

## ROBÔ EQUILIBRISTA APLICADO AO ENSINO DE TÉCNICAS PRÁTICAS DE CONTROLE EM ENGENHARIA

**Franco Mateus Gomes do Nascimento** – franco.nascimentodm@hotmail.com  
Instituto Federal de Educação, Ciência e Tecnologia do Maranhão - IFMA  
Endereço: Av. Getúlio Vargas - Monte Castelo, São Luís - MA  
65030-005 – São Luis – Maranhão

**Francisco Borges Carreiro** – fborges@ifma.edu.br  
Instituto Federal de Educação, Ciência e Tecnologia do Maranhão - IFMA  
Endereço: Av. Getúlio Vargas - Monte Castelo, São Luís - MA  
65030-005 – São Luis – Maranhão

**Resumo:** No presente artigo, um Robô Equilibrista é desenvolvido lançando mão de técnicas de controle clássicas, tal como o controle PID, com o objetivo final de ser um aporte ao ensino prático de técnicas de controle no âmbito da engenharia. O protótipo consiste no emprego de um modelo de pêndulo invertido, comumente descrito na literatura. Com o objetivo de desenvolvermos um software e hardware consideravelmente simples, de modo a possibilitar a sua implementação por alunos em formação, utilizamos ferramentas de fácil manuseio e baixo custo, tais como o microcontrolador NODEMCU ESP32, que além do baixo custo pode ser facilmente programado na IDE nativa do Arduino. Além do NODEMCU ESP32, o protótipo conta com sensor acelerômetro, giroscópio e magnetômetro MPU 6050, e motores DC com caixa de redução, interfaceados por um módulo Ponte-H, responsáveis pelo posicionamento angular e controle do Robô Equilibrista respectivamente.

**Palavras-chave:** Robô Equilibrista. PID. Ensino prático. Controle.

### 1 INTRODUÇÃO

Os sistemas computacionais passaram por grande evolução nas duas últimas décadas de modo que hoje é possível executar algoritmos com funções extremamente complexas em circuitos integrados cada vez menores e com uma programação cada vez mais simples. No entanto, mesmo com o rico aprimoramento das tecnologias computacionais, as técnicas de ensino de engenharia não acompanharam lado a lado esse desenvolvimento, de modo que ainda é possível constatar um ensino da teoria muito distante da prática, e nesse deficit encontramos um amplo espaço para desenvolver novas formas de aliar o ensino teórico da engenharia com a prática, de modo a desenvolver as competências tanto do aluno quanto do professor, que também precisa se reinventar num ambiente de constate desenvolvimento.

A implementação do robô equilibrista representa uma estratégia abrangente para o ensino de engenharia, tendo em vista que além de aplicar técnicas de controle é possível realizar uma

conexão multidisciplinar abrangendo tópicos voltados à eletrônica, lógica de programação, arquitetura de sistemas digitais, filtragem e processamento de sinais.

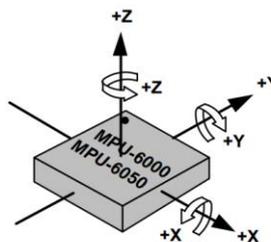
Utilizamos o modelo de um pêndulo invertido que necessita se manter em equilíbrio na posição vertical, para a montagem do protótipo do robô equilibrista. De acordo com Franklin (2013) “nos últimos anos, esse sistema existiu principalmente em laboratórios de sistemas de controle, como uma ferramenta educacional” (Franklin, 2013). No entanto, o protótipo proposto neste artigo tem como objetivo o desenvolvimento por qualquer aluno em formação, de maneira mais simples e instrutiva.

Tendo como objetivo principal o desenvolvimento de uma ferramenta simples de aprendizagem para o aluno, escolhemos o processador NODEMCU ESP32 que pode ser programado no ambiente de programação do Arduino. O ambiente de programação Arduino é aberto e gratuito, conta com muitas bibliotecas e exemplos de implementações disponíveis de forma gratuita assim como é familiar aos professores e alunos no âmbito da engenharia. Outros componentes como acelerômetros/giroscópio, motores de corrente contínua, que são encontrados com facilidade no mercado e foram utilizados.

## 2 PRINCIPIO DE FUNCIONAMENTO

O robô equilibrista usa o mesmo princípio do pêndulo invertido que tem modelo descrito por Franklin (2013). A planta utiliza-se de um sensor acelerômetro e giroscópio MPU6050 para definir a posição angular do robô no eixo vertical. O sensor é capaz de captar a aceleração linear e a velocidade angular em três eixos perpendiculares, conforme ilustra a figura 1.

