

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO**

Tiago Hártur Oliveira Feu

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

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.

Orientador: Prof. Dr. André Ferreira

Vitória – ES

2022

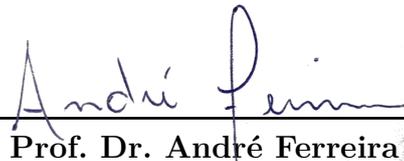
Tiago Hártur Oliveira Feu

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.

Aprovado em 25 de março de 2022.

COMISSÃO EXAMINADORA:



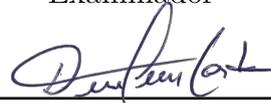
Prof. Dr. André Ferreira

Universidade Federal do Espírito Santo
Orientador



**Profa. Dra. Eliete Maria de Oliveira
Caldeira**

Universidade Federal do Espírito Santo
Examinador



Sargento B.M. Diego Turbay Costa

Corpo de Bombeiros Militar do Espírito
Santo
Examinador

Vitória – ES

2022

RESUMO

Neste trabalho é apresentado o desenvolvimento de um protótipo de simulador de incêndios para o Corpo de Bombeiros Militar do Estado do Espírito Santo (CBMES). A motivação inicial partiu de uma demanda do setor de educação de jovens e crianças do CBMES, com vistas a propiciar demonstrações *hands-on* sobre os métodos de combate a incêndios dentro de salas de aula nas escolas. O resultado do trabalho aqui descrito é um simulador completo que pode ser executado em computadores de baixo custo como uma *Raspberry Pi* e três controles remotos com a eletrônica embarcada necessária para se comunicar com uma interface virtual. O simulador desenvolvido provê uma interface muito mais real, com uma maior imersão deste público nas palestras realizadas, preservando a segurança, mesmo em locais fechados.

Palavras-chave: Combate a incêndios, eletrônica embarcada, *hands-on*, *Raspberry Pi*, simulador.

ABSTRACT

This work presents the development of a prototype of a fire simulator for the Military Fire Brigade of the State of Espírito Santo (CBMES). The initial motivation came from a demand from the youth and children education sector of CBMES, in order to provide hands-on demonstrations on firefighting methods inside classrooms. The result of the work described here is a complete simulator that can be run on low-cost computers like a Raspberry Pi and three controllers with the embedded electronics necessary to communicate with a virtual interface. The developed simulator provides a much more real interface, with a greater immersion of the students in the lectures held, preserving safety, even in closed places.

Keywords: Embedded electronics, firefighting, hands-on, Raspberry Pi, simulator.

LISTA DE FIGURAS

Figura 1 – Visão geral do protótipo	12
Figura 2 – Tetraedro do fogo	16
Figura 3 – Classes de incêndio	17
Figura 4 – Logo Pygame	18
Figura 5 – <i>Raspberry Pi</i> 4, Modelo B.	19
Figura 6 – Espectro eletromagnético	20
Figura 7 – Modulador de produto	22
Figura 8 – Modulação OOK	22
Figura 9 – Sinal OOK recebido	23
Figura 10 – Regeneração do sinal recebido	24
Figura 11 – Estrutura de usabilidade	25
Figura 12 – Diagrama geral do protótipo	28
Figura 13 – Attiny85	29
Figura 14 – Módulos RF	30
Figura 15 – Antena espiral 433 MHz	31
Figura 16 – Bateria de 9 V	31
Figura 17 – Regulador de tensão L7805CV	32
Figura 18 – Circuito do botão	33
Figura 19 – Conectores	33
Figura 20 – Simulação em 3D da placa de circuito impresso	34
Figura 21 – Placa de fenolite corroída em percloreto de ferro	35
Figura 22 – PCB com fabricação especializada	35
Figura 23 – Peças para montagem da caixa	36
Figura 24 – Extintor com circuito	37
Figura 25 – Fluxograma simplificado do <i>firmware</i>	38
Figura 26 – Protocolo de comunicação	40
Figura 27 – Tela inicial do jogo	41
Figura 28 – Fases do jogo	42
Figura 29 – Sprite de fumaça	43
Figura 30 – Rotina principal	43

LISTA DE TABELAS

Tabela 1 – Principais Rotinas	39
Tabela 2 – Custos do projeto	46

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>
MVP	<i>Minimum Viable Product</i>
NFPA	<i>National Fire Protection Association</i>
NR	Norma Regulamentadora
OOK	<i>On-off Keying</i>
PLA	Poliácido Láctico
PSK	<i>Phase Shift Keying</i>
RAM	<i>Random Access Memory</i>
RF	Radiofrequência
SD	<i>Secure Digital</i>
SRAM	<i>Static Random Access Memory</i>
SUS	<i>System Usability Scale</i>
UFES	Universidade Federal do Espírito Santo

USB *Universal Serial Bus*

Wi-Fi *Wireless Fidelity*

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	Ângulo 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	12
1.1	Apresentação	12
1.2	Justificativa	13
1.3	Objetivos	13
1.3.1	Objetivo Geral	13
1.3.2	Objetivos Específicos	13
2	EMBASAMENTO TEÓRICO	15
2.1	Combate a incêndio	15
2.1.1	Teoria do fogo	15
2.1.2	Classes de incêndio	16
2.1.3	Extintores	17
2.2	Motor de jogos	18
2.3	Plataforma	18
2.4	Comunicação via radiofrequência	20
2.4.1	Modulação ASK	21
2.4.2	Demodulação ASK(OOK)	23
2.5	Avaliação	23
2.5.1	Usabilidade	25
2.5.2	SUS - <i>System Usability Scale</i>	26
3	METODOLOGIA E ETAPAS DE DESENVOLVIMENTO	27
4	IMPLEMENTAÇÃO	28
4.1	Hardware	28
4.1.1	Microcontrolador	29
4.1.2	Módulos transmissor e receptor RF	30
4.1.3	Bateria e regulador de tensão	31
4.1.4	LED indicador, botão e conectores	32
4.1.5	Placa de circuito impresso	34
4.1.6	Caixa para acomodar os componentes	36
4.1.7	Montagem do controle	36
4.2	Firmware	38
4.2.1	Protocolo de comunicação	40
4.3	Software	40
4.3.1	Criação e utilização dos menus	41

4.3.2	<i>Sprites</i> e animações	42
5	RESULTADOS E TRABALHOS FUTUROS	45
5.1	Custos	45
5.2	Trabalhos futuros	45
	REFERÊNCIAS	48
	APÊNDICES	50
	APÊNDICE A – ESCALA DE USABILIDADE DO SISTEMA . . .	51
	APÊNDICE B – CÓDIGO DO CONSOLE	52

1 INTRODUÇÃO

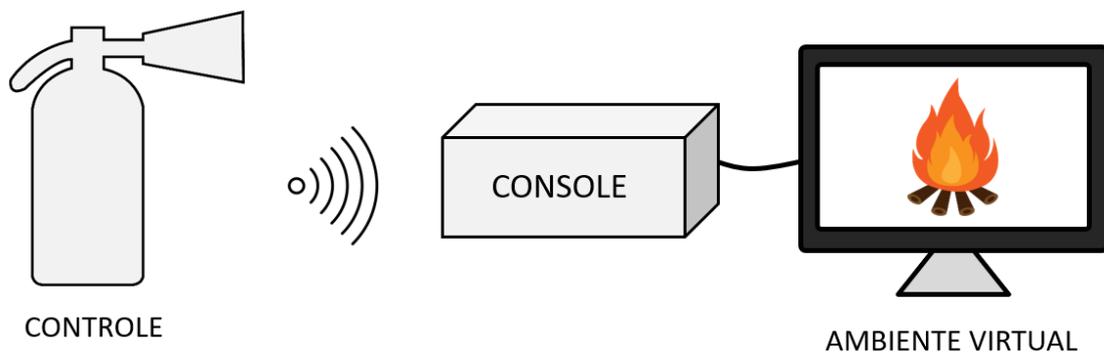
1.1 Apresentação

A motivação para o desenvolvimento desse trabalho surgiu a partir da demanda de um agente do Corpo de Bombeiros Militar do Estado do Espírito Santo (CBMES) por uma ferramenta que auxiliasse enquanto os bombeiros ministram 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. Existem, contudo, limitações impostas às demonstrações realizadas em ambientes fechados como auditórios e salas de aula, tornando inviável a liberação do agente extintor em locais como esses.

