

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia Eletrônica

**Sistema para o monitoramento e acionamento
de um plano inclinado com dois graus de
liberdade para uso em um laboratório de ensino.**

Autor: Letícia Ribeiro de Macedo
Orientador: Dr. Guillermo Alvarez Bestard

Brasília, DF
2022



Letícia Ribeiro de Macedo

Sistema para o monitoramento e acionamento de um plano inclinado com dois graus de liberdade para uso em um laboratório de ensino.

Monografia submetida ao curso de graduação em (Engenharia Eletrônica) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Eletrônica).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Dr. Guillermo Alvarez Bestard

Brasília, DF

2022

Letícia Ribeiro de Macedo Sistema para o monitoramento e acionamento de um plano inclinado com dois graus de liberdade para uso em um laboratório de ensino. / Letícia Ribeiro de Macedo. – Brasília, DF, 2022- 68 p. : il. (algumas color.) ; 30 cm.
Orientador: Dr. Guillermo Alvarez Bestard
Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2022.
1. Plano inclinado. 2. Faculdade UnB Gama. I. Dr. Guillermo Alvarez Bestard. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Sistema para o monitoramento e acionamento de um plano inclinado com dois graus de liberdade para uso em um laboratório de ensino.
CDU

Letícia Ribeiro de Macedo

Sistema para o monitoramento e acionamento de um plano inclinado com dois graus de liberdade para uso em um laboratório de ensino.

Monografia submetida ao curso de graduação em (Engenharia Eletrônica) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Eletrônica).

Trabalho aprovado. Brasília, DF, 01 de Dezembro de 2022:

Dr. Guillermo Alvarez Bestard
Orientador

Dr. Gilmar Silva Beserra
Convidado 1

Dr. José Alfredo Ruiz Vargas
Convidado 2

Brasília, DF
2022

Resumo

A Educação 4.0 vem para modificar os meios de aprendizagem manipulando nossa forma de obter conhecimento através da tecnologia. Os laboratórios são ferramentas importantes para auxiliar na evolução dos alunos nos estudos, são formas extras de garantir a compreensão acerca dos conceitos estudados com métodos tradicionais. O plano inclinado do tipo "*ball on plate*" destaca-se na área de sistema de controle por sua natureza multivariável e em sistemas embarcados pela dificuldade de implementar o controle. Neste trabalho foi desenvolvido o sistema de medição da posição da bola e o sistema de acionamento de uma plataforma de dois graus de liberdade, utilizando servo-motores. Foram implementadas duas malhas de controle com reguladores PID para testar o desempenho dos sistemas. Se tentou manter o custo do sistema baixo, utilizando *software* livre e que fora adaptável a diferentes microcontroladores, com o objetivo que em trabalhos futuros seja utilizado num laboratório remoto.

Palavras-chaves: Ball on plate, Controle PID, Educação 4.0, Laboratório, Plano inclinado.

Abstract

Education 4.0 comes to modify the forms of learning by changing our way of acquire knowledge through technology. Laboratories are important tools to help students progress in their studies, they are extra ways to ensure understanding of the concepts studied with traditional methods. They allow students to deepen their knowledge and become more engaged in the disciplines with the practical application of the themes studied. The inclined plane of the "ball on plate" type stands out in the control system area due to its multivariable nature and in embedded systems due to the difficulty of implementing the control. In this work, was developed a ball position measurement system and a two-degree-of-freedom platform drive system, using servomotors. Two control loops with PID regulators were implemented to test the performance of the systems. An attempt was made to keep the cost of the system low, using free software that was adaptable to different microcontrollers, with the aim of using it in a remote laboratory in future work.

Key-words: Ball on plate, Education 4.0, PID control, Laboratory.

Sumário

1	INTRODUÇÃO	13
1.1	Objetivos do trabalho	14
1.2	Organização do documento	14
2	REFERENCIAL TEÓRICO	15
2.1	Educação 4.0	15
2.2	Plano Inclinado	17
2.2.1	Função de transferência do sistema	18
2.2.2	Controlador PID	20
2.2.3	Exemplos na literatura de controle em plano inclinado	22
3	MATERIAIS E MÉTODOS	24
3.1	Proposta de solução	24
3.2	Sistema mecânico	27
3.3	Sistema de medição da posição da bola	29
3.3.1	Algoritmo para medir a posição da bola	30
3.4	Sistema de acionamento	34
3.5	Sistema de controle	36
3.5.1	Placa do circuito	36
3.5.2	Microcontrolador	37
3.5.2.1	Algoritmo para aquisição de dados e controle dos motores	39
3.5.2.2	Algoritmo para controle PID	40
3.6	Projeto de controle	43
4	RESULTADOS	48
4.1	Medição da posição bola	48
4.2	Controle da posição do plano inclinado	50
5	CONCLUSÕES	56
	Referências	57
	ANEXOS	60
	ANEXO A – DESENHO MECÂNICO	61

ANEXO B – ESTRUTURA FINAL EM ACRÍLICO	66
ANEXO C – PREPARAÇÃO DA IDE PARA PROGRAMAÇÃO DA PLACA	67

Lista de ilustrações

Figura 1 – Dimensões significativas de implementação de uma Indústria 4.0 de sucesso.	16
Figura 2 – Fases de transição em direção a Indústria 4.0.	16
Figura 3 – Diagrama de forças do plano inclinado.	18
Figura 4 – Diagrama de blocos de um controlador PID	20
Figura 5 – Estrutura do plano inclinado.	23
Figura 6 – Diagrama de blocos do laboratório remoto	26
Figura 7 – Desenho mecânico vista frontal	27
Figura 8 – Desenho mecânico vista lateral	28
Figura 9 – Desenho mecânico vista isométrica	29
Figura 10 – Raspberry Pi.	29
Figura 11 – Estrutura com a webcam conectada a Raspberry Pi.	30
Figura 12 – Limiar de cor encontrado no frame com resolução de 480 X 480.	31
Figura 13 – Contorno encontrado no rastreamento com resolução de 480 X 480.	31
Figura 14 – Fluxograma do código do rastreio da bola.	32
Figura 15 – Quadro de comunicação.	33
Figura 16 – Diagrama das conexões da comunicação UART.	34
Figura 17 – Servo motor	35
Figura 18 – Funcionamento do servo motor.	35
Figura 19 – Comunicação entre o microcontrolador e o servo motor.	36
Figura 20 – Placa para conexão dos componentes	36
Figura 21 – Esquemático do circuito	37
Figura 22 – Componentes do microcontrolador	38
Figura 23 – Raspberry Pi Pico	38
Figura 24 – Fluxograma do código do microcontrolador	40
Figura 25 – Fluxograma do código do PID	42
Figura 26 – Planta para o controle PID	44
Figura 27 – Planta para o controle simulado no SciLab	44
Figura 28 – Simulação no SciLab controlador P	45
Figura 29 – Simulação no SciLab controlador PD	45
Figura 30 – Simulação no SciLab controlador PI	46
Figura 31 – Simulação no SciLab controlador PID	46
Figura 32 – Estrutura de testes.	48
Figura 33 – Rastreamento da posição da bola	49
Figura 34 – Posição um da bola em milímetros no frame de 480X480.	52
Figura 35 – Controle um da bola em X na plataforma.	52

Figura 36 – Controle um da bola em Y na plataforma.	53
Figura 37 – Posição dois da bola em milímetros no frame de 480X480	53
Figura 38 – Controle dois da bola em X na plataforma.	54
Figura 39 – Posição dois da bola em Y na plataforma.	54
Figura 40 – Base superior	61
Figura 41 – Base lateral	62
Figura 42 – Braço câmera	63
Figura 43 – Cubo central	64
Figura 44 – Eixo base	65
Figura 45 – Estrutura em acrílico	66
Figura 46 – Gerenciador de placas na IDE do Arduino	67
Figura 47 – Instalação da placa da Raspberry Pi Pico na IDE do Arduino.	68
Figura 48 – Seleção da placa da Raspberry Pi Pico na IDE do Arduino.	68

Lista de tabelas

Tabela 1 – Lista de componentes	26
Tabela 2 – Especificações elétricas dos componentes	27
Tabela 3 – Testes realizados na medição da posição da bola.	49
Tabela 4 – Testes de erro na medição da posição da bola.	50
Tabela 5 – Testes realizados no eixo X.	51
Tabela 6 – Testes realizados no eixo Y.	51
Tabela 7 – Valores de PID.	51

Lista de abreviaturas e siglas

CPS	Sistemas ciber físicos
IoT	Internet das coisas
MIMO	Multiple input multiple output(múltiplas entradas múltiplas saídas)
2DOF	Dois graus de liberdade
PID	Proporcional integral derivativo
UART	Transmissor/receptor assíncrono universal
KP	Coefficiente de ação proporcional
KI	Coefficiente de ação integral
KD	Coefficiente de ação derivativa
TCC	Trabalho de conclusão de curso
RAM	Memória de acesso aleatório
PWM	Modulação por largura de pulso

Lista de símbolos

α	Parâmetro para o ângulo da placa a ser calculado
θ	Parâmetro para o ângulo do servo a ser calculado

1 Introdução

As revoluções industriais trouxeram grandes mudanças para a sociedade. As mudanças foram tecnológicas, socioeconômicas e culturais, houve a introdução de novos materiais como o ferro e o aço, novas fontes de energia como o carvão, criação de novos maquinários com isso uma menor utilização de força humana no trabalho, divisão maior do trabalho, aumento na inserção da ciência na indústria, produção em massa de bens (Britanica, 2021).

A quarta revolução industrial está acontecendo agora ela utiliza sistemas ciber físicos para controle e obtenção de dados da produção (Duarte, 2017). Existem quatro componentes básicos desse sistema: são sistemas ciber físicos, Internet das Coisas, Internet dos Serviços e Fábricas Inteligentes (De Carvalho et al., 2020).

Com isso, todas as camadas da sociedade serão alteradas, inclusive a educação. Entraremos na era da Educação 4.0 que transformará as formas de aprendizado. Para (Antonio et al., 2019) as tecnologias da informação e comunicação, por facilitarem a transmissão de dados propiciam novas formas de ensino a distância. Dessa maneira a tecnologia será uma grande aliada da educação remota. Alkhaldi et al. (2016) define laboratório remoto como um experimento que está localizado fisicamente longe do usuário, mas está conectado a rede por meio remoto, permitindo conduzir ele.

Os laboratórios remotos são desenvolvidos para acesso e manipulação de peças reais. O propósito do laboratório sobre o plano inclinado, desenvolvido neste trabalho é permitir que os alunos acessem futuramente o experimento, por meio de um *site web* que contará com a transmissão de vídeo e obtenção de dados em tempo real. Através das interfaces desenvolvidas, será possível a manipulação do experimento, programação do microcontrolador nas disciplinas de eletrônica embarcada.

Também poderá ser utilizado nas disciplinas de sistemas de controle e robótica, adicionando uma maior complexidade ao projeto de controle. O modelo desenvolvido tem dois graus de liberdade de movimentação e é considerado um clássico no ensino de sistemas não lineares multivariáveis. Por conta disso, o desenvolvimento do controlador não é um processo trivial e é tema de disciplinas na pós-graduação.

A Indústria 4.0 veio para modificar a sociedade por completo da mesma forma que as outras revoluções industriais fizeram. Isso significa que a educação também será atingida por essas transformações, os laboratórios remotos estão cada vez mais presentes nas universidades e escolas. O laboratório sobre plano inclinado pode ser utilizado por diversas matérias em diferentes engenharias, o que torna ele acessível a um número maior de alunos. A pandemia de Covid 19 reforçou a importância da construção de laboratórios a distância de disciplinas práticas em diversas áreas e não somente em engenharia.

O uso de laboratórios remotos para o ensino segundo [Esquembre \(2015\)](#) confirma a necessidade desse tipo de material para melhorar a aprendizagem e aumentar o envolvimento dos alunos nas aulas. O sistema de plano inclinado, modelo *ball on plate*, é muito popular no ensino de sistemas de controle visto que é de simples porém bem desafiadora a elaboração do projeto de controle ([Cheng and Tsai, 2016](#) apud [J. Hauser and Kokotovic, 1992](#)).

A elaboração de um laboratório aumenta a compreensão dos alunos acerca do que foi ensinado em sala de aula, melhora o engajamento dos alunos nas matérias que apresentam esse tipo de ferramenta e melhora a aprendizagem além de ser uma nova ferramenta para professores trabalharem e desenvolverem com alunos em Iniciação Científica.

1.1 Objetivos do trabalho

O objetivo geral deste trabalho é desenvolver o sistema de medição e de acionamento para mudar a posição da bola em um modelo de plano inclinado, e duas as malhas de controle independentes para testar o sistema.

Os objetivos específicos, que contribuem com o objetivo geral, são listados a seguir: 1) Desenvolver o sistema de medição da posição da bola no plano inclinado utilizando processamento de imagens. 2) Desenvolver o sistema de acionamento para mudar a posição da bola no plano inclinado. 3) Validar o comportamento estático e dinâmico dos sistemas de medição e de acionamento utilizando duas malhas de controle independentes.

Vale reforçar que este trabalho faz parte de um projeto maior, onde são implementadas várias práticas de laboratório com acesso remoto. Apenas uma parte de uma dessas práticas será desenvolvida neste trabalho.

1.2 Organização do documento

O documento se divide em 4 capítulos, o primeiro é o referencial teórico necessário para compreensão sobre o que é o projeto e os conceitos nele empregados, o segundo são os materiais e métodos onde são detalhados os requisitos e as funcionalidades de cada parte que compõem o desenvolvimento eletrônico e mecânico.

O terceiro capítulo mostra os resultados obtidos na construção do plano inclinado referente a planta de controle. O último apresenta as conclusões obtidas com tudo o que foi desenvolvido. Posteriormente se incluem as referências bibliográficas e três anexos que permitem aprofundar no desenho mecânico, no modelo desenvolvido e na preparação para a programação da placa.

2 Referencial teórico

Para uma melhor compreensão de conceitos que serão aplicados posteriormente no desenvolvimento deste trabalho nesta seção serão apresentados alguns conceitos fundamentais.

2.1 Educação 4.0

O Governo Alemão apresenta a Indústria 4.0 como um projeto emergente em que a manufatura e os sistemas de logística utilizarão Sistema de Produção Física Cibernética, onde eles usam intensivamente as rede de informações e comunicações para uma troca super automatizada de informações onde os processos de produção e negócios são combinados (MAK, 2016).

De maneira geral, a Indústria 4.0 pode ser definida como "se caracteriza pela capacidade de adaptação, a eficiência dos recursos e ergonomia, bem como a integração de clientes e parceiros de negócios em processos de negócios e de valor. Sua base tecnológica é composta por sistemas físicos/cibernéticos e a Internet das Coisas." (Santos, 2015, p.12)

Os conceitos de Indústria 4.0 e suas tecnologias são feitos baseados em criar uma processo que visa melhorar velocidade e dar uma maior versatilidade para os produtores (El Hamdi et al., 2019), para ele esses conceitos se baseiam em 3 princípios, que podem ser vistos na Figura 1.

Pela Figura 1 é possível ver 3 etapas muito bem definidas, a primeira delas é a digitalização de todos os processos físicos bem como a utilização de sistemas embarcados, sensores, a segunda é a parte de controle, onde os dados colhidos em tempo real são utilizados para tomada de decisões a respeito da produção e a terceira e última etapa seria a automação centrada nos seres humanos e na melhoria da experiência do utilizador dos produtos feitos.

Na Figura 2 é apresentada as 4 fases de transição para a implementação da Indústria 4.0 conforme El Hamdi (El Hamdi et al., 2019) essas fases são, o monitoramento que é parte de sensores e dados externos utilizados para monitorar os produtos e gerar notificação se houver alguma mudança. O controle que com o software embarcado nos produtos possibilita o controle de funcionalidades do produto e permite a personalização para cada usuário. A otimização em que as duas primeiras fases permite que o algoritmo realize otimizações e melhorias no produto e por fim a autonomia que com as três fases anteriores o produto se torna autônomo.

Conseqüentemente para que ocorra a implementação dos conceitos e fundamentos

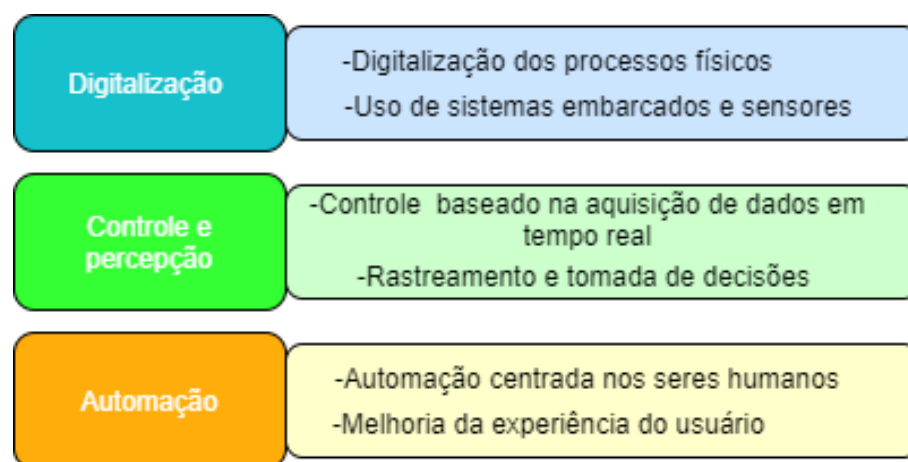


Figura 1 – Dimensões significativas de implementação de uma Indústria 4.0 de sucesso.

Fonte: Modificado de (El Hamdi et al., 2019, p.2).

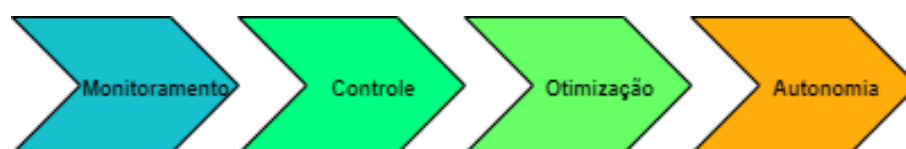


Figura 2 – Fases de transição em direção a Indústria 4.0.

Fonte: Modificado de (El Hamdi et al., 2019, p.2).

da Indústria 4.0 em uma empresa é necessário a implementação de sensores e sistemas embarcados que são os sistemas Ciber Físicos e Internet das Coisas, para a conectividade entre as máquinas, monitoramento constante da produção e obtenção de dados em tempo real.

A transformação da Indústria 4.0 não se limita apenas às indústrias, elas trarão mudanças em todas as camadas da sociedade, inclusive na educação.

A Educação 4.0 faz referência à quarta revolução industrial. De acordo com Digital (2021) a Educação 4.0 visa transformar a forma como aprendemos, aliando a tecnologia a nossa forma de obter conhecimento ainda segundo ele o conceito dessa nova forma de ensino é aliar a tecnologia como instrumento de interação estimulando o trabalho coletivo e criatividade, estimulando os alunos a se envolverem mais na sua educação.

