.

1. **Combustível:** É toda substância capaz de queimar e alimentar a combustão, ou seja, capaz de reagir com o oxigênio.
2. **Comburente:** É o elemento químico que se combina com o combustível na reação de combustão. O comburente mais comum é o oxigênio.
3. **Calor:** É a energia térmica transmitida de um sistema de maior temperatura para outro de menor temperatura. É o componente do tetraedro que dá inicio à reação de combustão.
4. **Reação em cadeia:** O calor irradiado atinge o combustível e o decompõe em partículas menores que se combinam com o oxigênio, irradiando mais calor e formando um ciclo constante.

Figura 2 – Tetraedro do fogo



Fonte: Produção do próprio autor.

2.1.2 Classes de incêndio

Na tentativa de agrupar incêndios pelas propriedades de seus materiais combustíveis, e com isto, tornar mais eficiente sua extinção, a *National Fire Protection Association - NFPA* elaborou uma classificação que divide os incêndios em 5 categorias principais (ROSA, 2015), conforme ilustrado na figura 3:

1. **Classe A:** Combustíveis sólidos e comuns, tal como madeira, papel, plástico, borracha, entre outros.
2. **Classe B:** Líquidos inflamáveis, líquidos combustíveis e gases inflamáveis, queimam em sua superfície e não deixam resíduos em sua queima.
3. **Classe C:** Equipamentos elétricos energizados, como motores, transformadores, quadros de distribuição, fios e etc.
4. **Classe D:** Metais combustíveis, como magnésio, zircônio e titânio.
5. **Classe K:** Óleo de cozinha, gorduras e graxa.

Figura 3 – Classes de incêndio



Fonte: Rodrigues Projetos, 2016.

Para cada tipo de incêndio dentro dessas categorias, são necessários cuidados específicos na hora da extinção do fogo, como escolher o extintor apropriado. Por conta disso, o simulador apresentá uma série de diferentes situações e o jogador deverá escolher qual é a melhor forma de extinguir o incêndio.

2.1.3 Extintores

Extintores são recipientes que contêm um determinado agente extintor para combate rápido a princípios de incêndio, ou seja, incêndios em sua fase inicial. Esses elementos classificam-se de acordo com a classe de incêndio a qual se destinam. Os mais comuns são (LIMA; ENCARNAÇÃO, 2016):

- Extintor de água:** É indicado para incêndios de classe A. Contém uma mistura de água e gás que é expelida ao pressionar o gatilho. Essa água expelida é responsável por resfriar o material, tornando a temperatura inferior ao ponto de ignição.
- Extintor de pó químico seco:** É indicado para incêndios de classe B. Contém uma mistura de um composto químico em pó (como o bicarbonato de sódio) e um gás propelente (como o dióxido de carbono) que, ao entrar em contato com as chamas, impede a continuidade da reação em cadeia.
- Extintor de gás carbônico:** É indicado para incêndios de classe C. Contém dióxido de carbono, um agente não condutor de eletricidade, que recobre o fogo e isola o oxigênio, que é indispensável à reação de combustão.

2.2 Motor de jogos

Motor de jogo, ou *game engine*, é uma biblioteca ou um pacote de funcionalidades, cujo objetivo é facilitar determinadas etapas no desenvolvimento de um jogo. Isso ocorre apresentando ao desenvolvedor uma série de ferramentas para que ele não necessite de desenvolver algumas funcionalidades do zero.

Dentre as funcionalidades tipicamente oferecidas por um motor de jogos estão: (i) um motor gráfico para renderizar gráficos 2D ou 3D, (ii) um motor de física para simular coisas como gravidade e detecção de colisão, (iii) suporte a animações e (iv) suporte à criação de efeitos sonoros. Além disso permite abstrair o *hardware*, permitindo que o desenvolvedor crie jogos sem conhecer profundamente a arquitetura da plataforma-alvo.

Nesse projeto, será utilizado o motor de jogos **Pygame**, observada na figura 4. Essa *engine* é um conjunto de módulos em Python voltados para o desenvolvimento de jogos e aplicações multi-media. É gratuita (licença LGPL) e bastante portátil, podendo ser executada em diversas plataformas e sistemas operacionais.

Figura 4 – Logo do Pygame



Fonte: pygame.org, 2021.

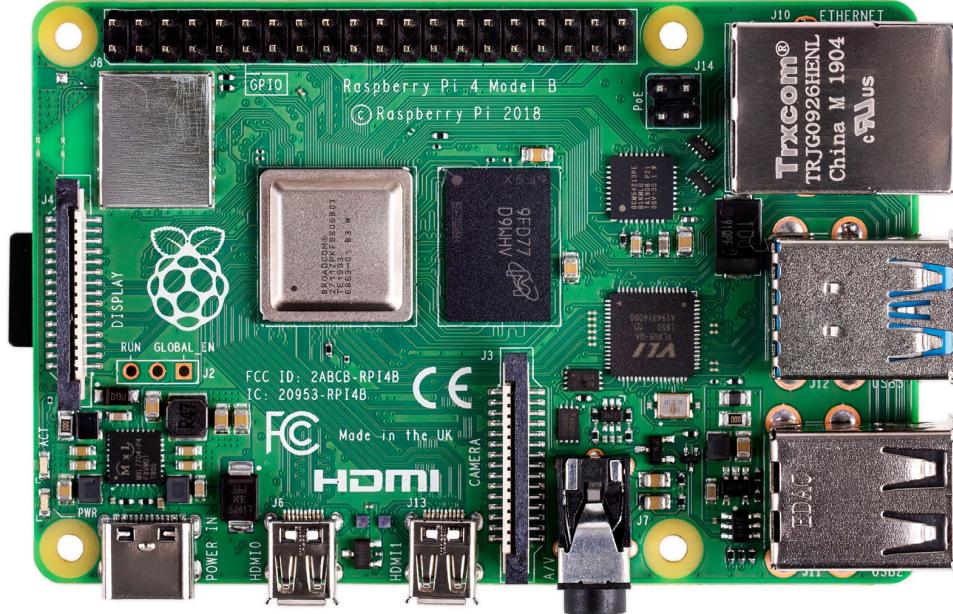
2.3 Plataforma

Para gerar o ambiente virtual e receber os comandos do jogador será utilizado um microcomputador embarcado, cuja principal função é tornar mais simples a utilização do kit, dispensando *hardwares* adicionais como computadores portáteis.

O microcomputador escolhido é o **Raspberry Pi 4 modelo B**. Apesar de ter as dimensões aproximadas de um cartão de crédito, tem nele todo o *hardware* necessário para executar

seu próprio sistema operacional, além de entradas para *mouse* e teclado, saída de vídeo, saída de áudio, interfaces *Universal Serial Bus* - USB e *Ethernet*, e pinos digitais de entrada e saída. O *hardware* descrito pode ser observado na figura 5.

Figura 5 – Raspberry Pi 4, Modelo B.



Fonte: Raspberry Foundation, 2018.

Entre suas especificações técnicas de maior interesse tem-se: Processador Quad Core 1,5 GHz Broadcom BCM2711 64 bits; 4 GB de memória RAM; Adaptador *wireless LAN* e Bluetooth 5.0; 40 pinos GPIO; 2 portas USB 2.0 e 2 portas USB 3.0; 2 saídas micro-HDMI; Entrada para cartão de memória micro SD (onde é armazenado o sistema operacional); Alimentação em 5 V; Dimensões físicas: 85 × 56 × 17 mm;

A interface do microcomputador com o usuário é possível através de alguns sistemas operacionais baseados em Linux. O mais utilizado deles é o **Raspbian**, que é um sistema operacional gratuito baseado no sistema Debian e otimizado para *Raspberry Pi*. Esse é o sistema escolhido para ser utilizado ao nesse trabalho.

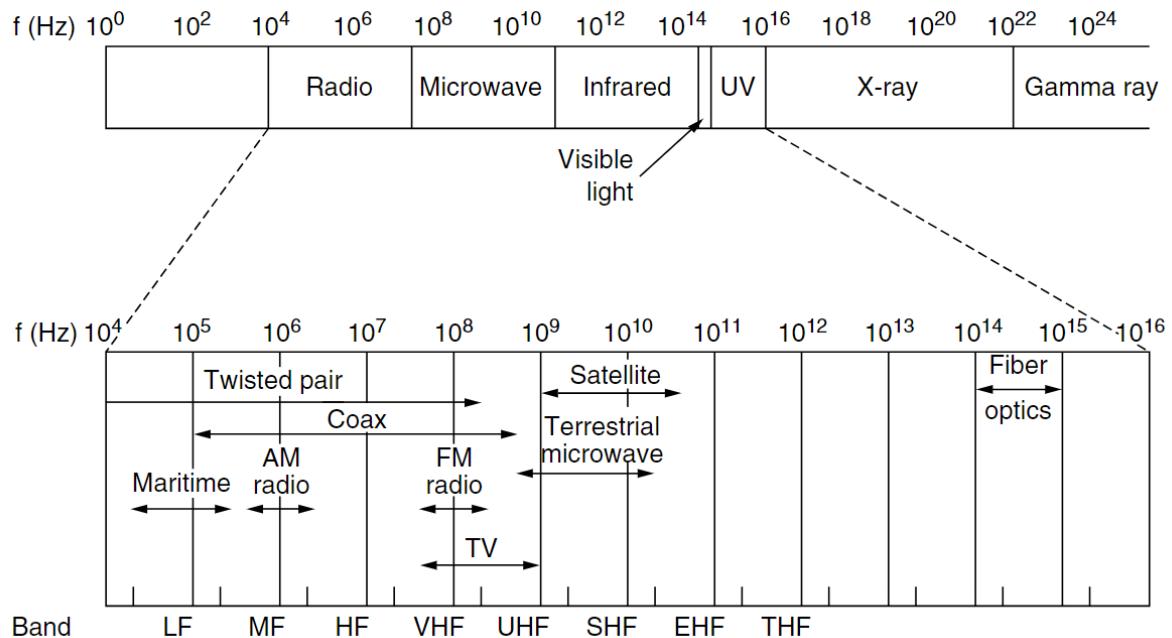
2.4 Comunicação via radiofrequênciа

O movimento de elétrons cria ondas eletromagnéticas que podem se propagar no espaço. Quando uma antena de tamanho adequado é conectada a um circuito elétrico, essas ondas eletromagnéticas podem ser transmitidas de maneira eficiente (transmissor) e recebidas