Na tentativa de contornar essas limitações, é apresentado aqui o desenvolvimento de 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.

Assim, ao longo deste trabalho é apresentada a construção de um kit educacional que simule situações de princípio de incêndio. Tal kit é composto dos controles remotos, que representam 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 comum, 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 *joysticks* incorporados à 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 a 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 a tais possibilidades, surgiu também o risco de incêndios.

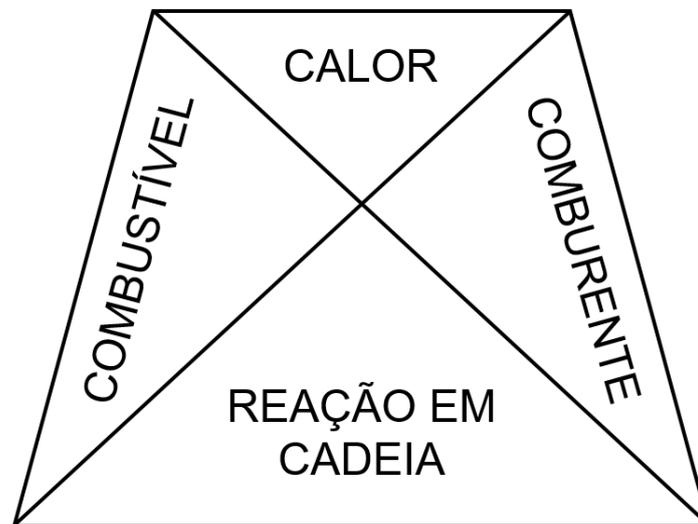
Por conta disso, de acordo com a Norma Regulamentadora 23 do Ministério do Trabalho e Previdência (NR 23), as organizações devem possuir em seus quadros pessoas capacitadas para utilizar equipamentos de proteção contra incêndios. O objetivo dessas pessoas é agir rapidamente no combate ao princípio de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

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, que queimam em sua superfície e não deixam resíduos em sua queima.
3. **Classe C:** Equipamentos elétricos energizados, como motores, transformadores, quadros de distribuição, fios, dentre outros.
4. **Classe D:** Metais combustíveis, como magnésio, zircônio e titânio.
5. **Classe K:** Óleo de cozinha, gorduras e graxa.

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

Figura 3 – Classes de incêndio



Fonte: Priorize SST, 2019.

usuário deve estar atento a classe de incêndio no momento de definir a melhor maneira 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 à qual se destinam. Os mais comuns são (LIMA; ENCARNAÇÃO, 2016):

1. **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.
2. **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 propulsor (como o dióxido de carbono) que, ao entrar em contato com as chamas, impede a continuidade da reação em cadeia.
3. **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 desenvolver algumas funcionalidades desde o início.

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 aspectos 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, utiliza-se o motor de jogos **Pygame**, cuja logo é apresentada na Figura 4. Essa *engine* é um conjunto de módulos em Python voltados para o desenvolvimento de jogos e aplicações multimídia. É gratuita e liberada sob a licença LGPL, que permite criar tanto jogos de código aberto quanto jogos comerciais. Além disso, é bastante portátil, podendo ser executada em diversas plataformas e sistemas operacionais.

Figura 4 – Logo Pygame



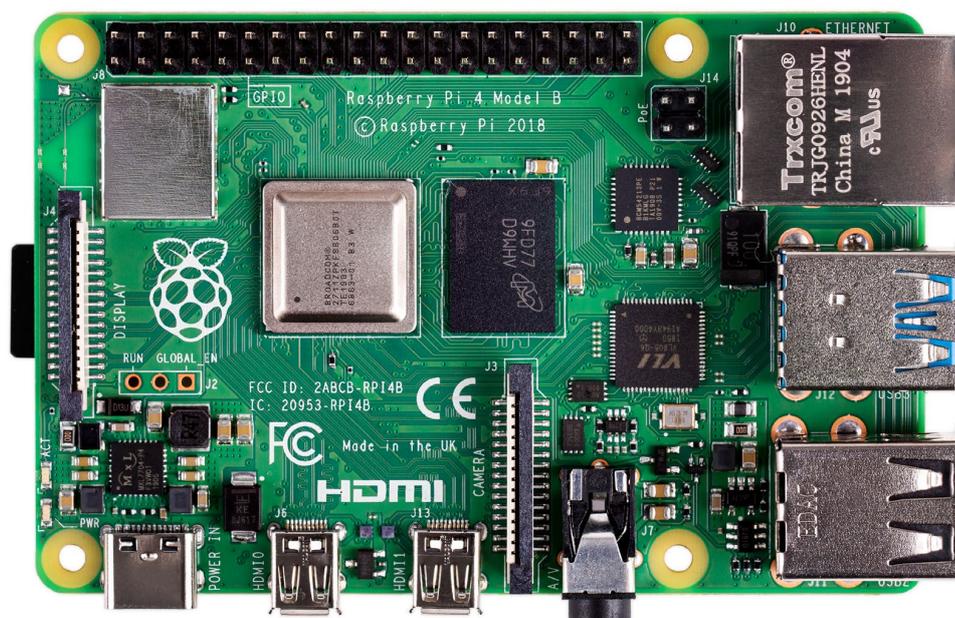
Fonte: pygame.org, 2021.

2.3 Plataforma

Para gerar o ambiente virtual e receber os comandos do jogador é utilizado um micro-computador 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: $85 \times 56 \times 17$ mm.

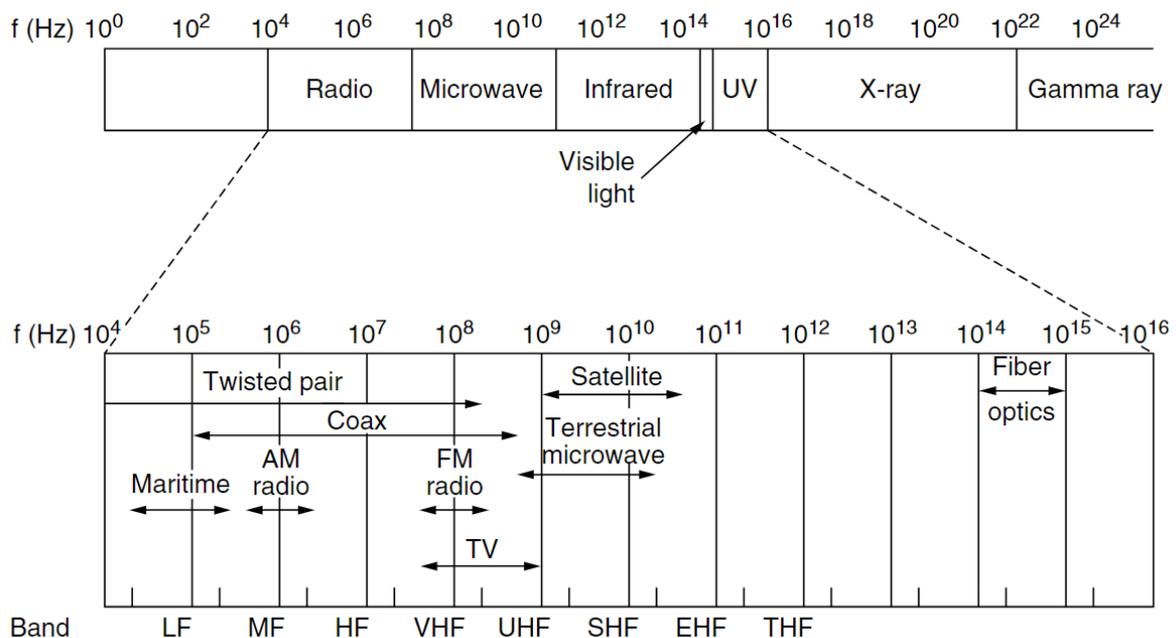
A interface do microcomputador com o usuário é possível através de alguns sistemas operacionais baseados em Linux. O mais utilizado deles é o **Raspbian**, que é um sistema operacional gratuito baseado nos sistemas Debian e otimizado para *RaspBerry Pi*. Esse é o sistema escolhido para ser utilizado nesse trabalho.