Para James (2021) a fim de que as universidades formem alunos bem-sucedido elas devem capacitar os alunos considerando que os sistemas ciber-físicos estão presentes em todos os lugares "Isso significa ensinar aos alunos essa tecnologia como parte do currículo, mudando completamente a abordagem de aprendizagem e utilizando essa tecnologia para melhorar a experiência universitária." (James, 2021, p.1).

Segundo James (2021) uma forma de garantir o sucesso dessa nova abordagem é acelerar a utilização de ensino remoto, colocando as aulas teóricas para serem remotas mas deixando as disciplinas práticas ainda de forma presencial, assim será exigido dos alunos a capacidade de se adaptarem a novos desafios preparando-os para possíveis mudanças em

suas carreiras no futuro.

[Digital \(2021\)](#) diz que existem 4 pilares para a Educação 4.0, eles são:

- 1. Modelo sistêmico : verificar o que existe atualmente e traçar estratégias para inovação;
- 2. Mudança do senso comum: utilizar informações científicas e tecnológicas que tratem da educação para implementar em sala;
- 3. Engenharia e gestão do conhecimento : verificar as aptidões dos alunos ;
- 4. Cibercultura: preparar as escolas para ofertar de forma eficiente esse novo estilo de ensino.

Partindo do conceito de Educação 4.0, a criação de laboratórios remotos são ferramentas de implementação da nova forma de ensino que a sociedade utilizará para transformar a educação.

A Educação 4.0 está presente neste trabalho uma vez que, o sistema de medição da posição da bola e acionamento aqui desenvolvido, é para o uso futuro em um laboratório remoto para práticas de ensino em engenharias, com o intuito de desenvolver um sistema com aquisição de dados em tempo real e manipulação da estrutura do plano inclinado de forma totalmente remota.

2.2 Plano Inclinado

O plano inclinado é um sistema que se movimenta em duas direções com o intuito de manter uma bola em um caminho definido, ainda que haja alguma perturbação ([Castro et al., 2013](#))

De acordo com [Bang and Lee \(2018\)](#), a maioria dos sistemas *Ball and Plate* implementados usa um *hardware 2-DOF* que somente se movimenta em relação ao eixo X e ao eixo Y e utiliza câmeras para ter informação sobre o posicionamento da bola.

Segundo [Wang et al. \(2007\)](#) o modelo *Ball on Plate* é um sistema multivariável não linear, é um dispositivo bidimensional eletromecânico que é uma extensão do modelo *Ball and Beam* e compreende problemas em diversas áreas como controle não linear, controle inteligente, processamento de imagens entre outros.

O plano inclinado é um circuito instável que sem um sistema de controle a bola desliza por ela sem estabilidade, o controle é usado para medir onde a bola está e assim determinar a movimentação que a plataforma deve fazer para controlar a sua localização ([Ismail, 2015](#)).

O sistema *ball on plate* se tornou popular no ensino de sistemas de controle pois é de fácil configuração e implementação (Cheng and Tsai, 2016 apud J. Hauser and Kokotovic, 1992).

O plano inclinado é um sistema multivariável, pois necessita que as duas saídas do sistema sejam controladas, a saída referente ao eixo X e a saída referente ao eixo Y. Ambas variáveis, aparentemente isoladas, estão interligadas por subsistemas que fazem com que a mudança de uma delas possa afetar também a outra.

O plano inclinado a ser detalhado na seção 3, é baseado no sistema descrito por Bang and Lee (2018), com dois graus de liberdade, onde a plataforma inclina em relação ao eixo X e ao eixo Y, para movimentar uma bola e usufruir do auxílio de uma câmera para obter as informações da posição dela. O sistema mecânico projetado na Seção 3.2, utiliza os mesmos fundamentos, com duas hastes que conectam os braços dos motores a plataforma para a movimentação da mesma.

2.2.1 Função de transferência do sistema

A função de transferência é a relação entre a entrada e a saída do sistema, Ody (2020) define as seguintes equações para encontrar a função de transferência do plano inclinado, a Figura 3 apresenta as forças aplicadas ao sistema.

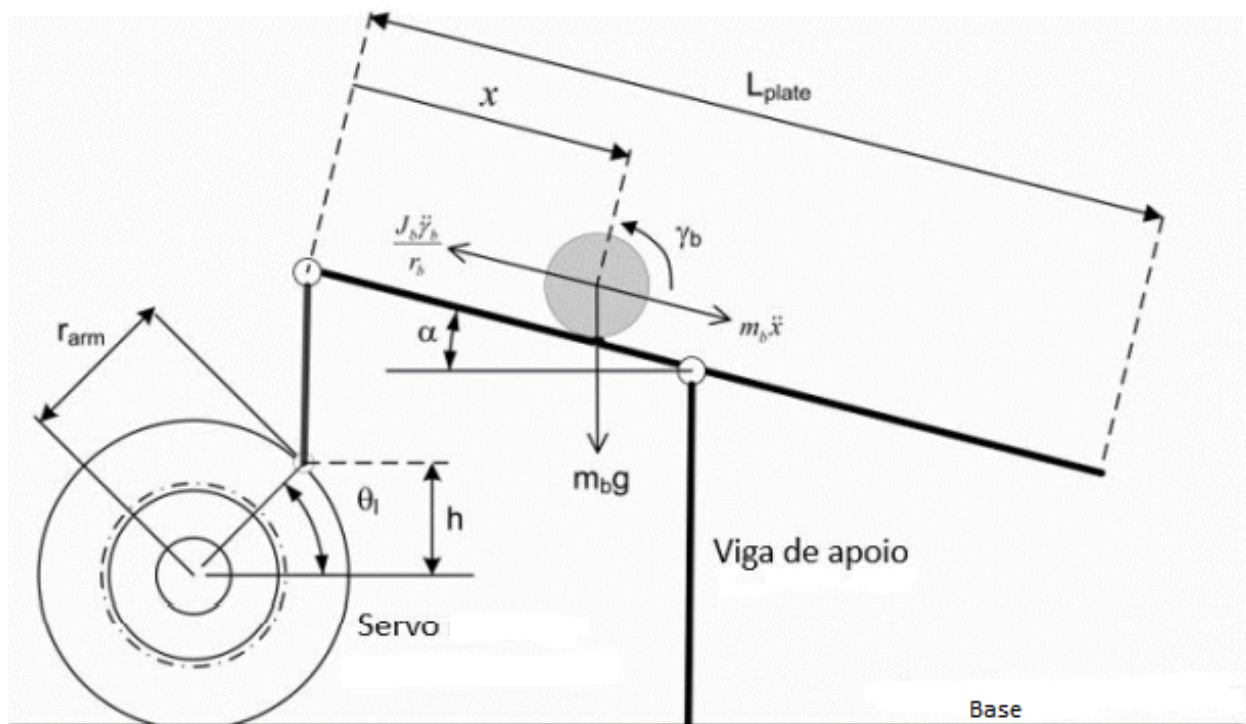


Figura 3 – Diagrama de forças do plano inclinado.

Fonte : Modificado de (Ody, 2020).

A aceleração em função do ângulo da placa, é dada pela equação 2.1, em que x'' é a aceleração, m_b é a massa da bola, r_b é o raio da bola, g é a gravidade, J_b é o momento de inércia da bola, α é o ângulo de inclinação da plataforma.

$$x''(t) = \frac{m_b(r_b)^2 g}{m_b(r_b)^2 + J_b} \text{sen}\alpha(t) \quad (2.1)$$

A relação entre o ângulo da placa e o braço do servo motor é determinado pelas equações 2.2 e 2.3, onde h é a altura entre o braço do servo e a haste que imprime movimento a plataforma, L_{plate} é a largura da plataforma, θ é o ângulo do servo e r_{arm} é o tamanho do braço do servo.

$$\text{sen}\alpha(t) = \frac{2h}{L_{plate}} \quad (2.2)$$

$$\text{sen}\theta(t) = \frac{h}{r_{arm}} \quad (2.3)$$

A relação entre o ângulo da placa e o do servo motor na planta, é determinado pela equação 2.4.

$$x''(t) = \frac{2r_{arm}}{L_{plate}} \frac{m_b(r_b)^2 g}{m_b(r_b)^2 + J_b} \text{sen}\theta(t) \quad (2.4)$$

Para realizar a linearização da equação considera-se que o ângulo do braço do servo se aproxima do ângulo do servo a equação resultante dessa aproximação é a 2.5.

$$x''(t) = \frac{2r_{arm}}{L_{plate}} \frac{m_b(r_b)^2 g}{m_b(r_b)^2 + J_b} \theta(t) \quad (2.5)$$

Realizando a transformada de Laplace da equação 2.5, a equação resultante é a 2.6.

$$\frac{X(s)}{\theta(s)} = \frac{2r_{arm}}{L_{plate}} \frac{m_b(r_b)^2 g}{(m_b(r_b)^2 + J_b)S^2} \quad (2.6)$$

Assumindo que a esfera que rola pela plataforma é maciça, o momento de inércia é determinado pela equação 2.7.

$$J_b = \frac{2m_b(r_b)^2}{5} \quad (2.7)$$

Substituindo a equação 2.7 na equação 2.6 e realizando as devidas simplificações a equação de transferência resultante do sistema é a equação 2.8.

$$\frac{X(s)}{\theta(s)} = \frac{2r_{arm}}{L_{plate}} \frac{5g}{7S^2} \quad (2.8)$$

As equações descritas anteriormente são usadas para auxiliar no cálculo da função de transferência do plano inclinado descrito na seção 3.6, a função de transferência é utilizada nas simulações da planta de controle elaborada para o projeto, que foram utilizadas para verificar diferentes valores do controlador.

2.2.2 Controlador PID

Segundo [Mario Cesar M. M. Campos \(2006\)](#) uma pesquisa feita em mais de 11 mil malhas de controle de refinarias, plantas químicas, indústrias de papel entre outras demonstrou que em 97% delas eram utilizados controladores PID, ainda de acordo com ele isso acontece devido a simplicidade na aplicação do ajuste dos parâmetros.

De acordo com [Ogata \(2011\)](#) mais da metade dos controladores industriais em uso atualmente utilizam controle PID e PID modificado, a utilidade dos controles PID está na sua aplicabilidade geral à maioria dos sistemas de controle, especialmente quando o modelo matemático da planta não é conhecido fazendo assim com que os métodos de projetos analíticos não possam ser utilizados, eles demonstram um controle satisfatório, porém em muitas situações eles não conseguem garantir um ótimo controle.

O controlador PID calcula o erro entre o valor da variável de entrada e o valor definido anteriormente que ela deve atingir. O controlador proporcional integral e derivativo funciona de forma proporcional (P), integral (I) e derivada (D) em relação ao erro do sistema ([Henz, 2017](#)), a Figura 4, mostra o diagrama de blocos de um controlador PID.

Os principais controladores são Controle Proporcional(P), Controle Proporcional e Integral(PI), Controle Proporcional e Derivativo (PD) e Controle Proporcional Integral e Derivativo (PID).

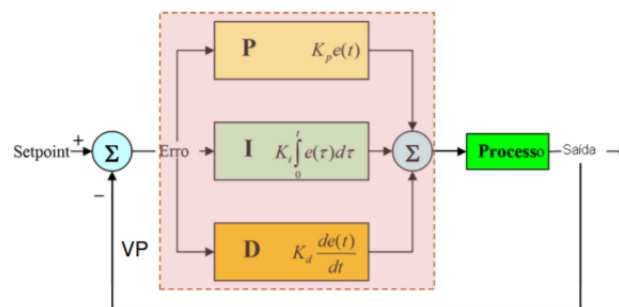


Figura 4 – Diagrama de blocos de um controlador PID

Fonte: ([Bojorge, 2017](#), p.4).

O controlador proporcional usa somente a diferença entre a variável do sistema e o ajuste, o ganho proporcional (K_p) é que define a variação na saída, a equação abaixo descreve como é calculado o ganho proporcional do sistema onde $u(t)$ é a saída, $e(t)$ é o erro

e u_0 é o valor inicial (NI, 2020). O controle proporcional, embora mais simples, geralmente apresenta um erro de estado estacionário significativo.

$$u(t) = K_p \cdot e(t) + u_0 \quad (2.9)$$

O controlador integral usa o somatório do erro ao longo do tempo, com isso a resposta vai mudando com o tempo. A única exceção é quando o erro do sistema é zero. A equação abaixo descreve como é calculado o ganho integral do sistema onde $u(t)$ é a saída, $e(t)$ é o erro que é somado dentro da integral de zero até t (NI, 2020). Com a ação integral é possível reduzir o erro de estado estacionário.

$$u(t) = K_i \int_0^t e(T) dT \quad (2.10)$$

O controlador derivativo faz com que caso a variável do sistema aumente rapidamente a derivada faz a saída decrescer, aumentar o fator do tempo força que o sistema fique mais sensível às alterações do erro acelerando a resposta do sistema, a equação abaixo descreve o cálculo para o ganho derivativo, onde $de(t)/dt$ é a derivada do erro no tempo (NI, 2020).

$$u(t) = K_d \frac{de(t)}{dt} \quad (2.11)$$

O controlador proporcional, integral e derivativo, mais conhecido como PID pode ser descrito pela soma dos controladores de forma individual, resultando na equação abaixo.

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(T) dT + K_d \frac{de(t)}{dt} u_0 \quad (2.12)$$

As equações aqui descritas possuem a finalidade de calcular o controlador feito e exposto na seção 3.5.2.2. As equações são empregadas para o cálculo do erro, diferencial e integral no algoritmo desenvolvido para o microcontrolador, efetuando o controle do plano inclinado.

As equações desenvolvidas para o uso no microcontrolador são detalhadas a seguir.

O erro é calculado utilizando a equação 2.13, subtraindo do *setpoint* o valor da entrada (*input*), depois é calculado a integral conforme equação 2.14, multiplicando o erro pelo tempo atual (dT), lido com a ajuda da biblioteca *Timelib.h*, e somando o resultado aos valores de erros lidos anteriormente.

$$erro = setPoint - input \quad (2.13)$$

$$somaErro+ = erro * dT \quad (2.14)$$

A diferencial é calculada pela equação 2.15, subtraindo o erro atual do valor obtido anteriormente no erro (*lastErro*) e dividindo o resultado pelo tempo atual.

$$diffErro = (erro - lastErro) / dT \quad (2.15)$$

Esses valores são utilizados para calcular o ângulo de saída do motor definido pela equação 2.16, multiplicando o erro pelo KP (coeficiente de ação proporcional), a diferencial do erro pelo KD (coeficiente de ação derivativa) é a soma do erro pelo KI (coeficiente de ação integral), somando essas multiplicações ao ângulo inicial (*anguloinit*) de estabilidade definido para que a plataforma fique reta obtém-se o valor de saída do servo.

$$angulo = anguloinit + (erro * kP + diffErro * kD + somaErro * kI) \quad (2.16)$$

A equação 2.16, é uma equivalência no tempo discreto da equação 2.12.

Onde:

- $erro * kP$ é equivalente a $K_p \cdot e(t)$,
- $somaErro * kI$ é equivalente a $\int_0^t e(T) dT$,
- $diffErro * kD$ é equivalente a $K_d \frac{de(t)}{dt} u_0$.

2.2.3 Exemplos na literatura de controle em plano inclinado

A maioria das implementações de planos inclinado modelo *ball on plate* na literatura consultadas neste trabalho, apresentam o controlador do tipo PID e utilizam como micro-controlador o ATmega, presente na placa de desenvolvimento do Arduino, como os planos inclinados desenvolvidos por (Ody, 2020), (Bang and Lee, 2018) e (Evangelista et al., 2019). Alguns exemplos de implementações são mostrados a seguir.

Evangelista et al. (2019) utiliza para o controle de um plano inclinado modelo *ball on plate* um controlador PID, (Proporcional-Integral-derivativo), com ganhos ajustados para o melhor funcionamento do sistemas e para aquisição da posição da bola é utilizada uma webcam, a Figura 5 mostra a estrutura utilizada neste projeto .

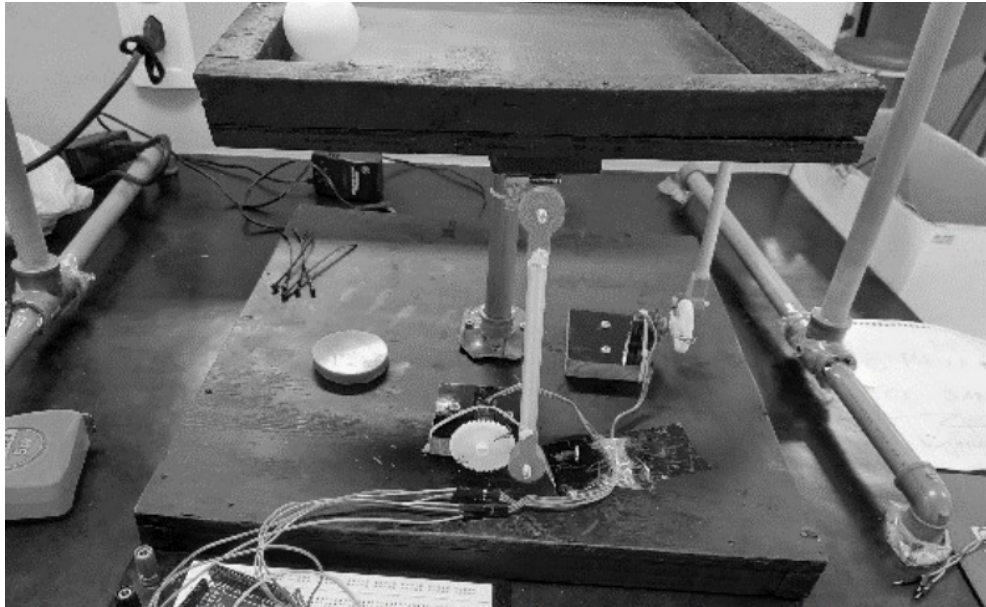


Figura 5 – Estrutura do plano inclinado.

Fonte: (Evangelista et al., 2019, p.3).

Rafael da Silveira Castro (2012) constrói o controle de um plano inclinado utilizando *Linear Quadratic Gaussian* e para o monitoramento do sistema utiliza o método de servovi-são baseado em posição.

Robayo Betancourt et al. (2019) demonstra a construção de projeto de um controlador Fuzzy e um controlador PID aplicado a construção de um plano inclinado utilizando o método do lugar geométrico das raízes e um sistema de interferência lógico.

