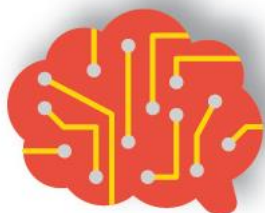
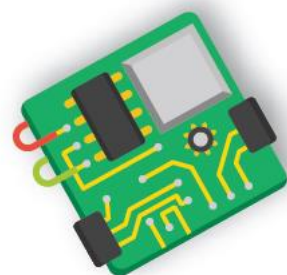
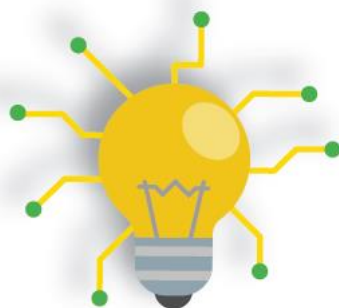


COMPUTAÇÃO & O MUNDO

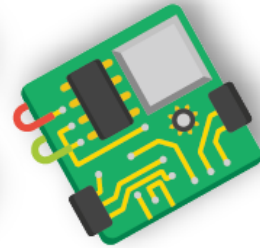
9º Ano



Livro do
Estudante

Luis Gustavo de Jesus Araujo
Bianca Leite Santana
Roberto Almeida Bittencourt

COMPUTAÇÃO & O MUNDO



Livro do Estudante - 9° ano
Projeto Computação Fundamental

Luis Gustavo de Jesus Araujo
Bianca Leite Santana
Roberto Almeida Bittencourt

<https://sites.google.com/view/computacaofundamental/>

Versão 1.0

Versão 1.0

Autores: Luis Gustavo de Jesus Araujo, Bianca Leite Santana e Roberto Almeida Bittencourt

Esta obra está sob licença Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0). Quaisquer dúvidas quanto a permissões consulte o link: <https://creativecommons.org/licenses/by-sa/4.0/>



Neste livro, nas aulas da Unidade I, utilizamos imagens de comandos e personagens do Scratch, que é um projeto da Fundação Scratch em colaboração com o grupo Lifelong Kindergarten no MIT Media Lab e está disponível gratuitamente em <https://scratch.mit.edu>. Tais imagens estão disponíveis sob a licença Attribution-ShareAlike 2.0 International (CC BY-SA 2.0). Nas aulas da Unidade IV, utilizamos imagens de comandos do mBlock, que é um projeto inteiramente baseado em Scratch e mantido pela Makeblock e está disponível gratuitamente em <https://www.mblock.cc/en-us/>. Também nas aulas da Unidade IV, usamos a plataforma de hardware livre Arduino, cujos dispositivos estão licenciados através da licença Attribution-ShareAlike (CC BY-SA). Algumas das aulas deste livro baseiam-se em ideias disponíveis em atividades do livro *Unplugged Computing... off-line activities and games for all ages* (1998), por Tim Bell, Ian H. Witten e Mike Fellows.

FICHA CATALOGRÁFICA

A663c Araujo, Luis Gustavo de Jesus.

Computação e o mundo: livro do estudante / Luis Gustavo de Jesus
Araujo, Bianca Leite Santana, Roberto Almeida Bittencourt. – Feira de
Santana: [s.n.], 2020.
140p.: il.

Edição do autor.
ISBN: 978-65-00-01450-1

1. Ciência da Computação 2. Programação de Computadores. 3. Pensamento
Computacional I. Título

CDD: 004
CDU: 004

Sumário

Sobre o Livro	01
Visão Geral	06

UNIDADE I - Simulações

Aula 01 - Impacto da Tecnologia na Cultura Humana	10
Aula 02 - Minha nova fábrica de chocolate	13
Aula 03 - A matemática do movimento	19
Aula 04 - Modelo Atômico de Bohr	23
Aula 05 - Sistema Solar	28
Aula 06 - Fazendo seu Projeto de Ciências (parte 1)	32
Aula 07 - Fazendo seu Projeto de Ciências (parte 2)	36

UNIDADE II - Manipulação de Imagens

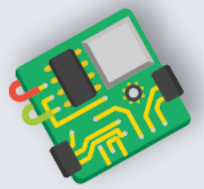
AULA 1 - Pixel a Pixel	40
AULA 2 - Fazendo Seus Próprios Filtros	42
AULA 3 - Mais Efeitos Utilizando Matemática	46
AULA 4 - Mais efeitos visuais	50
AULA 5 - Brincando com Chroma key	54
AULA 6 - Gerador de Memes	57
AULA 7 - Cidade na Lama	60

UNIDADE III - Python

AULA 1 - Fluxograma	64
AULA 2 - Apertadinho	72
AULA 3 - Enquetes	75
AULA 4 - Criptografia de César	78
AULA 5 - Jogo da Força	85
AULA 6 - A Droga da Obediência	90
AULA 7 - Ovos da Páscoa	93
AULA 8 - Jogo da Velha	96

UNIDADE IV - Arduino

AULA 1 - Construindo um Semáforo	102
AULA 2 - Medindo a Luminosidade em um Ambiente	108
AULA 3 - Criando um Piano com o Arduino	112
AULA 4 - Criando um Robô Dançarino	117
AULA 5 - Criando um Robô Guardião do Tesouro	121
AULA 6 - Como funciona o Computador?	125
AULA 7 - Criando seu Primeiro Protótipo com o Arduino	133
AULA 8 - Apresentando seu Primeiro Protótipo com o Arduino	135



Sobre o Livro

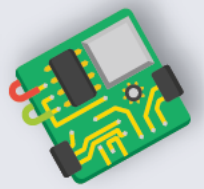
Este livro é concebido para os estudantes do nono ano do ensino fundamental. O objetivo principal deste livro é conectar os conhecimentos dos estudantes sobre a computação com o mundo em que vivem, além de aprofundar o uso da computação para expressar ideias na forma de software ou de hardware através da solução criativa de problemas de interesse dos estudantes e do mundo.

Na primeira unidade, iniciamos a conexão entre os conhecimentos prévios de pensamento computacional e programação em linguagens de blocos com o mundo das ciências naturais. Atividades de simulação de fenômenos científicos são propostas para os estudantes, permitindo aplicar seus conhecimentos prévios na expressão matemática e lógica de problemas do mundo real. No final da unidade, os estudantes fazem um projeto de simulação de ciências de seus próprios interesses.

Na segunda e terceira unidades, os estudantes fortalecem seu domínio do uso de uma linguagem de programação textual (Python), já iniciado no ano anterior através do contexto metodológico de figuras geométricas e da biblioteca Turtle. Na segunda unidade, o contexto metodológico de imagens digitais permite que eles reapliquem os conceitos de sequência, seleção, repetição e funções, adicionando algumas estruturas de dados básicas como vetores unidimensionais e bidimensionais. As imagens digitais permitem o feedback imediato sobre a execução do código do mesmo modo que o palco do Scratch e a tela do Turtle. Na terceira unidade, movemos para o contexto metodológico mais tradicional de programação usando entradas e saídas textuais. A linguagem permanece Python, para manter a familiaridade, e os estudantes aprendem a usar variáveis inteiras, reais e textuais, expressões lógicas e aritméticas e as estruturas de programação e de dados previamente aprendidas para construir jogos e pequenos programas. O uso de strings é potencializado com jogos de palavras. Assim, é possível se aproximar de um contexto mais tradicional, mas mantendo uma abordagem lúdica.

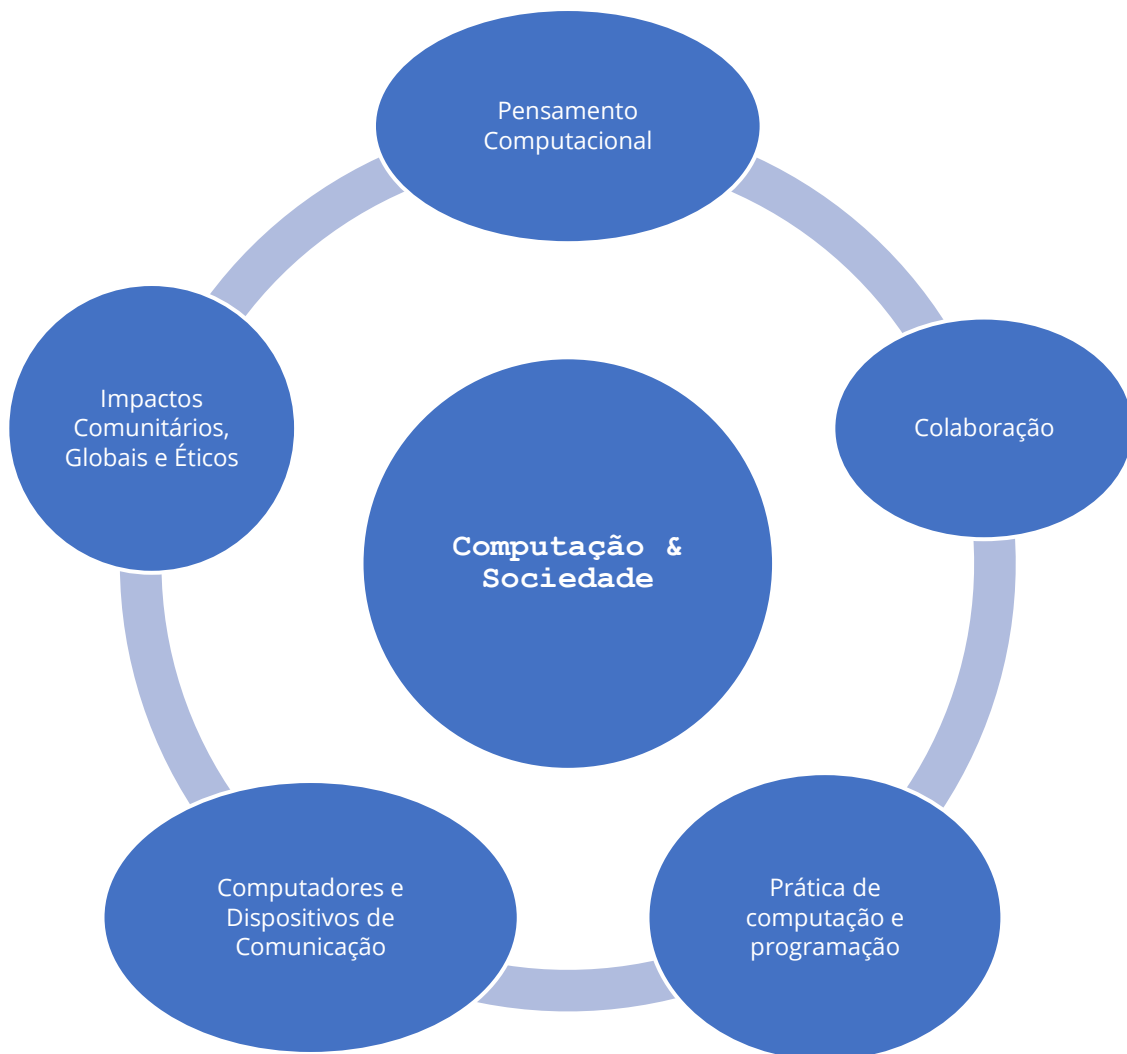
Finalmente, na quarta unidade, introduzimos o contexto metodológico de hardware livre através da plataforma Arduino e de projetos básicos de eletrônica digital. Com este contexto, os estudantes podem fazer seus

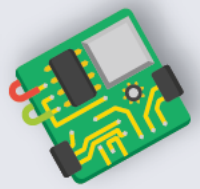




programas em linguagem de blocos comandar dispositivos eletrônicos e fazer simulações de situações realísticas como as existentes em cidades, envolvendo sensores, processadores e atuadores. A conexão com a robótica educacional trabalhada no ano anterior é natural, já que vários dos dispositivos usados se assemelham aos integrados à plataforma de robô usada no ano anterior. Do mesmo modo, os custos escolares são mantidos sob controle com os dispositivos Arduino e componentes eletrônicos de baixo custo.

Ao final de um ano com a proposta Computação e o Mundo, os estudantes exercitam ainda mais os eixos de pensamento computacional, colaboração, práticas de computação e programação e de computadores e dispositivos de comunicação, integrando-os ao mundo em que vivem, além de continuar a discutir os impactos comunitários, globais e éticos da computação.





Pensamento Computacional (PC)

PC1. Descrever como uma simulação pode ser usada para resolver um problema.

PC2. Usar passos básicos na solução de problemas algorítmicos para projetar soluções (por exemplo, declaração e exploração de problemas, exemplos de instâncias, design, implementação de uma solução, teste, avaliação).

PC4. Definir um algoritmo como uma sequência de instruções que pode ser processada por um computador.

PC5. Avaliar maneiras pelas quais diferentes algoritmos podem ser usados para resolver o mesmo problema.

PC6. Trabalhar com algoritmos de busca e classificação.

PC7. Descrever e analisar uma sequência de instruções que estão sendo seguidas (por exemplo, descrever o comportamento de um personagem em um videogame conforme orientado por regras e algoritmos).

PC8. Representar dados de várias formas, incluindo texto, sons, imagens e números.

PC9. Usar representações visuais de estados de problemas, estruturas e dados (por exemplo, gráficos, diagramas, diagramas de rede, fluxogramas).

PC10. Interagir com modelos e simulações de conteúdo específico (por exemplo, ecossistemas, epidemias, dinâmica molecular) para apoiar a aprendizagem e pesquisa.

PC11. Avaliar quais tipos de problemas podem ser resolvidos usando modelagem e simulação.

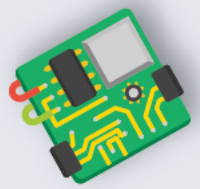
PC12. Analisar o grau em que um modelo de computador representa com precisão o mundo real.

PC13. Usar abstração para decompor um problema em subproblemas.

PC14. Compreender a noção de hierarquia e abstração na computação, incluindo linguagens de alto nível, tradução, conjunto de instruções, e circuitos lógicos.

PC15. Examinar as conexões entre os elementos da matemática e da ciência da computação, incluindo números binários, lógica, conjuntos e funções.

PC16. Fornecer exemplos de aplicações interdisciplinares do pensamento computacional.



Práticas de Computação e Programação (PCP)

PCP5. Reunir e manipular dados usando uma variedade de ferramentas digitais.

PCP10. Demonstrar compreensão dos algoritmos e sua aplicação prática.

PCP11. Implementar soluções de problemas usando uma linguagem de programação, incluindo: comportamento de looping, instruções condicionais, lógica, expressões, variáveis e funções.

PCP13. Identificar carreiras interdisciplinares que são aprimoradas pela ciência da computação.

PCP15. Coletar e analisar dados que são produzidos a partir de várias execuções de um programa de computador.

Colaboração (C)

C6. Criar, desenvolver, publicar e apresentar, de forma colaborativa, produtos (por exemplo, vídeos, podcasts, websites) usando recursos de tecnologia que demonstram e comunicam conceitos de currículo.

C7. Colaborar com colegas, especialistas e outras pessoas usando práticas colaborativas, como programação em pares, trabalho em equipes de projeto e participação em atividades de aprendizado ativo em grupo.

C8. Apresentar disposição necessária para colaboração: fornecer feedback útil, integrar feedback, compreender e aceitar múltiplas perspectivas, socialização.

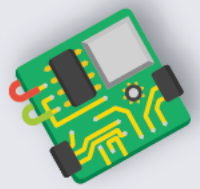
Computadores e Dispositivos de Comunicação (CDC)

CDC1. Reconhecer que os computadores modelam o comportamento inteligente (como encontrado em robótica, reconhecimento de fala e linguagem e animação por computador).

CDC3. Reconhecer que os computadores são dispositivos que executam programas.

CDC4. Identificar uma variedade de dispositivos eletrônicos que contêm processadores computacionais.

CDC5. Demonstrar uma compreensão da relação entre hardware e software.



CDC9. Descrever o que distingue os seres humanos das máquinas, concentrando-se na inteligência humana versus a inteligência das máquinas e nas maneiras pelas quais podemos comunicar.

CDC10. Descrever maneiras pelas quais os computadores usam modelos de comportamento inteligente (por exemplo, movimento do robô, compreensão da fala e da linguagem, e visão computacional).

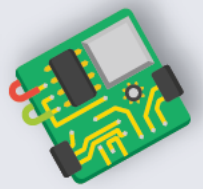
Impactos Comunitários, Globais e Éticos (IC)

IC3. Discutir questões básicas relacionadas ao uso responsável de tecnologia e informação, e as consequências do uso inadequado.

IC6. Identificar o impacto da tecnologia (por exemplo, redes sociais, cyberbullying, computação e comunicação móvel, tecnologias da Web, segurança cibernética e virtualização) na vida pessoal e na sociedade.

IC10. Demonstrar conhecimento das mudanças nas tecnologias da informação ao longo do tempo e os efeitos que essas mudanças têm sobre a educação, local de trabalho e sociedade.

IC14. Discutir como a distribuição desigual de recursos de computação em uma economia global levanta questões de equidade, acesso e poder.



Visão Geral

As aulas estão divididas em quatro unidades. Embora cada unidade tenha um tema central, tópicos dos cinco eixos de objetivos esperados de aprendizagem são trabalhados de maneira concomitante.

Em cada uma das aulas, os estudantes são incentivados a escreverem um diário de bordo descrevendo suas experiências.

UNIDADE I – Simulações

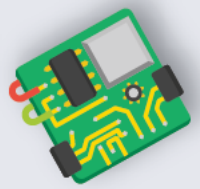
A primeira unidade do programa, composta por 7 aulas, tem como tema central o desenvolvimento de simulações utilizando o Scratch.

	AULA	CONTEÚDO
UNIDADE I	1 Impacto da Tecnologia na Cultura Humana C6, C7, C8, IC3, IC6, IC10 e IC14	Conceito e Exemplos de Tecnologia; Impacto da Tecnologia na vida humana; Questões éticas e o uso da Tecnologia.
	2 Minha nova fábrica de chocolate PC2, C7 e C8	Design de interfaces; Affordances; Mapeamento; Efeitos de transferência; Estereótipos populacionais; Ícones.
	3 A matemática do movimento PC1, PC2, PC4, PC5, PC7, PC10, PC11, PC12, PC15 e PCP11	Movimento uniformemente variado; Scratch; Variáveis, Entrada e saída, Loops, Condicionais, Contadores, Expressões aritméticas.
	4 Modelo Atômico de Bohr Mesmos objetivos da aula anterior	Modelo atômico de Bohr; Scratch; Variáveis, Entrada e saída, Loops, Condicionais, Contadores, Expressões aritméticas.
	5 Sistema Solar Mesmos objetivos da aula anterior	Sistema Solar; Movimento elíptico; Scratch; Variáveis, Entrada e saída, Loops, Condicionais, Contadores, Expressões aritméticas.
	6 Fazendo seu Projeto de Ciências (parte 1) PC1, PC2, PC4, PC5, PC7, PC10, PC11, PC12, PC13, PC15 e PCP11	Conceitos específicos de ciências; Scratch; Variáveis, Entrada e saída, Loops, Condicionais, Contadores, Expressões aritméticas, Funções matemáticas.
	7 Fazendo seu Projeto de Ciências (parte 2) Mesmos objetivos da aula anterior	

Os estudantes entendem como a matemática pode ser utilizada para criar modelos computacionais que representam fenômenos da natureza.

UNIDADE II – Manipulação de Imagens

A segunda unidade do programa, composta por 7 aulas, trabalha com a manipulação de imagens através da implementação de programas em linguagem Python.



	AULA	CONTEÚDO
UNIDADE II	1 Pixel a Pixel PC2, PC5, PC8, PC12, PC15, PCP10 e PCP11	Imagens Digitais; Pixels; Padrão de cores RGB; Python; Variáveis, sequência, loops simples e funções de bibliotecas.
	2 Fazendo Seus Próprios Filtros PC2, PC5, PC8, PC15, PCP10 e PCP11	Efeitos de imagem; Pixels; Funções do JES; Variáveis, expressões aritméticas, sequência, loops, listas e funções.
	3 Mais Efeitos Utilizando Matemática Mesmos objetivos da aula anterior	
	4 Mais efeitos visuais Mesmos objetivos da aula anterior	
	5 Brincando com Chroma key Mesmos objetivos da aula anterior	Chroma key; Pixels; Funções do JES; Variáveis, expressões aritméticas, condicionais, loops, listas e funções.
	6 Gerador de Memes Mesmos objetivos da aula anterior	Efeitos de imagem; Pixels; Funções do JES; Variáveis, expressões aritméticas, sequência, loops, listas e funções.
	7 Cidade na Lama PC1, PC5, PC6 e PC9	Grafos não dirigidos; Algoritmos sobre grafos.

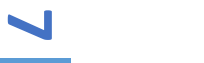
Os estudantes utilizam o ambiente de desenvolvimento JES para manipular propriedades dos pixels das imagens, implementando efeitos como escala de cinza, negativo, *chroma key*, dentre outros.

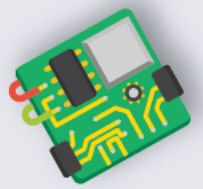
UNIDADE III – Programação em Python

A terceira unidade do programa, composta por 8 aulas, busca aprofundar as habilidades de programação dos estudantes com a linguagem Python.

