

ROBOT CLASS



Definição do Projeto

Este guia educacional é um recurso com o objetivo de orientar adolescentes em uma jornada de descoberta e aprendizado no campo da robótica. Neste curso, proporcionamos uma abordagem estruturada e ampla, buscando promover conhecimento tecnológico.

A metodologia pedagógica prioriza a participação ativa dos alunos, envolvendo-os em atividades práticas que estimulam o raciocínio lógico, a resolução de problemas e o trabalho em equipe. Através de uma progressão cuidadosamente elaborada, desde conceitos básicos até desafios mais avançados, na busca da garantia de que todos os estudantes possam acompanhar e se desenvolver de acordo com seu ritmo e capacidades individuais.

No Robot Class, vamos começar com os conceitos básicos e, gradualmente, avançar para projetos mais elaborados. Este guia é projetado para ser interativo e prático, permitindo que você aplique os conhecimentos de robótica e programação que você adquiriu em um contexto real e significativo.

Sumário

1. Introdução

1.1 Introdução ao projeto	2
---------------------------------	---

2. Robótica

2.1 O que é robótica?	3
-----------------------------	---

2.2 Aplicações da robótica na vida cotidiana	4
--	---

3. Passo a Passo

3.1 Programação	6
-----------------------	---

3.2 Como funciona	7
-------------------------	---

3.3 Estrutura do código	8
-------------------------------	---

4. Arduino

4.1 O que é Arduino?	12
----------------------------	----

4.2 Componentes do Arduino	13
----------------------------------	----

4.3 Exemplo prático de Arduino	16
--------------------------------------	----

5. Referências Bibliográficas

5.1 Referências do projeto	19
----------------------------------	----

1. Introdução

1.1 Projeto Social: Educação em Robótica para Jovens

Como dito anteriormente, este guia de aprendizado tem como objetivo servir como um recurso para orientar adolescentes em sua jornada de exploração e aprendizado na área da robótica. Oferecemos uma abordagem educacional estruturada e diversificada, com o propósito de fomentar o conhecimento tecnológico.

Nosso método pedagógico enfatiza a participação ativa dos estudantes, integrando-os em atividades práticas que incentivam o pensamento lógico, a resolução de problemas e o trabalho em equipe. Através de uma sequência de progressão cuidadosamente planejada, desde conceitos básicos até desafios mais complexos, nos esforçamos para garantir que todos os alunos possam seguir e se desenvolver de acordo com suas habilidades e ritmo individual.

2. Robótica

2.1 O que é robótica?

Robótica é a ciência que estuda as tecnologias associadas à construção de robôs.

Os robôs são mecanismos automáticos que utilizam circuitos integrados para realizarem atividades e movimentos humanos simples ou complexos. A robótica tem grande aplicação em diversas áreas desde a produção industrial, medicina até atividades domésticas.

O conceito sobre robótica surgiu no início do século XX, na obra “O Mentiroso” do autor de ficção científica Isaac Asimov. Foi o autor quem criou a palavra ‘robótica’ e foi também ele quem apresentou, no campo da ficção-científica, as Leis da Robótica.

O termo foi popularizado apenas em 1950 por conta do livro “Eu, Robô”, do mesmo autor. A obra levantou diversas discussões sobre a relação entre homens e máquinas.

A ideia da criação de máquinas é antiga. Na Grécia Antiga, acreditava-se que os gregos e romanos já desenvolviam diferentes tipos de máquinas capazes de realizar movimentos automatizados.

A edição 2023 do torneio de robótica “First Lego League” buscou abranger neste ano como tema da temporada “MasterPiece” e o tema geral é mundo das artes aliado às tecnologias e o torneio tem como objetivo estimular a exploração científica.

