**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA – CÂMPUS FLORIANÓPOLIS**

**DEPARTAMENTO ACADÊMICO DE ELETRÔNICA**

**CURSO DE GRADUAÇÃO EM ENGENHARIA ELETRÔNICA**

**NYCOLAS COELHO DE ABREU**

**TÍTULO DO TRABALHO: e subtítulo se houver**

**FLORIANÓPOLIS, 2022.**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA – CÂMPUS FLORIANÓPOLIS**

**DEPARTAMENTO ACADÊMICO DE ELETRÔNICA**

**CURSO DE GRADUAÇÃO EM ENGENHARIA ELETRÔNICA**

**NYCOLAS COELHO DE ABREU**

**TÍTULO DO TRABALHO: e subtítulo se houver**

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Engenheiro em Eletrônica.

Orientador:   
Prof. Renan Augusto Starke, Dr.

**FLORIANÓPOLIS, 2022.**

**PÁGINA PARA COLOCAÇÃO**

**DA FICHA DE IDENTIFICAÇÃO DA OBRA**

(A Ficha de identificação da obra deve ser elaborada de acordo com o padrão adotado pela biblioteca do IFSC a partir do formulário disponível em: <http://ficha.florianopolis.ifsc.edu.br/>

Observação: por questões de compatibilidade do *site*, recomenda-se que a Ficha de identificação seja gerada no navegador ***Mozilla Firefox***)

**TÍTULO DO TRABALHO**

**NYCOLAS COELHO DE ABREU**

Este trabalho foi julgado adequado para obtenção do título de Engenheiro em Eletrônica e aprovado na sua forma final pela banca examinadora do Curso de Engenharia Eletrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, \_\_\_ de \_\_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_.

Banca Examinadora:

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Nome do Orientador, Titulação

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Nome do Coorientador (se houver), Titulação

Instituição/Empresa

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Nome do Membro da Banca, Titulação

Instituição/Empresa

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Nome do Membro da Banca, Titulação

Instituição/Empresa

**RESUMO**

O resumo deve mostrar a natureza e o objetivo do trabalho, o método que foi empregado, os resultados e as conclusões. O resumo deve conter entre 150 e 500 palavras e constitui-se de um único parágrafo, sem recuo.

**Palavras-chave**: Primeira palavra-chave. Segunda palavra-chave. Terceira palavra-chave. Quarta palavra-chave (opcional). Quinta palavra-chave (opcional).

**ABSTRACT**

The abstract should show the nature and scope of work, the method that was used, the results and conclusions. The abstract may contain between 150 and 500 words, and it must be only one paragraph.

**Keywords**: First keyword. Second keyword. Third keyword. Fourth keyword (optional). Fifth keyword (optional).

**LISTA DE FIGURAS**

[Figura 1 – Código compilado e interpretado](#_heading=h.1ksv4uv) 17

Figura 2 – Exemplo de código escrito em C embarcado 17

[Figura 2 – Diagrama Fasorial 18](#_heading=h.2jxsxqh)

**LISTA DE QUADROS**

[Quadro 1 – Tipos de energia analisados 16](#_heading=h.3rdcrjn)

**LISTA DE TABELAS**

[Tabela 1 – Produção de petróleo na Bahia 17](#_heading=h.44sinio)

**LISTA DE ABREVIATURAS E SIGLAS**

| ANSI | American National Standards Institute |
| --- | --- |
| BIPES | Block based Integrated Platform for Embedded Systems |
| DAE | Departamento Acadêmico de Eletrotécnica |
| IFSC | Instituto Federal de Santa Catarina |
| IoT | Internet of Things |
| STL | Standard Template Library |