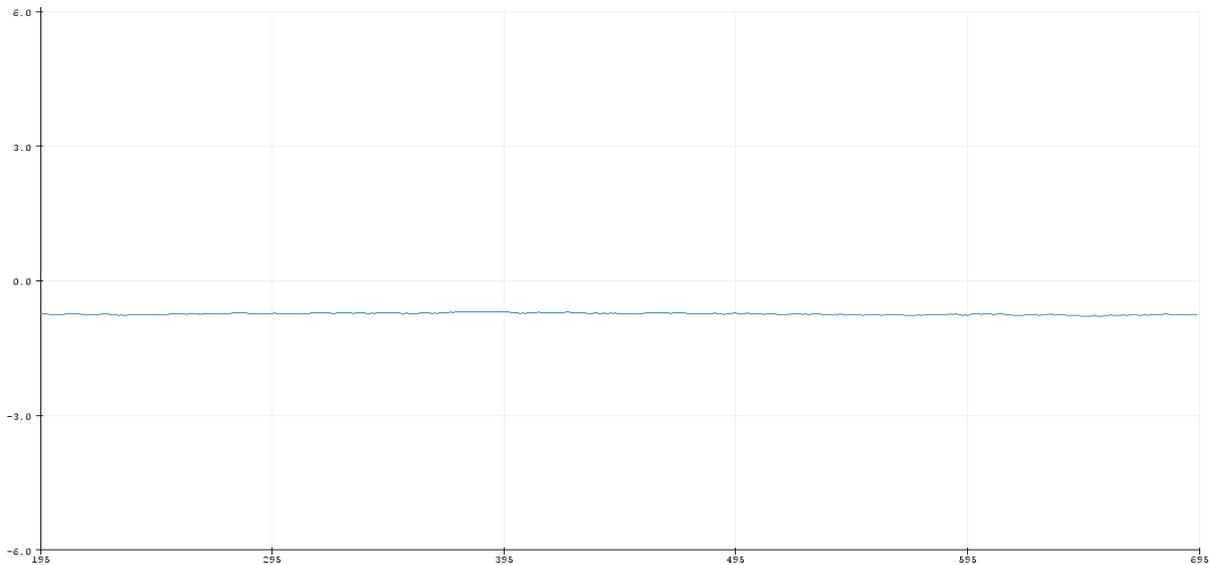
Figura 1: Orientação dos eixos de medição



Fonte: INVENSENSE (2013)

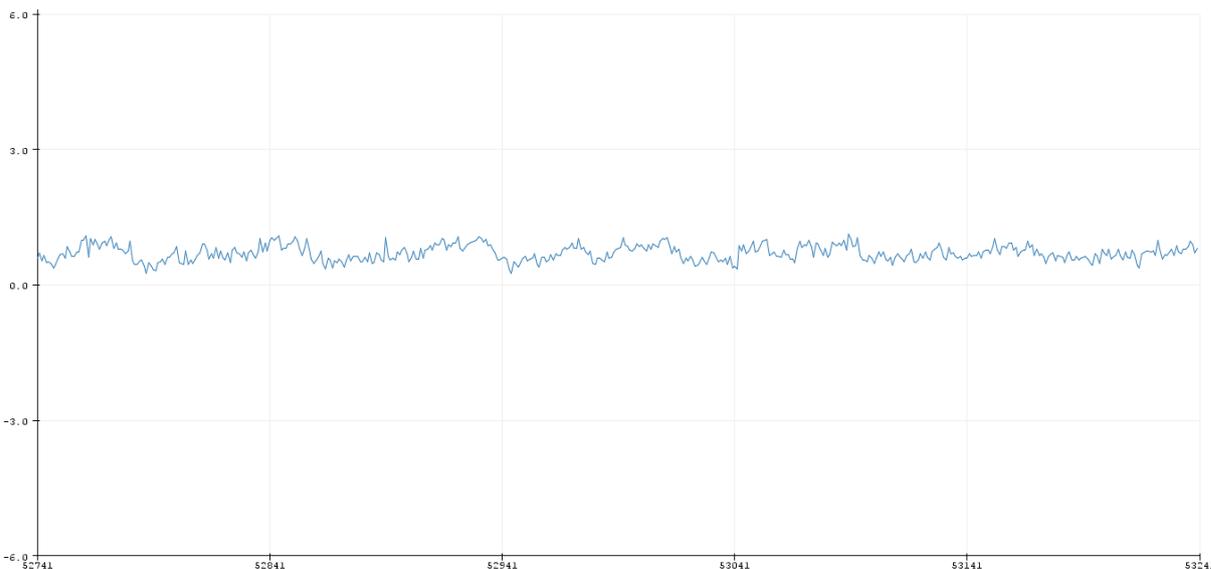
Apesar da precisão da MPU6050 as medições podem ser contaminadas com ruídos conforme demonstra Oliveira e Gonçalves (2017), para atenuar estes ruídos que podem prejudicar toda a dinâmica do sistema, optamos pela utilização do Filtro de Kalman, implementado em 1958 por Rudolf Kalman. A escolha deste método de filtragem se justifica pela sua ampla utilização nesse tipo de problema além de haver versão deste filtro implementado na forma como bibliotecas para a Arduino, tal como a biblioteca “Kalman.h” de autoria de Kristian Lauszus (2012). Através das figuras 2 e 3 é possível fazer um comparativo da condição dos dados recebidos da MPU6050, com e sem a filtragem.