2.4 Comunicação via radiofrequência

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 adiciona-se informação a uma onda eletromagnética é chamado de modulação, e a recuperação dessa informação é chamada demodulação. A modulação consiste em 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 ângulo 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 corretamente, 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 considera-se apenas a modulação ASK, especialmente voltada para sinais digitais.

2.4.1 Modulação ASK

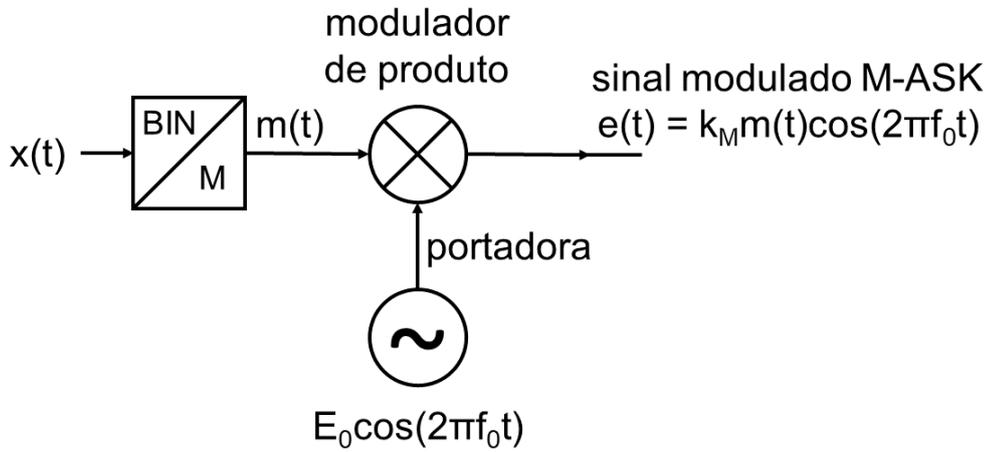
A modulação de amplitude pode ser obtida em um modulador de produto como o observado na Figura 7. Num modulador como esse um sinal digital de informação $x(t)$ é convertido em um sinal multinível $m(t)$ onde cada combinação de N bits é associada a um valor M de tensão (CARVALHO, 2009).

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 se tem nível lógico alto (bit = 1) e suprimir quando se tem nível lógico baixo (bit = 0),

Figura 7 – Modulador de produto

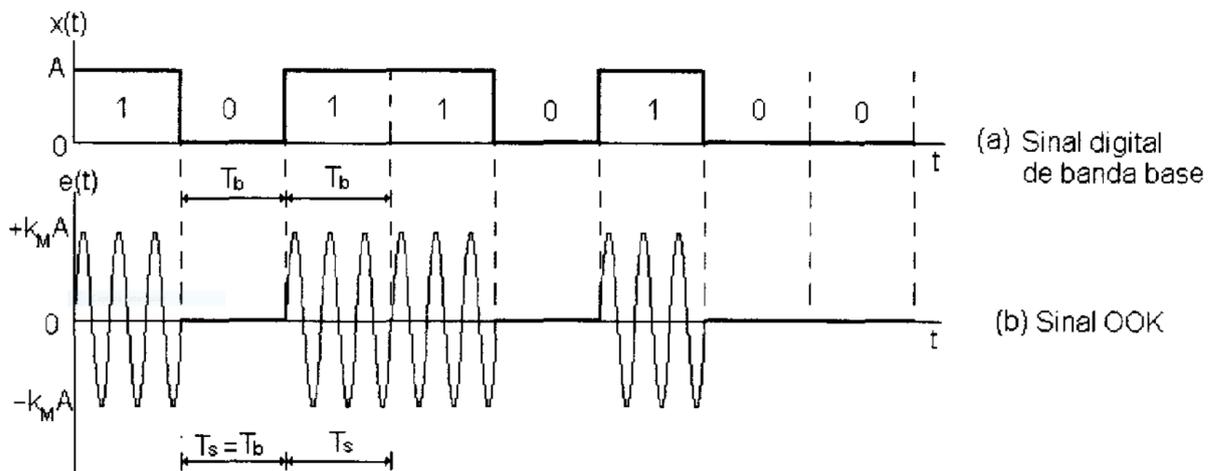


Fonte: Adaptado de Carvalho, 2008.

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

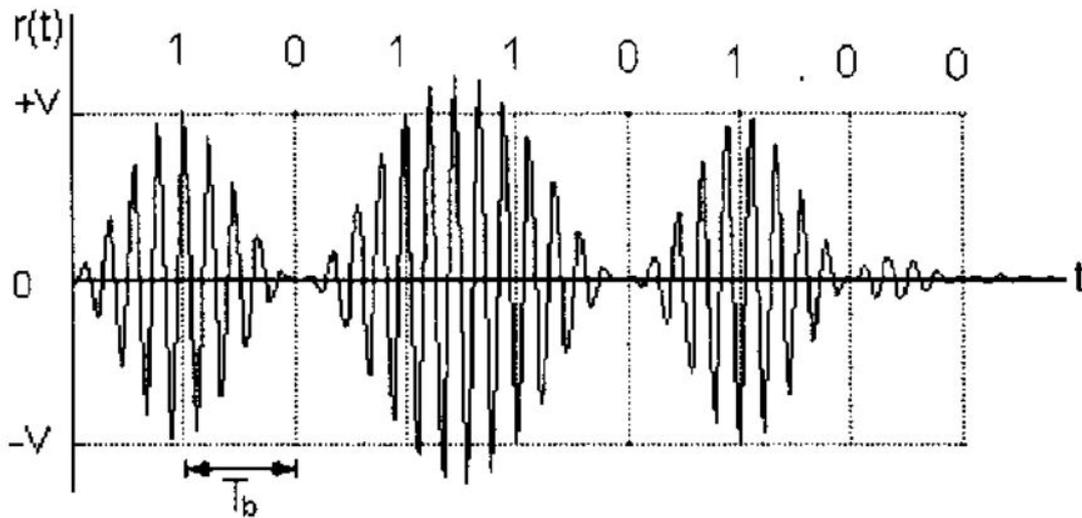


Fonte: Carvalho, 2008.

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.

Figura 9 – Sinal OOK recebido



Fonte: Carvalho, 2008.