**SUMÁRIO**

[**1**](#_heading=h.1fob9te) **INTRODUÇÃO 14**

[**1.1**](#_heading=h.3znysh7) **Justificativa 14**

[**1.2**](#_heading=h.2et92p0) **Definição do Problema 14**

[**1.3**](#_heading=h.tyjcwt) **Objetivo Geral 14**

[**1.4**](#_heading=h.3dy6vkm) **Objetivos Específicos 14**

[**1.5**](#_heading=h.1t3h5sf) **Estrutura do Trabalho 15**

[**2**](#_heading=h.4d34og8) **FUNDAMENTAÇÃO TEÓRICA 16**

[**2.1**](#_heading=h.17dp8vu) **Subtítulo Secundário 1 16**

[**2.2**](#_heading=h.26in1rg) **Subtítulo Secundário 2 16**

[2.2.1](#_heading=h.lnxbz9) Subtítulo Terciário 16

[*2.2.1.1*](#_heading=h.35nkun2) *Subtítulo Quaternário 17*

[**3**](#_heading=h.1y810tw) **METODOLOGIA 19**

[**3.1**](#_heading=h.4i7ojhp) **Métodos aplicados 19**

[**4**](#_heading=h.2xcytpi) **APRESENTAÇÃO DOS RESULTADOS 20**

[**4.1**](#_heading=h.1ci93xb) **Análise e discussão dos resultados 20**

[**5**](#_heading=h.3whwml4) **CONSIDERAÇÕES FINAIS 21**

[**5.1**](#_heading=h.2bn6wsx) **Sugestões para trabalhos futuros 21**

[**REFERÊNCIAS 22**](#_heading=h.qsh70q)

[**APÊNDICES 23**](#_heading=h.3as4poj)

[APÊNDICE A – Título 24](#_heading=h.1pxezwc)

[APÊNDICE B – Título 25](#_heading=h.49x2ik5)

[**ANEXOS 26**](#_heading=h.2p2csry)

[ANEXO A – Título 27](#_heading=h.147n2zr)

[ANEXO B – Título 28](#_heading=h.3o7alnk)

# 

# INTRODUÇÃO

O projeto de sistemas embarcados cresce de forma constante nos últimos anos. As aplicações são onipresentes na fabricação de produtos de consumo, veículos, telecomunicações, sistemas militares e de defesa do meio ambiente (MICHELI; SAMI, 1996).

Esses componentes estão se tornando mais complexos e possuem cada vez menos tempo para serem apresentados ao mercado. Portanto, surgiu a necessidade de criar um ambiente de projeto mais eficiente e de fácil utilização, onde o projetistas e jovens sem muita experiência possam utilizar da programação visual ou *no-code* para auxiliar em seus estudos e protótipos.

Atualmente, a base da programação de microcontroladores são linguagens de baixo nível como C, C++, Assembly e MicroPython, todas linguagens textuais em com sintaxes fortemente tipadas. Como esclarece Caballar (2020), a programação *no-code* seria a progressão natural do desenvolvimento de software, envolvendo a adição de camadas de abstração e escondendo as complexidades por trás do código textual, tornando a programação mais fácil para os desenvolvedores.

Ferramentas simples e intuitivas de programação em blocos já são amplamente utilizadas por desenvolvedores e estudantes, plataformas de programação em blocos como o Scratch, que foi projetado para ensinar programação para jovens, possui mais de 95 milhões de usuários e quase 115 milhões de projetos compartilhados (MIT; 2022) e também a Blockly,uma biblioteca criada pelo Google para facilitar a criação de linguagens visuais em bloco, utilizada em centenas de projetos, principalmente educacionais (GOOGLE; 2022).

Este trabalho tem como proposta o desenvolvimento de uma biblioteca de programação em blocos voltada para microcontroladores, utilizando plataformas de programação projetadas para estes ambientes.

A elaboração da biblioteca foi realizada pela plataforma *open* *source* BIPES, desenvolvida no Brasil e planejada para programação de sistemas embarcados e aplicações IoT (JUNIOR et al., 2020), e também na plataforma MicroBlocks um ambiente de programação projetado para kits de eletrônica, inspirado em Scratch, GP e Snap! (CABRERA; MALONEY; WEINTROP, 2019).