	AULA	CONTEÚDO
UNIDADE III	1 Fluxograma PC2, PC4, PC8, PC9, PC16, PCP5 e PCP10	Fluxogramas: Terminal, Seta de Fluxo, Processamento; Entrada de Dados, Saída de Dados; Decisão; Conector, Loops.
	2 Apertadinho PC2, PC4, PC8, PC9, PC15, PC16, PCP5, PCP10 e PCP11	Python; Variáveis, atribuição, entrada e saída, loops, condicionais, expressões lógicas e relacionais.
	3 Enquetes PC2, PC4, PC8, PC15, PC16, PCP5, PCP10, PCP11 e PCP15	Python; Variáveis, entrada e saída, loops, condicionais, contadores, expressões aritméticas.
	4 Criptografia de César PC2, PC4, PC8, PC15, PC16, PCP5, PCP10 e PCP11	Criptografia; Python; Variáveis, entrada e saída, loops, condicionais, strings, indexação de strings.
	5 Jogo da Forca PC2, PC4, PC8, PC15, PC16, PCP5, PCP10, PCP11 e PCP15	Python; Variáveis, entrada e saída, loops, condicionais, strings, indexação de strings, expressões lógicas.
	6 A Droga da Obediência PC2, PC4, PC8, PC15, PC16, PCP5, PCP10 e PCP11	Criptografia; Python; Variáveis, entrada e saída, loops, condicionais, strings, manipulação de strings.
	7 Ovos da Páscoa PC2, PC4, PC8, PC15, PC16, PCP5, PCP10, PCP11 e PCP15	Python; Variáveis, entrada e saída, loops, condicionais, expressões lógicas e aritméticas, strings, indexação de strings, Arquivos.
	8 Jogo da Velha PC2, PC4, PC8, PC15, PC16, PCP5, PCP10 e PCP11	Python; Variáveis, entrada e saída, loops, condicionais, strings, listas e sua indexação, funções.

Cada aula implementa projetos de propósito geral, como sistema de enquetes, jogos da forca, jogo da velha, dentre outros.





UNIDADE IV - Arduino

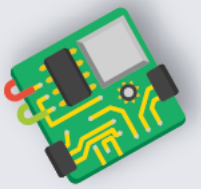
A quarta unidade do programa, composta por 8 aulas, apresenta aos estudantes o universo da eletrônica e robótica com Arduino.

	AULA	CONTEÚDO
UNIDADE IV	1 Construindo um Semáforo PC2, PC7, PCP10, PCP11, PCP13, C7, C8, CDC1, CDC3, CDC4, CDC9 e CDC10	Circuitos elétricos e eletrônicos; Arduino; mBlock; Sequências e loops.
	2 Medindo a Luminosidade em um Ambiente Mesmos objetivos da aula anterior	Arduino; Sensor de luminosidade LDR; Sinais digitais e analógicos; mBlock; Sequências, loops e condicionais.
	3 Criando um Piano com o Arduino Mesmos objetivos da aula anterior	Arduino; Push button; Buzzer; mBlock; Sequências, loops e condicionais.
	4 Criando um Robô Dançarino Mesmos objetivos da aula anterior	Arduino; Servomotor (Micro Servo 9g SG90 TowerPro); mBlock; Sequências, loops e funções.
	5 Criando um Robô Guardiã do Tesouro Mesmos objetivos da aula anterior	Arduino; Sensor ultrassônico (HSR04); mBlock; Sequências, loops e condições.
	6 Como funciona o Computador? PC14, CDC3 e CDC5	Arquitetura de hardware; Arquitetura de software.
	7 Criando seu Primeiro Protótipo com o Arduino Mesmos objetivos da aula 1	Circuitos eletrônicos; Arduino; Componentes; mBlock; Sequências, loops, condições e funções.
	8 Apresentando seu Primeiro Protótipo com o Arduino Mesmos objetivos da aula 1	Circuitos eletrônicos; Arduino; Componentes; mBlock; Sequências, loops, condições e funções; Apresentação de Slides.

Cada aula implementa um projeto diferente, utilizando LEDs, motores, sensores, dentre outros componentes.

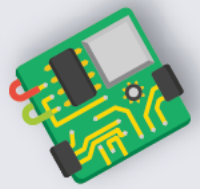
UNIDADE I

Simulações



AULA 1 - Impacto da Tecnologia na Cultura Humana

Nesta aula você irá refletir sobre o impacto das diversas tecnologias na cultura humana.

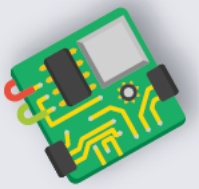


FOLHA DE ATIVIDADES

Refletindo sobre o Impacto da Tecnologia

Responda à seguinte questão: “Qual o impacto da tecnologia na cultura humana?”. Analise quando este impacto é positivo e quando ele é negativo. Além disso, que soluções você poderia propor para aumentar este impacto, em caso positivo, ou diminuir, em caso negativo. Apresente, ao menos, três exemplos dos impactos a seguir.

- **Comportamentos sociais potencializados pela tecnologia:** as novas tecnologias trouxeram muitos avanços para a sociedade. Com isto, a tecnologia modifica o nosso jeito de viver para o bem e para o mal. Assim, alguns comportamentos não desejados podem ser potencializados através do uso incorreto da tecnologia.
- **Uso da informação:** A informação é algo muito importante atualmente. Sabemos que diversas empresas têm como núcleo do seu negócio uso de informações. Hoje todos nós podemos facilmente divulgar informações em sites e redes sociais. Em consequência, também recebemos informações de várias pessoas. Será que sabemos distinguir entre uma informação verdadeira ou falsa?
- **Uso de redes sociais:** As redes digitais diminuem distâncias entre pessoas e possibilitam que pessoas que talvez nunca se conhecessem possam compartilhar experiências de vida ou conhecer a cultura de outros lugares. Mas à medida que as redes aproximam o que está longe, podem distanciar o que está perto?
- **Computação móvel:** Ao longo da história, os computadores foram diminuindo de tamanho, através do estudo de novas tecnologias. A Computação móvel permite que levemos aparelhos a vários lugares como cinema, teatro, shows, viagens de férias e outros, mas até que ponto essa onipresença dos computadores pode ser positiva ou negativa?
- **Privacidade:** a facilidade quanto ao compartilhamento de informações trouxe também a quebra da privacidade. Muitas pessoas postam sobre suas vidas de forma excessiva. Além disso, algum aspecto da sua vida pode estar online, em poucos segundos, sem sua autorização.
- **Segurança:** nossos computadores e celulares estão cheios de dados sobre nós: textos, fotos, vídeo, áudios e informações secretas, como senhas, por exemplo. Devemos tomar cuidado para que esses aparelhos não caiam em mãos erradas.
- **Acesso à tecnologia:** o mundo é muito grande e muito diversos. Você já deve ter ouvido que existem poucos com muitos e muitos com pouco. Este fato é real e se aplica a diversas coisas, como por exemplo a tecnologia. Já pensou que pode existir alguém no mundo que nunca teve acesso a um rádio, a uma televisão ou até mesmo a um celular?



AULA 2 – Minha nova fábrica de chocolate

Nesta aula você irá aprender com atividades práticas e desplugadas conceitos relacionados ao design de interfaces humano-máquina.



FOLHA DE ATIVIDADES

O problema das portas

Analise cada tipo de porta e marque o seu modo de abrir, de acordo com a figura apresentada. Você pode marcar mais de uma opção, se fizer sentido.

Porta Simples



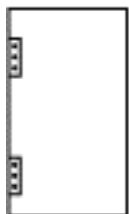
- Empurrar Lado Esquerdo
 Puxar Lado Direito
 Deslizar

Porta com Etiqueta



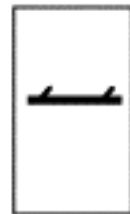
- Empurrar Lado Esquerdo
 Puxar Lado Direito
 Deslizar

Porta com Dobradiça



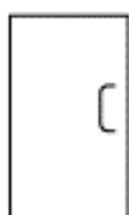
- Empurrar Lado Esquerdo
 Puxar Lado Direito
 Deslizar

Porta com Barra



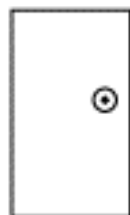
- Empurrar Lado Esquerdo
 Puxar Lado Direito
 Deslizar

Porta com Alavanca



- Empurrar Lado Esquerdo
 Puxar Lado Direito
 Deslizar

Porta com Maçaneta



- Empurrar Lado Esquerdo
 Puxar Lado Direito
 Deslizar

Porta com Painel



- Empurrar Lado Esquerdo
 Puxar Lado Direito
 Deslizar

Porta de Vidro

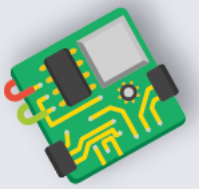


- Empurrar Lado Esquerdo
 Puxar Lado Direito
 Deslizar

Porta Deslizante



- Empurrar Lado Esquerdo
 Puxar Lado Direito
 Deslizar

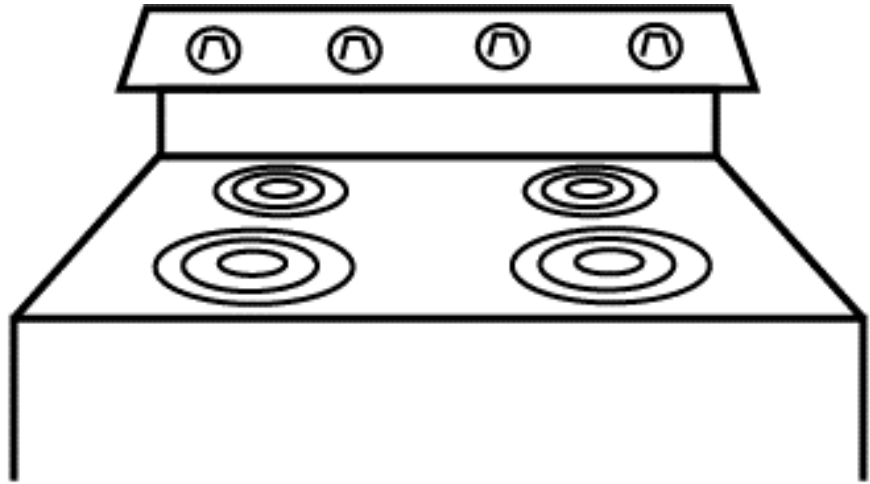


FOLHA DE ATIVIDADES

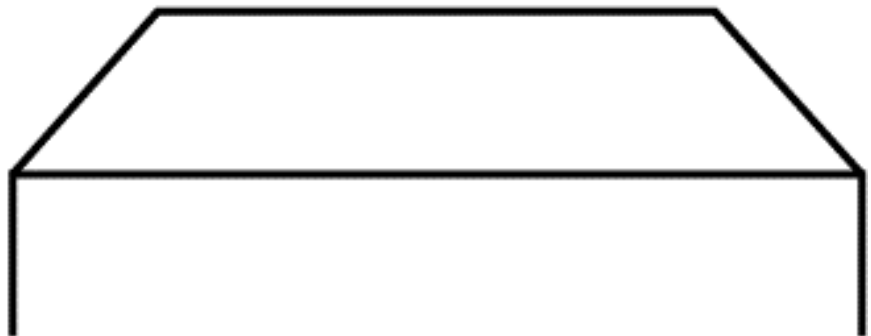
O problema dos fogões

Redesenhe o projeto do fogão para que ele fique mais fácil de utilizar.

VELHO



NOVO





FOLHA DE ATIVIDADES

Projetando um novo sistema de comunicação

A fábrica está cheia de correias transportadoras carregando potes de chocolate. Essas correias transportadoras são controladas manualmente pelos Oompa-Loompas, sob instruções de uma sala de controle central. Na antiga fábrica, isso era feito com um sistema de voz: a voz da pessoa da sala de controle saía de um alto-falante pelos controles da correia transportadora. Mas a fábrica era barulhenta e difícil de ouvir. **Descreva um esquema que use sinais visuais em substituição aos alto-falantes.** *Uma possibilidade é utilizar semáforos. Pergunte ao professor como funcionam os Semáforos no mundo dos Oompa-Loompas.*

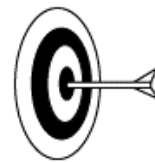
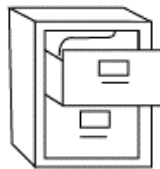
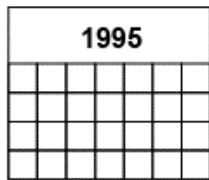


FOLHA DE ATIVIDADES

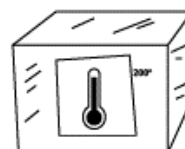
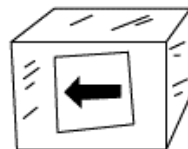
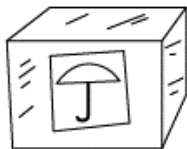
Identificando e criando ícones

Discuta, em grupo, para que servem esses ícones.

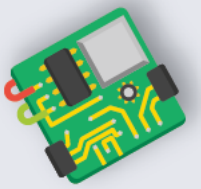
Em um Escritório



Em uma Caixa

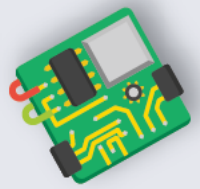


Agora, em grupo, crie ícones para a atividade descrita no cartão entregue pelo professor.



AULA 3 – A matemática do movimento

Nesta aula você deve investigar como a modelagem computacional pode ser utilizada para entender ou descrever fenômenos da natureza, através de simulações. Os projetos desenvolvidos nesta aula tratam da ideia de descrever a queda livre como um movimento com aceleração constante, de modo similar às ideias de Galileu Galilei.



FOLHA DE ATIVIDADES

A programação de computadores também pode ser empregada para a criação de modelos que representem o mundo real. As chamadas simulações utilizam a formalização matemática para construir modelos computacionais que imitam uma situação real. Um exemplo de utilização prática dos modelos computacionais é quando criamos jogos onde objetos caem, ou quando criamos animações que representem experimentos científicos.

Exercício 1 – A queda

Crie uma animação no Scratch onde uma bola deve “cair” no chão da tela. Ao criar sua animação, utilize um algoritmo que torne a queda da bola mais parecida com o modo como os objetos caem no mundo real.

Exercício 2 – Questões sobre a queda

Após finalizar a animação do Exercício 1, responda às questões a seguir:

1 – Ao jogar uma bola de uma determinada altura, ela mantém a mesma velocidade durante toda a queda?

2 – Os objetos em queda sempre aceleram? Em caso afirmativo, como?

3 – Como implementar uma simulação que represente a queda dos objetos? Descreva brevemente o algoritmo empregado.

4 – Na sua concepção, o que significa o termo **velocidade**?

5 – Na sua concepção, o que significa o termo **aceleração**?



Exercício 3 – Um modelo melhor para a queda

Na vida real, a queda é um movimento com aceleração constante. Esta ideia é uma das grandes descobertas do físico, matemático, astrônomo e filósofo Galileu Galilei.

Quando jogamos um objeto em repouso a partir de uma determinada altura, no primeiro instante, a velocidade é zero, mas, a cada instante que passa, há um aumento na velocidade. Do modo como os computadores representam este fenômeno, a cada repetição do programa, que corresponde a um "incremento de tempo", geramos um incremento constante na quantidade de pixels em que o objeto é deslocado na tela, o que corresponde, na prática, à velocidade. Assim, a **velocidade** é o incremento para a posição do objeto a cada instante e a **aceleração** é o incremento para a velocidade a cada instante. A queda respeita o que chamamos de movimento uniformemente variado, onde a velocidade aumenta e a aceleração é constante.

Recrie sua animação de bola caindo, empregando o algoritmo a seguir:

1. A velocidade inicial é zero.
2. A aceleração é um valor que pode ser alterado pelo usuário da simulação (utilize um controle deslizante para isso).
3. Repita os seguintes comandos até alcançar o chão:
 - a. Ande o deslocamento especificado pela velocidade.
 - b. A velocidade no instante atual = a velocidade no instante anterior + aceleração

Este modelo não é exatamente o modelo que representa a vida real, mas a queda seguindo esta representação é muito mais convincente!

Exercício 4 – Queda livre

O conceito de queda livre é o movimento resultante unicamente da aceleração provocada pela gravidade, que é uma das forças fundamentais da natureza. A gravidade consiste em uma atração gravitacional da Terra, que dá peso aos objetos. Por isso é que as coisas caem ao chão quando são soltas!

Na prática, quando jogamos algo do alto de um prédio, além da força gravitacional, há também a resistência do ar interferindo na queda. Para efeitos de aprendizagem, podemos considerar que a resistência do ar é desprezível (ou fingir que estamos jogando um objeto do alto de um prédio e estamos no vácuo total)! Considerando apenas a gravitação atuando sobre o objeto caindo, a fórmula que descreve a queda livre é:

$$y = y_0 + v_0 t - \frac{1}{2} g t^2$$

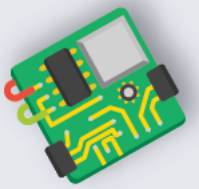
y é a posição do objeto no instante t ;

y_0 é a posição inicial do objeto;

v_0 é a velocidade inicial do objeto, se o objeto estava inicialmente em repouso então $v_0 = 0$;

g é a aceleração da gravidade, que, para os casos próximos à superfície do planeta Terra, pode ser aproximada para $g = 9,8 \text{ m/s}^2$.

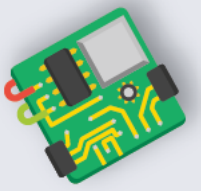
Adapte este modelo matemático da queda livre de modo que ele possa ser representando em uma animação no Scratch, que explique ao usuário os principais conceitos da queda livre.



DIÁRIO DE BORDO

Na aula de hoje

Ruled writing area for the journal entry, consisting of 23 horizontal lines.



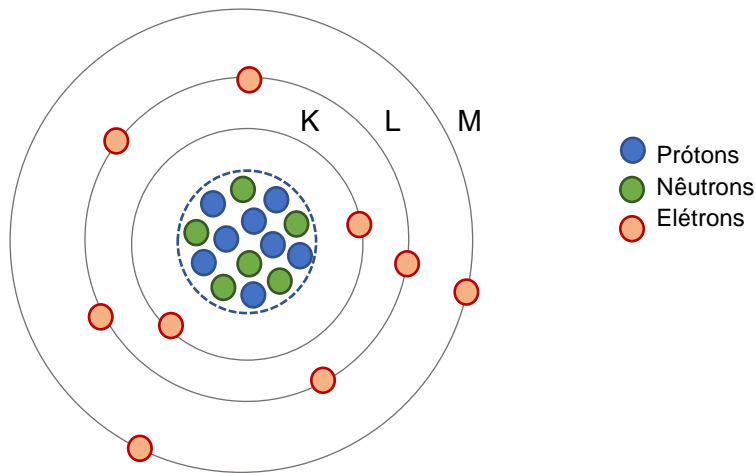
AULA 4 – Modelo Atômico de Bohr

Nesta aula você deve implementar uma simulação que represente o modelo atômico de Bohr para um elemento químico de sua escolha.



TÓPICOS RELEVANTES

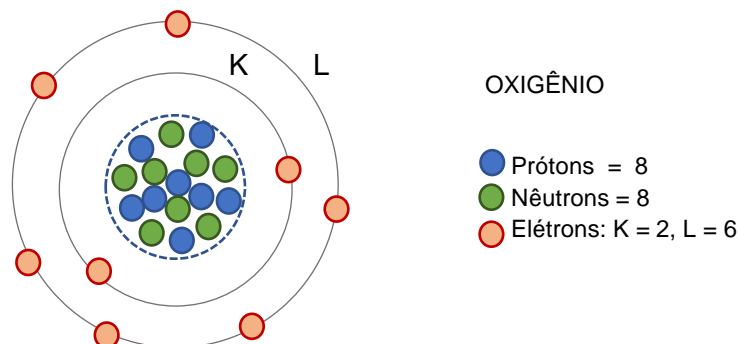
Átomo é uma unidade básica de matéria, que é composto por **prótons**, **nêutrons** e **elétrons**. Um próton é uma partícula de carga elétrica positiva, enquanto um elétron é uma partícula de carga elétrica negativa. O nêutron não possui carga elétrica. No modelo atômico proposto pelo físico dinamarquês Niels Bohr, o átomo possui um núcleo pequeno e carregado positivamente cercado por elétrons em órbita circular, a exemplo do átomo da figura abaixo.



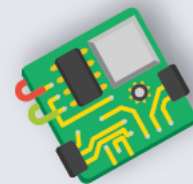
Algumas características do **modelo atômico de Bohr**:

- Existem sete camadas eletrônicas ao redor do núcleo e os elétrons estão distribuídos entre elas;
- As camadas são representadas pelas letras maiúsculas: K, L, M, N, O, P e Q. Existem sete níveis de energia, de modo que, à medida que as camadas se afastam do núcleo, aumenta a energia dos elétrons nelas localizados;
- Os elétrons seguem um movimento circular ao redor do núcleo do átomo.

Cada átomo possui uma quantidade específica de prótons, nêutrons, além dos elétrons de cada camada. À quantidade de camadas e de elétrons em cada camada do átomo, damos o nome de distribuição energética. Por exemplo: o átomo de oxigênio, que é um elemento que ocupa a posição 8 da tabela periódica, possui 8 prótons, 8 nêutrons e 8 elétrons.



O átomo de oxigênio possui duas camadas energéticas: a camada K possui 2 elétrons e a camada L possui 6 elétrons.



FOLHA DE ATIVIDADES

Nesta unidade, estamos aprendendo como a programação de computadores pode ser empregada para a criação de simulações. O átomo é um tema importante para o estudo da ciência, especialmente da química e, por isso, simulações de modelos atômicos são importantes ferramentas de estudos.

Exercício 1 – Escolha seu elemento químico

Escolha um dos elementos químicos da tabela periódica e responda às perguntas a seguir:

1 – Qual o elemento químico escolhido e seu número na tabela periódica?

2 – Quantos nêutrons o átomo do elemento escolhido possui em seu núcleo?

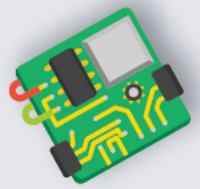
3 – Quantos prótons o átomo do elemento escolhido possui em seu núcleo?

3 – Quantos elétrons o átomo do elemento escolhido possui em seu núcleo?

4 – Quantas camadas energéticas o átomo do elemento escolhido possui?

