PROGRAMA DE PÓS-GRADUAÇÃO EM
ENSINO DE CIÊNCIAS E MATEMÁTICA

MANUAL PARA A CONSTRUÇÃO DO PROTÓTIPO DO ORGANISMO HUMANO COM CONTROLE FUZZY

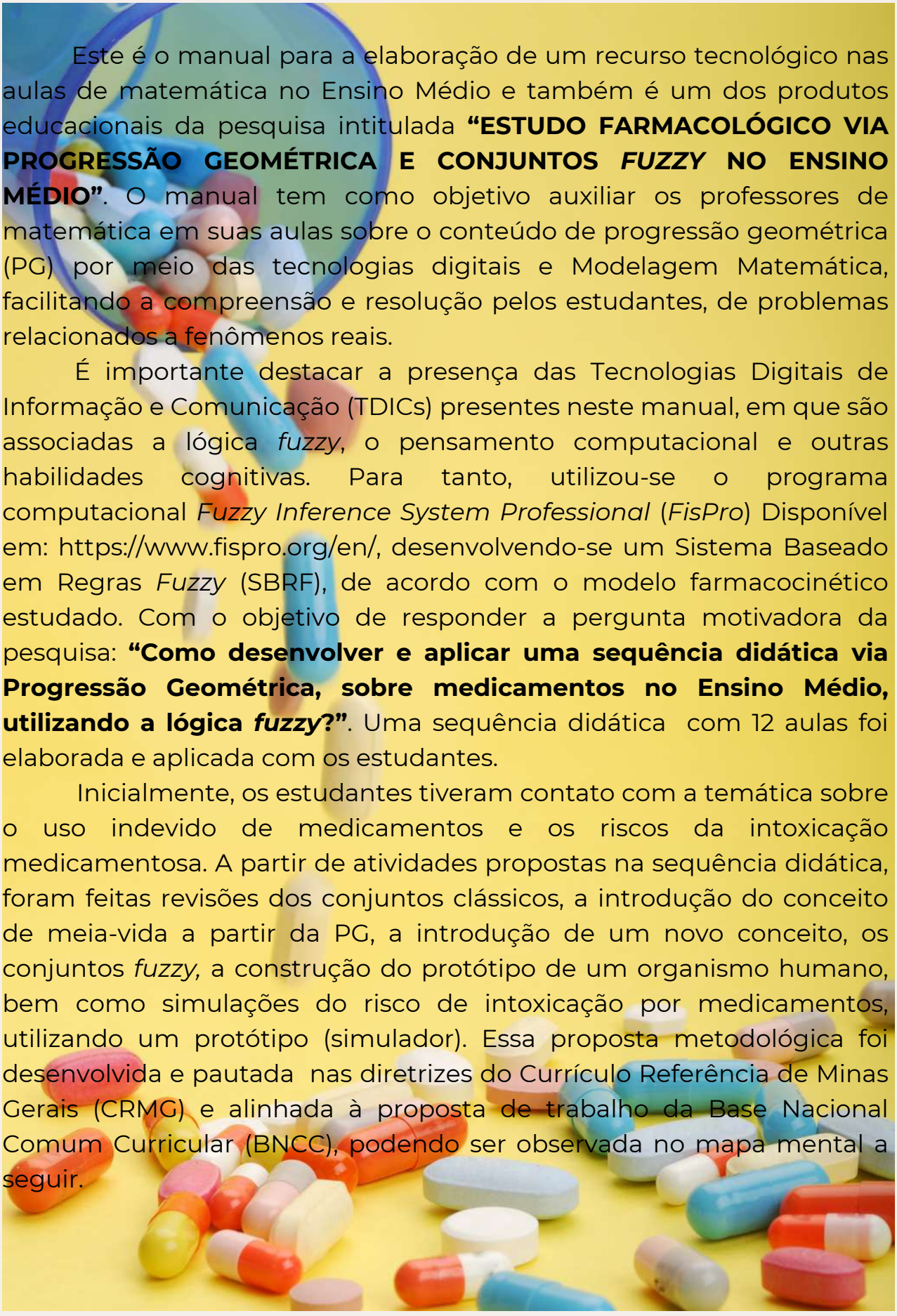
Graciela Nunes da Silva
Rosana Sueli da Motta Jafelice

Sumário

1	APRESENTAÇÃO.....4 .
1.1	MAPA MENTAL.....5
2	COMPOSIÇÃO DOS ELEMENTOS DO PROTÓTIPO.....6
2.1	COMPOSIÇÃO DOS ELEMENTOS DO PROTÓTIPO.....6
2.2	PROGRAMA *FISPRO*.....8
2.3	APLICATIVO *BLYNK*.....9
3.0	CONSTRUÇÃO DO PROTÓTIPO.....10
3.1	MATERIAIS UTILIZADOS PARA A CONSTRUÇÃO DO PROTÓTIPO.....10
3.2	MONTAGEM DO PROTÓTIPO.....13
3.3	SISTEMA BASEADO EM REGRAS FUZZY (SBRF) DO MODELO FARMACOCINÉTICO.....16
3.4	ESTRUTURA DA PARTE ELETRÔNICA DO MODELO FARMACOCINÉTICO.....*25*
3.4.1	CONECTANDO O ARDUÍNO IDE COM ESP 8266 E *BLYNK*.....26
3.4.2	SENSOR INTERNO DO PROTÓTIPO.....28
3.4.3	ALICATIVO *BLYNK* 30

4.0	CONSIDERAÇÕES FINAIS.....32
5.0	REFERÊNCIAS.....33
	APÊNDICE.....34

1. Apresentação



Este é o manual para a elaboração de um recurso tecnológico nas aulas de matemática no Ensino Médio e também é um dos produtos educacionais da pesquisa intitulada **“ESTUDO FARMACOLÓGICO VIA PROGRESSÃO GEOMÉTRICA E CONJUNTOS *FUZZY* NO ENSINO MÉDIO”**. O manual tem como objetivo auxiliar os professores de matemática em suas aulas sobre o conteúdo de progressão geométrica (PG) por meio das tecnologias digitais e Modelagem Matemática, facilitando a compreensão e resolução pelos estudantes, de problemas relacionados a fenômenos reais.

É importante destacar a presença das Tecnologias Digitais de Informação e Comunicação (TDICs) presentes neste manual, em que são associadas a lógica *fuzzy*, o pensamento computacional e outras habilidades cognitivas. Para tanto, utilizou-se o programa computacional *Fuzzy Inference System Professional* (*FisPro*) Disponível em: https://www.fispro.org/en/, desenvolvendo-se um Sistema Baseado em Regras *Fuzzy* (SBRF), de acordo com o modelo farmacocinético estudado. Com o objetivo de responder a pergunta motivadora da pesquisa: **“Como desenvolver e aplicar uma sequência didática via Progressão Geométrica, sobre medicamentos no Ensino Médio, utilizando a lógica *fuzzy*?”**. Uma sequência didática com 12 aulas foi elaborada e aplicada com os estudantes.

Inicialmente, os estudantes tiveram contato com a temática sobre o uso indevido de medicamentos e os riscos da intoxicação medicamentosa. A partir de atividades propostas na sequência didática, foram feitas revisões dos conjuntos clássicos, a introdução do conceito de meia-vida a partir da PG, a introdução de um novo conceito, os conjuntos *fuzzy,* a construção do protótipo de um organismo humano, bem como simulações do risco de intoxicação por medicamentos, utilizando um protótipo (simulador). Essa proposta metodológica foi desenvolvida e pautada nas diretrizes do Currículo Referência de Minas Gerais (CRMG) e alinhada à proposta de trabalho da Base Nacional Comum Curricular (BNCC), podendo ser observada no mapa mental a seguir.

1.1 Mapa Mental

1 Análise

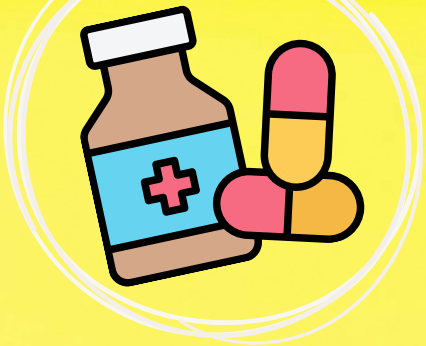
Escolha do tema contemporâneo transversal (TCT) saúde , possibilitando uma proposta metodológica interdisciplinar e inovadora.

2 Estratégia

Utilização do Pensamento Computacional aliado a lógica fuzzy, utilizando a modelagem matemática na resolução de problemas.

3 Objetivos

Abordagem do conceito da meia-vida dos medicamentos dentro do conteúdo da PG; conscientização do uso de medicamentos e de suas ações; modelagem do risco de intoxicação medicamentosa, simulação dos riscos com a utilização do protótipo..



4 Ação

Construção de um modelo farmacocinético, capaz de facilitar a compreensão e resolução pelos estudantes, de um problema relacionado a um fenômeno real "intoxicação medicamentosa".

5 Resolução

Encontro das respostas do problema a partir do modelo farmacocinético construído., utilizando o protótipo para simulações.

6 Revisão

Revisão da aprendizagem dos estudantes, a partir de uma ferramenta tecnológica, um jogo com perguntas e repostas em formato quiz , contemplando todo o assunto abordado.

2. Composição dos elementos do Protótipo

2.1 O Simulador do Organismo Humano

O Simulador do Organismo Humano (Protótipo), foi criado para permitir que os estudantes interajam com o modelo farmacocinético, aplicando o conhecimento adquirido no decorrer das atividades didáticas (Figuras 1 e 2). O controle do simulador é feito através do Arduíno, uma plataforma programável que pode ser configurada para várias funções, tais como acender e apagar luzes de LED ou ligar e desligar dispositivos eletrônicos. Para programar este protótipo, utiliza-se a linguagem C++ na placa eletrônica ESP8266, através da biblioteca fuzzy (ALVES, 2012). A placa eletrônica pode ser visualizada na Figura 3 e a plataforma de programação Arduíno é ilustrada na Figura 4. É importante salientar que a plataforma Arduíno é uma plataforma de programação muito utilizada na educação devido a sua praticidade e baixo custo.

Figura 1 - Protótipo

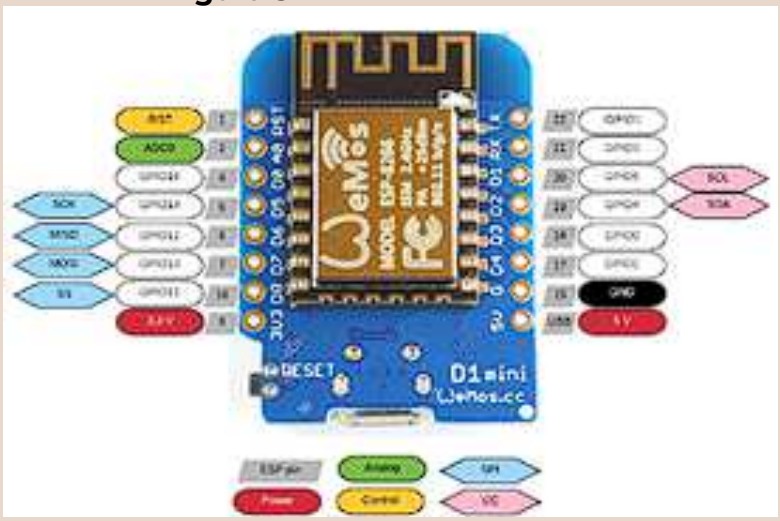
Fonte: Elaborado pela pesquisadora.

.

Figura 2 - Protótipo

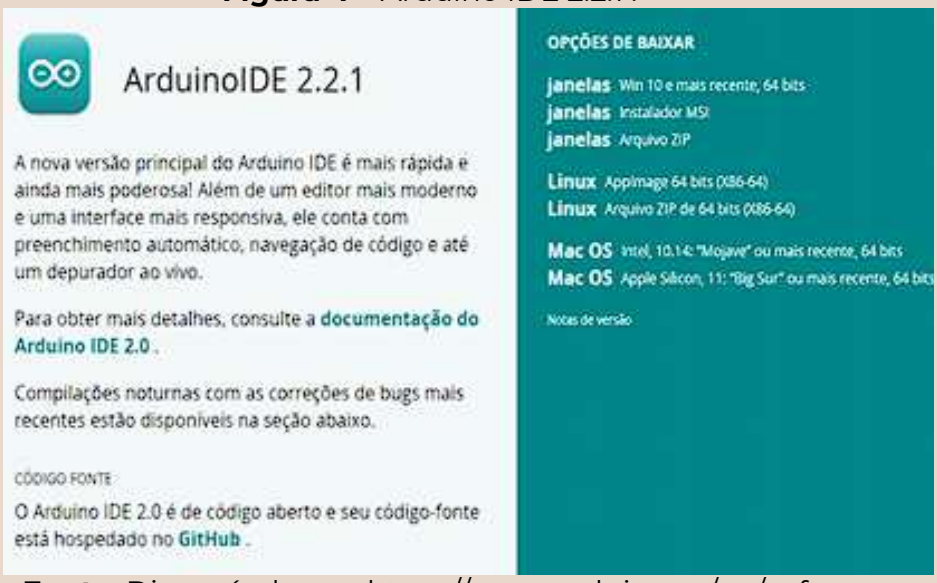
Fonte: Elaborado pela pesquisadora.