## Justificativa

A programação de microcontroladores é um processo que necessita de desenvolvedores e engenheiros qualificados com amplo conhecimento de uma determinada plataforma e linguagem de programação, além de requerer uma pesquisa minuciosa de diferentes documentos e *datasheets,* para que se consiga obter os conhecimentos necessários para a execução de um projeto. Como resultado, as bases de códigos geralmente carecem de modularidade e apresentam uma estrutura de difícil compreensão.

Para jovens e iniciantes, analisar e criar bases de código para microcontroladores se torna uma tarefa difícil e demorada. Por esses motivos, novas ferramentas e metodologias de programação foram criadas para facilitar e acelerar o processo de aprendizado e prototipagem. Um destes métodos é a programação em blocos, que introduz uma dimensão visual para apoiar o processo de aprendizagem, segundo Alrubaye, Ludi e Mkaouer (2019), este ambiente produz uma nova maneira de se escrever código, incluindo cores e formas, podendo reduzir a curva de aprendizado que os alunos possuem quando iniciam o aprendizado de programação em um ambiente baseado em texto.

A ferramenta de programação em blocos, além de facilitar a codificação para o usuário, prepara o ambiente de programação de forma automatizada, colocando em prática a ideia *plug-and-play*, o usuário não precisa se preocupar com diversas variáveis ​​de preparação, instalação e configuração da plataforma.

Diante disso, esse estudo propõe abordar a programação em blocos voltada para microcontroladores, implementando uma biblioteca de programação em blocos utilizando plataformas existentes e atuais, de forma a evidenciar sua praticidade, viabilidade e funcionamento.

## Definição do Problema

Dependendo da plataforma e da linguagem escolhida, a programação de sistemas embarcados pode ser complexa e demorada, envolvendo a compreensão de um conjunto de teoria, prática, conhecimento sobre paradigmas, sintaxe, semântica da linguagem, lógica, abstração e habilidades de raciocínio algorítmico. Esses aspectos dificultam o aprendizado para iniciantes e são uma barreira para aqueles que desejam criar protótipos rapidamente.

Dessa forma, diante das novas e velhas opções de linguagens e ferramentas de programação para sistemas embarcados, podemos utilizar alguma plataforma para criar protótipos simples e com um aprendizado facilitado?

## Objetivo Geral

O presente trabalho tem como objetivo a implementação e desenvolvimento de uma biblioteca de programação em blocos, visando a facilidade na prototipação e uma alternativa de aprendizado moderna para programação de microcontroladores.

## Objetivos Específicos

Com o objetivo geral apresentado, destaca-se os seguintes objetivos específicos:

1. Estudar as ferramentas e tecnologias disponíveis para o desenvolvimento da biblioteca;
2. Escolha dos componentes para construção do *hardware*;
3. Projetar a interface de comunicação com o microcontrolador;
4. Implementar a biblioteca;
5. Testar o funcionamento e a viabilidade;
6. Analisar os resultados alcançados;