Figura 2: Posição angular com filtragem



Fonte: Autores

Figura 3: Posição angular sem filtragem



Fonte: Autores

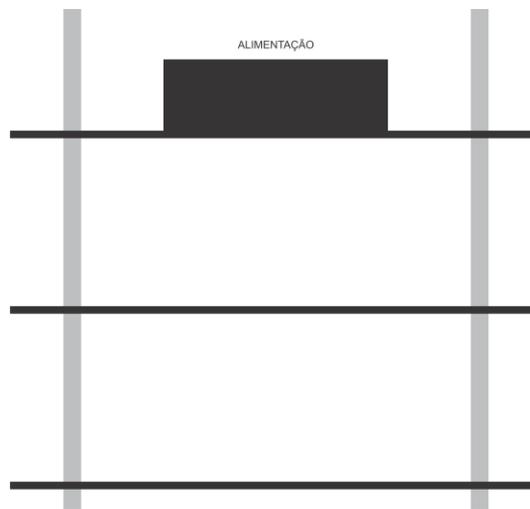
Através dos dados coletados e filtrados da MPU6050 é possível localizar o robô em torno do eixo vertical. De posse destes dados podemos implementar um algoritmo de controle PID que ficará hospedado no ESP32 e que será responsável em corrigir a posição angular do robô sempre de volta ao ponto de referência à medida que o mesmo tende a cair por ação gravitacional. Ou seja, se o robô tende a cair para o sentido positivo da orientação, os motores são acionados no sentido de rotação contrário gerando uma reação que tenderá a equilibrar o mesmo.

### 3 MONTAGEM ESTRUTURAL

Toda a estrutura do robô foi construída de maneira simples utilizando três chapas de material plástico rígido como patamares para a alocação dos componentes do *hardware* como

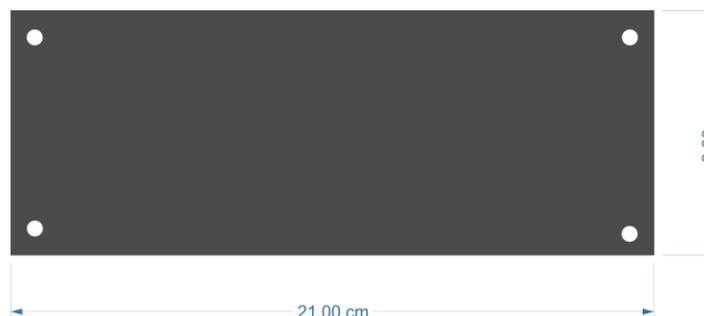
ilustram as figuras 4 e 5. As chapas foram conectadas através de barra roscada, utilizando porcas para a fixação das mesmas.

Figura 4: Vista frontal do robô



Fonte: Autores

Figura 5: Vista superior da chapa patamar



Fonte: Autores

#### 4 HARDWARE

O hardware é composto completamente por materiais de fácil acesso e também de baixo custo. A tabela 1 demonstra os materiais utilizados e seus respectivos valores aproximados.