por um segundo circuito elétrico (receptor) a uma certa distância. Esse é o princípio da comunicação sem fio. (TANENBAUM, 2003)

Dentro do espectro eletromagnético observado na figura 6, uma das porções mais utilizadas para comunicação é a faixa de radiofrequência (10 KHz a 100 GHz). Nessa faixa operam diversas tecnologias comuns do nosso dia-a-dia, como televisores, transmissões de rádio, comunicação via satélites e Wi-fi.

Figura 6 – Espectro eletromagnético



Fonte: Tanenbaum, 2003.

O processo através do qual adicionamos informação a uma onda eletromagnética é chamado de modulação, e a recuperação dessa informação é chamada demodulação. A modulação consiste variar um dos parâmetros de uma onda senoidal portadora $p(t)$ como a da equação (1) acompanhando um sinal de informação $x(t)$, também chamado de sinal modulador. Sendo E_0 a amplitude da onda portadora, f_0 sua frequência e θ o angulo de fase.

$$p(t) = E_0 \cdot \cos(2\pi f_0 t + \theta) \quad (1)$$

A onda portadora, que normalmente tem frequência muito mais alta, serve de suporte para a transmissão do sinal $x(t)$. A principal razão disso é facilitar a irradiação já que, para que uma transmissão ocorra certamente, a antena deve ter dimensões da ordem de

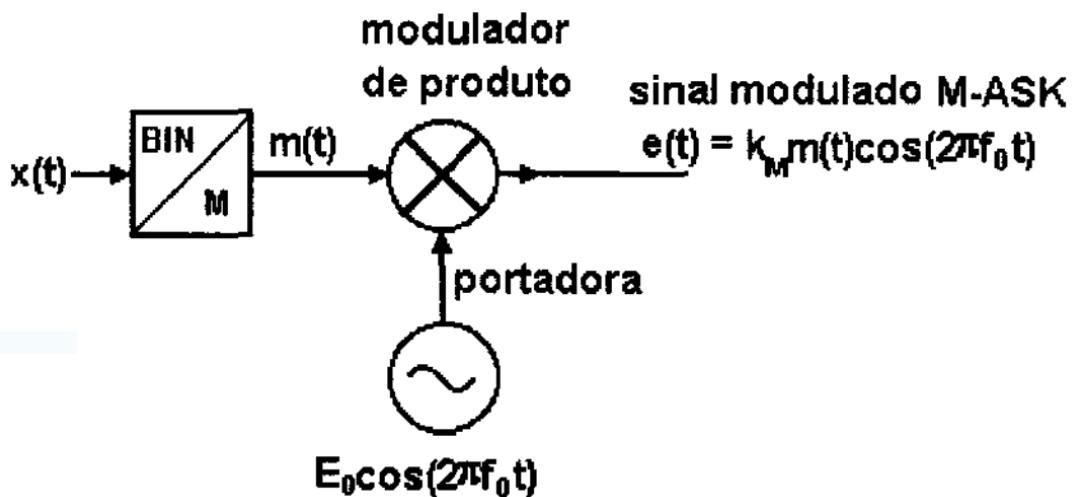
grandeza do comprimento de onda a ser irradiado. Ou seja, a frequência da onda portadora é escolhida pensando na viabilidade das dimensões da antena.

Existem três variáveis que podem ser controladas durante o processo de modulação, gerando assim três tipos de modulação: frequência, fase e amplitude - identificadas, respectivamente, por FSK (*frequency shift keying*), PSK (*phase shift keying*) e ASK (*amplitude shift keying*). Contudo, no escopo desse trabalho vamos tratar apenas da modulação ASK, especialmente voltada para sinais digitais.

2.4.1 Modulação ASK

A modulação de amplitude pode ser obtida num modulador de produto como o que vemos na figura 7. Num modulador como esse um sinal digital de informação $x(t)$ é convertido em um sinal multinível $m(t)$ onde cada combinação de N bits é associada a um valor M de tensão (CARVALHO, 2009).

Figura 7 – Modulador de produto



Fonte: Carvalho, 2008.

O resultado do produto de $m(t)$ e a portadora é um sinal modulado ASK $e(t)$ como o da equação (2), onde k_M é uma constante adimensional.

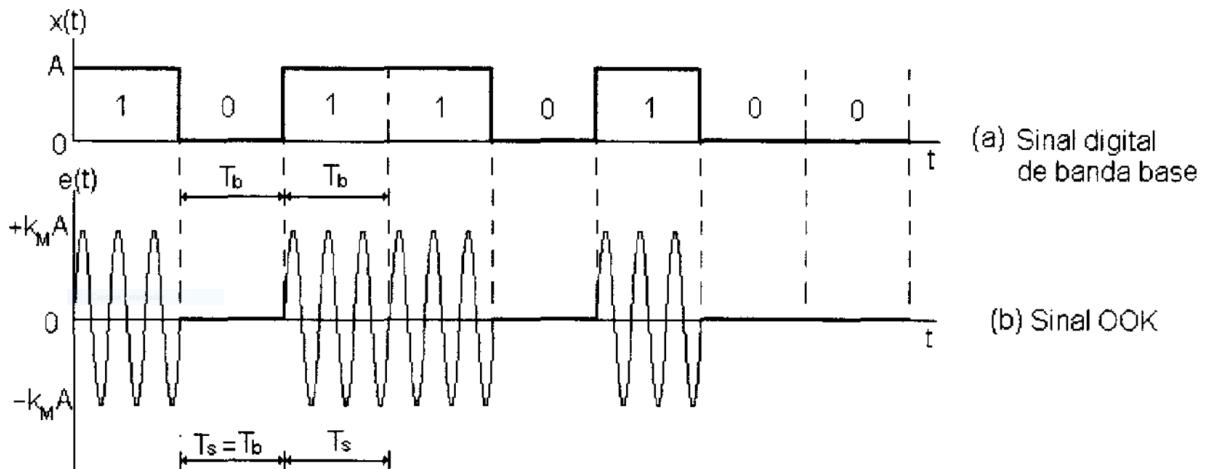
$$e(t) = k_M \cdot m(t) \cdot \cos(2\pi f_0 t) \quad (2)$$

Se o sinal binário unipolar $x(t)$, ou seja, um sinal que assume apenas dois níveis de tensão (zero ou um valor positivo A de amplitude) modula diretamente a portadora,

tem-se o sistema denominado BASK (Binary Amplitude Shift Keying) ou 2-ASK. Um caso particular do BASK é o OOK (On-Off Keying), que se caracteriza por transmitir a portadora quando temos nível lógico alto (bit = 1) e suprimir quando temos nível lógico baixo (bit = 0), como o sinal $e(t)$ da equação (3). Tal modulação é exemplificada na figura 8.

$$e(t) = \begin{cases} k_M A \cos(2\pi f_0 t) = E_0 \cos(2\pi f_0 t) & \dots \text{bit} = 1 \\ 0 & \dots \text{bit} = 0 \end{cases} \quad (3)$$

Figura 8 – Modulação OOK



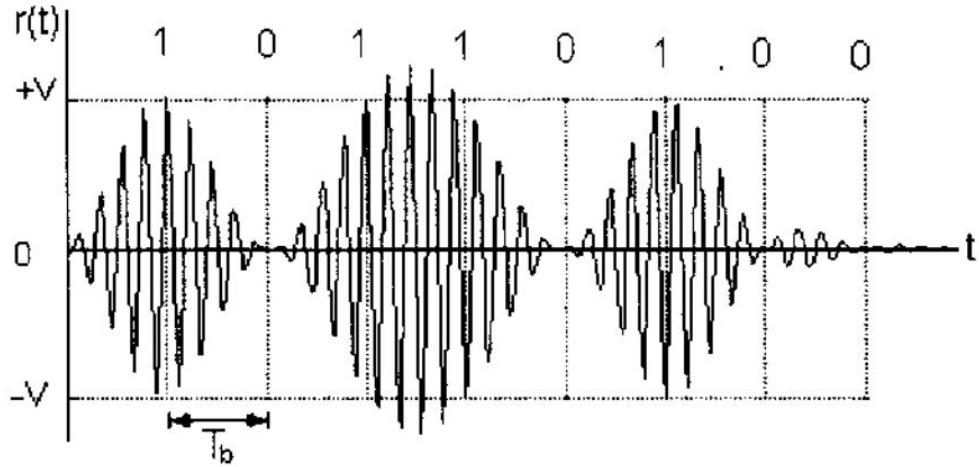
Fonte: Carvalho, 2008.