## Estrutura do Trabalho

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

# FUNDAMENTAÇÃO TEÓRICA

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

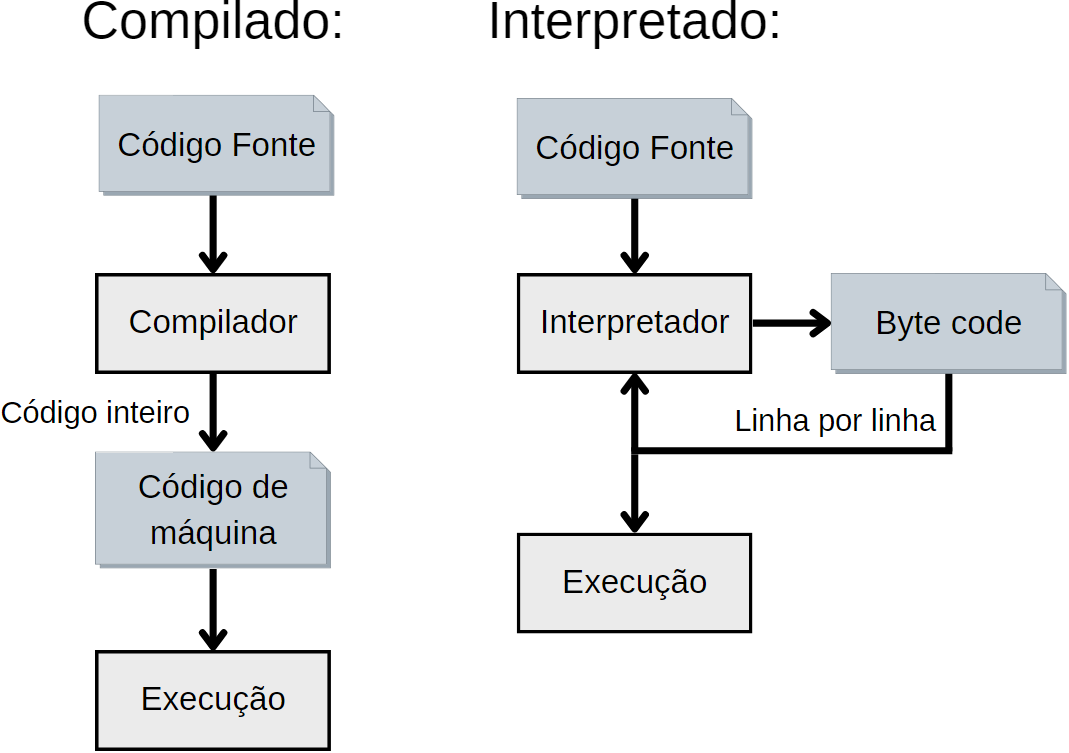
## Programação textual

De acordo com Sammet (1969), uma linguagem de programação é um conjunto de caracteres que possuem regras para combiná-los e uma especificação de seus efeitos quando executados em um computador. O nível de abstração deve ser tal que os usuários não precisem conhecer código de máquina e para que a linguagem seja amplamente independente, permitindo que o mesmo programa seja executado em computadores com arquiteturas diferentes.

As linguagens de programação podem ser divididas em duas categorias, sendo elas linguagens compiladas e linguagens interpretadas.

Em uma linguagem compilada, um compilador pré-traduz o programa em código de máquina para uma plataforma específica, podendo assim ser carregado na memória do computador e executado diretamente por ele (WATSON, 2017). Nas linguagens interpretadas não ocorre esta antecipação, o código é executado diretamente na plataforma pelo interpretador, traduzindo o programa em código de máquina linha por linha simultaneamente com a execução do programa (KWAME; MARTEY; CHRIS, 2017).

**Figura 1 – Código compilado e interpretado**



Fonte: Elaboração própria (2022).

Os microcontroladores são tipicamente programados em linguagens como Assembly, C e C++, como mostrado atualmente por Mazzei et al. (2015), devido ao desenvolvimento da tecnologia e ao aumento da capacidade de memória dos microcontroladores, é possível implementar máquinas virtuais em ambientes embarcados assim podendo interpretar linguagens modernas como Java, Python e MicroPython.

* + 1. Linguagem C

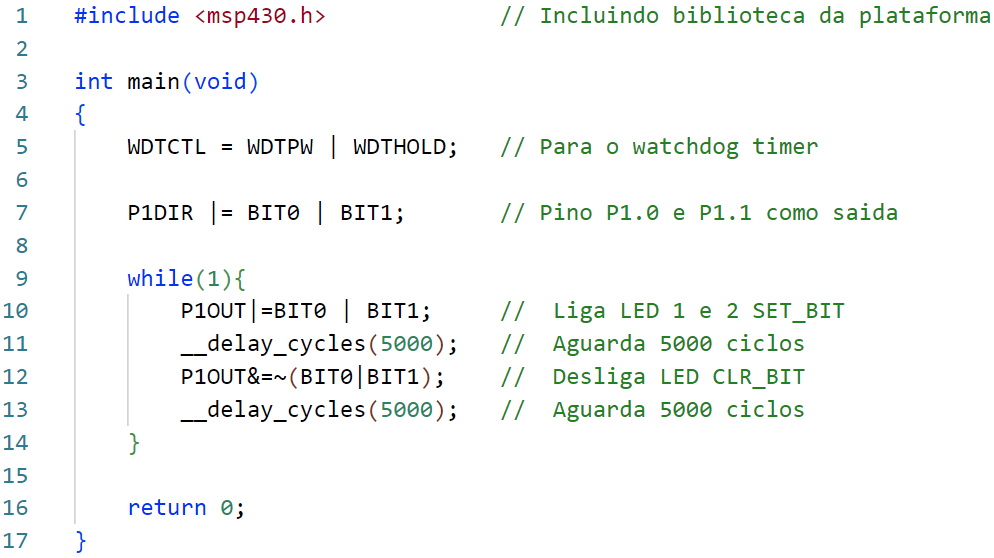
Como o criador da linguagem Ritchie (1993) afirma, C surgiu entre os anos de 1969 e 1973 como uma linguagem para ser utilizada no sistema operacional Unix, mas apenas em meados de 1980 a linguagem foi padronizada pelo comitê ANSI, fazendo seu uso se espalhar e colocando-a entre as linguagens mais usadas em toda a indústria de computadores.

