



PROFEPT

INSTITUTO FEDERAL
Mato Grosso do Sul

**PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO
PROFISSIONAL E TECNOLÓGICA**

CAMPUS CAMPO GRANDE


TM

**ELETROPRATICANDO:
ATIVIDADES PRÁTICAS ON-LINE INTEGRANDO
ELETRODINÂMICA E ARDUINO**

GUIA DO DOCENTE

HUGO EDUARDO PIMENTEL MOTTA SISCAR

ORIENTADOR - PROF. DR. DANTE ALIGHIERI ALVES DE MELLO



Siscar, Hugo Eduardo Pimentel Motta
S622e Eletropraticando: atividades práticas on-line integrando eletrodinâmica e arduino: guia do docente / Hugo Eduardo Pimentel Motta Siscar, Dante Alighieri Alves de Mello. – Campo Grande-MS, 2022.
126 f. : il. color.

Produto educacional (Mestrado em Educação Profissional e Tecnológica) – Programa de Pós-Graduação em Educação Profissional e Tecnológica, Instituto Federal de Mato Grosso do Sul-IFMS, Campus Campo Grande, 2022.

Orientador: Prof. Dr. Dante Alighieri Alves de Mello.

Inclui referências.

Inclui ISBN 978-65-00-57671-9.

1. Produto educacional. 2. Eletrodinâmica. 3. Arduino. I. Mello, Dante Alighieri Alves de. II. Instituto Federal de Mato Grosso do Sul. Programa de Pós-Graduação em Educação Profissional e Tecnológica. III. Título.

CDD 23. ed. 537.6

Autores da pesquisa

Hugo Eduardo Pimentel Motta Siscar

<http://lattes.cnpq.br/7879983413287026>

Graduação em Engenharia da Computação - UNIDERP

Especialização em Docência para Educação Profissional e Tecnológica - IFMS

Mestrado em Educação Profissional e Tecnológica - IFMS



Dante Alighieri Alves de Mello

<http://lattes.cnpq.br/8505049911605733>

Licenciatura Plena em Física - UFMS

Mestrado em Química - UFMS

Doutorado em Educação - UFMS



Ir para...

APRESENTAÇÃO

SUGESTÕES DE APLICAÇÃO

REFERENCIAL TEÓRICO

CRONOGRAMA PROPOSTO

AVALIAÇÃO DIAGNÓSTICA

ATIVIDADES PRÁTICAS

RESOLUÇÃO DAS ATIVIDADES

APÊNDICE - AMBIENTE VIRTUAL TINKERCAD PARA O DOCENTE

TM

APRESENTAÇÃO

Caro(a) docente,

A concepção deste material surgiu da elaboração de um Produto Educacional final da minha Dissertação de Mestrado, intitulado: "Proposta de aplicação de atividades práticas **on-line** integrando Eletrodinâmica e Arduino", sendo assim, alguns conteúdos aqui apresentados são referentes à esses temas.

As atividades abordarão conceitos de corrente elétrica, resistores, potenciômetros, Light Dependent Resistor (LDR), Light Emitter Diode (LED), Arduino e linguagem de programação.

Além deste material, foi elaborado um Caderno de Atividades para os Estudantes acompanharem as atividades propostas.

Você poderá clicar nos ícones presentes nas atividades para acessar alguns recursos.



Vídeos



Links



SUGESTÕES DE APLICAÇÃO

TM

Quem pode aplicar?



Docente da área de Física



Docente da área de Física + Docente da base técnica

Para quem aplicar?



Alunos do Curso Técnico em Informática ou áreas afins



Organização da turma em pequenos grupos

Quando aplicar?



Ao término ou durante a aplicação dos conteúdos de corrente elétrica e resistores

Oficina

REFERENCIAL TEÓRICO

TEORIA HISTÓRICO-CULTURAL DE LEV S. VYGOSTKY

TM

Premissas



Conceitos espontâneos e científicos
Mediação e Interação
Imitação
Zona de Desenvolvimento Proximal



CLIQUE PARA
SABER MAIS...

Contexto de aplicação

A metodologia da aplicação deve atender **às necessidades** do docente.

Pode ser aplicado **integralmente** ou em **partes**.



CRONOGRAMA PROPOSTO

TM

Encontro 01

10 min – Apresentação do Projeto de Pesquisa

50 min – Avaliação Diagnóstica

20 min – *Apresentação do Ambiente Virtual Tinkercad*

10 min – *Apresentação da Ferramenta Google Chat*

CRONOGRAMA PROPOSTO

TM

Encontro 02

30 min – Atividade 01

30 min – Atividade 02

10 min – Aula: LED

20 min – Atividade 03

CRONOGRAMA PROPOSTO

TM

Encontro 03

10 min – Correção da Atividade 03

50 min – Aula: Arduino

20 min – Prática orientada: Atividade 04

10 min – Atividade 05

CRONOGRAMA PROPOSTO

TM

Encontro 04

10 min – Correção da Atividade 05

10 min – Aula: Potenciômetro

25 min – Prática orientada: Atividade 06

10 min – Aula: LDR

20 min – Prática orientada: Atividade 08

10 min – Atividade 07

05 min – Atividade 09



CRONOGRAMA PROPOSTO

TM

Encontro 05

15 min – Correção da Atividade 07

15 min – Correção da Atividade 09

60 min – Avaliação da Aprendizagem



ATIVIDADES PROPOSTAS

1. ASSOCIAÇÃO DE RESISTORES EM SÉRIE
2. ASSOCIAÇÃO DE RESISTORES EM PARALELO
3. CALCULANDO O VALOR DO RESISTOR PARA LIGAR O LED
4. PISCA LED COM ARDUINO
5. SEMÁFORO
6. LENDO O VALOR DE UM POTENCIÔMETRO - POT
7. REGULANDO O BRILHO DO LED POR MEIO DO POT
8. LENDO O VALOR DE UM LDR
9. REGULANDO O BRILHO DO LED POR MEIO DO LDR



AVALIAÇÃO DIAGNÓSTICA

TM

<https://forms.gle/HYsX7RxiG3z2Dwqq5>

AVALIAÇÃO DIAGNÓSTICA

Questão 01 - A corrente elétrica está presente no dia a dia das pessoas, seja no celular, aparelhos eletrônicos, no uso da eletricidade em casa, dentre outros. Assim, conceitue corrente elétrica.

RESOLUÇÃO

AVALIAÇÃO DIAGNÓSTICA

Questão 02 - Alguns componentes elétricos estão presentes no nosso dia a dia mas nem sempre os percebemos. Um exemplo disso são os resistores.

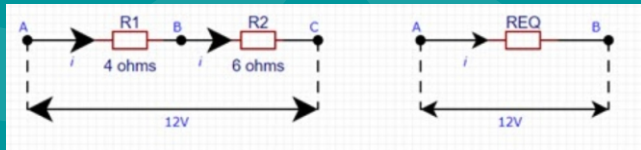
O que são resistores?

Onde eles são frequentemente utilizados?

RESOLUÇÃO

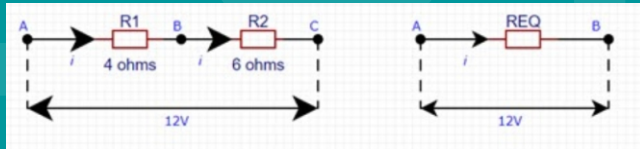
AVALIAÇÃO DIAGNÓSTICA

Questão 3 - Frequentemente, em determinados tipos de “pisca-pisca” de Natal, diversas lâmpadas são associadas em série. Em condições específicas as lâmpadas podem ser consideradas resistores ôhmicos. Com o objetivo de estudar estes circuitos um estudante associou em série por um condutor ideal dois resistores ôhmicos, de valores 4Ω e 6Ω . Uma bateria ideal fornece aos extremos da associação uma diferença de potencial (ddp) de $12V$, como mostra a figura abaixo.



AVALIAÇÃO DIAGNÓSTICA

Questão 3 - Com base nessas informações, calcule: (a) a resistência equivalente da associação? (b) a intensidade de corrente elétrica em cada resistor:



RESOLUÇÃO

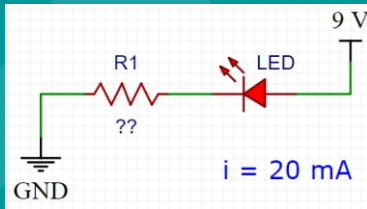
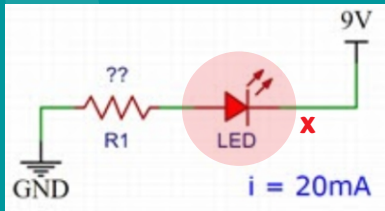
AVALIAÇÃO DIAGNÓSTICA

Questão 4 - (PUC) Três resistores idênticos de $R = 30\Omega$ estão ligados em paralelo com uma bateria de 12V. Pode-se afirmar que a resistência equivalente do circuito é de:

RESOLUÇÃO

AVALIAÇÃO DIAGNÓSTICA

Questão 5 - O circuito abaixo possui uma fonte de alimentação de 9V, um resistor R1 e um LED (light-emitting diode ou diodo emissor de luz). Considerando que o LED necessite de uma tensão de 2V e uma corrente de 20mA para funcionar corretamente, calcule o valor do resistor para acionar esse LED.



RESOLUÇÃO

Ambiente Virtual TinkerCad - para estudantes

Acesse o link: www.tinkercad.com

Clique em:

«Fazer login,
Inscrever-se
ou em
Participar da aula»

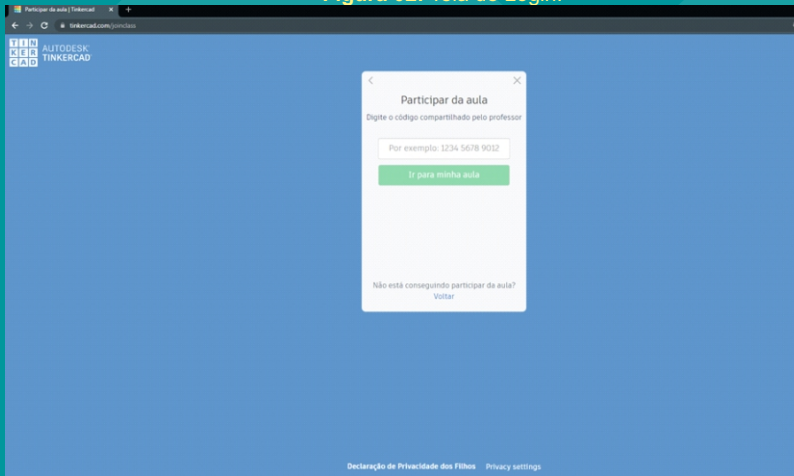
Figura 01: Tela inicial.



Ambiente Virtual Tinkercad

TM

Figura 02: Tela de Login.