2.4.2 Demodulação ASK(OOK)

Após a transmissão, cabe ao receptor traduzir o sinal $r(t)$ recebido, como o observado na figura 9, em informação novamente.

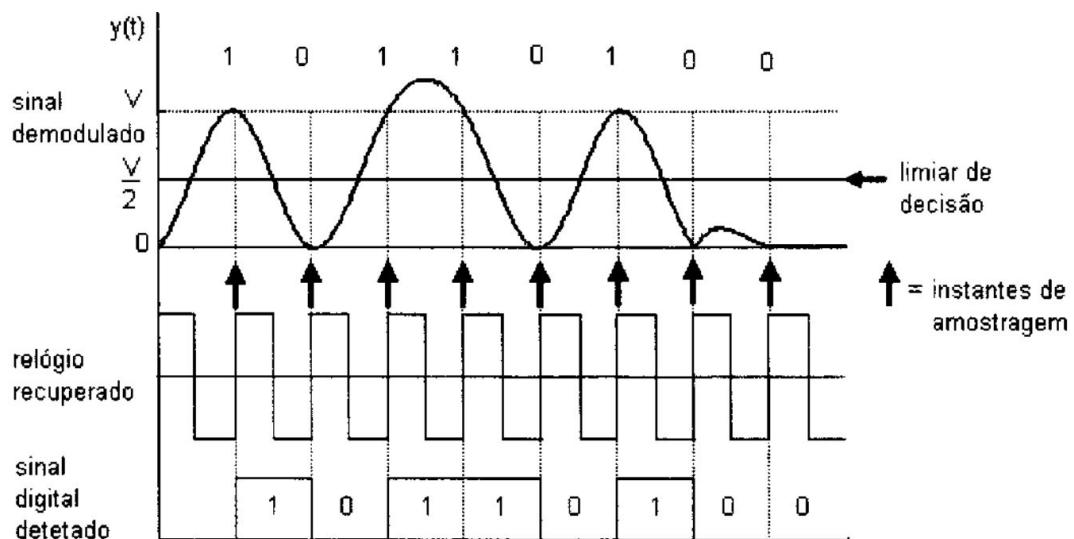
O processo de demodulação tem inicio com um detector de envoltória que transforma o sinal $r(t)$ em uma envoltória $y(t)$, como a da figura 10. Em seguida essa envoltória é amostrada segundo o tempo de 1 bit do sinal de informação $x(t)$. Caso a amostra ultrapasse um limiar de decisão definido, considera-se um bit de nível lógico alto. Caso contrario, considera-se um bit de nível lógico baixo.

Figura 9 – Sinal OOK recebido



Fonte: Carvalho, 2008.

Figura 10 – Regeneração do sinal recebido



Fonte: Carvalho, 2008.

2.5 Avaliação

O questionário é um instrumento desenvolvido cientificamente, composto de um conjunto de perguntas ordenadas de acordo com um critério pré-determinado, que deve ser respondido sem a presença do entrevistador (LAKATOS; MARCONI, 1999) e tem por objetivo coletar dados a respeito de um grupo.

Essa se mostra uma técnica bastante viável e pertinente quando se trata de problemas cujos objetos de pesquisa correspondem a questões de cunho empírico, envolvendo opinião, percepção, posicionamento e preferências dos pesquisados (CHAER; DINIZ; RIBEIRO, 2012). Além disso, garante o anonimato e é de fácil manejo na padronização dos dados, o que garante uniformidade. Por esses motivos, será a principal ferramenta utilizada pra avaliar a qualidade do protótipo desenvolvido.

No intuito de garantir a eficiência dessa ferramenta, são feitas algumas recomendações: (*i*) os temas escolhidos devem estar de acordo com os objetivos da pesquisa; (*ii*) o questionário deve ser limitado em sua extensão e em sua finalidade, pois um questionário muito longo causa cansaço e consequente desinteresse e um questionário muito curto pode não oferecer informações suficientes; (*iii*) as questões devem ser codificadas, a fim de facilitar a análise dos dados obtidos; (*iv*) deve estar acompanhado de orientações sobre como respondê-lo; (*v*) o aspecto e a estética devem ser observados (LAKATOS; MARCONI, 1999).

Após a elaboração do questionário, recomenda-se que se faça um pré-teste, ou seja, um teste antes de sua utilização definitiva. Esse teste pode ser feito por meio de sua aplicação a um pequeno grupo de respondentes. O pré-teste serve também para verificar se o questionário apresenta três importantes elementos: (*i*) fidedignidade: os resultados serão os mesmos, independentemente de quem o aplicou; (*ii*) validade: os dados coletados são necessários à pesquisa; (*iii*) operatividade: o vocabulário é acessível e o significado é claro (LAKATOS; MARCONI, 1999)

3 METODOLOGIA E ETAPAS DE DESENVOLVIMENTO

O desenvolvimento do simulador será composto das atividades listadas a seguir:

Etapa1. Estudo teórico

1. Estudo sobre técnicas de desenvolvimento de jogos, especialmente voltadas para a plataforma Pygame.
2. Revisão a respeito de protocolos de comunicação sem fio.

Etapa2. Especificações do projeto

3. Dimensionar as adequações necessárias ao extintor para ser utilizado como controle.
4. Definir o protocolo de comunicação a ser utilizado entre controle e console.
5. Definir os diferentes cenários de risco a serem simulados, de acordo com as necessidades do Corpo de Bombeiros.

Etapa3. Desenvolvimento do protótipo

6. Realizar as adequações físicas nos extintores.
7. Desenvolver o jogo no ambiente virtual.

Etapa4. Teste do protótipo

8. Realizar testes práticos e avaliar os resultados.

Etapa5. Documentação

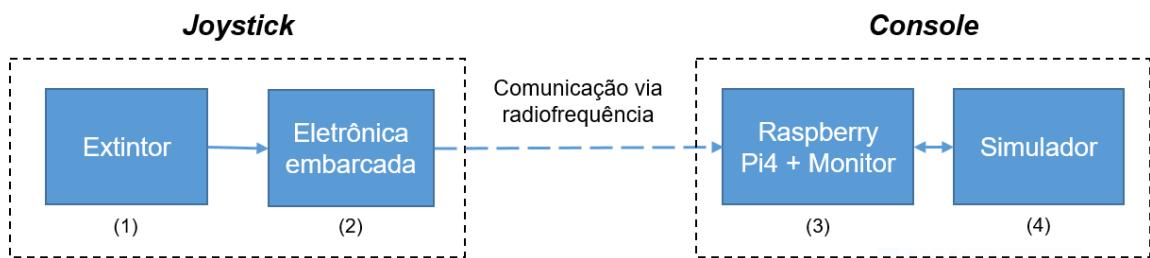
9. Documentar as etapas de desenvolvimento e desenvolver a dissertação do projeto de graduação.

4 IMPLEMENTAÇÃO

A implementação deste projeto se divide em três principais etapas:

1. *Hardware*, observado, principalmente, nos blocos 1 e 2 da figura 11.
2. *Firmware*, que controla a eletrônica embarcada do bloco 2 da figura 11.
3. *Software*, observado no bloco 4 da figura 11 sendo executado na Raspberry Pi.

Figura 11 – Diagrama geral



Fonte: Produção do próprio autor.

O simulador de incêndios é constituído pelo joystick, que é a junção do extintor de incêndios (bloco 1 da figura 11) com a eletrônica embarcada necessária para identificar o tipo de extintor sendo pressionado e enviar essa informação para o console via radiofrequência. Esse console contém um receptor de radiofrequência conectado a Raspberry Pi (bloco 3 da figura 11) a fim de controlar o simulador sendo executado.

4.1 Hardware

A eletrônica do projeto contém os componentes listados a seguir:

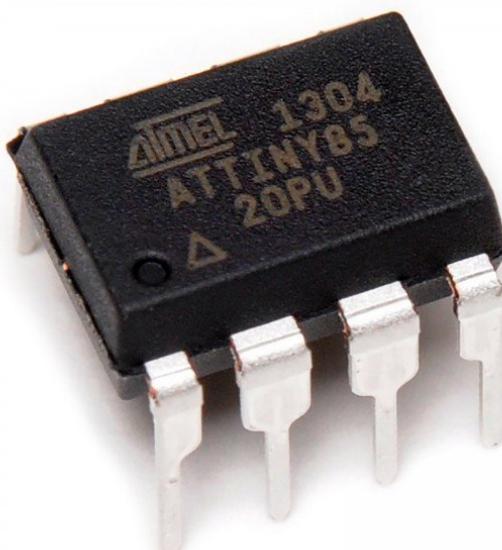
1. Microcontrolador;
2. Módulo transmissor RF e módulo receptor RF;
3. Bateria + Regulador de tensão;
4. LED indicador, botão e conectores;

4.1.1 Microcontrolador

O microcontrolador Attiny85 tem duas principais funções no joystick: identificar quando o extintor é pressionado e enviar essa informação através de um protocolo de comunicação até a Raspberry Pi.