Figura 3 - Placa ESP8266



Fonte: Solectroshop.

Figura 4 – Arduino IDE 2.2.1 .

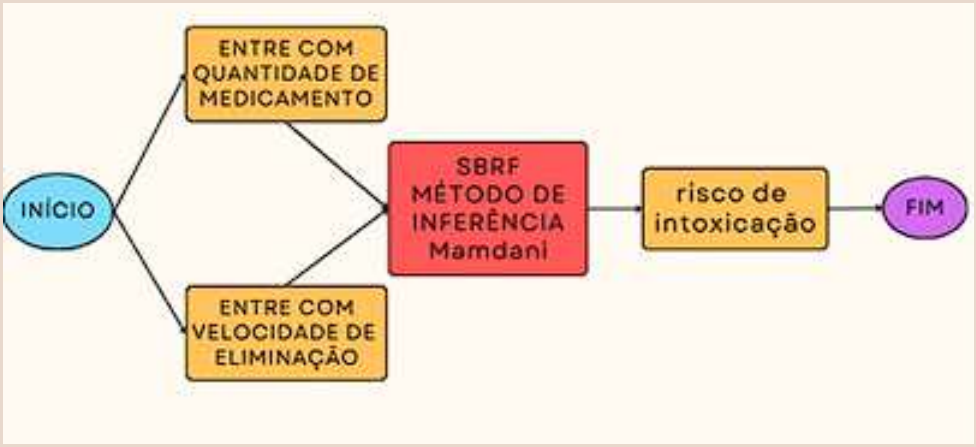


Fonte: Disponível em: <https://www.arduino.cc/en/software>.

2.2 Programa FisPro

O *FisPro* é um *software* computacional utilizado na matemática, possibilitando simulações neste projeto de pesquisa referente a dissertação de mestrado intitulada: **“ESTUDO FARMACOLÓGICO VIA PROGRESSÃO GEOMÉTRICA E CONJUNTOS FUZZY NO ENSINO MÉDIO”**. A partir da construção de um SBRF no *Fispro*, determina-se valores numéricos para o risco de intoxicação de medicamentos, ao inserir valores para velocidade de eliminação e a quantidade de medicamentos. A ferramenta é gratuita e oferece diversos recursos para o desenvolvimento desse projeto. O risco da intoxicação medicamentosa pelo organismo é obtido a partir da programação das variáveis linguísticas de entrada do SBRF que são velocidade de eliminação e quantidade de medicamento (Figura 5). Os métodos de inferência e defuzzificação utilizados são respectivamente, o de Mamdani e o centro de gravidade (JAFELICE; BARROS; BASSANEZI, 2023). Após a construção do SBRF no FisPro, a programação desse sistema é feita pela plataforma Arduíno, na linguagem C, utilizando-se a placa ESP8266, a partir de uma biblioteca *fuzzy*.

Figura 5 – Fluxograma do SBRF do modelo farmacocinético



Fonte: Elaborado pela pesquisadora.

2.3 Aplicativo *Blynk*

A interface entre o controle e monitoramento do risco de intoxicação é realizada a partir da plataforma *Blynk* desenvolvida para praticar a Internet das Coisas (Internet of Things– IoT) (https://blynk.io/), por meio de dispositivos móveis. Para isso, deve-se instalar o aplicativo da plataforma *Blynk* em um smartphone. O Wi-Fi presente na placa ESP 8266 possibilita a interação com o modelo por meio do SBRF. Em seguida, deve-se inserir a variáveis de entrada do SBRF: quantidade de medicamento e velocidade de eliminação. A quantidade de doses administradas serão captadas através do toque em um botão "sensor" instalado na boca do boneco e sensível ao toque. O risco de intoxicação é calculado de acordo com a quantidade de doses inseridas, bem como a velocidade de eliminação escolhida. A Figura 6 ilustra o funcionamento do *smartphone* a partir da plataforma *Blynk*.

Figura 6 – *Smartphone* com as variáveis de entrada e saída do SBRF do modelo farmacocinético

Fonte: Elaborado pela pesquisadora.

3. Construção do Protótipo

O protótipo é um simulador programado para receber comandos por meio de um smartphone, apontando o possível risco da intoxicação medicamentosa pelo organismo humano quando este valor é maior ou igual a 0,8. Para utilizar esse dispositivo, durante a aplicação da sequência didática foi introduzido a teoria dos conjuntos *fuzzy* e construído o SBRF do modelo em estudo no *software* livre FisPro com os estudantes. O SBRF é composto por 4 componentes como mostra a Figura 7: um processador de entrada responsável pela fuzzificação das variáveis de entrada; uma combinação de regras *fuzzy*, chamada base de regras; um método de inferência *fuzzy*; e um processador de saída que realiza a defuzzificação, gerando um número real.

Figura 7 – Esquema do Sistema Baseado em Regras *Fuzzy*

Fonte: Elaborado pela pesquisadora.

3.1 Materiais utilizados para a construção do protótipo

Para construção do protótipo do organismo humano, contou-se com o auxílio de um engenheiro e Mestre em Ensino de Ciências e Matemática da Universidade Federal de Uberlândia. Os materiais utilizados para a produção do protótipo são descritos na Tabela 1.

Tabela 1 - Lista de itens para a construção do protótipo

Lista de Materiais para a Construção do Protótipo			
Material	Nome	Descrição	Quant.
	[1]Boneco de plástico	Boneco de plástico translúcido (60 cm) representa um organismo humano.	1
	[2]Placa eletrônica ESP8266	Recebe e decodifica os dados acionando as luzes de LED.	1
	[3]LED WS 2812	Indica o nível do risco de intoxicação por meio da variação de cores.	1
	Micro chave[4]	Sensor de entrada, simula a administração do medicamento por meio do toque.	1

	[5] Prensa cabo 3/4	Suporte, acopla o tubo de alumínio ao dorso do protótipo e conduz a extensão com a luz de LED ao interior (abdômen do boneco).	2
	[6] Cabo USB	Conecta a placa ESP8266 ao computador.	1
	[7] Tubo Alumínio (1 cm de diâmetro e 12 cm de comprimento)	Interliga o botão e os fios que levam as informações a placa ESP8266.	1

[1] Adquirido pela pesquisadora.
[2,3] Disponível em: <https://www.smartkits.com.br/>.
[4,5,7] Disponível em:<https://produto.mercadolivre.com.br/>.
[6] Disponível em: <https://www.lojadofilmmaker.com>.

3.2 Montagem do protótipo

Para a construção do protótipo simulador do risco de intoxicação medicamentosa, utilizou-se um boneco de plástico translúcido a fim de que os sinais luminosos vindos de seu interior ficassem nítidos para a visualização. Assim, foi instalado na parte frontal do boneco (boca) uma micro chave sensível ao toque e capaz de captar os comandos vindos da parte externa. Para isso realizou-se uma perfuração pequena na boca do boneco para que a micro chave fixada na extremidade do tubo de alumínio, pudesse ser posicionada na boca do protótipo (Figura 8). Os cabos ligados à micro chave chegam até a placa eletrônica ESP8266 por meio do tubo de alumínio anexado por uma prensa ¾ na parte superior dorsal do boneco, (Figura 9). Dessa forma, foi feita uma perfuração com a mesma espessura da prensa para que esta ficasse bem fixada a estrutura (boneco).

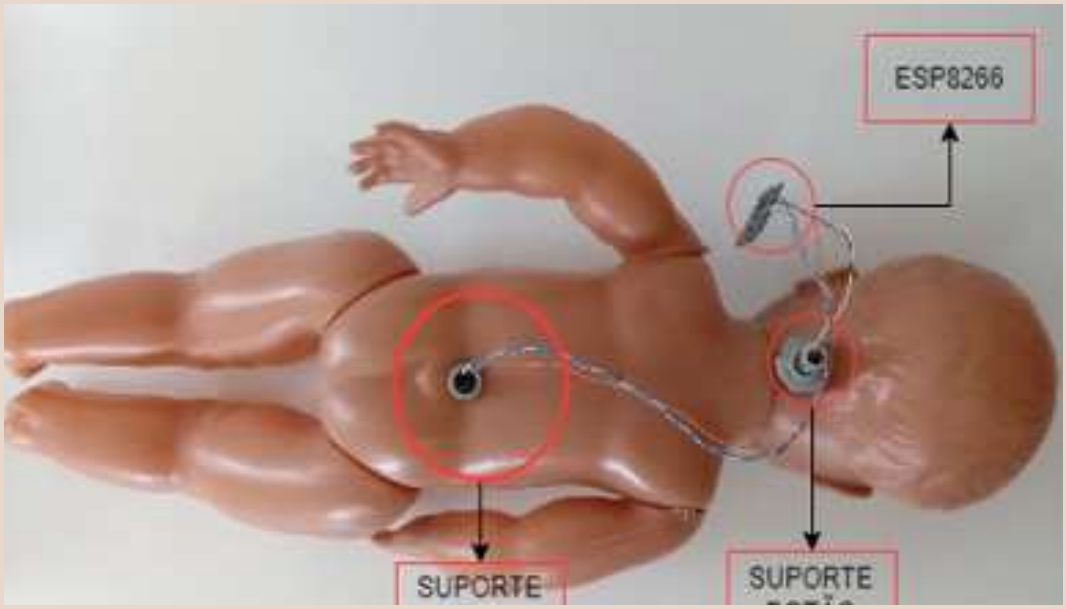
Em seu interior (abdômen) foi acoplado um ponto de luz Led. A entrada dos cabos de luz e sua condução é feita por meio da prensa de cabo ¾, posicionada no dorso do protótipo na parte inferior, também por meio de uma perfuração e fixação da prensa 3/4 (Figura 9). Tanto os cabos de entrada quanto os de saída estão ligados à placa ESP8266, responsável por receber e decodificar os dados. A placa se posiciona entre os cabos de entrada e saída na parte externa do protótipo (Figura 11).

Figura 8 - Botão “sensor” posicionado na boca do protótipo

Fonte: Elaborado pela pesquisadora.

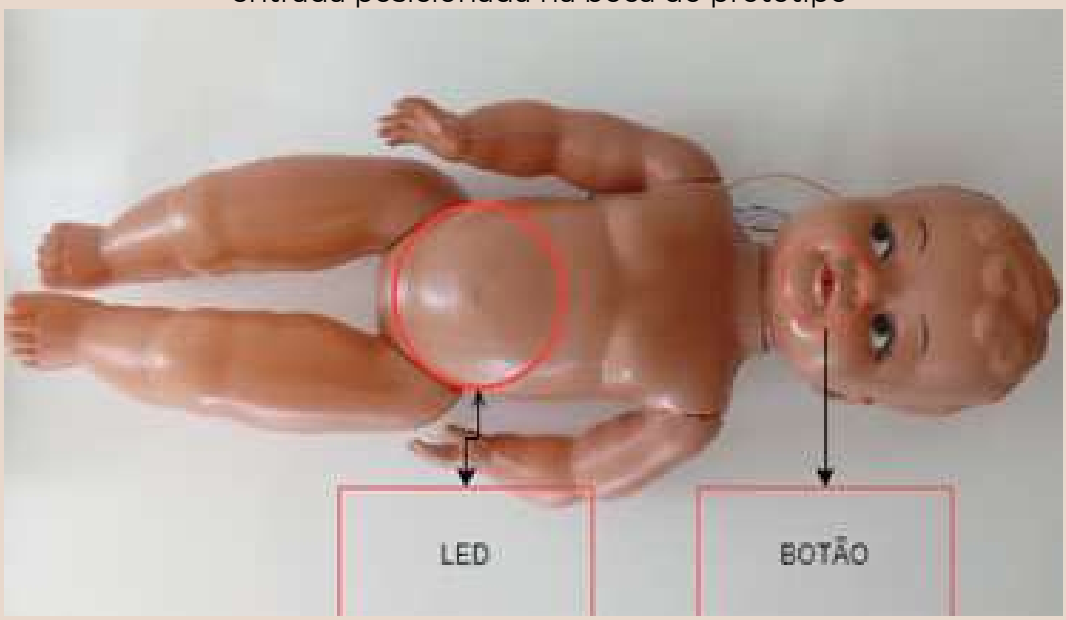
A Figura 9 mostra os suportes para a luz de Led no dorso do boneco. O botão e a luz de Led instalados na parte frontal são mostrados na Figura 10.