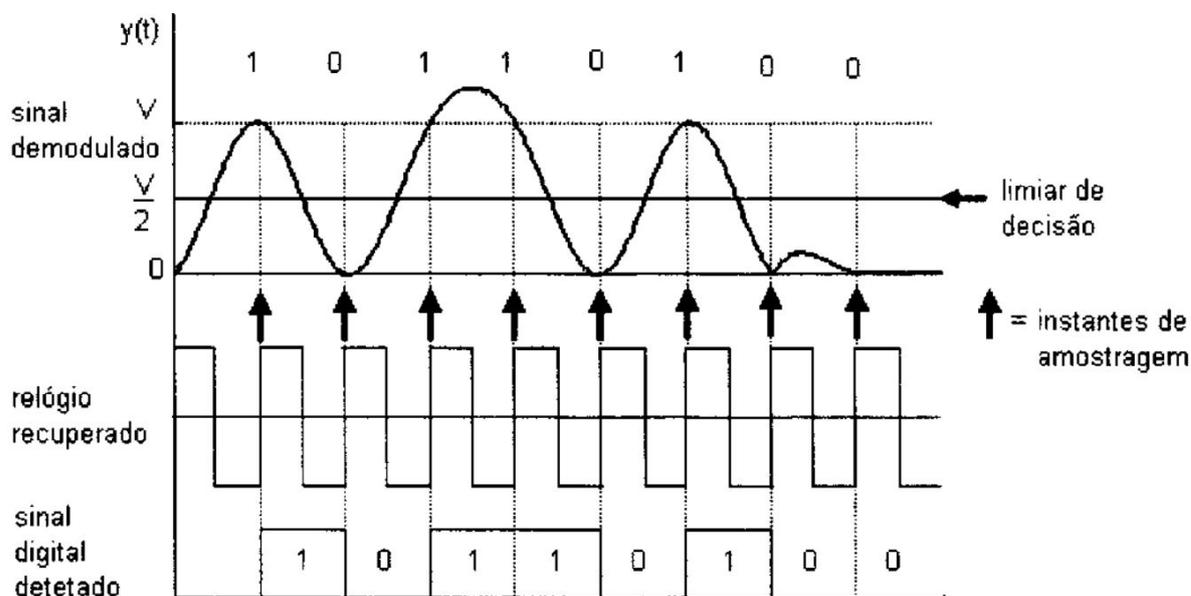
O processo de demodulação tem início com um detector de envoltória que transforma o sinal $r(t)$ em uma envoltória $y(t)$, como a da Figura 10. Em seguida essa envoltória é amostrada segundo o tempo de 1 bit do sinal de informação $x(t)$. Caso a amostra ultrapasse um limiar de decisão definido, considera-se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

2.5 Avaliação

O questionário é um instrumento desenvolvido cientificamente, composto de um conjunto de perguntas ordenadas de acordo com um critério pré-determinado (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,

Figura 10 – Regeneração do sinal recebido



Fonte: Carvalho, 2008.

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 para 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 conseqüente 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) o questionário 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).

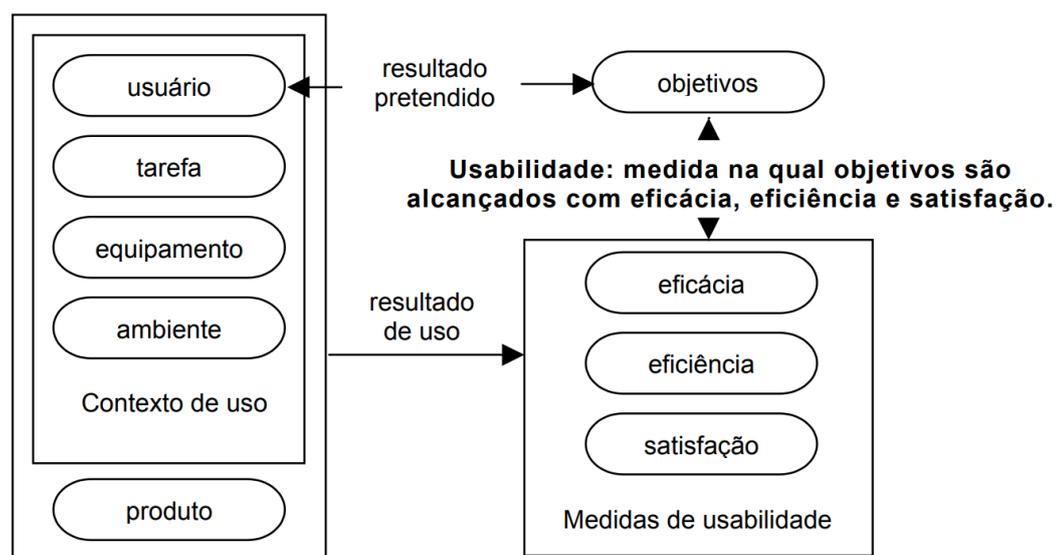
O último passo é definir a quantidade necessária de questionários respondidos para que a resposta obtida seja estatisticamente relevante. Segundo o censo escolar de 2020, o Espírito Santo tem cerca de 800 mil alunos matriculados em escolas de educação básica (INEP, 2021). Considerando esse como o público alvo da pesquisa, para obter um grau de confiança no resultado de 95% com um erro tolerável de até 10% serão necessários cerca de 100 formulários respondidos (ISRAEL, 1992).

2.5.1 Usabilidade

Dentro dos requisitos citados anteriormente, o atributo chave a ser observado é o da usabilidade. Segundo a NBR ISO 9241-11, a usabilidade é uma "Medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso" (ABNT, 2002). Ou seja, é necessário observar o produto desenvolvido dentro do contexto no qual ele será utilizado.

O processo de medição de usabilidade, segundo a norma, é dividido em três principais pontos: (i) eficácia: que é a acurácia com que os usuários completam as atividades; (ii) eficiência: recursos gastos em relação a acurácia dos usuários; (iii) satisfação: o conforto durante a utilização do produto. A interação desses quesitos com o produto pode ser observada no diagrama da Figura 11.

Figura 11 – Estrutura de usabilidade



Fonte: ABNT, 2002.

2.5.2 SUS - *System Usability Scale*

Pensando em um formulário que atenda todos esses requerimentos, foi escolhido o *System Usability Scale* (SUS). Esse é um sistema de avaliação contendo 10 afirmações simples onde o usuário deve selecionar uma opção numa escala de 1 a 5 dizendo o quanto ele concorda ou discorda com a afirmação (BROOKE, 1996).

As afirmações escolhidas para esse modelo de questionário são afirmações que, ao serem testadas, gerarão uma significativa variedade de respostas. Além disso, também são afirmações que avaliam vários aspectos da usabilidade, como: necessidade de suporte, treinamento e complexidade. Por se tratar de um questionário escrito originalmente em inglês, foi necessária uma adaptação ao português para poder ser aplicado (MARTINS et al., 2015).

Após o preenchimento do formulário, é possível calcular uma pontuação medindo a usabilidade geral do sistema em estudo. Essa pontuação é calculada somando a contribuição de cada um dos itens do questionário, que pode variar de 0 a 4. Para os itens 1, 3, 5, 7 e 9 a nota é a posição na escala menos 1. Já para os itens 2, 4, 6, 8 e 10 a nota é 5 menos a posição na escala. Por fim, essa soma de notas é multiplicada por 2,5 a fim de obter uma pontuação variando de 0 a 100, sendo a média de notas próxima a 68 pontos (SAURO; LEWIS, 2016).

O exemplo do formulário aplicado está presente no APÊNDICE A desse documento.

3 METODOLOGIA E ETAPAS DE DESENVOLVIMENTO

O desenvolvimento do simulador foi realizado de acordo com a seguinte sequência de etapas e atividades realizadas:

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. Dimensionamento das adequações necessárias ao extintor para ser utilizado como controle.
4. Definição do protocolo de comunicação a ser utilizado entre controle e console.
5. Definição dos diferentes cenários de risco a serem simulados, de acordo com as necessidades do Corpo de Bombeiros.