C é uma linguagem de programação compilada, de uso geral, com paradigmas estruturais, imperativos e procedurais, possui uma sintaxe reduzida, controle de fluxo moderno, estruturas de dados e um rico conjunto de operadores. Sua ausência de limitações e sua generalidade a tornam mais conveniente e eficiente para muitas tarefas do que linguagens supostamente mais poderosas (KERNIGHAN; RITCHIE, 1988).

Os microcontroladores são tradicionalmente programados em linguagem assembly, como Pardue (2015) afirma, cada plataforma possui sua própria sintaxe, dificultando o aprendizado e tornando o código obsoleto no momento em que se muda de família. Alunos, professores e hobistas tendem a pular o estudo detalhado de assembler e ir direto para C, que é mais simples e eficiente. Devido a quantidade crescente de compiladores e sua facilidade de implementação, C se tornou a linguagem padrão para programação em microcontroladores (BATES, 2008).

Todos os microcontroladores possuem periféricos e outros componentes que devem ser acessados e controlados ​​pelo programador. A linguagem C permite configurar e inicializar os bits sinalizadores nos registradores de controle para cada operação, além de facilitar o acesso aos endereços e a memória interna do dispositivo, dando ao programador total controle da plataforma que está sendo utilizada.

**Figura 2 – Exemplo de código escrito em C embarcado**



Fonte: Elaboração própria (2022).

* + 1. Linguagem C++

Como Stroustrup (2013) o criador da linguagem explica, C++ foi desenvolvida em 1983, projetada para tornar a programação mais eficiente e moderna. C++ é uma extensão da linguagem de programação C, dessa forma ela possui retrocompatibilidade com os códigos já existentes escritos em C.

C++ é uma linguagem de programação compilada, de propósito geral, assim como C, mas com paradigmas modernos, como abstração de dados, orientação a objetos e programação genérica. Ao longo do tempo, a linguagem passou por várias atualizações, a principal adição foi uma nova biblioteca padrão chamada STL em 1998, que forneceu ferramentas úteis como contêineres, algoritmos e iteradores de contêineres (STROUSTRUP, 2013).

Entre as maiores diferenças com C, pode-se afirmar que a programação orientada a objetos aplicada em C++ reduz o tempo de desenvolvimento, devido a facilidade de reutilização de código, e também a alocação dinâmica de memória utilizando ponteiros inteligentes, que diminui drasticamente a chance de falhas de segurança, em relação à linguagem C, estes pontos fazem a linguagem ser mais popular em projetos com grandes equipes (BARR, 1999).

C++ foi projetado para programação de sistemas e software embarcado, focado em grandes sistemas e com recursos limitados, com desempenho, eficiência e flexibilidade de uso como destaques do design (STROUSTRUP, 2014). Portanto, devido a todos os fatores, a utilização de C++ em microcontroladores se torna uma experiência mais moderna e completa quando se comparada com a linguagem C.