Tabela 1: Descrição de itens do hardware

QNTD.	ITEM	VALOR
01	NODEMCU ESP32	R\$ 40,00
02	BATERIAS DC, 4.2V, 8800mAh	R\$ 20,00
01	Ponte H L298N	R\$ 15,00
02	Motores DC com caixa de redução e rodas	R\$ 30,00
01	Conversor de Nível Logico 3.3V/5V	R\$ 7,50
01	Circuito regulador de tensão LM7805	R\$ 12,00
01	Switch	R\$ 0,60
01	MPU6050	R\$ 14,00

Fonte: Autores

As duas baterias descritas na tabela 1, são suficientes para alimentar todos os elementos do robô. No entanto, algumas conversões de níveis de tensão são necessárias ao longo do fluxo de informações no sistema. O Primeiro ocorre no LM7805 que reduz a tensão nominal das duas baterias em série de 8.4V para 5V que é a tensão de alimentação do ESP32. A segunda conversão ocorre no fluxo de informações entre o ESP32 e a Ponte H, visto que as saídas digitais do microcontrolador tem níveis de tensão de 3.3V, e a Ponte H exige como sinal lógico um nível de 5V.

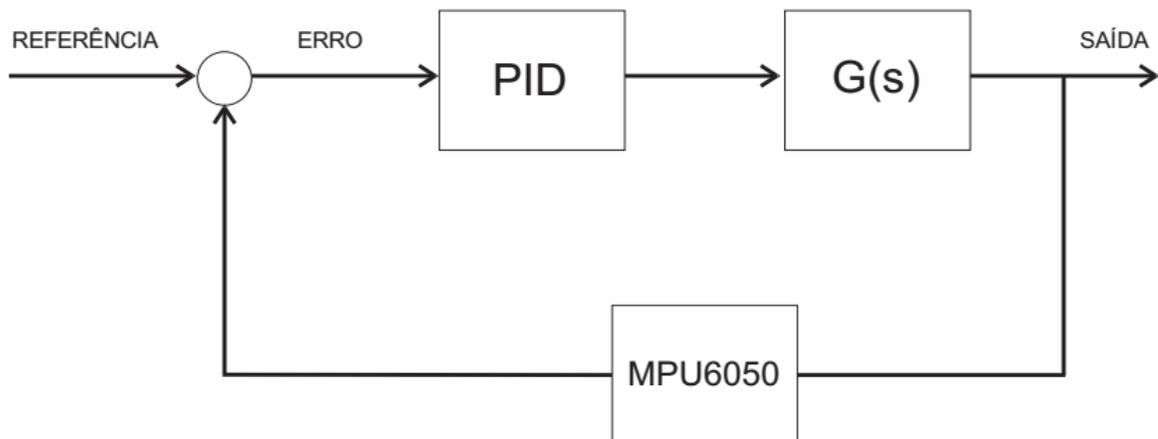
## 5 CONTROLE DO SISTEMA

O algoritmo foi implementado na IDE nativa do Arduíno, valendo-se da lógica descrita anteriormente no tópico 2. Para realizar o controle do robô em torno da posição de interesse é necessário lançar mão de uma técnica de controle que se adéque a característica não linear do sistema, conforme descreve Franklin (2013). Tendo em vista essa condição e o objetivo geral do trabalho, que é o de ser uma ferramenta dinâmica para o ensino de técnicas de controle na engenharia, escolhemos como técnica o controlador PID, pois o mesmo representa uma técnica poderosa e amplamente utilizada, tanto na prática como para o ensino teórico, visto que a grande maioria dos cursos na área de controle perpassam pelo controlador PID. Este controlador se vale de três termos, o Proporcional, Integral e Derivativo, cada um destes parâmetros está diretamente ligado as variáveis de comportamento do sistema, tais como, velocidade de resposta, erro, tempo de acomodação, sobressinal, etc. A Equação (1) define o controlador PID.

$$C = K_p e(t) + K_i \int e(t) dt + K_d \frac{d(e(t))}{dt} \quad (1)$$

O PID representa um controlador suficientemente eficiente na função de manter o robô equilibrado. Em específico para a topologia de patamares escolhida para o mesmo neste trabalho, apenas os termos Proporcional e Integral são suficientes para executar o controle principal e manter o sistema estável. No entanto, para efeitos de construção do código de controle todos os três termos do controlador PID são considerados, tendo em vista que, caso necessário o professor e/ou o aluno a frente do desenvolvimento do protótipo em ambiente acadêmico podem utilizar livremente o termo Derivativo em conjunto com os demais, a fim de garantir uma abordagem mais completa em torno deste controlador. A figura 6 ilustra o diagrama de blocos para o sistema implementado.