Figura 9 - Suporte para luz de Led, botão “sensor” e comandos placa eletrônica ESP8266



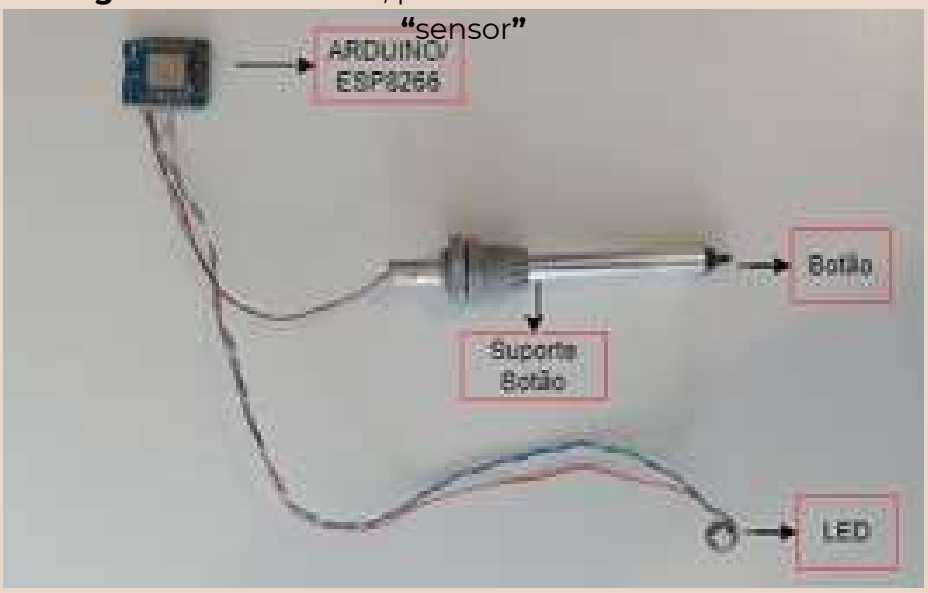
Fonte: Elaborado pela pesquisadora.

Figura 10- Luz de Led posicionada no abdômen do protótipo e sensor de entrada posicionada na boca do protótipo



Fonte: Elaborado pela pesquisadora.

Figura 11 - Luz de Led, placa eletrônica ESP8266 botão “sensor”



Fonte: Elaborado pela pesquisadora**.**

O SBRF que modela o risco de intoxicação é descrito na próxima seção. A luz será acionada representando sinais de alerta ao piscar e variar as cores. As cores no abdômen do protótipo estão programadas da seguinte forma:

- Se o risco está entre [0 ; 0,3), então a luz verde acende;
- Se o risco está entre [0,3 ; 0,8), então a luz amarela acende;
- Se o risco está entre [0,8 ; 1], então a luz vermelha acende.

3.3 Sistema Baseado em Regras *Fuzzy* (SBRF) do modelo farmacocinético

Nesta seção, apresenta-se o passo a passo da construção do SBRF do modelo estudado e os parâmetros utilizados para calcular os possíveis riscos de intoxicação pelo organismo.

1. Abra o software *FisPro* (Figura 12).

Figura 12 - Janela inicial do *software FisPro*

Fonte: Elaborado pela pesquisadora (*Software FisPro*).

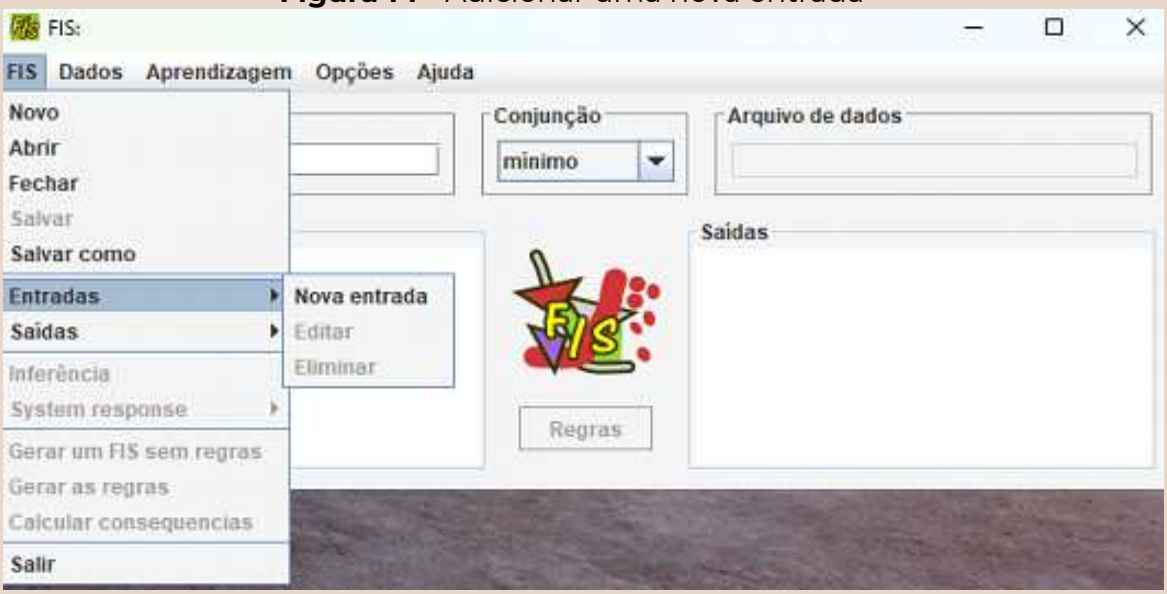
2. Dê um clique em FIS e, na sequência, clique em Novo para criar um novo documento (Figura 13).

Figura 13 - Janela do *software FisPro* para criar um novo documento

Fonte: Elaborado pela pesquisadora (*Software FisPro*).

3. Para inserir as variáveis de entrada clique sequencialmente em: FIS – Entrada – Nova Entrada (Figura 14).

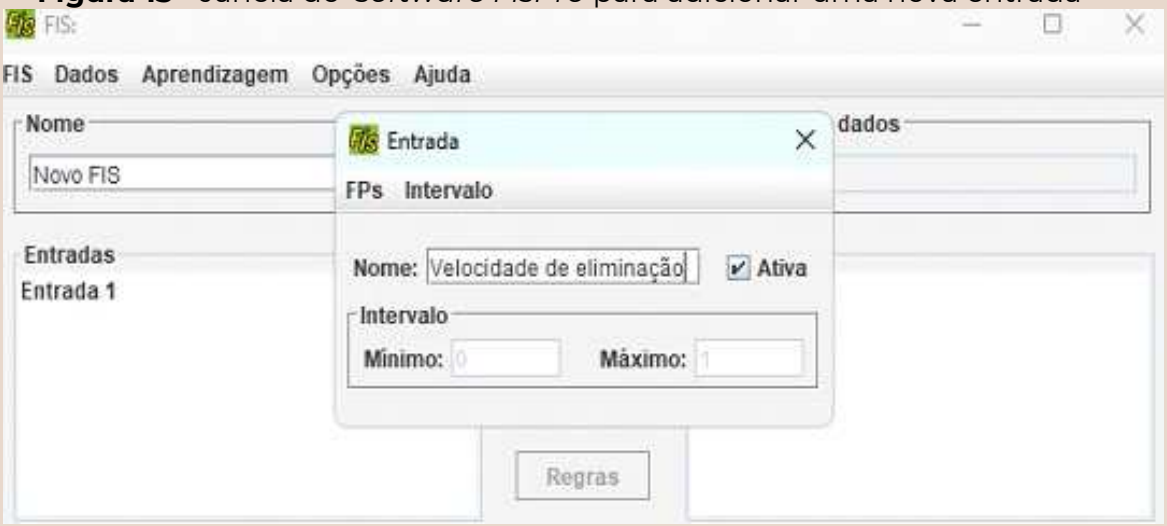
Figura 14 - Adicionar uma nova entrada



Fonte: Elaborado pela pesquisadora (*Software FisPro*).

4. Na janela, aberta a seguir, adicione o nome e o domínio da variável (Figura 15).

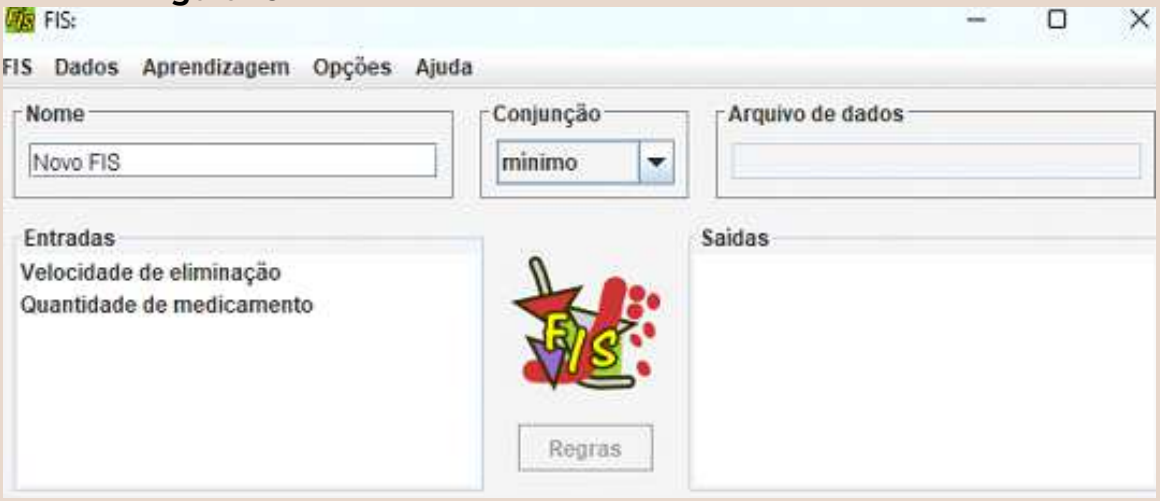
Figura 15 - Janela do *software FisPro* para adicionar uma nova entrada



Fonte: Elaborado pela pesquisadora (*Software FisPro*).

5. Adicione a próxima variável de entrada do modelo em estudo (Figura 16)

Figura 16 - Variáveis de entrada do modelo em estudo



Fonte: Elaborado pela pesquisadora (*Software FisPro*).

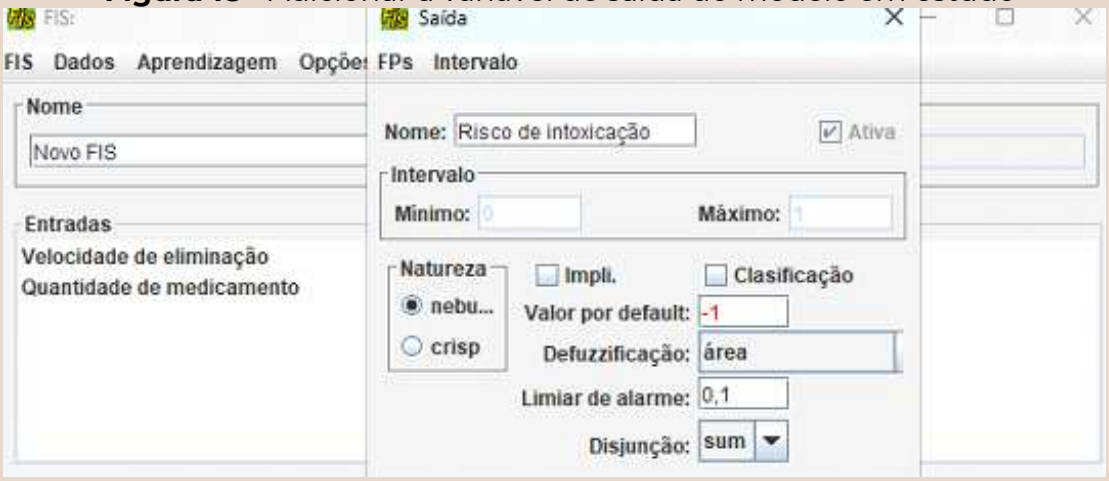
6. Adicione a variável de saída, a partir da sequência: Clique em FIS – Saídas – Adicionar saída (Figura 17).

Figura 17 - Adicionar as variáveis de saída do modelo em estudo

Fonte: Elaborado pela pesquisadora (*Software FisPro*).

7. Uma nova janela é aberta e nela adicione o nome da variável de saída, o intervalo do domínio da variável e a natureza fuzzy nebulosa (Figura 18).

