

UNIVERSIDADE SÃO JUDAS TADEU

GIOVANNI SANTOS GUIMARAES
MATHEUS DOS SANTOS BUENO
PEDRO AUGUSTO CALIXTO FERREIRA

ROBÔ PARA COMPETIÇÃO DO BALÃO – PROJETO INTERDICIPLINAR 3A

Robô Osíris

**São Paulo
2020**

GIOVANNI SANTOS GUIMARAES 81818398

MATHEUS DOS SANTOS BUENO 819224145

PEDRO AUGUSTO CALIXTO FERREIRA 818117605

ROBÔ PARA COMPETIÇÃO DO BALÃO – PROJETO INTERDICPLINAR 3A

Robô Osíris

Projeto Integrado Multidisciplinar para a composição da nota final das disciplinas correspondentes ao período letivo 2020/1 para o curso de Projeto Interdisciplinar 3A da Universidade São Judas Tadeu.

Orientador: Prof. Dr. Paulo Cesar da Silva
Emanuel

**SÃO PAULO
2020**

RESUMO

O projeto consiste em aplicar os conhecimentos obtidos da graduação até o ponto de entrega deste trabalho. A realização deste dar-se-á pela criação de um Robô que tem como foco a competição da semana de Engenharia da Universidade São Judas Tadeu 2020-1.

Palavras-chaves: Robô, são judas, competição, aplicação.

ABSTRACT

The project consists of applying the knowledge obtained from the graduation to the point of delivery of this work. This will be done by creating a Robot that focuses on the competition for Engineering Week at São Judas Tadeu 2020-1.

Key-word: Robot, são judas, competition, application.

LISTA DE ILUSTRAÇÕES

Imagem 1 – Pinos ESP-12.....	4
Imagem 2 –Funcionamento Blynk.....	5
Imagem 3 – Estatística de emprego no Estado de São Paulo no ano 2002	13
Imagem 4 – Estatística de emprego no Estado de São Paulo no ano 2002	13
Imagem 5 – Estatística de emprego no Estado de São Paulo no ano 2002	13
Imagem 6 – Estatística de emprego no Estado de São Paulo no ano 2002	13
Imagem 7 – Estatística de emprego no Estado de São Paulo no ano 2002	13
Imagem 8 – Estatística de emprego no Estado de São Paulo no ano 2002	13
Imagem 9 – Fluxograma sobre a função blynk do controle de velocidade.....	25
Imagem 10 – Fluxograma sobre a função blynk do controle de velocidade.....	25
Imagem 11 – Fluxograma sobre a função blynk do controle de direção.....	26
Imagem 12 – Fluxograma sobre a função blynk do leitor do joystick.....	27
Imagem 13 – Fluxograma sobre a função de leitura das portas seriais.....	28
Imagem 14 – Fluxograma sobre o interpretador de entrada de dados.....	29
Imagem 15 – Fluxograma sobre a função blynk autenticador do controle.....	30
Imagem 16 – Fluxograma sobre a função blynk conversor de variáveis de entrada.....	31
Imagem 17 – Fluxograma sobre conversor de velocidade.....	32
Imagem 18 – Fluxograma sobre o leitor dos pinos virtuais.....	33
Imagem 19 – Fluxograma sobre informações no monitor serial.....	34
Imagem 20 – Fluxograma sobre o leitor de saída dos pinos de dados.....	35
Imagem 21 – Fluxograma sobre informações do monitor serial.....	36
Imagem 22 – Fluxograma sobre a função que informa a velocidade ao monitor serial.....	37
Imagem 23 – Fluxograma sobre os comandos do controle.....	39
Imagem 24 – Fluxograma sobre a função direcional do robô.....	40
Imagem 25 – Fluxograma sobre movimentos simples.....	41
Imagem 26 – Fluxograma sobre a função de erro.....	42
Imagem 27 – Fluxograma sobre a conversão da velocidade para o monitor.....	43

Imagem 28 – Fluxograma que informa a direção escolhida para o aplicativo.....	44
Imagem 29 – Fluxograma que define a velocidade do robô indo para a direita.....	44
Imagem 30 – Fluxograma que define a velocidade do robô indo para a esquerda.....	45
Imagem 31 – Fluxograma que define a velocidade do robô indo para a frente.....	45
Imagem 32 – Fluxograma que define a velocidade do robô indo para ré.....	46
Imagem 33 – Fluxograma que define a velocidade do robô parado.....	46
Imagem 34 – Fluxograma que ativa os motores.....	47
Imagem 35 – Fluxograma que emite a informação sobre a rotação dos motores.....	49
Imagem 36 – Fluxograma que faz o robô ficar parado.....	50

LISTA DE ABREVIATURAS E SÍGLAS

ABNT - Associação Brasileira de Normas Técnicas

USJT- Universidade São Judas Tadeu

SUMÁRIO

1. INTRODUÇÃO.....	1
1.1 Contextualização.....	1
1.2 - Justificativa.....	1
1.3 Guerra do Robô Balão	2
2 OBJETIVOS.....	2
2.1 - Objetivos gerais	2
2.2 Objetivos específicos	3
2.3 Materiais utilizados.....	3
3. Tecnologias Empregadas.....	3
3.1 ESP8266-12	3
3.1.1 Introdução ESP8266-12.....	3
3.1.2 Programação ESP8266-12	3
3.1.3 ESP-12.....	3
3.1.4 Pinos ESP-12.....	4
3.2 Tecnologia Blynk.....	4
3.2.1 Introdução Blynk.....	4
3.2.2 Blynk App.....	5
3.2.3 Blynk Server	5
3.2.4 Blynk Libraries	5
3.3 IFTTT.....	5
3.3.1 Introdução IFTTT	6
3.3.2 Funcionamento IFTTT	6
3.4 Ponte H.....	7

4. Simulação	9
4.1 Colocando os componentes	9
4.2 A Simulação	10
4.3 Desenvolvimento do Aplicativo Blynk.....	11
4.4 Desenvolvimento a apresentação via assistente Google.....	12
4.5 Desenvolvendo tradução do IFTTT	13
5. Lógica da Programação	14
5.2 Modelagem 3D	14
6. Desenvolvimento do projeto.....	41
7. CONSIDERAÇÕES FINAIS	41
Apêndice A	42
8. REFERÊNCIAS BIBLIOGRÁFICAS	56

1. INTRODUÇÃO

1.1 Contextualização

Na engenharia os projetos práticos são de suma importância para o desenvolvimento profissional dos estudantes, pondo em prática conhecimentos que anteriormente eram apenas teóricos, sendo eles principalmente, linguagem C (largamente utilizada em programação e aplicada no mercado). Eletrônica e utilização de Arduino, que são amplamente comuns e necessários em projetos acadêmicos, tendo seu uso difundido devido sua facilidade e praticidade no manuseio. Além de aplicar conhecimentos sobre fluxogramas e cronogramas, habilidades fundamentais para qualquer atividade e ramo de trabalho ou estudo (CAVALHEIRA; Thaís, 2016)

Com o objetivo de complementar o conhecimento teórico dos estudantes a Universidade São Judas Tadeu desenvolveu as aulas de Laboratório de Projetos Interdisciplinar (LPI), pondo em prática os conhecimentos obtidos em todas as matérias no semestre letivo.

No LPI desse semestre os alunos de engenharia deverão realizar a construção de um robô de batalha para a “Guerra de Robô Balão”, pondo em prática conhecimentos já destacados anteriormente além do demais aptidões obtidos nas aulas de Análise de Circuitos Elétricos, Sinais Sistemas e Análise de transitórios e Conversão de Energia.

1.2 - Justificativa

A tecnologia desempenha um papel extremamente importante para a humanidade, nomeadamente em Engenharia de Controle e Automação, sendo suas aplicações diversas, vindo do desenvolvimento de coisas simples como canetas até ha carros autônomos. Em todas essas tecnologias algo em comum paira sobre elas, o planejamento e o desenvolvimento. A “Guerra de Robô Balão” possibilita o desenvolvimento das duas habilidades.

O planejamento de projetos se da ao momento em que existe a criação do projeto por meios não físicos, utilizando softwares para simulações e modelagens, aplicando conceitos importantes como a programação, internet das coisas, do inglês, *Intert of things* (IOT) e análise mecânica, nessa fase existe uma maior percepção do

projeto junto a uma aproximação com ferramentas importantes. (VALLIM; Marcos Banheti Rabello, 2000)

O desenvolvimento do projeto se dá ao fim de seu planejamento, pondo em prática todas as ideias anteriormente propostas, nessa fase ideias novas poderão emergir possibilitando uma melhoria ao projeto.

Com a junção de conhecimentos em diversos aspectos a “Guerra de Robô Balão” possibilita não apenas habilidades para a criação de um projeto, como também desenvolve habilidades como trabalho em equipe e comunicação. (LM OLIVEIRA, DP Teixeira, 2012)

1.3 Guerra do Robô Balão

O projeto “Guerra de Robô Balão” vem com o objetivo de aplicar os conhecimentos adquiridos nas aulas de Laboratório de Práticas Integradas ECA3. Usando a linguagem de programação C, Arduíno, Português para fazer os fluxogramas, entre outros, a fim de criar um Robô para competição que ocorre dentro da Universidade São Judas Tadeu (USJT), a peleja tem como fruto a aplicação as boas práticas de engenharia e a integração entre alunos. No maquinário será acoplado um balão, cabendo aos outros competidores estourá-lo, em contrapartida, teremos que estourar o balão dos adversários. (NAGAI ; Walter Aoiama, IZEKI; Claudia Akemi 2013)

2 OBJETIVOS

2.1 - Objetivos gerais

Com a importância dos projetos para os cursos de engenharia, este trabalho tem como objetivo desenvolver um robô de batalha utilizando tecnologias importantes para seu planejamento e sua execução. O projeto deverá ser planejado, simulado, modelado e analisado antes de seu desenvolvimento. Será necessário utilizar plataformas para o controle do robô assim como a programação para a efetividade e o funcionamento do projeto. Ao fim da implementação do projeto o robô de batalha deverá participar da competição “Guerra de Robô Balão” onde o mesmo tem o objetivo de estourar um balão acima de um robô oponente. Com isso, o aluno terá aplicado todos os conhecimentos fundamentais para formação acadêmica dispostos no semestre em questão.

2.2 Objetivos específicos

O projeto Osíris tem como objetivo específicos:

2.2.1 – Desenvolver um robô de batalha.

2.2.2 – Utilizar a plataforma Arduino para a movimentação do robô

2.2.3 – Desenvolver junto ao ESP8266 uma comunicação Wi-Fi entre o robô e a internet

2.2.5 – Desenvolver um aplicativo capaz de se comunicar com o ESP8266 que por sua vez irá mandar informações ao robô

2.2.5 – Realizar simulações na plataforma Proteus para a implementação do robô. A simulação irá verificar se a programação no Arduino e no ESP8266 estão funcionando como necessário.

2.2.6 – Modelar o robô nas plataformas Autocad e Inventor para verificar a mecânica e a organização em seu interior.

2.3 Materiais utilizados

1 ESP8266-12 – Comunicação IOT.	1 Power Bank Carregador de Celular.
1 L298N – Ponte H.	6 Jumpers Macho-Femia.
1 Protoboard.	3 Jumpers Macho-Macho.
2 Motores DC.	6 Jumpers sem junção.
2 Rodinhas – 70mm de diâmetro.	1 Desodorante Roll On.
4 Pilhas AAA.	
2 Suportes para pilhas AAA.	

3. TECNOLOGIAS EMPREGADAS

3.1 ESP8266-12

3.1.1 Introdução ESP8266-12

O ESP8266 é um microcontrolador que possui uma comunicação Wi-Fi, essa característica o faz ser largamente utilizado com um módulo Wi-Fi para outros microcontroladores (Tais como o Arduino). Capaz de hospedar paginas web e aplicativos, ser programado por diversas linguagens de programação tais como Lua e a IDE do Arduino o ESP8266 se torna uma ótima placa para se iniciar o aprendizado em soluções IOT(Internet Of Things ou Internet das Coisas)._(Oliveira.R.R, USO DO MICROCONTROLADOR ESP8266 PARA AUTOMAÇÃO RESIDENCIAL, Janeiro de 2017)

3.1.2 Programação ESP8266-12

Capaz de ser programado pela IDE do Arduino o ESP8266 se torna cada vez mais completo com a interação de uma grande quantidade de pessoas tanto com o Arduino quanto com a família ESP, sua velocidade e seu processamento tornam fácil a junção de sensores com sua placa, além de possibilitar uma junção com outras plataformas e microcontroladores.

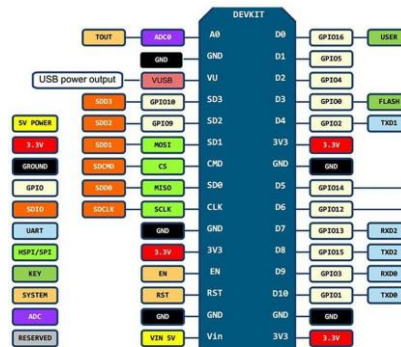
(Internet das Coisas com ESP8266, Arduino e Raspberry Pi, Junho de 2017)

3.1.3 ESP-12

Com uma quantidade mais elevada de pinos, o ESP-12 possibilita uma maior quantidade de conexões desse modo, trabalhos que necessitariam de uma junção entre ESP + microcontrolador qualquer podem ser feitas exclusivamente sozinhas, sem auxílio de mais microcontroladores.

3.1.4 Pinos ESP-12

Imagem 1 – Pinos ESP-12



Fonte: Mercado livre, 2020.

3.2 Tecnologia Blynk

3.2.1 Introdução Blynk

“Ao considerarmos o universo maker, é quase impossível não citarmos os projetos baseados em Arduino. O surgimento de novos dispositivos que também podem ser programados em Arduino, bem como a utilização de shields (placas que agregam funções aos dispositivos Arduino) ampliaram as possibilidades de projetos que podem ser desenvolvidos em Arduino. Paralelamente, o surgimento de serviços conectados à internet e o conceito de IoT (Internet Of Things) aumentaram a demanda por dispositivos que possuam conectividade e, assim, proporcionem o envio de dados à internet e o controle remoto destes dispositivos.” (Serrano.T.M, Embarcados, Maio de 2018)

Com isso em mente apresentamos o Blynk, um serviço baseado em um aplicativo interativo que por sua vez permite o controle de hardware programáveis de forma remota assim como a reportagem de dados do hardware para o aplicativo. Com o Blynk surge a possibilidade da construção de interfaces gráficas de controle de maneira rápida e intuitiva. O Blynk, possibilita a junção de mais de 400 placas de desenvolvimento baseadas em Arduino. (Mas não somente em arduino). *(Serrano.T.M, Embarcados, Maio de 2018)*

3.2.2 Blynk App

“O App Blynk é um aplicativo disponível para Android e iOS que permite ao usuário criar aplicações que interagem com o hardware. Através de um espaço próprio para cada projeto, o usuário pode inserir Widgets que implementam funções de controle (como botões, sliders e chaves), notificação e leitura de dados do hardware (exibindo em displays, gráficos e mapas).” (Serrano.T.M, Embarcados, Maio de 2018)

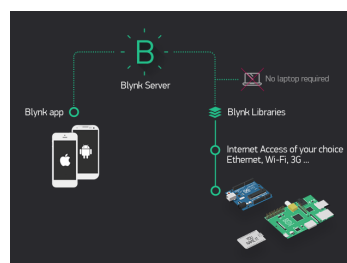
3.2.3 Blynk Server

“Toda comunicação entre o aplicativo e o hardware do usuário se dá através da cloud Blynk. O servidor é responsável por transmitir os dados ao hardware, armazenar estados do aplicativo e do hardware e também armazenar dados de sensores lidos pelo hardware mesmo se o aplicativo estiver fechado. Vale ressaltar que os dados armazenados no server Blynk podem ser acessados externamente através de uma API HTTP, o que abre a possibilidade de utilizar o Blynk para armazenar dados gerados periodicamente como dados de sensores de temperatura, por exemplo.” (Serrano.T.M, Embarcados, Maio de 2018)

3.2.4 Blynk Libraries

“Finalmente, do lado do hardware temos as bibliotecas Blynk para diversas plataformas de desenvolvimento. Essa biblioteca é responsável por gerir toda a conexão do hardware com o servidor Blynk e gerir as requisições de entrada e saída de dados e comandos. A forma mais fácil e rápida é utilizá-la como bibliotecas Arduino, no entanto, é possível obter versões da biblioteca para Linux (e Raspberry Pi!), Python, Lua, entre outras.” (Serrano.T.M, Embarcados, Maio de 2018).