A escolha desse modelo de microcontrolador foi feita, principalmente, por se tratar de uma plataforma amplamente conhecida, de fácil utilização e custo acessível. Além disso, por exercer uma função que não exige tanto do hardware, foi possível optar por um controlador pequeno e de baixo consumo energético, que pode ser observado na figura 12.

Figura 12 – Attiny85



Fonte: Bau da eletrônica, 2021.

Algumas especificações do Attiny85:

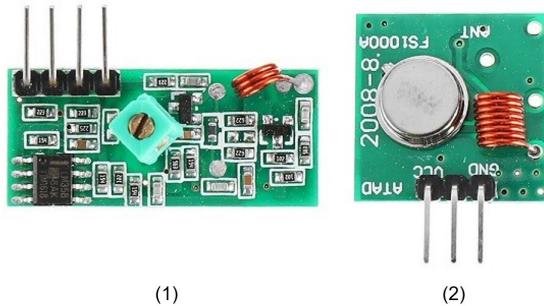
- Fabricado pela Atmel
- 8 pinos DIP, sendo 6 deles GPIO
- Memória RAM de 512 Bytes
- Frequência máxima de 20 MHz
- Dimensões de 9,3 x 6 x 3,3 mm (comprimento x largura x altura)

4.1.2 Módulo transmissor RF e módulo receptor RF

Módulos RF são componentes básicos para comunicação em radiofrequência presentes em vários sistemas do nosso dia-a-dia como sistemas de alarme, controles remoto, sistema de aquisição de dados entre outros.

Para a comunicação do Attiny85 com a Raspberry são utilizados dois módulos de radiofrequência de modulação ASK. O módulo 1 da figura 13 é o receptor (MX-05V), que fica conectado na Raspberry Pi através dos pinos GPIO. Já o módulo 2 (MX-FS-03V) da figura 13 é o receptor que fica junto do restante da eletrônica embarcada no joystick.

Figura 13 – Módulos RF



Fonte: Adaptado de Filipeflop, 2021.

Para melhorar a comunicação entre os dois módulos, foi conectada uma antena helicoidal metálica de tamanho apropriado a frequência de operação ao módulo transmissor, aumentando o alcance da comunicação. Essa antena pode ser vista na figura 14

Figura 14 – Antena espiral 433 MHz



Fonte: Filipeflop, 2021.

Dentre as características principais dos módulos, destacam-se:

- Transmissor: (i) alcance de 20 a 200 metros, (ii) tensão de operação de 3,5-12 V, (iii) frequência de transmissão de 433 MHz, (iv) taxa de transferência de 4 KB/s, (v) potência de transmissão de 10 mW e (vi) dimensões de 19 x 19 mm.
- Receptor: (i) tensão de operação de 5 V, (ii) frequência de operação de 433 MHz, sensibilidade de -105dB e (iii) dimensões de 30 x 14 x 7 mm.

4.1.3 Bateria + Regulador

Para alimentação do circuito foi utilizado uma bateria recarregável de 9 V e 250 mAh que pode ser observada na figura 15. O principal motivo para a escolha dessa bateria é a facilidade com que ela e seu carregador podem ser encontrados no mercado. Além disso, conseguem armazenar carga suficiente para a aplicação.

Figura 15 – Bateria de 9 V

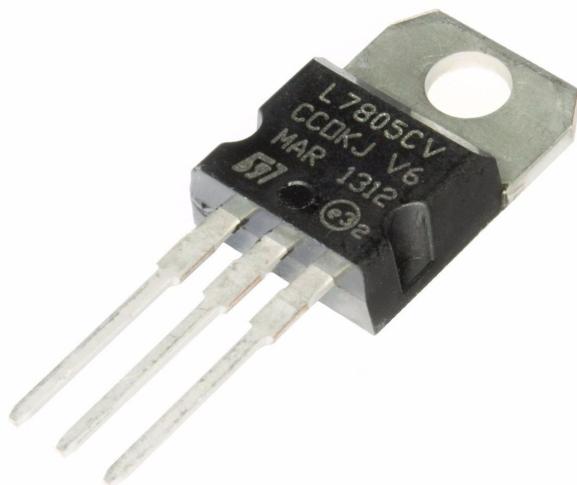


Fonte: Grupo Elgin, 2021.

Apesar dos pontos positivos citados da bateria, sua tensão de funcionamento é incompatível com os demais componentes que compõe o joystick. Por isso, é necessário a utilização de um regulador de tensão que reduza a tensão de operação da bateria de 9 V para a tensão de operação dos demais componentes, ou seja, 5 V. O regulador escolhido foi o L7805CV da STMicroelectronics que pode ser visto na figura 16.

Algumas características importantes desse componente:

Figura 16 – Regulador de tensão L7805CV



Fonte: Achei Componentes Eletrônicos, 2021.

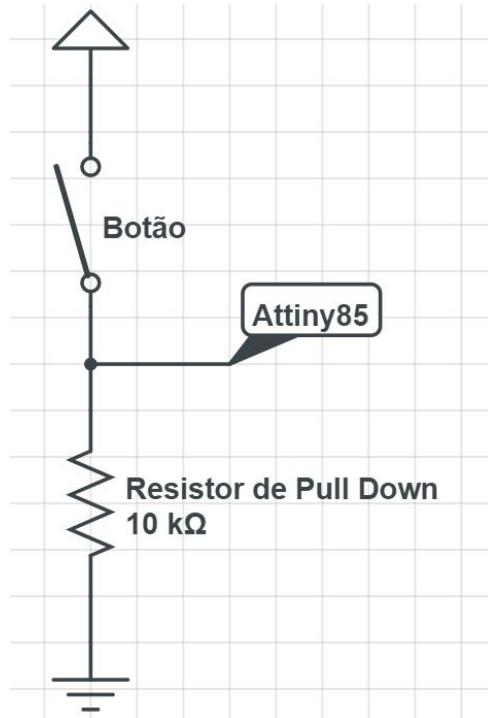
- Tensão de entrada de 5-35 V
- Tensão de saída de 5 V
- Temperatura operacional de até 125 °C
- Corrente de saída de até 1 A

4.1.4 LED indicador, botão e conectores

É utilizado um botão acoplado ao extintor de incêndios que é pressionado junto do gatilho do extintor. Esse botão é conectado a um resistor de *pull down* e à um pino de entrada do Attiny85, como o observado na figura 17. Dessa forma, enquanto o botão não é pressionado, a saída permanece conectada ao terra (0 V). Uma vez que o botão é pressionado, essa tensão na entrada do microcontrolador muda para a tensão da fonte que, nesse caso, é de 5 V.

Foi adicionado um LED indicador vermelho de 3 mm de diâmetro e 20 mA em série com um resistor de $330\ \Omega$ para auxiliar na detecção de problemas durante a utilização. Esse LED pode ser utilizado para identificar se o *firmware* entrou em uma determinada rotina, por exemplo.

Figura 17 – Circuito do botão



Fonte: Produção do próprio autor.

Para realizar a conexão da bateria à PCB foi utilizado um *clip* conector como o da figura 18. Já a conexão do botão foi realizada com um conector JST de duas vias como o da figura 18.

Figura 18 – Conectores



(1) Clip conector de bateria 9 V

(2) Conector JST polarizado de 2 vias

Fonte: Produção do próprio autor.

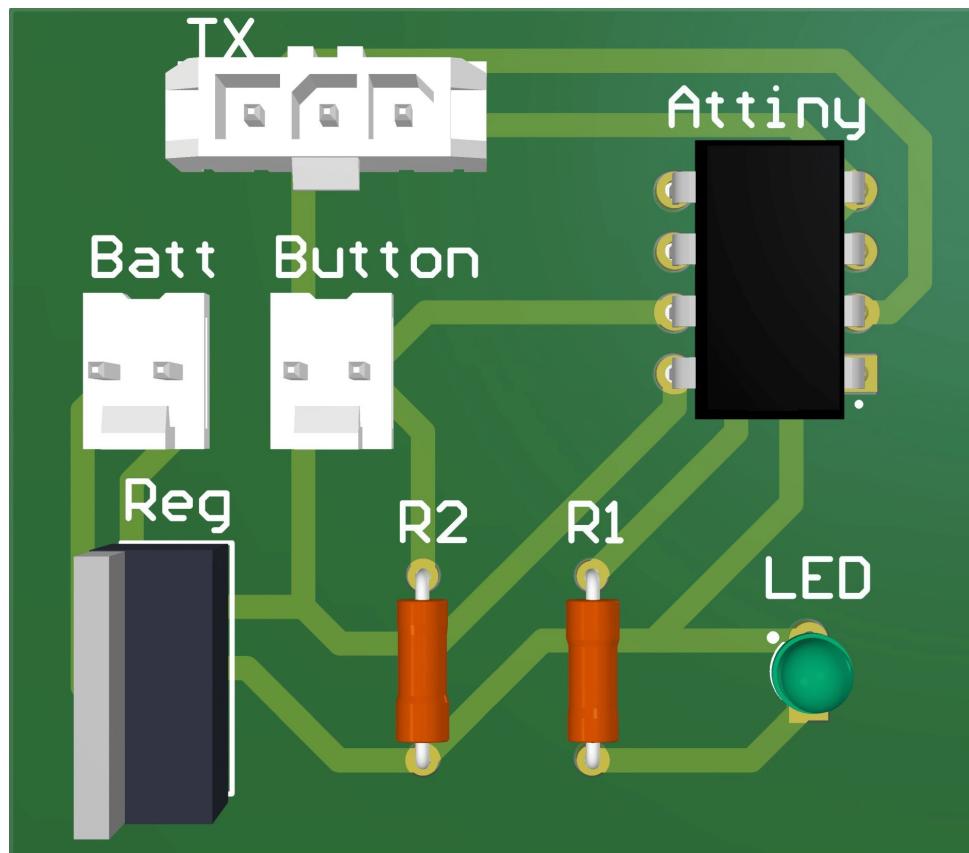
4.1.5 Placa de circuito impresso

Após a definição dos componentes necessários, o circuito foi desenvolvido em 4 etapas principais:

1. Desenvolvimento do circuito em 2D e 3D, como o da figura 19;
2. Montagem em *protoboard* e testes iniciais;
3. Confecção em placa de fenolite, como a da 20;
4. Confecção especializada, como a da figura 21;

A simulação de um modelo em três dimensões da PCB é um passo importante para estimarmos o tamanho dos componentes e como serão dispostos, além de uma estimativa do tamanho final da PCB. Podendo assim, projetar como essa ficará acomodada na versão final do projeto.

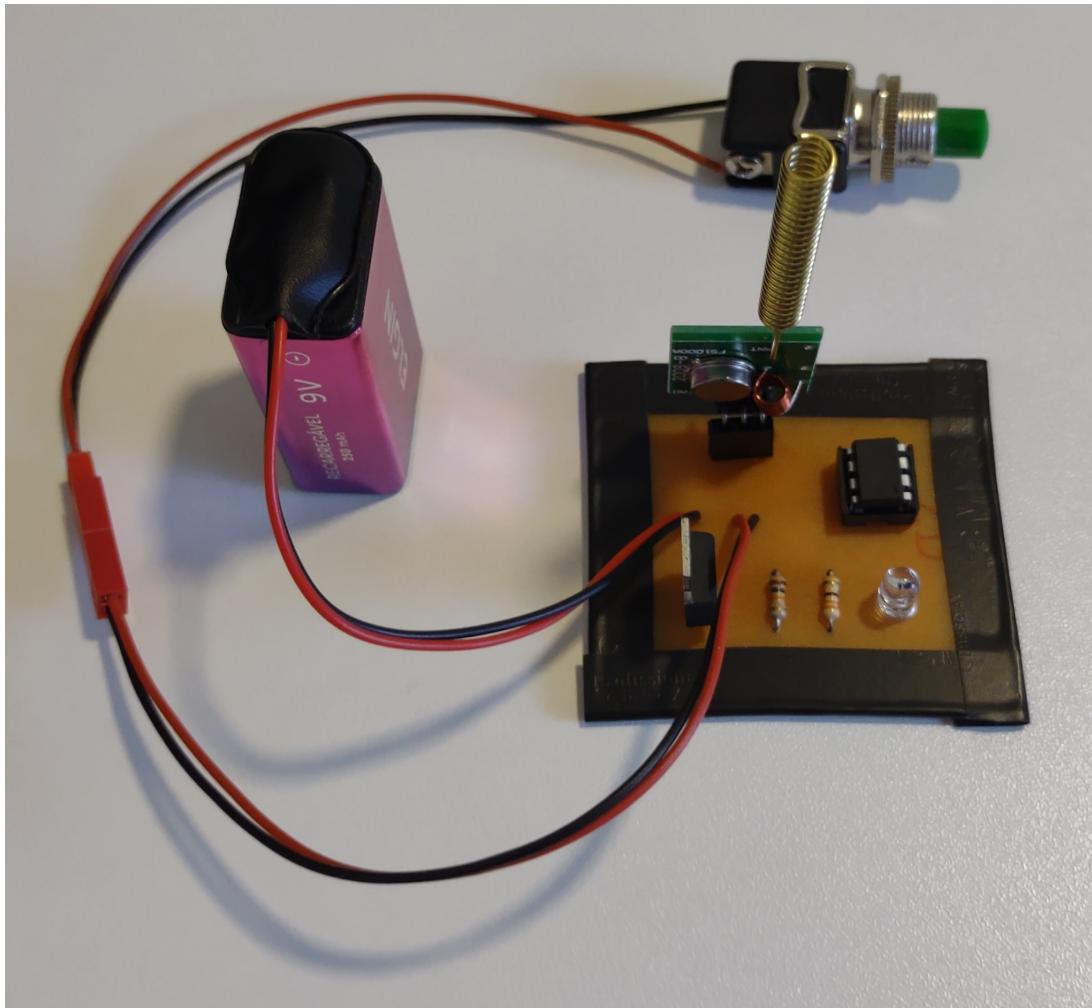
Figura 19 – Simulação em 3D da placa de circuito impresso



Fonte: Produção do próprio autor.

Após a modelagem do circuito e validação com testes em protoboard, foi confeccionada uma versão artesanal (figura 20), em fenolite, a fim de verificar se o roteamento do circuito foi adequado. Nela foram conectados todos os componentes necessários e, em seguida, foram realizados testes observando se as mensagens transmitidas eram recebidas corretamente pelo receptor presente na Raspberry Pi.

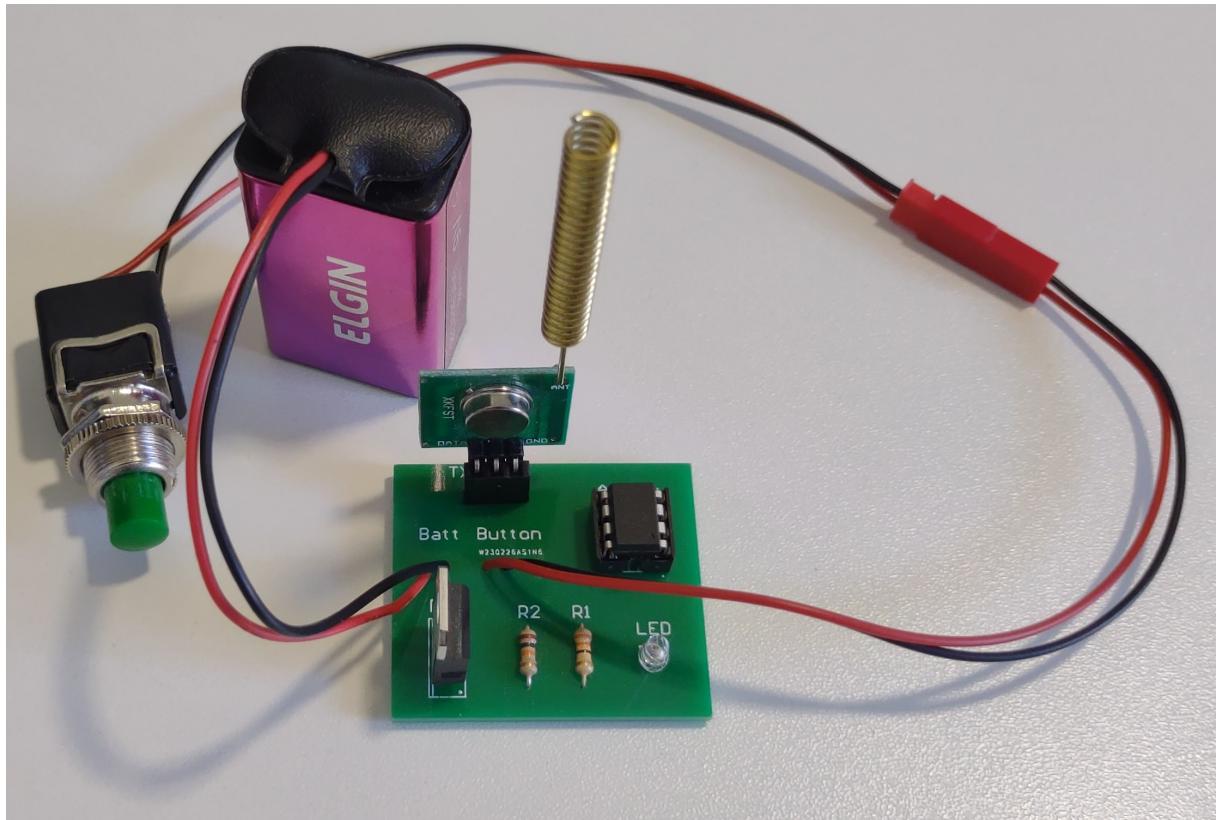
Figura 20 – Placa de fenolite corroída em percloreto de ferro



Fonte: Produção do próprio autor.