* + 1. MicroPython
    2. Sintaxe de linguagens textuais
    3. Ensino

## Subtítulo Secundário 2

As citações diretas com menos de três linhas “devem estar entre aspas e devem mostrar entre parênteses o ano e a página da obra consultada.” (AUTOR, ano, página). Já as citações com mais de três linhas devem ser recuadas da margem esquerda em 4 cm, tamanho da fonte 10, espaçamento simples e texto sem aspas (ABNT, 2002, p. 2).

Texto texto texto texto texto texto texto texto. Texto texto texto texto texto texto. Texto texto texto texto texto texto texto texto texto. Texto texto texto texto texto texto Texto texto texto texto texto texto. Texto texto texto texto texto texto. (AUTOR, ano, página).

### Subtítulo Terciário

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

#### Subtítulo Quaternário

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto, conforme exposto na Figura 1.

**Figura 1 – Motor Weg**



Fonte: WEG (2014).

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto como indica a Tabela 1.

**Tabela 1 – Produção de petróleo na Bahia**

| **Ano** | **Produção (1.000 t)** |
| --- | --- |
| **1996** | **2.536** |
| **1997** | **2.665** |
| **1998** | **3.056** |
| **1999** | **3.567** |

Fonte: Adaptado de ANP (2000).

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto. Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto. Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto. Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto, como evidencia a Figura 2.

**Figura 2 – Diagrama Fasorial**



Fonte: Silva (2020).

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto, conforme mostra a Equação 1.

|  | (1) |
| --- | --- |

**Quadro 1 – Tipos de energia analisados**

| **Ano** | **Tipos de energia** |
| --- | --- |
| 2017 | Mecânica |
| 2018 | Térmica |
| 2019 | Elétrica |
| 2020 | Química |
| 2021 | Atômica |

Fonte: Elaboração própria (2021).

# METODOLOGIA

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

## Métodos aplicados

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

# APRESENTAÇÃO DOS RESULTADOS

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

## Análise e discussão dos resultados

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

# CONSIDERAÇÕES FINAIS

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

## Sugestões para trabalhos futuros

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

**REFERÊNCIAS**

ALRUBAYE, H.; LUDI, S.; MKAOUER, M. **Comparison of block-based and hybrid-based programming environments in transferring programming skills to text-based environment**. Rochester Institute of Technology. 2019. Disponível em: https://doi.org/10.48550/arXiv.1906.03060. Acesso em: 12 out. 2022.

BARR, Michael. **Programming embedded systems in C and C++**. O'Reilly Media, Inc., 1999.

BATES, Martin P. **Programming 8-bit PIC microcontrollers in C: with interactive hardware simulation**. Newnes, 2008.

CABALLAR, Rina Diane. **Programming Without Code: the rise of no-code software development**. 2020. Disponível em: https://spectrum.ieee.org/programming-without-code-no-code-software-development. Acesso em: 14 out. 2022.

CABRERA, Lautaro; MALONEY, John H.; WEINTROP, David. **Programs in the Palm of your Hand: how live programming shapes children's interactions with physical computing devices**. 2019. Association For Computing Machinery, New York. Disponível em: https://dl.acm.org/doi/10.1145/3311927.3323138. Acesso em: 16 out. 2022.

GOOGLE. **Blockly**. Disponível em: https://developers.google.com/blockly. Acesso em: 14 out. 2022.

JUNIOR, A. G. D. S.; GONÇALVES, L. M. G.; CAURIN, G. A. D. P.; TAMANAKA, G. T. B.; HERNANDES, A. C.; & AROCA, R. V. **BIPES: Block Based Integrated Platform for Embedded Systems**. IEEE Access, 2020. Disponível em: https://ieeexplore.ieee.org/document/9246562

KERNIGHAN, Brian W.; RITCHIE, Dennis M. **The C programming language**. Pearson Education, 1988.

KWAME, Ampomah Ernest; MARTEY, Ezekiel Mensah; CHRIS, Abilimi Gilbert. **Qualitative assessment of compiled, interpreted and hybrid programming languages**. Communications on Applied Electronics (CAE), v. 7, n. 7, p. 8-13, 2017.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY(MIT). **Scratch**. Disponível em: https://scratch.mit.edu/. Acesso em: 14 out. 2022.