5 – Preencha a quantidade de elétrons em cada camada energética do átomo do elemento escolhido, deixando em branco as camadas que não existirem no átomo:

Camada energética	Quantidade de elétrons
K	
L	
M	
N	
O	
P	
Q	



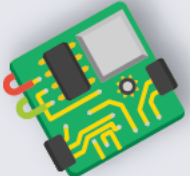
Exercício 2 – Simule seu elemento

Crie uma animação no Scratch que represente o modelo atômico do elemento químico escolhido por você no Exercício 1.

Dica: As equações paramétricas para o círculo podem ser descritas como:

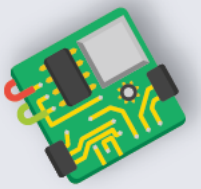
$$x = r \times \cos((a \times t) + d) \qquad y = r \times \sin((a \times t) + d)$$

Onde **r** é o raio do círculo, **a** é a aceleração da partícula, **t** é uma variável que é incrementada com o tempo e **d** é o deslocamento em relação à origem (0°) do movimento circular.



DIÁRIO DE BORDO

Na aula de hoje

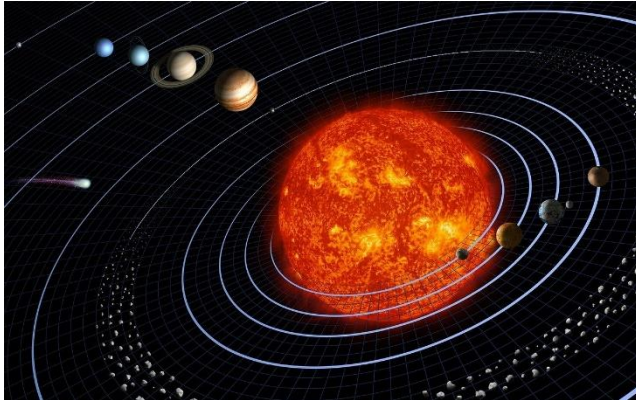


AULA 5 – Sistema Solar

Nesta aula você deve implementar uma simulação que represente o movimento dos planetas no Sistema Solar.



TÓPICOS RELEVANTES



Terra.

O **Sistema Solar** é formado pelos planetas, satélites naturais e corpos celestes que estão sob domínio gravitacional do Sol.

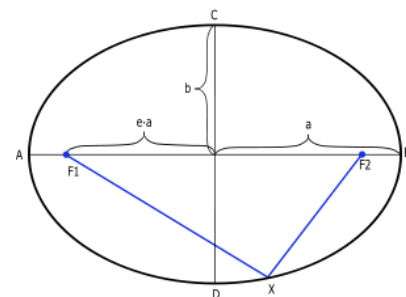
Assim, o Sol é a estrela central do Sistema Solar, além de ser muito grande em relação a todos os outros elementos do sistema. Em nível de comparação, o volume do Sol, ou seja, a quantidade de espaço que ele ocupa, é 1.300.000 vezes maior do que a

O Sistema Solar possui 8 planetas conhecidos classificados de acordo com a distância em relação ao Sol: **Mercúrio, Vênus, Terra, Marte, Júpiter, Saturno, Urano e Netuno**. Os quatro planetas mais próximos do Sol são chamados planetas telúricos e têm como características comuns a presença de crostas sólidas e núcleos com elevada porcentagem de ferro. Os quatro planetas mais afastados são chamados planetas gigantes, pois possuem dimensões consideravelmente maiores que os planetas telúricos.

Além dos planetas conhecidos, há outros 5 planetas classificados como planetas anões: **Ceres, Plutão, Haumea, Makemake e Éris**.

Todos os planetas e demais corpos do Sistema Solar descrevem uma **órbita elíptica** ao redor do Sol. Uma elipse, por sua vez, é uma figura geométrica como ilustrada no exemplo abaixo. Este movimento orbital é também chamado de movimento de translação. Além disso, o tempo que cada planeta leva para estabelecer uma volta completa em torno do Sol varia. Na Terra, é de um ano. Já em Júpiter, dura em torno de 12 anos terrestres.

Uma elipse no plano cartesiano possui dois focos F_1 e F_2 . O segmento de reta que passa por esses dois focos é chamado de eixo maior. A elipse também possui um eixo menor, que é perpendicular ao eixo maior. Quanto mais distantes estão os focos de uma elipse, maior é a sua **excentricidade** e , e quando os focos estão mais próximos, a elipse se parece mais com um círculo.



Fonte: Wikipédia

Para a maior parte dos planetas, a elipse orbital é pouco excêntrica e a trajetória é praticamente circular. Nas órbitas do Sistema Solar, o Sol se localiza em um dos focos da elipse, fazendo com que haja um ponto onde ocorre a máxima aproximação dos corpos em relação ao Sol e um ponto oposto, onde ocorre a máxima distância em relação ao Sol.

Os componentes do Sistema Solar giram ao redor de seu próprio eixo imaginário, estabelecendo um movimento de rotação, que pode ser no sentido horário ou anti-horário. Vênus e Urano exercem a rotação no sentido anti-horário, enquanto todos os outros planetas o fazem no sentido horário.



FOLHA DE ATIVIDADES

Nesta unidade, estamos aprendendo como a programação de computadores pode ser empregada para a criação de simulações. O Sistema Solar é um tema importante para o estudo da ciência e da astronomia, por isso, simulações do comportamento dos planetas são importantes ferramentas de estudos.

Exercício 1 – Sistema Solar

Implemente no Scratch uma simulação para o Sistema Solar. Sua simulação deve representar os 8 planetas conhecidos Mercúrio, Vênus, Terra, Marte, Júpiter, Saturno, Urano e Netuno. Para que sua simulação fique muito legal, seguem algumas dicas:

1ª Dica: Pesquise as proporções dos planetas. No Sistema Solar o Sol possui um volume muito maior do que todos os outros corpos e os planetas também possuem diferentes tamanhos. Faça uma representação estabelecendo os tamanhos dos atores de sua animação de modo que eles se pareçam com a proporção original.

2ª Dica: A órbita dos planetas é elíptica e a posição do Sol é um dos focos destas elipses. A fórmula paramétrica utilizada na aula passada para descrever uma trajetória circular pode ser adaptada para que o movimento do ator seja elíptico. Uma elipse cujo eixo maior está no eixo x pode ser representada pelas seguintes equações:

Equações para o círculo:

$$x = r \times \cos(\text{velocidade} \times t)$$
$$y = r \times \text{sen}(\text{velocidade} \times t)$$

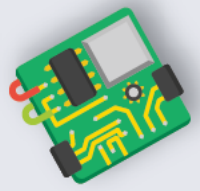
Equações para a elipse:

$$x = a \times \cos(\text{velocidade} \times t)$$
$$y = b \times \text{sen}(\text{velocidade} \times t)$$

Onde **r** é o raio do círculo, **a** é a metade do eixo maior (horizontal) da elipse e **b** é a metade do eixo menor (vertical) da elipse. Utilize os princípios desta fórmula para que os atores de sua simulação tenham uma trajetória elíptica.

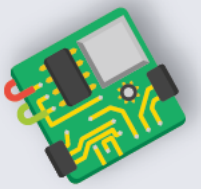
3ª Dica: Cada planeta leva um determinado tempo para completar sua órbita! Pesquise e adicione diferentes valores de aceleração ao movimento de cada planeta.

4ª Dica: Faça com que sua simulação também represente o movimento de rotação, que é quando os planetas giram em torno de si mesmos. Lembre-se que cada planeta tem um tempo de rotação específico. Faça sua pesquisa e descubra proporções de tempos de rotação para os planetas.



DIÁRIO DE BORDO

Na aula de hoje



AULA 6 - Fazendo seu Projeto de Ciências (parte 1)

Nesta aula você irá exercitar suas habilidades sobre programação no Scratch para idealizar projetos de ciência.

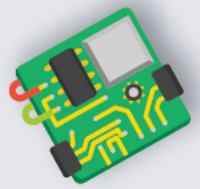


FOLHA DE ATIVIDADES

Idealizando meu projeto de ciências

1 - Qual o fenômeno da natureza escolhido para a simulação?

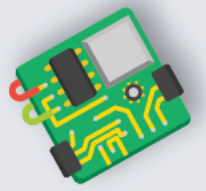
2 - Descreva o seu projeto de forma detalhada. Além disso, você pode utilizar desenhos para exemplificar. Se você usar fórmulas matemáticas, coloque-as aqui também.



3 - Marque com um "X" os conceitos que utilizará:

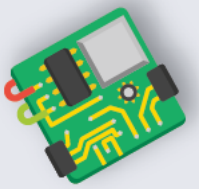
- Variáveis
- Entrada
- Saída
- Condicionais
- Loops
- Contadores
- Expressões aritméticas
- Funções matemáticas
- Ângulos

4 - Descreva aqui outros conceitos específicos de ciências que utilizará:



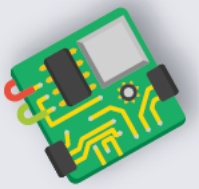
DIÁRIO DE BORDO

Na aula de hoje



AULA 7 - Fazendo seu Projeto de Ciências (parte 2)

Nesta aula você irá exercitar suas habilidades de programação no Scratch para implementar projetos de ciências.



FOLHA DE ATIVIDADES

Desenvolvendo meu Projeto de Ciências

1 – Com base na Folha de Atividades da aula anterior, implemente no Scratch o seu projeto de Ciências.

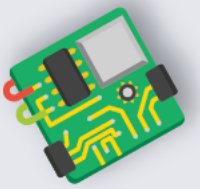
Passo 1: Selecione as imagens que utilizará.

Passo 2: Desenhe as imagens que não encontrou.

Passo 3: Selecione áudios que utilizará;

Passo 4: Coloque as imagens no seu projeto Scratch.

Passo 5: Implemente sua simulação, programando cada ator por vez.

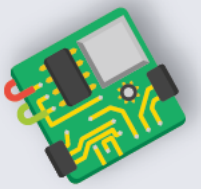


DIÁRIO DE BORDO

Na aula de hoje

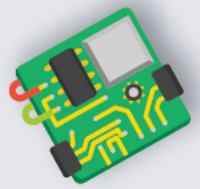
UNIDADE II

Manipulação de Imagens



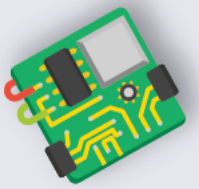
AULA 1 – Pixel a Pixel

Nesta aula você irá relembrar conceitos sobre o padrão de cores RGB, pixels, imagens digitais e manipularão os pixels através do JES de modo introdutório.



AULA 2 – Fazendo Seus Próprios Filtros

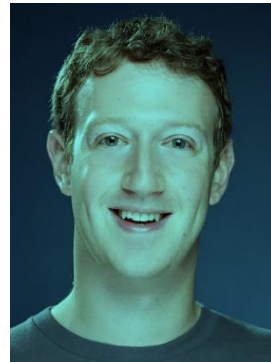
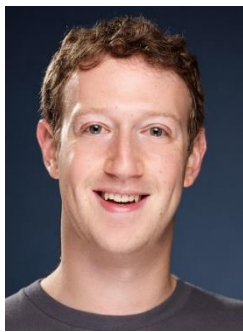
Nesta aula você irá criar efeitos visuais que modificam os valores dos pixels originais, como matiz, utilizando loops e operações aritméticas sobre os pixels com o uso da linguagem Python.



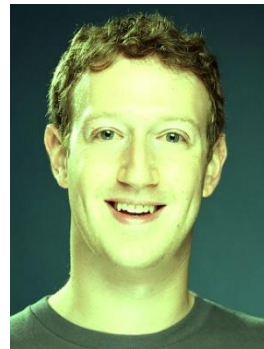
FOLHA DE ATIVIDADES - Criando efeitos de Matiz

Exercite suas habilidades com pixels para a criação de efeitos variados.

- ✓ **Matiz Vermelha.** Reduza do valor vermelho (*red*) pela metade, em todos os pixels da imagem, e obtenha o seguinte efeito.

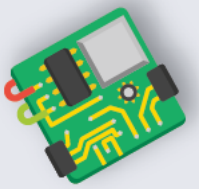


- ✓ **Matriz Verde.** Adicione 50% ao valor verde (*green*), em todos os pixels da imagem, e obtenha o seguinte efeito.

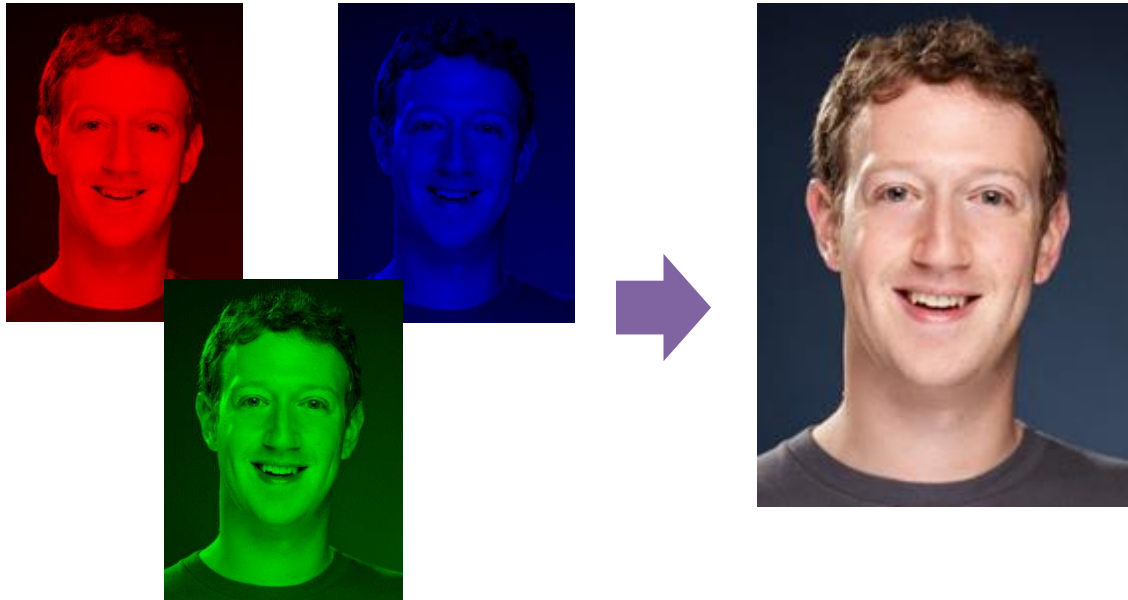


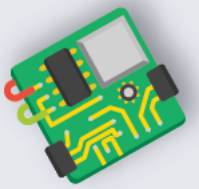
- ✓ **Matiz Azul.** Reduza 250 do valor azul (*blue*), em todos os pixels da imagem, e obtenha o seguinte efeito.





- ✓ **Desafio.** Você já sabe que as imagens RGB são formadas por 3 valores de cores básicas RGB: *Red*, *Green* e *Blue*. Experimente criar 3 imagens, cada uma com apenas uma dessas cores. Ao final, some os valores dos pixels das 3 imagens e obtenha a imagem original. Preparado?!





AULA 3 - Mais Efeitos Utilizando Matemática

Nesta aula você irá aplicar conhecimentos matemáticos para criar os efeitos de negativo, escala de cinza e espelhamento. Você também irá desenvolver programas, utilizando funções, para criar os efeitos.



FOLHA DE ATIVIDADES - Criando mais efeitos com operações matemáticas

Aqui, você irá exercitar suas habilidades matemáticas e lógicas para a criação de efeitos mais avançados com Python. Não esqueça de utilizar funções.

- ✓ **Crie um efeito de Negativo.** Um efeito de negativo é criado através da subtração do valor original do pixel do valor máximo que uma cor do pixel pode receber: $255 - \text{valor da cor}$.



- ✓ **Crie o efeito Escala de Cinza.** Um efeito escala de Cinza é criado utilizando a média das três cores RGB de cada pixel. Para cada pixel, é preciso somar os valores de cada cor e depois dividi-las por 3, calculando uma média. Esse novo valor deve ser colocado em cada cor RGB do mesmo pixel, ou seja, $(\text{red} + \text{green} + \text{blue}) / 3$

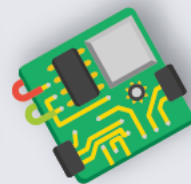




- ✓ **Crie o efeito de Espelhamento.** Um efeito de espelhamento é realizado copiando os pixels da primeira metade da imagem para a segunda metade, percorrendo-a na direção contrária. O loop padrão deve ser modificado para percorrer apenas metade do comprimento da imagem ($width / 2$). A cor de cada pixel percorrido é trocada pela cor do pixel na posição oposta. Por exemplo, em uma imagem de comprimento 100, o pixel 0,0 é oposto ao 99,0; o pixel 1,0 é oposto ao pixel 98,0. O pixel 20,10 é oposto ao pixel 79,10 e assim sucessivamente.

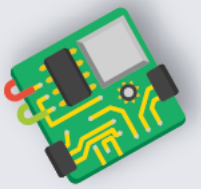


- ✓ **Novo Efeito.** Escolha uma personagem que você gosta. Misture dois ou mais efeitos ou realize modificações sobre os efeitos vistos para gerar novos efeitos. Escreva a seguir o que você fez para gerar este novo efeito e qual é o nome que você deu a ele.



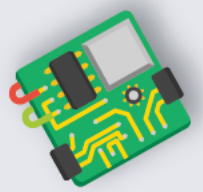
DIÁRIO DE BORDO

Na aula de hoje



AULA 4 - Mais efeitos visuais

Nesta aula você irá manipular imagens através de condicionais, criando recortes de imagens, efeito binário e adicionando bordas em imagens.



FOLHA DE ATIVIDADES

Criando efeitos com condicionais e funções com parâmetros

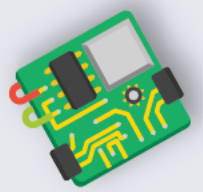
Chegou a hora de exercitar suas habilidades com Python e manipulação de imagens. Para isso, faça as seguintes atividades.

- ✓ **Crie um efeito binário.** Um efeito Binário é um efeito que transforma uma imagem com muitas cores em uma imagem com apenas duas cores. Para isso, você pode utilizar a comparação entre um limiar (valor fixo definido de intensidade da cor) e o valor da soma das cores RGB para decidir se pinta de uma cor ou de outra ($somaDasCores < limiar$). Crie uma função para fazer o efeito binário que receba como parâmetros as duas cores do efeito.



- ✓ **Crie um efeito de borda.** O efeito de borda já é um velho conhecido seu da primeira aula sobre imagens. Faça uma função para criar uma borda em uma imagem que utilize dois parâmetros: um para estipular a espessura da borda e outra para a cor da borda.





- ✓ **Crie o efeito de recorte.** Um efeito de recorte é um efeito realizado com mais de uma imagem. A ideia é copiar uma parte (retângulo) de uma imagem e adicioná-la em outra, sobrepondo a parte copiada sobre a segunda imagem. Veja este exemplo:

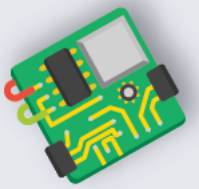


Faça uma função chamada `recorte` que receba dois parâmetros: a primeira imagem, que será copiada, e a segunda imagem, que receberá o recorte. Perceba que funções podem receber imagens como parâmetros.

Desafio 1. No efeito `recorte`, foram utilizadas duas imagens como parâmetros da função. Você consegue pensar em dois parâmetros adicionais para escolher o trecho da primeira imagem que será copiado? Que tal criar um código com estes novos parâmetros?

Desafio 2. No efeito de recorte, é possível deixar a imagem do fundo em escala de cinza e mover a foto recortada para o centro. Que tal tentar esta modificação?





AULA 5 - Brincando com *Chroma key*

Nesta aula você irá criar o efeito Chroma key em uma imagem, sobrepondo uma imagem sobre outra imagem de fundo e anulando uma cor padrão da imagem original, por exemplo, verde.

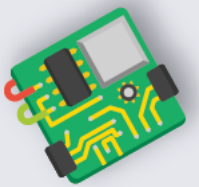


FOLHA DE ATIVIDADES

Implementando o efeito *Chroma key*

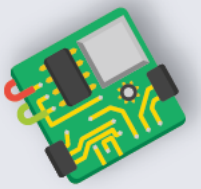
Vamos exercitar suas habilidades matemáticas e lógicas para a criação do efeito *Chroma key*.

- ✓ **Primeira parte.** Remova o fundo da imagem que você trouxe para a atividade. Lembre-se de ajustar o valor da cor (verde ou azul) para eliminar todo o fundo.
- ✓ **Segunda parte.** Agora que você já sabe remover uma cor padrão do fundo de uma imagem, experimente colar os pixels de outra imagem para compor o fundo, criando o efeito *Chroma key*. Você pode pensar em algum lugar especial para você.
- ✓ **Terceira parte:** Agora que você já está dominando a arte do *Chroma key*, que tal compor uma imagem com três imagens diferentes: *imagem 1 com fundo azul/verde, imagem 2 com fundo azul/verde e um fundo.*
- ✓ **Desafio.** É possível utilizar um ou mais efeitos combinados com o *Chroma key*? Exercite sua criatividade e crie um novo efeito.



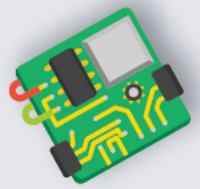
DIÁRIO DE BORDO

Na aula de hoje



AULA 6 – Gerador de Memes

Nesta aula você utilizará todos os conceitos aprendidos para criar um gerador de memes.



FOLHA DE ATIVIDADES - Gerador de Memes

1 - Escreva a sua proposta para o gerador de Memes.

Ex.: Vou adicionar o nome "MIGA, SUA LOUCA?!" na imagem e deixá-la em escala de cinza.