Figura 6: Diagrama de blocos do sistema



Fonte: Autores

Onde  $G(s)$  representa a função de transferência que governa o comportamento do robô.

É importante ressaltar que a implementação do algoritmo de controle embarcado em um microcontrolador, garante a possibilidade para que o operador possa realizar mudanças de forma fácil e rápida, isso também possibilita que a abordagem de controle possa ser definida pelo usuário. No presente trabalho utilizamos como técnica para definição do termo integral a Integração de Euler ou Acumulação, que segundo Ellis (2004) “é o método de integração digital mais comum”. A Equação 2 define a Integração de Euler.

$$C_N = C_{N-1} + TR_N \quad (2)$$

Onde  $C_N$  representa a integral acumulada,  $R_N$  o erro e  $T$  o tempo de cada amostra.

Porém o tipo de integração pode ser escolhido livremente, podendo ser um método mais preciso, como a Integração Trapezoidal.

Devemos tomar nota de que o método de filtragem descrito anteriormente no tópico 2, se faz extremamente necessário para que o controlador possa desempenhar a sua função de maneira satisfatória, tendo em vista que todos os três termos do controlador PID são funções que dependem diretamente do erro, que no caso do robô equilibrista representa a diferença entre a leitura da posição angular instantânea feita pela MPU6050 e a posição angular de referência, ou seja, o ponto de equilíbrio.

Após realizar toda a implementação do algoritmo de controle, se faz necessário definir as constantes de proporcionalidade de cada termo do controlador, ou seja, os ganhos  $K_P$ ,  $K_I$  e  $K_D$ . Este processo é comumente chamado de sintonia do controlador, e é possível encontrar vários métodos que norteiam essa sintonia, tais como Lugar das Raízes, Nise (2012) e Ziegler–Nichols, Ogata (2010). No entanto, como este trabalho visa construir uma ferramenta de ensino simplificada e eficiente, e a grande maioria dos métodos de sintonia citados acima tem uma complexidade considerável, já que a para a grande maioria é necessário estar de posse do modelo matemático do sistema, e tendo em vista que a obtenção deste tipo de modelagem física e matemática geralmente não é tarefa simples, optamos por utilizar como técnica de sintonia a chamada Sintonia por Zonas descrita por Ellis (2004). A sintonia por zonas pode ser realizada sem a modelagem matemática do sistema, ou seja, podemos utilizar como referência para a sintonia apenas o comportamento físico da planta a medida que os ganhos de controle são ajustados. Essa técnica nos dá também a possibilidade de sintonizar os parâmetros de forma independente, diminuindo assim a combinação de possíveis pontos de estabilidade.

Para o nosso sistema primeiramente é sintonizado, de modo que o operador dever aumentar gradativamente o valor deste ganho, e após cada incremento verificar o comportamento do robô, ou seja, a reação que o mesmo faz para manter-se na posição de equilíbrio. Como o tempo proporcional do controlador está diretamente ligado a velocidade de resposta um ganho  $K_P$  muito alto fará com que o robô responda a mudanças na posição angular de maneira muito brusca levando-o para a instabilidade, em contrapartida um ganho  $K_P$  muito baixo causará um movimento de reação muito lento e incapaz de trazer o robô a posição de equilíbrio. Portanto o ganho  $K_P$  deve ser ajustado de modo que o robô consiga reagir de forma rápida o suficiente para não cair, mas não tão rápida a ponto de criar instabilidade.

Terminado a sintonia da primeira zona, é necessário realizar a sintonia para a segunda zona, ou seja, para o ganho  $K_I$ . Este ajuste é feito de maneira análoga à primeira zona, porém é importante ter em mente que o  $K_I$  por sua vez está diretamente ligado ao erro de regime permanente. Em outras palavras o ajuste desta zona garantirá que o esforço de controle do robô na região próxima a referência seja satisfatório.

Com algumas tentativas já é possível perceber o robô tendendo a equilibrar-se. Para este trabalho foram necessárias vinte e seis iterações para que ambas as zonas fossem ajustadas de maneira satisfatória, e o robô permanecesse em equilíbrio.

Após todas as etapas descritas acima terem sido concluídas, é possível pôr o robô em funcionamento. A figura 7 ilustra o robô montado e em funcionamento.