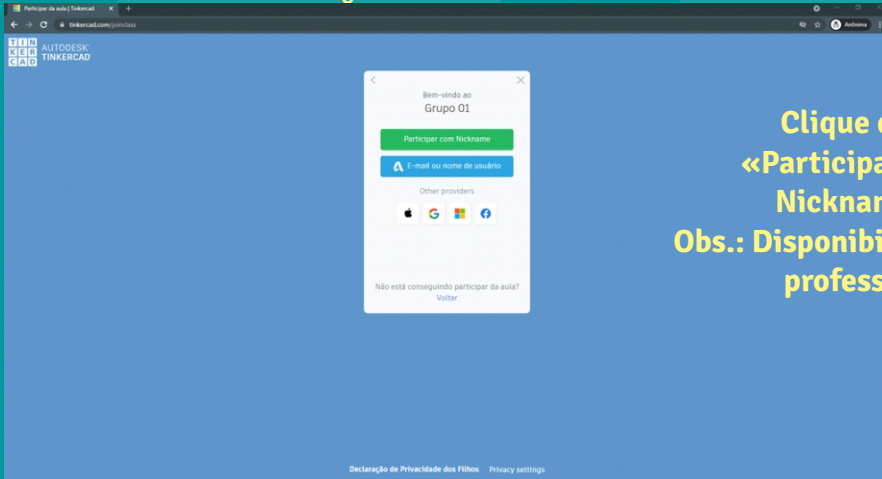
Digitar o **código** disponibilizado pelo **professor**

Fonte: Tinkercad (com adaptações).

Ambiente Virtual Tinkercad

TM

Figura 03: Nickname.

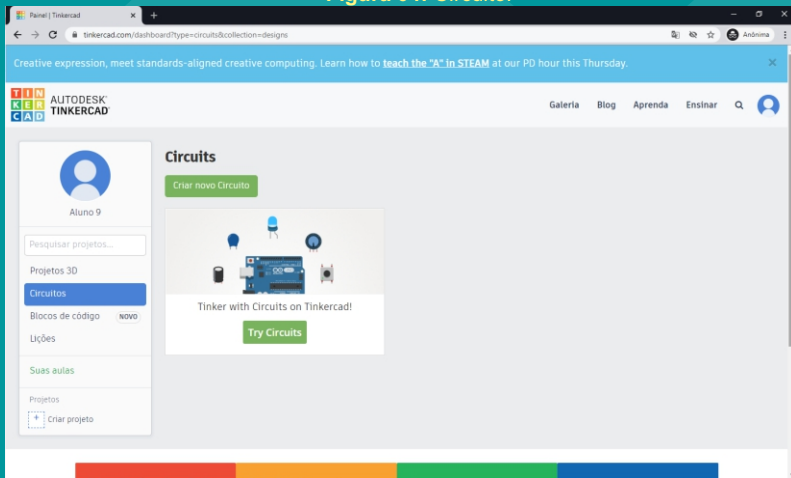


Clique em:
«Participar com Nickname»
Obs.: Disponibilizado pelo professor!

Fonte: Tinkercad (com adaptações).

Ambiente Virtual Tinkercad

Figura 04: Circuito.



Clique em:
«Circuitos»
e depois em
«Criar novo circuito»

Fonte: Tinkercad.

Ambiente Virtual TinkerCad

Figura 05: Componentes.



Clique nos
componentes e
arraste para
Área de Trabalho

Fonte: Tinkercad (com adaptações).

Ambiente Virtual TinkerCad

TM

Figura 06: Salvar arquivo.

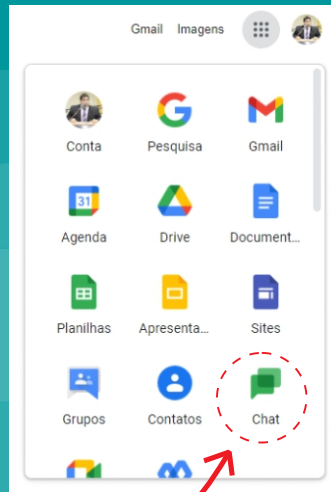
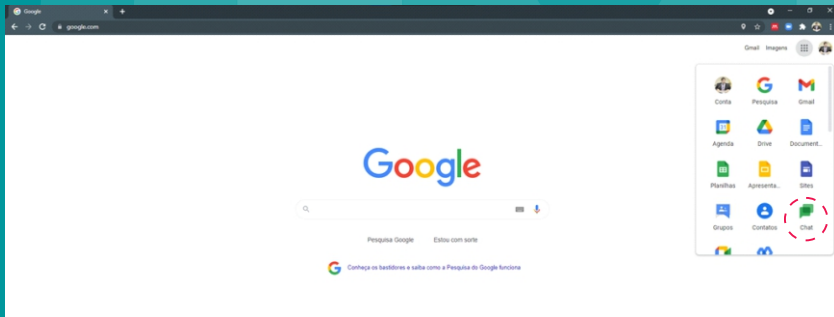


Clique em «Tremendous Fulffy» e renomeie a atividade. Ex.: «Atividade 01»

Fonte: Tinkercad (com adaptações).

Google Chat

1. Clicar no *Google Apps*
2. Clicar em *Chat*



Google Chat

PROFEPT - Grupo 2 - Chat

https://mail.google.com/chat/u/0/#chat/space/AAAAnujWdcs

Chat

Pesquisar no chat e em salas

Ativo

Chat

Salas

PROFEPT - Grupo 2

PROFEPT - Grupo 1

PROFEPT - Grupo 2

1 participante

Chat Arquivos Tarefas

Você criou esta sala nesta data: terça-feira, 30 de mar.

Adicionar pessoas/bots Compartilhar um arquivo Atribuir tarefas

O HISTÓRICO ESTÁ ATIVADO

As mensagens enviadas com o histórico ativado são salvas

Meet

Nova reunião

Minhas reuniões

p histórico está ativado



Encontro 02



ATIVIDADES

1. ASSOCIAÇÃO DE RESISTORES EM SÉRIE
2. ASSOCIAÇÃO DE RESISTORES EM PARALELO
3. CALCULANDO O VALOR DO RESISTOR PARA LIGAR O LED
4. PISCA LED COM ARDUINO
5. SEMÁFORO
6. LENDO O VALOR DE UM POTENCIÔMETRO - POT
7. REGULANDO O BRILHO DO LED POR MEIO DO POT
8. LENDO O VALOR DE UM LDR
9. REGULANDO O BRILHO DO LED POR MEIO DO LDR

RESISTORES ELÉTRICOS

Resistância elétrica é a capacidade de um objeto de opor-se à passagem de corrente elétrica, quando submetido a uma diferença de potencial. A figura abaixo apresenta a simbologia elétrica do resistor no padrão europeu e americano, respectivamente.

Figura – Simbologia elétrica do Resistor.

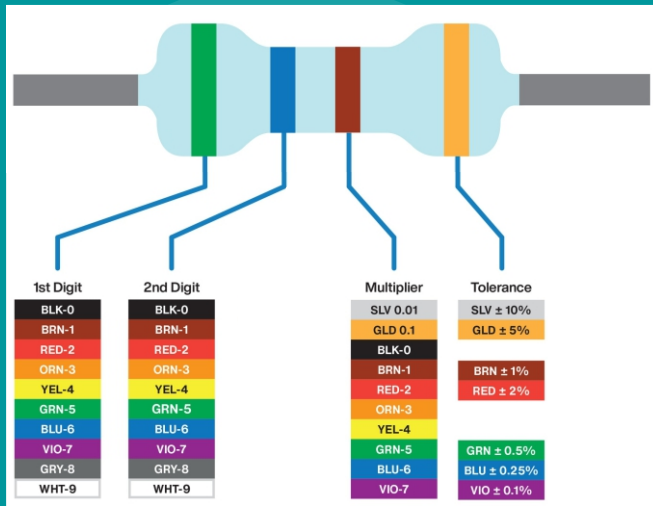


Fonte: Autoria própria.

No Sistema Internacional (SI) a unidade de medida da resistência elétrica é o ohm (Ω).

$$U = 1V \text{ (volt)} \quad \text{e} \quad R = \frac{U}{i}, \text{ temos } R = \frac{1V}{1A} = 1\Omega \text{ (ohm)}$$
$$i = 1A \text{ (ampère)}$$

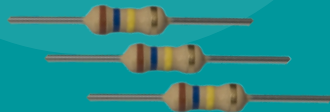
RESISTORES ELÉTRICOS



- 1ª cor: Verde - Primeiro algarismo = 5
2ª cor: Azul - Segundo algarismo = 6
3ª cor: Marrom - Valor do expoente da potência de base 10 = 10^1
4ª cor: Dourada - Tolerância de 5%

Valor da resistência: 560 Ω .

O valor da resistência pode variar de 532 Ω a 588 Ω .

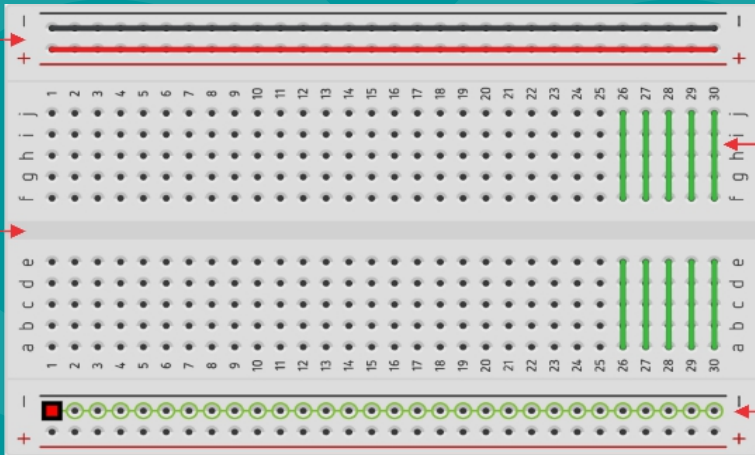


Fonte: [Adaptado] <https://br.mouser.com/technical-resources/conversion-calculators/resistor-color-code-calculator>

PROTOBOARD

TM

Figura – Visão do protoboard.



Alimentação
Barramentos
horizontais

Área de Trabalho
Barramentos verticais

Isolamento

Furos ou Pontos
Furo para encaixe dos
componentes

ATIVIDADE 01

Associação de resistores em série

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD, CALCULE:

1.  $\rightarrow R_{eq} = R_1 + R_2 = 330\Omega + 220\Omega = 550\Omega$

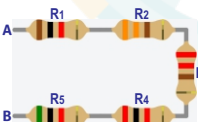
2.  $\rightarrow R_{eq} =$

3.  $\rightarrow R_{eq} =$

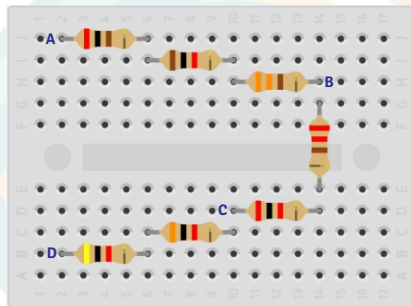
4.  $\rightarrow R_{eq} =$

5.  $\rightarrow R_{eq} =$

6.  $\rightarrow R_{eq} =$

7.  $\rightarrow R_{eq} =$

8.