Imagem 2: Funcionamento do Blynk.



Fonte: Serrano.T.M, Embarcados, Maio de 2018

3.3 IFTTT

3.3.1 Introdução IFTTT

Tibbets e Jesse Tane co-fundaram a IFTTT em 2010 e lançaram oficialmente o serviço em 2011. Sediada em São Francisco. Em novembro de 2016, a IFTTT aprimorou suas receitas que conectavam dois dispositivos, aplicativos ou serviços, transformando-os em applets, capazes de conectar vários dispositivos, aplicativos ou serviços.

O IFTTT deriva seu nome da instrução condicional de programação "se isso, então aquilo". O que a empresa fornece é uma plataforma de software que conecta aplicativos, dispositivos e serviços de diferentes desenvolvedores para acionar uma ou mais automações envolvendo esses aplicativos, dispositivos e serviços (MARTIN et al., 2019).

Como exemplo de automações que você pode executar com o IFTTT, podemos citar:

- Quando você faz uma chamada em seu telefone um registro dessa chamada é adicionado a uma planilha do Google.
- O IFTTT pode acender uma luz ao detectar um movimento em alguma sala (com dispositivos compatíveis associados).

Segundo Tibbets (2019), a IFTTT tem como objetivo conectar todos os aplicativos, dispositivos ou serviços, "Ajudamos todos os produtos e serviços a trabalharem bem juntos, de uma maneira que lhe traga confiança e os ajude a criar interações ricas em seus ecossistemas."

3.3.2 Funcionamento IFTTT

As automações são realizadas por meio de applets que são como macros que conectam vários aplicativos para executar tarefas automatizadas. Você pode ativar ou desativar um applet usando o site da IFTTT ou aplicativos móveis (e / ou os widgets IFTTT dos aplicativos móveis). Você também pode criar seus próprios applets ou fazer variações dos existentes por meio da interface direta e amigável do IFTTT (MARTIN et al., 2019).

Há uma variedade enorme de applets disponíveis. Suas coleções agrupam mini aplicativos para diferentes plataformas como iOS, Android e assistentes de voz, e mostram tudo, desde mini aplicativos para notícias e serviços meteorológicos até

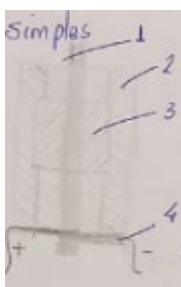
automação residencial. Os próprios usuários podem criar seus applets combinando vários "serviços" de aplicativos e definindo parâmetros de acionamento.

3.4 Ponte H

Para o movimento do robô, foram utilizados dois motores CC de 12V de tensão (corrente contínua, a corrente contínua é um tipo de fonte de energia, que não oscila com o tempo, se mantém estática), para acontecer o movimento para frente os dois motores giram ao mesmo tempo no mesmo sentido e com intensidade igual, e vice versa para o movimento para trás, para o movimento para esquerda o motor da esquerda para ou reduz sua velocidade, para poder o robô girar no eixo do motor. Para a direita vice e versa. Deste modo conseguimos um movimento livre e fácil de controlar

Os motores funcionam de uma maneira simples, quando são alimentados por uma fonte de energia temos o motor girando, quando os polos (que são como extremidades, aonde colocamos a entrada de energia) são invertidos o motor gira no sentido diferente, isso pois o sentido da corrente muda, então, o sentido do campo magnético também altera, temos o sentido de linhas magnéticas que interagem com o campo externo no motor, assim tendo o giro. Funciona da seguinte forma:

O eixo do motor ([1]Uma haste metálica) esta junta de uma bobina ([3] um enrolamento de fios que gera um campo magnético quando energizado) o que faz um rotor, ou seja, um rotor é o conjunto de uma haste metálica com uma bobina, é também a parte que dá o giro do motor. Como já havia dito, tem que existir um conflito de campos magnéticos, então temos ímãs em volta da bobina [2] que foram montados desta forma cilíndrica para que o giro do motor seja perfeito.

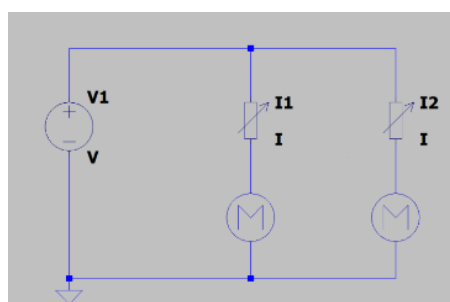


Quando a bobina é energizada [4] ela gera um campo magnético induzido, quando existe uma parte metálica (como a haste) temos uma magnetização da haste, ou seja temos um campo magnético completo, E este campo magnético está sujeito a um torque¹ magnético, que tende o material a girar em algum eixo. Como o rotor está magnetizado, ele sofre influência de campos magnéticos, em volta dele existe ímãs o que faz o rotor ficar estático, mas, o torque ainda tende o rotor a girar, o que encontra apenas um jeito, girar no mesmo eixo, aonde temos o giro de um motor (HIGASHI, EDUARDO 2006)

Torque¹: É uma força que todo corpo sofre, uma tendência de fazer o corpo girar. Uma porta por exemplo, quando empurramos uma porta, na parte de maçaneta fica fácil de abri-la. Quando empurramos perto da dobradiça fica difícil, pois está perto do eixo de rotação. (HALLIDAY, David, 2012)

O que usamos para o controle dos motores foi uma Ponte H, uma componente eletrônica que tem um certo tipo de chaveamento para dar movimentos sincronizados a motores. Os motores de corrente contínua que usamos apesar de serem simples, para fazer dois motores funcionarem em sincronia, ou fazer movimentos alternados fica mais complicado por meios convencionais, por exemplo uma ligação em paralelo ou em série.

Imagem 2:

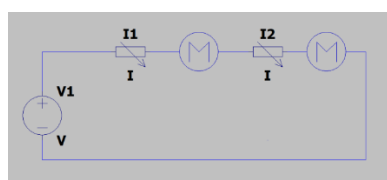


Fonte: Acervo pessoal

Esta ligação em paralelo, temos uma corrente dividida, então se eu aumentar a potência de um motor, o outro motor terá uma corrente reduzida e vice e versa, temos isso comprovado pelas Leis de Kirchoff, a lei das malhas e dos nós, quando temos um sistema em paralelo com mais de um nó (o nó é uma junção de fios, ou seja quando a partir de um ponto sai 3 ou mais fios), a tensão dada aos componentes é equivalente a da fonte mas a corrente será dividida

(WENTWORTH, M., S. 2006)

Imagem 4:



Fonte: Acervo pessoal

Temos também a ligação em série, agora com nenhum nó então temos a mesma corrente passando pelo sistema, então se aumentar a potencia de um motor

vai

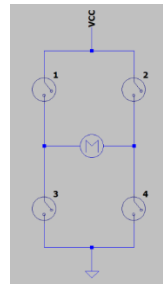
subir

dos

dois

Como a corrente que controla a intensidade do motor, escolhemos a Ponte H para poder fazer a distribuição de corrente e também manter os motores sincronizados.

Imagem 5:



Fonte: Acervo pessoal

O nome Ponte H é dado, por causa da sua arquitetura compacta, que lembra a letra H. Como vimos na seção de motores, quando invertemos os polos de ligação o sentido de rotação alterna, temos os chaveamentos que proporcionam esta mudança de uma forma mais fácil, podemos ver que para girar em um sentido fechamos as chaves 1 e 4, para outro sentido abrimos 1 e 4, e fechamos 2 e 3. (Patsko, Luís Fernando, 2006)

Como usamos dois motores serão necessárias 2 pontes H ligadas em paralelo para manter o funcionamento com a mesma tensão e distribuir a corrente. A diferença é que agora não é mais distribuído no nó, e sim por componente, mesmo que tenha uma estrutura complexa continua sendo um componente só, então a distribuição de energia é diferente e ela apenas recebe a corrente que precisa sem interferir no nó

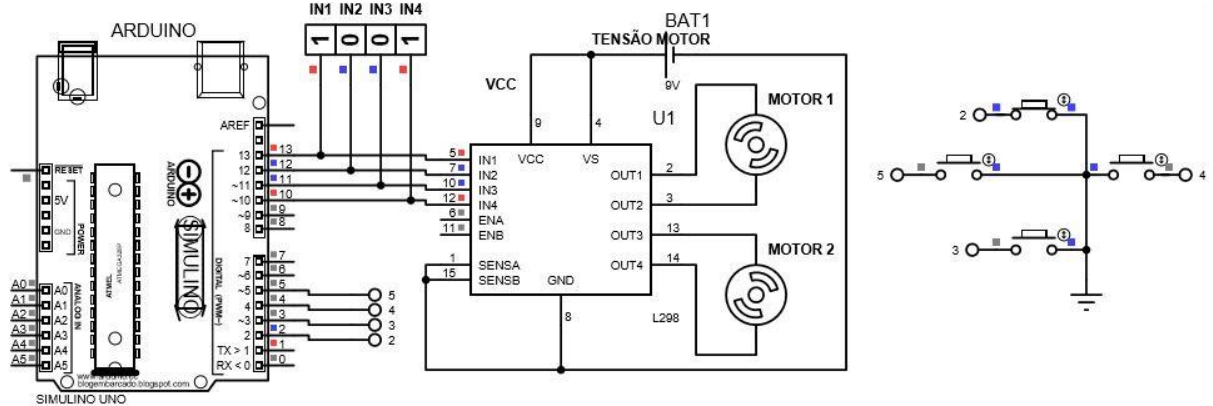
4. SIMULAÇÃO

4.1 Colocando os componentes

Para a realização das simulações do Projeto Osiris utilizamos a plataforma proteus. Entretanto a mesma não possibilitava a simulação do ESP8266, desse modo efetuamos as simulações com um Arduino Uno. Após locarmos o microcontrolador

inserir na simulação o módulo ponte H L298N. Para simularmos a comunicação entre o celular e o ESP-12 utilizamos 4 botões como mostrado na figura abaixo.

Imagem 6:



Fonte: Acervo pessoal

Para realizarmos a simulação devemos entender a conversão dos botões e do Arduino para o aplicativo Blynk e o ESP-12. Os botões que estão conectados aos pinos 2,3,4 e 5 correspondem aos botões virtuais do aplicativo, já os botões de saída 13,12,11 e 10 do Arduino correspondem respectivamente os botões D2,D5,D6 e D7 da placa ESP-12.

4.2 A Simulação

A simulação corresponde apenas aos movimentos do robô, sua comunicação não está totalmente englobada na simulação, e sim o entendimento da rotação dos motores. A figura abaixo mostra a lógica que ocorre ao apertar cada um dos botões.

Imagem 7:

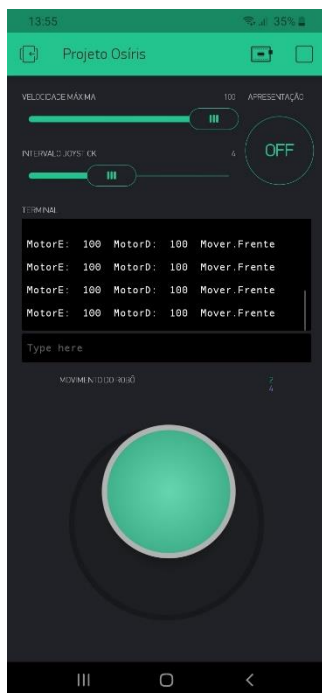


Fonte: Acervo pessoal

Após finalizar a simulação adquirimos assim a base do funcionamento do robô podendo identificar sua movimentação. Para a continuidade do projeto iniciamos a construção do aplicativo Blynk.

4.3 Desenvolvimento do Aplicativo Blynk

Após instalar o aplicativo e realizar o cadastro iniciamos um novo projeto denominado “projeto robô” feito esse processo começamos a montagem dos botões que seriam utilizados. Adicionamos ao projeto 3 Sliders, 1 Terminal e 1 Joystick como mostrado na figura abaixo.



O Slider “VELOCIDADE MÁXIMA” Indica a velocidade máxima que os motores E e D terão no funcionamento do robô. Seu pino virtual é o V2

O Slider “INTERVALO JOYSTICK” corresponde a sensibilidade do Joystick para a velocidade dos motores. Seu pino virtual é o V4

O TERMINAL é a parte responsável do projeto de informar as velocidades do motor assim como sua direção. Seu pino virtual é o V5

O Joystick se torna capaz da realização dos movimentos do robô. Fazendo o mesmo ir para a direção informada pelo Joystick. Seus pinos virtuais são o V0 e V1

O APRESENTAÇÃO é responsável pela apresentação via assistente Google. Seu pino virtual é o V7

Imagem de autoria própria

4.4 Desenvolvimento a apresentação via assistente Google.

Desenvolvendo IFTTT

Para realizar a junção entre o blynk e o assistente Google será necessário instalar o aplicativo IFTTT que permite a conexão.

Após instalado e se cadastrado no aplicativo será necessário apertar em “create” para criar uma função que conecte ambos os aplicativos, posteriormente click em “If This” e pesquise “Google Assistant” e click em “say a simple phrase” e preencha as informações como mostrado na imagem abaixo.

Feito o preenchimento da interação do assistente Google será necessário criar a resposta do assistente Google, para isso click em “Then That” e pesquise “Webhooks”, click em “make a we request” e preencha como mostrado na figura abaixo

The screenshot shows the IFTTT app interface for Google Assistant. It displays two triggers on the left and one action on the right.

Trigger 1: Say a simple phrase

- Description: This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase you choose. For example, say "Ok Google, I'm running late" to text a family member that you're on your way home.
- What do you want to say?: presenting osiris
- What's another way to say it? (optional):
- And another way? (optional):
- What do you want the Assistant to say in response?: O robô Osiris está sendo apresentado
- Language: English

Trigger 2: Make a web request

- Description: This action will make a web request to a publicly accessible URL. NOTE: Requests may be rate limited.
- URL: http://45.55.96.146 /XKUKVccdKCRXqfC9oljYAYStrFgdRP89/update/V7
- Method: PUT
- Content Type: application/json
- Body: ["1"]

A **Save** button is visible at the bottom right of the interface.

O espaço “What do you want to say?”

corresponde à fala

necessária para ativar a

ação via Google

Assistente, observe que a fala deverá ser em inglês.

O espaço “What do you want the assistant to say in response”

representa a resposta

dada pelo Assistente

Google. Está em

português pois haverá

uma tradução no meio do caminho.