Figura 7: Robô executando controle de equilíbrio



Fonte: Autores

## 6 CONCLUSÕES

É inegável que nos últimos anos as ferramentas tecnológicas voltadas ao ensino de engenharias desenvolveram-se fortemente, de modo que é necessário que tais técnicas sejam revistas e aprimoradas constantemente, com o intuito de sempre melhorar aprendizagem e a formação dos profissionais que saem da academia. Com o objetivo de contribuir com esse ambiente de inovações e avanços, este trabalho abordou a implementação de um Robô Equilibrista que visa ser uma ferramenta simples para o ensino de técnicas de projeto e sintonia de controladores em plantas reais.

Por meio da implementação e utilização desse protótipo, alunos poderão testar e desenvolver suas competências não só na área de controle, mas também nas diversas áreas afins que este trabalho enfoca. Outros pontos importantes a serem elencados são a simplicidade física do projeto, que facilita a familiaridade do aluno com a ferramenta, por exemplo no emprego da linguagem de programação nativa do Arduino que é amplamente difundida e que contém uma sintaxe simples e intuitiva. Além do baixo custo total do projeto levando em consideração que os recursos para inovação no ensino, nem sempre são de fácil acesso em todas as instituições.

Dito isso é possível constatar que a ferramenta apresentada neste artigo pode se mostrar promissora no ensino de engenharias na academia, de modo a tirar os alunos de um ambiente puramente teórico e transportá-los a uma realidade onde teoria e prática estão indissociáveis, além de criar o interesse e satisfação no aluno que pode desempenhar papel ativo no seu processo de aprendizagem, que na maioria das vezes é centralizado na figura do professor.

### ***Agradecimentos***

Aos Professores Dr. Ginalber Serra e Dr. Reginaldo Miranda pela assistência durante a execução do projeto. Ao IFMA por auxílio na forma de bolsa de iniciação científica e instalações de laboratório.

### **REFERÊNCIAS**

Arduino Software (IDE). Disponível em: <https://www.arduino.cc/en/Main/Software>. Acesso em: 25 ago. 2018.

ELLIS, George. **Control System Design Guide**. 4ª Ed. California: Elsevier Academic Press. 2004.

FRANKLIN, Gene F; POWELL, J. David; EMAMI-NAEINI, Abbas. **Sistemas de Controle para Engenharia**. 6ª Ed. Porto Alegre: Bookman, 2013.

HELLMAN, Hanna; SUNNERMAN, Henrik; **Two-Wheeled Self-Balancing Robot - Design and control based on the concept of an inverted pendulum**. Tese (Bacharelado) – KTH Royal Institute of Technology. Estocolmo, Suecia, 2015.

LAUSZUS, Kristian. **Kalman Filter**. Disponível em: <https://github.com/TKJElectronics/KalmanFilter>. Acesso em: 08 dez. 2018.

NISE, Norman S. **Control Systems Engineering**. 7ª Ed. California: John Wiley & Sons, 2015.

OGATA, Katsuhiko. **Modern Control Engineering**. 5ª Ed. New Jersey: Pearson Education: 2010.

OLIVEIRA, Waldri dos Santos; GONÇALVES, Eduardo. Nunes. **Implementação em c: filtro de Kalman, fusão de sensores para determinação de ângulos**. ForScience: revista científica do IFMG, Formiga, v. 5, n. 3, e00287, jul./dez. 2017

## SELF-BALANCING ROBOT APPLIED IN THE TEACHING OF PRACTICAL CONTROL TECHNIQUES IN ENGINEERING

**Abstract:** *In this article, a Self-Balancing Robot is developed using classic control techniques, such as PID control, with the ultimate goal of being a contribution to the practical teaching of control techniques in engineering. The prototype consists of the use of an inverted pendulum model, commonly described in the literature. With the objective of developing a very simple software and hardware to enable this implementation by students in training, we use tools of easy handling and low cost, such as the NODEMCU ESP32 microcontroller, which in addition to the low cost can be easily programmed in the Arduino IDE. In addition to the NODEMCU ESP32, the prototype features an accelerometer, gyroscope and magnetometer MPU 6050, and DC motors with gear reduction, interfaced by an H-Bridge module, responsible for the angular positioning and control of the Self-Balancing Robot respectively.*

**Key-words:** *Self-Balancing Robot. PID. Practical teaching. Control.*