2.2 Aplicações da robótica na vida cotidiana

A robótica tem várias aplicações na vida cotidiana, impactando diversos setores e tornando a vida mais conveniente e eficiente. Aqui estão algumas das principais aplicações:

- **Saúde:** Robôs cirúrgicos aprimoram procedimentos, possibilitando cirurgias menos invasivas em áreas como coração, estômago e cérebro;
- **Indústria Automobilística:** Robôs lideram soldagem e montagem, impulsionando produtividade e reduzindo custos;
- **Doméstica:** De limpeza a assistência a idosos, robôs tornam-se assistentes em nossas casas;
- **Inovação Tecnológica:** Integração da robótica em dispositivos cotidianos e eletrônicos é cada vez mais comum;
- **Militar:** Robôs monitoram terrenos, proporcionando imagens em tempo real para decisões rápidas.
- **Tecnologia:** Assistentes pessoais e dispositivos controlados por voz simplificam atividades diárias;

- **Educação:** Plataformas interativas com robôs enriquecem a educação, ensinando habilidades técnicas e científicas;
- **Transporte Autônomo:** Carros autônomos prometem dirigir independentemente, impactando positivamente o trânsito;
- **Impressão 3D:** Impressoras 3D constroem estruturas complexas, revolucionando a construção civil;
- **Nanotecnologia:** Nano robôs visam otimizar tratamentos médicos, minimizando efeitos colaterais.

3. Passo a passo

3.1 Programação

Programação é o processo de escrever instruções que serão realizadas pelo computador. Em essência, é a arte de desenvolver programas de computador por meio de um conjunto de comandos que orientam o computador.

Os programadores utilizam linguagens de programação para controlar o comportamento das máquinas, especialmente dos computadores. Essas linguagens permitem que os programadores se comuniquem com as máquinas de maneira mais clara e manipulável do que a linguagem nativa. As linguagens mais comuns incluem Python, Ruby, JavaScript, C++ e outras, e são usadas para desenvolver *softwares*, aplicativos, *websites* e jogos.

3.2 Como funciona?

Como dito no tópico acima, programação é o processo de escrever instruções em uma linguagem de programação específica no qual permite ao computador executar tarefas e resolver problemas. Aqui está uma visão geral de como funciona:

- **Código Fonte:** O código fonte é o conjunto de palavras e comandos escritos em uma linguagem de programação.
- **Algoritmos:** Algoritmos são sequências lógicas de passos para realizar uma tarefa.
- **Linguagens de Programação:** Existem muitas linguagens de programação, cada uma com suas particularidades e usos específicos. Alguns exemplos de linguagens de programação populares incluem C++, Objective-C, C#, e Swift.
- **Habilidades do Programador:** Além de escrever código, os programadores precisam ter habilidades cognitivas, como raciocínio lógico, habilidades matemáticas e capacidade de abstração.
- **Processo de Desenvolvimento:** O processo de programação geralmente começa com a definição de um problema e a criação de um algoritmo para resolvê-lo.
- **Execução do Programa:** Quando o programa é executado, o computador segue as instruções do código fonte para realizar a tarefa desejada.

3.3 Estrutura do Código

Variáveis: São espaços de memória que armazenam valores. Cada variável tem um nome e um tipo de dado que determina o tipo de valor que ela pode conter. Exemplos de tipos de dados incluem:

- **Ponto Flutuante (float):** armazena números decimais, como 3.14.

```
#include <iostream>

int main() {
    float numero = 3.14159; // Declaração de uma variável de ponto flutuante
    std::cout << "O valor de PI é aproximadamente: " << numero << std::endl;
    return 0;
}
```

- **Texto (string):** armazena sequências de caracteres, como "Olá, mundo!".

```
#include <iostream>
#include <string>

int main() {
    std::string minhaString = "Olá, Mundo!"; // Declaração e atribuição de uma string
    std::cout << minhaString << std::endl; // Impressão da string
    return 0;
}
```

- **Booleano (bool):** armazena valores verdadeiros (true) ou falso (false).

```
#include <iostream>

int main() {
    bool estaVerdadeiro = true; // Declaração e atribuição de um valor booleano
    std::cout << "O valor é: " << (estaVerdadeiro ? "Verdadeiro" : "Falso") << std::endl;
    return 0;
}
```

Estruturas de Controle: Mecanismos que permitem a execução condicional ou repetitiva de instruções dentro de um programa:

- **Condicionais (if, else if, else):** Permite executar um bloco de código se uma condição for verdadeira. Caso contrário, você pode fornecer alternativas. Exemplo:

```
#include <iostream>

int main() {
    int idade = 18; // Suponha que esta é a idade de uma pessoa

    if (idade >= 18) { // Verifica se a pessoa é maior ou igual a 18 anos
        std::cout << "Você é maior de idade." << std::endl;
    } else {
        std::cout << "Você é menor de idade." << std::endl;
    }

    return 0;
}
```