2 - Marque com um "X" os conceitos que utilizará:

loop sobre uma lista de pixels

loop dentro de loop

condicionais

funções

parâmetros em funções

operações aritméticas

operadores relacionais

outros: _____

3 - Marque com um "X" os efeitos que utilizará:

adicionar nomes

adicionar borda

matiz

negativo

escala de cinza

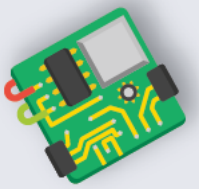
binário

recorte

Chroma key

outros: _____

4 - Crie o código para seu gerador de memes utilizando todos os conceitos e efeitos marcados.



AULA 2 – AULA 7 – Cidade na Lama

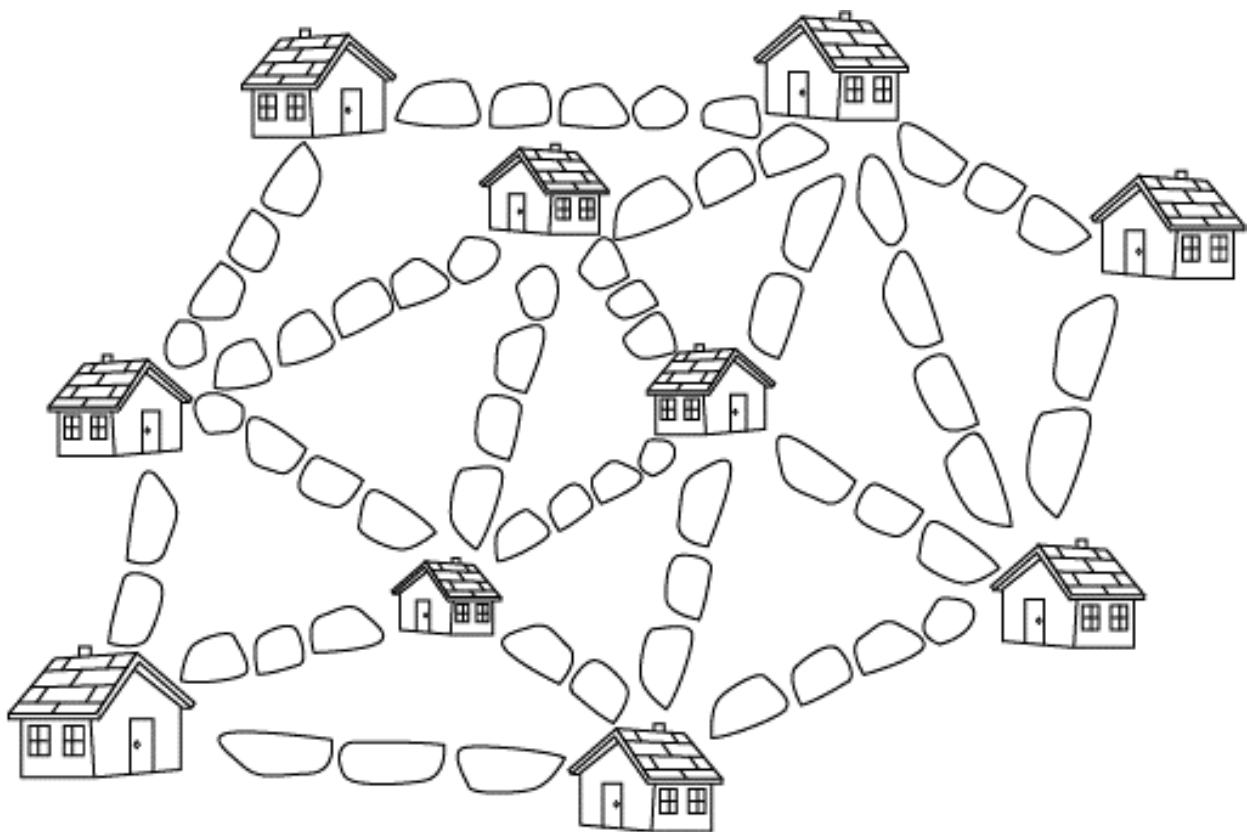
Nesta aula você irá aprender como o conceito de grafo pode ser utilizado em atividades do dia a dia através de uma atividade desplugada.



FOLHA DE ATIVIDADES – Cidade na Lama

Uma pequena cidade do Brasil sofre todos os anos com as fortes chuvas. Nesta cidade, sempre que chove, as pessoas têm dificuldade de sair de casa pois as ruas ficam cheias de lama. Por conta deste problema, o prefeito da cidade resolveu pavimentar as ruas. Mas, contando com poucos recursos, pediu para que os funcionários encontrassem uma solução mais barata para pavimentar as ruas. A solução deve permitir que todos os moradores possam sair de suas casas para qualquer outra casa através das ruas pavimentadas.

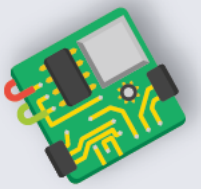
Esta imagem representa a cidade. O número de blocos de pavimentação entre cada casa representa o custo de pavimentação. Ou seja, se há dois blocos pavimentados entre as casas A e B, o custo desta pavimentação é 2. **Encontre o melhor percurso que ligue todas as casas, mas utilize a menor quantidade possível de blocos.**



Qual estratégia você usou para resolver o problema?

UNIDADE III

Python



AULA 1 – Fluxograma

Nesta aula você irá aprender sobre fluxogramas através de atividades de organização e criação de fluxogramas.



TÓPICOS IMPORTANTES

O fluxograma é mais uma forma de representar os algoritmos. No fluxograma, existem vários símbolos que funcionam como comando no Scratch ou como palavras reservadas do Python. Com exceção da Decisão e do Conector, esses símbolos se conectam a penas um outro símbolo. A Seta indica a ordem de execução de cada bloco, iniciando sempre o Início em direção ao final. Loops são representados pela ligação de um bloco com outro anterior, através da Seta de Fluxo.

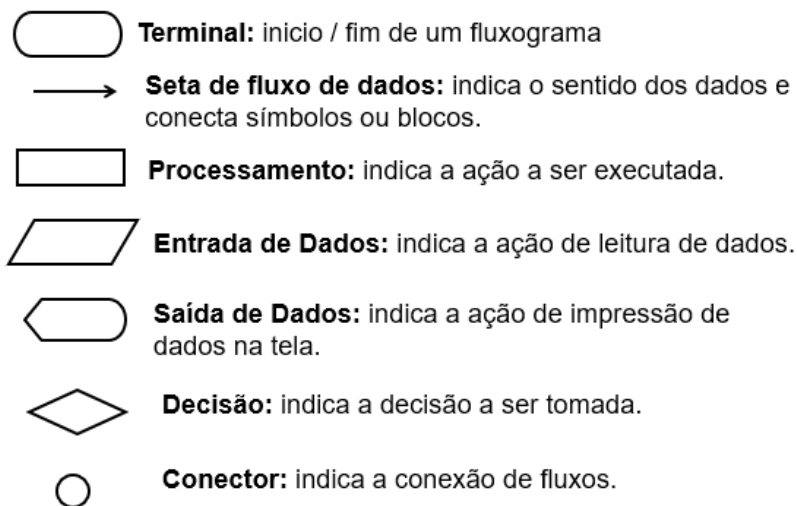
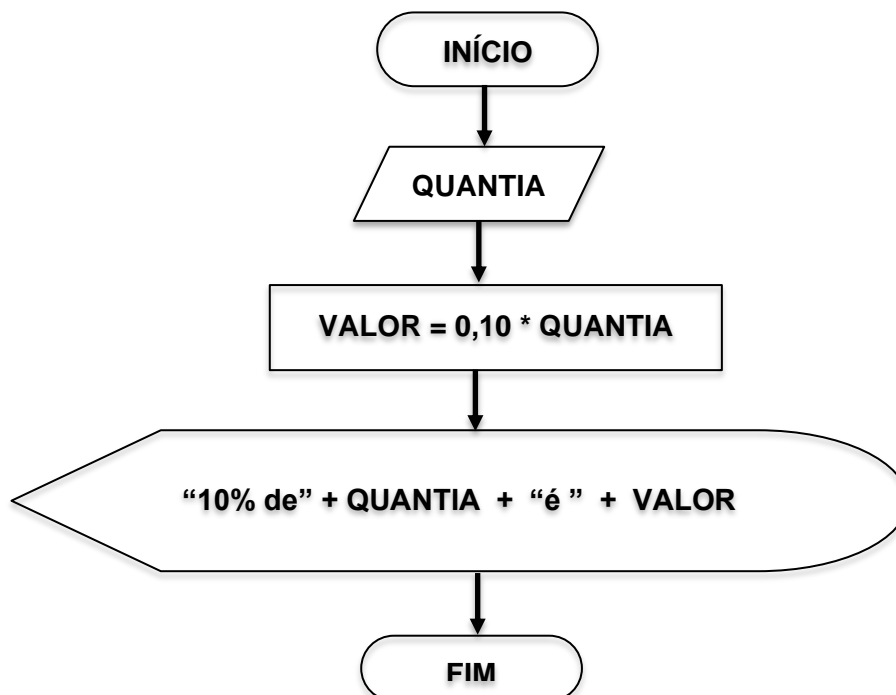
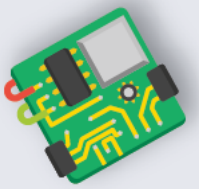


Figura 1 - Símbolos do Fluxograma

Vamos ver um exemplo que calcula e exibe 10% de uma quantia informada pelo usuário (N1). Caso o usuário informe o valor 100, a mensagem será “10% de 100 é 10”.



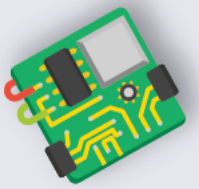


FOLHA DE ATIVIDADES - parte I

Agora que você sabe tudo sobre fluxogramas, exercite um pouco com esta atividade. Utilize os símbolos disponíveis na folha mais à frente. Recorte-os, cole-os e desenhe setas para completar os fluxogramas, indicando o fluxo de execução.

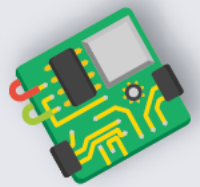
- a) Leia dois números, some-os e exiba o resultado.

- b) Leia dois números, calcule e exiba a média deles.



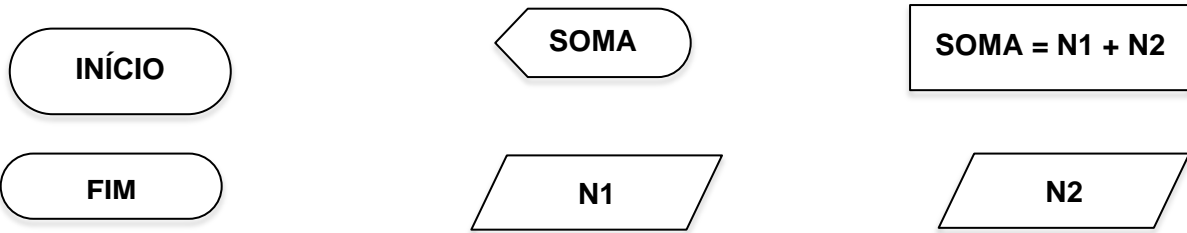
- c) Leia duas notas, calcule a média delas e verifique se a média é maior ou igual 5 (aprovação) ou menor que 5 (reprovação).

- d) Solicite um nome e exiba-o após a frase "Seu nome é ". O fluxograma deve continuar solicitando e mostrando os nomes até que o nome digitado seja a palavra "Sair".

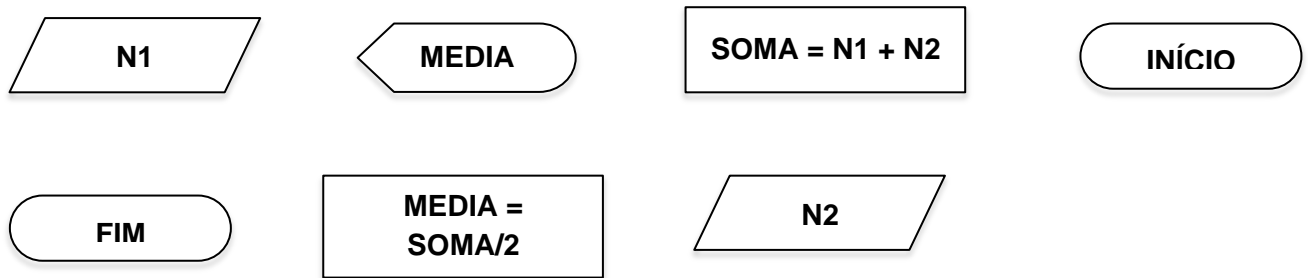


Recorte os símbolos e os organize no espaço reservado.

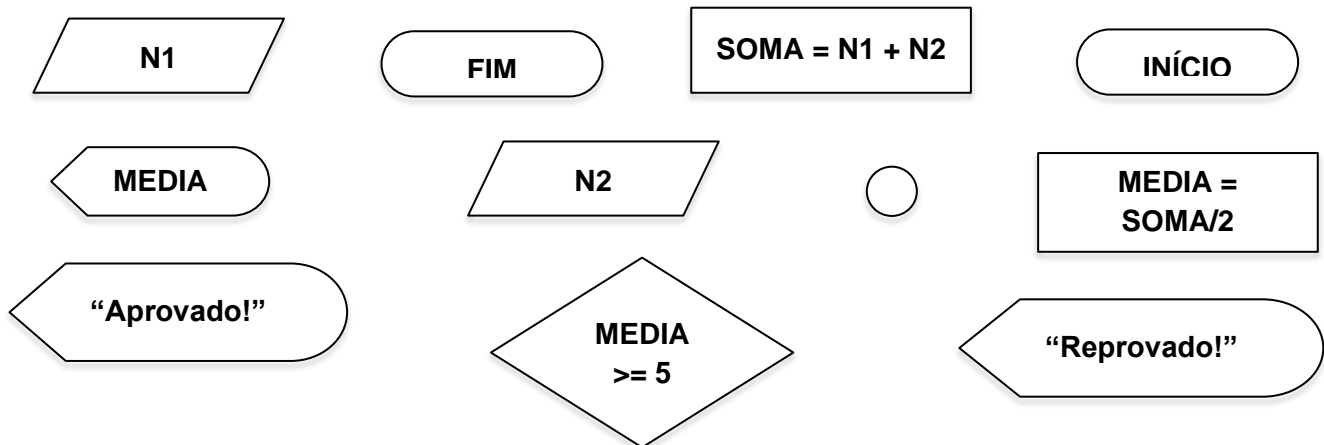
a)



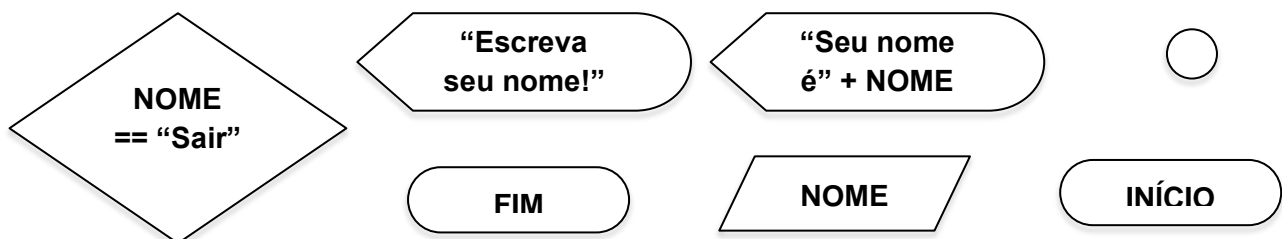
b)



c)



d)





FOLHA DE ATIVIDADES - parte II

Individualmente, desenhe os seguintes fluxogramas ou utilize um programa de computador.

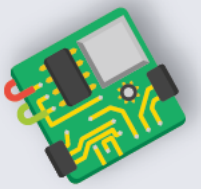
- a) Crie um algoritmo que calcule a área de um quadrado, dado o lado do quadrado.

- b) Crie um algoritmo que calcule a área de um triângulo, dada a base e a altura.



- c) Crie um algoritmo que receba os valores dos três ângulos internos de um triângulo e diga se ele é equilátero.

- d) Crie um algoritmo que solicite o nome de uma figura. Em seguida, exiba este nome cem vezes na tela.



AULA 2 – Apertadinho

Nesta aula você implementará o jogo Apertadinho utilizando conceitos como variáveis, entrada e saída, condicionais e loops.



FOLHA DE ATIVIDADES

O jogo Apertadinho consiste em ter um número sorteado entre um número mínimo e um número máximo (por exemplo, entre 0 e 100). A cada rodada, o jogador dá um palpite no intuito de acertar qual foi o número sorteado. A cada rodada, também, é reduzido o limite dos números mínimo e máximo para o jogador, “apertando” cada vez mais as possibilidades. Os critérios para esta redução são:

- a) Se o palpite é maior que o mínimo e menor que o número sorteado, ele será o novo mínimo.
- b) Se o palpite é menor que o máximo e maior que o número sorteado, ele será o novo máximo.

Vamos a um exemplo: O número sorteado é 50, o mínimo é 0 e o máximo é 100. O jogador escolhe o valor 40. Assim, o 40 será o novo mínimo e o máximo continua sendo 100. Na próxima rodada, o jogador escolhe o valor 60. Agora o novo máximo passará a ser 60. Deste modo, o jogador só poderá escolher números entre 40 e 60 na próxima jogada.

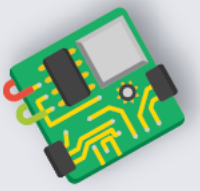
No jogo, o jogador ganha se acertar o número, mas perde se os números mínimo e máximo forem separados por apenas um número, ou seja, separados apenas pelo número sorteado.

Vamos a mais um exemplo: Os limites são 40 e 60. O jogador escolhe o número 51! Como 51 é maior que 50 e menor que 60 (máximo), os novos limites são 40 e 51. Na rodada seguinte o jogador escolhe o número 49. Logo, o limite mínimo será agora 49 e o máximo permanece 51. A única alternativa possível é o número 50, logo o jogador não conseguiu descobrir o valor.

Agora, vamos lá, com base em seus conhecimentos de programação e nas dicas dadas sobre o jogo Apertadinho:

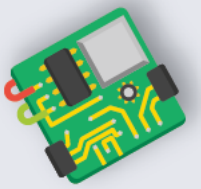
1 - Crie um fluxograma que represente as regras do jogo apertadinho.

2 - Crie um programa que implemente o jogo apertadinho utilizando a linguagem Python.



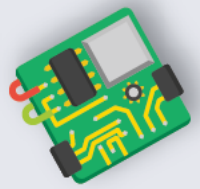
DIÁRIO DE BORDO

Na aula de hoje



AULA 3 – Enquetes

Nesta aula você criará um programa de enquetes, reforçando a aprendizagem dos conceitos de variáveis, entrada e saída, condicionais, loops, e adicionando o conceito de contadores.



FOLHA DE ATIVIDADES

Uma enquete é uma pesquisa de opinião, ou seja, um levantamento estatístico da opinião pública a respeito de algum tema. Muitas pesquisas de opinião trabalham com dados quantitativos e executam estatísticas sobre estes dados. Um exemplo é uma enquete com o objetivo de sondar os estudantes sobre qual meio de transporte eles utilizam para chegar à escola. Os participantes que respondem a essa enquete geram uma contagem para cada meio de transporte. Cálculos como a porcentagem de uso de cada meio de transporte podem ser realizados sobre estes dados, oferecendo uma visão ampla sobre a realidade social dos estudantes.

Exercício 1 - Quem é mais popular?

Vamos descobrir quem ou o que é mais popular na sua turma. Selecione abaixo um tipo de enquete:

- () Qual sua série de tv favorita?
- () Qual seu filme favorito?
- () Qual seu gênero de filmes favorito?
- () Qual seu gênero musical favorito?
- () Qual seu(sua) cantor(a) favorito(a)?
- () Qual sua banda favorita?
- () Qual seu(sua) youtuber favorito(a)?

Descreva as opções da sua enquete:

Opção 1: _____

Opção 2: _____

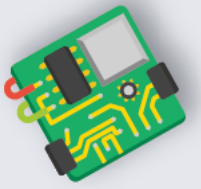
Opção 3: _____

Opção 4: _____

Opção 5: _____

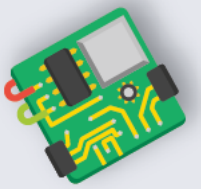
Opção 6: Nenhuma das opções anteriores.

Crie um programa que implemente sua enquete utilizando a linguagem Python.



DIÁRIO DE BORDO

Na aula de hoje



AULA 4 – Criptografia de César

Nesta aula você criará um programa que implementa o método de criptografia chamado cifra de César.



TÓPICOS RELEVANTES

A linguagem de programação Python é uma das linguagens mais fáceis de aprender quando se é novato na programação de computadores. Mesmo assim, é preciso prestar atenção em alguns aspectos para que seus programas funcionem corretamente.

Quando criamos uma variável em um programa em Python, esta variável pode ser de vários tipos: número inteiro, número real, texto, dentre outros tipos. Lembre-se que a variável é uma caixinha que guarda um valor. Em Python, uma variável pode armazenar diferentes tipos de valores ao longo da execução do programa. Mas, se em um determinado momento a caixinha está guardando um número, você não pode executar, sobre este valor, funções que são específicas para textos, por exemplo. Apenas quando a variável guarda um texto, é possível usar funções específicas para textos.

Já trabalhamos com números, mas hoje vamos trabalhar com variáveis textuais, por isso, vamos estudar um pouco o que são *strings* e como manipulá-las:

Strings (str)

Uma **string** representa um texto, ou seja, um conjunto encadeado de caracteres. Caractere, por sua vez, é um símbolo gráfico. São exemplos de caracteres as letras do alfabeto, os algarismos, sinais de pontuação de texto, dentre outros. Em uma *string*, os caracteres estão dispostos em uma determinada ordem, e esta ordem importa bastante. Por exemplo, uma variável que guarda o nome 'Bianca' só faz sentido para o usuário respeitando a ordem das letras do nome, ou seja, o seu encadeamento.