Etapa3. Desenvolvimento do protótipo

6. Realização das adequações físicas nos extintores.
7. Desenvolvimento do jogo no ambiente virtual.

Etapa4. Teste do protótipo

8. Realização de testes práticos e avaliação dos resultados.

Etapa5. Documentação

9. Documentação das etapas de desenvolvimento e redação da 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 12.
2. *Firmware*, que controla a eletrônica embarcada do bloco 2 da Figura 12.
3. *Software*, observado no bloco 4 da Figura 12 sendo executado na *Raspberry Pi* (bloco 3).

Figura 12 – Diagrama geral do protótipo



Fonte: Produção do próprio autor.

O sistema é constituído pelo *joystick*, que é a junção do extintor de incêndios (bloco 1 da Figura 12) 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 à *Raspberry Pi* (bloco 3 da Figura 12) a fim de controlar o simulador sendo executado.

4.1 Hardware

A eletrônica do projeto contém os componentes listados a seguir:

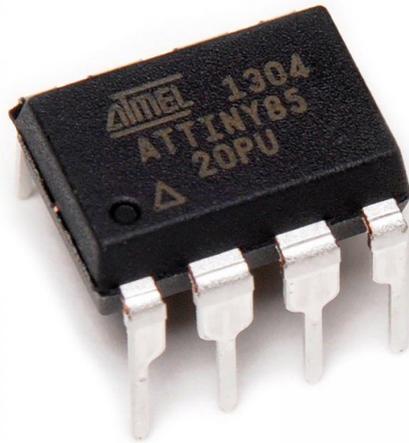
1. Microcontrolador;
2. Módulos transmissor e receptor RF;
3. Bateria e regulador de tensão;
4. LED indicador, botão e conectores.

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 13.

Figura 13 – Attiny85



Fonte: Bau da eletrônica, 2021.

Algumas especificações do Attiny85 segundo o *datasheet* do fabricante (ATMEL, 2013):

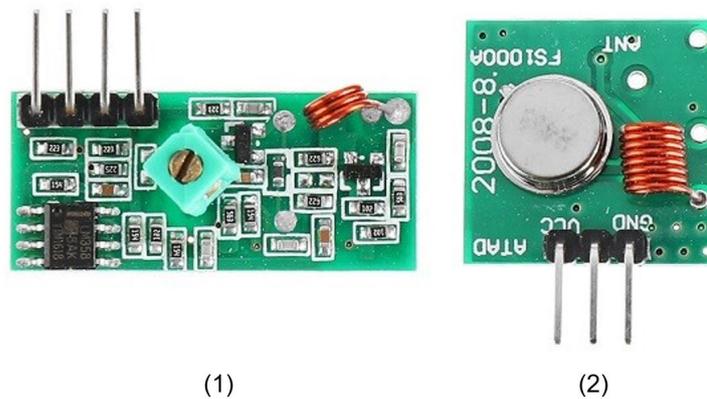
- Fabricado pela Atmel
- 8 pinos DIP, sendo 6 deles GPIO
- 512 Bytes de memória SRAM
- Frequência máxima de 20 MHz
- Dimensões de 9,3 x 6,3 x 3,3 mm (comprimento x largura x altura)

4.1.2 Módulos transmissor e receptor RF

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

Para a comunicação do Attiny85 com a *Raspberry* são utilizados dois módulos de radiofrequência de modulação ASK. O módulo 1 da Figura 14 é o receptor (MX-05V), que fica conectado na *Raspberry Pi* através dos pinos GPIO. Já o módulo 2 (MX-FS-03V) da Figura 14 é o receptor que fica junto do restante da eletrônica embarcada no *joystick*.

Figura 14 – Módulos RF



Fonte: Adaptado de Filipeflop, 2021.

Para melhorar a comunicação entre os dois módulos, foi conectada uma antena helicoidal metálica de tamanho apropriado à frequência de operação ao módulo transmissor, aumentando o alcance da comunicação. Essa antena pode ser vista na Figura 15.

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 -105 dB e (iii) dimensões de 30 x 14 x 7 mm.

Figura 15 – Antena espiral 433 MHz



Fonte: Filipeflop, 2021.

4.1.3 Bateria e regulador de tensão

Para alimentação do circuito foi utilizada uma bateria recarregável de 9 V e 250 mAh que pode ser observada na Figura 16. O principal motivo para a escolha dessa bateria é a facilidade com que ela e seu carregador podem ser encontrados no mercado. Além disso, consegue armazenar carga suficiente para a aplicação, visto que em um ensaio realizado com o sistema em funcionamento e o LED indicador aceso, a bateria manteve o sistema energizado por cerca de 7 horas.

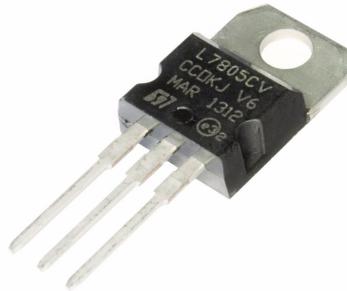
Figura 16 – Bateria de 9 V



Fonte: Grupo Elgin, 2021.

Apesar dos pontos positivos citados da bateria, sua tensão de funcionamento é incompatível com os demais componentes que compõem o *joystick*. Por isso, é necessária 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 17.

Figura 17 – Regulador de tensão L7805CV



Fonte: Achei Componentes Eletrônicos, 2021.

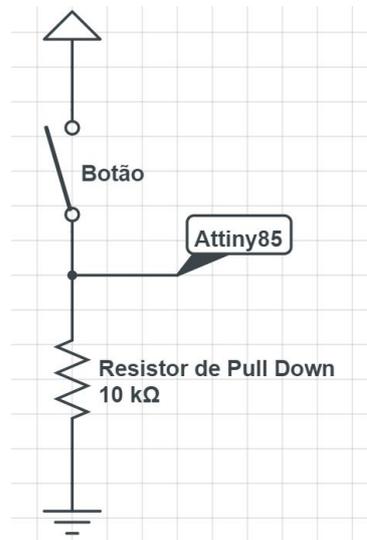
Algumas características importantes desse componente são:

- 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
- Proteção contra curto-circuito

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 a um pino de entrada do Attiny85, como pode ser observado na Figura 18. 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.

Figura 18 – Circuito do botão



Fonte: Produção do próprio autor.

Foi adicionado um LED indicador vermelho de 3 mm de diâmetro e 20 mA em série com um resistor de 330 Ω 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.

Para realizar a conexão da bateria à PCB foi utilizado um *clip* conector como o da Figura 19.1. Já a conexão do botão foi realizada com um conector JST de duas vias como o da Figura 19.2.

Figura 19 – Conectores



(1) *Clip* conector de bateria 9 V

(2) Conector JST polarizado de 2 vias

Fonte: Produção do próprio autor.