Tendo por base as implementações aqui apresentadas, elas exibem o Arduino como microcontrolador mais utilizado pela simplicidade da IDE de programação, além de apresentar diversas bibliotecas já desenvolvidas para ele. Utilizam o controlador PID por ser o que tem a maior facilidade de compreensão. Com isso o PID é o controlador escolhido para ser aplicado neste trabalho justamente por ser mais simples e o microcontrolador a *Raspberry Pi Pico* por permitir a utilização da IDE do Arduino para a sua programação.

3 Materiais e métodos

O plano inclinado aqui apresentado é parte do desenvolvimento de um projeto de iniciação científica realizado em conjunto com outros alunos de engenharia eletrônica da faculdade do gama da Universidade de Brasília, para o desenvolvimento de práticas de laboratórios remotos, esse projeto é desenvolvido baseando-se nas transformações que ocorrerão na educação com o advento das transformações ocasionadas pela revolução da Indústria 4.0, conforme apresentado anteriormente na seção 2.1, a revolução acontecerá também na educação .

Durante o desenvolvimento do trabalho foram definidos os seguintes requisitos funcionais:

- O sistema deve permitir medir a posição da bola no plano inclinado (sistema de medição) com um período adequado para o controle (a ser determinado nos testes);
- O sistema deve permitir mudar a posição da bola no plano inclinado (sistema de acionamento) com uma dinâmica adequada para o controle (a ser determinada nos testes);
- Que ambos sistemas estejam disponíveis e funcionais para conectar diferentes plataformas de controle (poder conectar diferentes microcontroladores)

Durante o desenvolvimento do trabalho foram definido os seguintes requisitos não funcionais:

- A estrutura do plano inclinado deve suportar bolas de diferentes pesos e tamanhos ;
- O rastreamento da bola (medição da posição) deve suportar bolas de diferentes cores.

3.1 Proposta de solução

A proposta de solução deste trabalho é desenvolver um sistema de medição para a posição da bola, e um controlador para alterar e controlar a posição da bola na plataforma. Em trabalhos futuros parte do que foi desenvolvido neste trabalho será convertido em um laboratório remoto. Com a finalidade de que os alunos possam acessar os recursos para programação do microcontrolador, nas disciplinas de eletrônica embarcada ou elaborar projeto de controle nas disciplinas de sistemas de controle.

A solução propõe-se a utilizar componentes simples, para que o projeto aqui desenvolvido possa ser replicado por qualquer instituição de ensino superior sem que seja necessário um alto investimento.

Para a elaboração do plano inclinado, são utilizados dois servos motores para imprimir movimentos à plataforma, cada um em um eixo diferente. O posicionamento da bola em cima da plataforma é realizado por processamento de imagem utilizando a *Raspberry Pi*, uma alternativa a essa implementação é a utilização de telas touchscreen resistivas para medir a posição da bola como utilizado por [Bang and Lee \(2018\)](#).

Todos esses sensores e atuadores estão enviando e recebendo informações de um microcontrolador. Para a conexão de todos os componentes é utilizada uma PCB.

Na Figura 6 é exposto o diagrama de blocos do plano inclinado. Nele é possível ver todas as etapas de comunicação dos componentes necessários para a prática.

A *Raspberry Pi* é a responsável por parte da comunicação entre o aluno e o laboratório. Ela é a responsável por permitir o acesso de forma remota ao experimento e transmitir as imagens em tempo real, além de processar a posição da bola.

Conforme a Figura 6, o aluno terá acesso a *Raspberry Pi* através de uma *VPN*, e usará um *site web* para acessar a transmissão da prática, para programação do microcontrolador é empregado um USB virtual. O acesso a *VPN*, ao *site web* e a programação do microcontrolador por um USB virtual são desenvolvidos por outros alunos de iniciação científica.

O microcontrolador se comunica com os servos pelas portas digitais e a posição da bola é enviada ao microcontrolador por comunicação serial *UART*.

O desenvolvimento deste trabalho contempla medir a posição da bola no plano inclinado, movimentar a bola ao longo da plataforma, mostrar um exemplo de código para microcontrolador e definir a comunicação serial entre o microcontrolador e a *Raspberry Pi*. As demais partes mostradas na Figura 6, são realizadas por outros alunos.

A Tabela 1 apresenta a lista dos componentes, bem como a quantidade de cada um utilizado na construção do plano inclinado. Neste trabalho não é utilizado o acelerômetro para medir a inclinação da plataforma, pois apenas o rastreamento da bola na plataforma foi suficiente para a implementação do sistema.

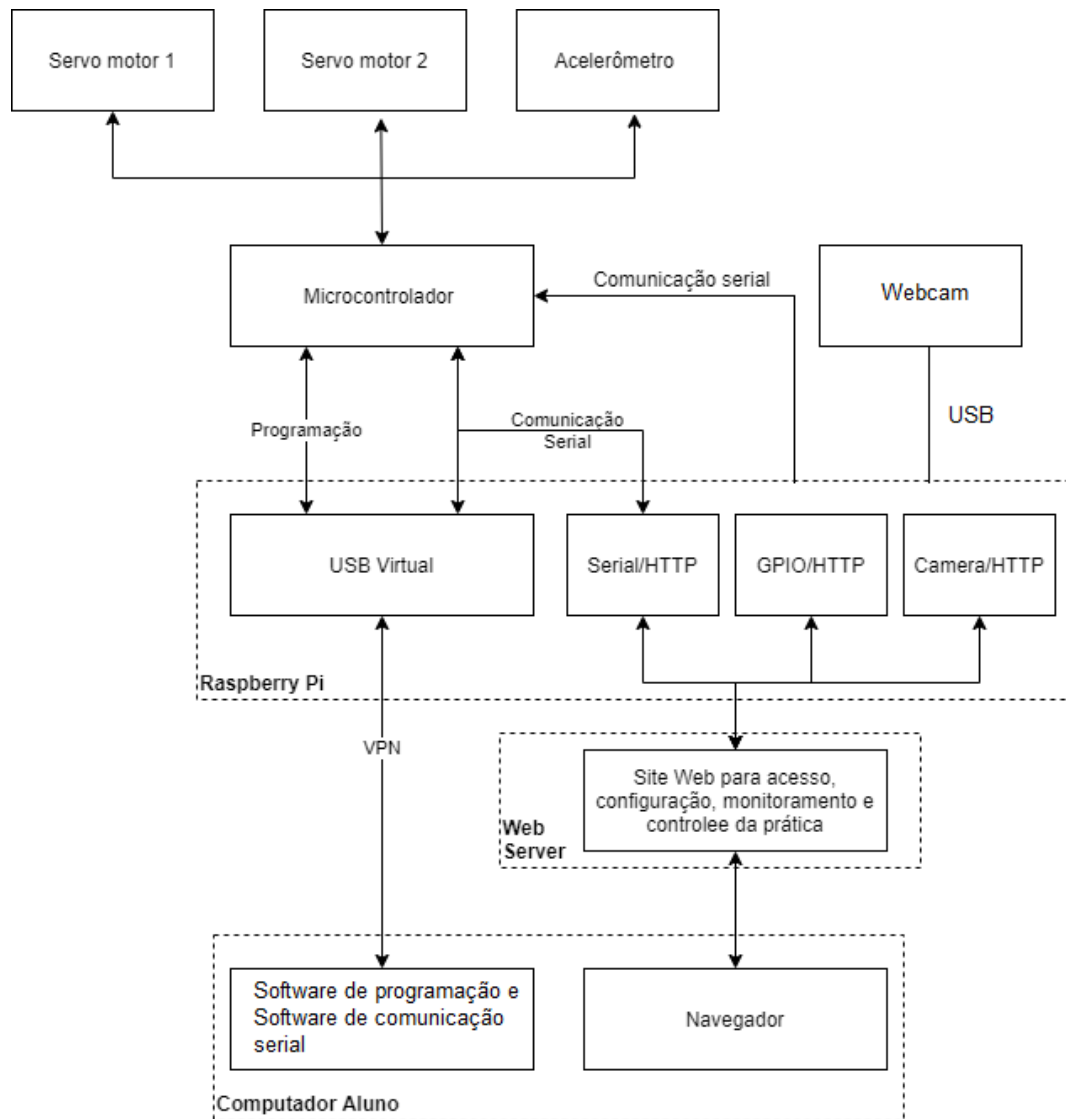


Figura 6 – Diagrama de blocos do laboratório remoto

Fonte: Elaboração Própria.

Tabela 1 – Lista de componentes

Componente	Modelo	Quantidade
Acelerômetro	MPU6050	1
Servo motor	Futaba S3003	2
Microcontrolador	Raspberry Pi Pico	1
Raspberry Pi	3	1
Webcam	LEY-52	1

Fonte: Elaboração Própria.

A Tabela 2 descreve as especificações elétricas de cada componente do projeto, com a tensão de alimentação e a intensidade de corrente necessária.

Tabela 2 – Especificações elétricas dos componentes

Componente	Tensão elétrica de alimentação (V)	Intensidade da corrente [mA]
Acelerômetro	3.3	3.6
Servo motor	5	7.2
Microcontrolador	5	95
Raspberry Pi	5	500
Webcam	5	0.25

Fonte: Elaboração Própria.

3.2 Sistema mecânico

A estrutura para o laboratório obrigatoriamente deve ser pequena para reduzir o espaço de armazenamento. O desenho mecânico foi feito no software *SolidWorks*, a estrutura tem altura total de 52.5 centímetros e largura de 30 centímetros.

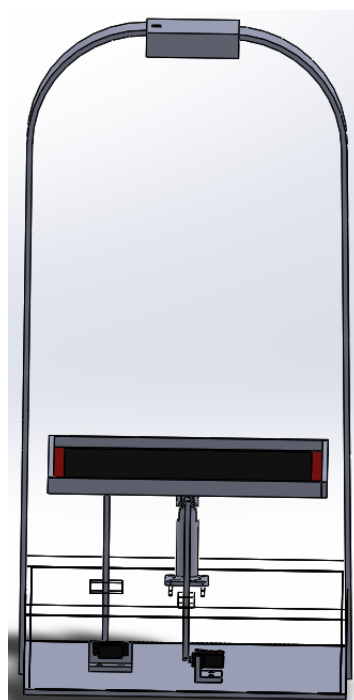


Figura 7 – Desenho mecânico vista frontal

Fonte: Modificado de (Sobrinho, 2019).

Como pode ser observado na Figura 7, os servos motores, o microcontrolador e a placa ficarão na base da estrutura, no topo da estrutura localiza-se a câmera, a base (mostrada na figura em preto) responsável por segurar a bola, tem 25 centímetros de largura para evitar que a bola escape foi colocado limitadores nas 4 laterais, eles possuem 1,5 centímetros de altura cada, eles oferecem também um reforço na estrutura. A movimentação da plataforma é possível por conta do eixo móvel que liga a base da estrutura a ela, a distância entre a plataforma e a câmera é de 39,40 centímetros.

As Figuras 8 e 9 apresentam as vistas laterais e isométricas respectivamente, para mais informações no anexo A estão os desenhos mecânicos com as especificações de cada parte da estrutura. A estrutura final foi feita em acrílico de 5mm de espessura e como o projeto não exige grandes esforços mecânicos a estrutura oferece resistência suficiente para o que foi projetado.

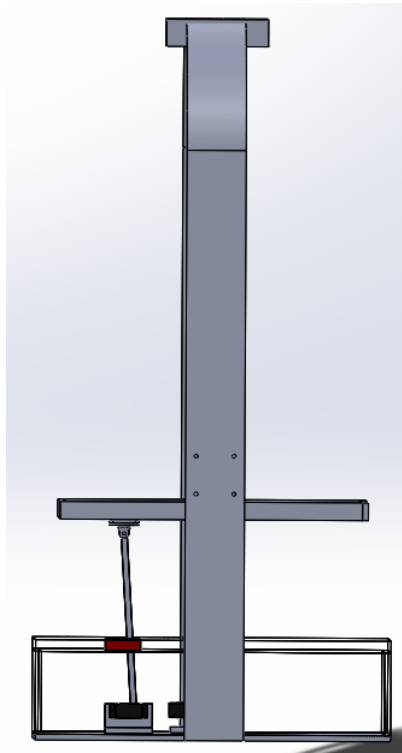


Figura 8 – Desenho mecânico vista lateral

Fonte: Modificado de (Sobrinho, 2019).

As duas hastes, que podem ser vistas na Figura 9, conectam os braços dos servos motores a base da plataforma e servem para imprimir movimento a mesma. A parte superior dos braços laterais possui um suporte para a câmera que recebe a imagem da bola, o braço central é móvel e permite que a plataforma se movimente em diferentes direções conforme é visto no Anexo B.

A restrição na movimentação é de 30 graus para mais ou para menos em cada eixo, assumindo o ângulo inicial do servo em 90 graus, valores acima disso podem ocasionar a derrubada da bola da plataforma ainda que existam limitadores nas laterais.

O desenho mecânico da estrutura exibido acima e a estrutura desenvolvida neste trabalho são baseados no plano inclinado apresentado por Bang and Lee (2018) na Seção 2.2 e por Evangelista et al. (2019) na Seção 2.2.3.

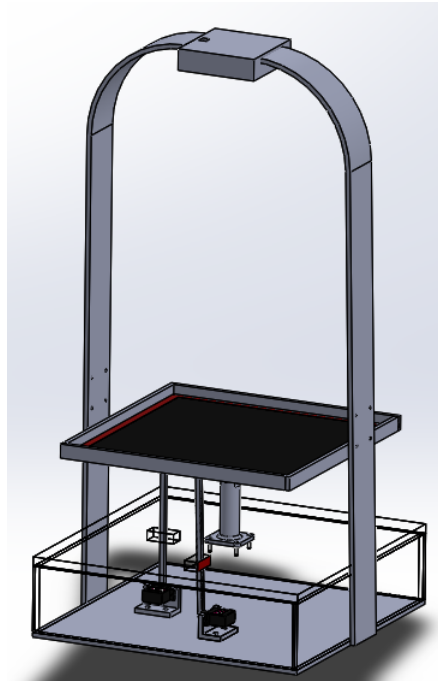


Figura 9 – Desenho mecânico vista isométrica

Fonte: Modificado de (Sobrinho, 2019).

3.3 Sistema de medição da posição da bola

A detecção da posição da bola é feita por processamento de imagem, usando uma *Raspberry Pi*, como a da Figura 10. A câmera da *Raspberry Pi* capta a imagem da bola em cima da plataforma. A bola é rastreada por detecção de cor utilizando a biblioteca *OpenCV* (OpenCV, 2021).

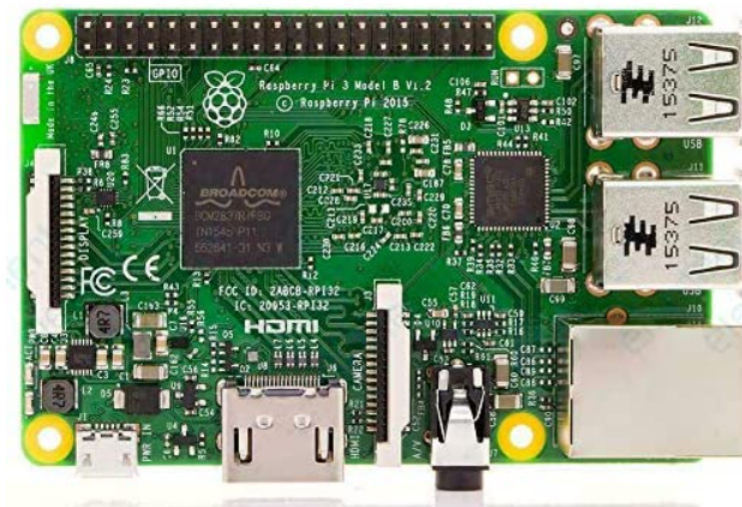


Figura 10 – Raspberry Pi.

Fonte: (Pi, 2022a)

A Figura 11, mostra a estrutura do plano inclinado, o suporte para a câmera fica localizado na parte superior, é feito de cano PVC, nessa parte é colocada a *webcam* para obter a imagem do posicionamento da bola. A *Raspberry Pi* fica na base, recebe as imagens da webcam, faz o processamento das imagens e envia as coordenadas para a *Raspberry Pi Pico*.

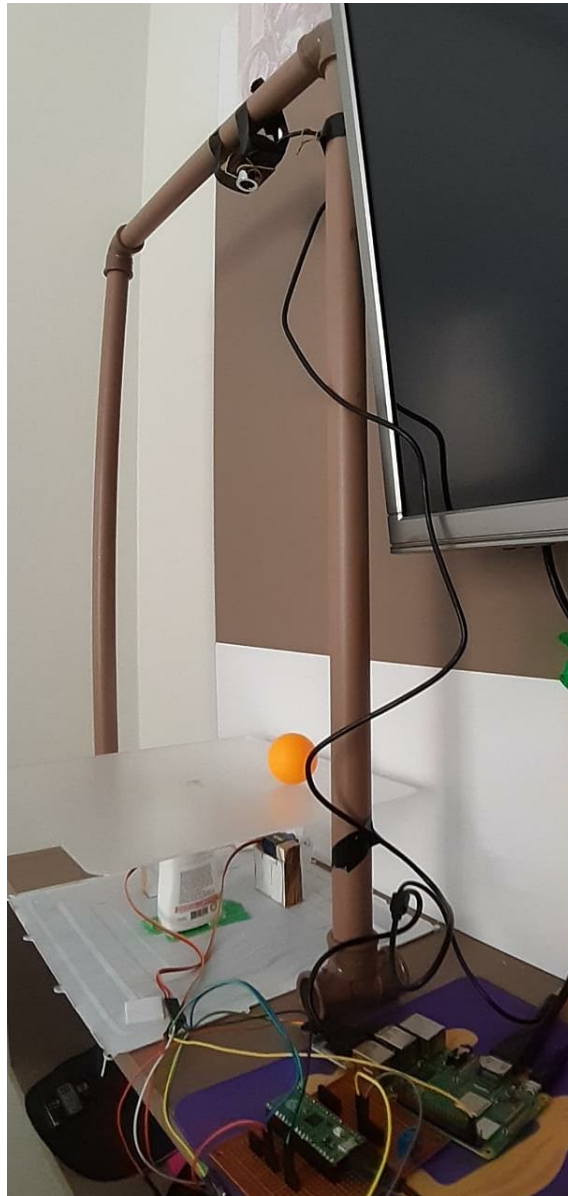


Figura 11 – Estrutura com a webcam conectada a Raspberry Pi.

Fonte: Elaboração Própria.

3.3.1 Algoritmo para medir a posição da bola

O *OpenCV* é uma biblioteca para *softwares* de visão computacional e aprendizado de máquina de código aberto, ela oferece mais de 2500 algoritmos para diversas finalidades como detecção de rostos, identificação de objetos, rastrear objetos em movimento, encontrar imagens semelhantes de um banco de dados de imagens, remover olhos vermelhos de ima-

gens tiradas com flash, seguir os movimentos dos olhos, reconhecer cenários entre diversas outras. Possui interfaces em *C++*, *Python*, *Java* e *MATLAB* além de suportar diferentes sistemas operacionais como, *Windows*, *Linux*, *Android* e *Mac OS* (OpenCV, 2022).