Strings em Python são chamadas de **str**. Uma variável *string*, ou seja, uma variável do tipo `str`, representa um texto com zero ou mais caracteres, que devem ser representados entre apóstrofes ou aspas. Exemplos: 'a', 'casa', "carro", "z", 'fl', 'Bianca é linda', ' ', '"', '"', '+', '#\$%@\$%', 'qwerty1234!@#\$', etc.

Operações com strings em Python

Vamos manipular algumas *strings* como exemplo. Para começar, vamos armazenar algumas frases em variáveis. Exemplo:

```
frase1 = 'Bianca é linda'
frase2 = 'Luís é inteligente'
frase3 = 'Roberto é legal'
```

Podemos somar estas variáveis. Somar *strings* significa concatená-las, ou seja, colocá-las uma após a outra. Por exemplo:

```
frase4 = frase1 + frase2 + frase3
```

O resultado guardado em `frase4` será 'Bianca é lindaLuís é inteligenteRoberto é legal', que não estará perfeito, pois a soma simplesmente junta as frases, sem considerar os espaços que seriam necessários para compreendermos melhor o texto. Podemos manipular esta soma colocando caracteres específicos entre as frases. Por exemplo:

```
frase4 = frase1 + '. ' + frase2 + '. ' + frase3 + '.'
```



O resultado será: 'Bianca é linda. Luís é inteligente. Roberto é legal.'

Podemos acessar elementos específicos de uma *string*, considerando que o primeiro caractere está na posição 0, o segundo caractere está na posição 1, o terceiro caractere está na posição 2, e assim por diante. Por exemplo:

```
frase = 'Bianca é linda'  
letra1 = frase[0]  
letra2 = frase[12]
```

O valor na variável `letra1` será 'B', que é o primeiro caractere da *string* `frase`, enquanto que o valor na variável `letra2` será 'd', que é o penúltimo caractere da *string* `frase`. Estas posições dentro de uma *string* são também chamadas de índices.

Podemos percorrer *strings* através de comandos do tipo `for`. Por exemplo:

```
alfabeto = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'  
  
for letra in alfabeto:  
    showInformation('O caractere atual é: ' + letra)
```

O laço `for` vai percorrer todos os caracteres do alfabeto e, em cada execução, vai guardar um dos caracteres da variável `alfabeto` na variável `letra`. O laço `for` acima pode ser interpretado como "para cada caractere na variável `alfabeto`, repita o comando `showInformation()`".

Este mesmo esquema com o laço `for` nos permite procurar por um determinado caractere em uma *string*. Por exemplo:

```
frase = 'a bola rola'  
  
for letra in frase:  
    if letra == 'a':  
        showInformation('Encontrei a letra a no texto!' )
```

Assim, o laço `for` vai percorrer todos os caracteres da variável `frase` e, em cada vez, vai verificar se o caractere é igual à letra 'a'. Se verdadeiro, vai mostrar a mensagem.



FOLHA DE ATIVIDADES

A **cifra de César** é uma das primeiras técnicas de criptografia de que se tem registro na história. Recebeu este nome em homenagem ao conquistador romano Júlio César, que a empregava para proteger importantes mensagens políticas ou militares. Nesta técnica, cada letra da mensagem é substituída por outra que aparece no alfabeto após a letra original, saltando um número fixo de vezes. Os desafios a seguir ajudarão você a implementar um programa, na linguagem Python, que faz o processo de cifragem e decifragem de acordo com a criptografia de César.

Desafio 1 - Solicitando informações

Você precisa solicitar informações do usuário, como:

- se o usuário deseja cifrar ou decifrar a mensagem que inseriu no programa,
- a chave da criptografia usada e
- a mensagem a ser cifrada ou decifrada.

Para solicitar as informações, utilize o comando `requestString`, como no exemplo abaixo:

```
texto = requestString('Digite a mensagem (apenas letras maiusculas):')
```

Desafio 2 - Definindo informações

Você precisa definir uma *string* para ser o alfabeto e uma *string* vazia para armazenar a mensagem cifrada ou decifrada.

Desafio 3 - Implementando a Cifragem

O processo de cifragem da criptografia de César segue a lógica a seguir:

Chave: 3

Mensagem original: **EU AMO ESTUDAR**

Alfabeto normal

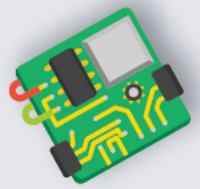
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	O	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Nova posição

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	O	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Mensagem cifrada: **HX DPR HVWXGDR**

Você pode implementar a cifragem através do algoritmo a seguir para cada caractere da mensagem inserida:



Passo 1: Se o caractere for um espaço ou qualquer outro que não seja uma letra do alfabeto, você deve adicioná-lo diretamente ao final da mensagem cifrada

```
texto_convertido = texto_convertido + caractere
```

Passo 2: Se o caractere não for um espaço:

Passo 2.1: Encontre o número dele no alfabeto normal.

```
num = alfabeto.find(caractere)
```

Passo 2.2: Adicione a chave de criptografia a este número e encontre a posição do caractere substituto.

```
posicao = num + chave
```

Passo 2.3: Verifique se o resultado da posição não ultrapassa a quantidade de letras no alfabeto. Se ultrapassar, você deve implementar uma lógica para encontrar a posição correta no alfabeto cifrado.

Passo 2.4: Adicione o caractere ao final da string que armazena a mensagem cifrada

```
texto_convertido = texto_convertido + alfabeto[posicao]
```

Desafio 4 - Implementando a Decifragem

O processo de decifragem da criptografia de César segue a lógica a seguir:

Mensagem cifrada: **WJH YDV**

Chave: 5

Alfabeto cifrado

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Posição do alfabeto normal

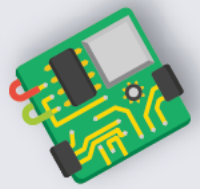
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Mensagem decifrada: **BOM DIA**

Você pode implementar a decifragem através do algoritmo a seguir para cada caractere da mensagem inserida:

Passo 1: Se o caractere for um espaço ou qualquer outro que não seja uma letra do alfabeto, você deve adicioná-lo diretamente ao final da mensagem cifrada

```
texto_convertido = texto_convertido + caractere
```



Passo 2: Se o caractere não for um espaço:

Passo 2.1: Encontre o número dele no alfabeto normal.

```
num = alfabeto.find(caractere)
```

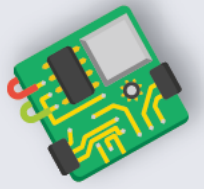
Passo 2.2: Subtraia a chave de criptografia a este número e encontre a posição do caractere substituto.

```
posicao = num - chave
```

Passo 2.3: Verifique se o resultado da posição é menor que zero. Se for, você deve implementar uma lógica para encontrar a posição correta no alfabeto.

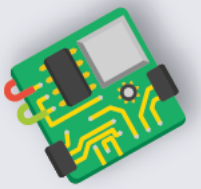
Passo 2.4: Adicione o caractere ao final da string que armazena a mensagem cifrada

```
texto_convertido = texto_convertido + alfabeto[posicao]
```



DIÁRIO DE BORDO

Na aula de hoje



AULA 5 – Jogo da Forca

Nesta aula você irá desenvolver o jogo da Forca utilizando conceitos de strings, indexação e contadores.



TÓPICOS RELEVANTES

Da aula anterior, já sabemos bastante sobre **strings**. Vale lembrar que podemos acessar elementos específicos de uma *string*, considerando que o primeiro caractere está na posição 0, o segundo caractere está na posição 1, o terceiro caractere está na posição 2, e assim por diante. Por exemplo:

```
frase = 'Bianca é linda'  
letra1 = frase[0]  
letra2 = frase[12]
```

O valor na variável `letra1` será 'B', que é o primeiro caractere da *string* `frase`, enquanto que o valor na variável `letra2` será 'd', que é o penúltimo caractere da *string* `frase`. Estas posições dentro de uma *string* são também chamadas de índices.

Aprendemos na aula anterior que o laço `for` nos permite procurar por um determinado caractere em uma *string*. Por exemplo:

```
frase = 'a bola rola'  
  
for letra in frase:  
    if letra == 'a':  
        showInformation('Encontrei a letra a no texto!')
```

Assim, o laço `for` vai percorrer todos os caracteres da variável `frase` e, em cada vez, vai verificar se o caractere é igual à letra 'a'. Se verdadeiro, vai mostrar a mensagem.

Percorrendo strings através de índices

Além deste modo de percorrer uma *string* usando `for`, podemos utilizar o valor do índice da *string*. Este índice assume os valores de 0 até o tamanho da *string* subtraído de um. Por exemplo, a *string* 'Bola' tem tamanho 4 pois tem 4 caracteres. É possível variar os índices desta *string* de 0 até 3 [0 é 'B', 1 é 'o', 2 é 'l', 3 é 'a'].

A função `len()`, que retorna o tamanho da *string*, pode ajudar neste processo.

Por exemplo, se `nome = 'Ada'` e `sobrenome = 'Lovelace'`, a chamada à função `len(nome)` deve retornar o valor 4, enquanto que `len(sobrenome)` deve retornar o valor 10.

A função `range()`, que fornece uma lista de números de 0 até o valor passado para a função subtraído de um, também pode ser útil neste processo.

Por exemplo, a chamada à função `range(5)` retorna a lista de cinco números de 0 a 4, ou seja: [0, 1, 2, 3, 4].

Combinando `range()` com `len()`, podemos saber os índices dos caracteres que precisamos percorrer em uma *string*. Assim, se `sobrenome = 'Lovelace'`, a chamada a `range(len(sobrenome))` deve gerar uma lista de tamanho 8, ou seja, os índices a serem percorridos serão [0, 1, 2, 3, 4, 5, 6, 7], o que pode ser feito usando um laço `for`.



No código a seguir, é possível fazer o percurso da *string* pelos índices com o auxílio da função `len()` e da função `range()`.

```
frase = 'a bola rola'

for indice in range(len(frase)):
    if frase[indice] == 'a':
        showInformation('Encontrei a letra a no texto!')
```

O laço `for` vai percorrer todos os caracteres da variável `frase` e, em cada vez, vai verificar se o caractere da frase naquela posição do `indice` é igual à letra `'a'`. Se verdadeiro, vai mostrar a mensagem, assim como no laço anterior.



FOLHA DE ATIVIDADES

O Jogo da Forca é um jogo muito famoso. Com uma palavra escondida, o jogador deve adivinhar quais letras compõem a palavra misteriosa, fornecendo uma letra a cada rodada. Com base no seu conhecimento do jogo, crie um programa do Jogo da Forca em Python. Para facilitar o seu processo de construção do programa, você pode seguir alguns passos básicos, descritos a seguir.

Crie variáveis: Você pode usar uma variável para guardar a palavra e outra variável para guardar a palavra apenas com as letras que o jogador acertou (`palavraEscondida`). As letras que não foram acertadas devem ser escondidas com o caractere *sublinhado* (`_`), também chamado de *underscore*. Por exemplo, considerando que o jogador acertou as letras A, X e I, as variáveis teriam:

`palavra` `ABACAXI` `palavraEscondida` `A_A_AXI`

Crie um contador: As chances do jogador não podem ser infinitas. Crie um contador para contabilizar as chances. Ele pode iniciar de zero e ir aumentando a cada rodada. Enquanto este contador for menor que o número máximo de chances que você escolher, o jogador poderá continuar dando palpites. Chegando a zero, o jogador perdeu o jogo. Você pode usar um laço `while` ou um laço `for` para controlar as rodadas.

Requisite a letra: Dentro do laço da rodada, exiba uma mensagem pedindo uma letra ao jogador. Utilize a função `requestString` para capturar a letra do teclado.

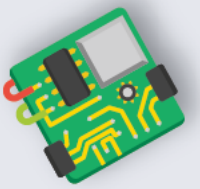
Verifique a letra: Dentro do laço da rodada, percorra a `string` letra por letra e verifique se a letra escolhida é equivalente a alguma letra em `palavra`. Para cada letra percorrida, faça uma das duas opções:

- Se a letra escolhida for igual à letra que está sendo percorrida em `palavra`, junte esta letra a uma variável `string` temporária;
- Se a letra escolhida não for igual à letra que está sendo percorrida em `palavra`, junte a letra em `palavraEscondida` que esteja na mesma posição à variável `string` temporária.

No final do percurso, a variável `string` temporária estará mais próxima da `palavra`. Então, faça `palavraEscondida` guardar o valor desta `string` temporária.

Isso fará com que, ao acertar todas as letras da `palavra`, você tenha a palavra correta guardada em `palavraEscondida`.

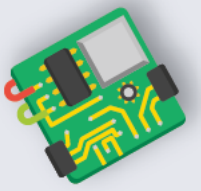
Critério de Parada: Pense um pouco como seria o critério de parada, quando o jogador esgotar suas chances ou quando ele ganhar. É possível adicionar uma condição ao laço `while` ou usar um comando `break` para interromper o laço `for`, faça de acordo com o tipo de laço escolhido anteriormente.



DIÁRIO DE BORDO

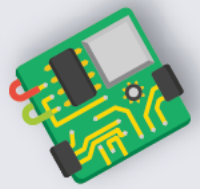
Na aula de hoje

Series of horizontal lines for writing.



AULA 6 – A Droga da Obediência

Nesta aula você criará um programa que implementa o método de criptografia do livro de Pedro Bandeira intitulado A Droga da Obediência.



FOLHA DE ATIVIDADES

O livro *A Droga da Obediência* de Pedro Bandeira descreve um grupo de jovens, intitulado *Os Karas*, que investigavam e ajudavam a desvendar crimes. O grupo cria uma série de codificações para poder trocar mensagens em segurança durante as investigações.

Código Tennis-Polar:

Este código substitui as letras da palavra *tenis* pelas letras correspondentes da palavra *polar*. As letras da mensagem que não estão na palavra *tenis* permanecem iguais. Assim:

T = P
E = O
N = L
I = A
S = R

Exemplos:

CASA = CARA (Apenas o S está na palavra *tenis*)

ESCOLA = ORCOLA (Apenas E e S estão na palavra *tenis*)

Código dos Karas:

Este código substitui as vogais e mantém as consoantes. Assim:

A = AIS
E = ENTER
I = INIS
O = OMBER
U = UFTER

Exemplos:

CASA = CAISSAIS

ESCOLA = ENTERSCOMBERLAIS

Desafio 1 – Código dos Karas

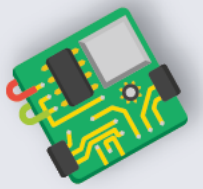
Implemente um programa que codifique e decodifique mensagens com o Código dos Karas.

Desafio 2 – Combinando códigos

Implemente um programa que codifique e decodifique mensagens com a seguinte lógica:

Codificação: Transforma o texto seguindo o Tennis-Polar e, em seguida, transforma o texto seguindo o Código dos Karas.

Decodificação: Transforma o texto seguindo o Código dos Karas e, em seguida, transforma o texto seguindo o Tennis-Polar.



DIÁRIO DE BORDO

Na aula de hoje



AULA 7 – Ovos da Páscoa

Nesta aula você desenvolverá o código para encontrar uma mensagem escondida em um arquivo de texto através da linguagem Python.



FOLHA DE ATIVIDADES

Na Computação, o termo Ovos de Páscoa, ou *Easter Eggs*, é muito utilizado para designar mensagens escondidas em um programa de computador. Há muito tempo, alguns programadores esconderam os seus nomes dentro dos jogos de console. Esse conteúdo só poderia ser acessado se os jogadores realizassem alguns passos definidos. Hoje em dia, empresas como a Microsoft, Facebook e Google possuem vários *Easter Eggs* para divertir os usuários. Por exemplo, se digitarmos “Do a Barrel Roll” no buscador da Google vamos ter uma surpresa!

Nesta atividade, há uma mensagem escondida em um arquivo de texto. Analise o texto e verifique o que a mensagem esconde. Em seguida, crie uma estratégia, baseada no padrão que você descobriu, para que o seu programa possa exibir a mensagem.

```
Esse texto tem Uma mensagem escondida. O Unico que pode
desvendar esse misterio e UM(a) bravo(a) guErreiro(a),
eXtremamente habilidoso(a) e Curioso(a) sobrE os segredos
LENdarios da criptografia. esse guerreiro(a) deve dominar as
arTEs imortais da PROGRAMAcAo. nao sera facil, mas certamente
voce sera capaz de DescObRir a mensagem!
```

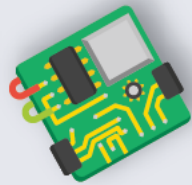
Abrindo o arquivo: Para abrir o arquivo de texto, você pode utilizar a função *open*. Esta função permite abrir um arquivo que está no seu computador, bastando passar para a função o caminho do arquivo, ou seja, em qual pasta o arquivo está no seu computador e o nome dele. O arquivo pode ser aberto de diversas formas, para leitura (modo *read*) ou para escrita (*write*). Neste caso, vamos apenas ler a mensagem, então, usaremos o modo de leitura (*read*), identificado pela letra ‘r’, na função *open*. Após a execução da função, todo o conteúdo do arquivo estará na variável texto.

```
texto = open('C:/Usuário/Desktop/texto_unidade3_aula7.txt', 'r')
```

Percorrendo o arquivo: Para percorrer o arquivo, nós podemos utilizar dois laços, um dentro do outro (também chamados de laços aninhados). O primeiro, para percorrer linha por linha. É possível exibir a linha inteira. O segundo, para percorrer cada letra de cada linha. Caso o arquivo só tenha uma linha, não precisaremos de dois laços. O código a seguir exibe o conteúdo da linha e depois cada letra desta linha.

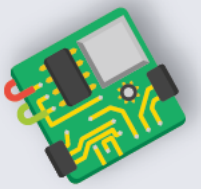
```
for linha in texto:
    print(linha)
    for letra in linha:
        print(letra)
```

Identificando a mensagem: Agora você está pronto para implementar o seu código. Utilize suas habilidades com condicionais, loops e strings para exibir a mensagem.



DIÁRIO DE BORDO

Na aula de hoje



AULA 8 – Jogo da Velha

Nesta aula você criará um programa que implementa o Jogo da Velha. Neste jogo, o jogador enfrenta o computador.



FOLHA DE ATIVIDADES

Na aula de hoje, você vai implementar um Jogo da Velha inteligente, onde o jogador enfrenta o computador! Siga os desafios propostos:

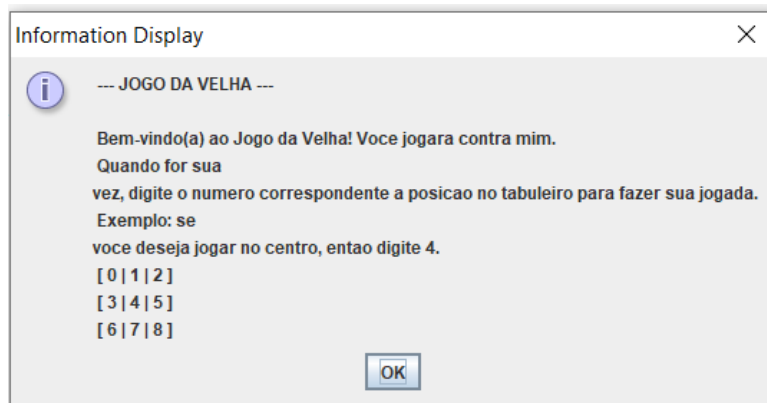
Desafio 1 – Definindo variáveis necessárias

Você vai precisar de algumas variáveis para armazenar:

- Quantas jogadas o jogo já teve;
- O tabuleiro com as nove posições do jogo;
- Quem é o jogador da vez;
- As jogadas do jogador humano;
- As jogadas do computador;
- O vencedor.

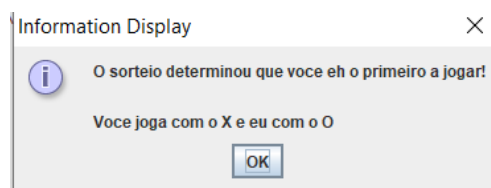
Desafio 2 – Apresentando o Tabuleiro

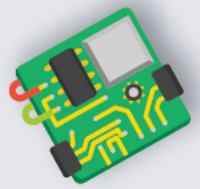
Crie a tela de apresentação do seu jogo. Você pode informar as instruções principais:



Desafio 3 – Quem joga primeiro?

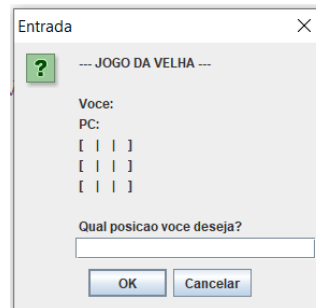
Defina quem é o primeiro jogador, o humano ou o computador. Use uma variável específica para armazenar quem é o jogador da vez. Informe ao usuário quem é o primeiro jogador.





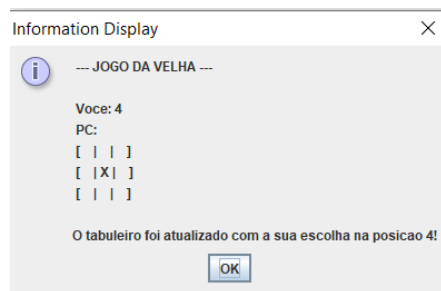
Desafio 4 – O que acontece quando o jogo está na vez do jogador humano?

O jogador humano deve escolher uma posição livre no tabuleiro. Para isso, crie uma tela em que o usuário visualize o estado atual do tabuleiro e insira a posição em que deseja jogar.



Ao inserir a mensagem, o programa deve:

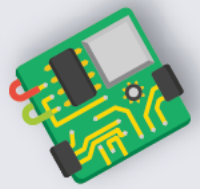
- Marcar na variável específica a posição do tabuleiro escolhida;
- Adicionar a posição à lista de jogadas do jogador humano;
- Incrementar o número de jogadas;
- Determinar que é a vez do jogador computador;
- Apresentar o tabuleiro atualizado.