O espaço “URL”

corresponde ao caminho

que o Google Assistente deverá tomar indo no IP e token do aplicativo Blynk e entrando no botão virtual 7 (V7)

O espaço “Body” corresponde a informação dada ao botão V7, no caso a imagem está ligando.

Imagem de autoria própria

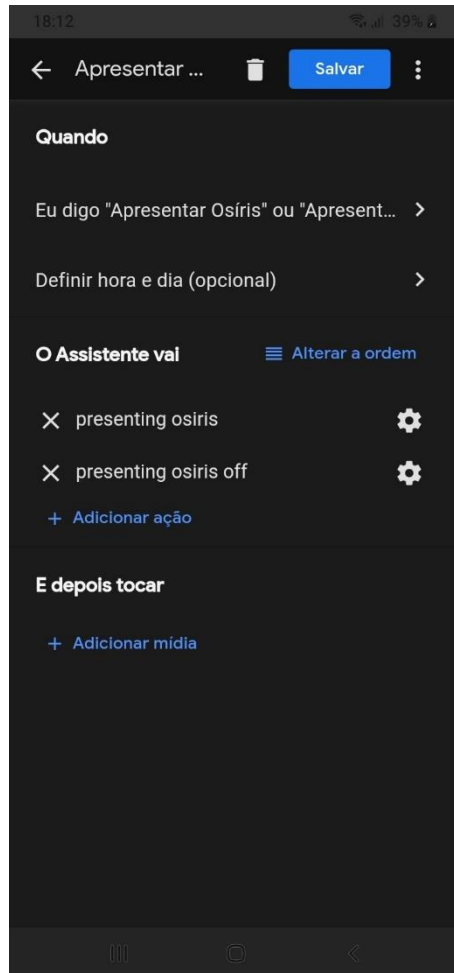
Imagem de autoria própria

O mesmo procedimento será feito para realizar o desligamento do mesmo botão, entretanto o usuário deverá falar “presenting osiris off”, o Assistente não responderá nada e a informação dada ao botão V7 será de desligamento.

4.5 Desenvolvendo tradução do IFTTT

Para que o IFTTT entenda palavras proferidas em Português BR será necessário o auxílio da ferramenta rotinas do próprio Google, para isso é necessário entrar no aplicativo “Google”, clicar em “Mais”, entre em configurações, selecione a

opção “Google Assistente”, vá para a aba “Assistente”, click em “Rotinas” e por fim selecione “+ Adicionar uma rotina”. Feito todos esses processos preencha as informações como mostrado abaixo.



No espaço “Quando” será necessário colocar a fala que o Assistente Google irá entender, nesse caso foram postas duas frases, são elas “Apresentar Osiris” e “Apresentar Robô”

No espaço “O Assistente vai” será colocado a resposta dada ao comando, sendo ela a frase em inglês posta no IFTTT, desse modo, ao ser dita a frase em português, a rotina manda ao IFTTT uma mensagem em inglês sendo elas “presenting osiris” para ativar o botão V7 e posteriormente “presenting osiris off” para desativar o botão V7.

Imagem de autoria própria

5. LÓGICA DA PROGRAMAÇÃO

5.2 Modelagem 3D

Para a realização da modelagem 3D será necessário baixar alguns blocos correspondentes aos componentes utilizados na construção do projeto, são eles, um blocos do ESP-12, protoboard, L298N, Motores DC, Rodinhas, pilhas, suporte de pinhas e um desodorante Roll On. Após baixas o blocos utilizaremos o software

AutoCad 2018 para a realização da modelagem, onde será necessário apenas modelar a Power Bank, o suporte onde serão locados os demais componentes e a parte superior do robô. A seguir uma figura da modelagem do robô.

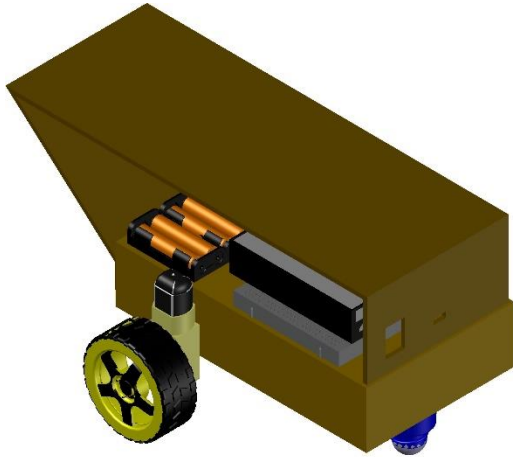


Imagem de autoria própria

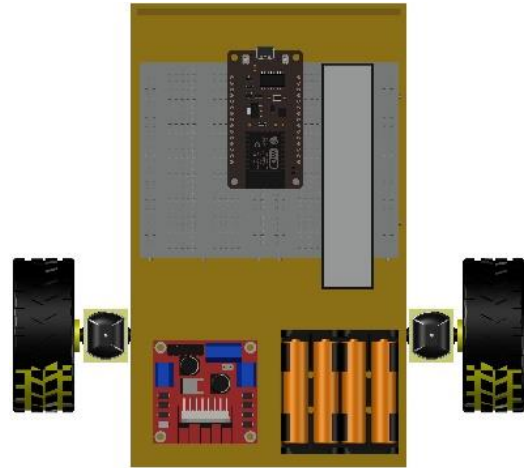
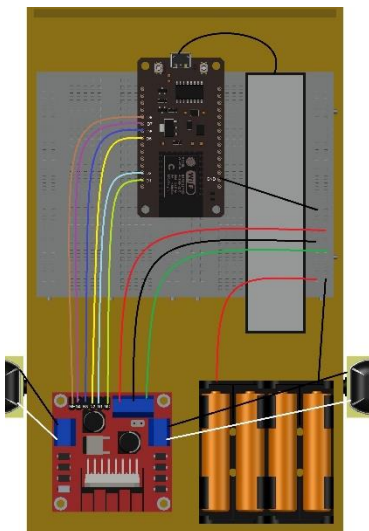


Imagem de autoria própria

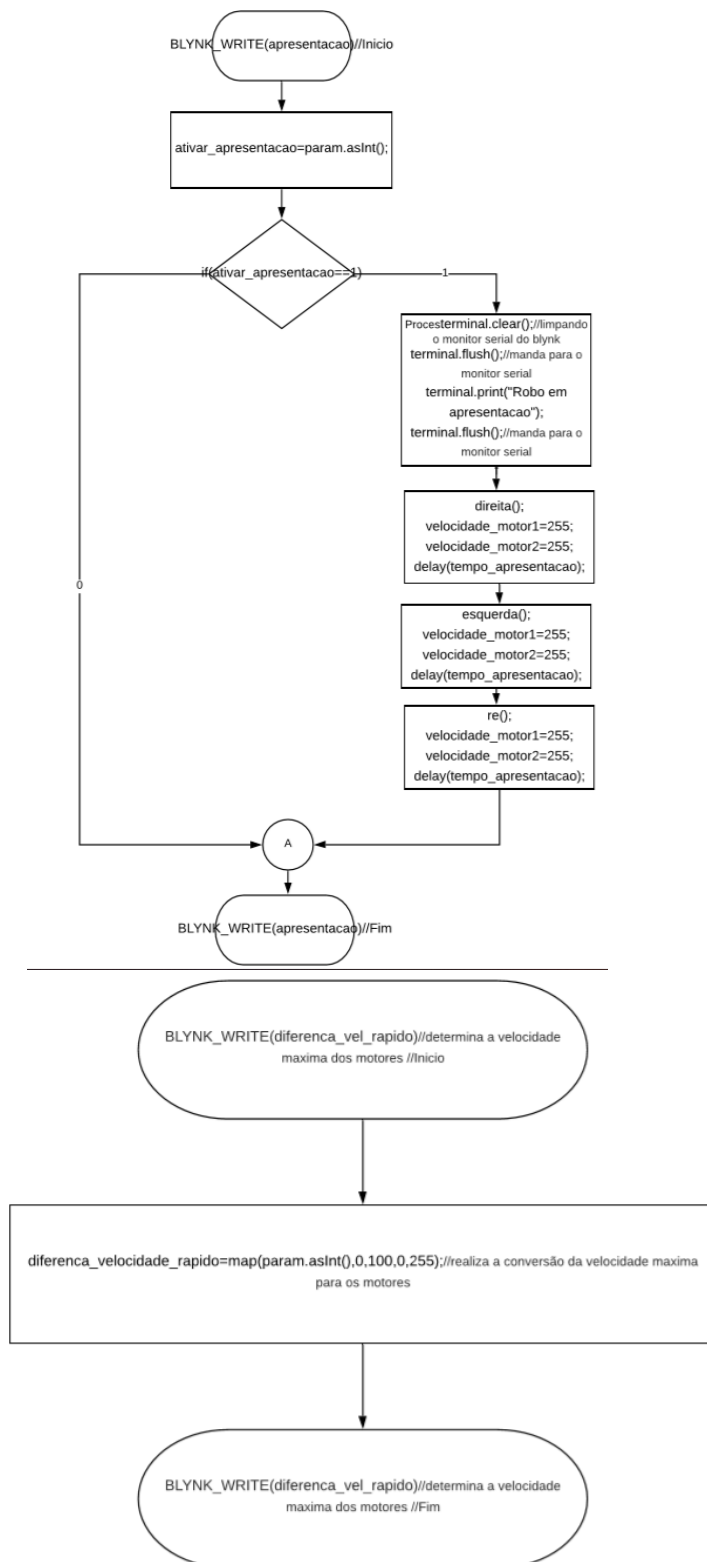
Observe que o desodorante Roll On será responsável de promover a liberdade do robô para as curvas pois seu eixo é totalmente livre, podendo girar em qualquer direção. Podemos também notar que as 4 pilhas estão em série.

As conexões de cada componente foram feitas como mostrado na imagem a seguir



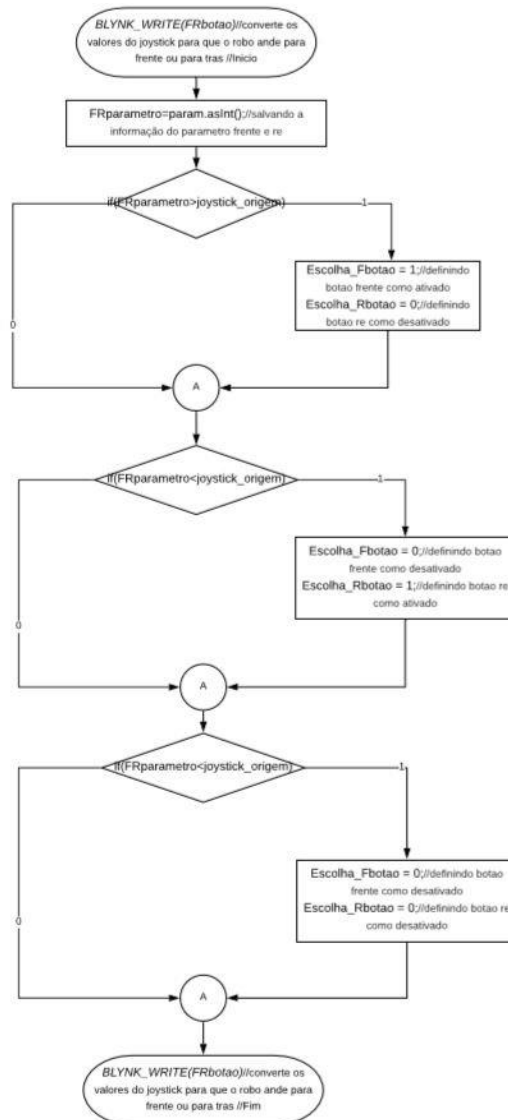
Observe que o GND do ESP8266 está junto ao GND do L298N e das pilhas, isso se dá para que todos os componentes estejam no mesmo referencial. As conexões da placa do ESP8266 e do L298N estão mais explicadas no Apêndice A correspondente à programação.