MAZZEI, D.; MONTELISCIANI, G.; BALDI, G.; FANTONI, G. **Changing the programming paradigm for the embedded in the IoT domain**. IEEE 2nd World Forum on Internet of Things (WF-IoT). IEEE, 2015. p. 239-244.

MICHELI, Giovanni de; SAMI, Mariagiovanna. **Hardware/Software Co-Design**. Londres: Kluwer Academic Publishers, 1996.

PARDUE, Joe. **C programming for microcontrollers**. Smiley Micros. com, 2005.

RITCHIE, Dennis M. **The development of the C language**. ACM Sigplan Notices, v. 28, n. 3, 1993.

SAMMET, Jean E. **Programming languages: History and fundamentals**. Prentice-Hall, Inc., 1969.

STROUSTRUP, Bjarne. **The C++ programming language**. Pearson Education, 2013.

STROUSTRUP, Bjarne. **The essence of C++**. Video. Edinburgh: The University of Edinburgh, 2014.

WATSON, Des. **Compilers and Interpreters**. Springer, 2017.

ANTUNES, Pedro F. **Ambiente de robótica educacional**. 2016. 120 f. Dissertação

(Mestrado em Engenharia Mecânica) – Universidade Federal de Minas Gerais, Belo

Horizonte, 2016.

BRASIL. **Lei nº 9.887,** **de 7 de dezembro de 1999**. Altera a legislação tributária federal. Disponível em: http://www.in.gov.br/mp\_leis/leis\_texto.asp?Id=LEI%209887. Acesso em: 22 dez. 1999

BRASIL. Ministério do Trabalho e Emprego. **NR 10**: Segurança em instalações e serviços em eletricidade. 7 dez. 2004. Disponível em: http://portal.mte.gov.br/data/files/FF8080812BE4CA7C012BE520074E5264/nr\_10.pdf. Acesso em: 6 maio 2011

CECHINEL, Carolina M. **Estudo da exposição ocupacional dos profissionais das**

**técnicas radiológicas em medicina nuclear**. 58f. 2017. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Radiologia) – Departamento Acadêmico de Saúde e Serviços, IFSC, Florianópolis, 2017. Disponível em: http://sites.florianopolis.ifsc.edu.br/radiologia/files/2017/10/2017-CAROLINA-MARTINSCECHINEL.-ESTUDO-DA-EXPOSI%C3%87%C3%83O-OCUPACIONAL-DOSPROFISSIONAIS0ADAS-T%C3%89CNICAS-RADIOL%C3%93GICAS-EM-MEDICINANUCLEAR.pdf. Acesso em: 03 abr. 2018

FUNDAÇÃO OSWALDO CRUZ. **Cuidado ao paciente**. Disponível em: http://pdf.datasheetcatalog.com/datasheet\_pdf/philips/100124A\_to\_100124Y.pdf. Acesso em: 14 mar. 2018.

SANTOS, Max M. D. **Veículos elétricos e híbridos**: fundamentos, características e aplicações. São Paulo: Érica, 2020.

SILVA, Maria. M. L. Crimes da era digital. **.Net**, Rio de Janeiro, nov. 1998. Seção Ponto de Vista. Disponível em: http://www.brazilnet.com.br/contexts/brasilrevistas.htm. Acesso em: 28 nov. 1998.

SOUZA, L. S.; BORGES, A. L.; REZENDE, J. O. Influência da correção e do preparo do solo sobre algumas propriedades químicas do solo cultivado com bananeiras. In: REUNIÃO BRASILEIRA DE FERTILIDADE DO SOLO E NUTRIÇÃO DE PLANTAS, 21, 1994, Petrolina. **Anais** [...] Petrolina: EMBRAPA, CPATSA, 1994. p. 3-4.

**APÊNDICES**

APÊNDICE A – Título

APÊNDICE B – Título

**ANEXOS**

ANEXO A – Título

ANEXO B – Título