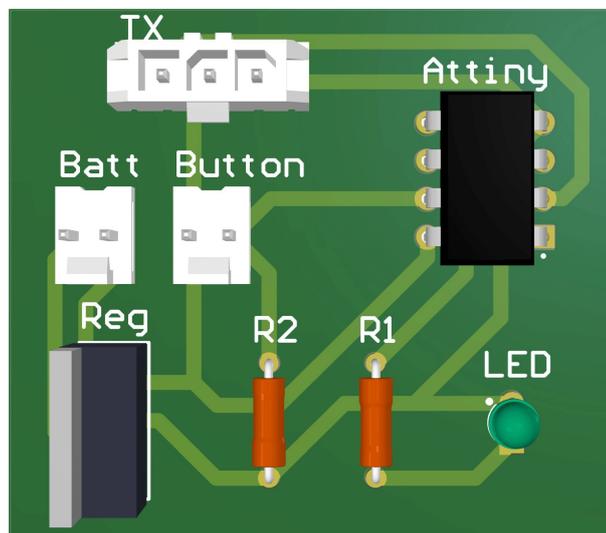
4.1.5 Placa de circuito impresso

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

1. Desenvolvimento do circuito em 2D e 3D, sendo o último como o modelo da Figura 20;
2. Montagem em *protoboard* e testes iniciais;
3. Confeção em placa de fenolite, como mostrado na Figura 21;
4. Confeção especializada, como a da Figura 22.

A simulação de um modelo em três dimensões da PCB é um passo importante para estimar o tamanho dos componentes e como serão dispostos, além de uma estimativa do tamanho final da PCB. Dessa forma, é possível projetar a PCB de acordo com os requisitos de projeto.

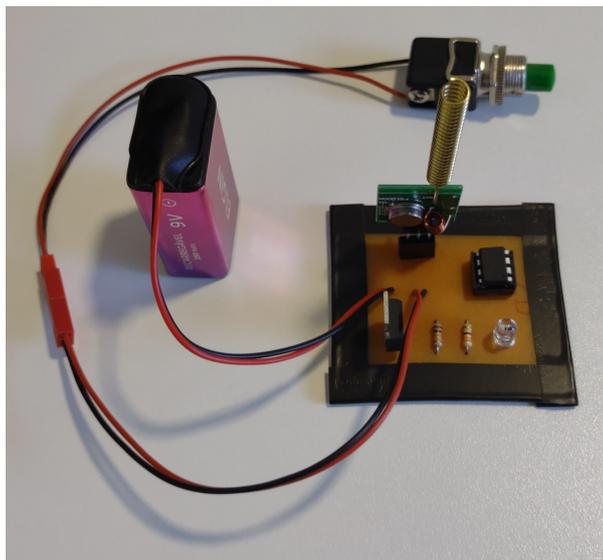
Figura 20 – Simulação em 3D da placa de circuito impresso



Fonte: Produção do próprio autor.

Após a modelagem do circuito e a validação com testes em *protoboard*, foi confeccionada uma versão de teste (Figura 21), 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*.

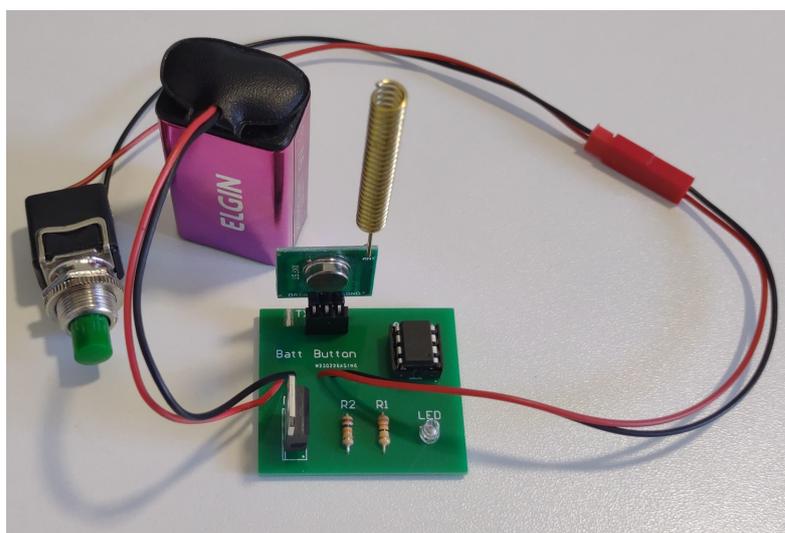
Figura 21 – Placa de fenolite corroída em percloroeto de ferro



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 22, 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 e identificação dos componentes na camada sem cobre (para o caso de PCBs com apenas uma camada). Esses processos reduzem ruídos e aumentam a vida útil da PCB.

Figura 22 – PCB com fabricação especializada



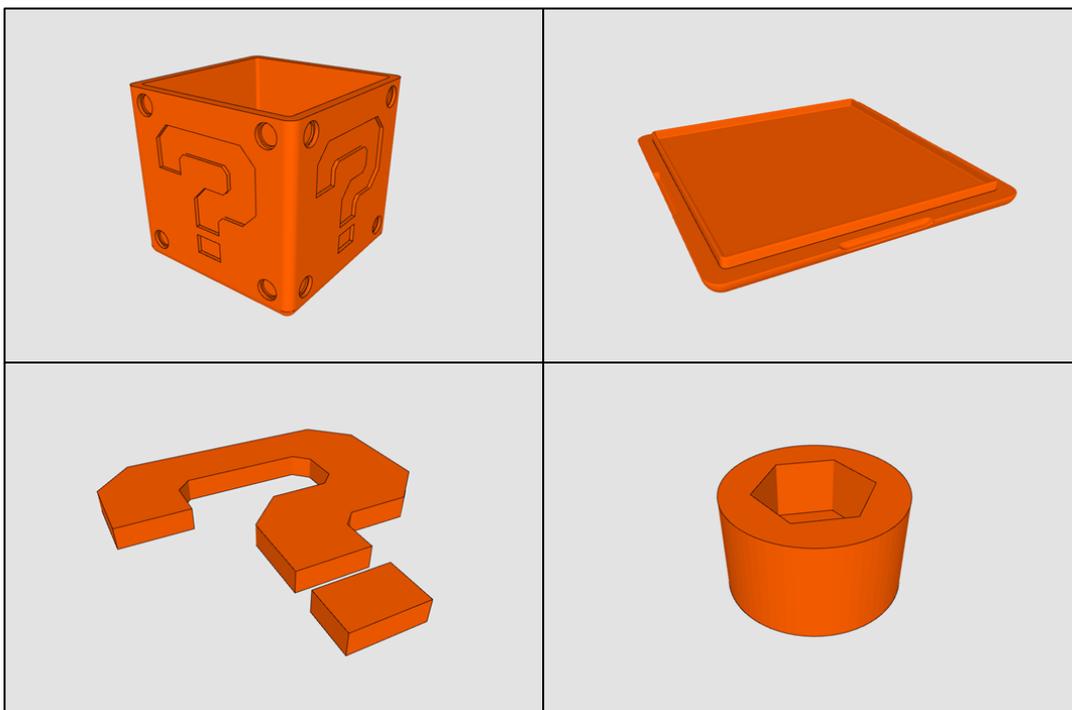
Fonte: Produção do próprio autor.

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 um material termoplástico biodegradável chamado PLA (poliácido láctico) 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 quatro peças diferentes: a caixa, a tampa e dois ornamentos que devem ser impressos em cores diferentes para destaque. Após a impressão, há uma segunda etapa em que os ornamentos são colados na lateral da caixa. Esses modelos 3D estão presentes na Figura 23.

Figura 23 – 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 24. Nessa montagem tem-se o botão fixado na alça de transporte através de um furo localizado sob o botão. Esse botão está

conectado à 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 24, 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 do extintor de incêndio como: peso, formato, rótulos informativos e a etiqueta de validade.

Figura 24 – Extintor com circuito



Fonte: Produção do próprio autor.