Fluxograma1: Controle de velocidade



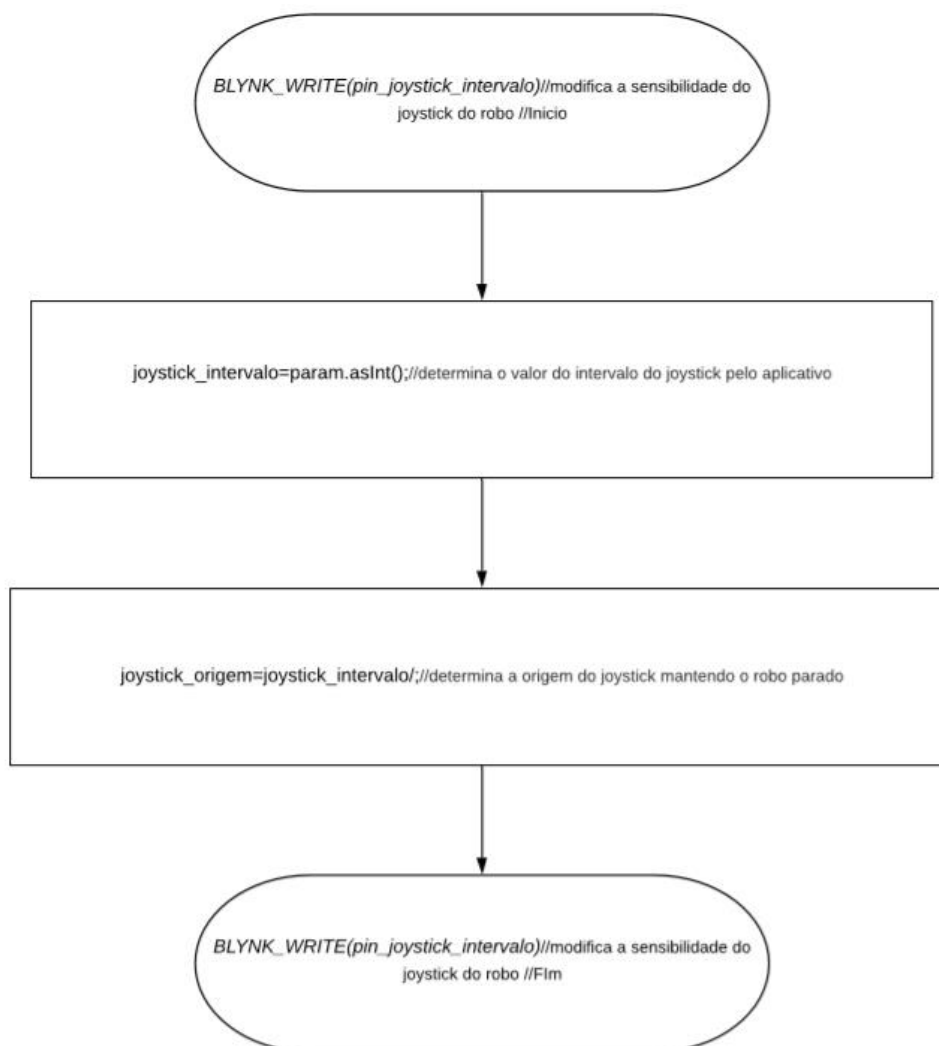
Fonte: Acervo pessoal

Fluxograma 3: Blynk para controle de direção



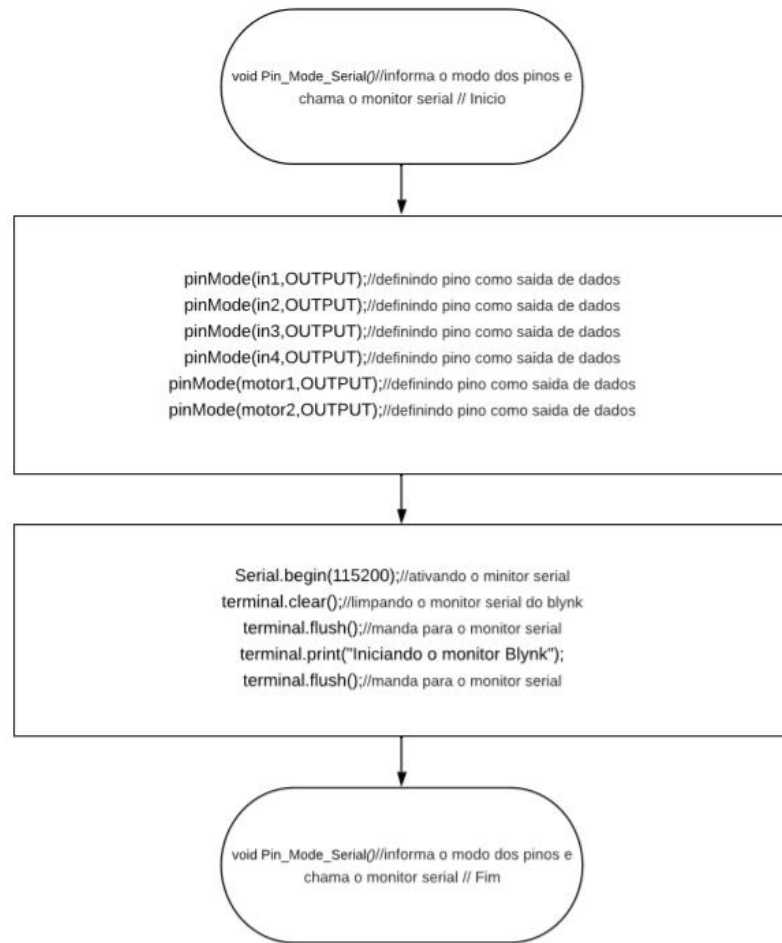
Fonte: Acervo pessoal

Fluxograma 4: Leitor do joystick



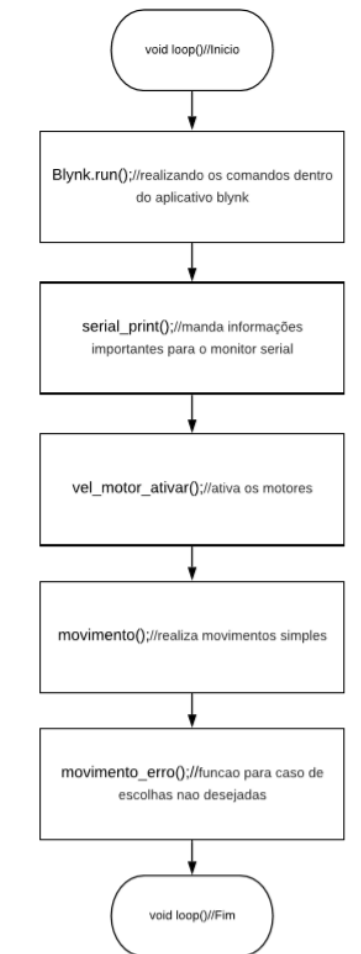
Fonte: Acervo pessoal

Fluxograma 5: Interpretador das portas seriais



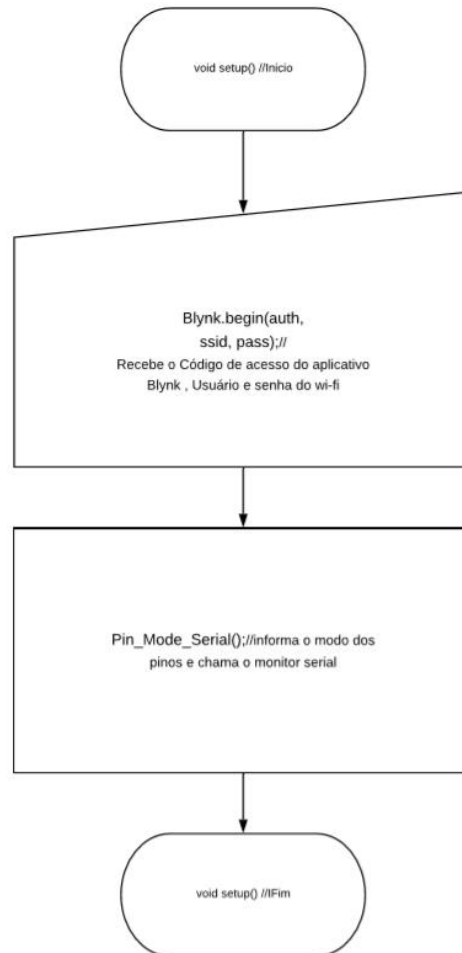
Fonte: Acervo pessoal

Fluxograma 6: Interpretador de entrada de dados



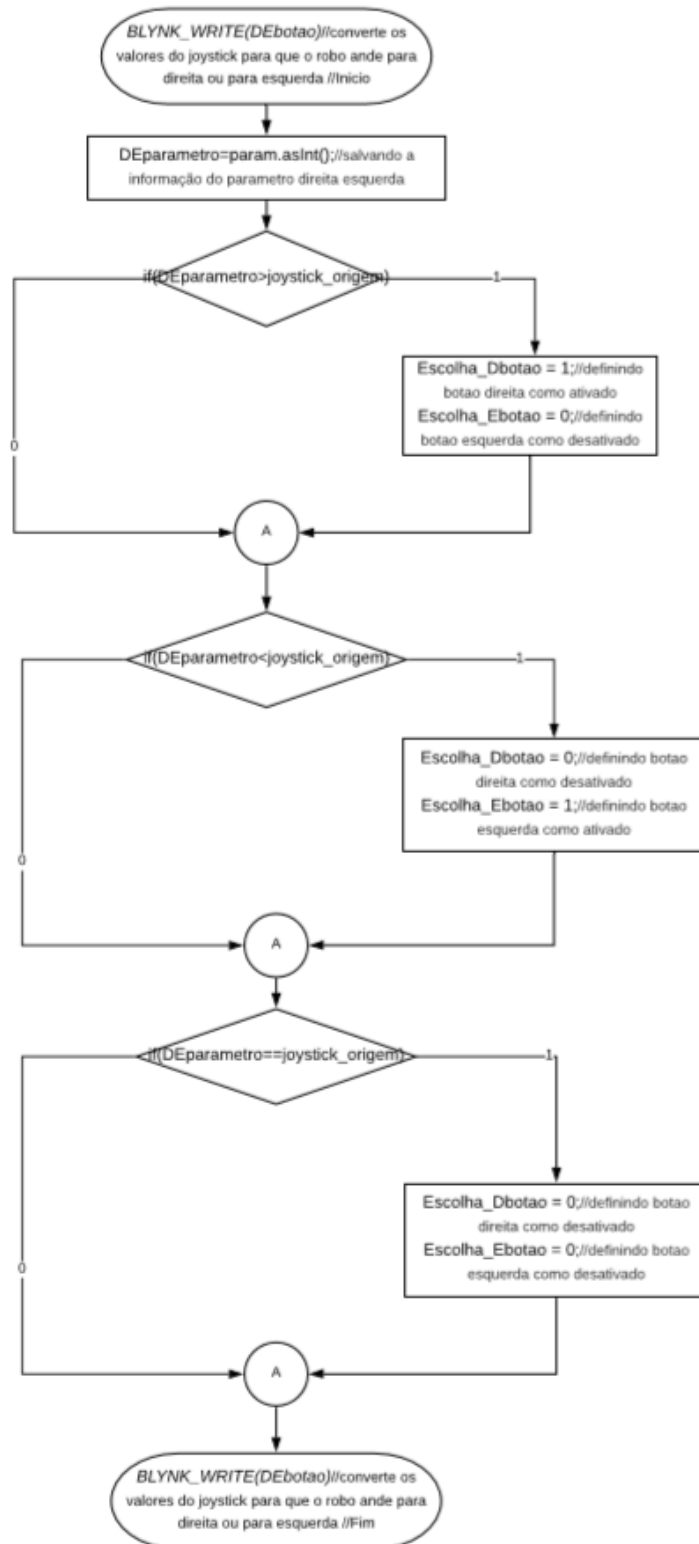
Fonte: Acervo pessoal

Fluxograma 7: Autenticador do controle



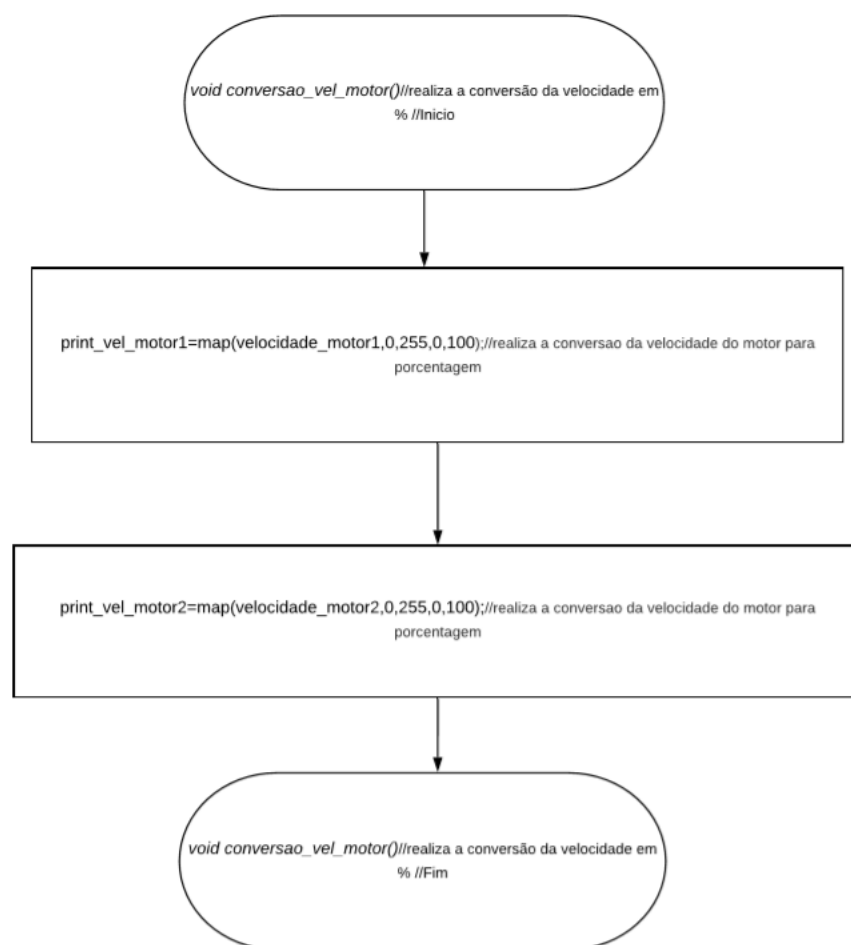
Fonte: Acervo pessoal

Fluxograma 8: Conversor de variáveis de entrada



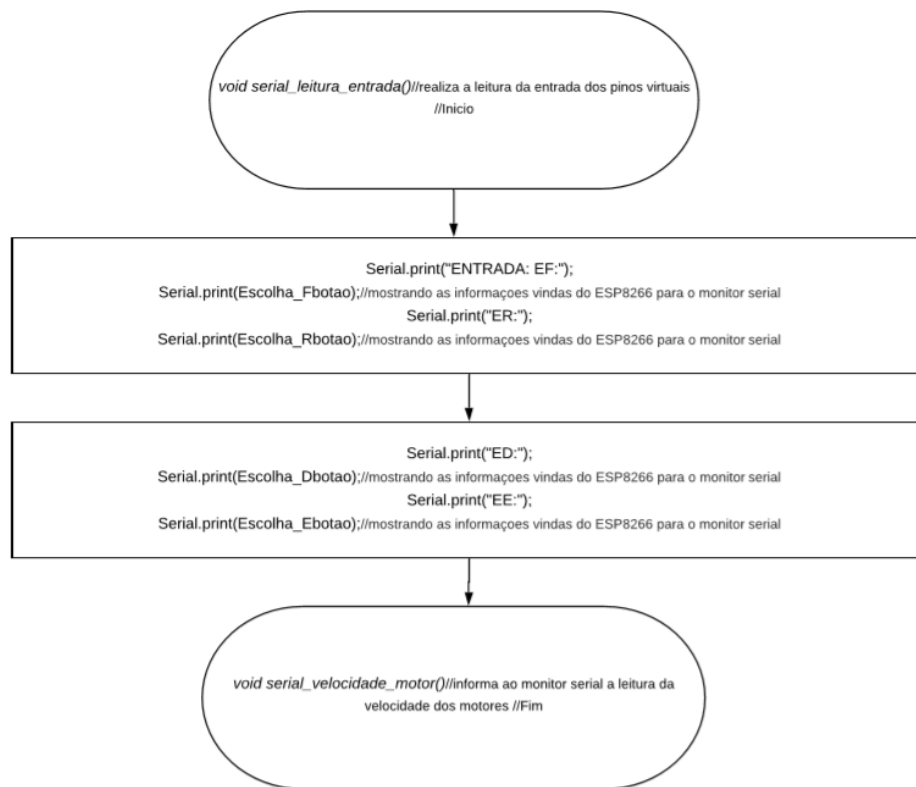
Fonte: Acervo pessoal

Fluxograma 9: Conversor de velocidade



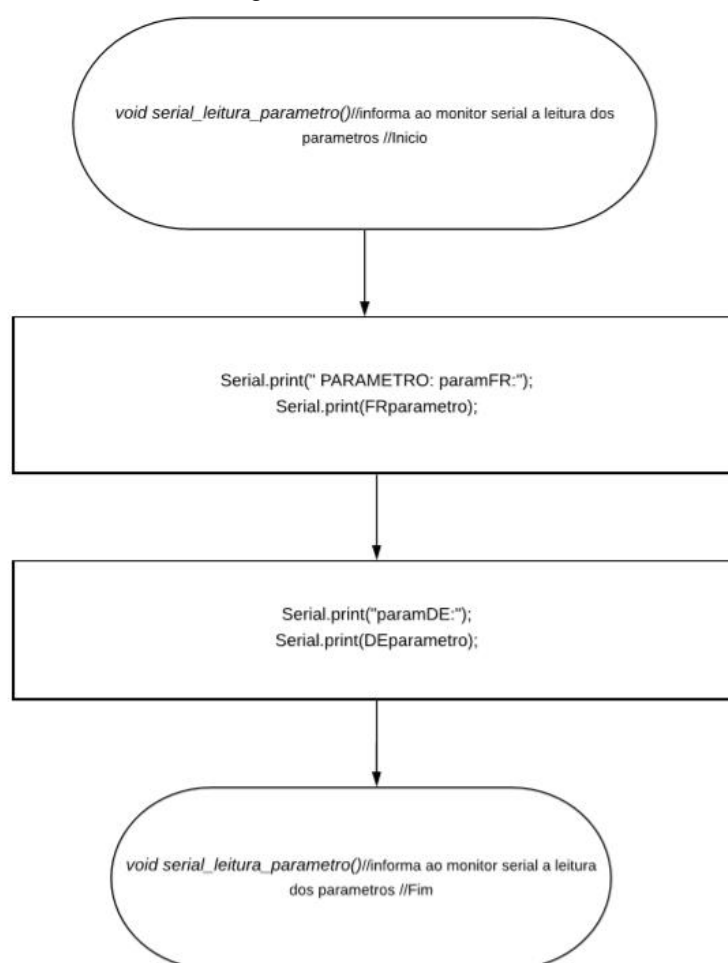
Fonte: Acervo pessoal

Fluxograma 10: Leitor dos pinos virtuais



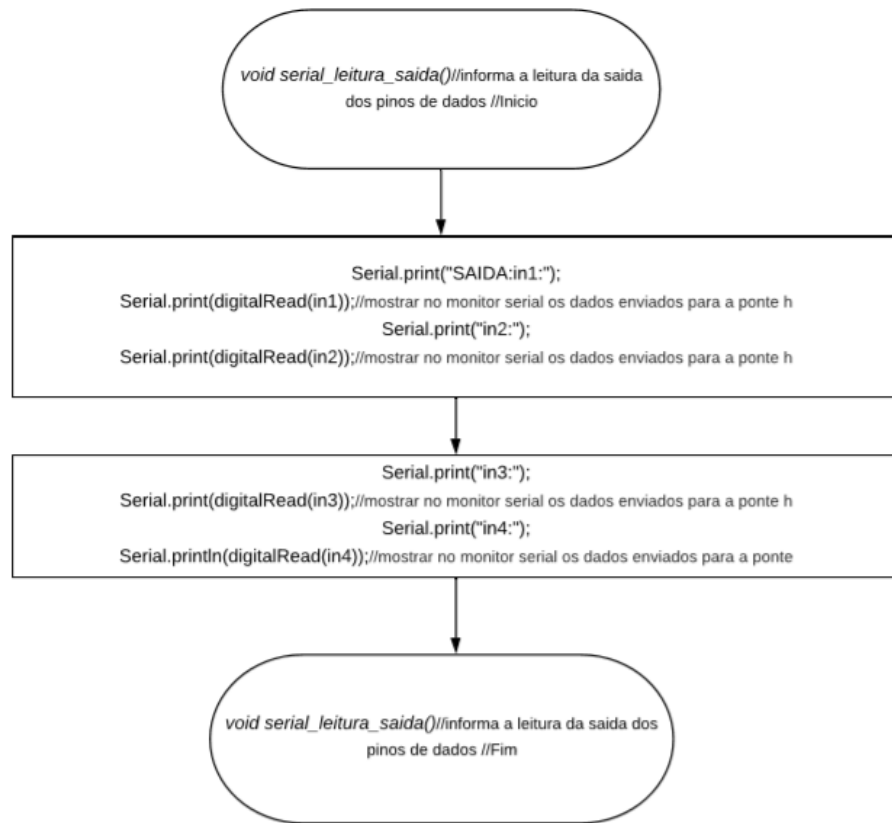
Fonte: Acervo pessoal

Fluxograma 11: Informa o monitor serial



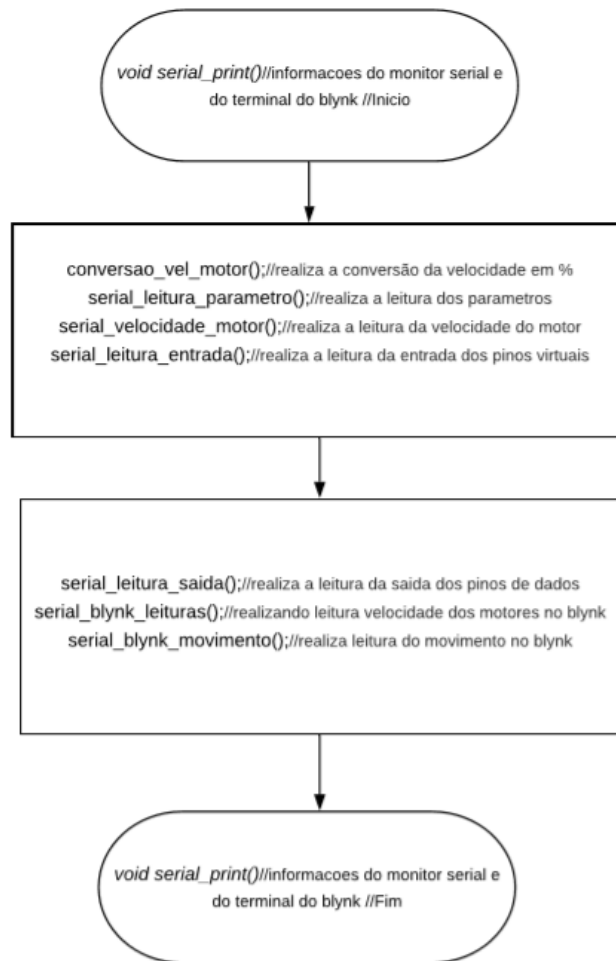
Fonte: Acervo pessoal

Fluxograma 12: Leitor de saída dos pinos de dados



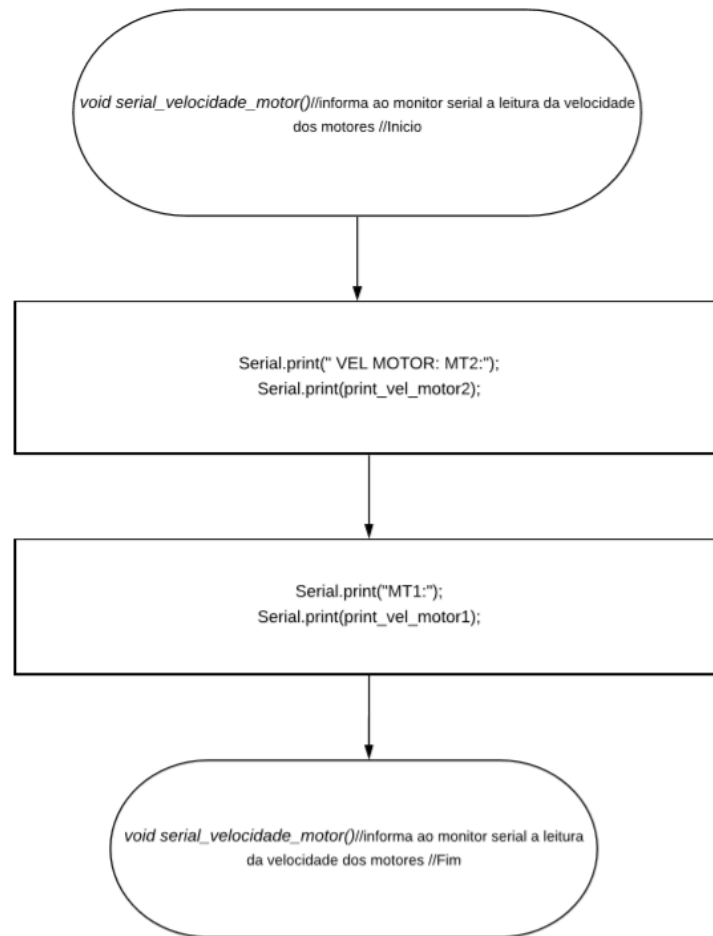
Fonte: Acervo pessoal

Fluxograma 13: Informações do monitor serial



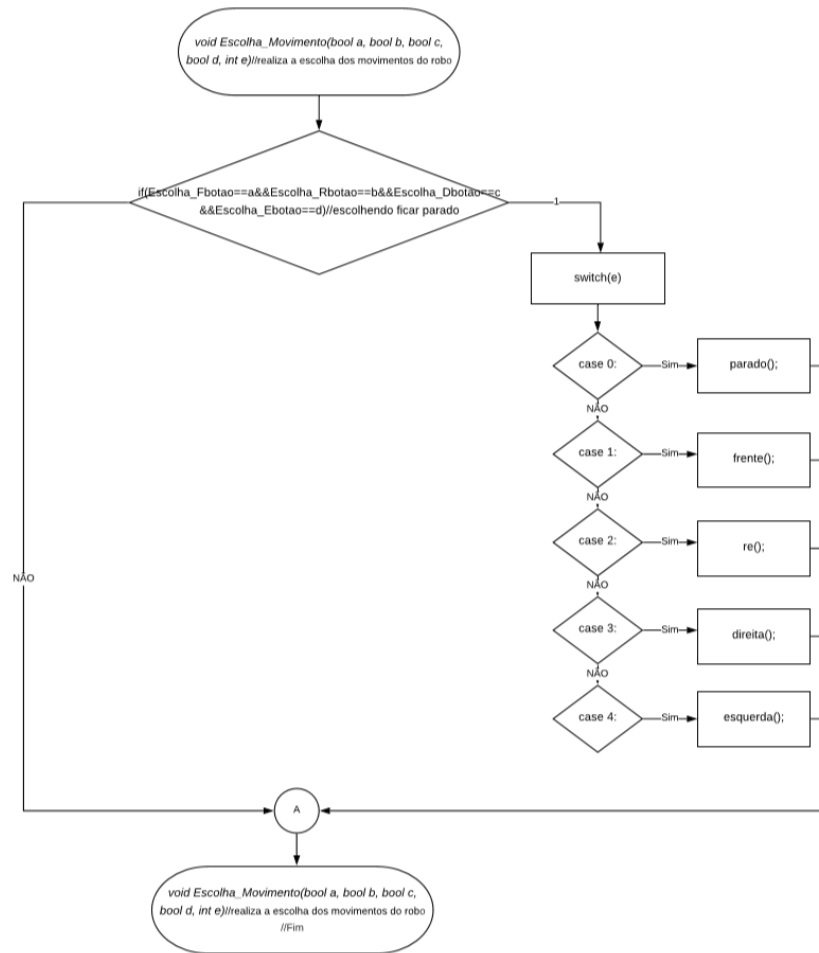
Fonte: Acervo pessoal

Fluxograma 14: Informa a velocidade para o monitor serial



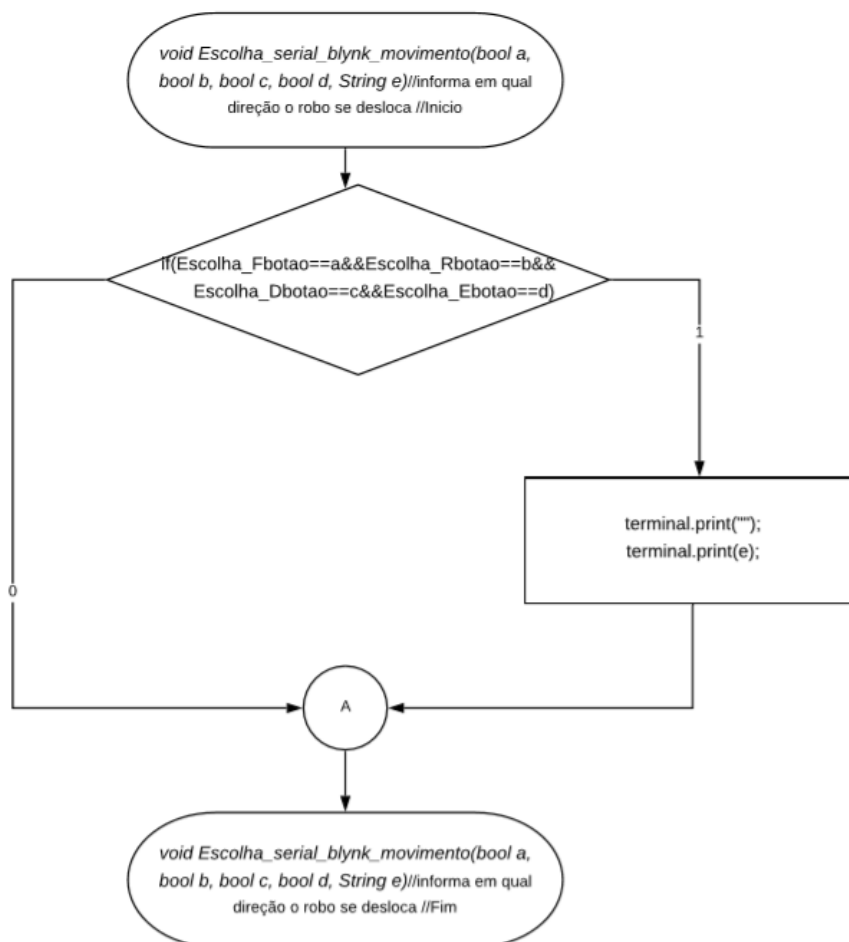
Fonte: Acervo pessoal

Fluxograma 15: Comandos do controle



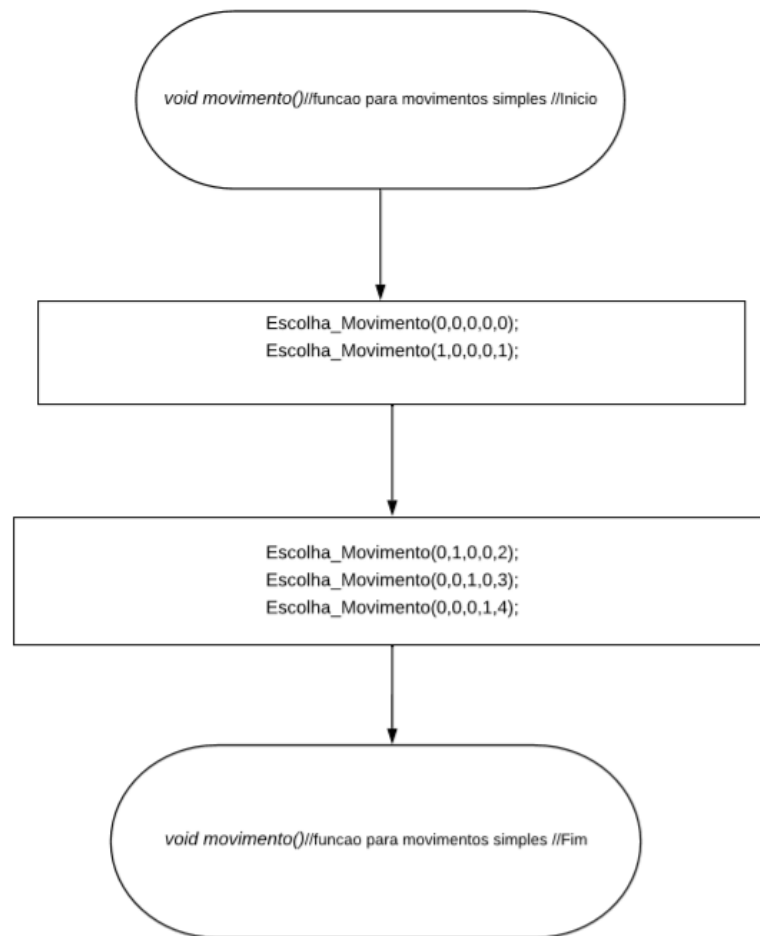
Fonte: Acervo pessoal

Fluxograma 16: Direcional do robô



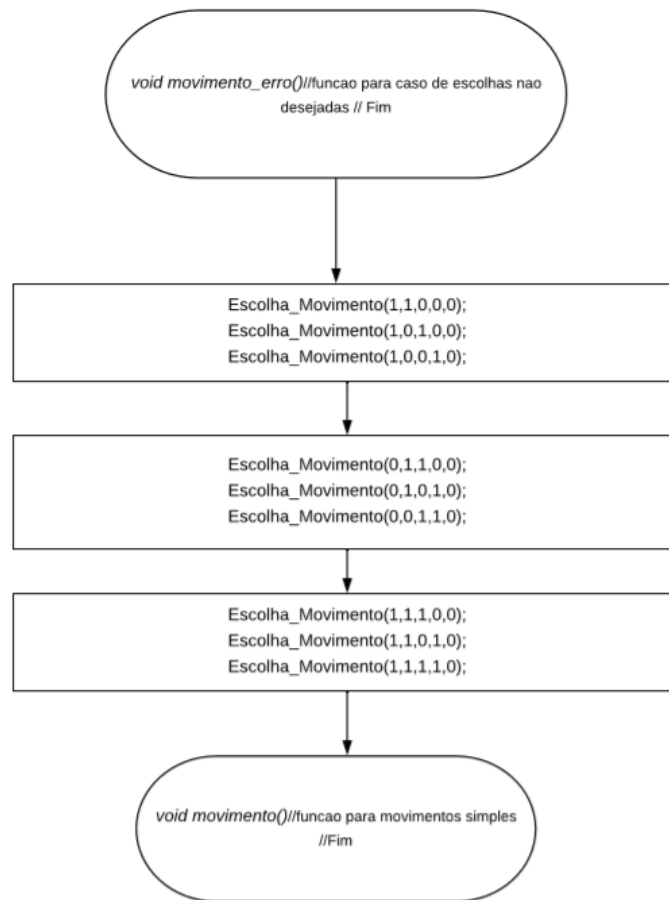
Fonte: Acervo pessoal

Fluxograma 17: Movimentos simples



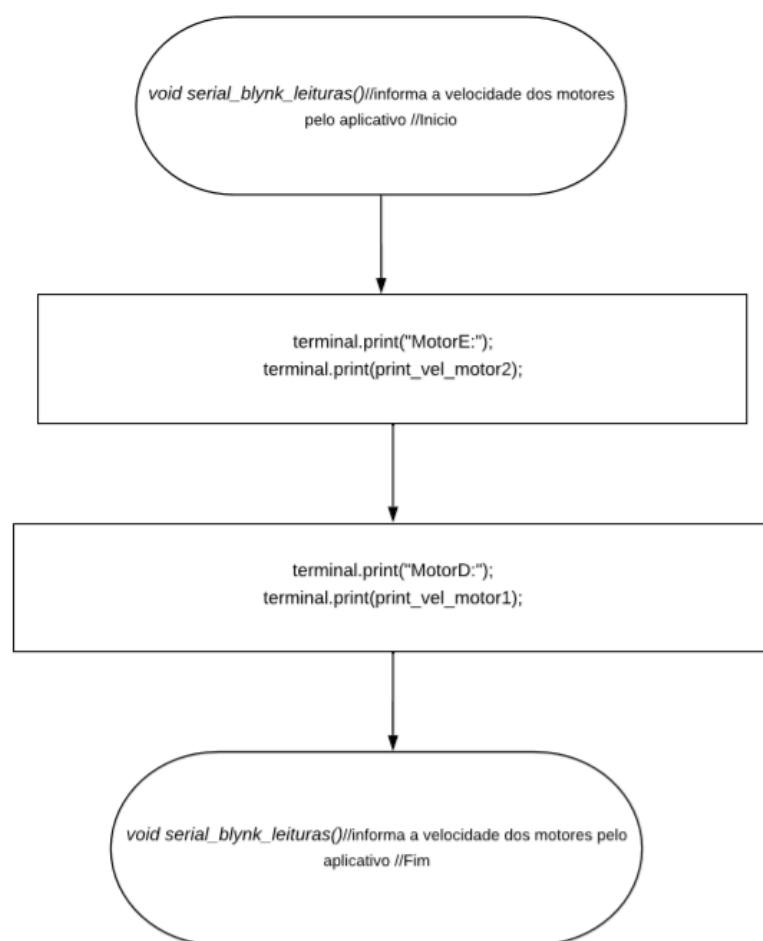
Fonte: Acervo pessoal

Fluxograma 18: Função de erro



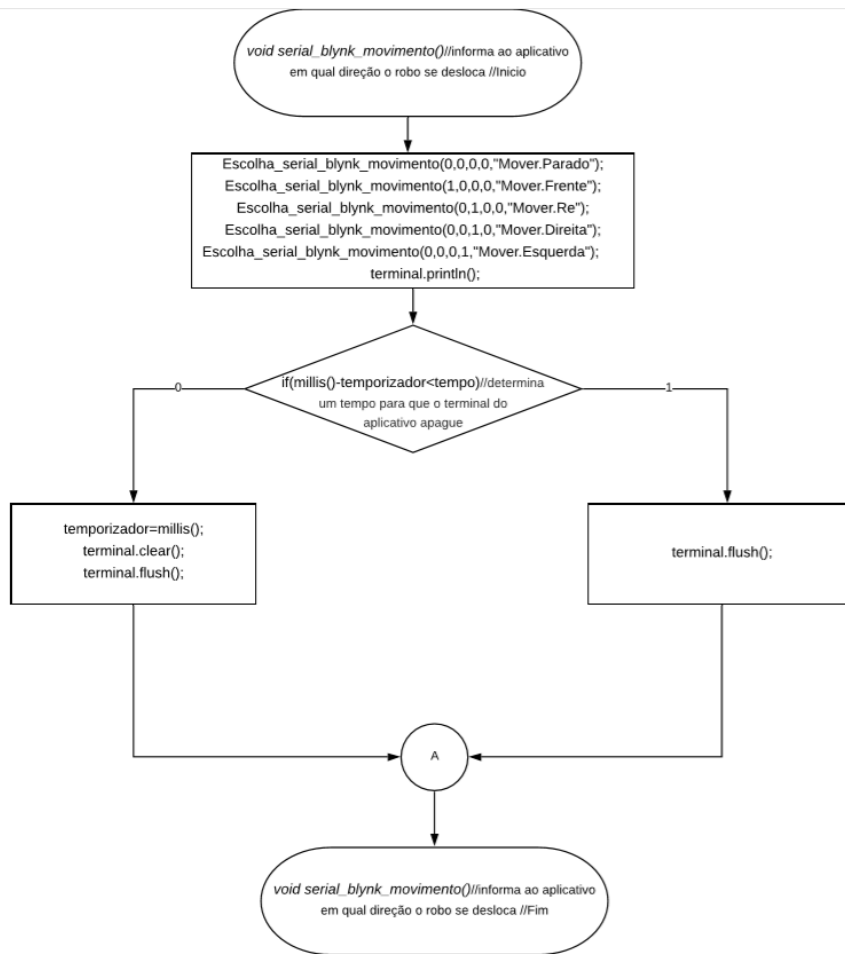
Fonte: Acervo pessoal