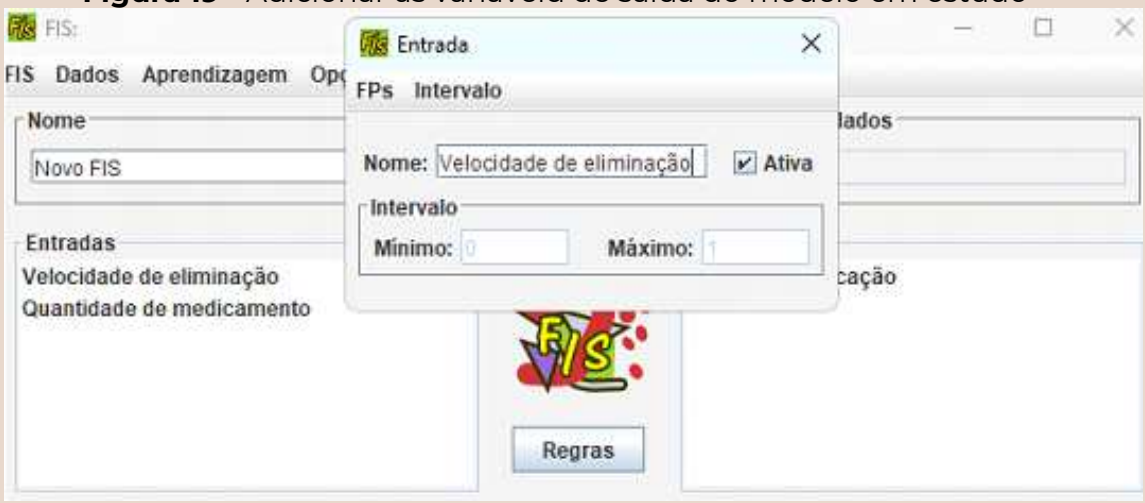
Figura 18 - Adicionar a variável de saída do modelo em estudo



Fonte: Elaborado pela pesquisadora (*Software FisPro*).

8. Em seguida, adicione os termos linguísticos às variáveis. Para isso, dê dois cliques sobre os nomes para que a janela de entrada se abra (Figura 19).

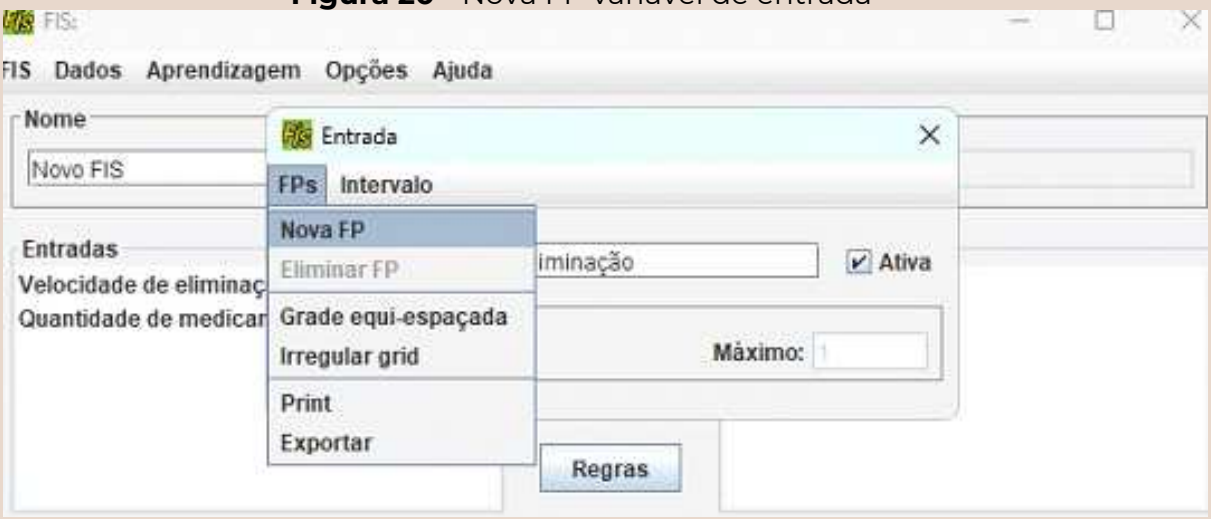
Figura 19 - Adicionar as variáveid de saída do modelo em estudo



Fonte: Elaborado pela pesquisadora (*Software FisPro*).

9. Para inserir os termos linguísticos clique em FPs (Funções de Pertinência) e, em seguida, em Nova FP (Figura 20).

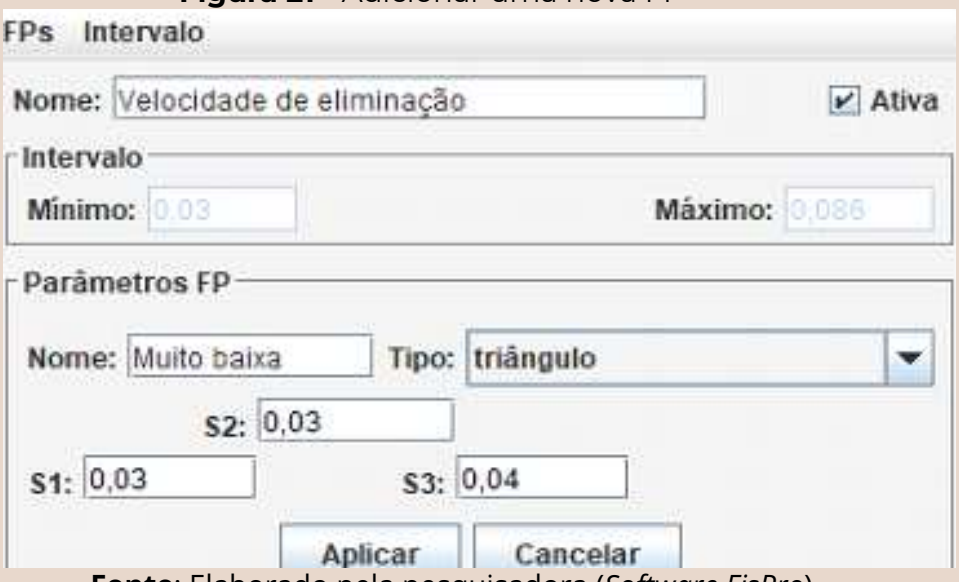
Figura 20 - Nova FP variável de entrada



Fonte: Elaborado pela pesquisadora (*Software FisPro*).

10. Insira os termos linguísticos das variáveis de entrada e saída, em seguida observe o nome da variável e o respectivo intervalo do domínio. Em FP, insira os nomes dos termos linguísticos, o tipo e os intervalos para o domínio (Figura 21).

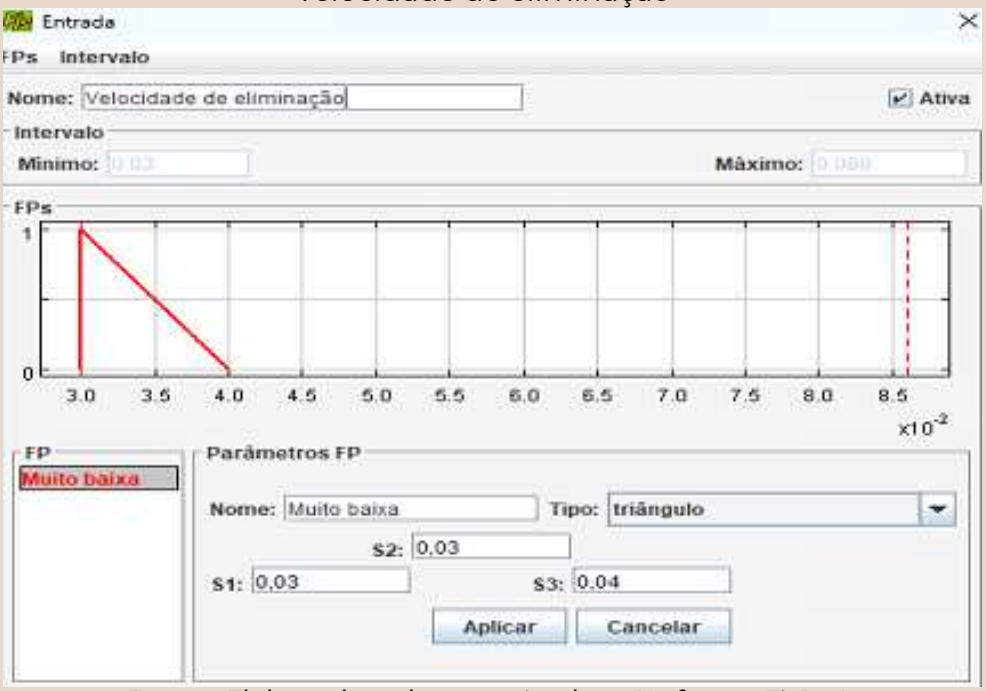
Figura 21 - Adicionar uma nova FP



Fonte: Elaborado pela pesquisadora (*Software FisPro*).

11. Clique em aplicar para abrir a janela da variável e o gráfico (Figura 22).

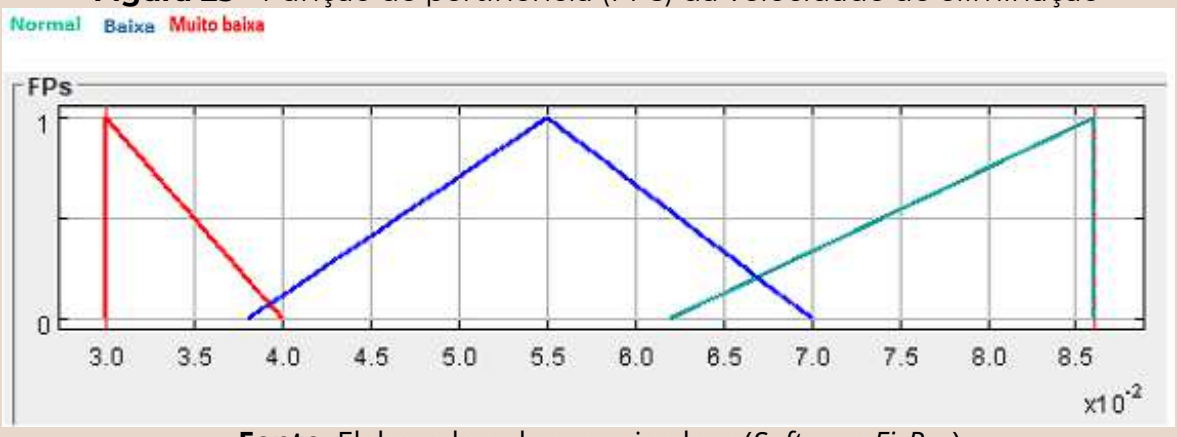
Figura 22 - Função de pertinência da variável de entrada velocidade de eliminação



Fonte: Elaborado pela pesquisadora (*Software FisPro*).

12. Na Figura 23 são apresentados os gráficos das funções da variável de entrada velocidade de eliminação. Os termos linguísticos utilizados são: Muito baixa, Baixa e Normal.

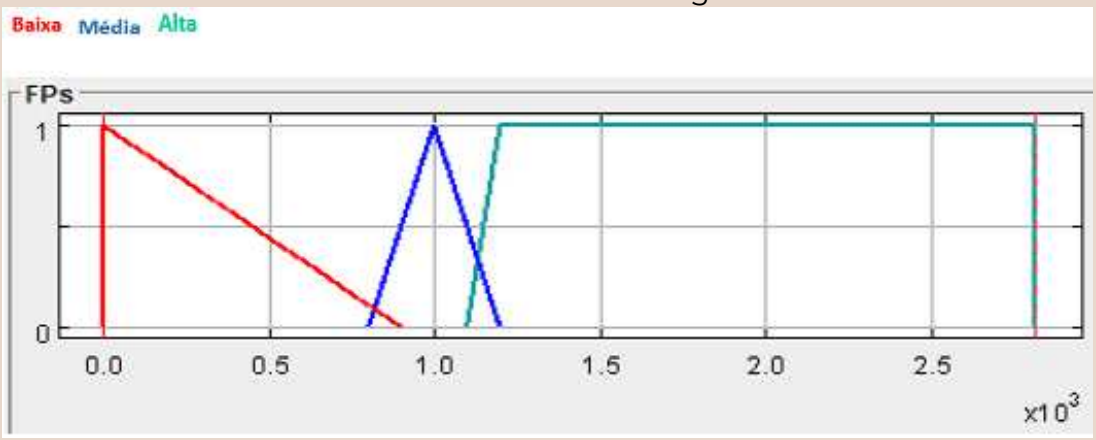
Figura 23 - Função de pertinência (FPs) da velocidade de eliminação



Fonte: Elaborado pela pesquisadora (*Software FisPro*).