O código utilizado aqui foi desenvolvido por Tropicic (2021), o propósito do código é rastrear objetos por detecção de cor ele usa a biblioteca *OpenCV* para encontrar os contornos de objetos na imagem, usando os limites superiores e inferiores no espaço de cores, sendo usado como um range de detecção, o código é escrito em *Python*, ele é implementado na *Raspberry Pi* usando o *software Thonny*.

A Figura 14, detalha o fluxograma de como é feito o rastreamento da bola na plataforma. O código implementado, funciona através da captura de um frame. Em seguida transforma as cores desse frame de *RGB* para as do espaço de cores *HSV* (matiz, saturação e valor) usando a função *ColorBGR2HSV*, aplica uma máscara com os *thresholds* de cores desejados, por exemplo laranja, conforme pode ser visto na Figura 12, onde um objeto dessa cor foi encontrado na máscara. Posteriormente encontra os contornos do objeto laranja na imagem, usando a função *FindContours*, como pode ser visto na Figura 13.



Figura 12 – Limiar de cor encontrado no frame com resolução de 480 X 480.

Fonte: Elaboração Própria.

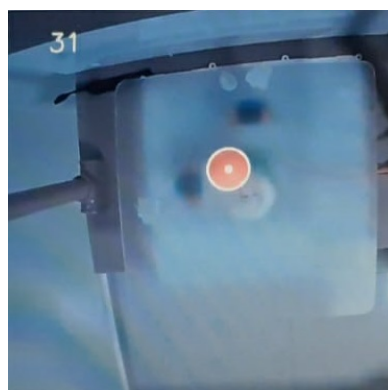


Figura 13 – Contorno encontrado no rastreamento com resolução de 480 X 480.

Fonte: Elaboração Própria.

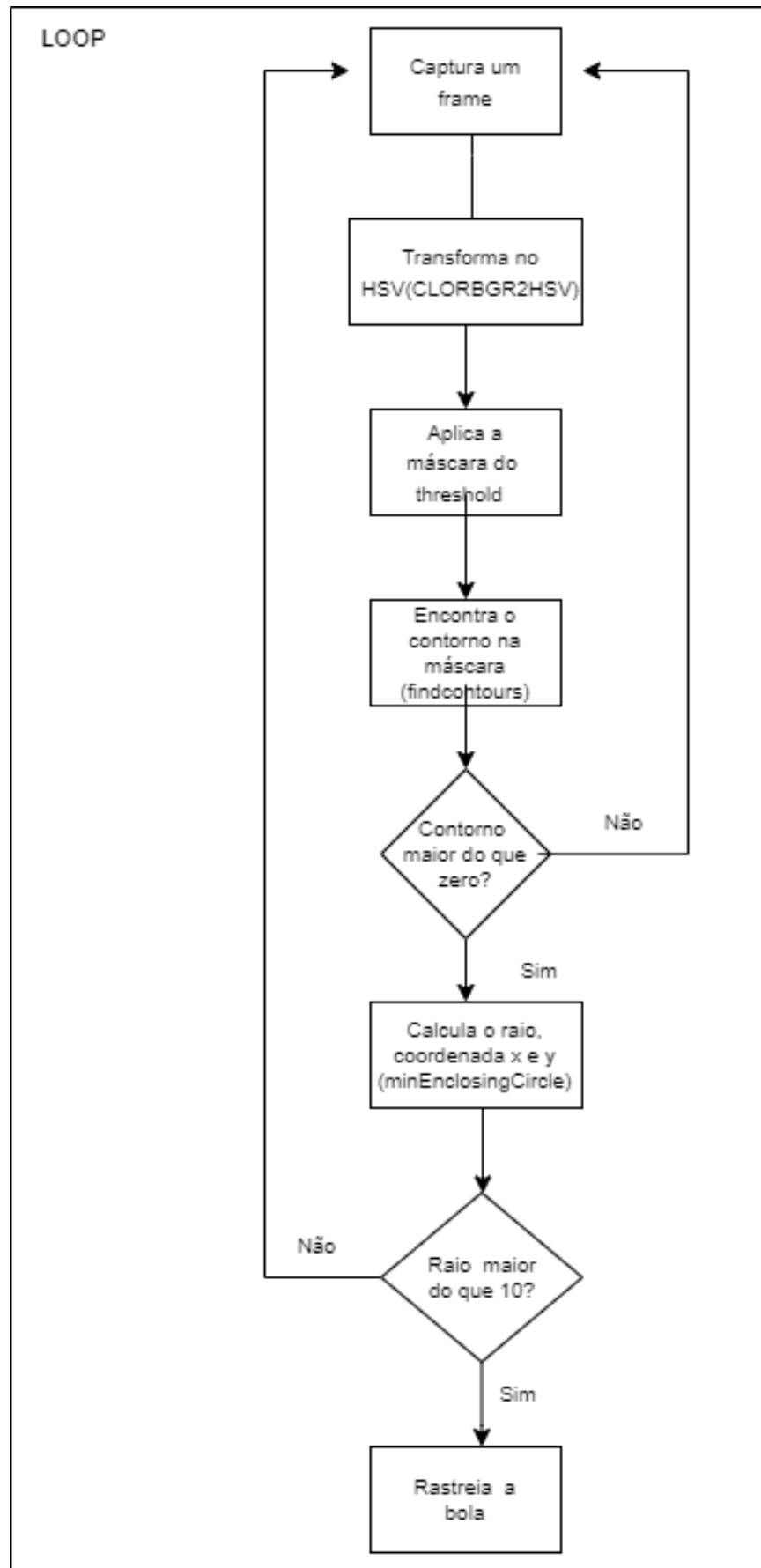


Figura 14 – Fluxograma do código do rastreo da bola.

Fonte: Elaboração Própria.

Se o contorno do objeto encontrado no frame for maior do que zero, isso indica que ele encontrou um objeto laranja no frame, passando a calcular o raio e as coordenadas com a função *minEnclosingcircle*, para que o rastreamento seja realizado o raio mínimo tem que ser maior do que dez milímetros, o frame tem resolução de 480 x 480, com pouco mais de 230 mil pixels.

Um problema que pode ocorrer ao implementar o código é a divergência no limiar de cor do objeto a ser rastreado. Para melhorar o rastreamento e fazer com que o objeto seja rastreado corretamente pode ser realizado o ajuste manual dos valores desejados de limiares da matiz, de saturação, e do valor mínimo e máximo.

Após o objeto ser detectado e o rastreamento ser iniciado, as coordenadas da posição da bola são enviadas pela serial *UART*.

O RS232 é um protocolo de comunicação assíncrona, ele envia dados em uma única direção, sem a necessidade de um *clock* para sincronizar o transmissor e o receptor. O envio de dados é iniciado com um *bit* inicial, seguido por dados binários de 8 *bits*, *bit* de paridade e *bit* de parada, que são enviados um após o outro (Swaroop, 2020).

A comunicação *UART* aqui aplicada tem taxa de transmissão de 19200 *bits* por segundo, sendo enviado pela *Raspberry Pi* um *byte* ao microcontrolador sem bit de paridade. É necessário apenas um *byte* para enviar as coordenadas pela *UART*.

O padrão de envio é exibido na Figura 15, são enviados pela *Raspberry Pi* um *byte*. Do *bit* zero ao *bit* 2 são enviados o valor da coordenada de X, o *bit* 3 é o separador que é uma vírgula, do *bit* 4 ao *bit* 6 são os valores da coordenada Y, e o *bit* 7 é o final do envio representado pelo barra n, os valores são enviados em formato de *String* codificados em *ASCII*.

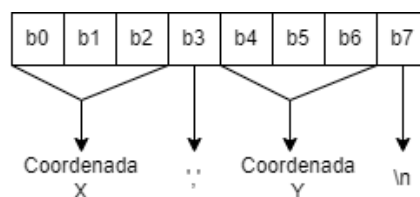


Figura 15 – Quadro de comunicação.

Fonte: Elaboração Própria.

O diagrama das conexões para a comunicação *UART* é apresentado na Figura 16. A *Raspberry Pi* transmite através do pino físico de número 8 (GPIO 14), de TX (transmissão), que deve ser conectado ao pino físico de número 7 (GPIO 5), de RX (recepção) do microcontrolador os valores das coordenadas. Os pinos de transmissão e recepção estão no nível de 5V. Os pinos de terra devem ser ligados em um terra comum.

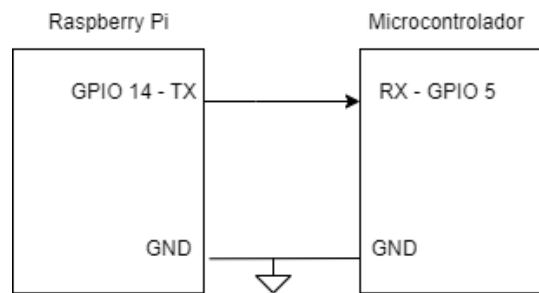


Figura 16 – Diagrama das conexões da comunicação UART.

Fonte: Elaboração Própria.

As únicas alterações realizadas no código do Tomislav Tropic para a implementação do rastreamento da bola foram, acrescentar o envio das coordenadas pela serial *UART*, conforme a Figura 15 e retirar a parte do processamento do movimento dos motores que na implementação dele era realizada no mesmo código em *Python*. O *link* para o repositório com o código utilizado pode ser encontrado em: <https://github.com/UnB-Projeto-Industria-4-0/UnB_industria4.0_plano_inclinado/tree/main/Rastreamento>.

3.4 Sistema de acionamento

"O termo "servo" refere-se ao mecanismo de controle. Em termos de tecnologia, o mecanismo que realiza o controle é denominado "mestre" e o mecanismo que está sendo controlado é denominado "escravo". Ambos os termos, "servo" e "escravo" derivam de "servus", a palavra latina para escravo" (Aspina, 2020, p.1) de forma simplificada os servos motores fazem exatamente o que é ordenado.

O servo motor no plano inclinado tem a função de aplicar todo movimento desejado a base para que a bola que está em cima dela se movimente e realize as mudanças que foram determinadas pelos usuários, o servo motor foi escolhido justamente por realizar movimentos de alta eficiência e precisão sem a necessidade de um driver para controlá-lo.

O microcontrolador irá enviar um sinal digital que determina quantos graus ele deve se mover, os graus são definidos após a aplicação do controle. O ângulo de saída do servo motor não deve ser maior do que 25°, pois ângulos maiores do que esses podem acabar derrubando a bola da base, na Figura 17 é exibido o servo motor.

Os servos aplicam o movimento na plataforma através de uma haste conectada no braço do motor e na parte inferior da plataforma, algo similar ao que pode ser visto na Figura 3.

O tempo que o servo leva para atingir a posição desejada é na ordem de milissegundos (ms), de acordo com o pulso enviado para ele, variando o tempo entre 15 e 25 milissegundos (Straub, 2016), como pode ser visto na Figura 18.



Figura 17 – Servo motor

Fonte: (Kits, 2021)

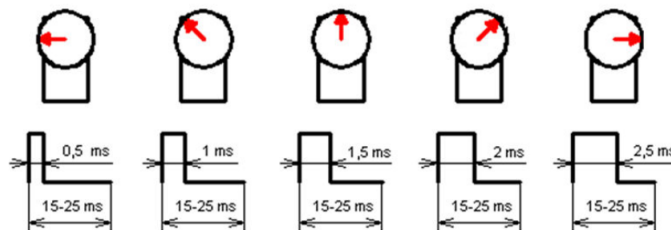


Figura 18 – Funcionamento do servo motor.

Fonte: (Straub, 2016)

O servo motor funciona com modulação por largura de pulso, em que o pulso define o quanto o servo deve movimentar. O pulso recebido pelo servo é um sinal enviado pelo microcontrolador que nesse caso é o proporcional ao ângulo gerado pelo código do microcontrolador a ser detalhado na Seção 3.5.2.1.

Tomando por base o ângulo inicial do servo que é fixado em 96 graus o servo pode movimentar entre 71 e 121 graus, com o menor grau de movimento possível sendo igual a 0.1, a largura do pulso de saída (*PWM*) é estabelecida pela equação 2.16, que é calculada pelo microcontrolador, em que basicamente é definido o tempo em que o sinal a ser enviado para o servo deve ficar em nível lógico alto.

A Figura 19, mostra a comunicação entre o microcontrolador e o servo. No microcontrolador são utilizados dois pinos, os *GPIOs* 20 e 21 para funcionarem como as saídas dos *PWMs* para os dois eixos, eles devem ser conectados aos pinos de sinais de cada um dos servos motores. O sinal de saída possui resolução de 250 valores diferentes.

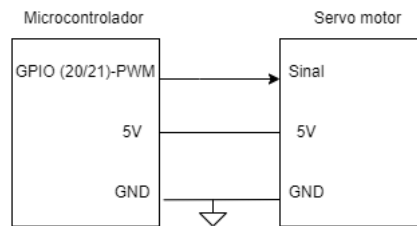


Figura 19 – Comunicação entre o microcontrolador e o servo motor.

Fonte: Elaboração Própria.

3.5 Sistema de controle

A seguir é apresentado a placa onde todos os componentes estarão conectados e o controlador responsável por controlar a aquisição de dados de posição e acionamento dos motores.

3.5.1 Placa do circuito

A placa desenvolvida para a conexão dos componentes do plano inclinado pode ser vista na Figura, 20, ela possui 3 saídas para servomotores, com saídas comuns de *GPIO*, *VCC* e *GND*, uma saída para comunicação *UART*, e uma saída para comunicação *I2C* ou *SPI*, que nos pinos da *Raspberry Pico* são os mesmos, os pinos centrais são para a conexão da *Raspberry Pi Pico*.

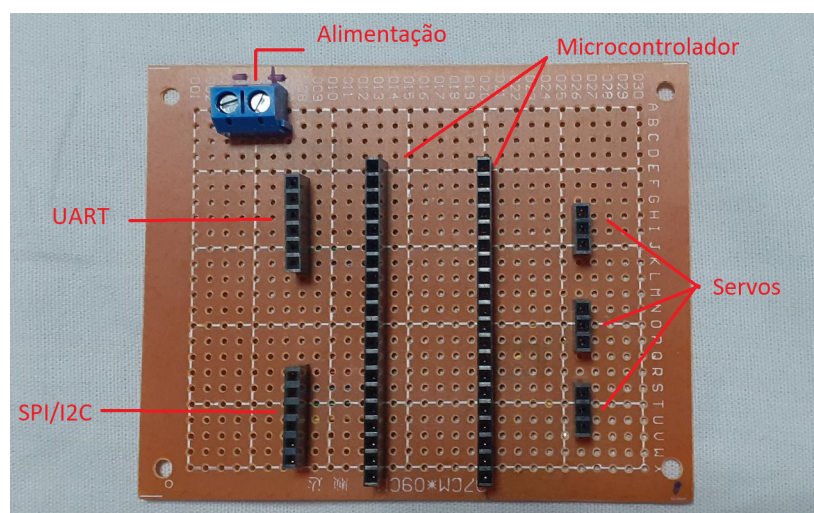


Figura 20 – Placa para conexão dos componentes

Fonte: Elaboração Própria.

A placa conta ainda com alguns pinos extras em caso de futuras adaptações do projeto e alimentação externa para os componentes. O esquemático do desenho eletrônico de

todos os componentes pode ser visto na Figura 21, ele foi desenvolvido no *software Altium Designer*.

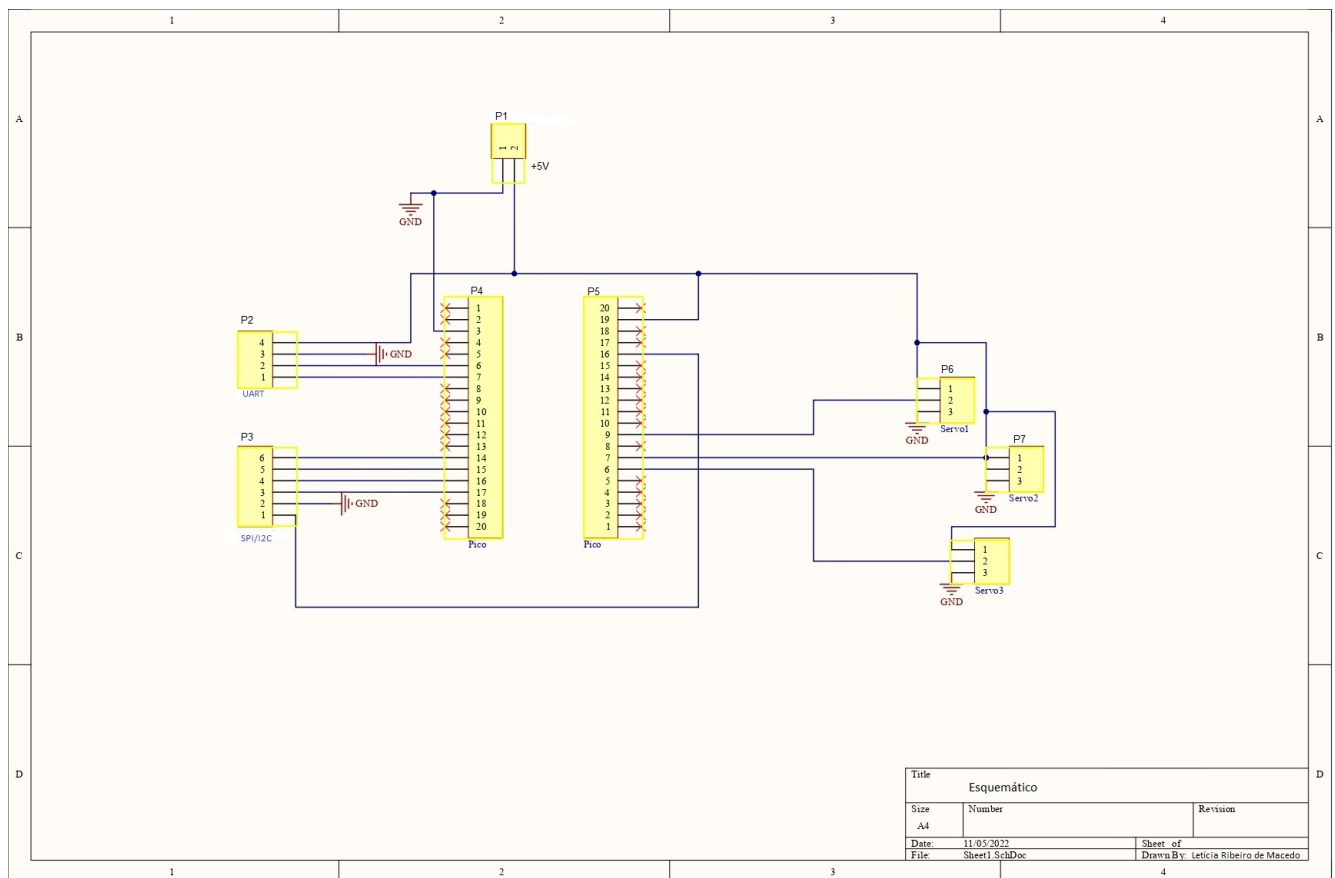


Figura 21 – Esquemático do circuito

Fonte: Elaboração Própria.