Desafio 4 – O que acontece quando está na vez do jogador PC?

O jogador PC ou seja, o computador, deve fazer uma jogada. Sinta-se livre para elaborar uma jogada mais sofisticada, avaliando as chances de vitória, por exemplo. Após definir a posição de jogada, o programa deve:

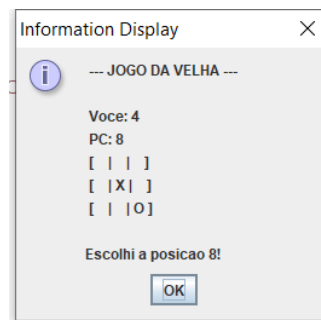
- Marcar na variável específica a posição do tabuleiro escolhida;
- Adicionar a posição à lista de jogadas do jogador humano;
- Incrementar o número de jogadas;
- Determinar que é a vez do jogador humano;
- Apresentar o tabuleiro atualizado.



Dica: Uma das ideias mais simples é se o computador simplesmente sortear uma posição livre no tabuleiro. O programa deve sortear um número e verificar se ele corresponde a uma posição livre no tabuleiro. Se sim, prossiga, senão, você deve sortear outro número. Para isso, você pode utilizar uma função de sorteio de números aleatórios chamada `randint`.

```
escolha = randint(0,8)
while tabuleiro[escolha] == 'X' or tabuleiro[escolha] == 'O' :
    escolha = randint(0,8)
```

Depois do sorteio é importante informar a escolha ao jogador PC e mostrar o tabuleiro atualizado.



Desafio 5 – Checando o vencedor

Sempre que uma jogada for feita, você deve verificar se houve vitória. Considerando que o tabuleiro possui 9 posições, que podem ser acessadas pelos índices 0 a 8 da lista, estas são as opções de vitória:

0	1	2

3	4	5

6	7	8

0		
	4	
		8

0		
3		
6		

	1	
	4	
	7	

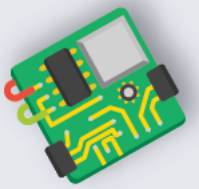
		2
		5
		8

		2
	4	
6		

Ao todo, oito possibilidades devem ser verificadas. Você deve verificar se o jogador humano ganhou, se o jogador PC ganhou ou se a quantidade de jogadas ultrapassou 9 (então ninguém ganhou). Você deve utilizar o resultado dessas verificações para continuar o jogo, repetindo as jogadas, ou parar o jogo e informar o resultado final.

UNIDADE IV

Arduino



AULA 1 - Construindo um Semáforo

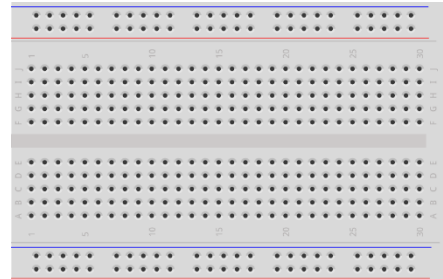
Nesta aula você irá conhecer o universo da eletrônica com Arduino. Através de três projetos simples, você aprenderá como programar a placa Arduino para incluir comportamentos específicos em circuitos com LEDs.



TÓPICOS RELEVANTES

Nesta unidade vamos manipular diversos componentes durante a montagem dos projetos com Arduino. Os componentes utilizados para a montagem dos projetos da aula de hoje são:

- **Protoboard:** é uma placa que serve como base para a montagem dos circuitos. Possui uma lógica específica para estabelecer as conexões.



- **LED:** diodo emissor de luz (em inglês: *light-emitting diode*). Componente utilizado para a emissão de luz. Possui polaridade, que pode ser identificada do seguinte modo: terminal negativo é o menor terminal.

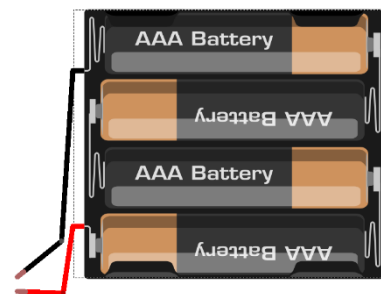


- **Jumper:** fio utilizado para estabelecer conexões entre os componentes. Podem ser do tipo macho/macho, quando tem as duas extremidades pontiagudas, macho/fêmea, quando tem apenas uma das extremidades pontiagudas ou fêmea/fêmea, quando nenhuma das extremidades é pontiaguda.

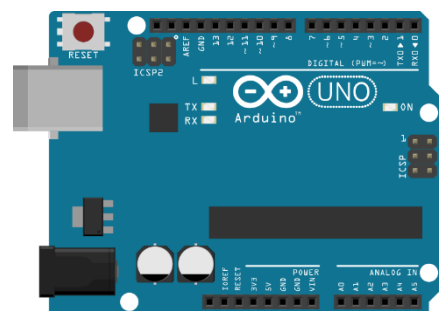
- **Resistor:** Componente que pode ser utilizado com a finalidade de transformar energia elétrica em energia térmica, por meio do efeito Joule, ou para limitar a corrente elétrica em um circuito. Em nossos projetos, o resistor tem como função limitar a corrente elétrica e utilizaremos um modelo 220R sempre associado aos LEDs.



- **Suporte para pilhas:** é um suporte onde pilhas AA podem ser encaixadas e combinadas para obter uma determinada voltagem. Cada pilha AA tem 1,5V e nossos suportes acomodam 4 pilhas em série, ou seja, 6V.



- **Arduino:** Placa de circuito que possui um circuito integrado interno e diversos módulos, podendo ser programada para a criação de protótipos.





FOLHA DE ATIVIDADES

Seja bem-vindo ao universo da eletrônica com Arduino! Nesta unidade, você irá trabalhar um bocado com esta placa que é amplamente utilizada para a criação de protótipos tecnológicos. Sabe como é, né? Você já é praticamente um engenheiro criador de tecnologias! Na aula de hoje, vamos implementar três projetos.

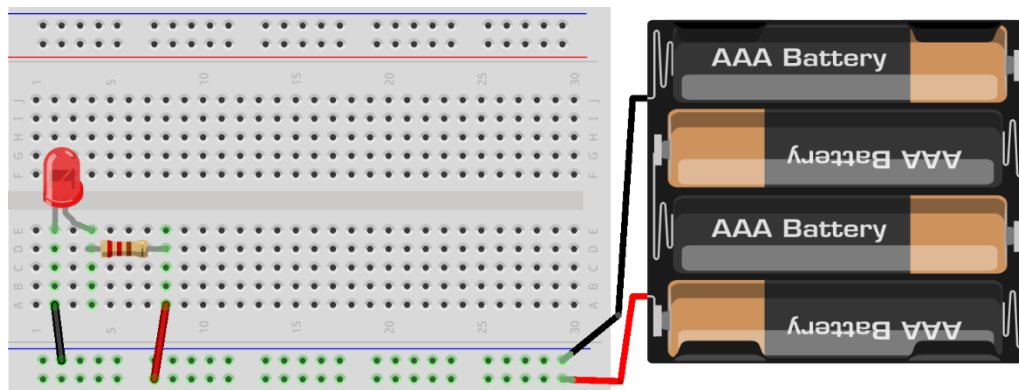
Projeto 1 - Ligando um LED

OBJETIVO: Acender um LED, alimentando a *protoboard* com quatro pilhas AA de 1,5 V.

COMPONENTES NECESSÁRIOS:

Quantidade	Item
1	LED (com a cor que você desejar)
1	Resistor 220R
2	Jumpers do tipo macho/macho
1	Suporte para quatro pilhas AA
4	Pilhas AA 1,5 V

ESQUEMÁTICO:



Projeto 2 - Ligando Três LEDs

OBJETIVO: Acender três LEDs, com as cores vermelho, amarelo e verde. O circuito deve ser alimentado por quatro pilhas AA de 1,5 V.

COMPONENTES NECESSÁRIOS:

Quantidade	Item
3	LED (cores: vermelho, amarelo e verde)
3	Resistor 220R
6	Jumpers do tipo macho/macho
1	Suporte para quatro pilhas AA
4	Pilhas AA 1,5 V

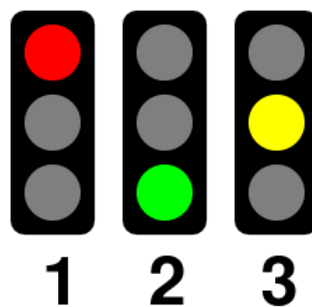


ESQUEMÁTICO: Como você tem se mostrado um engenheiro supertalentoso, não vamos oferecer o esquemático desta vez. Utilize a mesma lógica do circuito do Projeto 1!

Projeto 3 – Construindo um Semáforo

OBJETIVO: Construir um circuito que tenha o comportamento de um semáforo.

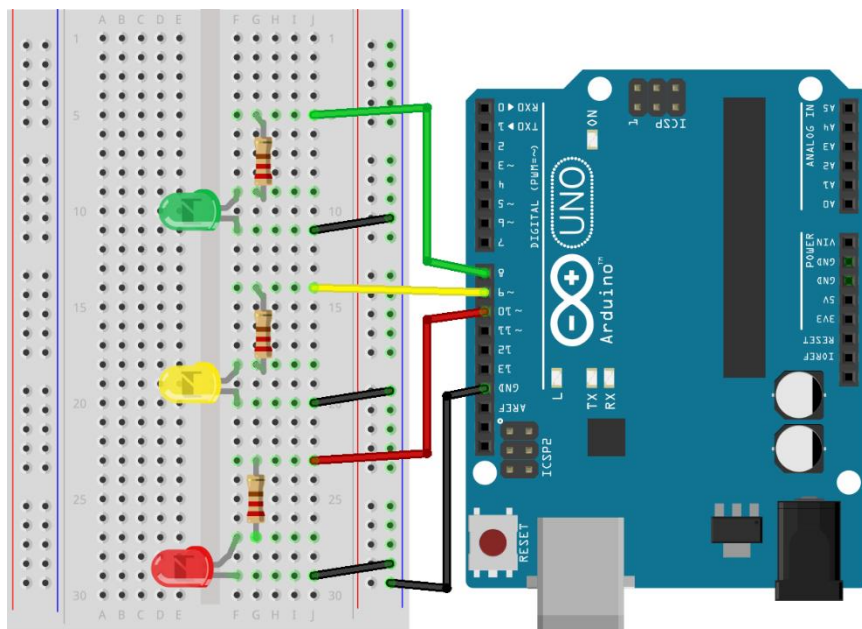
REQUISITOS: O circuito deve ter três LEDs, com as cores vermelho, amarelo e verde. O circuito tem três estados: deve passar um tempo no sinal vermelho, depois deve passar um tempo no sinal verde e, por fim, um tempo menor no sinal amarelo. Este processo de mudança de estados deve se repetir sempre!

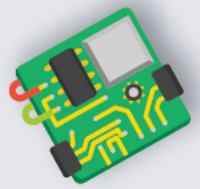


COMPONENTES NECESSÁRIOS:

Quantidade	Item
1	Arduino Uno
3	LED (cores: vermelho, amarelo e verde)
3	Resistor 220R
6	Jumpers do tipo macho/macho

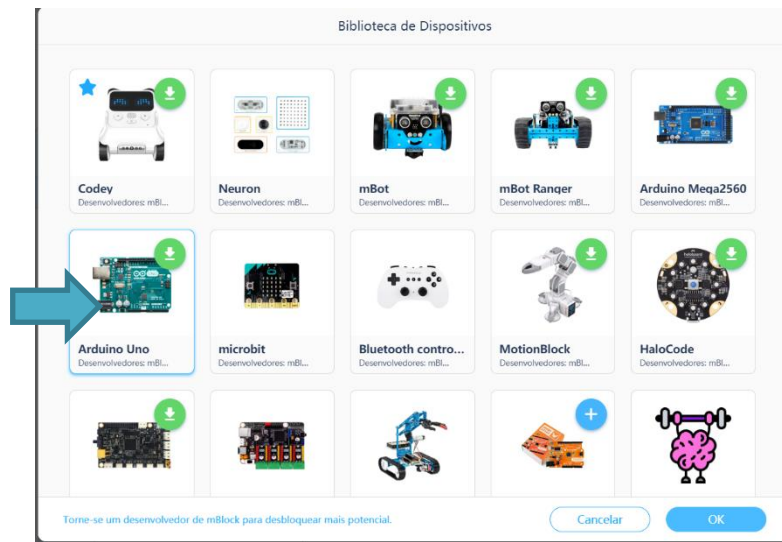
ESQUEMÁTICO:



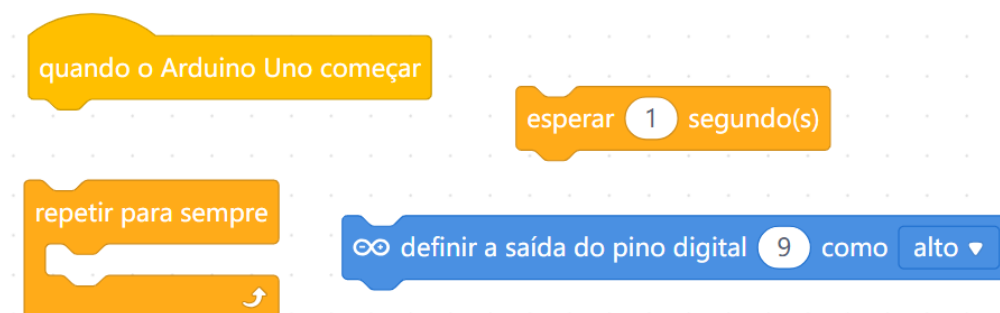


PROGRAMAÇÃO NO MBLOCK: Para criar um circuito com comportamentos bem definidos, você deve utilizar o Arduino e programá-lo de acordo com o objetivo do circuito. O Arduino Uno pode ser programado através do mBlock (você lembra do mBlock, né?). Como você é um programador muito experiente, não é necessário que apresentemos o programa inteiro, mas vamos te dar algumas dicas.

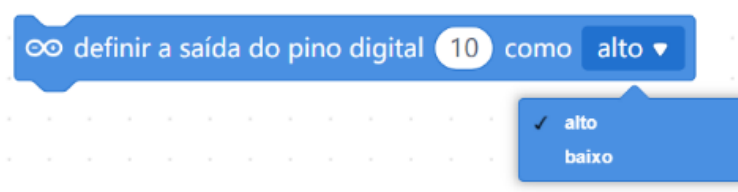
1ª Dica: Ao abrir o mBlock, remova o dispositivo padrão “Codey” e adicione o dispositivo Arduino Uno.

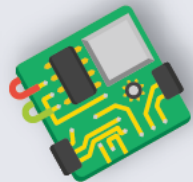


2ª Dica: Para programar o Arduino, você deve utilizar os comandos abaixo. Pense num algoritmo para o seu programa.



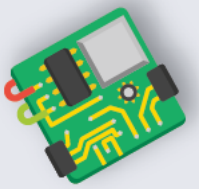
O comando *Definir saída do pino digital...* é o mais importante! Nele, você deve especificar a porta digital do Arduino onde o componente está conectado e se, através dessa porta, passará ou não corrente. Por exemplo, o LED vermelho está conectado à porta digital 10 do Arduino. Para programá-lo, vamos utilizar este comando, passando 10 como parâmetro. Para ligar o LED, devemos escolher a opção “alto” e, para desligar o LED, devemos escolher a opção “baixo”.





DIÁRIO DE BORDO

Na aula de hoje



AULA 2 – Medindo a Luminosidade em um Ambiente

Nesta aula você irá conhecer o sensor de luminosidade LDR e criará um circuito capaz de medir a luminosidade em um ambiente.



FOLHA DE ATIVIDADES

Continuando a nossa aventura pelo universo da Eletrônica, hoje é dia de medir luminosidade através do sensor chamado *Light Dependent Resistor* – LDR, um resistor dependente de luz.

Calculando o resultado da conversão Analógico-Digital!

Na aula anterior, ao trabalhar com LEDs, utilizamos as portas digitais do Arduino. Portas digitais trabalham com apenas dois valores de sinais: 0V ou 5V. Ou seja, ou tem sinal ou não tem sinal naquele pino! O sensor LDR, que é tema de estudo da aula de hoje, é um sensor analógico.

Para utilizar o LDR, precisamos conectá-lo a uma porta analógica do Arduino. Apesar de só trabalhar internamente com sinais digitais, o Arduino utiliza de um conversor analógico-digital que transforma a tensão, que é analógica, num valor digital. Um valor da tensão num pino de entrada analógica é convertido em um valor de 0 a 1023. Se o valor lido for 0, então a tensão na entrada do pino é 0V, mas se o valor lido for 1023, então a tensão lida é 5V. Como você é fera na matemática, responda as questões a seguir:

Qual valor será lido pela porta analógica do Arduino, se ele receber uma tensão de entrada de 1V?

Qual valor será lido pela porta analógica do Arduino, se ele receber uma tensão de entrada de 4V?

Qual valor será lido pela porta analógica do Arduino, se ele receber uma tensão de entrada de 2,5V?

Projeto 1 – Medindo a Luminosidade

OBJETIVO: Construir um circuito capaz de medir a luminosidade em um ambiente.

REQUISITOS: Este circuito terá dois LEDs, vermelho e azul, e um sensor de luminosidade LDR. Ao medir a luminosidade do ambiente, devem ser identificados 3 níveis de luminosidade: escuro, parcialmente escuro e claro. Ao detectar o nível escuro, apenas o LED vermelho deve ser ligado. Ao identificar o nível parcialmente escuro, apenas o LED azul deve ser ligado. Ao identificar o nível claro, os dois LEDs devem permanecer apagados. O valor de entrada para o nível escuro é entre

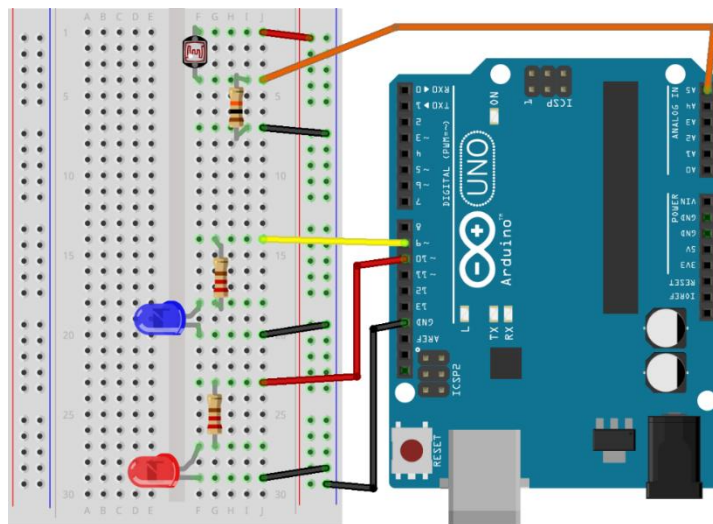


0 e 2,5V; o valor de entrada para o nível parcialmente escuro é entre 2,5 e 4V; e o valor para o nível claro é acima de 4V.

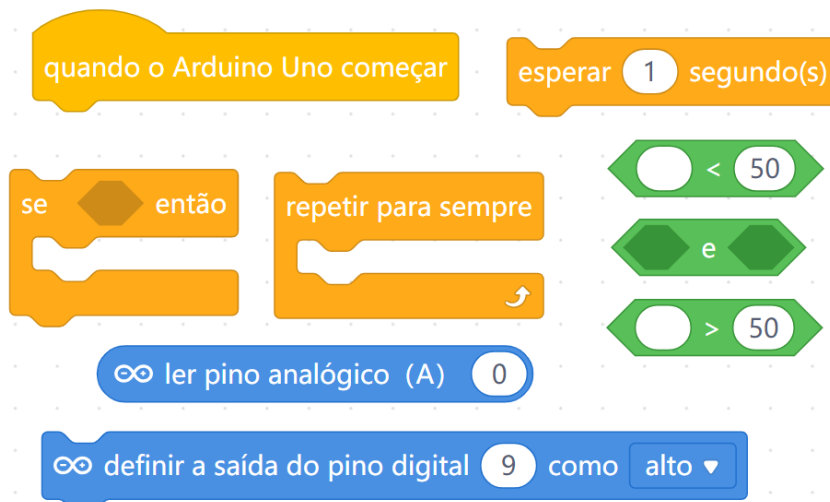
COMPONENTES NECESSÁRIOS:

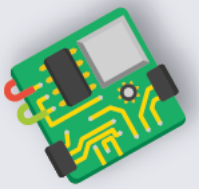
Quantidade	Item
1	Arduino Uno
1	LED (com a cor que você desejar)
1	Resistor 220R
1	Sensor LDR
1	Resistor de 10K
8	Jumpers do tipo macho/macho

ESQUEMÁTICO:



PROGRAMAÇÃO NO MBLOCK: Para criar o programa do circuito que mede luminosidade, você deve utilizar os comandos abaixo, podendo usar outros blocos caso ache necessário. Como você é um programador expert, deve pensar num algoritmo que faça a leitura do valor do sensor LDR, identifique em qual nível de luminosidade o valor lido se encaixa, e definir as ações correspondentes, de acordo com o objetivo definido para o circuito.





AULA 3 – Criando um Piano com o Arduino

Nesta aula você irá desenvolver o projeto de Piano com o Arduino, utilizando um buzzer, para emitir sons, e push buttons, que funcionam como botões.



FOLHA DE ATIVIDADES

Continuando a nossa aventura pelo universo da Eletrônica, hoje é dia de fazer barulho com dois novos componentes: o *buzzer* e o *push button*!