Utilizando o recurso,

calcule:



A ~ B: _____

B ~ C: _____

C ~ D: _____

A ~ C: _____

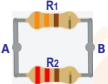
B ~ D: _____

A ~ D: _____

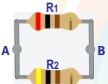

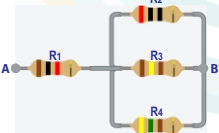
ATIVIDADE 02

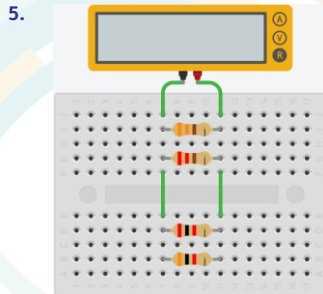
Associação de resistores em paralelo

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD, CALCULE:

- 

$$1/R_{eq} = 1/R_1 + 1/R_2 = 1/330 + 1/220 = 132\Omega$$
ou, no caso de cálculo com somente dois resistores:

$$R_{eq} = (R_1 \times R_2) / (R_1 + R_2) = (330 \times 220) / (330 + 220) = 132\Omega$$
- 
- 
- 



LED - Light Emitter Diode

Scopacasa (2004, p. 5) aponta que o LED é um componente do tipo bipolar, ou seja, tem um terminal chamado ânodo e outro, chamado cátodo. Dependendo de como for polarizado, permite ou não a passagem da corrente elétrica e, conseqüentemente, a geração ou não de luz.

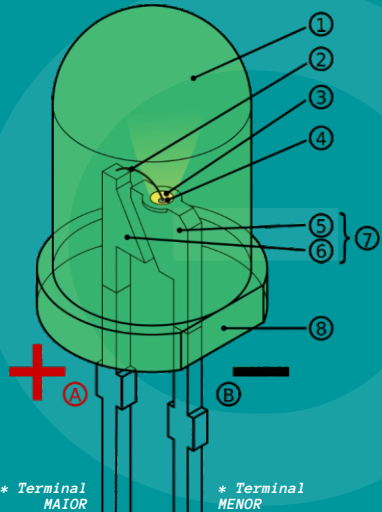
Figura – Simbologia elétrica do LED.



Fonte: Autoria própria.

A função do diodo nos circuitos eletrônicos é permitir a passagem da corrente elétrica em apenas uma direção. Caso a corrente tente retornar o diodo impede que ela o faça (MCROBERTS, 2011, p.49).

LED - Light Emitter Diode



* Terminal
MAIOR

* Terminal
MENOR

* Quando ainda não tiveram seus terminais cortados.
O terminal negativo também pode ser indicado
pelo chanfrado, conforme rótulo 8.

Rótulo

Descrição

A	Ânodo
B	Cátodo
1	Lente Epóxi / Encapsulamento
2	Fio condutor
3	Cavidade reflexiva
4	Semicondutor
5	Bigorna
6	Plaquetária
7	Estrutura
8	Borda plana

LED - Light Emitter Diode

TM

Tabela – Relação: cor x tensão x corrente do LED

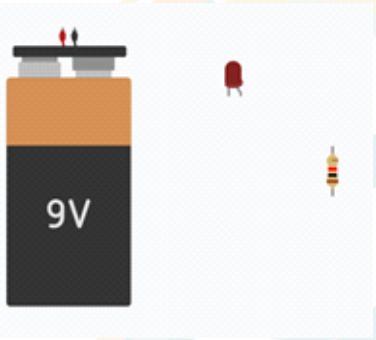
Cor do LED	Faixa de tensão	Corrente máxima
Vermelho	1,8 V - 2,0 V	20 mA
Amarelo	1,8 V - 2,0 V	20 mA
Laranja	1,8 V - 2,0 V	20 mA
Verde	2,0 V - 2,5 V	20 mA
Azul	2,5 V - 3,0 V	20 mA
Branco	2,5 V - 3,0 V	20 mA

ATIVIDADE 03

Calculando o valor do resistor para ligar o LED

O OBJETIVO DESTA ATIVIDADE É LIGAR O LED.

PARA QUE ISTO ACONTEÇA VOCÊ DEVE DETERMINAR O VALOR DO RESISTOR E INTERLIGAR OS COMPONENTES.



A. QUAL O VALOR DO RESISTOR?



B. A SIMULAÇÃO ACONTECEU CORRETAMENTE NA PRIMEIRA TENTATIVA? CASO A RESPOSTA SEJA NEGATIVA, DESCREVER O QUE FOI FEITO PARA CORRIGIR O PROBLEMA.

C. COM O AUXÍLIO DO MULTÍMETRO DIGITAL, INFORME O VALOR DA TENSÃO NOS TERMINAIS DO RESISTOR.



D. INFORME TAMBÉM O VALOR DA TENSÃO NOS TERMINAIS DO LED.



E. TESTE COM RESISTORES DE 10 Ω, 220 Ω e 10 MΩ. DESCREVA O QUE ACONTECE.

- 10 Ω -
- 220 Ω -
- 10 MΩ -



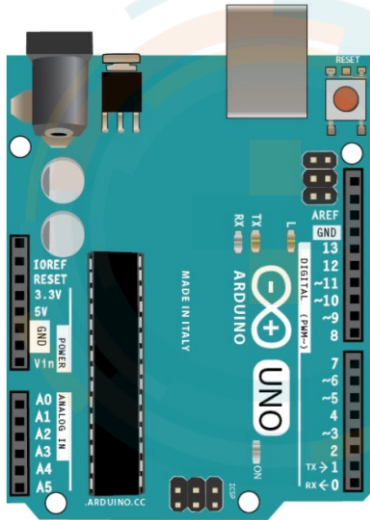
Encontro 03



ATIVIDADES

1. ASSOCIAÇÃO DE RESISTORES EM SÉRIE
2. ASSOCIAÇÃO DE RESISTORES EM PARALELO
3. CALCULANDO O VALOR DO RESISTOR PARA LIGAR O LED
4. PISCA LED COM ARDUINO
5. SEMÁFORO
6. LENDO O VALOR DE UM POTENCIÔMETRO - POT
7. REGULANDO O BRILHO DO LED POR MEIO DO POTENCIÔMETRO
8. LENDO O VALOR DE UM LDR
9. REGULANDO O BRILHO DO LED POR MEIO DO LDR

ARDUINO



Arduino é o que chamamos de uma plataforma de computação física embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de hardware e software (MCROBERTS, 2011, p. 20).

ARDUINO - Modelos



Arduino Uno



Arduino Leonardo



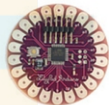
Arduino Ethernet



Arduino Pro



Arduino Mega 2560



Arduino LilyPad



Arduino BT



Arduino Nano



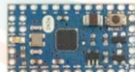
Arduino Mega ADK



Arduino Fio



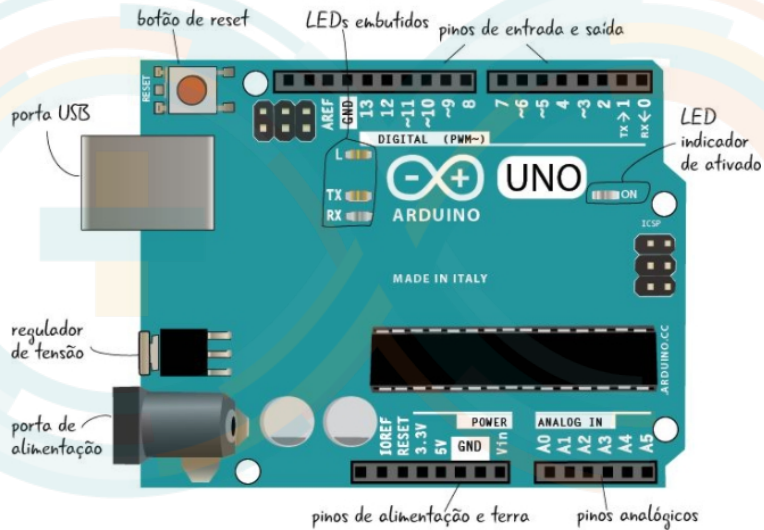
USB/Serial Light Adapter



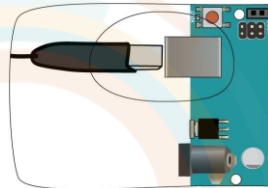
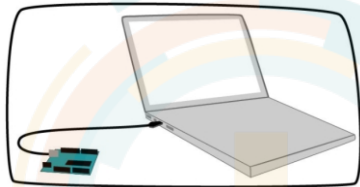
Arduino Mini

Fonte:
<http://ardufc.blogspot.com/2012/10/modelos-do-arduino.html>

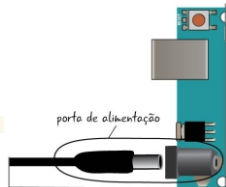
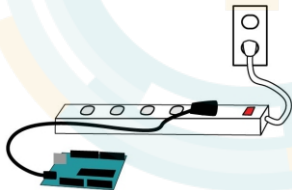
ARDUINO UNO



ARDUINO UNO - Alimentação



Fonte: (CULKIN E HAGAN, 2018, p. 32 e 33).



Fonte: (CULKIN E HAGAN, 2018, p. 35 e 36).

ARDUINO UNO - Alimentação

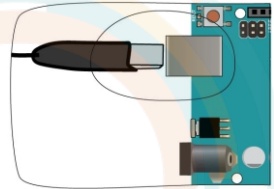
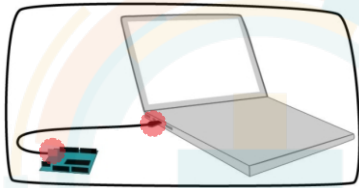
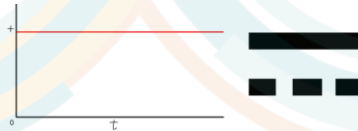


Figura: Forma de onda da corrente contínua e símbolo DC.



Fonte: (CULKIN e HAGAN, 2018, p. 149).

ARDUINO UNO - Alimentação

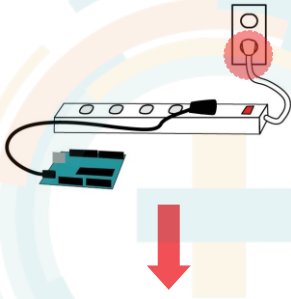
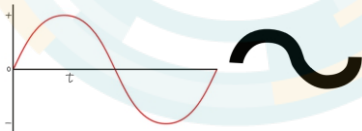


Figura: Forma de onda da corrente alternada e símbolo AC.



Fonte: (CULKIN e HAGAN, 2018, p. 149).

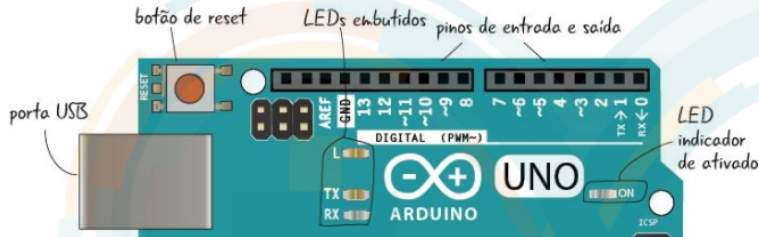


Figura: Forma de onda da corrente contínua e símbolo DC.