Fluxograma 19: Conversão da velocidade para o monitor



Fonte: Acervo pessoal

Fluxograma 20: Informa a direção escolhida para o aplicativo



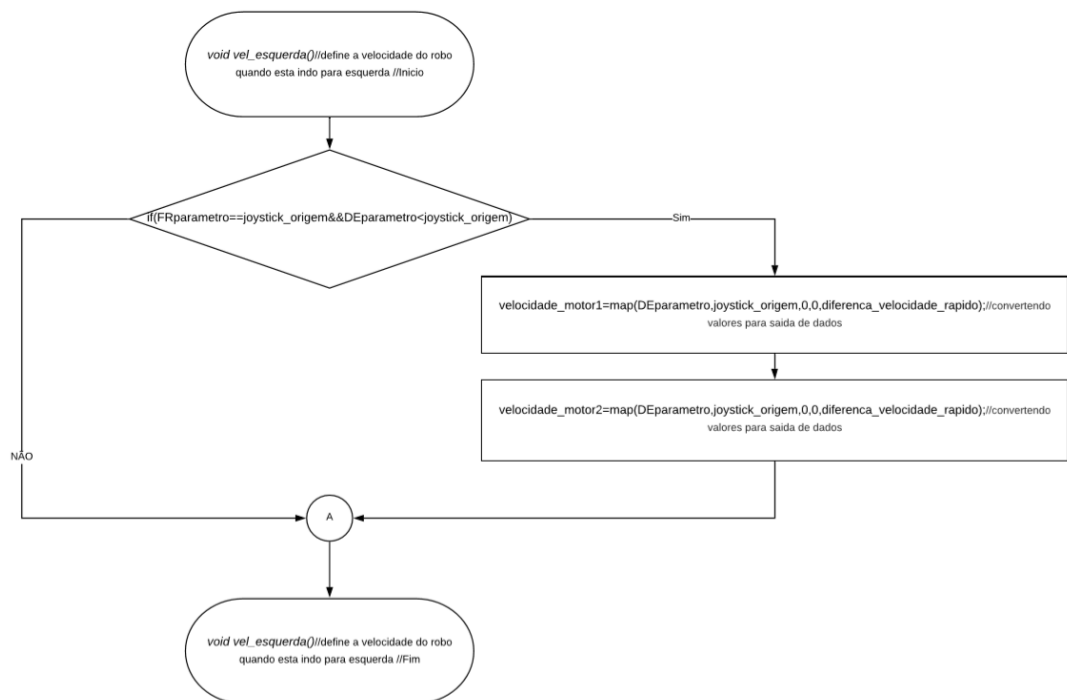
Fonte: Acervo pessoal

Fluxograma 21: Define a velocidade do robô indo para a direita



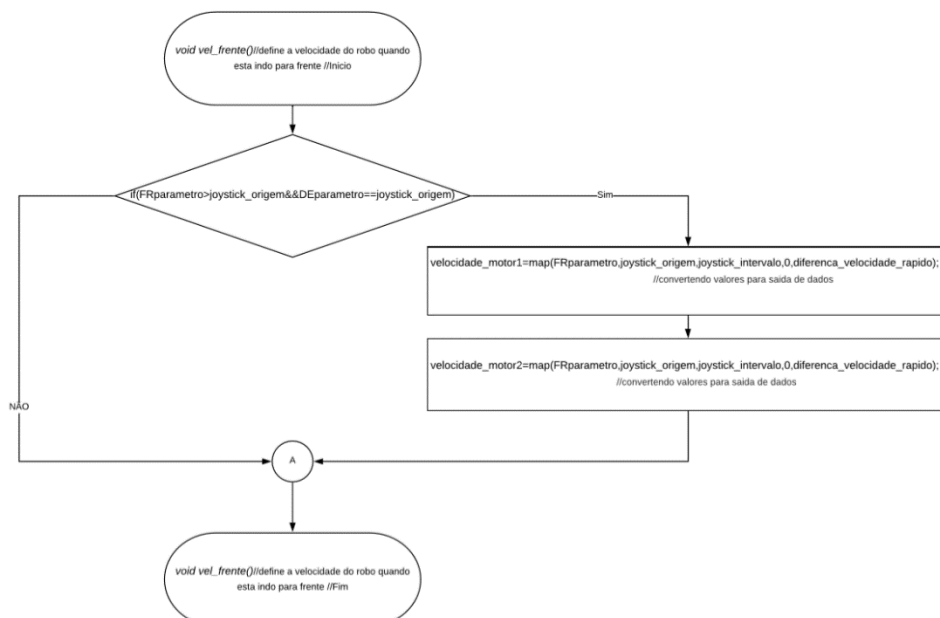
Fonte: Acervo pessoal

Fluxograma 22: Define a velocidade do robô indo para a esquerda



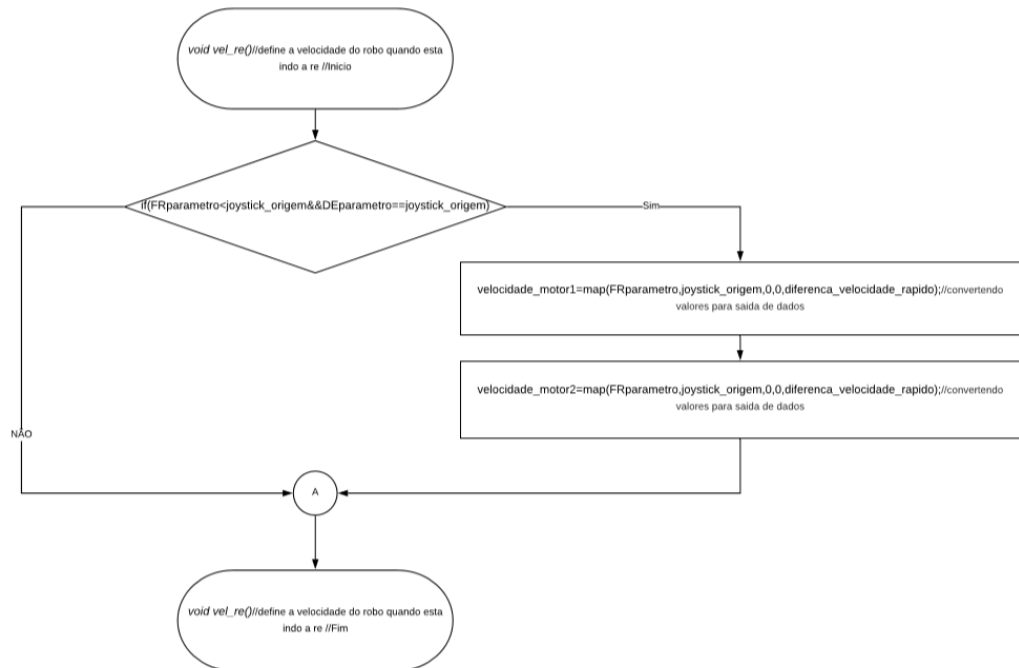
Fonte: Acervo pessoal

Fluxograma 23: Define a velocidade do robô indo para a frente



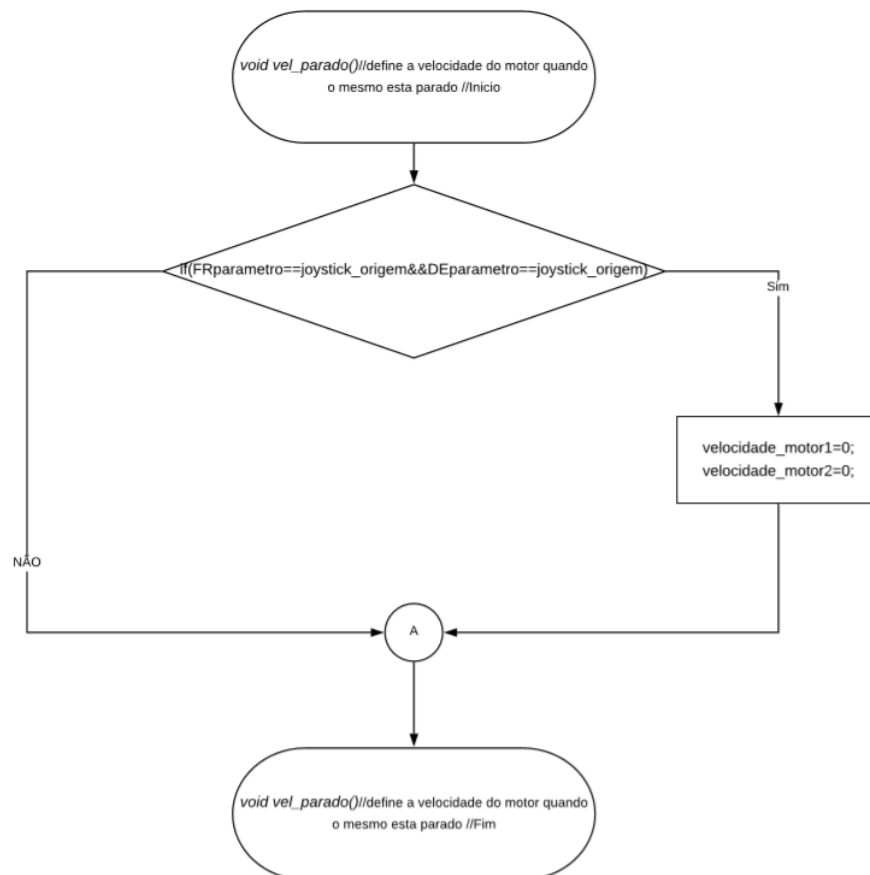
Fonte: Acervo pessoal

Fluxograma 24: Define a velocidade do motor indo para a ré



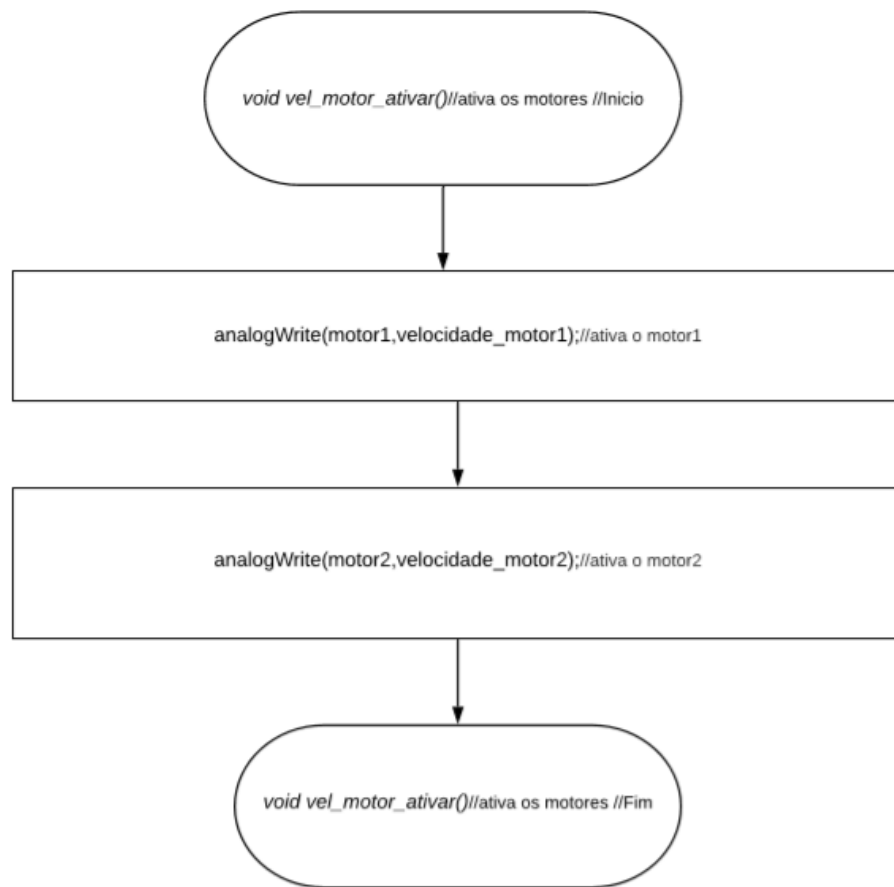
Fonte: Acervo pessoal

Fluxograma 25: Define a velocidade do motor parado



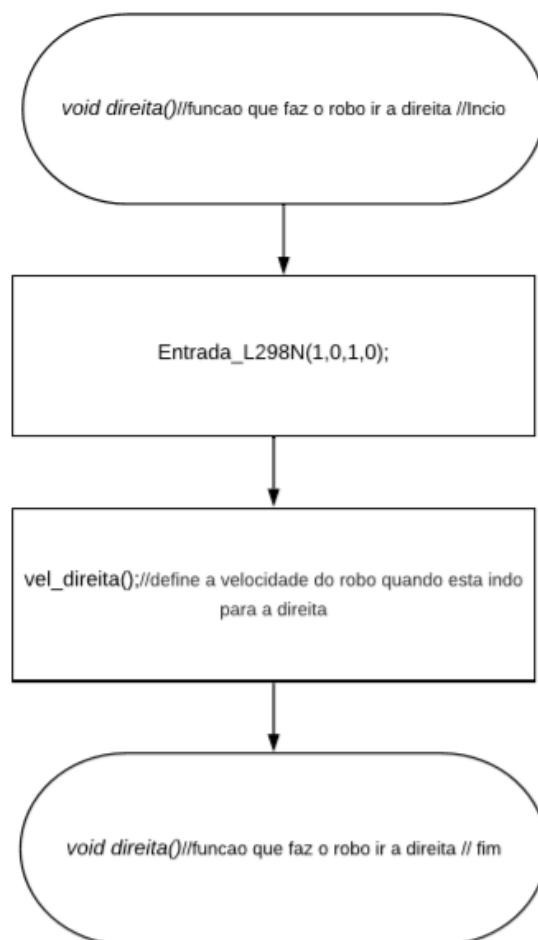
Fonte: Acervo pessoal

Fluxograma 26: Ativa os motores



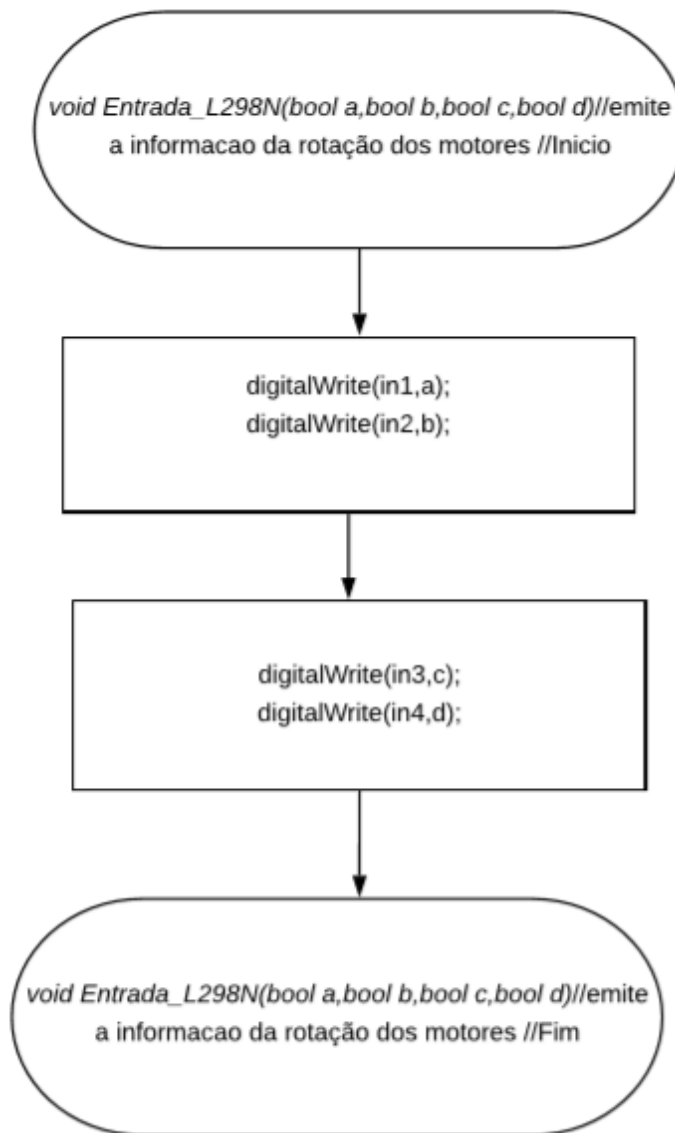
Fonte: Acervo pessoal

Fluxograma 27: Faz o robô ir para a direita



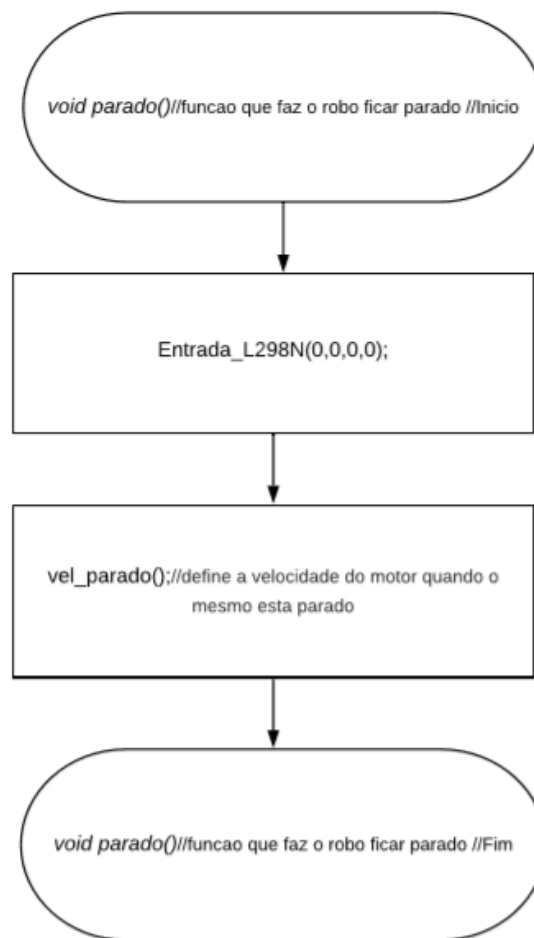
Fonte: Acervo pessoal

Fluxograma 28: Emite informação sobre a rotação dos motores



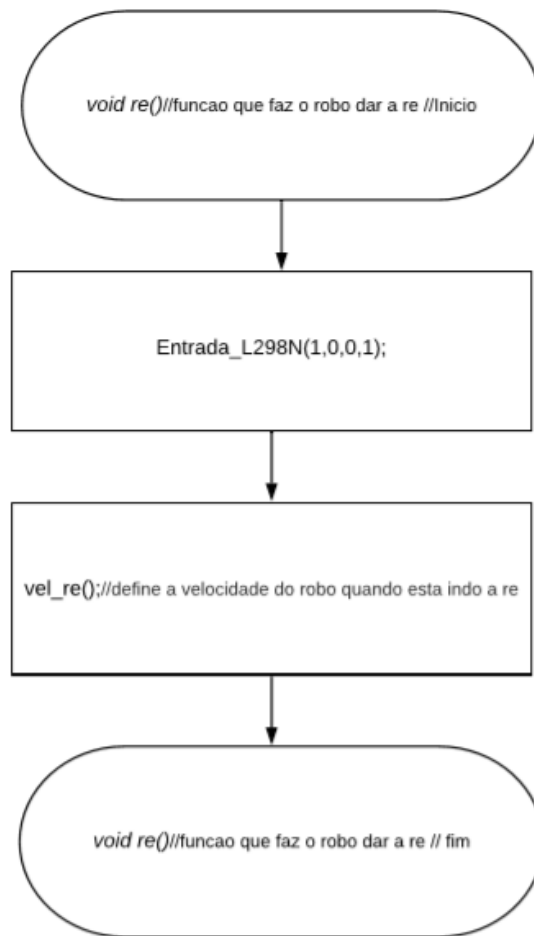
Fonte: Acervo pessoal

Fluxograma 29: Faz o robô ficar parado



Fonte: Acervo pessoal

Fluxograma 30: Faz o robô dar ré



Fonte: Acervo pessoal

6. DESENVOLVIMENTO DO PROJETO

7. CONSIDERAÇÕES FINAIS

Com o termino deste trabalho confirma-se utilidade das práticas e da competição em si, na aprendizagem e na formação acadêmica técnica do indivíduo. Percebe-se que somente colocando em prova, em testes, na produção de algo podemos perceber os desafios do tema, procuramos e adquirimos conhecimentos novos, além de perceber e sanar dúvidas até então não percebidas ou levantadas.

Fica evidente a inviabilidade de adquirir todo conhecimento necessário para uma graduação em engenharia sobre um regime semestral, sem o auxilio de aulas