Projeto 1 – Piano Arduino

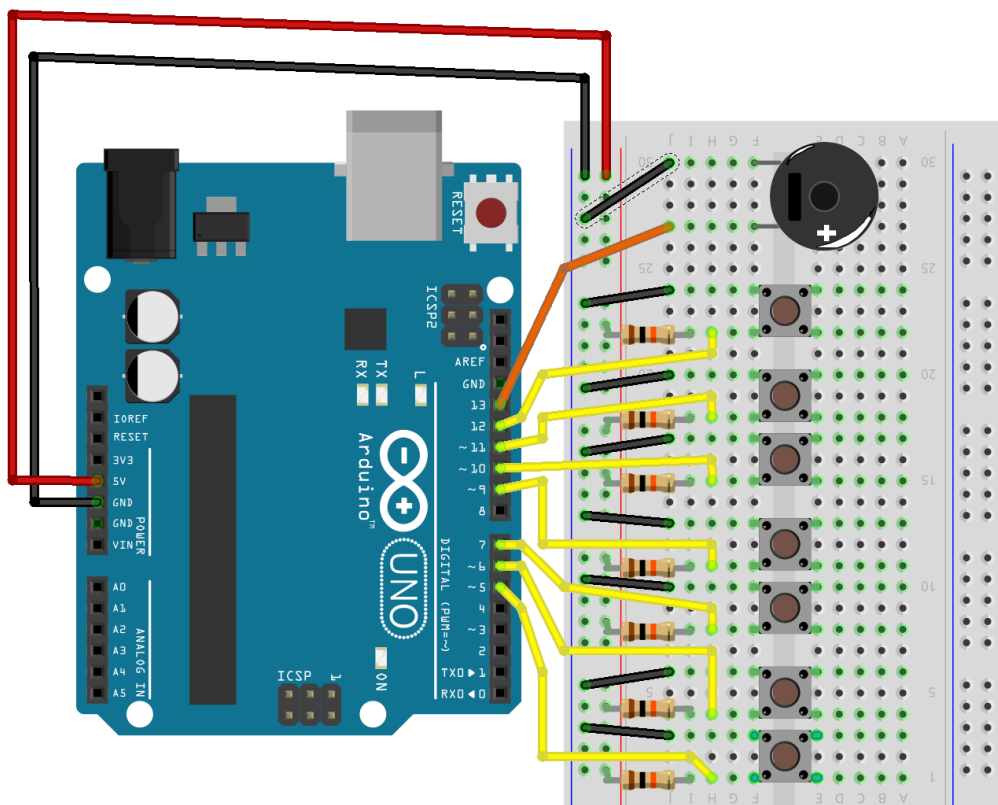
OBJETIVO: Construir um piano capaz de tocar as sete notas musicais: Dó, Ré, Mi, Fá, Sol, Lá, Si.

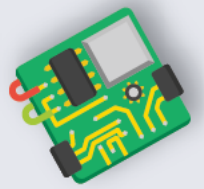
REQUISITOS: Este circuito terá sete *push buttons*, cada um correspondente a uma nota musical, e um *buzzer* que reproduzirá as notas quando os *push buttons* forem pressionados.

COMPONENTES NECESSÁRIOS:

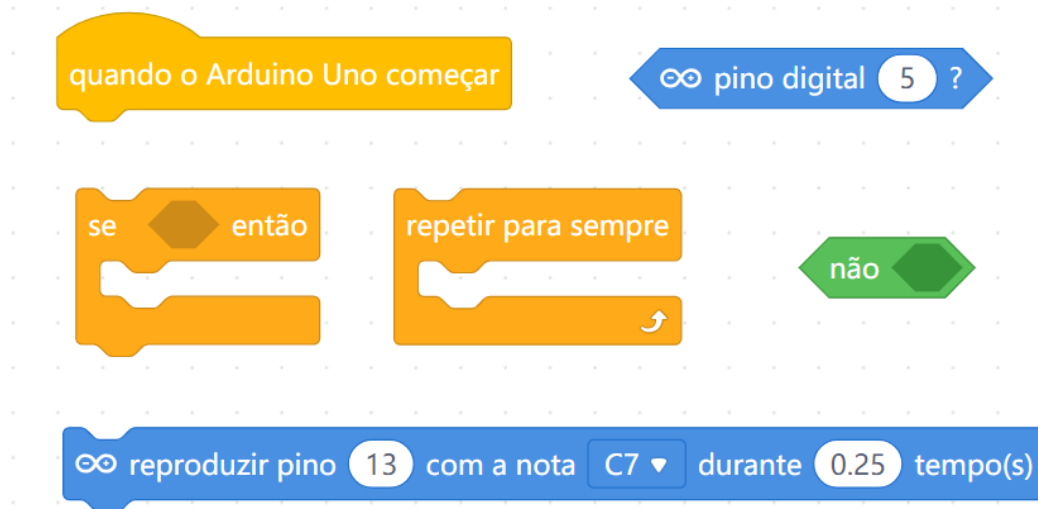
Quantidade	Item
1	Arduino Uno
7	<i>Push button</i>
7	Resistor 10K
1	<i>Buzzer</i>
18	Jumpers do tipo macho/macho
1	Protoboard de 400 pontos

ESQUEMÁTICO:





PROGRAMAÇÃO NO MBLOCK: Para criar o programa do circuito piano, você deve utilizar os comandos abaixo, podendo usar outros blocos caso ache necessário. Como você é um programador expert, deve pensar num algoritmo que identifique quando um dos *push buttons* é pressionado e toque a nota correspondente a esta 'tecla' do piano no *buzzer*.



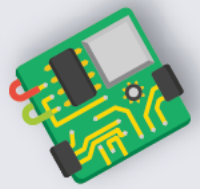
Projeto 2 – Minipiano Arduino

OBJETIVO: Construir um piano capaz de tocar quatro notas musicais (Dó, Ré, Mi, Fá) e acender um LED correspondente a cada nota, promovendo um efeito visual no circuito.

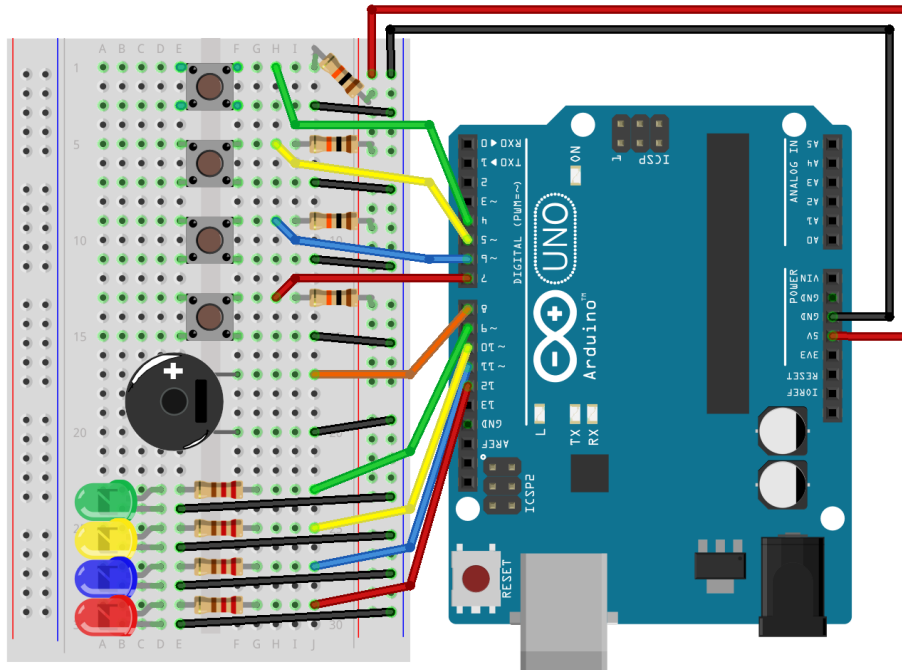
REQUISITOS: Este circuito terá quatro *push buttons*, cada um correspondente a uma nota musical. O *buzzer* deve reproduzir as notas musicais, de acordo com o *push button* pressionado. Cada 'tecla' do minipiano terá um LED associado, que deve ser aceso sempre que a tecla estiver pressionada.

COMPONENTES NECESSÁRIOS:

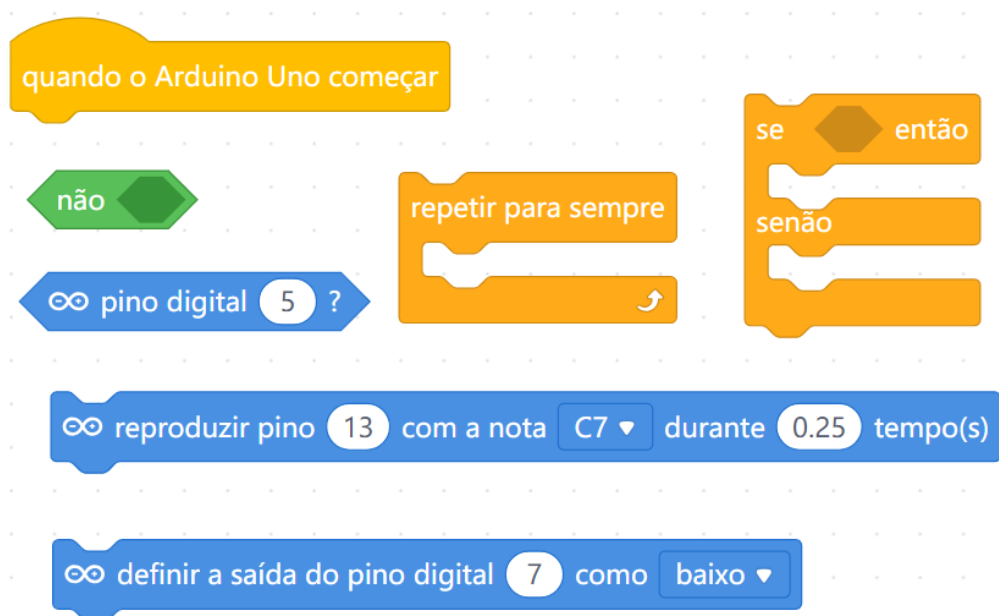
Quantidade	Item
1	Arduino Uno
4	<i>Push button</i>
4	Resistor 10K
4	LED (cores distintas)
4	Resistor 220R
1	<i>Buzzer</i>
20	Jumpers do tipo macho/macho
1	Protoboard de 400 pontos

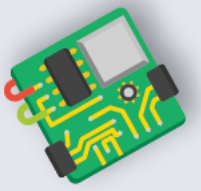


ESQUEMÁTICO:



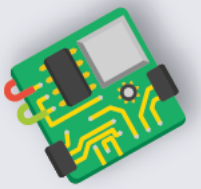
PROGRAMAÇÃO NO MBLOCK: Para criar o programa do circuito minipiano, você deve utilizar os comandos abaixo, podendo usar outros blocos caso ache necessário. Como você é um programador expert, deve pensar num algoritmo que identifique quando um dos *push buttons* é pressionado, toque a nota correspondente a esta 'tecla' do piano no *buzzer* e acenda o LED correspondente, e o apague quando a tecla for solta.





DIÁRIO DE BORDO

Na aula de hoje



AULA 4 – Criando um Robô Dançarino

Nesta aula você irá desenvolver o projeto do Robô Dançarino, envolvendo servomotores e o Arduino.



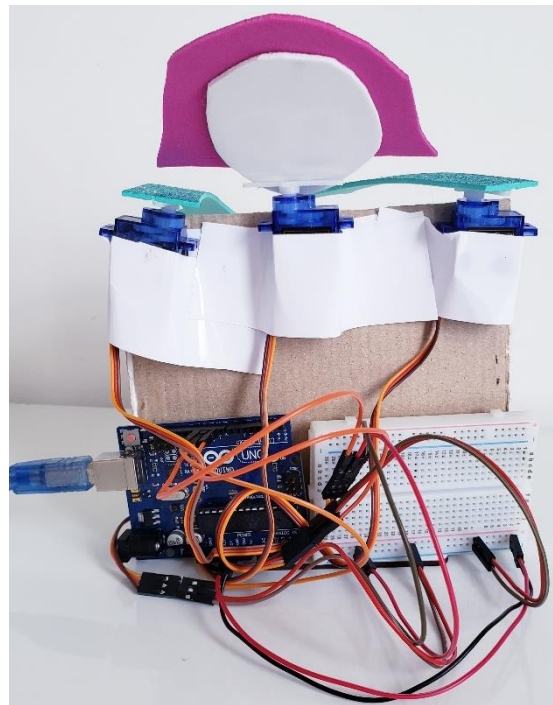
FOLHA DE ATIVIDADES

Continuando a nossa aventura pelo universo da eletrônica, hoje é dia de dançar com um novo componente: o Micro Servo 9g SG90 TowerPro! Este servomotor é capaz de posicionar seu “braço” entre 0° e 180°.

Projeto 1 – Robô Dançarino

OBJETIVO: Construir um robô que execute movimentos dos braços e da cabeça de maneira sincronizada com uma música de escolha do projetista.

REQUISITOS: Este circuito terá três servomotores. O projetista deve escolher uma música e criar uma sequência para cada servomotor, simulando a coreografia da música escolhida. Antes da montagem do circuito, o projetista deve criar uma base para o robô, incluindo os braços e a cabeça. Um servo faz o movimento da cabeça, um servo faz o movimento do braço direito e outro servo faz o movimento do braço esquerdo.

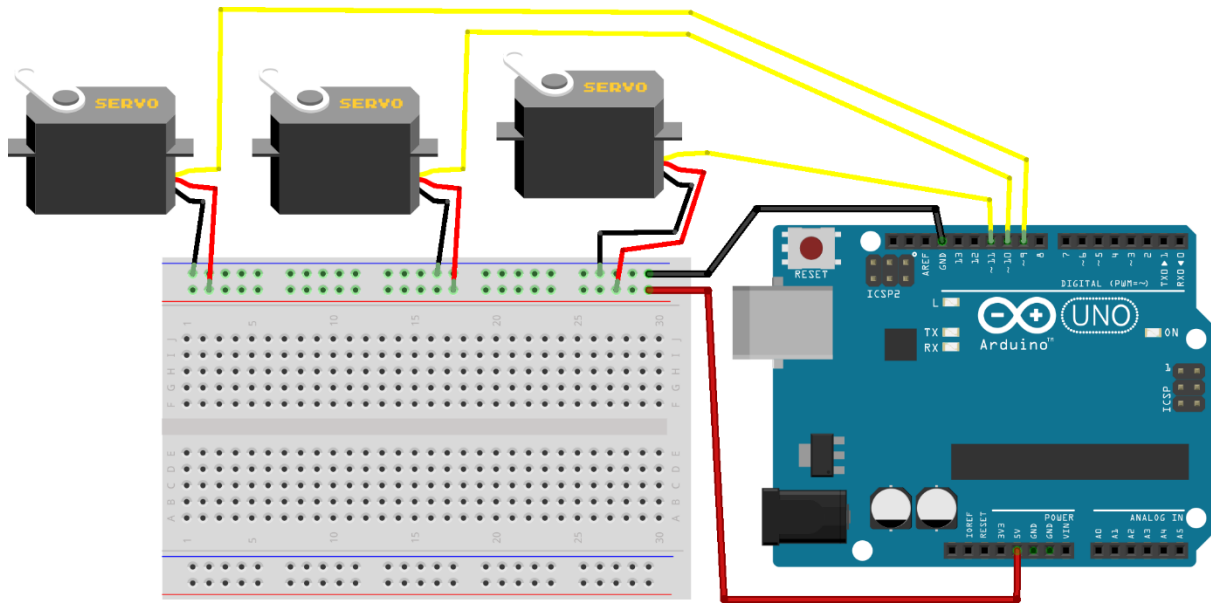


COMPONENTES NECESSÁRIOS:

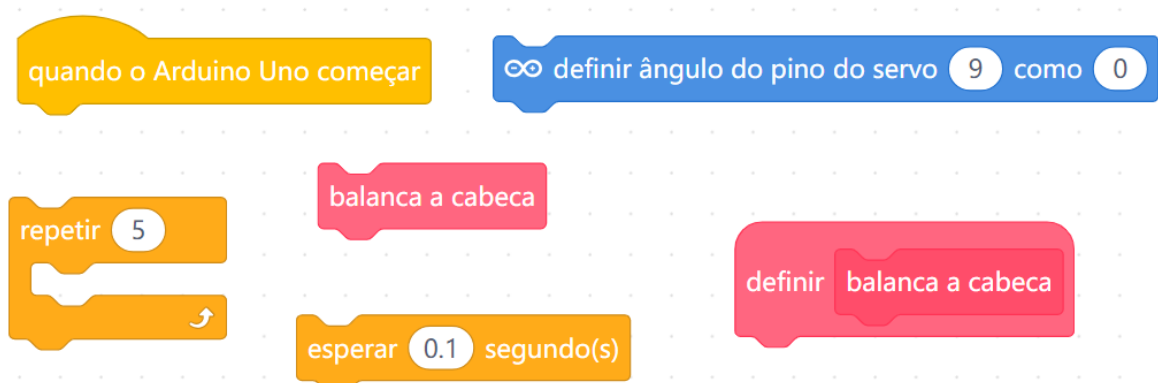
Quantidade	Item
1	Arduino Uno
3	Micro Servo 9g SG90 TowerPro
1	Protoboard de 400 pontos
11	Jumpers do tipo macho/macho



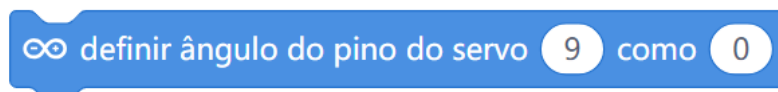
ESQUEMÁTICO:

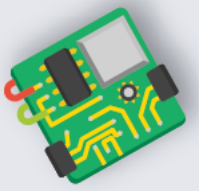


PROGRAMAÇÃO NO MBLOCK: Para criar o programa do Robô Dançarino, você deve utilizar os comandos abaixo, podendo usar outros blocos caso ache necessário. Como você é um programador expert, deve pensar num algoritmo que estabeleça uma sequência de ações para os servos. Para que seu programa não fique muito grande, você pode utilizar a criação de blocos, na categoria Meus Blocos. Por exemplo, definir um bloco para quando o robô balanceia a cabeça.



O comando que especifica a posição do servomotor é *Definir Ângulo Do Pino Servo...* Nele, você deve especificar em que porta digital do Arduino o Servo está conectada e o ângulo em que seu braço deve ser posicionado. O modelo de servomotor que trabalhamos só se posiciona entre 0° e 180°.





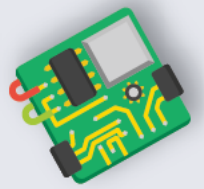
DIÁRIO DE BORDO

Na aula de hoje



AULA 5 – Criando um Robô Guardião do Tesouro

Nesta aula você irá trabalhar com o sensor ultrassônico e deve alterar o projeto do Robô Dançarino, transformando-o no Robô Guardião do Tesouro.



FOLHA DE ATIVIDADES

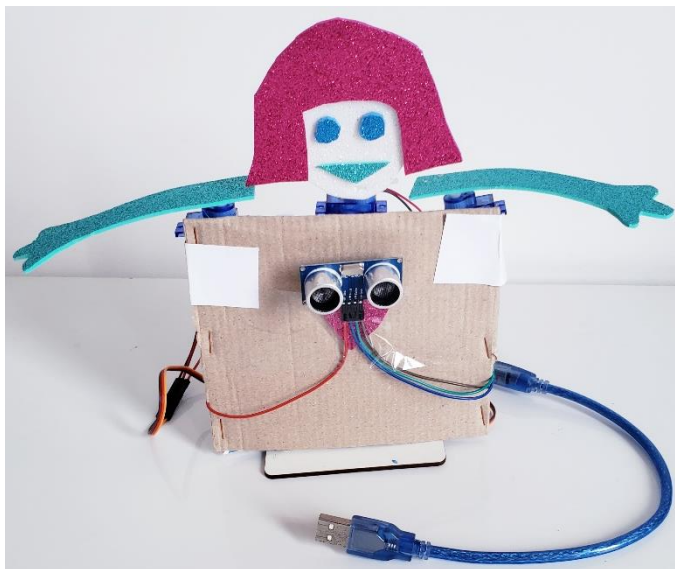
Continuando a nossa aventura pelo universo da eletrônica, hoje é dia de conhecer um novo componente: o Sensor Ultrassônico HSR04! Este sensor é capaz de medir distâncias entre 3 centímetros e 4 metros.

Projeto 1 – Robô Guardião do Tesouro

OBJETIVO: Construir um robô que identifique intrusos e proteja um objeto.

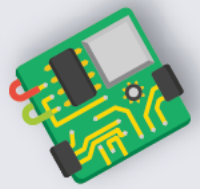
REQUISITOS: Este circuito terá três servomotores e um sensor ultrassônico. Um servo faz o movimento da cabeça, um servo faz o movimento do braço direito e o outro servo faz o movimento do braço esquerdo. O projetista deve definir um comportamento para os braços e a cabeça do robô se um objeto for detectado pelo sensor ultrassônico a uma distância menor do que 20 cm. O projetista pode alterar este limite de 20 cm caso julgue necessário durante a fase de teste do projeto.

O projetista deve aproveitar o projeto criado de Robô Dançarino, e promover alterações neste circuito, adicionando o sensor ultrassônico.