Fonte: (CULKIN e HAGAN, 2018, p. 149).

ARDUINO UNO - Pinos de entrada e saída digital



Níveis de Tensão Elétrica CMOS



As entradas digitais só podem assumir dois estados:

Nível lógico **alto** - HIGH: 5 volts.

Nível lógico **baixo** - LOW: 0 volts.

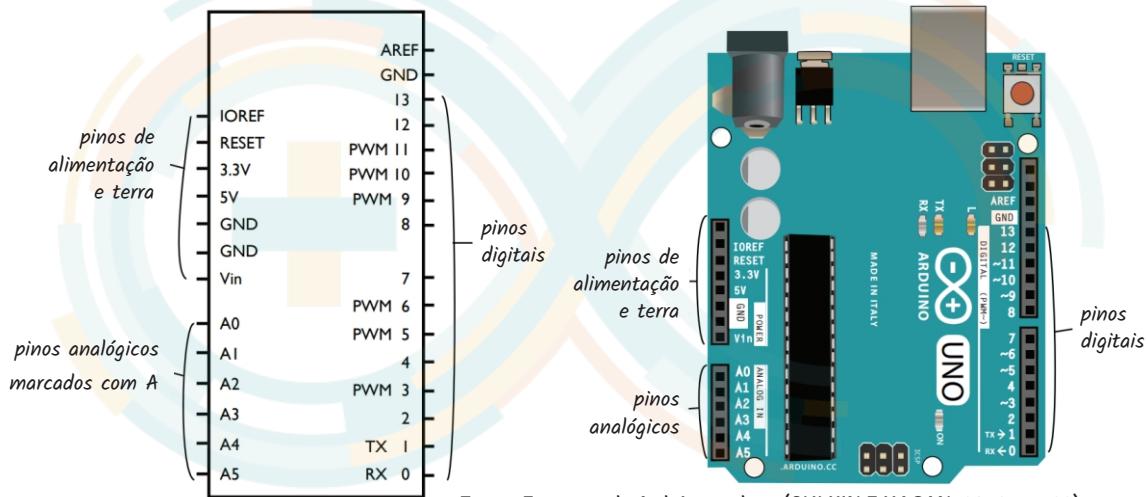


Fonte:

<http://www.bosontreinamentos.com.br/electronica/el-electronica-digital/o-que-sao-portas-logicas/>

Fonte: <https://portal.vidadesilicio.com.br/entradas-e-saidas-digitais/>

ARDUINO UNO - Esquema



Fonte: Esquema do Arduino e placa (CULKIN E HAGAN, 2018, p. 122).

Noções básicas de programação em C++

*A linguagem do Arduino é baseada em C/C++.
Conforme a Documentação de Referência da
Linguagem Arduino (ARDUINO.CC, 2021), a linguagem
de programação do Arduino pode ser dividida em três
partes principais: estruturas, valores (variáveis e
constantes) e funções.*

Alguns elementos de sintaxe

Quadro – Operadores aritméticos.

Operação	Descrição
+	adição
-	subtração
*	multiplicação
/	divisão
%	resto da divisão

Fonte: Autoria própria.

Quadro – Operadores de comparação.

Operação	Descrição
==	igual
!=	diferente
<	menor
>	maior
<=	menor igual
>=	maior igual

Fonte: Autoria própria.

Quadro – Operadores booleanos.

Operação	Descrição
&&	e
	ou
!	não

Fonte: Autoria própria.

Operadores aritméticos são usados para desenvolver operações matemáticas. No Quadro 4 apresentamos uma lista resumida de operações aritméticas do C/C++.

Operadores relacionais, ou de comparação, realizam comparações entre as variáveis.

Estruturas de controle

As instruções de controle, ou condicionais, controlam o fluxo do seu código. A instrução *if* (*se*) é um exemplo de uma estrutura de controle, cujo propósito é verificar se determinada condição foi ou não atingida. Caso a condição seja verdadeira, a instrução executa o código dentro de seu bloco de código (MCROBERTS, 2011, p.62). Se utilizarmos a instrução *if* junto com *else* (*senão*), no caso da sentença *if* for falsa, serão executados os comandos entre as chaves do *else*.

```
1  if (condição1) {  
2      comando a;  
3  }  
4  else if (condição2) {  
5      comando b;  
6  }  
7  else {  
8      comando c;  
9  }
```

Podemos analisar o código acima da seguinte forma: se (*if*) a determinada *condição1* for satisfeita, execute os comandos entre chaves (*comando a*), caso contrário se (*else if*), ou seja, se a *condição1* não for satisfeita, verifica a *condição2*. Se a *condição2* for satisfeita, executa o *comando b*, caso contrário será executado o *comando c*.

Estruturas de repetição

A estrutura de repetição permite que uma sequência de instruções seja repetida quantas vezes o programador desejar.

A função `loop()` já faz algo semelhante.

Os principais comandos de repetição são o `for()` e o `while()`.

Variáveis

Quadro – Tipos de variáveis.

Tipo de variável	Descrição	Exemplo de código
byte	Um número inteiro de 8 bits com um intervalo de 0 a 255.	byte <u>myVariable</u> = 215;
int	Abreviação de número inteiro; um número inteiro de 16 bits com um intervalo 32.767 a -32.768.	<u>int myVariable</u> = 32.000;
long	Um número inteiro logo armazenado como um valor de 32 bits com um intervalo de 2.147.483.647 a -2.147.483.648.	<u>long myVariable</u> = 320.000
float	Um número que tem um ponto decimal, armazenado como um valor de 32 bits com um intervalo de 3,4028325E + 38 a -3,4028325E + 38	<u>float myVariable</u> = 3,14;
unsigned int	Um número inteiro sem quaisquer valores negativos, armazenado como um valor de 16 bits com um intervalo de 0 a 65.545.	<u>unsigned int myVariable</u> = 45.000;
unsigned long	Um número inteiro sem quaisquer valores negativos, armazenado como um valor de 32 bits com um intervalo de 0 a 4.294.967.296.	<u>unsigned long myVariable</u> = 569.124;
word	Um número sem sinal, armazenado como um valor de 16 bits com um intervalo de 0 a 65.545 (o mesmo que um número inteiro sem sinal).	word <u>myVariable</u> = 53.000;
boolean	Uma variável booleana que detém um dos dois valores: verdadeiro ou falso.	<u>boolean myVariable</u> = true;
char	Um único caractere ASCII, armazenado como um número de 8 bits, ou byte. Isso permite que os caracteres possuam funções aritméticas executadas neles. O caractere 'c' é armazenado como o número 99, então você pode adicionar 2 a ele e obter 'e'.	char <u>myVariable</u> = 'c'; char <u>myVariable</u> = 99;

Fonte: EVANS, NOBLE e HOCHENBAUM, 2013, p. 398 – 399.

McRoberts (2011, pg. 42) define uma variável como um local que podemos armazenar dados. Na linguagem C++ o nome da variável deve iniciar com uma letra e o restante pode ser formado por letras, números e underscores (underline).

Funções

Em linguagens de programação, funções são um conjunto de instruções que podem ser chamadas em qualquer parte do programa. Com esses conjuntos de instruções, a função pode ou retornar um valor ou executar uma determinada tarefa.

O Arduino possui algumas funções predefinidas: `pinMode()`, `digitalWrite()`, `delay()`, entre outras. Também podem ser criadas pelo programador, geralmente antes da função `setup()`.

ATIVIDADE 04

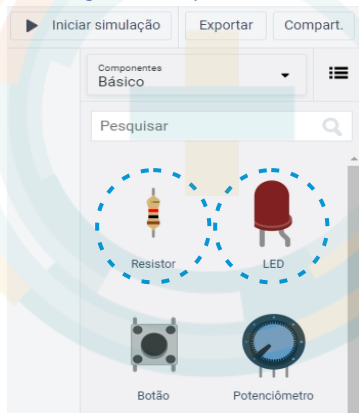
Pisca LED com Arduino

Objetivo: Acender um **LED** por um segundo e depois apagar por um segundo, repetidamente.

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD

1º
PASSO

Figura 1: Componentes.



ADICIONAR OS COMPONENTES

CLIQUE NOS COMPONENTES E ARRASTE PARA ÁREA DE TRABALHO

- ARDUINO
- LED
- RESISTOR

SABENDO QUE A TENSÃO DE SAÍDA NO PINO 13 DO ARDUINO É DE 5V, DETERMINE O VALOR DO RESISTOR.



Fonte: os autores.

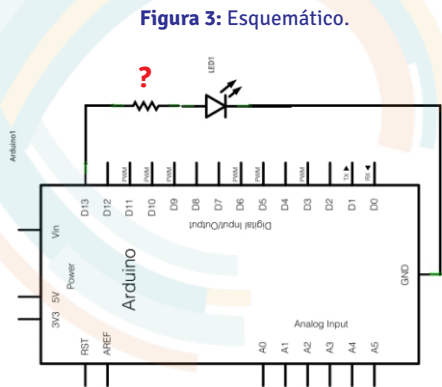
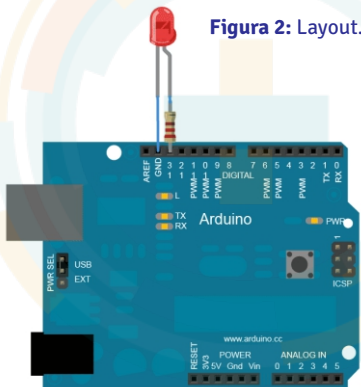
ATIVIDADE 04

Pisca LED com Arduino

Objetivo: Acender um **LED** por um segundo e depois apagar por um segundo, repetidamente.

2º
PASSO

INTERLIGAR OS COMPONENTES



Fonte: <https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink>

ATIVIDADE 04

Pisca LED com Arduino

Objetivo: Acender um **LED** por um segundo e depois apagar por um segundo, repetidamente.

3º
PASSO

CÓDIGO

CLICAR →

 Código

 Iniciar simulação

Exportar

Compart.

```

01 void setup () {
02     pinMode (LED_BUILTIN, OUTPUT);
03 }
04
05 void loop () {
06     digitalWrite (LED_BUILTIN, HIGH);
07     delay(1000);
08     digitalWrite (LED_BUILTIN, LOW);
09     delay(1000);
10 }
    
```



DIAGRAMA DE
TEMPO

4º
PASSO

INICIAR SIMULAÇÃO

AUMENTE/DIMINUA O DELAY E OBSERVE O QUE ACONTECE!

ATIVIDADE 04

Pisca LED com Arduino

ANALISANDO O CÓDIGO

Linha 1. Executa a função `setup()`. Possui apenas uma instrução, `pinMode()`, que diz ao Arduino que desejamos definir o modo de um de seus pinos como saída (OUTPUT).