Uma vez que o roteamento da PCB foi validado, foram confeccionadas novas placas de circuito impresso como a da figura 21, dessa vez, por uma empresa especializada. A opção pela prototipagem especializada gera alguns ganhos significativos na qualidade de confecção como: metalização de furos, máscara de solda, identificação dos componentes na camada sem cobre (para o caso de PCBs com apenas uma camada) e, consequentemente, reduz ruídos e aumenta a vida útil da PCB.

Figura 21 – PCB com fabricação especializada



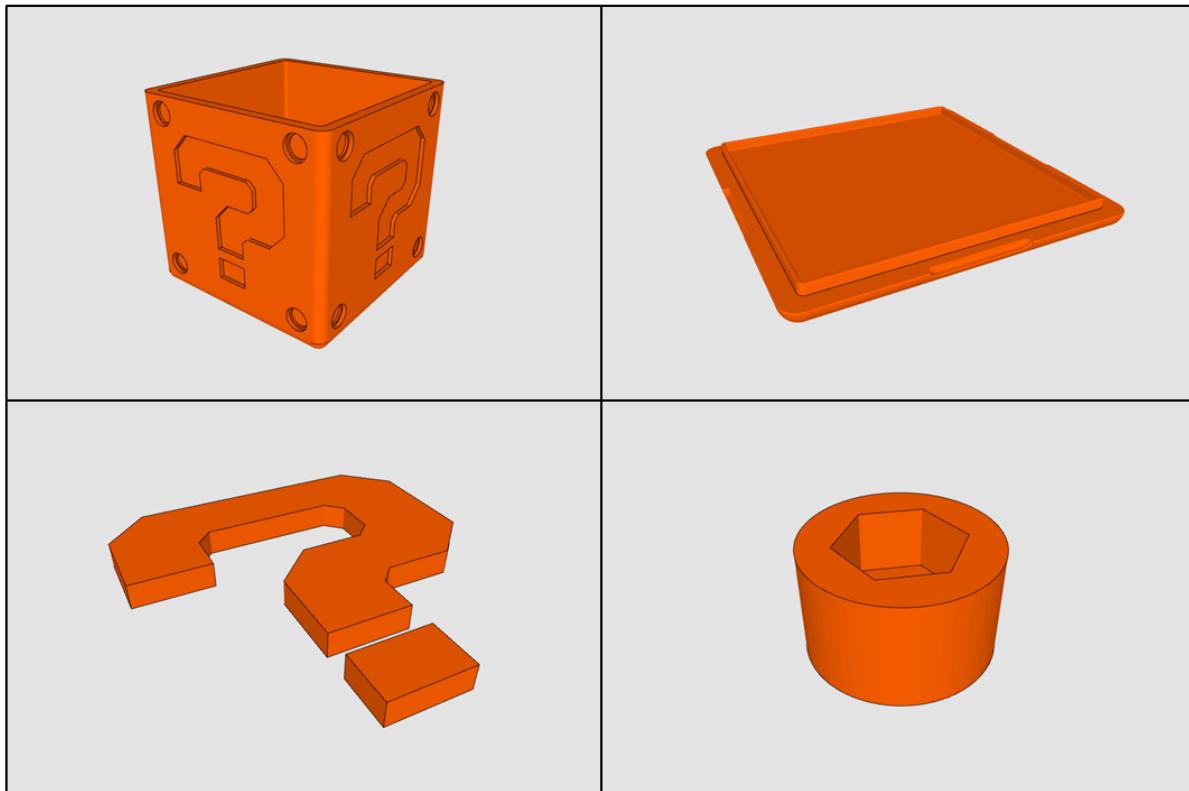
Fonte: Produção do próprio autor.

4.1.6 Caixa para acomodar os componentes

Para acomodar os componentes eletrônicos junto ao extintor de incêndios foi utilizada uma caixa em formato cúbico impressa em material termoplástico (PLA) através de uma impressora 3D. Por ser um simulador voltado majoritariamente a um público infantil, a ideia do formato da caixa é tornar a utilização uma atividade mais lúdica.

A fim de facilitar a confecção desse modelo, a impressão foi dividida em 4 peças diferentes: a caixa, a tampa e dois ornamentos que devem ser impressos em cores diferentes para ter destaque. Após a impressão, há uma segunda etapa em que os ornamentos são colados na lateral da caixa. Esses modelos 3D são os presentes na figura 22.

Figura 22 – Peças para montagem da caixa



Fonte: Produção do próprio autor.

4.1.7 Montagem do controle

A montagem final do controle é observada na figura 23. Nessa montagem temos o botão fixado na alça de transporte através de um furo localizado sob o botão. Esse botão está conectado a PCB através dos conectores JST que passam por baixo da alça e chegam até a caixa na lateral.

Os demais componentes ficam localizados nessa caixa fixada na lateral do extintor. Apesar de se encontrar aberta na figura 23, durante a utilização ela será devidamente fechada para evitar contato direto do usuário com o circuito.

Dessa maneira, apesar das situações de princípio de incêndio serem simuladas, o usuário conseguirá observar características importantes da ferramenta como: peso, formato, rótulos informativos e a etiqueta de validade.

Figura 23 – Extintor com circuito



Fonte: Produção do próprio autor.

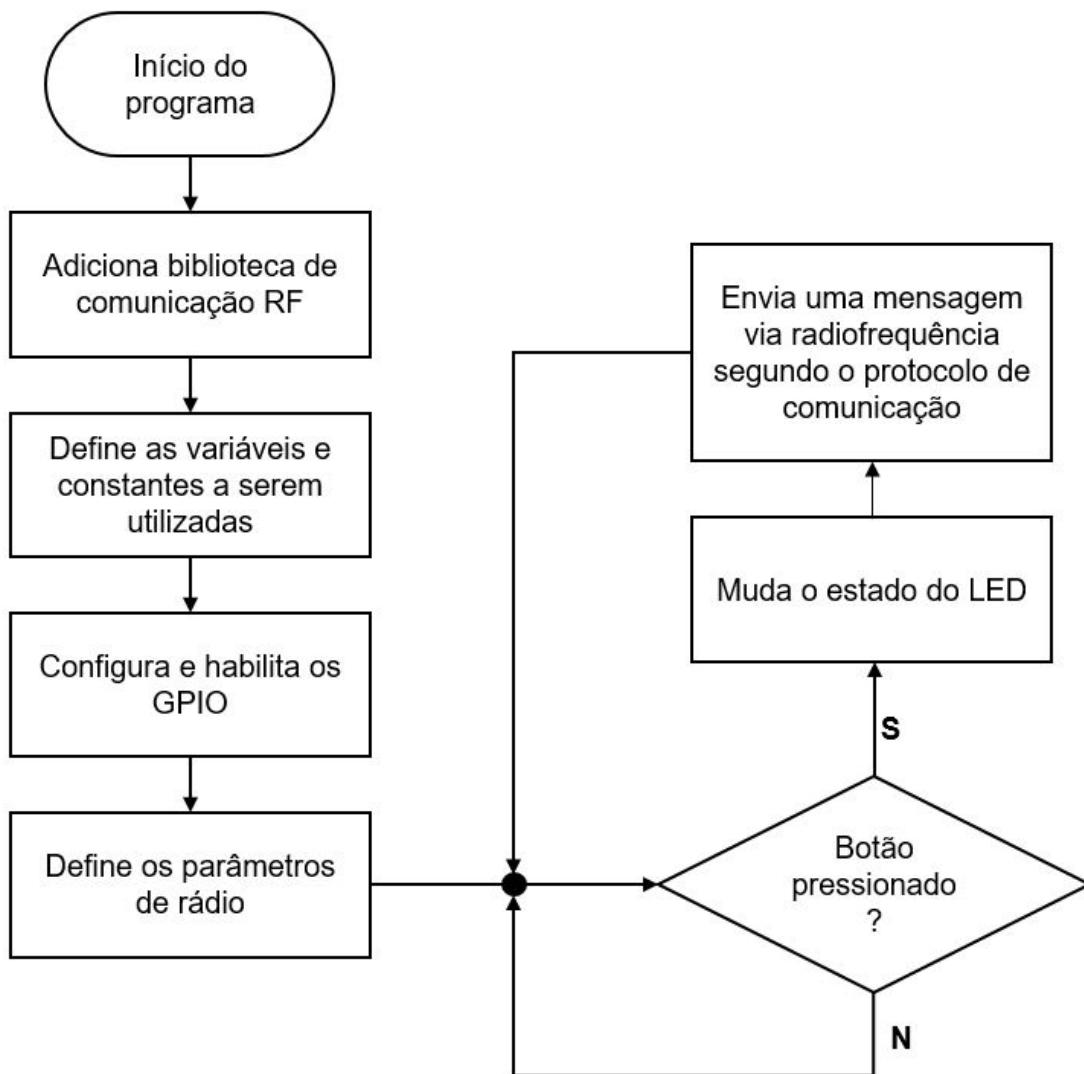
4.2 Firmware

O *firmware*, que pode ser observado no APÊNDICE A desse documento, tem três partes principais:

1. **Definições iniciais:** nessa seção são feitas as definições de variáveis, configurações dos pinos GPIO e definição dos parâmetros do módulo RF.
2. **Checagem do botão:** nessa seção ocorre a checagem do estado do botão, incluindo tratamento de *debounce*.
3. **Envio da informação:** por fim, ao pressionar o botão, ocorre o envio dos dados de acordo com o protocolo de comunicação

Essas rotinas são organizadas de acordo com o que é apresentado no fluxograma da figura 24

Figura 24 – Fluxograma simplificado do *firmware*



Fonte: Produção do próprio autor.