- **Loops (for, while, do-while):** Permitem executar um bloco de código repetidamente enquanto uma condição for verdadeira. Útil para processar listas de dados ou realizar tarefas repetitivas. Exemplo:

```
#include <iostream>

int main() {
    for (int i = 1; i <= 10; i++) { // Inicialização, condição e incremento
        std::cout << i << " "; // Comando dentro do laço
    }
    std::cout << std::endl;
    return 0;
}
```

Estruturas de Dados: São maneiras de organizar e armazenar dados para que possam ser usados de forma eficiente. Exemplos incluem:

- **Arrays:** Coleções ordenadas de elementos do mesmo tipo, acessados por um índice.

```
#include <iostream>

int main() {
    int numeros[5] = {1, 2, 3, 4, 5};
    // Impressão dos elementos do array
    for (int i = 0; i < 5; i++) {
        std::cout << numeros[i] << " ";
    }
    std::cout << std::endl;

    return 0;
}
```

- **Operadores Lógicos e Aritméticos:** São símbolos usados para realizar operações matemáticas e lógicas, como adição (+), subtração (-), multiplicação (*), divisão (/), e comparações (==, <, >, etc.).

- **Listas:** Coleções flexíveis de elementos, onde os elementos podem ser adicionados ou removidos dinamicamente.

```
#include <iostream>
#include <list>

int main() {
    std::list<int> minhaLista;
    // Inserção de elementos na lista
    minhaLista.push_back(1);
    minhaLista.push_back(2);
    minhaLista.push_back(3);
    // Iteração sobre a lista e impressão dos elementos
    for (int num : minhaLista) {
        std::cout << num << " ";
    }
    std::cout << std::endl;

    return 0;
}
```

- **Funções:** Blocos de código reutilizáveis que realizam uma tarefa específica e podem ser chamados pelo nome da função em diferentes partes do programa.

```
#include <iostream>
//Função
void imprimirHelloWorld() {
    std::cout << "Hello World!" << std::endl;
}
int main() {
    imprimirHelloWorld();
    return 0;
}
```

4. Arduino

4.1 O que é Arduino?

Arduino é uma plataforma que possibilita o desenvolvimento de projetos eletrônicos. Em outras palavras, é uma plataforma de prototipagem eletrônica.

O Arduino é constituído de *hardware* e *software*, tornando assim possível a realização de diversos projetos tecnológicos. A placa tem como principal componente o *microcontrolador*, que é um tipo de processador bem menor do que o convencional. O *microcontrolador* executa os programas e avalia qualidade das entradas e saídas, ou seja, dos canais pelos quais é possível a comunicação entre mundo externo e digital.

O Arduino funciona a partir de códigos de programação, onde pode ser livremente destinado a diversos tipos de funções, podendo comandar desde sensores eletrônicos até módulos de alta complexidade.