Linha 2. Inicializando o pino `LED_BUILTIN` como um pino de saída;

Linha 3. Término de execução da função `setup()`

Linha 5. Executa a função `loop()`. Esta é a função principal do programa e executa continuamente enquanto o Arduino estiver ligado

Linha 6. Atribui nível lógico alto para o pino de saída, ou seja, fornece uma tensão de 5V no pino 13, ligando o LED.

ATIVIDADE 04

Pisca LED com Arduino

ANALISANDO O CÓDIGO

Linhas 7 e 9. Essa instrução diz ao Arduino para esperar 1.000 milissegundos, ou seja, 1 segundo.

Linha 8. Atribui nível lógico baixo para o pino de saída, ou seja, fornece uma tensão de 0V no pino 13, desligando o LED.

Linha 10. Término de execução da função `loop()`. Reinicia desde o princípio e assim infinitamente até que o Arduino seja desligado ou o botão `Reset` pressionado.

ATIVIDADE 05

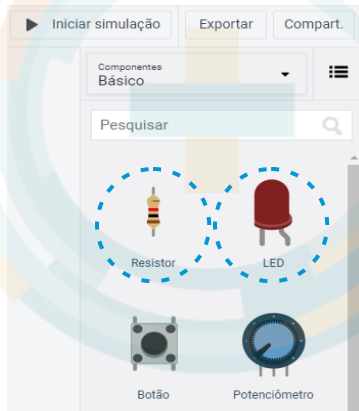
Semáforo

Objetivo: Simular o funcionamento de um semáforo.

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD

1º
PASSO

Figura 1: Componentes.

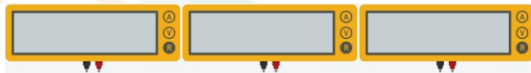


ADICIONAR OS COMPONENTES

CLIQUE NOS COMPONENTES E ARRASTE PARA ÁREA DE TRABALHO

- 1 ARDUINO
- 3 LED's
- 3 RESISTORES

SABENDO QUE A TENSÃO DE SAÍDA NOS PINOS DO ARDUINO É DE 5V, DETERMINE O VALOR DE CADA RESISTOR.



ATIVIDADE 05

Semáforo

ATIVIDADE
EM GRUPO

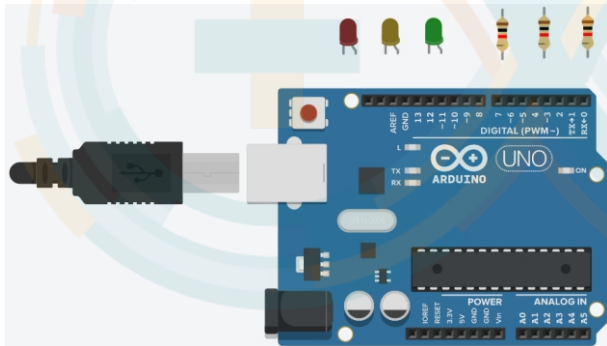
Objetivo: Simular o funcionamento de um semáforo.

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD

2º
PASSO

INTERLIGAR OS COMPONENTES

Figura 2: Layout.



DESENVOLVA O
ESQUEMÁTICO

Enviar no formato de imagem
no Google Chat

ATIVIDADE 05

Semáforo

ATIVIDADE
EM GRUPO

Objetivo: Simular o funcionamento de um semáforo.

3º
PASSO

CODIFICAR

CLICAR →

Código

Iniciar simulação

Exportar

Compart.

Para desenvolver o código, considere os seguintes parâmetros:

Semáforo **Vermelho**: 2 segundos

Semáforo **Amarelo**: 1 segundo

Semáforo **Verde**: 2 segundos



4º
PASSO

INICIAR SIMULAÇÃO

Observar o acionamento dos LEDs e os respectivos intervalos de tempo.

ATIVIDADE 05

Semáforo

5º
PASSO

DESENVOLVA O DIAGRAMA DE TEMPO DA TENSÃO (V) x TEMPO (s)

O estudante deve **representar** no gráfico a **variação da tensão** no decorrer do tempo, conforme os parâmetros.

Exemplo: no intervalo de tempo de 1 a 3 segundos, a **tensão** do semáforo vermelho está em **5V** enquanto os demais estão em **0V**.

Diagrama de Tempo - Semáforo Vermelho

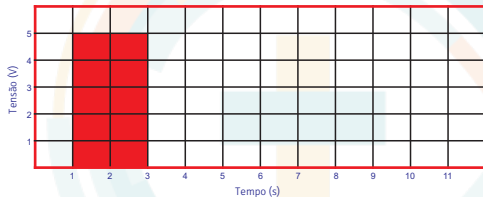


Diagrama de Tempo - Semáforo Verde

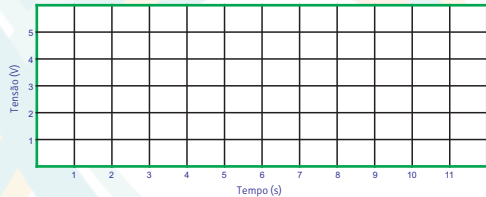
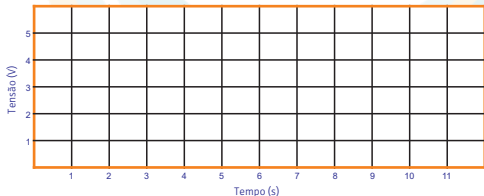


Diagrama de Tempo - Semáforo Amarelo



Grupo 01

<https://bit.ly/2ZemuWq>

Grupo 02

<https://bit.ly/3AsJZIT>

Grupo 03

<https://bit.ly/2Z8TvTH>



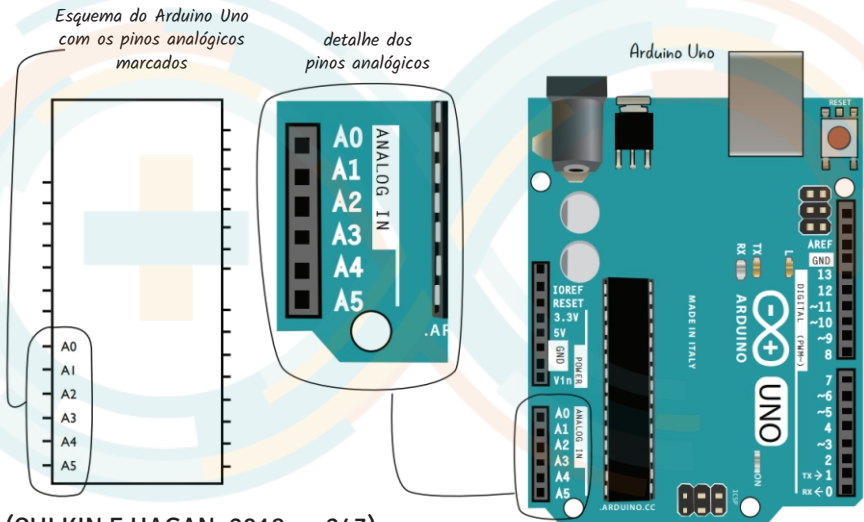
Encontro 04



ATIVIDADES

1. ASSOCIAÇÃO DE RESISTORES EM SÉRIE
2. ASSOCIAÇÃO DE RESISTORES EM PARALELO
3. CALCULANDO O VALOR DO RESISTOR PARA LIGAR O LED
4. PISCA LED COM ARDUINO
5. SEMÁFORO
6. LENDO O VALOR DE UM POTENCIÔMETRO - POT
7. REGULANDO O BRILHO DO LED POR MEIO DO POTENCIÔMETRO
8. LENDO O VALOR DE UM LDR
9. REGULANDO O BRILHO DO LED POR MEIO DO LDR

ARDUINO UNO - Pinos de entrada analógicos

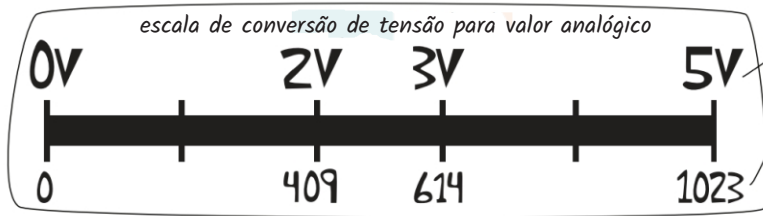


Fonte: (CULKIN E HAGAN, 2018, p. 243).

ARDUINO UNO - Conversor Analógico Digital

CONVERSOR AD DE 10 Bits

O sinal de entrada é transformado em valores entre 0 e 1023.



valores de tensão

valores analógicos equivalentes

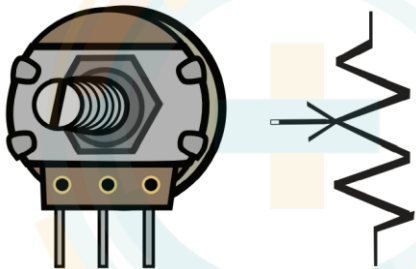
000000000 (0)	- 0 V
000000001 (1)	- 5 mV
000000010 (2)	- 10 mV
....	
00110011001 (409)	- 2 V
....	
1001100110 (614)	- 3 V
....	
1111111111 (1023)	- 5 V

Fonte: (CULKIN E HAGAN, 2018, p. 258).

ARDUINO UNO - Pinos de entrada analógicos

Nome da função	O que ela faz	Argumentos da função	Intervalo de valores	
digitalRead()	Lê o valor de um pino de entrada digital	O número do pino a ser lido	Lê os valores 1 ou 0 do pino	<pre>digitalRead(pino) val = digitalRead(7);</pre>
digitalWrite()	Escreve um valor para um pino de saída	O número do pino a ser escrito e o valor	Escreve 1 ou 0 no pino	<pre>digitalWrite(pino, valor) digitalWrite(13, HIGH);</pre>
analogRead()	Lê o valor de um pino de entrada analógica	O número do pino a ser lido	Lê um número inteiro entre 0 e 1023	<pre>analogRead(pino) val = analogRead(analogPin);</pre>
analogWrite()	Escreve um valor para um pino de saída com PWM	O número do pino a ser escrito e o valor	Escreve um inteiro entre 0 e 255 no pino, o que resulta em um valor de tensão entre 0 e 5 volts	<pre>analogWrite(pino, valor)</pre>

Potenciômetro



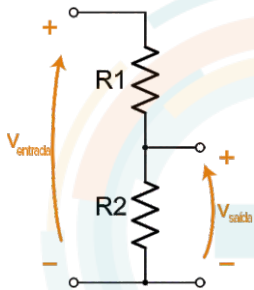
Um potenciômetro é um tipo de resistor variável, o que significa que o valor da resistência pode mudar. De acordo com a variação do seletor, é possível aumentar ou diminuir o valor da resistência dependendo da direção e extensão do giro.

Fonte: (CULKIN E HAGAN, 2018, p. 244).