laboratoriais. Transformando tal atividade em uma competição, instiga ainda mais a curiosidade e empenho do aluno, forçando-o, a dedicar-se mais ao projeto

APÊNDICE A

```
/******
*****INCLUINDO BIBLIOTECAS E DEFININDO PINOS*****
*****/

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#define in1 4//pino D2
#define in2 14//pino D5
#define in3 12//pino D6
#define in4 13//pino D7
#define motor1 5//pino D1 responsavel pela saida da velocidade do motor1
#define motor2 15//pino D8 responsavel pela saida da velocidade do motor2
#define FRbotao V0//Botão de entrada para ir para frente
#define DEbotao V1//Botão de entrada para usar a ré
#define diferenca_vel_rapido V2//Botão de entrada para diferença de velocidade dos
motores.

#define pin_joystick_intervalo V4//define o intervalo do joystick
#define monitor_blynk V5//ativa o monitor do blynk
#define apresentacao V7//realiza a apresentacao do robo osiris
WidgetTerminal terminal(monitor_blynk);//definindo pino virtual V5 como monitor serial
do blynk

/******
*****VARIAVEIS*****
*****

*****

*****
*****/

int ativar_apresentacao;
int Escolha_Fbotao;//Verifica a leitura do botão frente
```

```

int Escolha_Rbotao;//Verifica a leitura do botão ré
int Escolha_Dbotao;//Verifica a leitura do botão direito
int Escolha_Ebotao;//Verifica a leitura do botão esquerdo
int FRparametro;//verifica o valor passado pelo joystick indo para frente a re
int DEparametro;//verifica o valor passado pelo joystick indo para direita ou esquerda
int joystick_intervalo=10;//o intervalo o joystick deve ser par
int joystick_origem=joystick_intervalo/2;//realiza o calculo da origem do joystick
int diferenca_velocidade_rapido=255;//promove a diferenca de velocidades entre os
motores

int velocidade_motor1;//promove a conversão para a velocidade do motor 1
int velocidade_motor2;//promove a conversão para a velocidade do motor 2