13. Na Figura 24 são apresentados os gráficos das funções da variável de entrada quantidade de medicamento. Os termos linguísticos são: Baixa, Média e Alta. A variável “quantidade de medicamento” possui o domínio de 0 a 3000 mg.

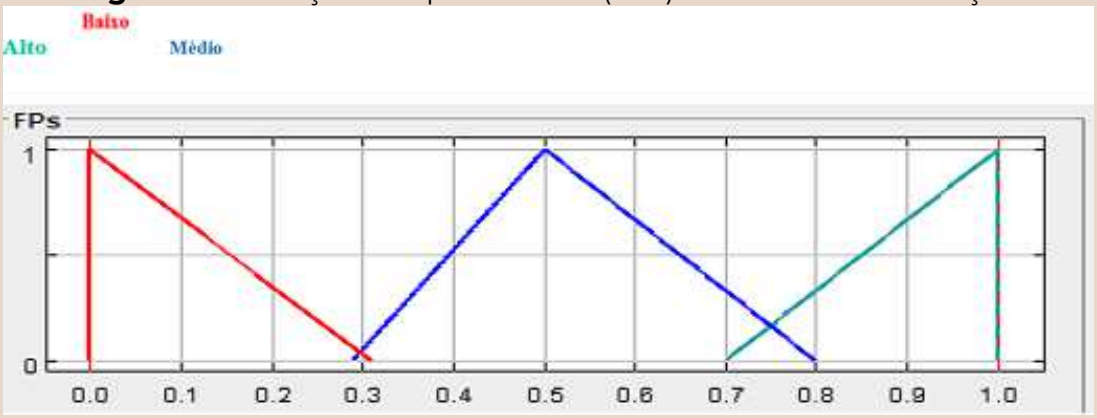
Figura 24 - Funções de pertinência (FPs) da quantidade de medicamento em mg



Fonte: Elaborado pela pesquisadora (*Software FisPro*).

14. A Figura 25 reserva-se os gráficos das funções de pertinência do risco de intoxicação do organismo. Os termos linguísticos do risco de intoxicação são: Baixo, Médio e Alto. O domínio da variável de saída risco de eliminação está entre 0 e 1.

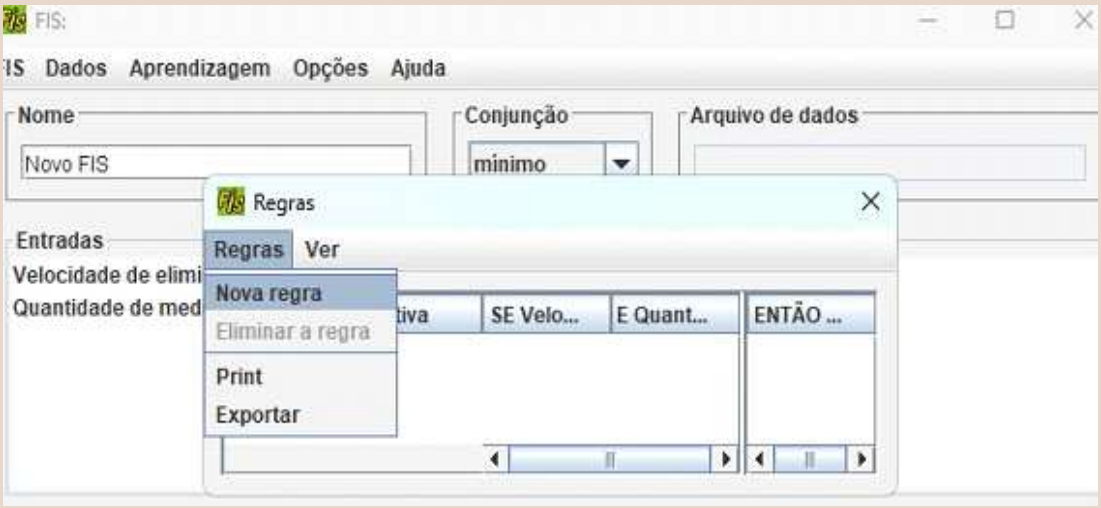
Figura 25 - Funções de pertinência (FPs) do risco de intoxicação



Fonte: Elaborado pela pesquisadora (*Software FisPro*).

15. Para construir a base de regras clique em Regras e, posteriormente em Nova Regra (Figura 26). Na sequência insira as possibilidades de combinação das variáveis de entrada.

Figura 26 -Janela de Regras



Fonte: Elaborado pela pesquisadora (*Software FisPro*).

As regras fuzzy são apresentadas na Tabela 2.

Tabela 2 - Base de regras *fuzzy*

Quantidade de Medicamento/ Velocidade	**Baixa**	**Média**	**Alta**
Muito baixa	Médio	Alto	Alto
Baixa	Médio	Médio	Alto
Normal	Baixo	Médio	Alto

Fonte: Elaborado pela pesquisadora.

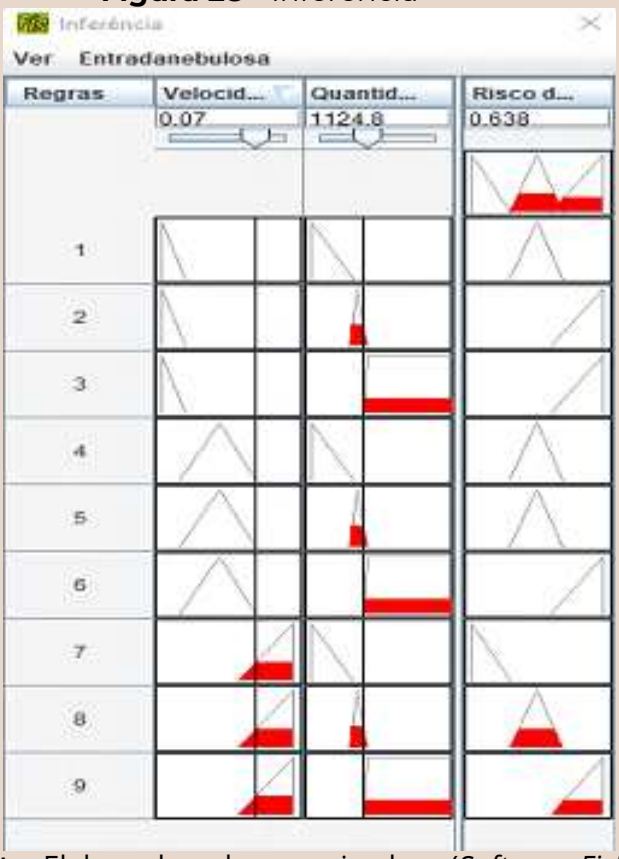
16. Para verificar o método de inferência clique em FIS e, em seguida, em Inferência de acordo com a Figura 27.

Figura 27 - Janela de Inferências

Fonte: Elaborado pela pesquisadora (*Software FisPro*).

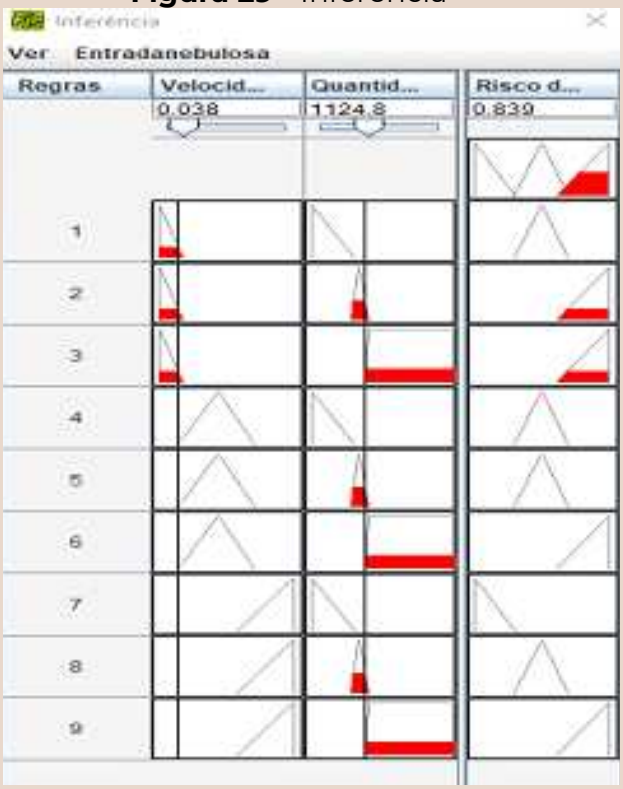
Nas Figuras 28 e 29 são inseridos valores para velocidade de eliminação e quantidade de medicamento e é determinado o valor do risco de intoxicação.

Figura 28 - Inferência



Fonte: Elaborado pela pesquisadora (*Software FisPro*).

Figura 29 - Inferência

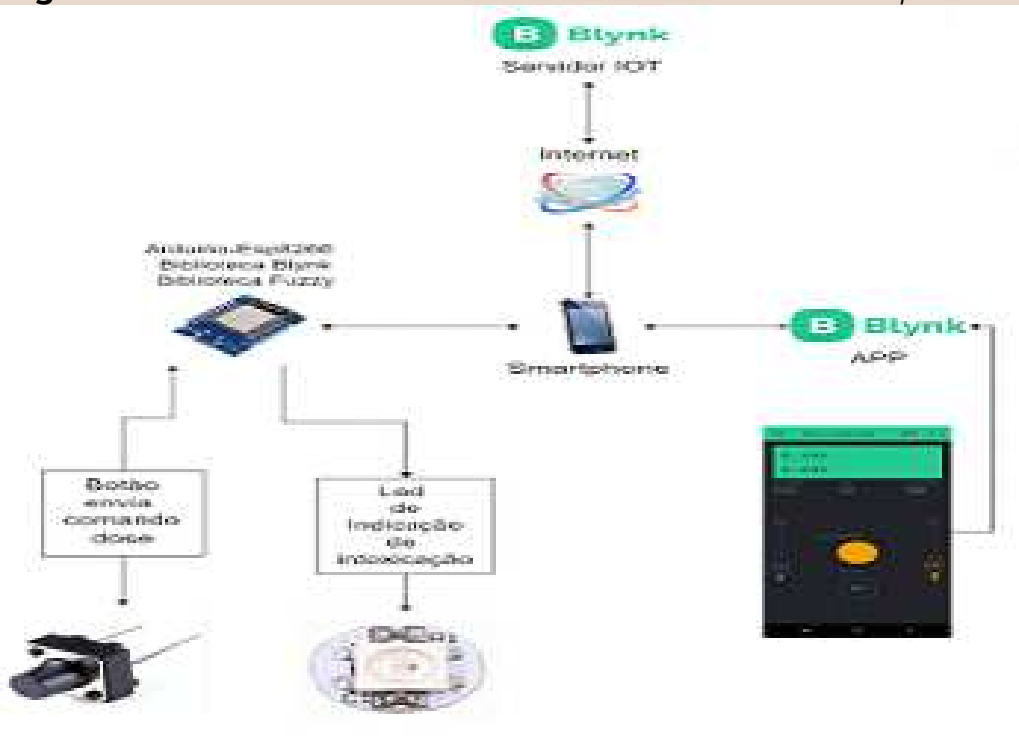


Fonte: Elaborado pela pesquisadora (*Software FisPro*).

3.4 Estrutura da parte eletrônica do modelo farmacocinético

Para que haja a interface de controle e monitoramento do risco de intoxicação foi utilizado a plataforma *Blynk*, por meio de dispositivos móveis como já foi dito anteriormente. O esquema a seguir demonstra a conexão entre os sensores. O controle do simulador é feito a partir do Arduíno, utilizando a comunicação sem fio Wi-Fi (Figura 30).

Figura 30 - Estrutura de conexão dos sensores com os *smartphones*

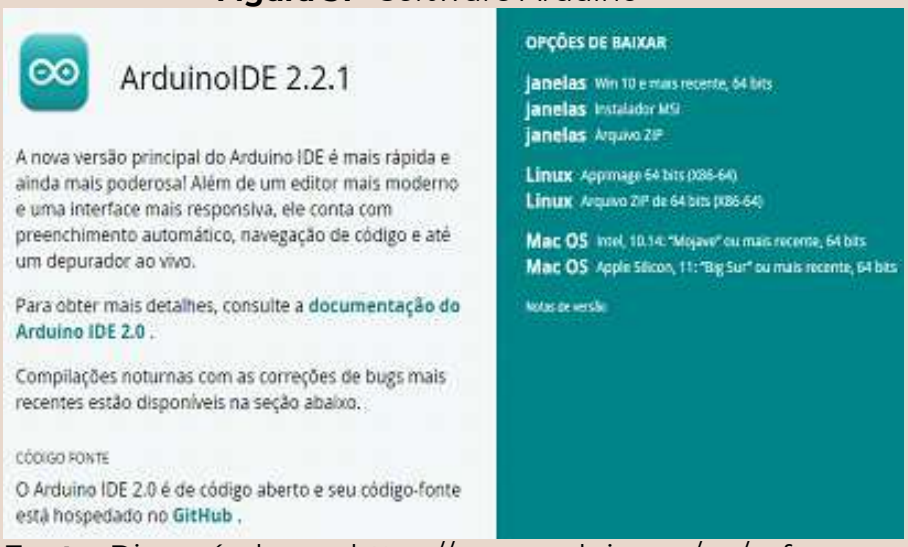


Fonte: Elaborado pela pesquisadora.