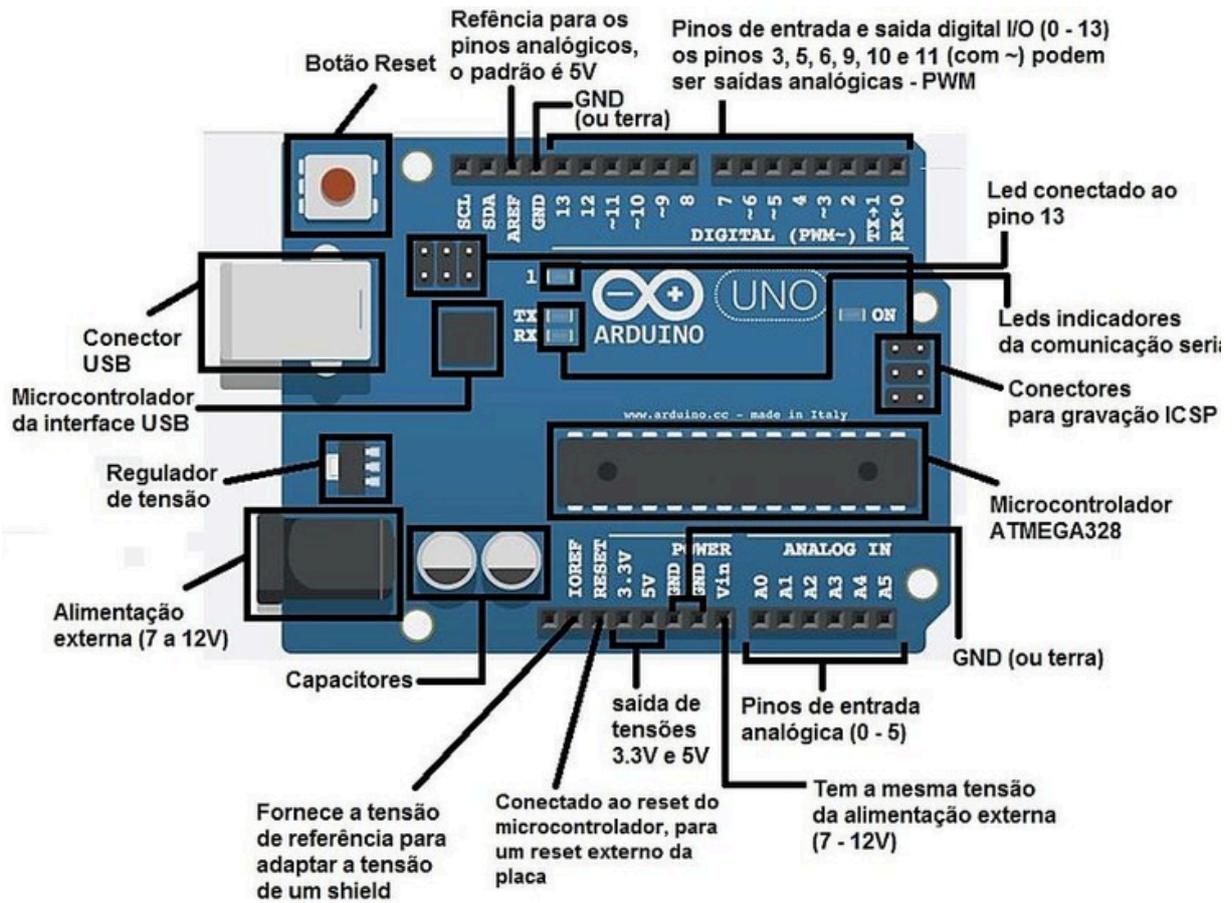
A programação é feita por meio do programa IDE Arduino, o qual pode ser baixado diretamente no site oficial arduino.cc, além disso, a conexão com o computador é feita via cabo usb, permitindo que os comandos definidos no programa sejam devidamente transferidos até a placa.

4.2 Componentes principais do Arduino

Um Arduino é composto principalmente pelos seguintes componentes:

- **Microcontrolador:** Este é o cérebro do Arduino, responsável por executar o código escrito. Os modelos Arduino geralmente utilizam microcontroladores da família Atmel AVR, embora alguns modelos mais recentes usam microcontroladores ARM.
- **Portas de Entrada/Saída (I/O):** As portas de entrada/saída permitem que o Arduino realize tarefas. Elas podem ser digitais (capazes de ler ou escrever sinais digitais, ou seja, ligado/desligado) ou analógicas (capazes de ler sinais analógicos, como uma voltagem variável).
- **Pinos de Alimentação:** Estes são os pinos onde você conecta a fonte de energia ao Arduino. Geralmente, há um pino para a alimentação de 5 volts e um ou mais pinos para aterramento (GND).
- **Conector USB:** O conector USB permite a conexão de um Arduino a um computador para programação e comunicação serial.

- **Regulador de Tensão:** O regulador de tensão converte a tensão de entrada (geralmente 7-12V) para a voltagem de operação do Arduino (geralmente 5V ou 3.3V), garantindo que o microcontrolador e outros componentes funcionem corretamente.
- **Cristal Oscilador:** Este componente fornece ao microcontrolador a base de tempo necessária para executar o código de forma precisa.
- **Botão de Reset:** Este botão permite reiniciar o programa em execução no Arduino.
- **LEDs de Indicação:** Muitos Arduinos vêm com LEDs integrados que podem ser úteis para indicar o status do dispositivo, como alimentação ou atividade de comunicação.



4.3 Exemplo prático de Arduino

Neste exemplo, utilizaremos a plataforma Tinkercad.