3.5.2 Microcontrolador

Conforme (Kerschbaumer, 2018, p.10) microcontroladores são circuitos integrados que possuem em seu interior todos os componentes necessários ao seu funcionamento dependendo unicamente da fonte de alimentação externa. Pode-se dizer que os microcontroladores são computadores de um único *chip*. Os microcontroladores diferem dos sistemas tradicionais por já terem os seus periféricos dentro do próprio componente.

Um microcontrolador é definido como "um circuito integrado programável que contém toda a estrutura (arquitetura) de um microcomputador, isto é, dentro de um microcontrolador podemos encontrar uma *CPU* (Unidade Central de Processamento), memória *RAM*, memória *EEPROM*, portas de entrada/saída." (Nobuo Oki, 2013, p.1). Eles são dispositivos com *clock*, memória de programa, memória *RAM*, podem ter diversos periféricos como conversor analógico/digital, *PWM*, interface *SPI*, *I2C*, *UART* (Pereira, 2008). A Figura 22 mostra os componentes presentes em um microcontrolador.

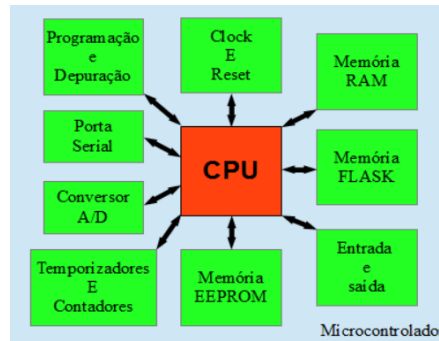


Figura 22 – Componentes do microcontrolador

Fonte: (Kerschbaumer, 2018, p.10)

O microcontrolador é o responsável pela leitura dos dados de todos os sensores presentes no sistema e a tomada de decisões em relação aos valores lidos. Ele é o componente que une todos os componentes em vista disso ele irá processar a leitura da posição da bola proveniente da *Raspberry Pi* e irá indicar qual deverá ser o deslocamento dos servo motores.

O modelo selecionado foi a *Raspberry Pi Pico*, lançada em 2021, que pode ser vista na Figura 23, a *Raspberry Pi Pico* foi escolhida por ter microcontrolador RP 2040 *ARM dual core*, *clock* de 133 MHz, memória *RAM* de 256 KB, memória *FLASH* de 2MB, com duas comunicações *SPI*, duas *I2C*, duas *UART*, três conversores *ADC* de 12 bits, 40 pinos de *GPIO*, tem a versatilidade de poder ser programada tanto em *C++* como em *Micro Python* (Pi, 2022b), a Figura 23 também apresenta os pinos da placa.

Essas configurações de tamanho de memória e diferentes tipos de comunicações são importantes para o plano inclinado que utiliza a *UART* para ler os dados de coordenada e enviar os valores de saída dos servos. Para adaptações futuras as configurações também são importantes no projeto, pois permitem escrever programas com mais instruções do que o que foi desenvolvido aqui, além de permitir a adição de outros componentes para o auxílio da implementação como por exemplo o acelerômetro para medir a inclinação da plataforma, entre outros.

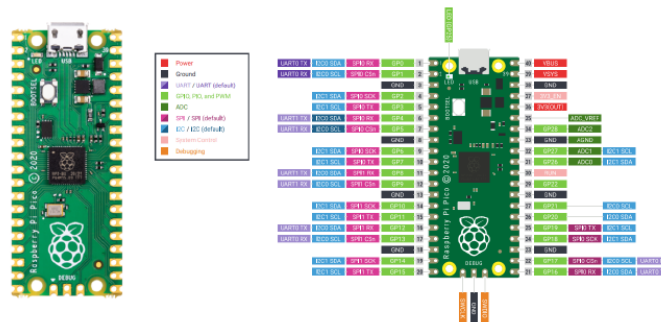


Figura 23 – Raspberry Pi Pico

Fonte: (Pi, 2022b)

O exemplo de código desenvolvido para a implementação do plano inclinado pode ser acessado pelo *link* : <https://github.com/UnB-Projeto-Industria-4-0/UnB_industria4.0_plano_inclinado>, o fluxograma de funcionamento dele é descrito nas Figuras 24 e 25.

Para o desenvolvimento do código foi utilizada a IDE do Arduino, ela possui compatibilidade com a *Raspberry Pi Pico* após a instalação da placa *Arduino Mbed OS RP2040 Boards* na IDE. A preparação da IDE pode ser vista no Anexo C.

3.5.2.1 Algoritmo para aquisição de dados e controle dos motores

O código desenvolvido para a implementação do plano inclinado pode ser visto no *link*: <https://github.com/UnB-Projeto-Industria-4-0/UnB_industria4.0_plano_inclinado>, o fluxograma dele é descrito nas Figuras 24 e 25.

O código do microcontrolador para a tomada de decisões inicia com a declaração das variáveis, em seguida é inicializado o *Baud Rate*(taxa de transmissão de dados) da comunicação serial e os motores são ligados às portas selecionadas.

O *loop* principal inicializa com a aquisição de dados da serial, com a função *Serial.available*, caso ela seja diferente de zero significa que algum valor chegou por ela, os valores que chegam são lidos com a função *Serial.readStringUntil(separador).toFloat()*;; que lê os dados até o separador definido no envio, que para o primeiro dado foi uma "," e o segundo dado o " n" e são convertidos para float.

Após a leitura eles são armazenados como as coordenadas, em seguida a função PID é chamada para calcular o ângulo de saída dos motores, o algoritmo da função é melhor detalhado na seção 3.5.2.2. Após o cálculo dos ângulos de saída os valores encontrados são escritos nos servo, usando a função *servo.write(angulo)* para isso é utilizada a biblioteca *Servo.h*.

A *Servo.h* é uma biblioteca desenvolvida para o controle de servo motores utilizando o Arduino mas que pela compatibilidade da placa *Raspberry Pico* com a IDE do arduino pode ser usada na programação da mesma.

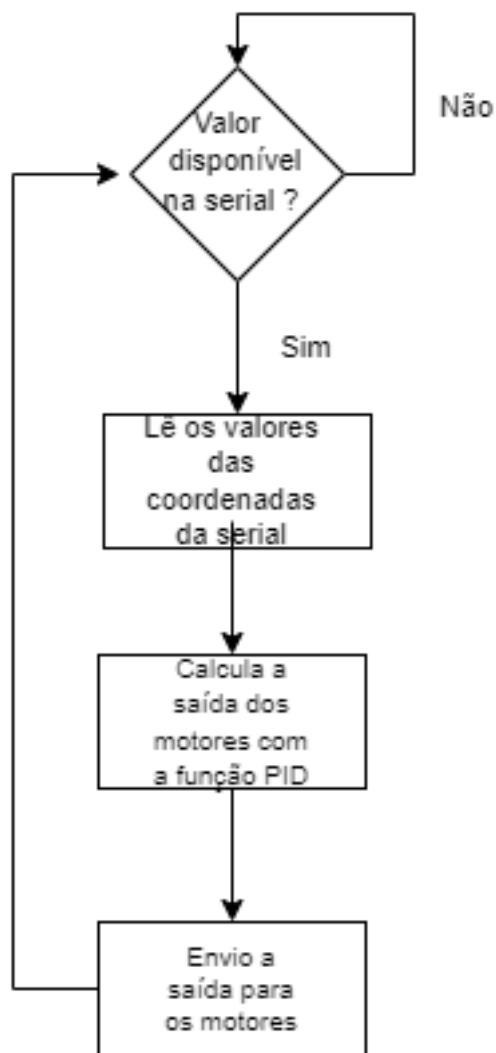


Figura 24 – Fluxograma do código do microcontrolador

Fonte: Elaboração Própria.

3.5.2.2 Algoritmo para controle PID

O algoritmo de controle PID é realizado dentro do microcontrolador em forma de uma função denominada PID, que é chamada logo após a leitura dos valores das coordenadas da serial. Ela realiza os cálculos necessários para determinar o ângulo de saída dos motores dos eixos X e Y, determinando assim a movimentação da bola em cima da plataforma.

Para tal feito é realizada a equivalência da equação 2.12 apresentada na Seção 2.2.2, e que serão discutidas a seguir.

A lógica da função para o cálculo do PID de cada um dos eixos, começa com a leitura do tempo atual (dT) utilizando a biblioteca *TimeLib.h*, o tempo é utilizado para auxiliar no cálculo da integral, da derivada e do ângulo de saída. A leitura do tempo é feita com a finalidade de aproximar os cálculos dos coeficientes de erro das equações definidas na Seção

2.2.2.

Após a leitura do tempo atual, é realizado o cálculo do erro definido pela equação 2.13 entre o *setpoint* e a coordenada. Em seguida o erro é somado no tempo com a equação 2.14.

A diferencial no tempo é calculada com a equação 2.15. Seguidamente é calculado o ângulo de saída do motor, com a equação 2.16, nela o ângulo inicial do motor é incrementado ao somatório da multiplicação do erro pelo KP com a soma do erro vezes o KI com a diferencial do erro vezes o KD.

O valor é verificado logo depois, a fim de aferir se o ângulo ultrapassa o limite de 25° para mais ou para menos.

A Figura 24, mostra o fluxograma tanto para o eixo X quanto para o eixo Y, do código desenvolvido para o controle da plataforma.

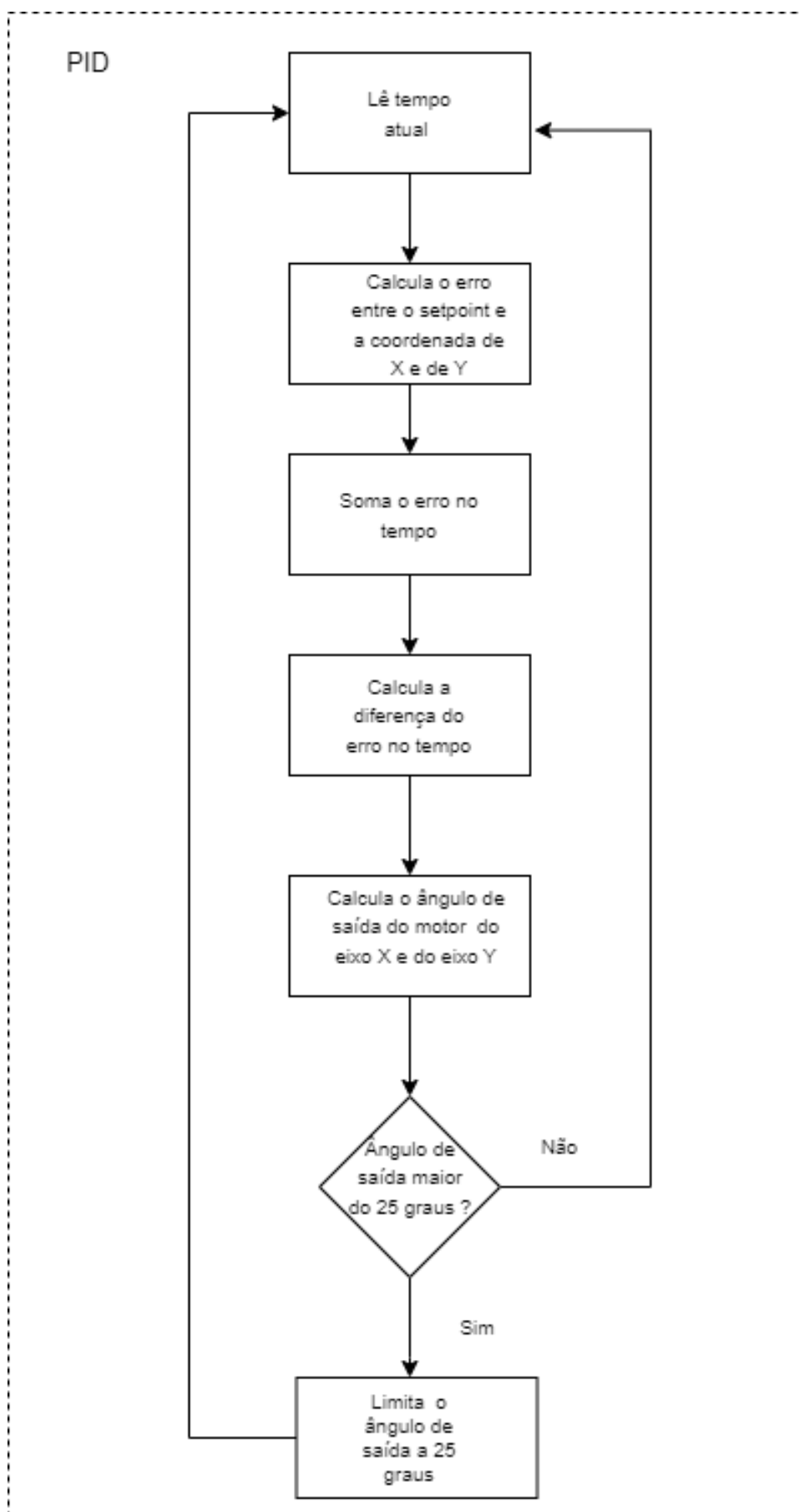


Figura 25 – Fluxograma do código do PID

Fonte: Elaboração Própria.

3.6 Projeto de controle

O plano inclinado é um sistema de controle multivariável, o controle tem que ser feito em duas variáveis, tem que ser controlada a inclinação da plataforma no eixo X e a inclinação no eixo Y e com isso as saídas controladas são os ângulos dos dois motores.

O controle mais clássico que pode ser aplicado a sistemas multivariáveis e que são mais utilizados na indústria é o controle PID, conforme descrito anteriormente por Ogata (2011), as principais estratégias dele são:

- Controlador proporcional;
- Controle proporcional e integral;
- Controle proporcional e derivativo;
- Controle proporcional integral e derivativo.

O controle construído para o plano inclinado é um controlador independente para cada variável, cada uma possui uma malha separada que não interage com a outra, o ideal é a construção de um controlador que leve em consideração as interações entre as malhas porém esse é um escopo fora do conteúdo deste trabalho. O controle é realizado em cada eixo sozinho, sendo perceptível os distúrbios causados por um eixo no outro.

A função de transferência do plano inclinado foi descrita anteriormente no tópico 2.2.1, tomando por base a equação definida para o plano desenvolvida por Ody (2020), a função de transferência do plano inclinado é a equação 2.8, considerando que o $rarm$ é 3 cm, o $Lplate$ é 24 cm, e a gravidade g é de $9,81 \text{ m/s}^2$, a equação resultante para o modelo aqui definido fica:

$$\frac{X(s)}{\theta(s)} = \frac{1,75}{s^2} \quad (3.1)$$

A Figura 26, mostra a planta de controle desenvolvida para o plano inclinado. É a mesma planta para os dois eixos, mudando apenas a variável a ser controlada em cada um dos controladores. No eixo X a saída do controlador U caracteriza o controle da posição da bola em X, e no eixo Y a saída do controlador U caracteriza o controle da posição da bola em Y. O controlador escolhido foi o PID após a realização de testes no sistema e simulação.

O erro é calculado pelo setpoint definido menos o valor da posição atual da bola na plataforma. Em seguida é calculado o ganho aplicado ao erro em P, I e D, somado esses valores e aplicados ao sistema do plano inclinado, que foi definido anteriormente na equação 3.1. O processo se repete em cada período de controle.

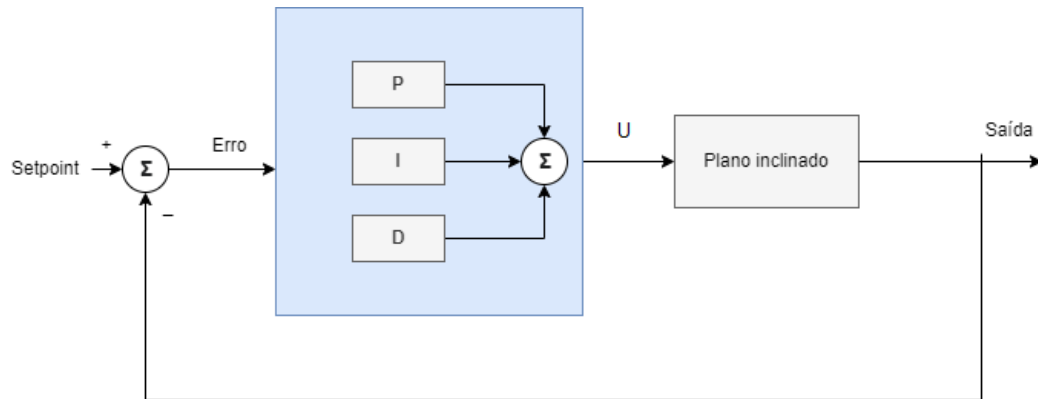


Figura 26 – Planta para o controle PID

Fonte : Elaboração Própria.

Para o ajuste dos valores de PID foi utilizado o método de tentativa e erro, começando com valores baixos para P, I e D e em seguida aumentando os valores até chegar no que melhor funcionou para o sistema. Começando os aumentos pelo P, depois I e por último D. O critério de parada foi definido pela resposta do sistema, se os valores escolhidos não causassem o efeito esperado de aproximar a bola do setpoint, eram trocados.

A fim de verificar o comportamento da planta e a escolha de qual controlador (P, PD, PI ou PID) utilizar na implementação deste trabalho, foi realizada a simulação utilizando o *software SciLab*, simulando uma resposta ao degrau unitário. A simulação serviu para verificar os valores de PID que posteriormente foram testados na implementação do plano inclinado.

A Figura 27 mostra a montagem da planta no *software*, os valores de PID são diferentes para cada uma das simulações.

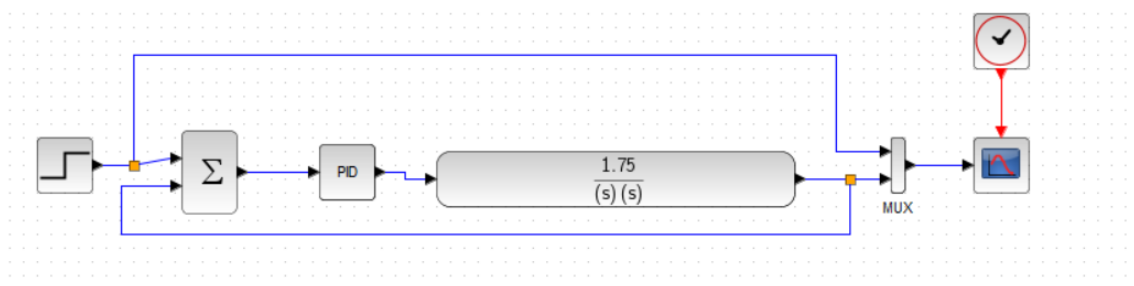


Figura 27 – Planta para o controle simulado no SciLab

Fonte : Elaboração Própria.