3.4.1 Conectando o Arduíno IDE com ESP8266 e Blynk

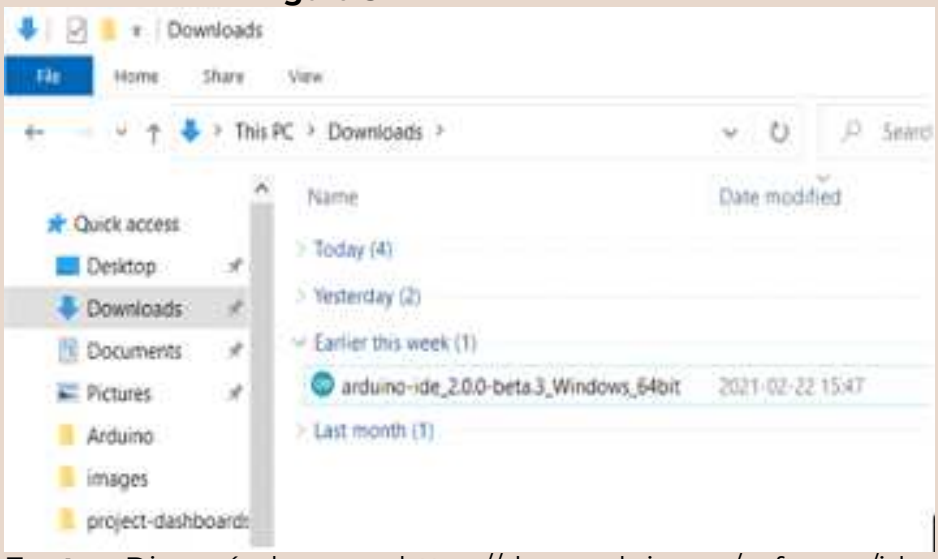
Para a programação do Arduíno IDE é necessário inicialmente baixar o editor e realizar a instalação do *software*, veja Apêndice. Esse procedimento poderá ser feito com o auxílio de um tutorial disponível em: https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-v2-downloading-and-installing e seguindo as instruções no guia como demonstram as Figuras 31, 32 e 33. É necessário escolher o sistema operacional, no modelo em estudo o sistema escolhido foi o Windows.

Figura 31 - *Software* Arduíno



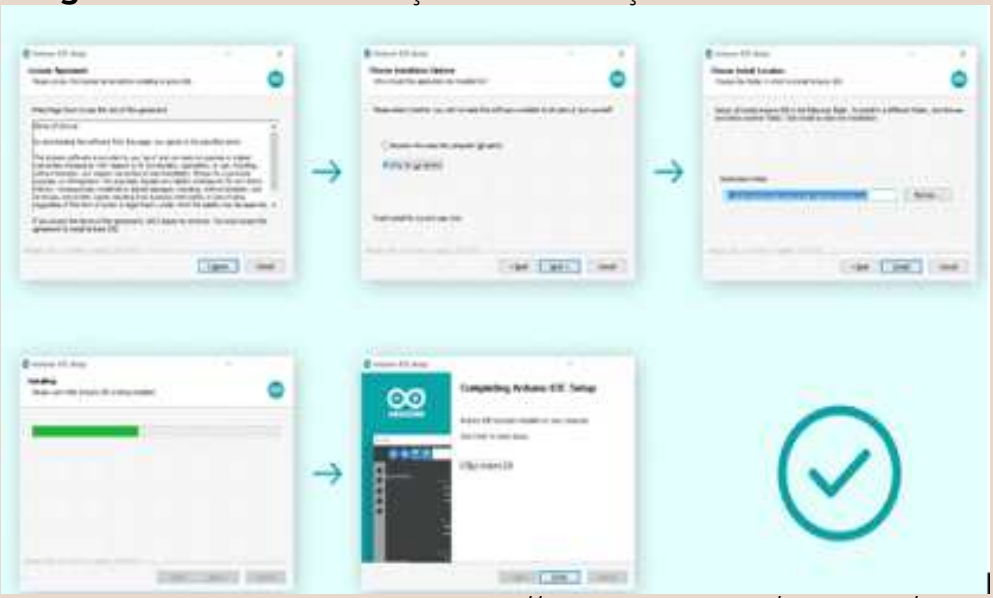
Fonte: Disponível em: <https://www.arduino.cc/en/software>.

Figura 32 - *Software* Arduíno



Fonte: Disponível em: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-downloading-and-installing>.

Figura 33 - Guia de instruções de instalação do Arduíno IDE 2.0



Fonte: Disponível em: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-downloading-and-installing>.

Após a instalação do Arduíno IDE e programação do módulo Wi-Fi ESP8266 é possível acessar o gerenciador de bibliotecas para adicionar a biblioteca *Blynk*. Esse procedimento poderá ser feito utilizando o tutorial para a instalação de bibliotecas disponível em: https://docs.arduino.cc/software/ide-v1/tutorials/installing-libraries.

3.4.2 Sensor interno do protótipo

O sensor interno localizado no abdômen do protótipo sinaliza o risco de intoxicação do organismo por meio de cores. As variáveis de entrada e saída foram programadas na plataforma Arduíno e estão associadas à placa ESP 8266. Por meio do aplicativo Blynk, há a interação dos estudantes, a partir da inserção das variáveis de entrada do SBRF. As doses administradas são captadas pelo toque em um botão “sensor”. Os dados são enviados e decodificados e a partir do SBRF obtém-se a variável de saída, o risco de intoxicação. O ponto de luz localizado no abdômen do protótipo, aciona cores diferentes de acordo com os valores, do risco de intoxicação do organismo (Figuras 34, 35 e 36).

• Risco de intoxicação entre [0 ; 0,3), a luz verde acende;

Figura 34 - Sinal de luz de Led verde localizada no abdômen do protótipo

Foto: Acervo da pesquisadora.

• Risco de intoxicação entre [0,3 ; 0,8), a luz amarela acende;

Figura 35 - Sinal de luz de Led amarela localizada no abdômen do protótipo

Foto: Acervo da pesquisadora.

• Risco de intoxicação entre [0,8 ; 1], a luz vermelha acende.

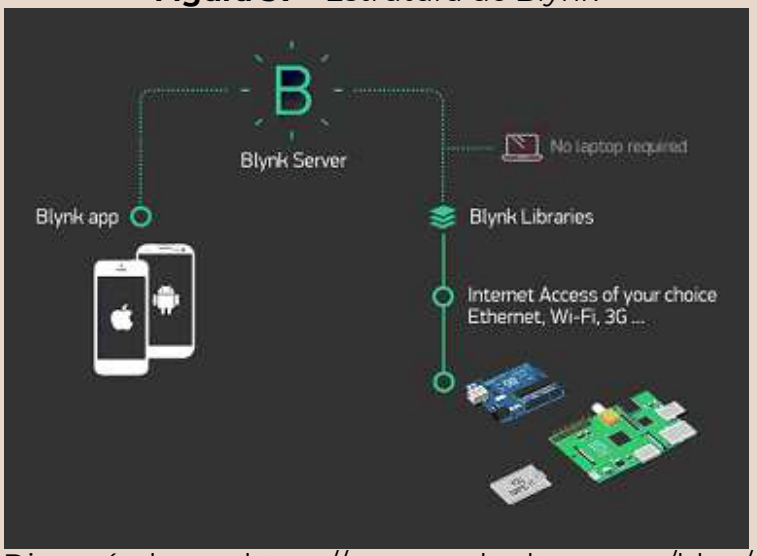
Figura 36 - Sinal de luz de Led vermelha localizada no abdômen do protótipo

Foto: Acervo da pesquisadora.

3.4.3 Aplicativo *Blynk*

A Plataforma *Blynk* foi desenvolvida para praticar a Internet das Coisas (Internet of Things – IoT) (*BLYNK*) possibilitando interconectar objetos em rede a partir da internet. O *Blynk* é um aplicativo que permite controlar dispositivos programáveis e enviar dados de sensores e módulos para aplicativos móveis de forma remota (Figura 37). Além disso, oferece várias funcionalidades adicionais sem a exigência de conhecimentos avançados em programação.

Figura 37 - Estrutura do *Blynk*



Fonte: Disponível em: <https://www.makerhero.com/blog/projetos-de-iot-com-blynk-e-nodemcu/>.

A utilização do *Blynk* com o ESP 8266 associado a plataforma Arduíno permite o desenvolvimento de projetos Iot como o modelo farmacocinético apresentado. O *download* e instalação do App em um dispositivo móvel poderá ser feito a partir da loja de aplicativos do celular, porém em Android basta acessar o *Play Store* do aparelho e dar início a instalação. O tutorial para a criação de um projeto no *Blynk* App está disponível em https://embarcados.com.br/introducao-ao-blynk-app/#Criando-o-primeiro-projeto-no-Blynk-App. Na Figura 38 pode-se observar os estudantes em simulações com o protótipo.

Figura 38 - Simulações dos estudantes com o protótipo

Foto: Acervo da pesquisadora.

4. Considerações Finais

Os estudos realizados a partir da pesquisa, foco deste trabalho, demonstraram a importância da associação da Modelagem Matemática às tecnologias educacionais, permitindo a abordagem da lógica *fuzzy,* um conceito novo para o Ensino Médio. O TCT saúde possibilitou um trabalho interdisciplinar entre os componentes curriculares Matemática e Química, facilitando a compreensão dos conteúdos e promovendo uma visão integradora do conhecimento. A utilização do pensamento computacional como recurso para aprendizagem, demonstrou a relevância do uso das tecnologias digitais tanto para a investigação quanto para a aplicação matemática, desenvolvendo competências e habilidades cognitivas e socioemocionais tendo a realidade como referencial, de acordo com as orientações da BNCC (BRASIL, 2018). A avaliação da aprendizagem dos estudantes ocorreu de forma contínua e associada a aplicação da sequência didática, sendo o jogo aplicado ao final do processo. Este foi elaborado na Platoforma *Kahoot* em um formato *quiz,* sendo uma proposta lúdica e descontraída, despertando a motivação e entusiasmo entre os estudantes.

Assim, a pesquisadora espera que esse material possa ser utilizado por professores em experiências didáticas interessantes, colaborando para o protagonismo e construção das aprendizagens de forma crítica e propositiva.

5. Referências

ARDUINO. Plataforma *online* (S/D). Disponível em: <https://www.arduino.cc/en/software>Acesso em: 04 fev. 2022.

ALVES. A. J. Uma biblioteca Fuzzy para Arduino e Sistemas Embarcados. Blog ZeRoKoL. Teresina – PI.; 28 de set de 2012. Disponível em: < https://blog.zerokol.com/2012/09/ arduinofuzzy-uma-biblioteca-fuzzy-para.html>. Acesso em: 04 ago. 2022.

BLYNK. Plataforma *online* (S/D). Disponível em: <https://blynk.io/>. Acesso em: 04 out. 2022.

BRASIL. *Base Nacional Comum Curricular (BNCC)*, Brasília: MEC, 2018. Disponível em: <http//basenacionalcomum.mec.gov.br/>. Acesso em: 02 set. 2022.

_____. Secretaria de Educação Fundamental. *Parâmetros Curriculares Nacionais:* apresentação dos Temas Contemporâneos Transversais, ética/Secretaria de Educação Fundamental. Brasília: MEC/SEF, 1997.

CANVA. Plataforma *online* S/D). Disponível em: < https://www.canva.com/pt_br/>. Acesso em: 04 set. 2022.

FISPRO. Fuzzy Inference System Professional. Software Livre. Disponível em: <https://www.fispro.org/en/> Acesso em: 16 ago. 2022.

JAFELICE, R. S. M.; BARROS, L. C.; BASSANEZI, R. C. Teoria dos Conjuntos Fuzzy com Aplicações. 3.ed. São Carlos, SP: SBMAC, 2023, 136p (Notas em Matemática Aplicada; v.17).

KAHOOT. Plataforma *online*, (S/D). Disponível em: < https:// kahoot.com>. Acesso em: 15 de nov. 2022.