Tinkercad é uma plataforma de desenvolvimento de projetos eletrônicos que permite aos usuários criar, testar e compartilhar seus projetos de hardware e software de maneira colaborativa. A plataforma é baseada em um modelo de computação em nuvem, o que significa que você pode trabalhar em seus projetos de qualquer lugar, acessando-os através de um navegador da web.



Link de acesso ao Tinkercad: <https://www.tinkercad.com/>

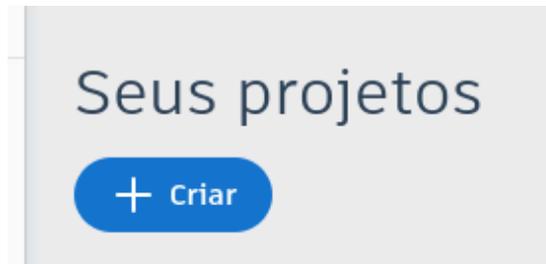
Aqui está um exemplo prático de como é possível usar o Tinkercad para criar um protótipo simples com um Arduino. Vamos criar um projeto que lê um valor analógico de um sensor e exibe o valor na tela serial do Arduino.

Passo 1: Crie uma nova conta no Tinkercad.

- Acesse o site do Tinkercad e crie uma nova conta.
- Depois de criar sua conta, você será levado para a página inicial do Tinkercad, onde você pode começar a criar seu primeiro projeto.

Passo 2: Crie um novo projeto

- Clique em "Criar", em seguida selecione "Circuito" e dê um nome ao seu projeto, como "LedLigaDesliga".

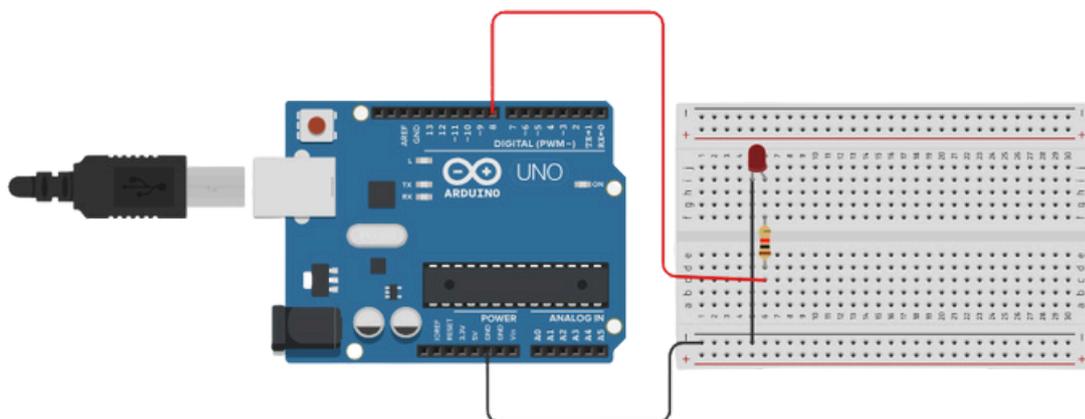


- Selecione o tipo de hardware que você está usando, neste caso, um Arduino (de modo mais específico, um Arduino Uno R3).



Passo 3: Adicione componentes ao seu projeto

- No painel do projeto, clique em "Componentes".
- Organize os componentes como na imagem abaixo.



Passo 4: Escreva o código

- No painel do projeto, clique em "Código" e comece a escrever o código para ler o valor do sensor e exibi-lo na tela serial.
- Aqui está um exemplo básico de código que você pode usar:

```
int vermelho = 8;

void setup()
{
  pinMode(vermelho, OUTPUT);
}

void loop()
{
  digitalWrite(vermelho, HIGH);
  delay(5000); // Wait for 1000 millisecond(s)
  digitalWrite(vermelho, LOW);
  delay(5000); // Wait for 1000 millisecond(s)
}
```

Passo 5: Teste seu projeto

- Depois de escrever o código, clique em "Iniciar simulação" para simular o funcionamento do seu projeto.



Passo 6: Compartilhe seu projeto

- Quando estiver satisfeito com seu projeto, clique em "Compartilhar" para permitir que outros vejam e interajam com seu projeto.