Na tabela 1 são descritas as principais funções presentes no código.

Tabela 1 – Principais Rotinas

Nome	Parâmetros	Aplicação	Retorno
pinMode	O número do pino a ser utilizado e sua função (<i>input</i> / <i>output</i>).	Define a utilização de um pino e se ele será um pino de entrada ou de saída.	-
digitalRead	O número do pino digital a ser verificado.	Verifica o valor digital de um determinado pino.	O valor digital observado (<i>high</i> / <i>low</i>).
digitalWrite	O número do pino e seu valor digital.	Modifica o valor do pino digital.	-
millis	-	Retorna o número de milissegundos passados desde o inicio da execução do programa atual.	O tempo em milissegundos.
enableTransmit	O número do pino conectado ao pino de dados do módulo RF.	Habilita a comunicação RF em um determinado pino	-
setPulseLength	O tamanho do pulso a ser enviado.	Define o parâmetro de tamanho de pulso da mensagem RF	-
setProtocol	O número do protocolo que será utilizado.	Define o protocolo utilizado no envio da mensagem.	-
setRepeatTransmit	A quantidade de vezes que uma mensagem será enviada.	Repete a transmissão de uma mensagem um determinado número de vezes.	-
send	O dado a ser enviado.	Envia o dado definido. Caso seja um inteiro, deve-se definir o tamanho da mensagem.	-

Fonte: Produção do próprio autor.

4.2.1 Protocolo de comunicação

O os dados são enviados seguindo o protocolo de comunicação da figura 25.

Figura 25 – Protocolo de comunicação



Fonte: Produção do próprio autor.

Nesse protocolo são enviados 8 bits, sendo os quatro primeiros um preambulo que identifica se tratar de um dado do joystick. Os dois bits seguintes identificam de qual dos joysticks esse dado se origina. E os dois últimos bits finalizam o Byte.

Considerando os três tipos mais comuns de extintores de incêndio citados anteriormente, os dados são organizados da seguinte forma:

- Dado = "01", para extintores de água;
- Dado = "10", para extintores de pó químico;
- Dado = "11", para extintores de gás carbônico;

4.3 Software

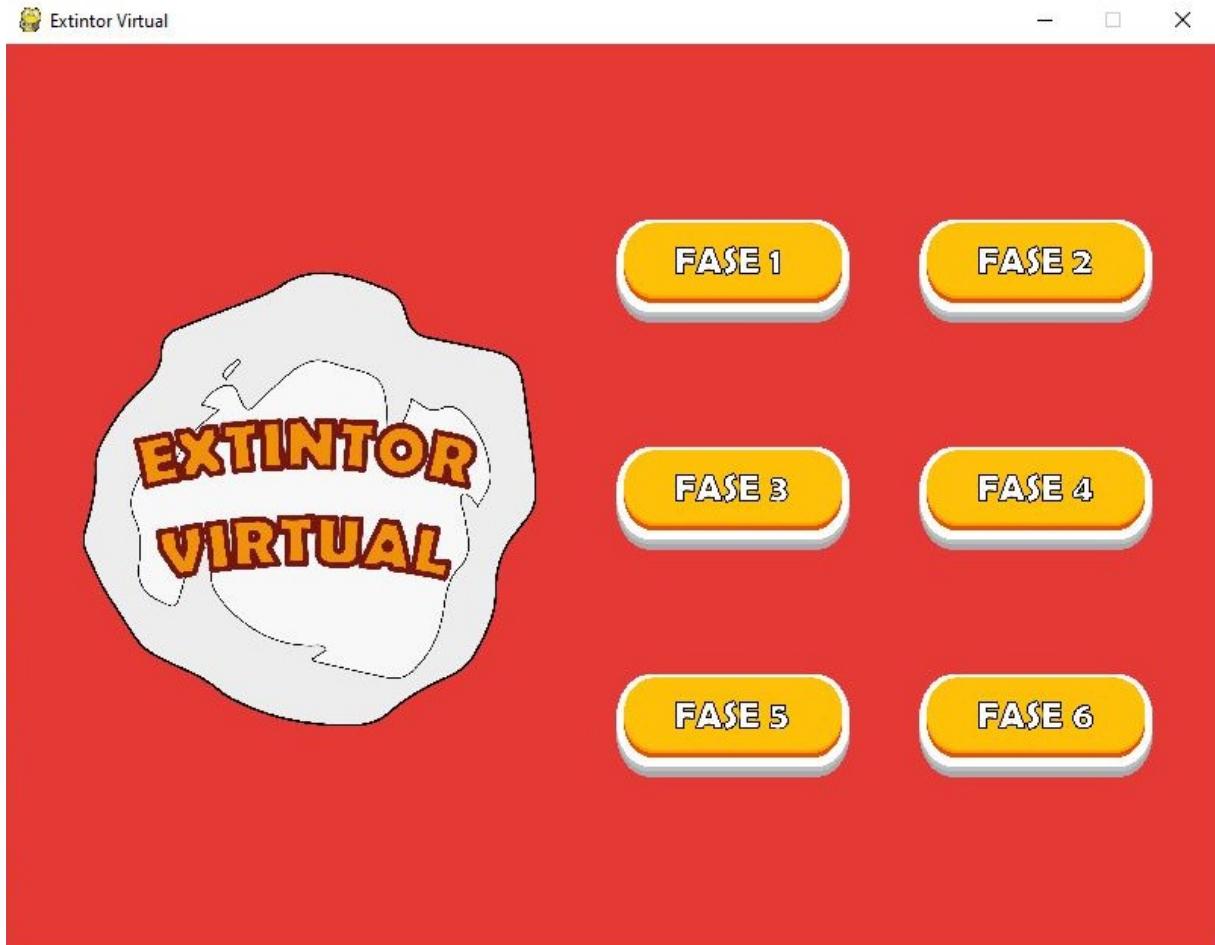
O software do simulador é todo desenvolvido em Python. Tanto no que diz respeito ao recebimento de informações através da biblioteca rpi_rf, quanto na interface gráfica através da *engine* Pygame.

4.3.1 Criação e utilização dos menus

A tela do menu principal contém a logo com o nome do simulador a esquerda (Extintor Virtual) e a direita se localizam os diferentes cenários a serem escolhidos pelo usuário,

como pode ser observado na figura 26. Essa escolha deve ser feita através da utilização de um mouse conectado a Raspberry Pi.

Figura 26 – Tela inicial do jogo

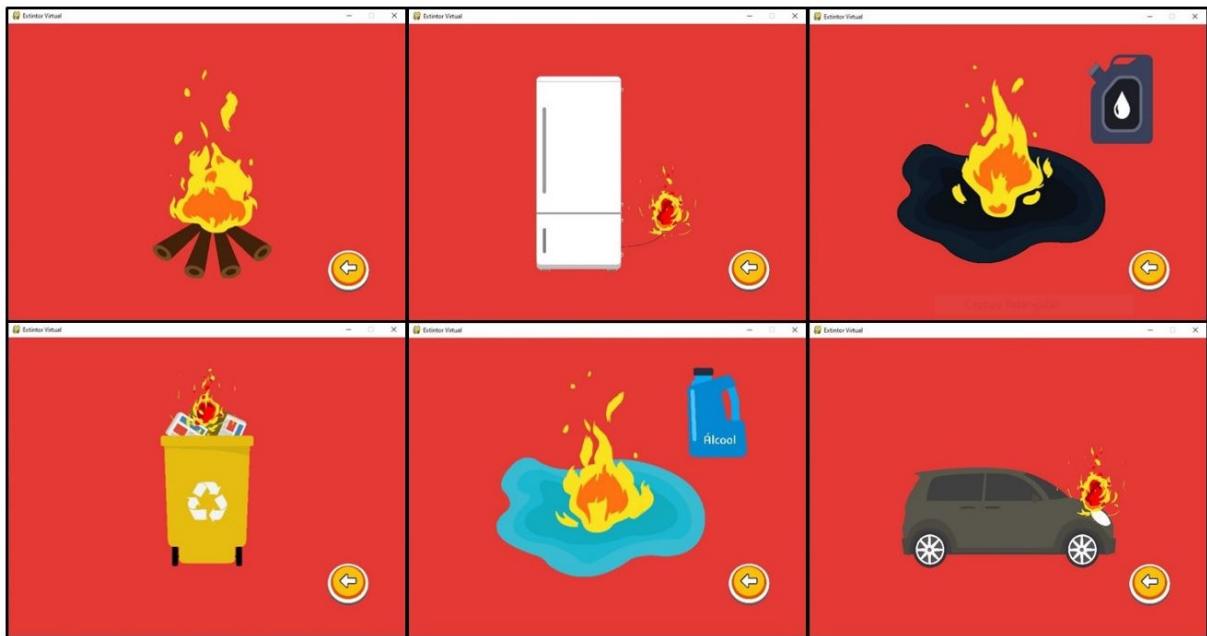


Fonte: Produção do próprio autor.