APÊNDICE

SOFTWARE C.

```
#INCLUDE <FASTLED.H>


#DEFINE BLYNK_TEMPLATE_ID ""
#DEFINE BLYNK_DEVICE_NAME "TESTE"
#DEFINE BLYNK_AUTH_TOKEN ""

#DEFINE FASTLED_INTERRUPT_RETRY_COUNT 0


#INCLUDE <WIRE.H>

#INCLUDE <FUZZY.H>

#INCLUDE <ESP8266WIFI.H>

#INCLUDE <BLYNKSIMPLEESP8266.H>

#DEFINE FASTLED_INTERNAL

#DEFINE NUM_LEDS 1

#DEFINE DATA_PIN 4


CRGB LEDS[NUM_LEDS];

CHAR AUTH[] = BLYNK_AUTH_TOKEN;

CHAR SSID[] = "REDE WIFI";
CHAR PASS[] = "SENHA DA REDE WIFI";

BLYNKTIMER TIMER_VERIFICA_BOTAO;
BLYNKTIMER TIMERMEDEVOLUME;
WIDGETLED LED1(V2);
WIDGETLCD LCD(V10);
```

```
#DEFINE BLYNK_GREEN     "#23C48E"
#DEFINE BLYNK_BLUE      "#04C0F8"
#DEFINE BLYNK_YELLOW    "#ED9D00"
#DEFINE BLYNK_RED       "#D3435C"
#DEFINE BLYNK_DARK_BLUE "#5F7CD8"


FLOAT VELOCIDADE = 0; //VELOCIDADE
FLOAT QUANTIDADE_INICIAL = 0; //QUANTIDADE
FLOAT QUANTIDADE[16] = {};
INT   DOSE = 0;

FLOAT DOSES[50]={};
FLOAT DOSES_ACUMULADAS[50]={};
FLOAT DOSE_INICIAL = 600;

STRING STR_1;
STRING STR_2;

INT SLIDER=0;
BLYNK_WRITE(V3)
{

   VELOCIDADE = PARAM.ASFLOAT();
   BLYNK.VIRTUALWRITE(V8, VELOCIDADE);
   BLYNK.VIRTUALWRITE(V0, VELOCIDADE);
   SLIDER = 1;

   //FUZZY_CALC();
   //BLYNK.VIRTUALWRITE(V4, OUTPUT_FUZZY);
}


BLYNK_WRITE(V1)
{
   QUANTIDADE_INICIAL = PARAM.ASFLOAT();
   BLYNK.VIRTUALWRITE(V9, QUANTIDADE_INICIAL);
   BLYNK.VIRTUALWRITE(V4, QUANTIDADE_INICIAL);
   SLIDER = 1;
   //FUZZY_CALC();
   //BLYNK.VIRTUALWRITE(V4, OUTPUT_FUZZY);
}
```

```
BLYNK_WRITE(V7)
{

   IF(PARAM.ASINT() == 1) {     // IF BUTTON SENDS 1

      DOSE = 0;
      BLYNK.VIRTUALWRITE(V5, DOSE);
      BLYNK.VIRTUALWRITE(V9, QUANTIDADE_INICIAL);
      LEDS[0] = CRGB::BLACK;
      FASTLED.SHOW();
      CALC_DOSES_ACUMULADAS();
      LCD.CLEAR(); //USE IT TO CLEAR THE LCD WIDGET
      LCD.PRINT(0, 0, "PRONTO PARA"); // USE: (POSITION X: 0-15,
POSITION Y: 0-1, "MESSAGE YOU WANT TO PRINT")
      LCD.PRINT(0, 1, "SIMULAÇÃO");
      }

}

FLOAT Y,Y_1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10,Y11,Y12,Y13,Y14,Y15;

// FUZZY
FUZZY *FUZZY = NEW FUZZY();

// FUZZYINPUT  DE ELIMINAÇÃO
FUZZYSET *VELOCIDADE_MUITOBAIXA = NEW FUZZYSET(0.03, 0.03,
0.03, 0.04);
FUZZYSET *VELOCIDADE_BAIXA = NEW FUZZYSET(0.038, 0.055, 0.055,
0.07);
FUZZYSET *VELOCIDADE_NORMAL = NEW FUZZYSET(0.062, 0.086,
0.086, 0.086);
```

```
// FUZZYINPUT QUANTIDADE DE MEDICAMENTO
FUZZYSET *QUANTIDADE_PEQUENO = NEW FUZZYSET(0.0, 0.0, 0.0,
900);
FUZZYSET *QUANTIDADE_MEDIO = NEW FUZZYSET(800, 1000,1000,
1200);
FUZZYSET *QUANTIDADE_ALTO = NEW FUZZYSET(1100, 1200, 3000,
3000);

// FUZZYOUTPUT RISCO DE INTOXICAÇÃO
FUZZYSET *RISCO_BAIXO = NEW FUZZYSET(0.0, 0.0, 0.0, 0.31);
FUZZYSET *RISCO_MEDIO = NEW FUZZYSET(0.29, 0.5, 0.5, 0.7);
FUZZYSET *RISCO_ALTO = NEW FUZZYSET(0.6, 1.0,1.0, 1.0);



VOID SETUP()
{


// FUZZYINPUT
  FUZZYINPUT *VELOCIDADE_ELIMINACAO = NEW FUZZYINPUT(1);

  VELOCIDADE_ELIMINACAO-
>ADDFUZZYSET(VELOCIDADE_MUITOBAIXA);
  VELOCIDADE_ELIMINACAO->ADDFUZZYSET(VELOCIDADE_BAIXA);
  VELOCIDADE_ELIMINACAO->ADDFUZZYSET(VELOCIDADE_NORMAL);
  FUZZY->ADDFUZZYINPUT(VELOCIDADE_ELIMINACAO);

  // FUZZYINPUT
  FUZZYINPUT *QUANTIDADE_EDICAMENTO = NEW FUZZYINPUT(2);

QUANTIDADE_EDICAMENTO-
>ADDFUZZYSET(QUANTIDADE_PEQUENO);
  QUANTIDADE_EDICAMENTO->ADDFUZZYSET(QUANTIDADE_MEDIO);
  QUANTIDADE_EDICAMENTO->ADDFUZZYSET(QUANTIDADE_ALTO);
  FUZZY->ADDFUZZYINPUT(QUANTIDADE_EDICAMENTO);


  // FUZZYOUTPUT
  FUZZYOUTPUT *RISCO_INTOXICACAO = NEW FUZZYOUTPUT(1);
```

```
  RISCO_INTOXICACAO->ADDFUZZYSET(RISCO_BAIXO);
  RISCO_INTOXICACAO->ADDFUZZYSET(RISCO_MEDIO);
  RISCO_INTOXICACAO->ADDFUZZYSET(RISCO_ALTO);
  FUZZY->ADDFUZZYOUTPUT(RISCO_INTOXICACAO);


  // BUILDING FUZZYRULE1------

  FUZZYRULEANTECEDENT
*IF_VELOCIDADE_MUITOBAIXA_AND_QUANTIDADE_PEQUENO = NEW
FUZZYRULEANTECEDENT();
  IF_VELOCIDADE_MUITOBAIXA_AND_QUANTIDADE_PEQUENO-
>JOINWITHAND(VELOCIDADE_MUITOBAIXA, QUANTIDADE_PEQUENO);


  FUZZYRULECONSEQUENT *THENRISCO_INTOXICACAO1 = NEW
FUZZYRULECONSEQUENT();
  THENRISCO_INTOXICACAO1->ADDOUTPUT(RISCO_MEDIO);


  FUZZYRULE *FUZZYRULE1 = NEW FUZZYRULE(1,
IF_VELOCIDADE_MUITOBAIXA_AND_QUANTIDADE_PEQUENO,
THENRISCO_INTOXICACAO1);
  FUZZY->ADDFUZZYRULE(FUZZYRULE1);

  // BUILDING FUZZYRULE2------

   FUZZYRULEANTECEDENT
*IF_VELOCIDADE_MUITOBAIXA_AND_QUANTIDADE_MEDIO = NEW
FUZZYRULEANTECEDENT();
  IF_VELOCIDADE_MUITOBAIXA_AND_QUANTIDADE_MEDIO-
>JOINWITHAND(VELOCIDADE_MUITOBAIXA, QUANTIDADE_MEDIO);



  FUZZYRULECONSEQUENT *THENRISCO_INTOXICACAO2 = NEW
FUZZYRULECONSEQUENT();
  THENRISCO_INTOXICACAO2->ADDOUTPUT(RISCO_ALTO);
```

```
FUZZYRULE *FUZZYRULE2 = NEW FUZZYRULE(2,
IF_VELOCIDADE_MUITOBAIXA_AND_QUANTIDADE_MEDIO,
THENRISCO_INTOXICACAO2);
  FUZZY->ADDFUZZYRULE(FUZZYRULE2);


  // BUILDING FUZZYRULE3------

  FUZZYRULEANTECEDENT
*IF_VELOCIDADE_MUITOBAIXA_AND_QUANTIDADE_ALTO = NEW
FUZZYRULEANTECEDENT();
  IF_VELOCIDADE_MUITOBAIXA_AND_QUANTIDADE_ALTO-
>JOINWITHAND(VELOCIDADE_MUITOBAIXA, QUANTIDADE_ALTO);


  FUZZYRULECONSEQUENT *THENRISCO_INTOXICACAO3 = NEW
FUZZYRULECONSEQUENT();
  THENRISCO_INTOXICACAO3->ADDOUTPUT(RISCO_ALTO);


  FUZZYRULE *FUZZYRULE3 = NEW FUZZYRULE(3,
IF_VELOCIDADE_MUITOBAIXA_AND_QUANTIDADE_ALTO,
THENRISCO_INTOXICACAO3);
  FUZZY->ADDFUZZYRULE(FUZZYRULE3);


  // BUILDING FUZZYRULE4------

  FUZZYRULEANTECEDENT
*IF_VELOCIDADE_BAIXA_AND_QUANTIDADE_PEQUENO = NEW
FUZZYRULEANTECEDENT();
  IF_VELOCIDADE_BAIXA_AND_QUANTIDADE_PEQUENO-
>JOINWITHAND(VELOCIDADE_BAIXA, QUANTIDADE_PEQUENO);

FUZZYRULECONSEQUENT *THENRISCO_INTOXICACAO4 = NEW
FUZZYRULECONSEQUENT();
  THENRISCO_INTOXICACAO4->ADDOUTPUT(RISCO_MEDIO);
```

```
FUZZYRULE *FUZZYRULE2 = NEW FUZZYRULE(2,
IF_VELOCIDADE_MUITOBAIXA_AND_QUANTIDADE_MEDIO,
THENRISCO_INTOXICACAO2);
  FUZZY->ADDFUZZYRULE(FUZZYRULE2);


  // BUILDING FUZZYRULE3------

  FUZZYRULEANTECEDENT
*IF_VELOCIDADE_MUITOBAIXA_AND_QUANTIDADE_ALTO = NEW
FUZZYRULEANTECEDENT();
  IF_VELOCIDADE_MUITOBAIXA_AND_QUANTIDADE_ALTO-
>JOINWITHAND(VELOCIDADE_MUITOBAIXA, QUANTIDADE_ALTO);


  FUZZYRULECONSEQUENT *THENRISCO_INTOXICACAO3 = NEW
FUZZYRULECONSEQUENT();
  THENRISCO_INTOXICACAO3->ADDOUTPUT(RISCO_ALTO);


  FUZZYRULE *FUZZYRULE3 = NEW FUZZYRULE(3,
IF_VELOCIDADE_MUITOBAIXA_AND_QUANTIDADE_ALTO,
THENRISCO_INTOXICACAO3);
  FUZZY->ADDFUZZYRULE(FUZZYRULE3);


  // BUILDING FUZZYRULE4------

  FUZZYRULEANTECEDENT
*IF_VELOCIDADE_BAIXA_AND_QUANTIDADE_PEQUENO = NEW
FUZZYRULEANTECEDENT();
  IF_VELOCIDADE_BAIXA_AND_QUANTIDADE_PEQUENO-
>JOINWITHAND(VELOCIDADE_BAIXA, QUANTIDADE_PEQUENO);

FUZZYRULECONSEQUENT *THENRISCO_INTOXICACAO4 = NEW
FUZZYRULECONSEQUENT();
  THENRISCO_INTOXICACAO4->ADDOUTPUT(RISCO_MEDIO);
```

```
FUZZYRULE *FUZZYRULE4 = NEW FUZZYRULE(4,
IF_VELOCIDADE_BAIXA_AND_QUANTIDADE_PEQUENO,
THENRISCO_INTOXICACAO4);
  FUZZY->ADDFUZZYRULE(FUZZYRULE4);

  //BUILDING FUZZYRULE5------

  FUZZYRULEANTECEDENT