Referências Bibliográficas

ENGENHARIA HÍBRIDA. "História da robótica: marcos importantes". Disponível em: <https://www.engenhariahibrida.com.br/post/explorando-a-fascinante-historia-da-robotica>. Acesso em: 11 fev. 2024.

LINKEDIN. "Explorando a fascinante história da robótica". Disponível em: <https://www.linkedin.com/pulse/explorando-fascinante-hist%C3%B3ria-da-rob%C3%B3tica-engenhariahibrida>. Acesso em: 11 fev. 2024.

LINKEDIN. "Robótica cotidiano". Disponível em: <https://pt.linkedin.com/pulse/rob%C3%B3tica-cotidiano-renan-assun%C3%A7%C3%A3o-chaves>. Acesso em: 11 fev. 2024.

PEOPLE.COM.BR. "7 maneiras como a robótica pode mudar o mundo". Disponível em: <https://www.people.com.br/noticias/robotica/7-maneiras-como-a-robotica-pode-mudar-o-mundo>. Acesso em: 11 fev. 2024.

PORTAL DA INDÚSTRIA. "Robótica". Disponível em: <https://www.portaldaindustria.com.br/industria-de-a-z/robotica/>. Acesso em: 11 fev. 2024.

RESEARCHGATE.NET. "Partes principais da placa Arduino Uno". Disponível em: <https://www.researchgate.net/publication/353380465/figure/fig4/AS:1048329521733643@1626952643134/Partes-principais-da-placa-Arduino-Uno.jpg>. Acesso em: 10 fev. 2024.

TERRA. "Robôs na vida moderna". Disponível em: <https://www.terra.com.br/byte/ciencia/robos-na-vida-moderna,bbcae160f4d8f310VgnCLD2000000ec6eb0aRCRD.html>. Acesso em: 11 fev. 2024.

UNIVERSAL ROBOTS. "A história da robótica dos autômatos antigos aos cobots e outros robôs modernos". Disponível em: <https://www.universal-robots.com/br/blog/a-hist%C3%B3ria-da-rob%C3%B3tica-dos-aut%C3%B4matos-antigos-aos-cobots-e-outros-rob%C3%B4s-modernos/>. Acesso em: 11 fev. 2024.

UNIVERSAL ROBOTS. "Os robôs vão dominar o mundo: perspectivas industriais da automação e robótica". Disponível em: <https://www.universal-robots.com/br/blog/os-rob%C3%B4s-v%C3%A3o-dominar-o-mundo-perspectivas-industriais-da-automac%C3%A3o-rob%C3%B3tica/>. Acesso em: 11 fev. 2024.

USINAINFO. "O que é Arduino?". Disponível em: <https://www.usinainfo.com.br/blog/o-que-e-arduino/>. Acesso em: 11 fev. 2024.

VICTORVISION. "O que é Arduino?". Disponível em: <https://victorvision.com.br/blog/o-que-e-arduino/>. Acesso em: 17 fev. 2024.

VIDESILICIO.COM.BR. "Arduino Uno". Disponível em: <https://i0.wp.com/portal.vidadesilicio.com.br/wp-content/uploads/2017/05/Arduino-Uno.jpg?resize=712%2C388&ssl=1>. Acesso em: 09 fev. 2024.

FVML. "Linguagem de programação utilizada". Disponível em: <https://www.fvml.com.br/2022/11/linguagem-de-programacao-utilizada-arduino.html>. Acesso em: 10 fev. 2024.

HOREADODECODAR. "O que é programação?". Disponível em: <https://horadecodar.com.br/o-que-e-programacao/>. Acesso em: 10 fev. 2024.

MANUALDOMAKER. "A linguagem do Arduino é C ou C++?". Disponível em: <https://www.manualdomaker.com/article/a-linguagem-do-arduino-e-c-ou-c/>. Acesso em: 10 fev. 2024.

TINKERCAD. Imagem do Tinkercad. Disponível em: <https://yt3.googleusercontent.com/ytc/Alf8zZQcG5Za9Z8TVJbbQBduCWm3TW6Qp3x4kHiRqpXGuQ=s900-c-k-c0x00ffffff-no-rj>. Acesso em: 18 fev. 2024.

TRIPLETEN. "O que é programação e como ela funciona?". Disponível em: <https://tripleten.com.br/blog/o-que-e-programacao-e-como-ela-funciona/>. Acesso em: 24 fev. 2024.