int print_vel_motor1;//Mostra a velocidade do motor nos monitores
int print_vel_motor2;//Mostra a velocidade do motor nos monitores
unsigned long temporizador;//faz a contagem do tempo
int tempo=5000;//faz a contagem de tempo para apagar o monitor serial do blynk
int tempo_apresentacao=3000;
char auth[] = "*****"; // Código de acesso do aplicativo

Blynk

char ssid[] = "*****"; // Usuario do WiFi
char pass[] = "*****"; //Senha do WiFi

/*****
*****

*****PRINCIPAIS*****

*****

*****

*****/

void setup()
{
  Blynk.begin(auth, ssid, pass);//entrando no usuario e a senha do wifi e no aplicativo blynk
  Pin_Mode_Serial();//informa o modo dos pinos e chama o monitor serial
}

void loop()
{

```

```

    Blynk.run();//realizando os comandos dentro do aplicativo blynk
    serial_print();//manda informações importantes para o monitor serial
    vel_motor_ativar();//ativa os motores
    movimento();//realiza movimentos simples
    movimento_erro();
}

/*****
*****

*****FUNÇÕES

PINOS*****

*****/

void Pin_Mode_Serial();//informa o modo dos pinos e chama o monitor serial
{
    pinMode(in1,OUTPUT);//definindo pino como saída de dados
    pinMode(in2,OUTPUT);//definindo pino como saída de dados
    pinMode(in3,OUTPUT);//definindo pino como saída de dados
    pinMode(in4,OUTPUT);//definindo pino como saída de dados
    pinMode(motor1,OUTPUT);//definindo pino como saída de dados
    pinMode(motor2,OUTPUT);//definindo pino como saída de dados
    Serial.begin(115200);//ativando o monitor serial
    terminal.clear();//limpando o monitor serial do blynk
    terminal.flush();//manda para o monitor serial
    terminal.print("Iniciando o monitor Blynk");
    terminal.flush();//manda para o monitor serial
}

/*****
*****

*****FUNÇÕES          LEITURA          PIN

VIRTUAL*****

*****/

BLYNK_WRITE(apresentacao)