A Figura 28, simula um controlador P, ele apresenta uma saída instável, o valor atribuído a P, para a simulação foi de $P= 0.02$, o controlador tem uma saída senoidal e não chega a estabilidade.

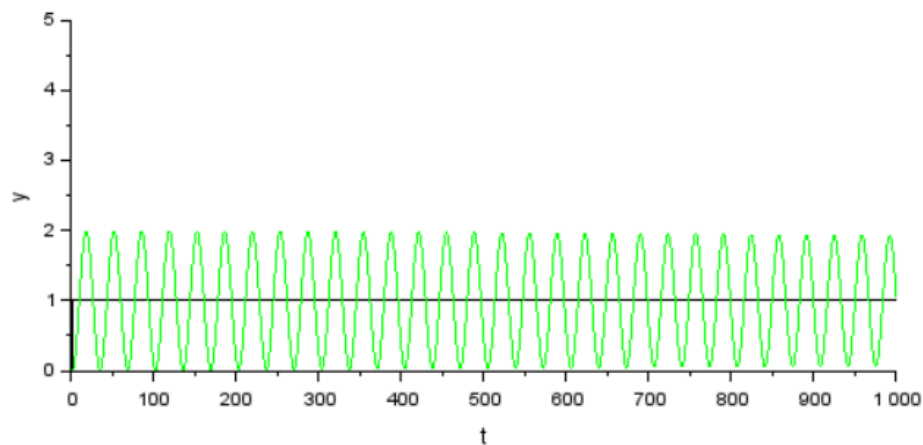


Figura 28 – Simulação no SciLab controlador P

Fonte : Elaboração Própria.

A Figura 29, simula um controlador PD, o valor atribuído foi de $P=0.5$ e $D = 0,3$. Ele apresenta um *overshoot* bem alto do *setpoint*, e chega a estabilidade com o t igual a 30.

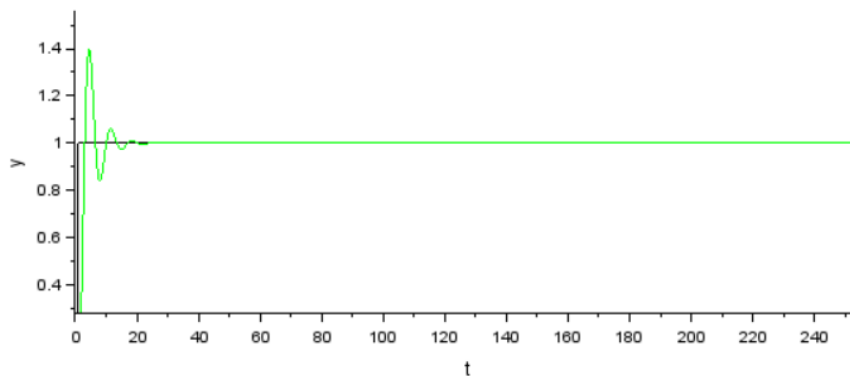


Figura 29 – Simulação no SciLab controlador PD

Fonte : Elaboração Própria.

A Figura 30, simula um controlador PI, com $P = 0.3$ e $I = 0,001$. A saída dele é uma senóide. Ele apresenta um sistema instável pois o ganho proporcional tem um valor alto.

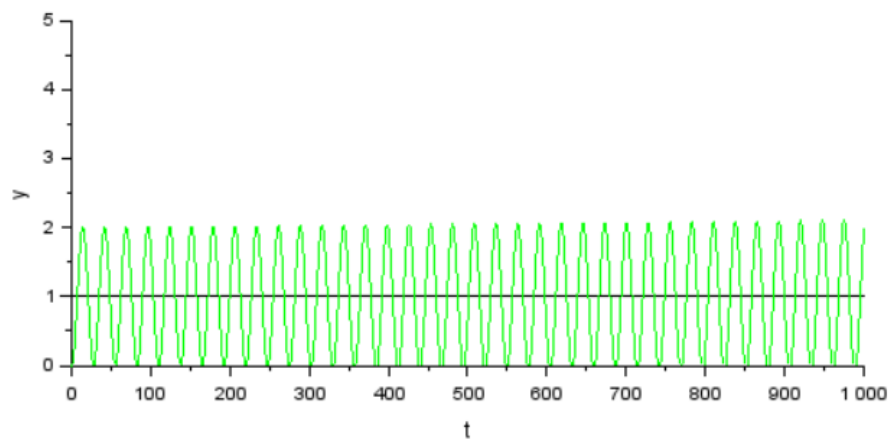


Figura 30 – Simulação no SciLab controlador PI

Fonte : Elaboração Própria.

A Figura 31, simula um controlador PID, com $P = 0.30$, $I = 0.01$, e $D = 0.5$, ele apresenta um *overshoot* do *setpoint* mas é um bem mais baixo do que o controlador PD e em seguida chega a estabilidade com o t igual a 25.

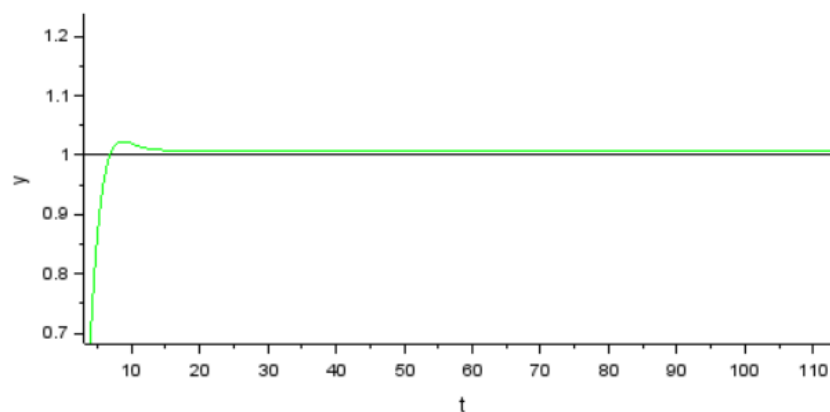


Figura 31 – Simulação no SciLab controlador PID

Fonte : Elaboração Própria.

A simulação atestou a funcionalidade do controlador PID na planta do plano inclinado e permitiu definir valores aproximados das constantes do PID para começar o ajuste do controlador na prática.

O controlador escolhido para ser utilizado no plano inclinado aqui desenvolvido foi o PID, apesar dele possuir um tempo de acomodação um pouco alto, nos testes práticos o PID se adaptou melhor aos requisitos do projeto.

O controle é feito por *software*, o fluxograma do código utilizado para calcular o ângulo de saída do motor é descrito na Figura 25. Tomando o valor atual da posição da bola em

X e Y e comparando com o valor do *setpoint* definido para cada variável, com isso é calculado o valor do erro, os valores para PID são definidos no código.

O valor da saída para os motores é calculado a partir do ângulo base para estabilidade da plataforma que por ajuste foi definido em 96° para o eixo X e 96° para o eixo Y. O cálculo do ângulo de saída dos motores é feito usando a equação 2.16, o erro é calculado na equação 2.13, a integral do erro é calculada na equação 2.14 e a diferença do erro é calculada na equação 2.15, essas equações são equivalentes as equações 2.9, 2.10, 2.11, 2.12 (NI, 2020).

4 Resultados

Com o intuito de caracterizar a solução proposta e testar o sistema de medição e de acionamento, aqui serão apresentados os resultados obtidos na implementação do plano inclinado, para a realização dos testes foi utilizada a estrutura apresentada na Figura 32.



Figura 32 – Estrutura de testes.

Fonte: Elaboração Própria.

4.1 Medição da posição bola

Os primeiros testes foram realizados no rastreamento da bola ao longo da plataforma, pois o rastreamento é a parte principal para o funcionamento do sistema, sem a posição da bola não é possível controlar nada.

Foram realizados cinco testes no rastreamento da bola como pode ser visto na Tabela 3. No primeiro teste não foi viável rastrear a bola pois o *threshold* de cores para a cor da bola testada que era laranja não estava configurado corretamente. Após o ajuste, para os valores corretos o rastreamento funcionou bem em todas as demais tentativas.

Tabela 3 – Testes realizados na medição da posição da bola.

Teste	Rastreou
1	Não
2	Sim
3	Sim
4	Sim
5	Sim

Fonte: Elaboração Própria.

A Figura 33 mostra o resultado do rastreamento da bola. A captura tem resolução de 480 x 480, com pouco mais de 230 mil *pixels*.

É perceptível que o contorno ao redor dela compreende perfeitamente a circunferência da bola, exibindo o ótimo funcionamento do código que faz o rastreamento. Bolas de diferentes cores podem ser utilizadas no projeto, definindo as cores desejadas do espaço de cores no código e ajustando assim os limiares para rastreamento.

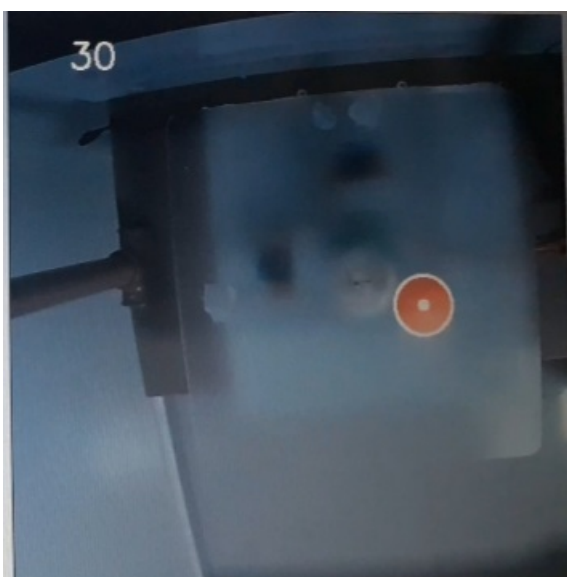


Figura 33 – Rastreamento da posição da bola

Fonte: Elaboração Própria.

A resolução da medição é de 3 milímetros. Pelos testes executados e que podem ser conferidos na Tabela 4, o erro na medição da posição da bola, pode variar em até dois milímetros para mais ou para menos. Considerando que a bola usada nos testes possui tamanho

de 30 milímetros, o erro máximo obtido nos testes foi de 2 milímetros, e a resolução da medição é de 3 milímetros o erro na medição está dentro do esperado, já que o erro não vai fazer com que o motor seja acionado sem necessidade

Para obtenção dos valores do raio foi utilizada a função *Print* do *Python* na *Raspberry Pi*, solicitando o envio dos valores do posição atual, para o terminal executor do programa.

Tabela 4 – Testes de erro na medição da posição da bola.

Teste	Posição real (mm)	Posição medida (mm)	Erro (mm)
1	417	416	1
2	235	235	0
3	302	301	1
4	277	279	2
5	386	385	1

Fonte: Elaboração Própria.

4.2 Controle da posição do plano inclinado

A fim de captar os dados para a análise, do controle da plataforma, os testes foram feitos colocando a bola em diferentes partes da plataforma e o valor definido do *setpoint* foi o mesmo para todos os testes.

Para realizar as medições dos valores da posição da bola ao longo da plataforma, foi utilizada a função *Print* do *Python* na *Raspberry Pi*, separando os valores das posições de X e de Y com vírgulas, os valores eram armazenados no terminal do executor do programa. Após a finalização do teste os valores foram transferidos para o *Excel* para tratamento dos dados e em seguida plotar os gráficos.

Os resultados obtidos com o controlador ajustado são apresentados nas Tabelas 5 e 6, elas indicam o erro entre o *setpoint* e a coordenada, ele foi calculado efetuando a diferença entre o valor do *setpoint* e o valor da coordenada. Foram considerados como acertos quando a bola ficava com um erro entre a coordenada e o *setpoint* de valor menor ou igual a 8 milímetros, o que corresponde a um critério de estabilidade de 3 %.

Para os testes aqui realizados o critério de estabilidade definido é entre 3 % e 5 %.

Considerando os dez testes realizados, a coordenada X possui uma taxa de falha (testes em que o erro entre a coordenada e o *setpoint* foi maior do que 8 milímetros), um pouco menor do que a coordenada Y.

A taxa de falha da coordenada X fica em 20 % dos testes totais realizados, de dez testes dois não atingiram o *setpoint*, enquanto que a coordenada Y apresenta 30% de falha, dos dez testes realizados em três o *setpoint* não foi atingido.

Tabela 5 – Testes realizados no eixo X.

Tentativa	Setpoint X (mm)	Coordenada X (mm)	Erro (mm)
1	300	304	4
2	300	311	11
3	300	307	7
4	300	298	2
5	300	305	5
6	300	300	0
7	300	297	3
8	300	291	9
9	300	299	1
10	300	300	0

Fonte: Elaboração Própria.

Tabela 6 – Testes realizados no eixo Y.

Tentativa	Setpoint Y (mm)	Coordenada Y (mm)	Erro (mm)
1	236	248	12
2	236	234	2
3	236	228	8
4	236	226	10
5	236	236	0
6	236	229	6
7	236	239	3
8	236	237	1
9	236	225	11
10	236	241	5

Fonte: Elaboração Própria.

Para que a estabilidade do sistema fosse atingida os valores de PID encontrados são apresentados na Tabela 7, os valores foram obtidos através de diversos testes com diferentes valores, modificando os valores de P, I e D e testando o funcionamento do sistema, além da realização da simulação apresentada na Seção 3.6.

Os valores de PID encontrados na simulação não foram utilizados como os valores finais de PID da estrutura, pois não apresentaram o mesmo comportamento. Os valores de PID foram definidos os mesmos para os dois eixos, após os testes na estrutura, os valores encontrados mudavam apenas o valor de I que ficava em 0,019, por simplificação ficou os mesmo valores para ambos os eixos.

Tabela 7 – Valores de PID.

P	I	D
0.570	0.020	0.625

Fonte: Elaboração Própria.

A Figura 34 mostra a posição da bola no *frame* que captura coisas ao redor da plataforma e por isso a trajetória não coincide exatamente com o centro do *plot*.

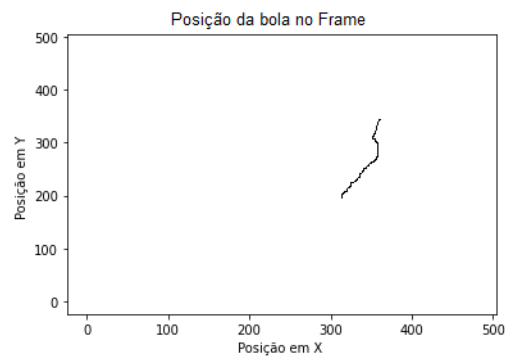


Figura 34 – Posição um da bola em milímetros no frame de 480X480.

Fonte: Elaboração Própria.

A Figura 35 mostra o controle aplicado ao eixo X, para a posição obtida de X na Figura 34. Nela é possível verificar a posição, o *setpoint*, e o erro, em milímetros, o ângulo de saída do motor em graus ao longo do tempo. É observável que ao longo da movimentação da bola pela plataforma o erro vai diminuindo chegando quase a zero.

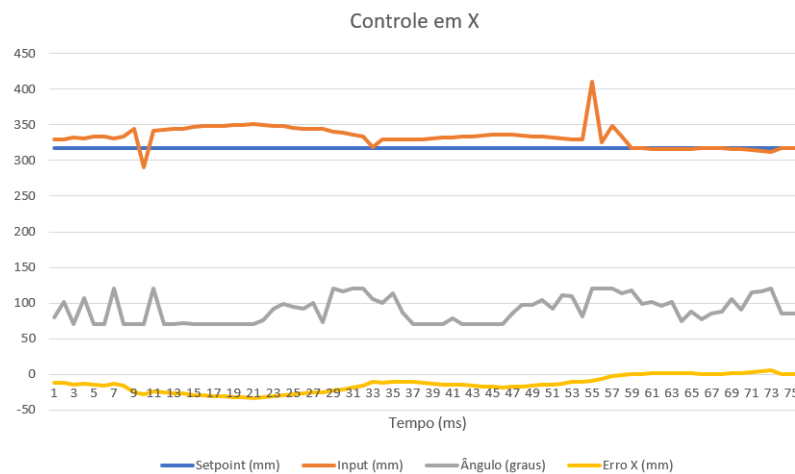


Figura 35 – Controle um da bola em X na plataforma.

Fonte: Elaboração Própria.

A Figura 36 expõe o controle aplicado ao eixo Y, para a posição obtida de Y na Figura 34. Nela, é possível averiguar a posição, o *setpoint*, e o erro em milímetros, o ângulo de saída do motor em graus ao longo do tempo. É observável que ao longo da movimentação da bola pela plataforma o erro vai diminuindo chegando quase a zero.

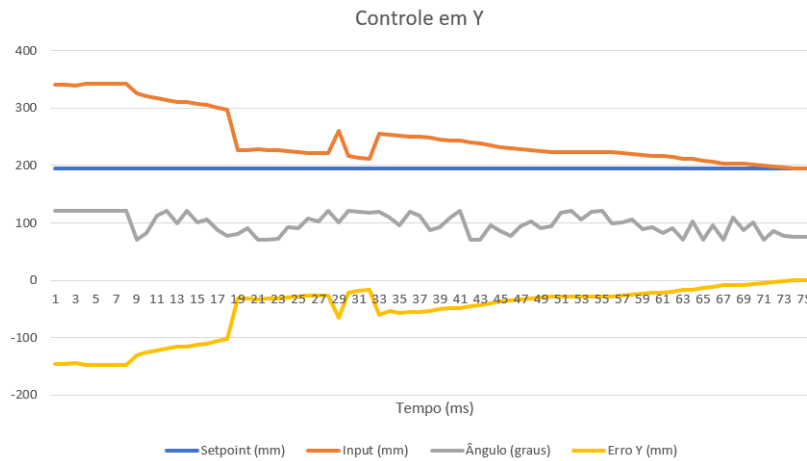


Figura 36 – Controle um da bola em Y na plataforma.

Fonte: Elaboração Própria.

O erro quadrático médio é definido pela equação 4.1, aqui ele é usado para calcular a variação do erro em estado estacionário do controle.

$$EQM = \sqrt{\frac{1}{n} \sum_{i=1}^n (f_i - o_i)^2} \tag{4.1}$$

Usando a equação 4.1 para calcular o erro quadrático médio do controle em X e em Y apresentados nas Figuras 35 e 36. O valor obtido para X é $Erro_x = 2.626$ o valor obtido para Y é $Erro_y = 5.504$.