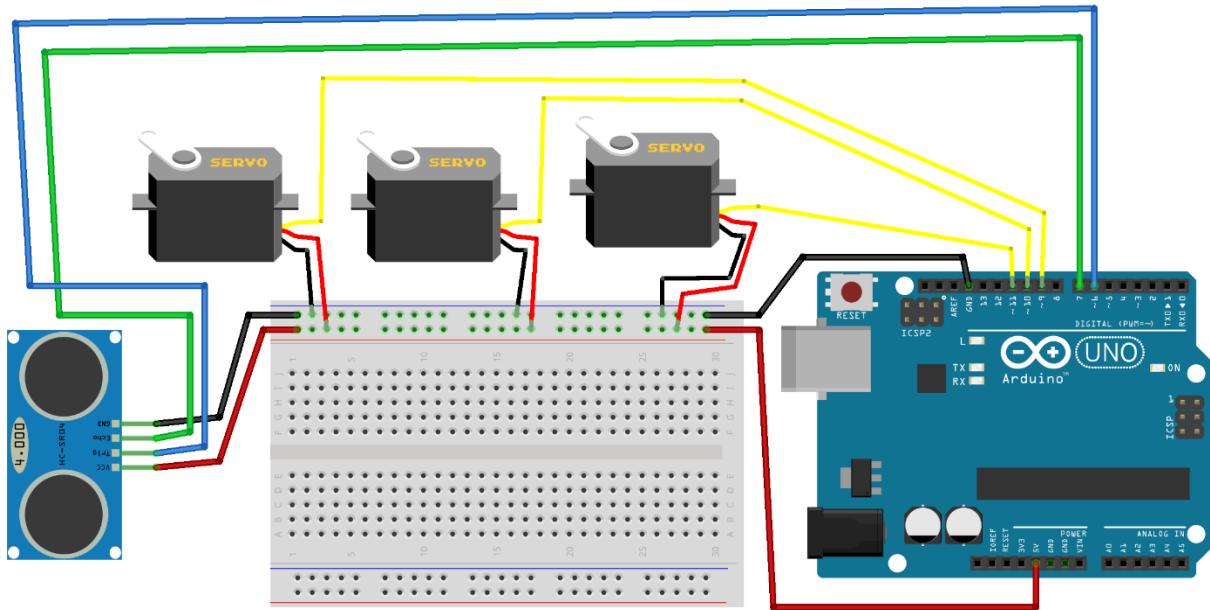


COMPONENTES NECESSÁRIOS:

Quantidade	Item
1	Arduino Uno
1	Sensor Ultrassônico HSR04
3	Micro Servo 9g SG90 TowerPro
1	Protoboard de 400 pontos
15	Jumpers do tipo macho/macho



ESQUEMÁTICO:



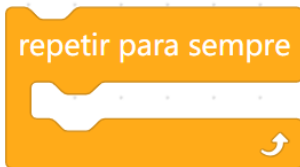
PROGRAMAÇÃO NO MBLOCK: Para criar o programa do Robô Guardiã do Tesouro, você deve utilizar os comandos abaixo, podendo usar outros blocos caso ache necessário. Como você é um programador expert, deve pensar num algoritmo onde o robô fique com os braços abertos e, se identificar um objeto a uma distância menor do que 20 cm, então o robô fecha os braços. O projetista também pode definir um comportamento para a cabeça do robô.

quando o Arduino Uno começar



ler pin trigonométrico do sensor ultrasónico 13 pin de eco 12

repetir para sempre



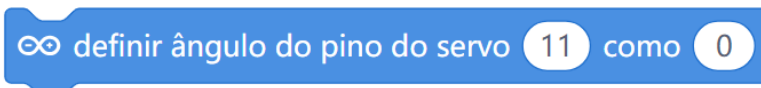
se então



esperar 0.5 segundo(s)



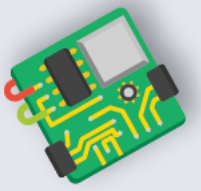
definir ângulo do pino do servo 11 como 0



O comando que especifica a posição do servomotor é *Ler pin trigonométrico do sensor ultrasónico...* Nele, você deve especificar em que porta digital do Arduino o terminal *Trig* do sensor está conectado e o pino digital onde o terminal *Echo* do sensor está conectado.

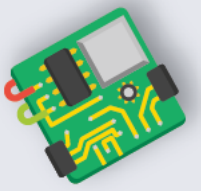
ler pin trigonométrico do sensor ultrasónico 13 pin de eco 12





DIÁRIO DE BORDO

Na aula de hoje



AULA 6 – Como funciona o Computador?

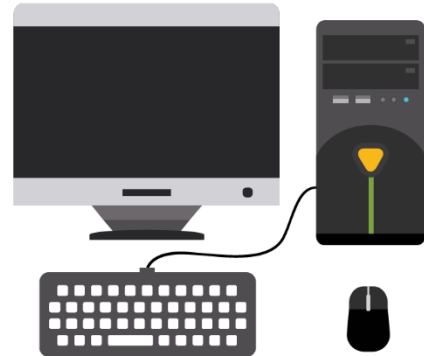
Nesta aula você irá aprender sobre a arquitetura de hardware dos computadores.



TÓPICOS RELEVANTES

Um computador é uma máquina destinada ao processamento de informação. Os computadores costumam ser classificados de acordo com a capacidade de processamento e sua função. Assim, são alguns tipos de computadores: microcomputadores, videogames, servidores, supercomputadores, desktops, smartphones, sistemas embarcados (lembra do Arduino?), dentre outros.

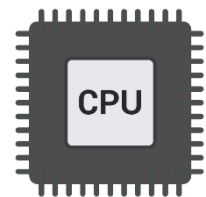
Na prática, os computadores são dispositivos eletrônicos programáveis que recebem dados, executam operações lógicas e matemáticas em alta velocidade e exibem o resultado destas operações.



Desde os primeiros computadores da década de 1940, a tecnologia utilizada nos computadores digitais mudou bastante. Apesar disso, quase todos os computadores ainda utilizam uma variante da arquitetura proposta pelo matemático John von Neumann na primeira metade do século XX. A chamada Arquitetura de von Neumann tem quatro partes principais: unidade lógica e aritmética, unidade de controle, memória e dispositivos de entrada e saída. A unidade lógica e aritmética e a unidade de controle se combinam formando o processador. Para interligar todas essas partes, existe a placa-mãe, que conecta os dispositivos através de barramentos e fios.

Processador

Todo computador possui um processador, também conhecido como CPU (*Central Processing Unit*). O processador é composto pela unidade lógica e aritmética, unidade de controle e registradores:



- **Unidade Lógica e Aritmética (ULA):** é a parte do processador que executa as operações aritméticas e lógicas entre números. As ULAs podem ter apenas as operações de adição e subtração, mas algumas ULAs podem incluir outras operações como a multiplicação e divisão. Também possuem as operações lógicas (por exemplo: E, Ou, Não).
- **Unidade de controle (UC):** é a parte do processador que controla seu funcionamento interno além de gerar sinais que controlam o funcionamento externo à CPU. A função mais importante da UC é executar as instruções dos programas. Ela recebe uma instrução da memória, interpreta-a e gera os sinais para as outras partes do computador (ULA, memória, dispositivos de entrada e saída) para executá-la.
- **Registradores:** são a parte do processador dedicada ao armazenamento de dados. Os registradores são necessários pois evitam que o processador acesse a memória do computador continuamente. Cada registrador é uma pequena porção de memória que pode ser lida e escrita de maneira mais rápida que a memória primária do computador.



Memória

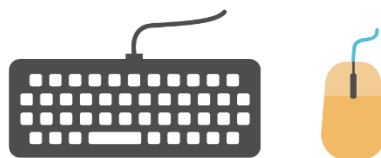
Todo computador precisa armazenar dados e a parte responsável por isso é a memória. A memória pode ser de leitura e escrita ou apenas de leitura. Leitura significa o processador receber dados da memória, enquanto escrita significa o processador enviar dados para a memória.

Um computador normalmente tem mais de um componente de memória. Geralmente, a memória está dividida em memória primária e memória secundária:

- **Memória primária:** é uma memória que pode ser acessada diretamente pelo processador. Fazem parte desta memória:
 - **Memória cache:** memória de acesso rápido, situada muito próxima à ULA do processador.
 - **Memória de acesso aleatório (Random Access Memory - RAM):** é uma memória composta por células de armazenamento numeradas, capazes de armazenar instruções a serem executadas pelo processador, ou dados necessários para que o processador execute alguma instrução. É uma memória volátil, ou seja, quando o computador é desligado os dados são perdidos.
 - **Memória apenas de leitura (Read-Only Memory - ROM):** é uma memória não volátil e que só pode ser lida. Esse tipo de memória armazena o *firmware* do computador, que é um programa que controla as funções principais do hardware de modo que o computador possa ser ligado e estabeleça comunicação adequada com os dispositivos de entrada e saída.
- **Memória Secundária:** é usada para gravar os dados em geral, como fotos e documentos. Este tipo de memória não perde os dados com o desligamento do computador. A principal memória secundária do computador é o **Disco Rígido (Hard Disk - HD)**. Ela também armazena os programas instalados de forma que possam ser executados sempre que o computador for ligado. Além disso, armazena os arquivos com dados de modo que possam sempre ser acessados quando o computador está ligado. Alguns computadores mais atuais usam a **memória SSD (solid-state drive)** no lugar do HD. As memórias secundárias possuem acesso mais lento que as memórias primárias. Também são considerados memória secundária de um computador os CDs, DVDs, e pen drives.

Dispositivos de entrada e saída

Os dispositivos de entrada e saída são também chamados de periféricos. Dispositivos de entrada codificam a informação passada pelo usuário do computador em tipos de dados digitais que podem ser processados pelo computador. São exemplos de dispositivos de entrada: teclado, mouse, webcam, scanner, microfone e touchpad.





Os dispositivos de saída decodificam a informação de modo que ela possa ser entendida pelo usuário do computador. São exemplos de dispositivos de saída: monitor, projetor, alto-falantes, fones de ouvido e impressora.



Alguns dispositivos periféricos podem ser tanto de entrada como de saída. Por exemplo, placas de rede com fio ou sem fio, dispositivos *Bluetooth* são dispositivos de entrada e saída.

Outros elementos

Além dos elementos processador, memória e dispositivos de entrada e saída, existem outros componentes que fazem parte da arquitetura do computador. São eles: a placa-mãe, a fonte de alimentação e o gabinete que armazena todos os componentes.

A placa-mãe é uma placa de circuito com diversas entradas, chamadas *slots*, que conectam todos os componentes do computador. Através da placa-mãe processador, memória RAM, disco rígido, placa gráfica e os demais componentes são interligados. A organização de uma placa-mãe varia conforme o modelo e fabricante, mas a função principal sempre se mantém. A principal função da placa-mãe é permitir o tráfego de informação, mas ela também alimenta alguns periféricos com a energia elétrica que recebe da fonte de alimentação.

A fonte de alimentação do computador é um componente que converte a voltagem de energia elétrica disponível nas tomadas, algo em torno de 110 ou 220 volts, para voltagens menores que são suportadas pelos componentes do computador, como 12 volts ou menos. Em computadores como notebooks, a fonte também costuma ser uma bateria, permitindo que o computador funcione desconectado da internet.

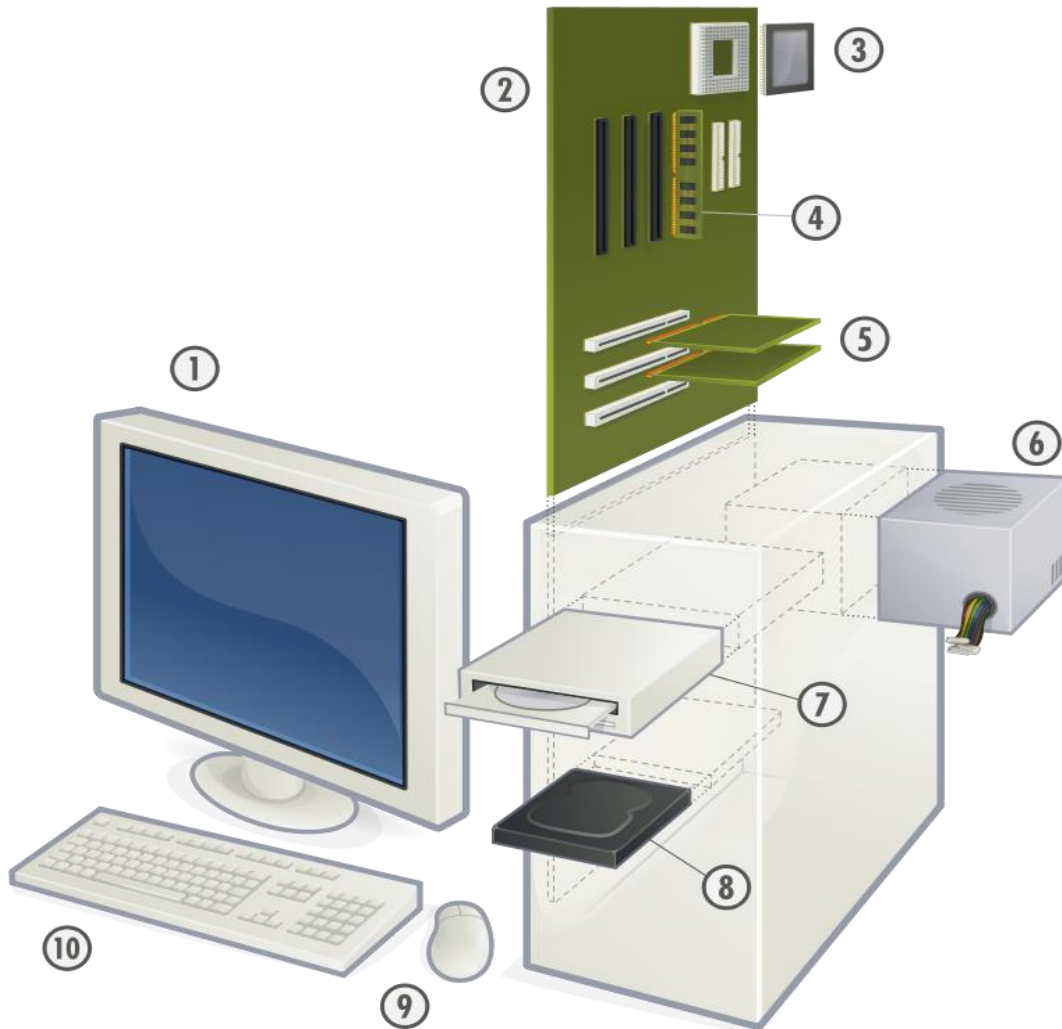
O gabinete computador é o compartimento que tem a maioria dos componentes, incluindo a placa-mãe e os dispositivos a ela acoplados. Normalmente, o monitor, o teclado e o mouse ficam de fora do gabinete. Muitas vezes o gabinete é chamado de CPU, mas sabemos que CPU é, na verdade, o processador do computador.



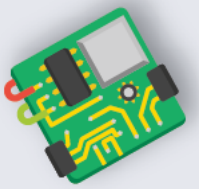
FOLHA DE ATIVIDADES

Exercício 1 – Partes do Computador

A Figura abaixo exibe um computador desktop tradicional com suas partes evidenciadas. Coloque o nome do dispositivo e classifique, quando pertinente, a parte listada como sendo processador, memória ou dispositivo de entrada e saída:



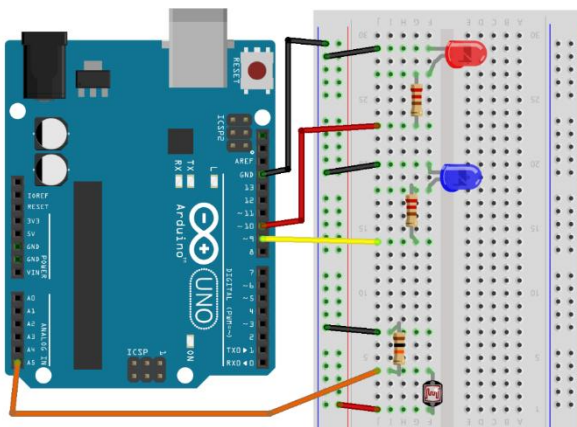
- 01- _____
- 02- _____
- 03- _____
- 04- _____
- 05- _____
- 06- _____
- 07- _____
- 08- _____
- 09- _____



Exercício 2 - Processamento, entrada ou saída?

Durante esta unidade criamos vários projetos com Arduino. Muitos destes projetos recebem informações do ambiente ou do usuário, processam estas informações e retornam alguma resposta. Para cada um dos projetos a seguir, identifique os componentes que estão encarregados do processamento e memória, entrada de dados ou saída de dados.

Medindo a luminosidade do ambiente:

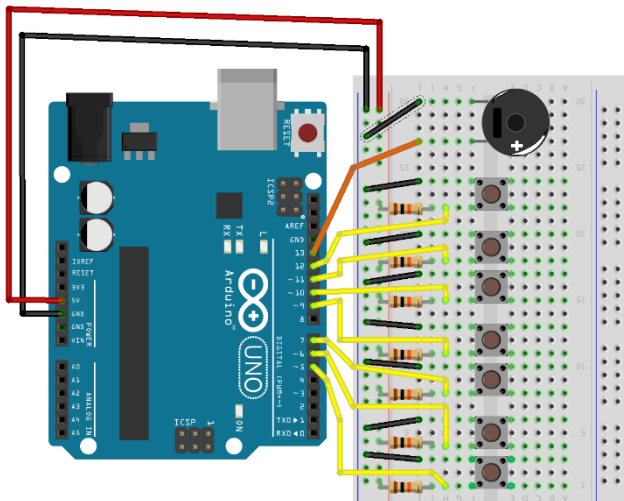


LEDS:

LDR:

ARDUINO:

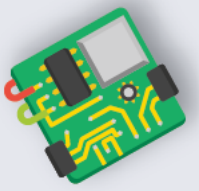
Piano com Arduino:



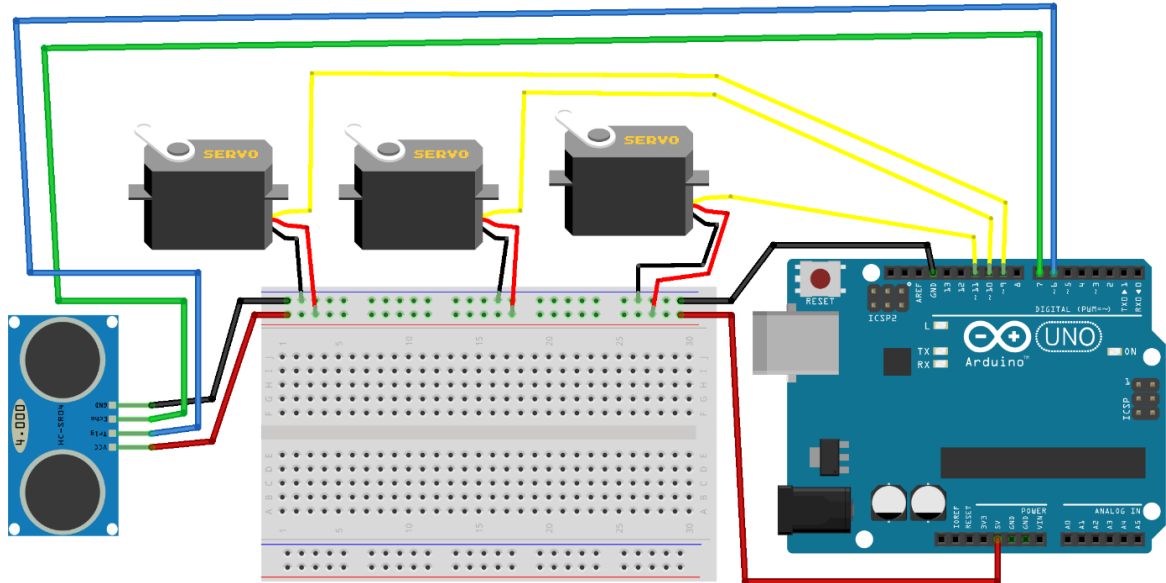
PUSH-BUTTONS:

BUZZER:

ARDUINO:



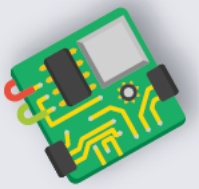
Robô Guardiã do Tesouro:



SENSOR ULTRASSÔNICO:

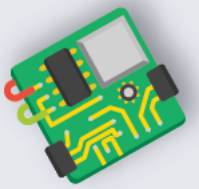
SERVO MOTOR:

ARDUINO:



AULA 7 – Criando seu Primeiro Protótipo com o Arduino

Nesta aula você irá criar um protótipo de um produto de seu interesse através de um circuito com o Arduino e outros componentes eletrônicos aprendidos ao longo da unidade.



FOLHA DE ATIVIDADES

Nesta unidade, já trabalhamos com diversos tipos de circuitos com Arduino. Também aprendemos diversos aspectos relevantes de robótica e eletrônica, como as limitações inatas dos componentes eletrônicos e as dificuldades de programar um comportamento autônomo e inteligente. Dentre várias possibilidades:

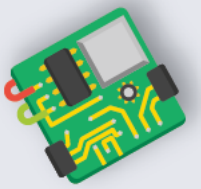
- Aprendemos a ligar e desligar LEDs;
- Aprendemos a fazer barulho com o *Buzzer*;
- Aprendemos a programar circuitos com botões;
- Aprendemos a criar movimentos com o servomotor;
- Aprendemos a medir luminosidade com o sensor LDR;
- Aprendemos a medir distâncias com o sensor ultrassônico.

Protótipo com Arduino

Agora que você conhece muito sobre eletrônica com Arduino, crie seu primeiro protótipo de um produto eletrônico. Seu circuito deve ser o mais inteligente possível e pode se parecer com um robô. Você também pode usar a sua criatividade e embarcá-lo em criações artesanais, como nos projetos do robô dançarino e do robô guardião do tesouro.

Seu circuito deve ter pelo menos quatro dos componentes a seguir, sendo que um deles deve ser obrigatoriamente um sensor:

- LED;
- *Buzzer*;
- *Push button*;
- Servomotor;
- Sensor de luminosidade;
- Sensor Ultrassônico.



AULA 8 – Apresentando seu Primeiro Protótipo com o Arduino

Nesta aula você irá apresentar o protótipo de produto criado com o Arduino para a comunidade.