Potenciômetro - Aplicações

- 1. REGULAR CONTRASTE DO DISPLAY LCD**
- 2. REGULAR CONTRASTE DAS TV's (MAIS ANTIGAS)**
- 3. REGULAR VOLUME AMPLIFICADOR DE SOM**
- 4. DIVISOR DE TENSÃO**
- 5. ENTRE OUTRAS APLICAÇÕES...**

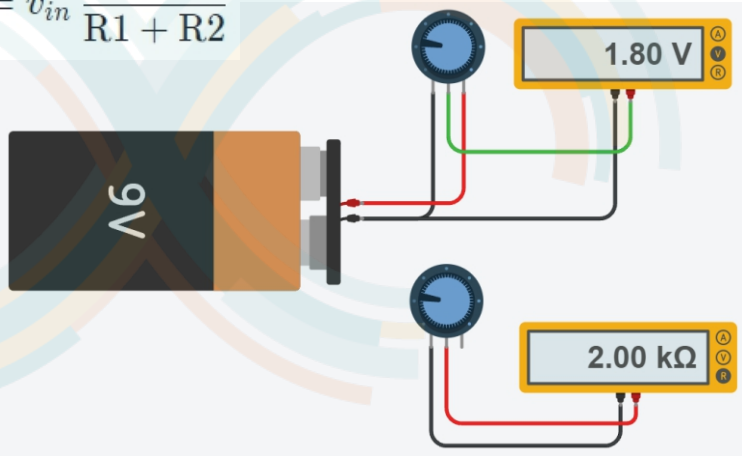
Potenciômetro - Aplicações



$$v_{out} = v_{in} \frac{R2}{R1 + R2}$$

Fonte:

<https://pt.khanacademy.org/science/electrical-engineering/ee-circuit-analysis-topic/ee-resistor-circuits/a/ee-voltage-divider>



ATIVIDADE 06

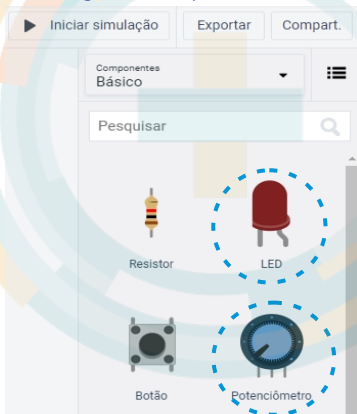
Potenciômetro

Objetivo: Realizar a leitura do **potenciômetro** por meio da por analógica do Arduino.

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD

1º
PASSO

Figura 1: Componentes.



ADICIONAR OS COMPONENTES

CLIQUE NOS COMPONENTES E ARRASTE PARA ÁREA DE TRABALHO

- ARDUINO
- POTENCIÔMETRO 10 KΩ

Fonte: os autores.

ATIVIDADE 06

Potenciômetro

Objetivo: Realizar a leitura do **potenciômetro** por meio da porta analógica do Arduino.

2º
PASSO

INTERLIGAR OS COMPONENTES

Figura 2: Layout.

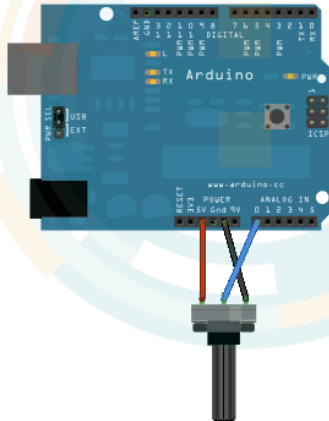
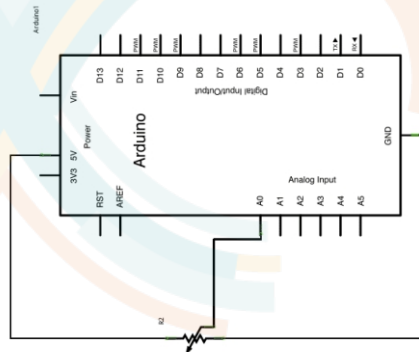


Figura 3: Esquemático.



Fonte: <https://www.arduino.cc/en/Tutorial/BuiltInExamples/AnalogReadSerial>

ATIVIDADE 06

Potenciômetro

Objetivo: Realizar a leitura do **potenciômetro** por meio da porta analógica do Arduino.

3º
PASSO

CÓDIGO

CLICAR →

Código

Iniciar simulação

Exportar

Compart.

```
// a rotina de configuração é executada uma vez quando você pressiona reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}
//a rotina de loop é executada indefinidamente:
void loop() {
  // leia a entrada no pino analógico 0:
  int sensorValue = analogRead(A0);
  // imprime o valor que você leu:
  Serial.println(sensorValue);
  delay(2); // atraso entre as leituras para estabilidade
}
```

4º
PASSO

INICIAR SIMULAÇÃO

VERIFICAR SERIAL MONITOR

ANALISANDO O CÓDIGO

Linha 1. Executa a função `setup()`.

Linha 2. Inicia a comunicação serial entre a placa do Arduino e o computador a 9600 bits por segundo;

Linha 3. Término de execução da função `setup()`.

Linha 5. Executa a função `loop()`. É a função principal do programa e executa continuamente enquanto o Arduino estiver ligado.

Linha 6. Determina uma variável, `sensorValue`, do tipo inteira para armazenar o valor lido do potenciômetro (valor entre 0 a 1023), que está conectado no pino A0.

Linha 7. Essa instrução imprime as informações na janela do serial monitor.

Linha 8. Aguarda 1milissegundo.

Linha 9. Término de execução da função `loop()`. Reinicia desde o princípio e assim infinitamente até que o Arduino seja desligado ou o botão Reset pressionado.

LDR - Light Dependent Resistor

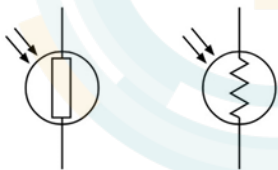
São dispositivos cuja resistência varia com a intensidade da luz que incide sobre eles.



Fonte:

<https://commons.wikimedia.org/wiki/File:LDR-gs-2012.jpg>

A resistência diminui com o aumento da intensidade da luz.



Fonte:

https://www.usinainfo.com.br/resistor/ldr-sensor-de-luminosidade-5mm-2982.html?search_query=ldr&results=19

Conforme a incidência de luz diminui sobre o LDR, a resistência aumenta.

LDR - Aplicações

1. ACENDIMENTO AUTOMÁTICO DE LUZ - ILUMINAÇÃO PÚBLICA
2. ROBÔS SEGUIDORES DE LINHA
3. AGRICULTURA
4. MOVIMENTAÇÃO DE PLACAS FOTOVOLTAICAS
5. ENTRE OUTRAS APLICAÇÕES...

ATIVIDADE 08

Lendo o valor de um LDR

Objetivo: Realizar a leitura do LDR por meio da porta analógica do Arduino.

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD

1º
PASSO

Figura 1: Componentes.



Fonte: Elaborado pelo autor.

ADICIONAR OS COMPONENTES

CLIQUE NOS COMPONENTES E ARRASTE PARA ÁREA DE TRABALHO

- ARDUINO
- RESISTOR 10 KΩ
- LDR ou Fotoresistor

ATIVIDADE 08

Lendo o valor de um LDR

Objetivo: Realizar a leitura do LDR por meio da porta analógica do Arduino.

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD

2º
PASSO

INTERLIGAR OS COMPONENTES

Figura 2: Layout.

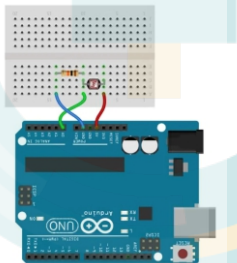
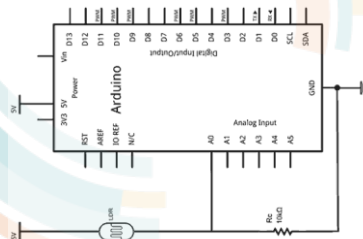


Figura 3: Esquemático.



Fonte:
<https://create.arduino.cc/projecthub/krivanja/working-with-light-dependent-resistor-ldr-1ded4f>

Fonte: <https://www.luisllamas.es/medir-nivel-luz-con-arduino-y-fotoresistencia-ldr/>

ATIVIDADE 08

Lendo o valor de um LDR

Objetivo: Realizar a leitura do LDR por meio da porta analógica do Arduino.

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD

3º
PASSO

CÓDIGO

```
01 int LDR = A0;
02 int valueLDR = 0;
03
04 void setup() {
05     Serial.begin(9600);
06 }
07
08 void loop() {
09     valueLDR = analogRead(LDR);
10     Serial.print("LDR Value is: ");
11     Serial.println(input_val);
12     delay(1000);
13 }
```

CLICAR →

Código

Iniciar simulação

Exportar

Compart.

4º
PASSO

INICIAR SIMULAÇÃO

Observar o que acontece no Serial Monitor quando muda a incidência de luz no ambiente.

ATIVIDADE 08

Lendo o valor de um LDR

ANALISANDO O CÓDIGO

Linha 1. Pino de entrada analógica ao qual o LDR está conectado.

Linha 2. Determina uma variável, `valueLDR`, do tipo inteira para armazenar o valor lido do LDR (valor entre 0 a 1023), que está conectado no pino A0. Inicialmente é atribuído o valor zero.

Linha 4. Executa a função `setup()`. Possui apenas uma instrução, `Serial.begin(9600)`.

Linha 5. Inicia a comunicação serial entre a placa do Arduino e o computador a 9600 bits por segundo.

Linha 6. Término de execução da função `setup()`.

Linha 8. Executa a função `loop()`. É a função principal do programa e executa continuamente enquanto o Arduino estiver ligado.

ATIVIDADE 08

Lendo o valor de um LDR

ANALISANDO O CÓDIGO

Linha 9. A variável `valueLDR` recebe o valor lido do LDR.

Linhas 10 e 11. Essas instruções imprimem as informações na janela do serial monitor.

Linha 12. Aguarda 1 segundo.

Linha 13. Término de execução da função `loop()`. Reinicia desde o princípio e assim infinitamente até que o Arduino seja desligado ou o botão Reset pressionado.

ATIVIDADE 07

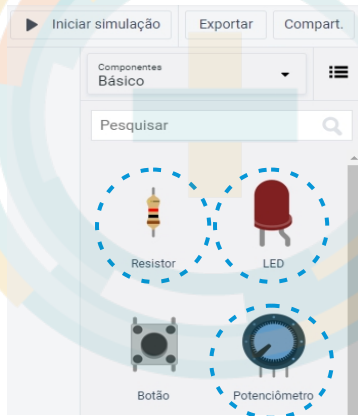
Regulando o brilho do LED por meio do Potenciômetro

Objetivo: Realizar o controle de brilho do LED por meio da variação do potenciômetro.

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD

1º
PASSO

Figura 1: Componentes.



ADICIONAR OS COMPONENTES

CLIQUE NOS COMPONENTES E ARRASTE PARA ÁREA DE TRABALHO

- ARDUINO
- LED
- RESISTOR
- POTENCIÔMETRO 10 KΩ
- PROTOBOARD (Placa de Testes)

Fonte: Elaborado pelo autor.