*IF_VELOCIDADE_BAIXA_AND_QUANTIDADE_MEDIO = NEW
FUZZYRULEANTECEDENT();
  IF_VELOCIDADE_BAIXA_AND_QUANTIDADE_MEDIO-
>JOINWITHAND(VELOCIDADE_BAIXA, QUANTIDADE_MEDIO);


  FUZZYRULECONSEQUENT *THENRISCO_INTOXICACAO5 = NEW
FUZZYRULECONSEQUENT();
  THENRISCO_INTOXICACAO5->ADDOUTPUT(RISCO_MEDIO);


  FUZZYRULE *FUZZYRULE5 = NEW FUZZYRULE(5,
IF_VELOCIDADE_BAIXA_AND_QUANTIDADE_MEDIO,
THENRISCO_INTOXICACAO5);
  FUZZY->ADDFUZZYRULE(FUZZYRULE5);


  // BUILDING FUZZYRULE6------

  FUZZYRULEANTECEDENT
*IF_VELOCIDADE_BAIXA_AND_QUANTIDADE_ALTO = NEW
FUZZYRULEANTECEDENT();
  IF_VELOCIDADE_BAIXA_AND_QUANTIDADE_ALTO-
>JOINWITHAND(VELOCIDADE_BAIXA, QUANTIDADE_ALTO);


  FUZZYRULECONSEQUENT *THENRISCO_INTOXICACAO6 = NEW
FUZZYRULECONSEQUENT();
  THENRISCO_INTOXICACAO6->ADDOUTPUT(RISCO_ALTO);


  FUZZYRULE *FUZZYRULE6 = NEW FUZZYRULE(6,
IF_VELOCIDADE_BAIXA_AND_QUANTIDADE_ALTO,
THENRISCO_INTOXICACAO6);
  FUZZY->ADDFUZZYRULE(FUZZYRULE6);
```

```
// BUILDING FUZZYRULE7------

  FUZZYRULEANTECEDENT
*IF_VELOCIDADE_NORMAL_AND_QUANTIDADE_PEQUENO = NEW
FUZZYRULEANTECEDENT();
  IF_VELOCIDADE_NORMAL_AND_QUANTIDADE_PEQUENO-
>JOINWITHAND(VELOCIDADE_NORMAL, QUANTIDADE_PEQUENO);


  FUZZYRULECONSEQUENT *THENRISCO_INTOXICACAO7 = NEW
FUZZYRULECONSEQUENT();
  THENRISCO_INTOXICACAO7->ADDOUTPUT(RISCO_BAIXO);


  FUZZYRULE *FUZZYRULE7 = NEW FUZZYRULE(7,
IF_VELOCIDADE_NORMAL_AND_QUANTIDADE_PEQUENO,
THENRISCO_INTOXICACAO7);
  FUZZY->ADDFUZZYRULE(FUZZYRULE7);


   // BUILDING FUZZYRULE8------

  FUZZYRULEANTECEDENT
*IF_VELOCIDADE_NORMAL_AND_QUANTIDADE_MEDIO = NEW
FUZZYRULEANTECEDENT();
  IF_VELOCIDADE_NORMAL_AND_QUANTIDADE_MEDIO-
>JOINWITHAND(VELOCIDADE_NORMAL, QUANTIDADE_MEDIO);


  FUZZYRULECONSEQUENT *THENRISCO_INTOXICACAO8 = NEW
FUZZYRULECONSEQUENT();
  THENRISCO_INTOXICACAO8->ADDOUTPUT(RISCO_MEDIO);


  FUZZYRULE *FUZZYRULE8 = NEW FUZZYRULE(8,
IF_VELOCIDADE_NORMAL_AND_QUANTIDADE_MEDIO,
THENRISCO_INTOXICACAO8);
  FUZZY->ADDFUZZYRULE(FUZZYRULE8);
```

```
// BUILDING FUZZYRULE9------

 FUZZYRULEANTECEDENT
*IF_VELOCIDADE_NORMAL_AND_QUANTIDADE_ALTO = NEW
FUZZYRULEANTECEDENT();
  IF_VELOCIDADE_NORMAL_AND_QUANTIDADE_ALTO-
>JOINWITHAND(VELOCIDADE_NORMAL, QUANTIDADE_ALTO);


  FUZZYRULECONSEQUENT *THENRISCO_INTOXICACAO9 = NEW
FUZZYRULECONSEQUENT();
  THENRISCO_INTOXICACAO9->ADDOUTPUT(RISCO_ALTO);


  FUZZYRULE *FUZZYRULE9 = NEW FUZZYRULE(9,
IF_VELOCIDADE_NORMAL_AND_QUANTIDADE_ALTO,
THENRISCO_INTOXICACAO9);
  FUZZY->ADDFUZZYRULE(FUZZYRULE9);

  FASTLED.ADDLEDS<NEOPIXEL, DATA_PIN>(LEDS, NUM_LEDS);
  LEDS[0] = CRGB::BLACK;
  FASTLED.SHOW();
  SERIAL.BEGIN(9600);
  BLYNK.BEGIN(AUTH, SSID, PASS);
  BLYNK.VIRTUALWRITE(V5, 3);
  PINMODE(5, INPUT_PULLUP);
  TIMER_VERIFICA_BOTAO.SETINTERVAL(500L, VERIFICA_BOTAO);
  VELOCIDADE = 0.07;
  QUANTIDADE_INICIAL= 400;
  BLYNK.VIRTUALWRITE(V9, QUANTIDADE_INICIAL);
  BLYNK.VIRTUALWRITE(V4, QUANTIDADE_INICIAL);
  BLYNK.VIRTUALWRITE(V4, DOSE);
  CALC_DOSES_ACUMULADAS();



  LCD.CLEAR(); //USE IT TO CLEAR THE LCD WIDGET
  LCD.PRINT(0, 0, "SISTEMA INICIADO"); // USE: (POSITION X: 0-15, POSITION
Y: 0-1, "MESSAGE YOU WANT TO PRINT")
  LCD.PRINT(0, 1, "----------------");

  DELAY(1000);
```

```
}

VOID LOOP()
{
BLYNK.RUN();
TIMER_VERIFICA_BOTAO.RUN();

IF (SLIDER == 1){
   STR_1 = STRING(VELOCIDADE,3);
   STR_1 = "VELOCIDADE:" + STR_1 ;
   STR_2 = STRING(QUANTIDADE_INICIAL);
   STR_2 = "QUANTIDADE:" + STR_2;
  LCD.CLEAR(); //USE IT TO CLEAR THE LCD WIDGET
  LCD.PRINT(0, 0, STR_1); // USE: (POSITION X: 0-15, POSITION Y: 0-1,
"MESSAGE YOU WANT TO PRINT")
  LCD.PRINT(0, 1, STR_2);
  SLIDER = 0;

}

}

VOID VERIFICA_BOTAO(){

    IF (DIGITALREAD(5) == LOW) {
       LEDS[0] = CRGB::BLACK;
       FASTLED.SHOW();
        DOSE++;
        STRING STR_TEMP =  STRING(DOSE, 3);

BLYNK.VIRTUALWRITE(V5, DOSE);
      // BLYNK.VIRTUALWRITE(V9, DOSES_ACUMULADAS[DOSE]);

FUZZY_CALC(DOSE,VELOCIDADE,DOSES_ACUMULADAS[DOSE]);
       }

    ELSE{
//       LEDS[0] = CRGB::BLACK;
//       FASTLED.SHOW();
//       SERIAL.PRINTLN("BAIXO");
       }

}
```

```
VOID CALC_DOSES_ACUMULADAS(){

   FOR (BYTE I = 1; I < 20; I = I + 1) {
     DOSES[I]=0;
     DOSES_ACUMULADAS[I]=0;

   }

   DOSES[0]=QUANTIDADE_INICIAL;
   //BLYNK.VIRTUALWRITE(V9, QUANTIDADE_INICIAL);
   FOR (BYTE I = 1; I < 20; I = I + 1) {
      DOSES[I] = DOSES[I-1]/2;
      //SERIAL.PRINTLN(DOSES[I-1]);
   }

   FOR (BYTE I = 1; I < 20; I = I + 1) {
      FOR (BYTE J = 0; J < I; J= J + 1) {

         DOSES_ACUMULADAS[I] = DOSES_ACUMULADAS[I] + DOSES[J];
     }
   }



}
 VOID FUZZY_CALC(INT DOSE_,FLOAT VELOCIDADE_K, FLOAT Y_QUANT) {

  INT I=0;
  //QUANTIDADE_INICIAL = Y_QUANT;
  //VELOCIDADE = VELOCIDADE_K;
  SERIAL.PRINTLN("DOSE: ");
  SERIAL.PRINTLN(DOSE_);
  SERIAL.PRINTLN("\N\N\NENTRADAS: ");
  SERIAL.PRINT("\T\T\TVELOCIDADE MEDICAMENTO: ");
  SERIAL.PRINT(VELOCIDADE_K);
  SERIAL.PRINT(", QUANTIDADE MEDICAMENTO: ");
  SERIAL.PRINT(Y_QUANT);
```

```
FUZZY->SETINPUT(1,VELOCIDADE_K);
  FUZZY->SETINPUT(2,Y_QUANT);


  FUZZY->FUZZIFY();

//  FOR(I=1;I<28;I++){
//    SERIAL.PRINT(I);
//    SERIAL.PRINT('-');
//    SERIAL.PRINTLN(FUZZY->ISFIREDRULE(I));
//  }



  SERIAL.PRINTLN("VELOCIDADE: ");
  SERIAL.PRINT("\TIDADE: MUITO BAIXA-> ");
  SERIAL.PRINT(VELOCIDADE_MUITOBAIXA->GETPERTINENCE());
  SERIAL.PRINT(", BAIXA-> ");
  SERIAL.PRINT(VELOCIDADE_BAIXA->GETPERTINENCE());
  SERIAL.PRINT(", NORMAL-> ");
  SERIAL.PRINTLN(VELOCIDADE_NORMAL->GETPERTINENCE());

  SERIAL.PRINT("\TQUANTIDADE: PEQUENA-> ");
  SERIAL.PRINT(QUANTIDADE_PEQUENO->GETPERTINENCE());
  SERIAL.PRINT(",  MEDIA-> ");
  SERIAL.PRINT(QUANTIDADE_MEDIO->GETPERTINENCE());
  SERIAL.PRINT(",  ALTA-> ");
  SERIAL.PRINTLN(QUANTIDADE_ALTO->GETPERTINENCE());
```

```
FLOAT OUTPUT1 = FUZZY->DEFUZZIFY(1);
//
//
  SERIAL.PRINTLN("SAÍDA: ");

  SERIAL.PRINT("\TRISCO ROMPIMENTO: BAIXO-> ");
  SERIAL.PRINT(RISCO_BAIXO->GETPERTINENCE());
  SERIAL.PRINT(", MEDIO-> ");
  SERIAL.PRINT(RISCO_MEDIO->GETPERTINENCE());
  SERIAL.PRINT(", ALTO-> ");
  SERIAL.PRINTLN(RISCO_ALTO->GETPERTINENCE());




  BLYNK.VIRTUALWRITE(V8, OUTPUT1);

  STR_1 = STRING(OUTPUT1,3);
  STR_1 = "RISCO:" + STR_1 ;
  STR_2 = STRING(Y_QUANT);
  STR_2 = "DOSES:" + STR_2;
  LCD.CLEAR(); //USE IT TO CLEAR THE LCD WIDGET
  LCD.PRINT(0, 0, STR_1); // USE: (POSITION X: 0-15, POSITION Y: 0-1,
"MESSAGE YOU WANT TO PRINT")
  LCD.PRINT(0, 1, STR_2);

IF( OUTPUT1 >=0 && OUTPUT1 <0.3){
     LEDS[0] = CRGB::GREEN;
      FASTLED.SHOW();
      LED1.SETCOLOR(BLYNK_GREEN);
     }
   IF( OUTPUT1 >=0.3 && OUTPUT1 <0.8){
     LEDS[0] = CRGB::YELLOW;
      FASTLED.SHOW();
      LED1.SETCOLOR(BLYNK_YELLOW);
     }
 IF( OUTPUT1 >=0.8 && OUTPUT1 <=1.0){
     LEDS[0] = CRGB::RED;
      FASTLED.SHOW();
      LED1.SETCOLOR(BLYNK_RED);
     }

  SERIAL.PRINTLN("RESULTADO: ");
  SERIAL.PRINT("\T\T\TRISCO DE INTOXICAÇÃO: ");
  SERIAL.PRINT(OUTPUT1);
```