As Figuras 37 e 34 exibem a bola lançada de diferentes pontos para chegar ao *setpoint*. Elas expõem a posição da bola ao longo do frame, no primeiro caso a bola levou 1,4 segundos para chegar ao setpoint enquanto que no segundo foi aproximadamente 0,75 segundos.

A Figura 37 mostra o caminho percorrido pela bola em relação aos eixos X e Y com o controle PID aplicado em cada eixo separadamente. Após aproximadamente 1,4 segundos a bola chegou ao centro da plataforma, atingindo o um valor próximo ao *setpoint* definido.

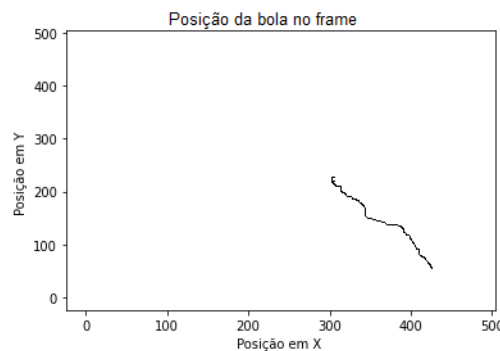


Figura 37 – Posição dois da bola em milímetros no frame de 480X480

Fonte: Elaboração Própria.

A Figura 38 expõe o controle aplicado ao eixo X, para a posição obtida de X na Figura 37. Nela é possível averiguar a posição, o *setpoint*, e o erro em milímetros, e o ângulo de saída do motor em graus ao longo do tempo. É observável que ao longo da movimentação da bola pela plataforma o erro vai diminuindo chegando quase a zero.

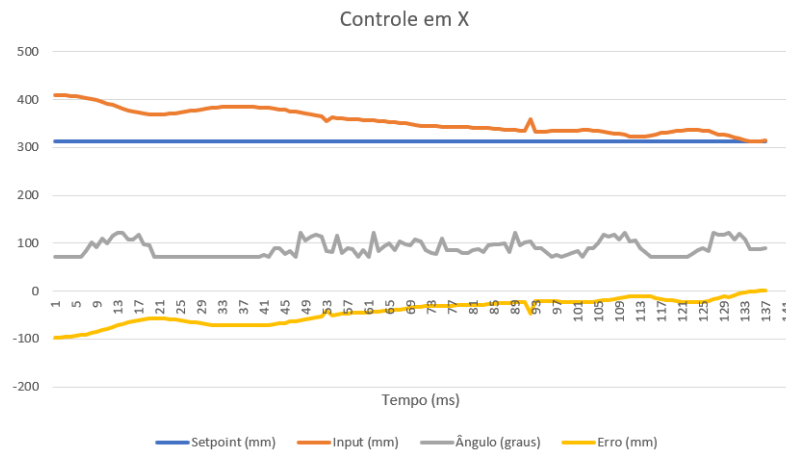


Figura 38 – Controle dois da bola em X na plataforma.

Fonte: Elaboração Própria.

A Figura 39 expõe o controle aplicado ao eixo Y, para a posição obtida de Y na Figura 37. Nela é possível averiguar a posição, o *setpoint*, e o erro em milímetros, e o ângulo de saída do motor em graus ao longo do tempo. É observável que ao longo da movimentação da bola pela plataforma o erro vai diminuindo chegando quase a zero.

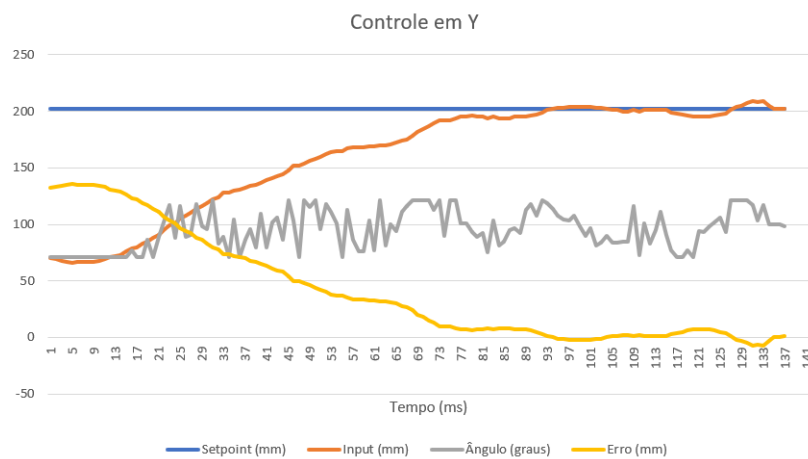


Figura 39 – Posição dois da bola em Y na plataforma.

Fonte: Elaboração Própria.

Usando a equação 4.1 para calcular o erro quadrático médio do controle em X e em Y apresentados nas Figuras 38 e 39. O valor obtido para X é $Erro_x = 5.465$ o valor obtido para Y é $Erro_y = 4.596$.

Os valores de erro quadrático médio, medidos para as posições encontradas nos dois testes apresentam um valor satisfatório de variação. O valor perfeito seria zero, o máximo aqui obtido foi 5.504, que pode ser considerado um valor bastante aceitável.

Comparando as diferentes posições da bola apresentadas nas Figuras 34 e 37, com a bola iniciando a trajetória em diferentes posições para chegar ao *setpoint* definido, o valor de PID encontrado para a estabilidade funciona bem para as diferentes posições de início selecionadas, as trajetórias não são lineares pois existem interações entre as malhas de controle que causam perturbações na movimentação da bola pela extensão da plataforma.

5 Conclusões

Os resultados obtidos no rastreamento da bola comprovam o bom funcionamento do código utilizado, a detecção por cor funciona não somente para objetos na cor laranja ou que apresentam um grande contraste entre as cores da bola e da plataforma, desde que os limites do espaço de cores estejam bem delimitados o rastreamento por detecção de cor é possível de ser realizado.

Em relação a manipulação dos atuadores para a movimentação da bola, o controle é eficiente para a implementação do plano inclinado, a construção é feita com materiais de baixo custo que permitem ser replicado em outras universidades. Mostram que o projeto aqui desenvolvido é uma opção simples e barata de aumentar atividades práticas nas disciplinas de engenharia. Vale ressaltar que o projeto aqui apresentado faz parte de um projeto de iniciação científica com o objetivo de desenvolver laboratórios remotos.

A estabilidade do sistema foi atingida usando o controlador PID, controlando a posição da bola na plataforma, foram feitos vários testes e simulações ajustando os ganhos do controlador para que a saída do sistema tivesse a maior estabilidade possível. A aplicação do controle PID em um sistema como esse é uma forma simplificada de demonstrar o principal controlador usado na indústria.

Analisando os resultados e a construção do sistema é possível afirmar que o projeto aqui desenvolvido demonstra possíveis ganhos no aprendizado dos alunos em sistemas de controle e também em sistemas embarcados, ao incentivar o desenvolvimento da planta de controle e do software embarcado para o microcontrolador.

O projeto é flexível permite a troca dos componentes por outros similares, utiliza plataformas de livre acesso para ser construído viabilizando sua adaptação a diferentes usos em universidades.

Algumas melhorias que podem ser realizadas futuramente é a utilização de atuadores que possuem uma precisão maior e mais potentes, a aplicação de outros modelos de controladores, aplicando controladores que levam em consideração as interações entre as malhas de controle e conversão do projeto em um laboratório completamente remoto.

Referências

; . Nenhuma citação no texto.

Alkhalidi T, Pranata I, Athauda RI. A review of contemporary virtual and remote laboratory implementations: observations and findings. *Journal of Computers in Education* 2016;3(3):329–351. Citado na página 13.

Antonio M, Carvalho GDE, Tulha C. Uso de Laboratórios Remotos na educação a distância no Brasil : uma revisão sistemática The Use of Remote Laboratories and e-learning in Brazil : a systematic review 2019;(i):195–209. Citado na página 13.

Aspina. What is a servo motor? 2020;<<https://www.aspina-group.com/en/learning-zone/columns/what-is/004/>>, *Online*, acessado em 29 de abril de 2021. Citado na página 34.

Bang H, Lee YS. Implementation of a Ball and Plate Control System Using Sliding Mode Control. *IEEE Access* 2018;6(8):32401–32408. Citado 5 vezes nas páginas 17, 18, 22, 25 e 28.

Bojorge N. Sintonia de Malhas de Controle 2017;<http://www.professores.uff.br/ninoska/wp-content/uploads/sites/57/2017/10/Control_Aula16_Sintonia_2sem17-1.pdf>. Citado na página 20.

Britanica Teoe. Industrial Revolution. *Encyclopedia Britannica* 2021;<<https://www.britannica.com/event/Industrial-Revolution.>>, *Online*, acessado em 10 de abril de 2021. Citado na página 13.

Castro RdS, Barth JMO, Flores JV, Salton AT. Modelagem e Implementação de um Sistema Ball and Plate Controlado por Servo-Visão. XI Simpósio Brasileiro de Automação Inteligente (SBAI) 2013;p. 85–91. Citado na página 17.

Cheng CC, Tsai CH. Visual servo control for balancing a ball-plate system. *International Journal of Mechanical Engineering and Robotics Research* 2016;5(1):28–32. Citado 2 vezes nas páginas 14 e 18.

De Carvalho GG, Tiosso F, Reis HM. Indústria 4.0. *Revista Interface Tecnológica* 2020;17(2):256–268. Citado na página 13.

Digital S. Educação 4.0 – Tudo o que você precisa saber 2021;<<https://sae.digital/educacao-4-0/>>, *Online*, acessado em 05 de agosto de 2021. Citado 2 vezes nas páginas 16 e 17.

Duarte AYS. Proposta de integração entre ferramentas de avaliação de ciclo de vida do produto e Indústria 4.0 (Industrie 4.0): estudo de caso da indústria têxtil e de confecção 2017;0:120. <<http://repositorio.unicamp.br/handle/REPOSIP/322461>>. Citado na página 13.

El Hamdi S, Abouabdellah A, Oudani M. Industry 4.0: Fundamentals and Main Challenges. *International Colloquium on Logistics and Supply Chain Management, LOGISTIQUA 2019* 2019;p. 1–5. Citado 2 vezes nas páginas 15 e 16.

- Esquembre F. Facilitating the Creation of Virtual and Remote Laboratories for Science and Engineering Education. IFAC-PapersOnLine 2015;48(29):49–58. <<http://dx.doi.org/10.1016/j.ifacol.2015.11.212>>. Citado na página 14.
- Evangelista LV, Roberto M, Guerra B. Montagem e controle de um sistema ball plate . 2019;. Citado 3 vezes nas páginas 22, 23 e 28.
- Henz RM. Desacopladores e projeto no espaço de estados: Extensão de Benchmark em planta industrial de quatro tanques de estados 2017;. Citado na página 20.
- Ismail. DESIGN AND IMPLEMENTATION OF BALL AND PLATE CONTROL SYSTEM USING PID CONTROLLER 2015;(September):1–88. Citado na página 17.
- J Hauser SS, Kokotovic PV. Nonlinear control via approximate input-output linearization: The ball and beam example. IEEE Trans Automatic Control 1992;37(3):392–398. Citado 2 vezes nas páginas 14 e 18.
- James F. Everything You Need to Know About Education 4.0 2021;<<https://www.qs.com/everything-you-need-to-know-education-40/>>, *Online*, acessado em 05 de agosto de 2021. Citado na página 16.
- Kerschbaumer R. Microcontroladores 2018;. Citado 2 vezes nas páginas 37 e 38.
- Kits S. Servo Motor S3003 Futaba 2021;<<https://www.smartkits.com.br/servo-motor-s3003-futaba>>, *Online*, acessado em 09 de setembro de 2021. Citado na página 35.
- MAK MONNMA Bahrin. Industry 4.0: A Review on Industrial Automation and Robotic. Jurnal Teknologi (Sciences Engineering) 2016;p. 137–143. Citado na página 15.
- Mario Cesar M M Campos HCGT. Controles típicos de equipamentos e processos industriais 2006;1:21–33. Citado na página 20.
- NI. Explicando a Teoria PID 2020;<<https://www.ni.com/pt-br/innovations/white-papers/06/pid-theory-explained.html>>, *Online*, acessado em 03 de outubro de 2021. Citado 2 vezes nas páginas 21 e 47.
- Nobuo Oki SCAM. Microcontroladores - PIC 2013;<https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/microcontroladores-_pic-1.pdf>, *Online*, acessado em 22 de agosto de 2021. Citado na página 37.
- Ody LL. PROJETO E DESENVOLVIMENTO DE UMA PLATAFORMA BALL AND PLATE PARA O ESTUDO DE SISTEMAS DE CONTROLE 2020;. Citado 3 vezes nas páginas 18, 22 e 43.
- Ogata K. Engenharia de controle moderno 2011;5:520–542. Citado 2 vezes nas páginas 20 e 43.
- OpenCV. Introduction to OpenCV.js and Tutorials 2021;<https://docs.opencv.org/master/df/d0a/tutorial_js_intro.html>, *Online*, acessado em 30 de setembro de 2021. Citado na página 29.
- OpenCV. About OpenCv 2022;<<https://opencv.org/about/>>. Citado na página 31.
- Pereira F. Microcontroladores PIC 2008;<<https://www.espressif.com/en/products/socs/esp32>>, *Online*, acessado em 21 de agosto de 2021. Citado na página 37.

- Pi R. Raspberry Pi 3 Model B+ 2022;<<https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>>, *Online*, acessado em 29 de Agosto de 2022. Citado na página 29.
- Pi R. Raspberry Pi Pico 2022;<<https://www.raspberrypi.com/products/raspberry-pi-pico/>>, *Online*, acessado em 29 de Agosto de 2022. Citado na página 38.
- Robayo Betancourt FI, Brand Alarcon SM, Aristizabal Velasquez LF. Fuzzy and PID controllers applied to ball and plate system. In: 2019 IEEE 4th Colombian Conference on Automatic Control (CCAC); 2019. p. 1–6. Citado na página 23.
- Santos P. Tecnologia você está preparado para viver a revolução da indústria 4.0? 2015;. Citado na página 15.
- Rafael da Silveira Castro JVFATS Jacson Miguel Olszanecki Barth. MODELAGEM E IMPLEMENTAÇÃO DE UM SISTEMA BALL AND PLATE CONTROLADO POR SERVO-VISÃO 2012;. Citado na página 23.
- Sobrinho FRGAB. Projeto PROIC-UnB: Laboratório com base na Indústria 4.0 para o ensino em cursos de engenharia 2019;. Citado 8 vezes nas páginas 27, 28, 29, 61, 62, 63, 64 e 65.
- Straub MG. FUNCIONAMENTO DO SERVO MOTOR FUTABA S3003 COM ARDUINO 2016;<<https://www.usinainfo.com.br/blog/servo-motor-posicao-180o-4-2kgcm-tipo-futaba-s3003/>>. Citado 2 vezes nas páginas 34 e 35.
- Swaroop. What is RS232 Protocol and How it Works? 2020;<<https://www.codrey.com/embedded-systems/rs232-serial-communication/>>, *Online*, acessado em 18 de Novembro de 2022. Citado na página 33.
- Tropic T. Ball on Plate 2021;<<https://github.com/ttropic/Ball-on-plate>>, *Online*, acessado em 05 de agosto de 2022. Citado na página 31.
- Wang H, Tian Y, Sui Z, Zhang X, Ding C. Tracking control of ball and plate system with a double feedback loop structure. Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, ICMA 2007 2007;p. 1114–1119. Citado na página 17.

Anexos

ANEXO A – Desenho mecânico

O anexo A contém os desenhos mecânicos para o plano inclinado.

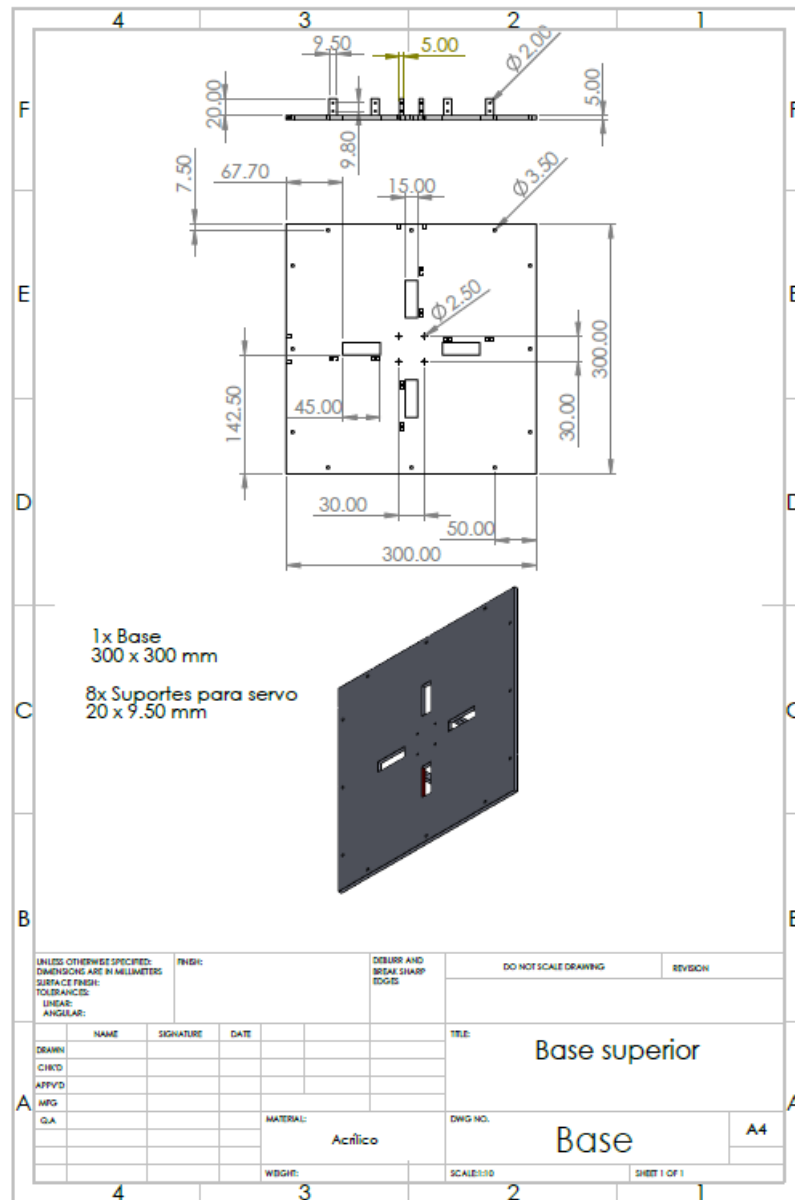


Figura 40 – Base superior

Fonte: Modificado de [Sobrinho \(2019\)](#)