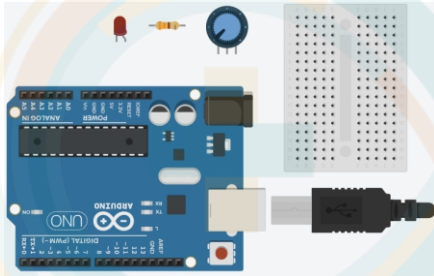
ATIVIDADE 07

Regulando o brilho do LED por meio do Potenciômetro

Objetivo: Realizar o controle de brilho do LED por meio da variação do potenciômetro.

2º
PASSO

INTERLIGAR OS COMPONENTES



DESENVOLVA O
ESQUEMÁTICO

Enviar no formato de imagem
no Google Chat

3º
PASSO

CODIFICAR

CLICAR →

Código

Iniciar simulação

Exportar

Compart.

4º
PASSO

INICIAR SIMULAÇÃO

Variar o potenciômetro e observar o que acontece com o LED.

ATIVIDADE 09

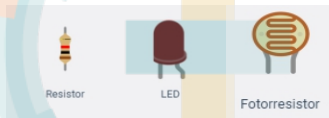
Acionando um LED por meio do LDR

Objetivo: Ligar / Desligar o LED conforme a intensidade luminosa do ambiente.

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD

1º
PASSO

Figura 1: Componentes.



Fonte: Elaborado pelo autor.

ADICIONAR OS COMPONENTES

CLIQUE NOS COMPONENTES E ARRASTE PARA ÁREA DE TRABALHO

- ARDUINO
- LED
- 2 RESISTORES
- LDR
- PROTOBOARD (Placa de Testes)

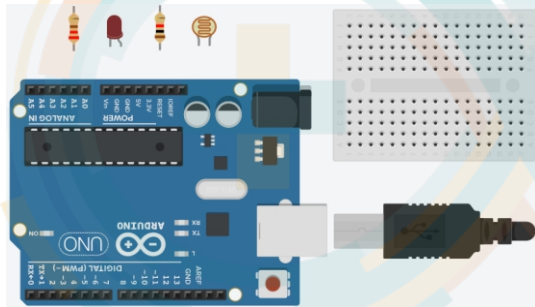
ATIVIDADE 09

Acionando um LED por meio do LDR

Objetivo: Ligar / Desligar o LED conforme a intensidade luminosa do ambiente.

2º
PASSO

INTERLIGAR OS COMPONENTES



DESENVOLVA O
ESQUEMÁTICO

Enviar no formato de imagem
no Google Chat



RESOLUÇÃO

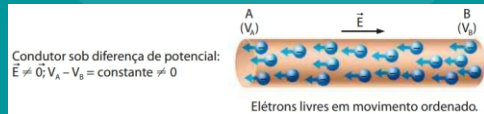
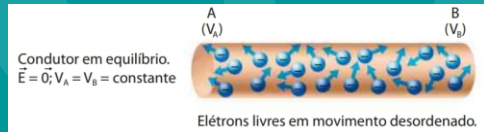


RESOLUÇÃO - AVALIAÇÃO DIAGNÓSTICA

Questão 01 - A corrente elétrica está presente no dia a dia das pessoas, seja no celular, aparelhos eletrônicos, no uso da eletricidade em casa, dentre outros. Assim, conceitue corrente elétrica.

A corrente elétrica é definida como **movimento ordenado de elétrons livres** no interior de um condutor.

[...] os elétrons livres movimentam-se no interior de um condutor sempre no sentido do aumento de potencial, ou seja, do potencial elétrico menor para o maior — na figura, da extremidade B (V_B) para a extremidade A (V_A) (YAMAMOTO e FUKE, 2016, p. 96).



RESOLUÇÃO - AVALIAÇÃO DIAGNÓSTICA

Questão 02 - Alguns componentes elétricos estão presentes no nosso dia a dia mas nem sempre os percebemos. Um exemplo disso são os resistores.

O que são resistores?

Onde eles são frequentemente utilizados?

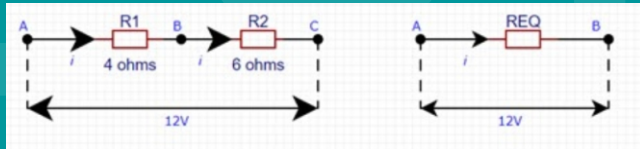
Resistência elétrica é a capacidade de um objeto de opor-se à passagem de corrente elétrica, quando submetido a uma diferença de potencial. Os condutores que têm exclusivamente a função de converter energia elétrica em calor são chamados de resistores (YAMAMOTO e FUKE, 2016, p. 108).

Chuveiro / Secador de cabelo / Ferro de passar / Churrasqueira elétrica / Placas eletrônicas...

TM

RESOLUÇÃO - AVALIAÇÃO DIAGNÓSTICA

Questão 3 - Com base nessas informações, calcule: (a) a resistência equivalente da associação? (b) a intensidade de corrente elétrica em cada resistor:



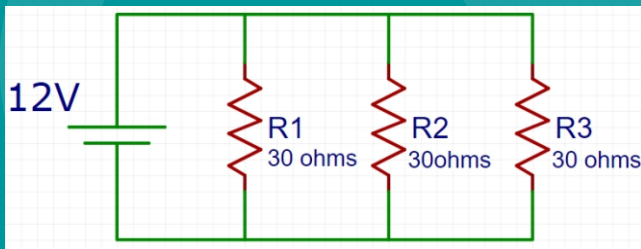
Associação em série

(a) $R_{eq} = R_1 + R_2$
 $R_{eq} = 4 + 6$
 $R_{eq} = 10 \Omega$

(b) $U = R \cdot i$
 $12 = 10 \cdot i$
 $i = 12/10$
 $i = 1,2 \text{ A}$

RESOLUÇÃO - AVALIAÇÃO DIAGNÓSTICA

Questão 4 - (PUC) Três resistores idênticos de $R = 30\Omega$ estão ligados em paralelo com uma bateria de 12V. Pode-se afirmar que a resistência equivalente do circuito é de:



Associação em paralelo

Obs.: Resistores iguais!

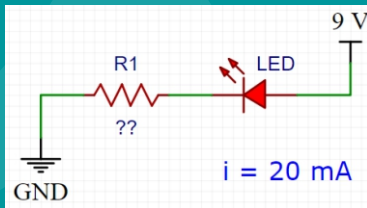
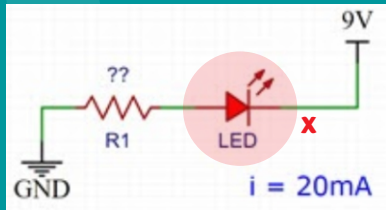
$$R_{eq} = R / 3$$

$$R_{eq} = 30/3$$

$$R_{eq} = 10 \Omega$$

RESOLUÇÃO - AVALIAÇÃO DIAGNÓSTICA

Questão 5 - O circuito abaixo possui uma fonte de alimentação de 9V, um resistor R1 e um LED (light-emitting diode ou diodo emissor de luz). Considerando que o LED necessite de uma tensão de 2V e uma corrente de 20mA para funcionar corretamente, calcule o valor do resistor para acionar esse LED.



Calculando o resistor do LED:

$$R = (U - U_{\text{LED}}) / i$$

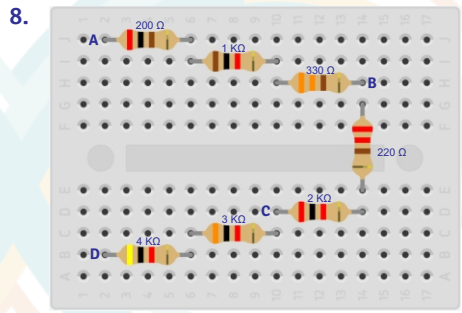
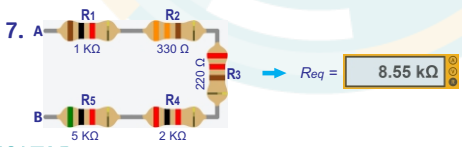
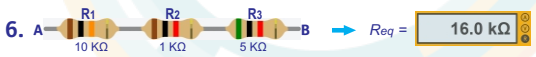
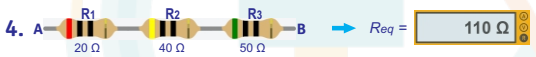
$$R = (9 - 2) / 0,02$$

$$R = 350 \Omega$$

RESOLUÇÃO - ATIVIDADE 01

Associação de resistores em série

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD, CALCULE:



Utilizando o recurso,



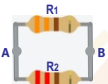
calcule:

- A ~ B: 1.530 Ω
- B ~ C: 2.220 Ω
- C ~ D: 7.000 Ω
- A ~ C: 3.750 Ω
- B ~ D: 9.220 Ω
- A ~ D: 10.750 Ω

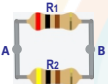
RESOLUÇÃO - ATIVIDADE 02

Associação de resistores em paralelo

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD, CALCULE:

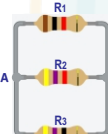
1. 
$$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} = \frac{1}{330} + \frac{1}{220} = \frac{1}{132} \Omega$$

ou, no caso de cálculo com somente dois resistores:
$$R_{eq} = (R_1 \times R_2) / (R_1 + R_2) = (330 \times 220) / (330 + 220) = 132 \Omega$$

2. 
$$R_1 = 200 \Omega \quad R_{eq} = (R_1 \times R_2) / (R_1 + R_2)$$

$$R_2 = 400 \Omega \quad R_{eq} = 200 \times 400 / 200 + 400$$

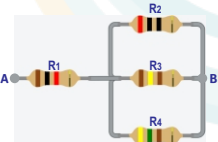
$$R_{eq} = 133,33 \Omega$$

3. 
$$R_1 = 1 \text{ K}\Omega$$

$$R_2 = 4,7 \text{ K}\Omega \quad \frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$$

$$R_3 = 47 \Omega \quad \frac{1}{R_{eq}} = \frac{1}{1.000} + \frac{1}{4.700} + \frac{1}{47}$$

$$R_{eq} = 44,46 \Omega$$

4. 
$$R_1 = 1 \text{ K}\Omega$$

$$R_2 = 20 \Omega \quad \frac{1}{R_{eq_{2-4}}} = \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4}$$

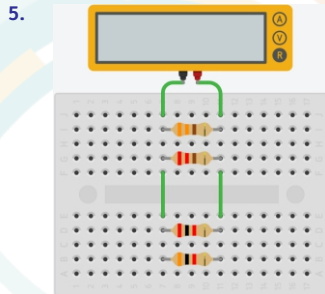
$$R_3 = 140 \Omega \quad \frac{1}{R_{eq_{2-4}}} = \frac{1}{20} + \frac{1}{140} + \frac{1}{450}$$

$$R_4 = 450 \Omega \quad \frac{1}{R_{eq_{2-4}}} = 374/6.300$$

$$R_{eq_{2-4}} = 16,84 \Omega$$

Parte 1 - Resolvendo a associação em paralelo
Parte 2 - Resolvendo a associação em série
$$R_{eq} = R_1 + R_{eq_{2-4}} = 1.000 + 16,84$$

$$R_{eq} = 1.016,8 \Omega$$



$R_1 = 330 \Omega$

$R_2 = 220 \Omega$

$R_3 = 2 \text{ K}\Omega$

$R_4 = 3 \text{ K}\Omega$

119 Ω

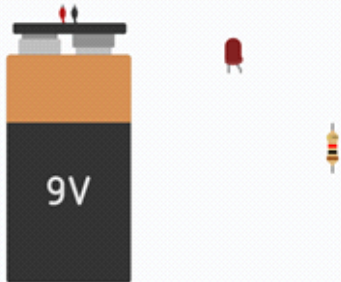
1.02 $\text{k}\Omega$

RESOLUÇÃO - ATIVIDADE 03

Calculando o valor do resistor para ligar o LED

O OBJETIVO DESTA ATIVIDADE É LIGAR O LED.