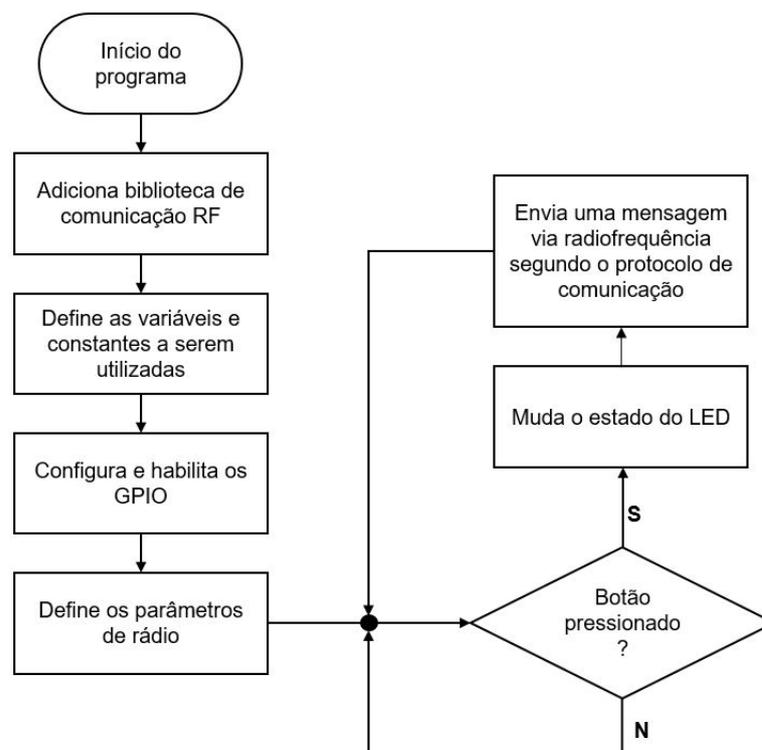
4.2 Firmware

O *firmware*, que pode ser observado no APÊNDICE B 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. **Verificação do botão:** nessa seção ocorre a verificação 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 25

Figura 25 – 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 início 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

Os dados são enviados seguindo o protocolo de comunicação da Figura 26.

Figura 26 – Protocolo de comunicação



Fonte: Produção do próprio autor.

Nesse protocolo são enviados 8 bits, sendo os quatro primeiros um preâmbulo 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.

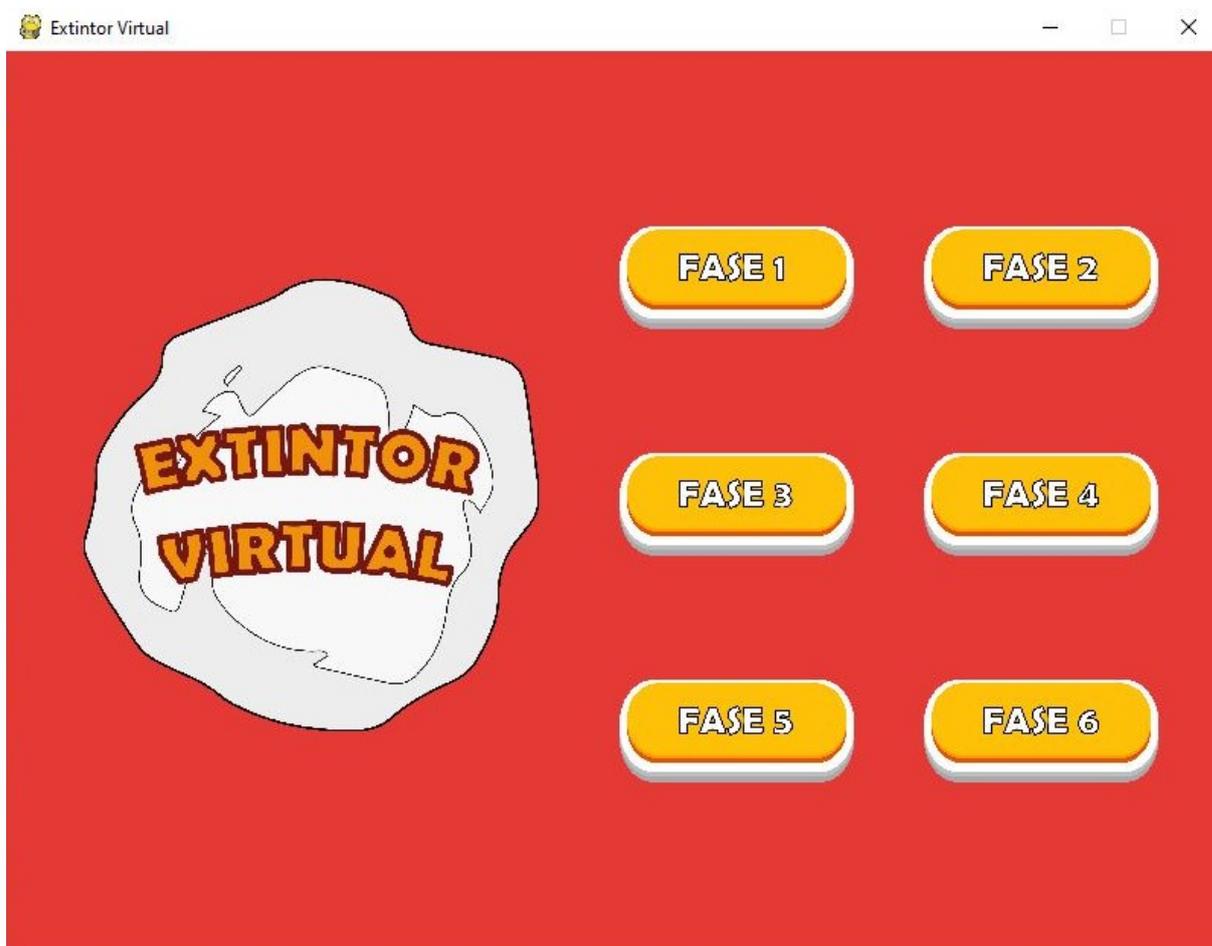
4.3 Software

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 à esquerda (Extintor Virtual) e à direita se localizam os diferentes cenários a serem escolhidos pelo usuário, como pode ser observado na Figura 27. Essa escolha deve ser feita através da utilização de um mouse conectado à *Raspberry Pi*.

Figura 27 – Tela inicial do jogo



Fonte: Produção do próprio autor.

Cada um desses cenários ilustra uma situação de princípio de incêndio com materiais diferentes, gerando distintas 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 28, tem-se 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 28 – Fases do jogo



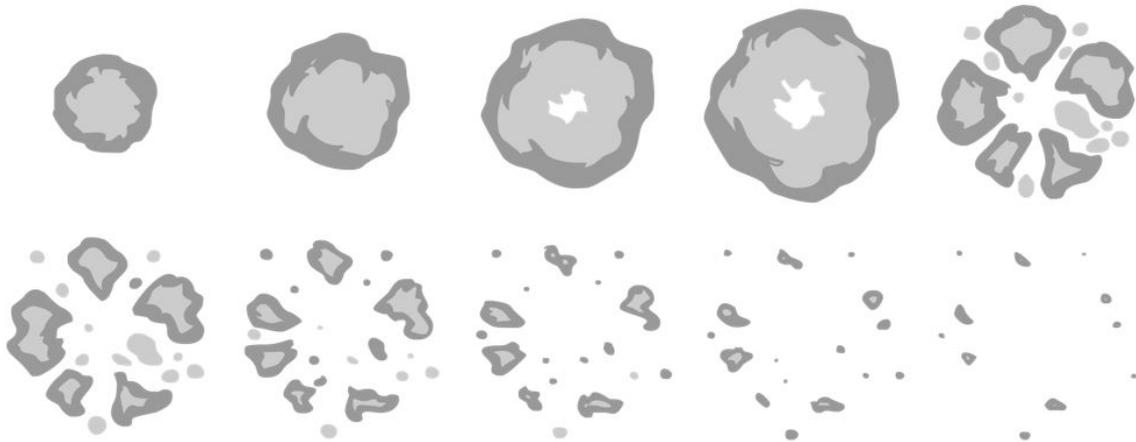
Fonte: Produção do próprio autor.