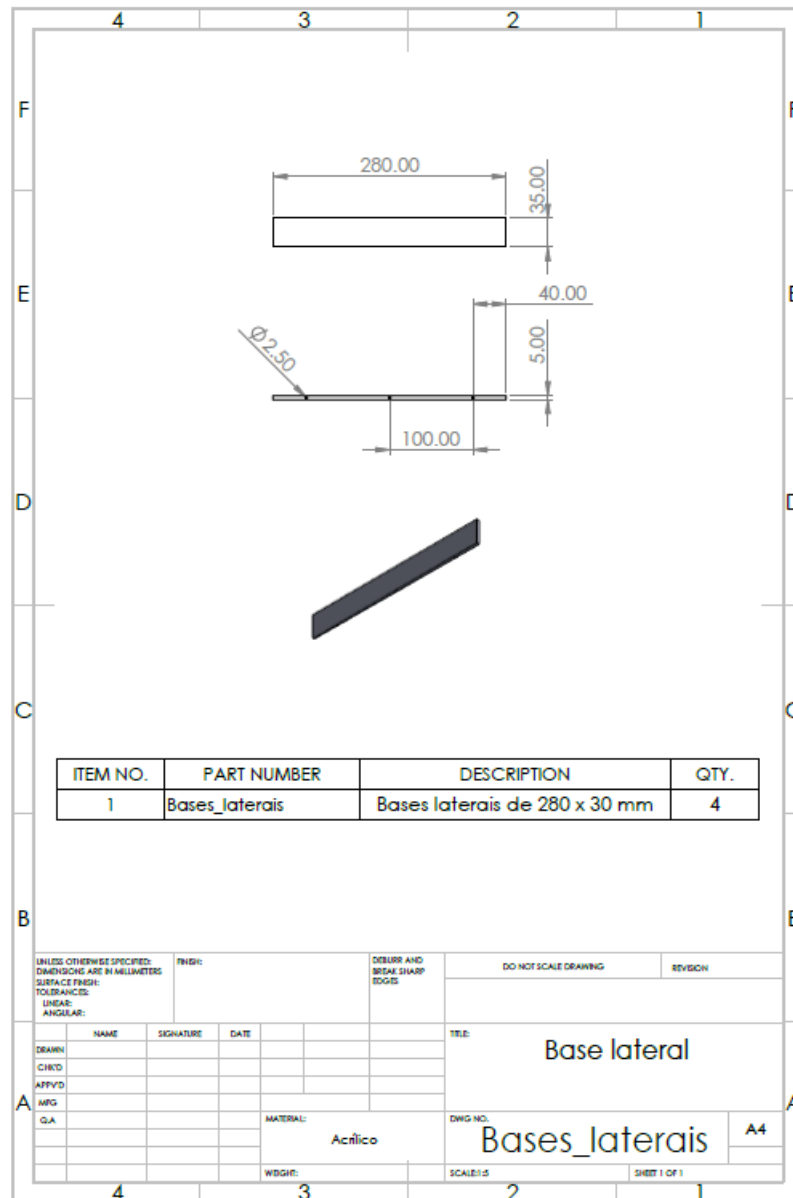


Figura 41 – Base lateral
Modificado de [Sobrinho \(2019\)](#)

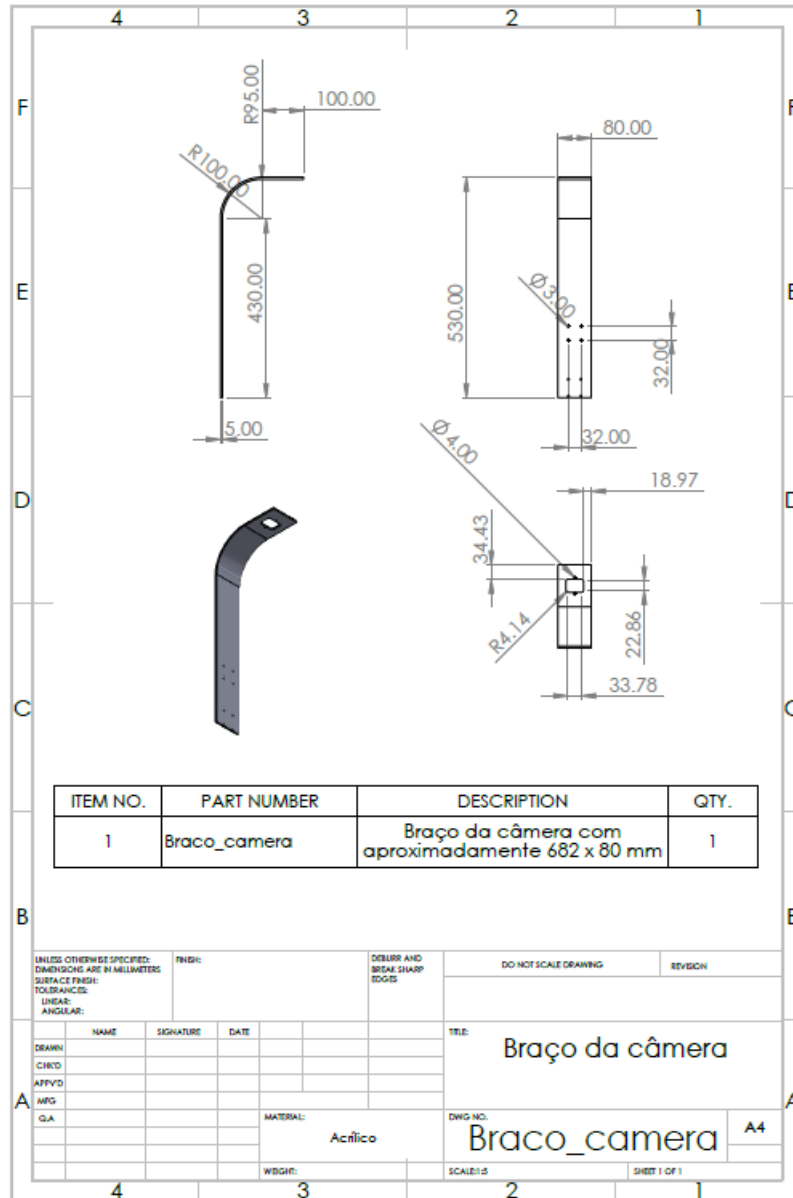


Figura 42 – Braço câmera
 Modificado de Sobrinho (2019)

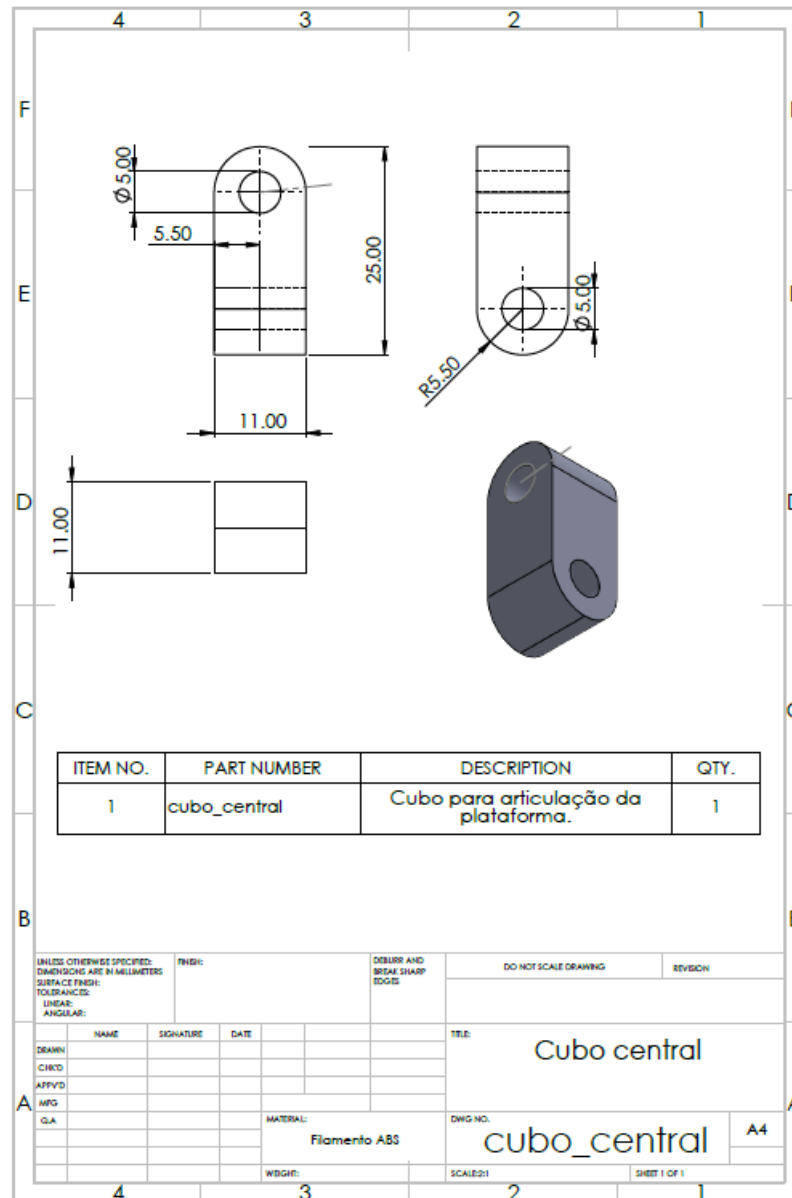


Figura 43 – Cubo central
 Modificado de [Sobrinho \(2019\)](#)

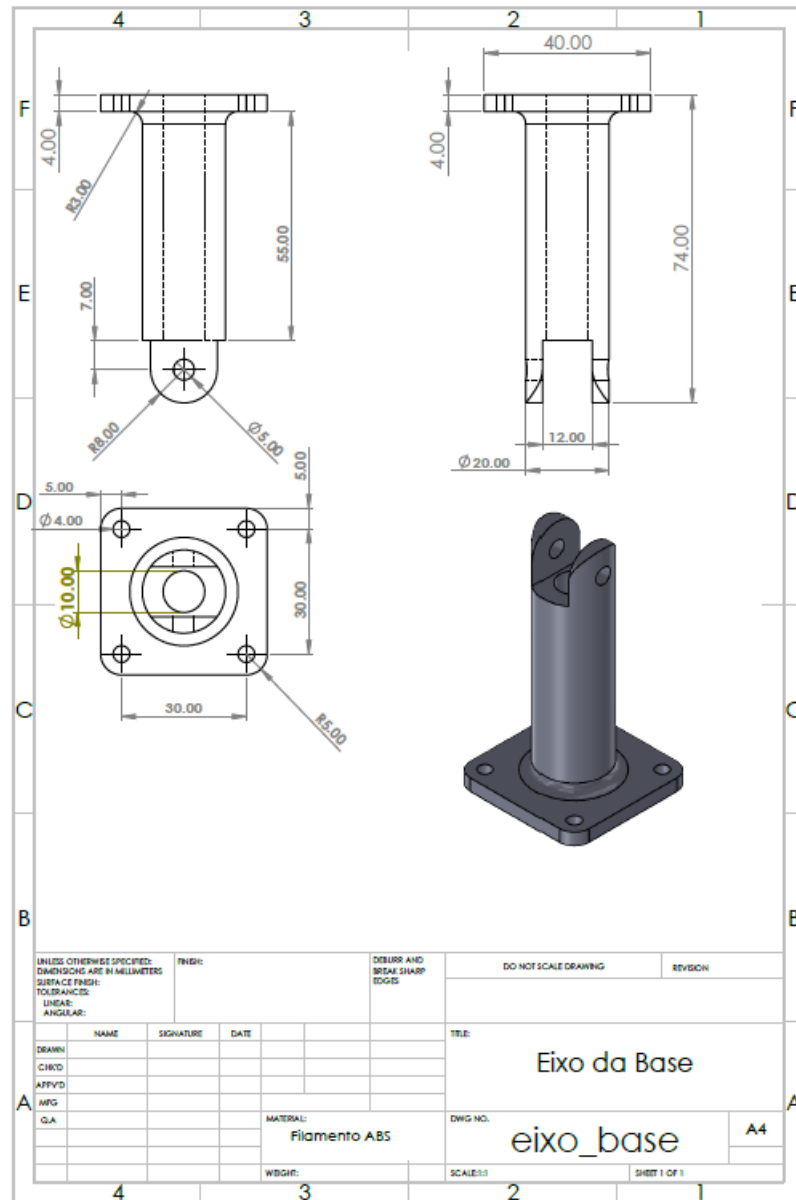


Figura 44 – Eixo base

Modificado de [Sobrinho \(2019\)](#)

ANEXO B – Estrutura final em acrílico

A figura abaixo apresenta a estrutura final do plano inclinado em acrílico .



Figura 45 – Estrutura em acrílico

ANEXO C – Preparação da IDE para programação da Placa

A preparação da IDE para a programação da *Raspberry Pico*, começa com a abertura do gerenciador de placas, como pode ser visto na Figura 46, realizando os seguintes procedimentos Ferramentas > Placa > Gerenciador de Placas.

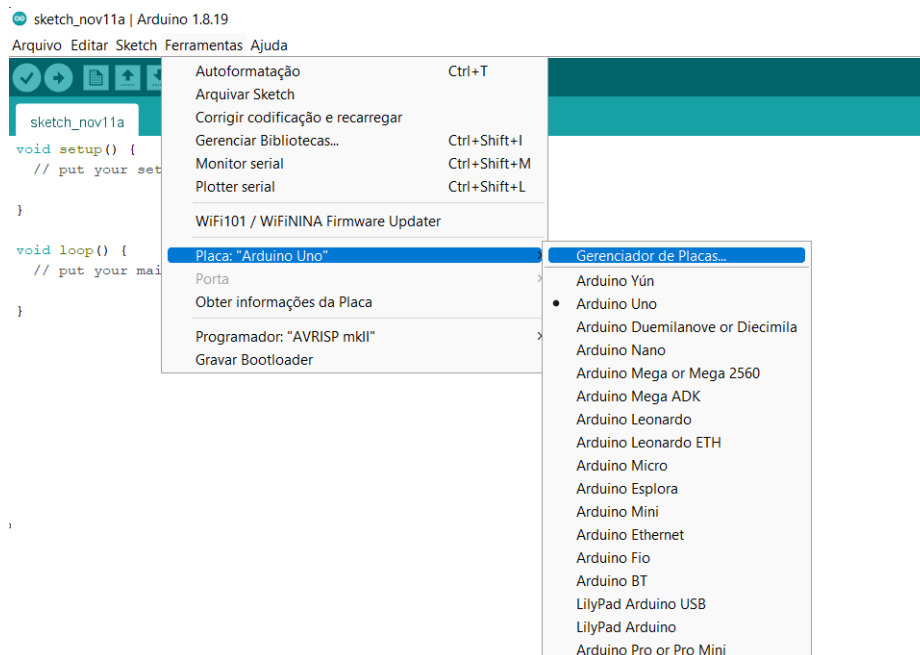


Figura 46 – Gerenciador de placas na IDE do Arduino

Fonte: Elaboração Própria.

Após o gerenciador de placas ser aberto buscar por Pico nas placas existentes, e instalar a *Arduino Mbed OS RP2040 Boards*, como pode ser visto na Figura 47.

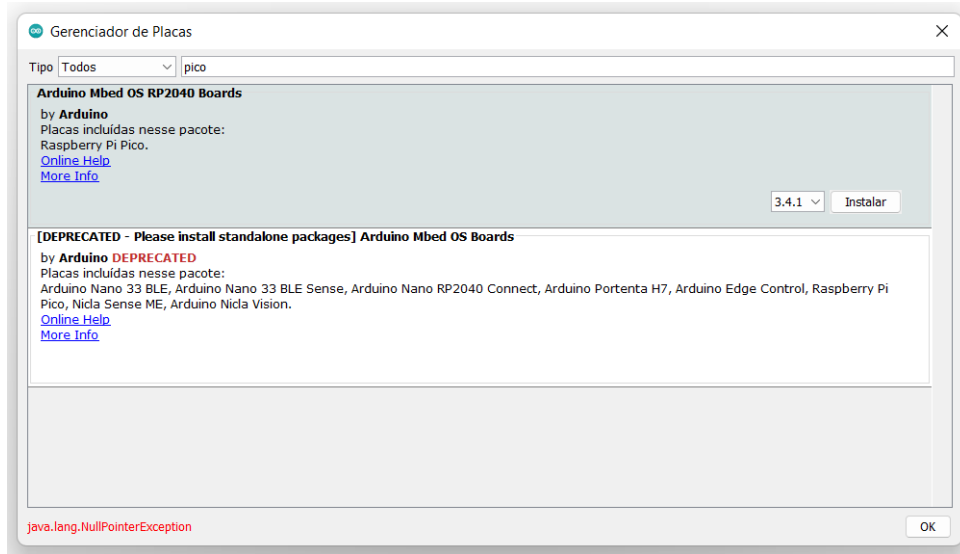


Figura 47 – Instalação da placa da Raspberry Pi Pico na IDE do Arduino.

Fonte: Elaboração Própria.

Para começar a programar a Pico basta pesquisar pela placa em ferramentas e selecionar a placa como pode ser visto na Figura 48.

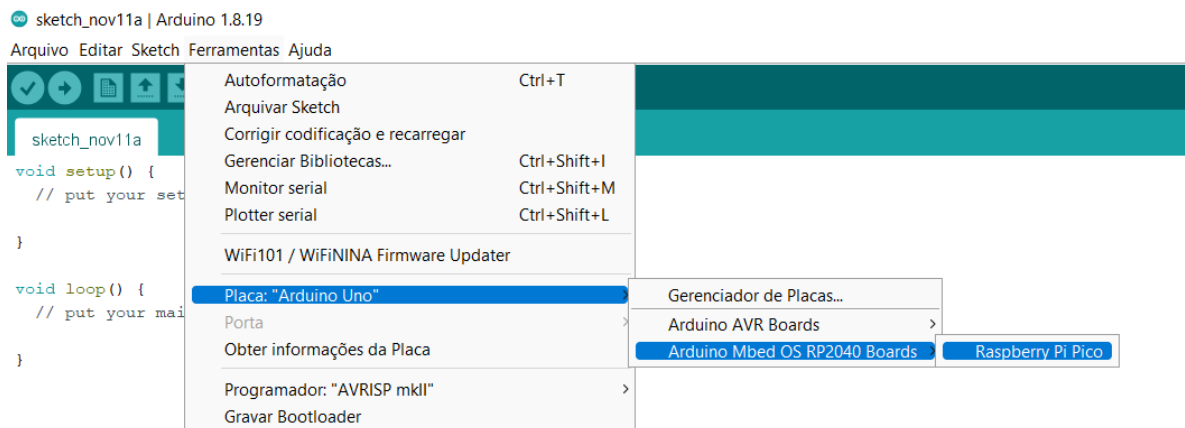


Figura 48 – Seleção da placa da Raspberry Pi Pico na IDE do Arduino.

Fonte: Elaboração Própria.