```

```

{
  ativar_apresentacao=param.asInt();
  if(ativar_apresentacao==1)
  {
    terminal.clear();//limpando o monitor serial do blynk
    terminal.flush();//manda para o monitor serial
    terminal.print("Robo em apresentacao");
    terminal.flush();//manda para o monitor serial
    frente();
    velocidade_motor1=255;
    velocidade_motor2=255;
    delay(tempo_apresentacao);
    direita();
    velocidade_motor1=255;
    velocidade_motor2=255;
    delay(tempo_apresentacao);
    esquerda();
    velocidade_motor1=255;
    velocidade_motor2=255;
    delay(tempo_apresentacao);
    re();
    velocidade_motor1=255;
    velocidade_motor2=255;
    delay(tempo_apresentacao);
  }
}

BLYNK_WRITE(pin_joystick_intervalo)//modifica a sensibilidade do joystick do robo
{
  joystick_intervalo=param.asInt();//determina o valor do intervalo do joystick pelo
aplicativo
  joystick_origem=joystick_intervalo/2;//determina a origem do joystick mantendo o robo
parado
}

```

```

    BLYNK_WRITE(diferenca_vel_rapido)//determina a velocidade maxima dos motores
    {
        diferenca_velocidade_rapido=map(param.asInt(),0,100,0,255);//realiza a
conversão da velocidade maxima para os motores
    }

    BLYNK_WRITE(FRbotao)//converte os valores do joystick para que o robo ande para frente ou
para tras
    {
        FRparametro=param.asInt();//salvando a informação do parametro frente e re
        if(FRparametro>joystick_origem)
        {
            Escolha_Fbotao = 1;//definindo botao frente como ativado
            Escolha_Rbotao = 0;//definindo botao re como desativado
        }
        if(FRparametro<joystick_origem)
        {
            Escolha_Fbotao = 0;//definindo botao frente como desativado
            Escolha_Rbotao = 1;//definindo botao re como ativado
        }
        if(FRparametro==joystick_origem)
        {
            Escolha_Fbotao = 0;//definindo botao frente como desativado
            Escolha_Rbotao = 0;//definindo botao re como desativado
        }

    }

    BLYNK_WRITE(DEbotao)//converte os valores do joystick para que o robo ande para direita ou
para esquerda
    {
        DEparametro=param.asInt();//salvando a informação do parametro direita esquerda
        if(DEparametro>joystick_origem)
        {
            Escolha_Dbotao = 1;//definindo botao direita como ativado

```



```

    Escolha_Ebotao = 0;//definindo botao esquerda como desativado
}
if(DEparametro<joystick_origem)
{
    Escolha_Dbotao = 0;//definindo botao direita como desativado
    Escolha_Ebotao = 1;//definindo botao esquerda como ativado
}
if(DEparametro==joystick_origem)
{
    Escolha_Dbotao = 0;//definindo botao direita como desativado
    Escolha_Ebotao = 0;//definindo botao esquerda como desativado
}
}

/*****
*****

*****FUNÇÕES*****
*****SERIAL*****
*****PRINTS*****
*****
*****/

void serial_print();//informacoes do monitor serial e do terminal do blynk
{
    conversao_vel_motor();//realiza a conversão da velocidade em %
    serial_leitura_parametro();//realiza a leitura dos parametros
    serial_velocidade_motor();//realiza a leitura da velocidade do motor
    serial_leitura_entrada();//realiza a leitura da entrada dos pinos virtuais
    serial_leitura_saida();//realiza a leitura da saida dos pinos de dados
    serial_blynk_leituras();//realizando leitura velocidade dos motores no blynk
    serial_blynk_movimento();//realiza leitura do movimento no blynk
}

void conversao_vel_motor();//realiza a conversão da velocidade em %
{
    print_vel_motor1=map(velocidade_motor1,0,255,0,100);//realiza a conversao da
    velocidade do motor para porcentagem

```

```
    print_vel_motor2=map(velocidade_motor2,0,255,0,100);//realiza a conversao da
    velocidade do motor para porcentagem
```

```
}
```

```
void serial_leitura_parametro();//informa ao monitor serial a leitura dos parametros
```

```
{
```

```
    Serial.print(" PARAMETRO: paramFR:");
```

```
    Serial.print(FRparametro);
```

```
    Serial.print("    paramDE:");
```

```
    Serial.print(DEparametro);
```

```
}
```

```
void serial_velocidade_motor();//informa ao monitor serial a leitura da velocidade dos motores
```

```
{
```

```
    Serial.print(" VEL MOTOR: MT2:");
```

```
    Serial.print(print_vel_motor2);
```

```
    Serial.print("    MT1:");
```

```
    Serial.print(print_vel_motor1);
```

```
}
```

```
void serial_leitura_entrada();//realiza a leitura da entrada dos pinos virtuais
```

```
{
```

```
    Serial.print(" ENTRADA: EF:");
```

```
    Serial.print(Escolha_Fbotao);//mostrando as informacoes vindas do ESP8266 para o
    monitor serial
```

```
    Serial.print("    ER:");
```

```
    Serial.print(Escolha_Rbotao);//mostrando as informacoes vindas do ESP8266 para o
    monitor serial
```

```
    Serial.print("    ED:");
```

```
    Serial.print(Escolha_Dbotao);//mostrando as informacoes vindas do ESP8266 para o
    monitor serial
```

```
    Serial.print("    EE:");
```

```
    Serial.print(Escolha_Ebotao);//mostrando as informacoes vindas do ESP8266 para o
    monitor serial
```

```
}
```

```
void serial_leitura_saida();//informa a leitura da saida dos pinos de dados
```

```

{
  Serial.print(" SAIDA: in1:");
  Serial.print(digitalRead(in1)); //mostrar no monitor serial os dados enviados para a ponte h
  Serial.print(" in2:");
  Serial.print(digitalRead(in2)); //mostrar no monitor serial os dados enviados para a ponte h
  Serial.print(" in3:");
  Serial.print(digitalRead(in3)); //mostrar no monitor serial os dados enviados para a ponte h
  Serial.print(" in4:");
  Serial.println(digitalRead(in4)); //mostrar no monitor serial os dados enviados para a ponte
h
}

void serial_blynk_leituras() //informa a velocidade dos motores pelo aplicativo
{
  terminal.print("MotorE: ");
  terminal.print(print_vel_motor2);
  terminal.print(" MotorD: ");
  terminal.print(print_vel_motor1);
}

void Escolha_serial_blynk_movimento(bool a, bool b, bool c, bool d, String e) //informa em
qual direção o robo se desloca
{
  //OBS de a até o d estao representados os parametros de escolhas dos
movimentos, ja o e representa o texto emitido

if(Escolha_Fbotao==a&&Escolha_Rbotao==b&&Escolha_Dbotao==c&&Escolha_Ebo
tao==d)
{
  terminal.print(" ");
  terminal.print(e);
}
}

void serial_blynk_movimento() //informa ao aplicativo em qual direção o robo se desloca
{

```

```

Escolha_serial_blynk_movimento(0,0,0,0,"Mover.Parado");
Escolha_serial_blynk_movimento(1,0,0,0,"Mover.Frente");
Escolha_serial_blynk_movimento(0,1,0,0,"Mover.Re");
Escolha_serial_blynk_movimento(0,0,1,0,"Mover.Direita");
Escolha_serial_blynk_movimento(0,0,0,1,"Mover.Esquerda");
terminal.println();
if(millis()-temporizador<tempo)//determina um tempo para que o terminal do aplicativo
apague
{
    terminal.flush();
}
else
{
    temporizador=millis();
    terminal.clear();
    terminal.flush();
}

}

/*****
*****

*****FUNÇÕES*****ESCOLHA
MOVIMENTO*****
*****
*****/

void Escolha_Movimento(bool a, bool b, bool c, bool d, int e)//realiza a escolha dos
movimentos do robo
{
    //observe que de a ao d estao as variaveis responsaveis para a escolha da
    direção, e o e representa o movimento a ser escolhido

    if(Escolha_Fbotao==a&&Escolha_Rbotao==b&&Escolha_Dbotao==c&&Escolha_Ebo
    tao==d)//escolhendo ficar parado

```

```

{
    switch(e)
    {
        case 0:
            parado();
            break;
        case 1:
            frente();
            break;
        case 2:
            re();
            break;
        case 3:
            direita();
            break;
        case 4:
            esquerda();
            break;
    }
}

```

void movimento()//funcao para movimentos simples

```

{
    Escolha_Movimento(0,0,0,0,0);
    Escolha_Movimento(1,0,0,0,1);
    Escolha_Movimento(0,1,0,0,2);
    Escolha_Movimento(0,0,1,0,3);
    Escolha_Movimento(0,0,0,1,4);
}

```

void movimento_erro()//funcao para caso de escolhas nao desejadas

```

{
    Escolha_Movimento(1,1,0,0,0);
    Escolha_Movimento(1,0,1,0,0);
}

```

```

        Escolha_Movimento(1,0,0,1,0);
        Escolha_Movimento(0,1,1,0,0);
        Escolha_Movimento(0,1,0,1,0);
        Escolha_Movimento(0,0,1,1,0);
        Escolha_Movimento(1,1,1,0,0);
        Escolha_Movimento(1,1,0,1,0);
        Escolha_Movimento(1,1,1,1,0);
    }

    /*****
*****
*****FUNÇÕES
MOVIMENTO*****
*****
*****/

    /*****
*****
*****VELOCIDADE
MOTORES*****/

    void vel_motor_ativar()//ativa os motores
    {
        analogWrite(motor1,velocidade_motor1);//ativa o motor1
        analogWrite(motor2,velocidade_motor2);//ativa o motor2
    }

    void vel_parado()//define a velocidade do motor quando o mesmo esta parado
    {
        if(FRparametro==joystick_origem&&DEparametro==joystick_origem)
        {
            velocidade_motor1=0;
            velocidade_motor2=0;
        }
    }

    void vel_frente()//define a velocidade do robo quando esta indo para frente

```

```

{
    if(FRparametro>joystick_origem&&DEparametro==joystick_origem)
    {

velocidade_motor1=map(FRparametro,joystick_origem,joystick_intervalo,0,diferenca
_velocidade_rapido);//convertendo valores para saida de dados

velocidade_motor2=map(FRparametro,joystick_origem,joystick_intervalo,0,diferenca
_velocidade_rapido);//convertendo valores para saida de dados
    }
}
void vel_re()//define a velocidade do robo quando esta indo a re
{
    if(FRparametro<joystick_origem&&DEparametro==joystick_origem)
    {

velocidade_motor1=map(FRparametro,joystick_origem,0,0,diferenca_velocidade_rap
ido);//convertendo valores para saida de dados

velocidade_motor2=map(FRparametro,joystick_origem,0,0,diferenca_velocidade_rap
ido);//convertendo valores para saida de dados
    }
}
void vel_direita()//define a velocidade do robo quando esta indo para direita
{
    if(FRparametro==joystick_origem&&DEparametro>joystick_origem)
    {

velocidade_motor1=map(DEparametro,joystick_origem,joystick_intervalo,0,diferenca
_velocidade_rapido);//convertendo valores para saida de dados

velocidade_motor2=map(DEparametro,joystick_origem,joystick_intervalo,0,diferenca
_velocidade_rapido);//convertendo valores para saida de dados

```

```

    }
}

void vel_esquerda()//define a velocidade do robo quando esta indo para esquerda
{
    if(FRparametro==joystick_origem&&DEparametro<joystick_origem)
    {

velocidade_motor1=map(DEparametro,joystick_origem,0,0,diferenca_velocidade_rapido);
//convertendo valores para saida de dados

velocidade_motor2=map(DEparametro,joystick_origem,0,0,diferenca_velocidade_rapido);
//convertendo valores para saida de dados

    }
}

/*****
*****

*****MOVIMENTO
SIMPLES*****/

void Entrada_L298N(bool a,bool b,bool c,bool d)//emite a informacao da rotaçao dos
motores
{
    //observe que de a até o d as variaveis são responsaveis por ativar cada tipo
de rotacao

    digitalWrite(in1,a);
    digitalWrite(in2,b);
    digitalWrite(in3,c);
    digitalWrite(in4,d);
}

void parado()//funcao que faz o robo ficar parado
{
    Entrada_L298N(0,0,0,0);
    vel_parado();//define a velocidade do motor quando o mesmo esta parado
}

```


void frente()//funcao que faz o robo ir para frente

```
{  
  Entrada_L298N(0,1,1,0);  
  vel_frente();//define a velocidade do robo quando esta indo para frente  
}
```

void re()//funcao que faz o robo dar a re

```
{  
  Entrada_L298N(1,0,0,1);  
  vel_re();//define a velocidade do robo quando esta indo a re  
}
```

void direita()//funcao que faz o robo ir a direita

```
{  
  Entrada_L298N(1,0,1,0);  
  vel_direita();//define a velocidade do robo quando esta indo para a direita  
}
```

void esquerda()//funcao que faz o robo ir a esquerda

```
{  
  Entrada_L298N(0,1,0,1);  
  vel_esquerda();//define a velocidade do robo quando esta indo para a esquerda  
}
```

8. REFERÊNCIAS BIBLIOGRÁFICAS

SILVA NETTO, Abner da; SILVEIRA, Marco Antônio Pinheiro da. Gestão da segurança da informação: fatores que influenciam sua adoção em pequenas e médias empresas. **JISTEM-Journal of Information Systems and Technology Management**, v. 4, n. 3, p. 375-397, 2007.

<https://repositorio.ufpb.br/jspui/handle/123456789/16068> Acesso em: Maio 26, 2020 as 20:08

<https://www.embarcados.com.br/ifttt-plataforma-para-iot/> Acesso em: Maio 26, 2020 as 20:08

Martin, J. A. (2019, Janeiro 31). Computerworld. Acesso em: Maio 26, 2020, Disponível em : <https://www.computerworld.com/article/3239304/what-is-ifttt-how-to-use-if-this-then-that-services.html>

HALLIDAY, David, RESNICK, Robert, WALKER, Jearl. Fundamentos de Física - Vol. 3 - Eletromagnetismo, 10ª edição. [Digite o Local da Editora]; [Digite o Nome da Editora], [Inserir ano de publicação]. 9788521632092. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788521632092/>.

WENTWORTH, M., S. Fundamentos de Eletromagnetismo.]. 978-85-216-2670-1. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/978-85-216-2670-1/>. Acesso em: 26 May 2020

Em direção à melhoria do ensino na área tecnológica: a experiência de uma disciplina de introdução à engenharia de controle e automação /

Acesso em: 26 May 2020 <https://repositorio.ufsc.br/handle/123456789/78477>

Relato de experiência com metodologia ativa de aprendizagem em uma disciplina de programação básica com ingressantes dos cursos de Engenharia da Computação, Engenharia de Controle e Automação e Engenharia Elétrica

Acesso em: 26 May 2020

<http://retec.eti.br/retec/index.php/retec/article/view/29>

<https://www.embarcados.com.br/introducao-ao-blynk-app/-Blynk>

<http://www.pipe.ufpr.br/portal/defesas/dissertacao/105.pdf>

https://www.robocore.net/upload/attachments/ponte_h_590.pdf