Cada um desses cenários ilustra uma situação de princípio de incêndio com materiais de diferentes, gerando diferentes classes de incêndio. O objetivo do usuário é utilizar o extintor correto para resolver cada classe de incêndio.

Por exemplo, considerando a primeira fase do jogo vista no canto superior esquerdo da figura 27, temos um princípio de incêndio em pedaços de madeira. Esse é um princípio de incêndio enquadrado na CLASSE A. Portanto, é necessário um extintor adequado a esse tipo que, nesse caso, pode ser o extintor de água. Caso o usuário utilize o extintor correto, o fogo simulado é apagado.

Figura 27 – Fases do jogo

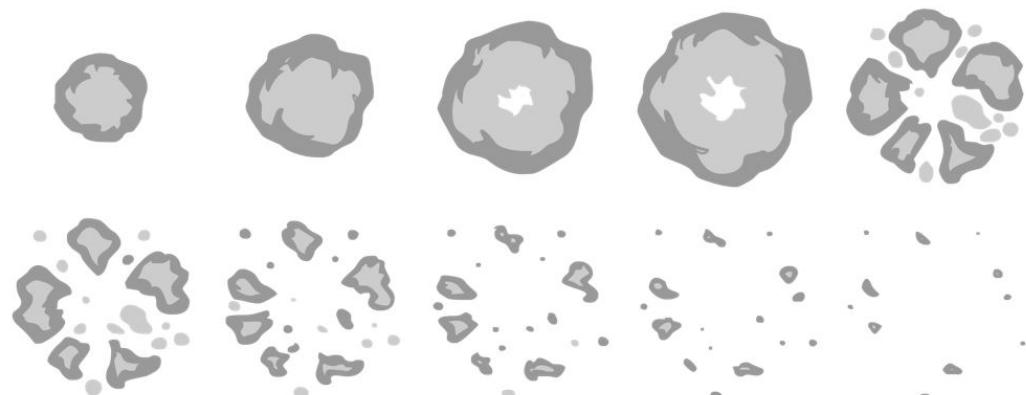


Fonte: Produção do próprio autor.

4.3.2 *Sprites* e animações

O simulador é desenvolvido através de dois tipos diferentes de *sprites*. Aqueles que são utilizados para compor o cenário e os botões do menu e, portanto, estáticos e outros como os da figura 28, que são sobrepostos na tela em sequência a uma certa velocidade e assim criam a animação de fumaça.

Figura 28 – Sprite de fumaça



Fonte: Produção do próprio autor.

Essa sobreposição das imagens é programada para ocorrer a uma determinada taxa de quadros. A cada atualização da tela, é desenhado o próximo *sprite* da sequência da animação. Além disso, a cada novo quadro, o programa também checa se há novas mensagens do controle e, caso aja, realiza as modificações necessárias. A função principal tem estrutura semelhante ao código mostrado a seguir.

```

1 while True:
2     # set framerate
3     clock.tick(12)
4     redrawGameWindow()
5
6     # check if there is a new message
7     if rfdevice.rx_code_timestamp != timestamp:
8         timestamp = rfdevice.rx_code_timestamp
9
10    # check if is a joystick message
11    if str(rfdevice.rx_code) == '157':
12        # throw smoke on screen
13        smokeOn = True
14        # show message log
15        logging.info(str(rfdevice.rx_code) +
16                      " [pulselength " + str(rfdevice.rx_pulselength) +
17                      ", protocol " + str(rfdevice.rx_proto) + "]")
18        time.sleep(0.01)
19
20    # if the x button is pressed, quit
21    for event in pygame.event.get():
22        if event.type == pygame.QUIT:
23            rfdevice.cleanup()
24            sys.exit()
```

A função *clock.tick(12)* define a taxa de atualização de quadros do jogo. Em seguida é chamada a função *redrawGameWindow()* que atualiza os *sprites* da tela. Depois que a tela é atualizada, é checado se há novas mensagens vindas do controle e, caso haja, uma *flag* é definida a fim de ativar a fumaça na tela e apagar a animação de fogo. Por fim, há uma função que finaliza o jogo caso o botão de sair seja apertado.

5 RESULTADOS E TRABALHOS FUTUROS

5.1 Custos

Considerando 3 controles e 1 console, teremos os seguintes custos descritos na tabela 2:

Tabela 2 – Custos do projeto

Item	Custo unitário	Quantidade	Custo total
Attiny85	R\$ 22,90	3 Unidades	R\$ 68,70
Regulador L7805CV	R\$ 2,50	3 Unidades	R\$ 7,50
Resistor 0,25 W	R\$ 0,10	6 Unidades	R\$ 0,60
Clip de bateria 9 V	R\$ 2,40	3 Unidades	R\$ 7,20
Conector JST polarizado	R\$ 3,90	3 Unidades	R\$ 11,70
Botão	R\$ 4,50	3 Unidades	R\$ 13,50
Módulos RF	R\$ 12,90	3 Unidades	R\$ 38,70
Antena 433 MHz	R\$ 4,90	3 Unidades	R\$ 14,70
Bateria 9V	R\$ 33,50	3 Unidades	R\$ 100,50
LED 3 mm vermelho	R\$ 0,20	3 Unidades	R\$ 0,60
PCB	R\$ 30,00	3 Unidades	R\$ 90,00
PLA	R\$ 0,30	90 Metros	R\$ 27,00
Raspberry Pi 4 + fonte	R\$ 709,90	1 Unidade	R\$ 706,90
Total			R\$ 1093,60

Fonte: Produção do próprio autor.

REFERÊNCIAS

- CARVALHO, R. M. Comunicações analógicas e digitais. [S.l.]: LTC, 2009. Citado na página 20.
- CHAER, G.; DINIZ, R. R. P.; RIBEIRO, E. A. A técnica do questionário na pesquisa educacional. Revista Evidência, v. 7, n. 7, 2012. Citado na página 23.
- COX, J. Improve Learning with Hands-On Classroom Games, Activities. 2014. Disponível em: <<http://www.teachhub.com/improve-learning-hands-classroom-games-activities>>. Acesso em: 01 maio 2018. Citado na página 12.
- HOLSTERMANN, N.; GRUBE, D.; BÖGEHOLZ, S. Hands-on activities and their influence on students' interest. Research in Science Education, Springer, v. 40, n. 5, p. 743–757, 2010. Citado na página 12.
- LAKATOS, E. M.; MARCONI, M. d. A. Técnicas de pesquisa: planejamento e execução de pesquisas, elaboração, análise e interpretação dos dados. São Paulo: Atlas, 1999. Citado 2 vezes nas páginas 22 e 23.
- LEE, V. S. Involved students in "hands-on" project such as research, case studies, or real-life activities. IDEA, 2012. Citado na página 12.
- LIMA, S. de T. C.; ENCARNAÇÃO, L. G. Curso de formação de brigadistas eventuais. [S.l.], 2016. Citado 2 vezes nas páginas 14 e 16.
- ROSA, R. C. da. Classes de incêndio. In: _____. Prevenção e combate a incêndio e primeiros socorros. [S.l.]: Instituto Federal do Rio Grande do Sul, 2015. p. 18–19. Citado na página 15.
- TANENBAUM, A. S. Redes de Computadores. trad. 4 ed. Rio de Janeiro: Elsevier, 2003. Citado na página 19.

Apêndices

APÊNDICE A – CÓDIGO DO CONSOLE

```

1 #include <RCSwitch.h>           // rf librarie
2
3 const int buttonPin = 2;         // pushbutton pin number
4 const int ledPin = 1;           // define led pin
5 int buttonState;               // current input pin reading
6 int lastButtonState = LOW;     // prev. input pin reading
7 int ledState = LOW;
8
9 unsigned long lastDebounceTime = 0; // last time output pin was toggled
10 unsigned long debounceDelay = 50; // debounce time
11
12 RCSwitch mySwitch = RCSwitch();
13
14 void setup() {
15   // GPIO setup
16   pinMode(buttonPin, INPUT);
17   pinMode(ledPin, OUTPUT);
18
19   // radio setup
20   mySwitch.enableTransmit(3);      // define DATA pin
21   mySwitch.setPulseLength(320);    // define pulselength
22   mySwitch.setProtocol(1);        // default is 1
23   mySwitch.setRepeatTransmit(15); // define number tries
24 }
25
26 void loop() {
27   // read the state of the switch into a local variable:
28   int reading = digitalRead(buttonPin);
29
30   // if the switch changed, due to noise or pressing:
31   if (reading != lastButtonState) {
32     // reset the debouncing timer
33     lastDebounceTime = millis();
34   }
35
36   if ((millis() - lastDebounceTime) > debounceDelay) {
37     // if the button state has changed:
38     if (reading != buttonState) {
39       buttonState = reading;
40
41       if (buttonState == HIGH) {
42         // change LED state
43         ledState = !ledState;

```

```
44     digitalWrite(ledPin, ledState);
45 }
46 // send radio message according to protocol
47 mySwitch.send("10011101");
48 }
49 }
50 // save the reading as lastButtonState
51 lastButtonState = reading;
52 }
```