4.3.2 Sprites e animações

Na computação gráfica, um *sprite* é uma imagem ou animação bidimensional sobreposta em uma cena. Esses são os elementos com os quais o usuário vai interagir durante o jogo e representam os personagens e materiais presentes. Nesse caso, 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 29, que são sobrepostos na tela em sequência a uma certa velocidade e assim criam a animação de fumaça.

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 verifica se há novas mensagens do controle e, caso existam, realiza as modificações necessárias. A rotina principal tem estrutura semelhante ao código mostrado na Figura 30.

Figura 29 – Sprite de fumaça



Fonte: Produção do próprio autor.

Figura 30 – Rotina principal

```

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()

```

Fonte: Produção do próprio autor.

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, o sistema verifica se há uma nova mensagem vinda 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 acionado.

5 RESULTADOS E TRABALHOS FUTUROS

Finalizado o desenvolvimento, tem-se como resultado um MVP (*Minimum Viable Product*) a ser validado no dia a dia das demonstrações do corpo de bombeiros. Esse produto inicial consiste em duas partes principais: (i) o simulador, que é a soma do *software* do jogo, da *Raspberry Pi* e do receptor de radiofrequência e (ii) o *joystick* que é a junção dos demais componentes eletrônicos acomodados dentro da caixa plástica.

A intenção de fazer um produto inicialmente mais simples é não empregar tempo excessivo em otimizações antes de deixar que os usuários finais utilizem o sistema e indiquem suas sugestões de melhoria.

5.1 Custos

Por se tratar de um setor com orçamento limitado, uma das principais preocupações ao longo do desenvolvimento do protótipo foi o custo. O pensamento foi justamente a escolha de dispositivos mais comuns e baratos, para que o produto pudesse ser facilmente replicado por pessoas interessadas. Considerando três controles e um console, tem-se os seguintes custos descritos na Tabela 2.

5.2 Trabalhos futuros

Após as primeiras sessões de teste, foram identificados alguns pontos que podem ser melhorados. Dessas melhorias, as principais estão relacionadas à fidedignidade do simulador. Por exemplo: alguns extintores devem ter o esguicho apontado para a base do fogo, outros como o de pó químico, devem ser utilizados através de vários jatos curtos. Também devem ser observados durante a utilização outros fatores como a direção do vento, uma vez que o extintor deve sempre ser utilizado a favor do vento. Além disso, deve-se testar o gatilho do extintor antes de dirigir-se ao local do sinistro (LIMA; ENCARNAÇÃO, 2016).

Considerando esses cuidados no manuseio dos equipamentos, foram propostas algumas adições a serem implementadas:

- Adição de um sensor que indique a direção para a qual o esguicho está sendo apontado. Uma possibilidade para resolver esse problema é um sensor de inclinação, a fim de medir se o esguicho está na inclinação correta. Uma segunda possibilidade é a

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
<i>Raspberry Pi</i> + fonte	R\$ 949,90	1 Unidade	R\$ 949,90
Total			R\$ 1330,60

Fonte: Produção do próprio autor.

utilização de um conjunto de emissor e receptor infravermelho para detecção do nível de alinhamento entre o extintor e a base do fogo (virtual). Esta informação poderia ser utilizada para modular a altura da chama, ou seja, quanto melhor o alinhamento, mais rápido o fogo se extingue.

- Adição de uma verificação se o usuário realmente testou o extintor antes de se dirigir para o local do sinistro.
- Adição de novas informações no simulador a respeito do ambiente do incêndio, como a direção do vento, para que os usuários possam se atentar a isso durante a utilização.
- A proximidade da chama influencia na velocidade de extinção do fogo. Dessa forma, é interessante uma mecânica de jogo que controle a velocidade de extinção do fogo baseado na distância do controle ao simulador. Através da mecânica de controle de distância, também seria possível emitir alertas de perigo caso o usuário esteja próximo demais da chama.

Além das possibilidades de melhorias já descritas, cabe mencionar a importância da avaliação da usabilidade do sistema desenvolvido através da métrica SUS em cenário prático. Neste sentido, o público alvo e os instrutores poderão apreciar o simulador com a oportunidade de sugerir aperfeiçoamentos sob o ponto de vista do usuário.

REFERÊNCIAS

- ABNT, A. B. D. N. T. NBR 9241-11: Requisitos ergonômicos para trabalho de escritórios com computadores parte 11 – orientações sobre usabilidade. Rio de Janeiro, 2002. Citado na página 25.
- ATMEL. Datasheet: Atmel 8-bit AVR Microcontroller with 2/4/8K. San Jose, USA, 2013. Citado na página 29.
- BROOKE, J. Sus: a “quick and dirty” usability. Usability evaluation in industry, v. 189, n. 3, 1996. Citado na página 26.
- CARVALHO, R. M. Comunicações analógicas e digitais. [S.l.]: LTC, 2009. Citado na página 21.
- 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 24.
- 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 13.
- 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 13.
- INEP. Censo da educação básica 2020. Brasília, 2021. Citado na página 25.
- ISRAEL, G. D. Determining sample size. University of Florida Cooperative Extension Service, Institute of Food and . . . , 1992. Citado na página 25.
- 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 23 e 24.
- LEE, V. S. Involved students in "hands-on" project such as research, case studies, or real-life activities. IDEA, 2012. Citado na página 13.
- LIMA, S. de T. C.; ENCARNAÇÃO, L. G. Curso de formação de brigadistas eventuais. [S.l.], 2016. Citado 3 vezes nas páginas 15, 17 e 45.
- MARTINS, A. I. et al. European portuguese validation of the system usability scale (sus). Procedia Computer Science, Elsevier, v. 67, p. 293–300, 2015. Citado na página 26.
- 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 16.
- SAURO, J.; LEWIS, J. R. Quantifying the user experience: Practical statistics for user research. [S.l.]: Morgan Kaufmann, 2016. Citado na página 26.

TANENBAUM, A. S. Redes de Computadores. trad. 4 ed. Rio de Janeiro: Elsevier, 2003.
Citado na página 20.

Apêndices

APÊNDICE A – ESCALA DE USABILIDADE DO SISTEMA

NOME (OPCIONAL): _____ IDADE: _____

SUS

INSTRUÇÕES: Para cada afirmação abaixo selecione a caixa (apenas uma por linha) que melhor descreve sua impressão a respeito do jogo.

	Discordo totalmente			Concordo totalmente	
	1	2	3	4	5
Acho que gostaria de utilizar este produto com frequência.	<input type="checkbox"/>				
Considerarei o produto mais complexo do que necessário.	<input type="checkbox"/>				
Achei o produto fácil de utilizar.	<input type="checkbox"/>				
Acho que necessitaria de ajuda de um técnico para conseguir utilizar este produto.	<input type="checkbox"/>				
Considerarei que as várias funcionalidades deste produto estavam bem integradas.	<input type="checkbox"/>				
Achei que este produto tinha muitas inconsistências	<input type="checkbox"/>				
Suponho que a maioria das pessoas aprenderia a utilizar rapidamente este produto.	<input type="checkbox"/>				
Considerarei o produto muito complicado de utilizar.	<input type="checkbox"/>				
Senti-me muito confiante a utilizar este produto.	<input type="checkbox"/>				
Tive que aprender muito antes de conseguir lidar com este produto.	<input type="checkbox"/>				

Fonte: Produção do próprio autor.

APÊNDICE B – 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 }
```