PARA QUE ISTO ACONTEÇA VOCÊ DEVE DETERMINAR O VALOR DO RESISTOR E INTERLIGAR OS COMPONENTES.



A. QUAL O VALOR DO RESISTOR?

350 Ω

B. A SIMULAÇÃO ACONTECEU CORRETAMENTE NA PRIMEIRA TENTATIVA? CASO A RESPOSTA SEJA NEGATIVA, DESCREVER O QUE FOI FEITO PARA CORRIGIR O PROBLEMA.

C. COM O AUXÍLIO DO MULTÍMETRO DIGITAL, INFORME O VALOR DA TENSÃO NOS TERMINAIS DO RESISTOR.

6.90 V

D. INFORME TAMBÉM O VALOR DA TENSÃO NOS TERMINAIS DO LED.

2.07 V

E. TESTE COM RESISTORES DE 10 Ω , 220 Ω e 10 M Ω . DESCREVA O QUE ACONTECE.

10 Ω - A corrente no LED atual é de 395 mA, enquanto o máximo absoluto é de 20 mA.

220 Ω - A corrente no LED atual é de 30.9 mA, enquanto o máximo recomendado é de 20 mA. A vida útil do LED pode ser reduzida.

10 M Ω - O valor do resistor é muito alto. A corrente elétrica é muito próxima a 0,0 A. O LED não acende.

RESOLUÇÃO - ATIVIDADE 05

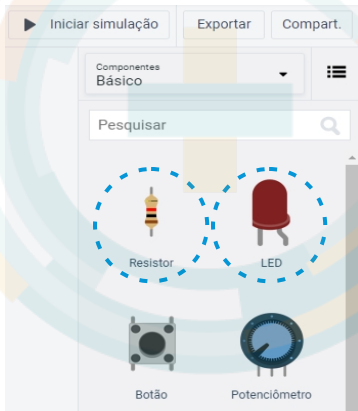
Semáforo

Objetivo: Simular o funcionamento de um semáforo.

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD

1º
PASSO

Figura 1: Componentes.

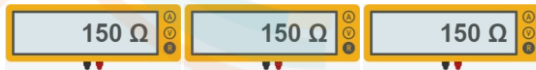


ADICIONAR OS COMPONENTES

CLIQUE NOS COMPONENTES E ARRASTE PARA ÁREA DE TRABALHO

- 1 ARDUINO
- 3 LED's
- 3 RESISTORES

SABENDO QUE A TENSÃO DE SAÍDA NOS PINOS DO ARDUINO É DE 5V, DETERMINE O VALOR DE CADA RESISTOR.



Fonte: os autores.

RESOLUÇÃO - ATIVIDADE 05

Semáforo

Objetivo: Simular o funcionamento de um semáforo.

UTILIZANDO SIMULADOR VIRTUAL TINKERCAD

2º
PASSO

INTERLIGAR OS COMPONENTES

Figura 2: Layout.

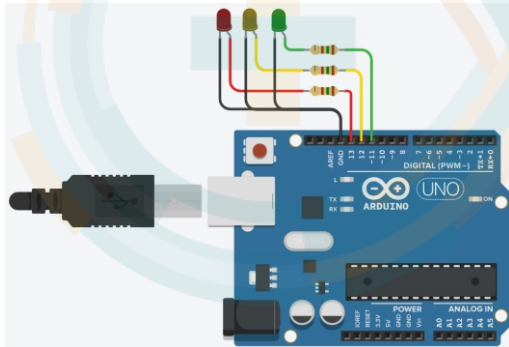
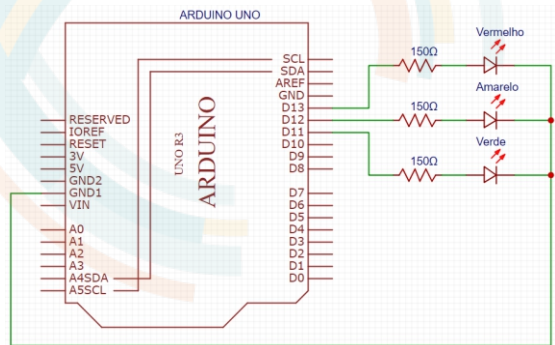


Figura 3: Esquemático.



RESOLUÇÃO - ATIVIDADE 05

Semáforo

ATIVIDADE
EM GRUPO

Objetivo: Simular o funcionamento de um semáforo.

3º
PASSO

CÓDIGO

CLICAR →

Código

Iniciar simulação

Exportar

Compart.

```
// Atividade 05 - Semáforo
```

```
int vmPin = 13; // LED vermelho conectado ao pino 13
int amPin = 12; // LED amarelo conectado ao pino 12
int vdPin = 11; // LED verde conectado ao pino 11
```

```
void setup () {
  pinMode (vmPin, OUTPUT); // Define o vmPin como saída
  pinMode (amPin, OUTPUT); // Define o amPin como saída
  pinMode (vdPin, OUTPUT); // Define o vdPin como saída
}
```

RESOLUÇÃO - ATIVIDADE 05

Semáforo

ATIVIDADE
EM GRUPO

```
void loop () {  
  
  // Estado 1 - Vermelho: Ligado | Amarelo: Desligado | Verde: Desligado  
  digitalWrite (vmPin, HIGH); // LIGA o LED vermelho  
  digitalWrite (amPin, LOW);  // DESLIGA o LED amarelo  
  digitalWrite (vdPin, LOW);  // DESLIGA o LED verde  
  delay(2000);                // aguarda dois segundos  
  
  // Estado 2 - Vermelho: Desligado | Amarelo: Ligado | Verde: Desligado  
  digitalWrite (vmPin, LOW);  // DESLIGA o LED vermelho  
  digitalWrite (amPin, HIGH); // LIGA o LED amarelo  
  digitalWrite (vdPin, LOW);  // DESLIGA o LED verde  
  delay(1000);                // aguarda um segundo  
  
  // Estado 3 - Vermelho: Desligado | Amarelo: Desligado | Verde: Ligado  
  digitalWrite (vmPin, LOW);  // DESLIGA o LED vermelho  
  digitalWrite (amPin, LOW);  // DESLIGA o LED amarelo  
  digitalWrite (vdPin, HIGH); // LIGA o LED verde  
  delay(2000);                // aguarda dois segundos  
}
```

RESOLUÇÃO - ATIVIDADE 05

Semáforo

ATIVIDADE
EM GRUPO

Objetivo: Simular o funcionamento de um semáforo.

Diagrama de Tempo - Semáforo Vermelho

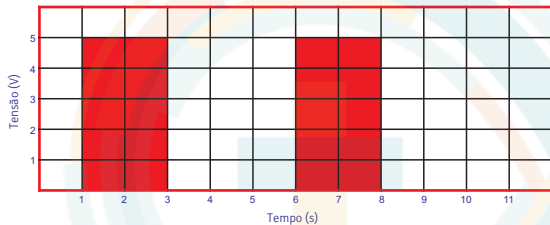


Diagrama de Tempo - Semáforo Verde

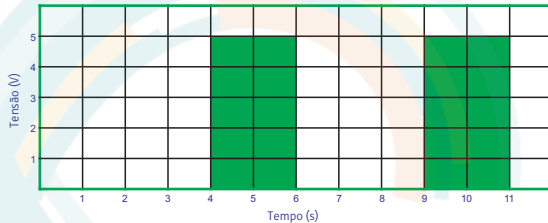
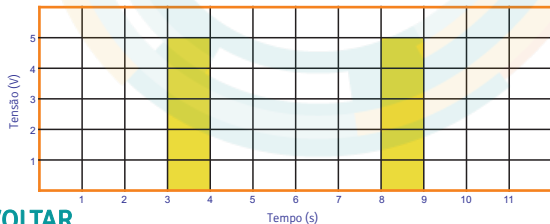


Diagrama de Tempo - Semáforo Amarelo



Grupo 01
<https://bit.ly/2ZemuWq>

Grupo 02
<https://bit.ly/3AsJZIT>

Grupo 03
<https://bit.ly/2Z8TvTH>



Encontro 05



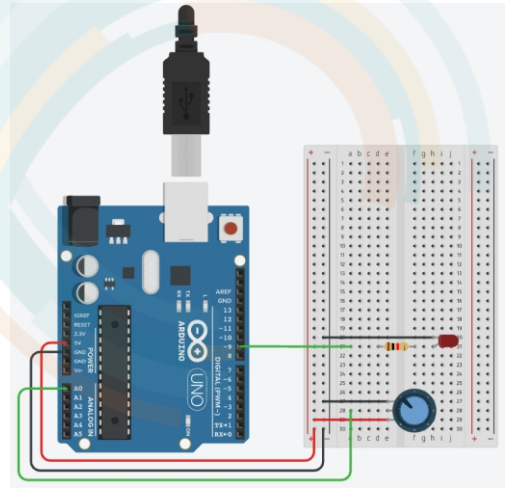
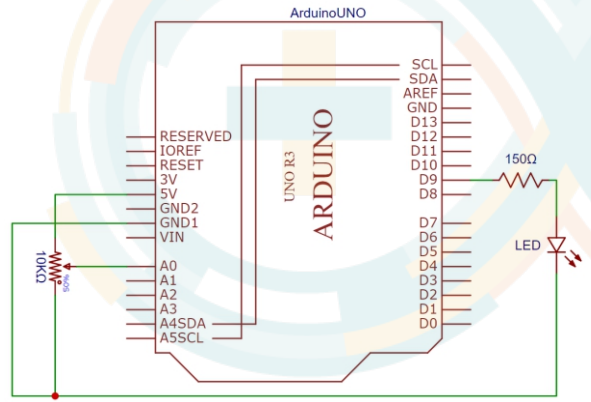
RESOLUÇÃO - ATIVIDADE 07

Regulando o brilho do LED por meio do Potenciômetro

Objetivo: Realizar o controle de brilho do LED por meio da variação do potenciômetro.

2º
PASSO

INTERLIGAR OS COMPONENTES



RESOLUÇÃO - ATIVIDADE 07


Regulando o brilho do LED por meio do Potenciômetro

3º
PASSO

CÓDIGO

CLICAR →

 Código

 Iniciar simulação

Exportar

Compart.

```

01 //Pino de entrada analógica ao qual o potenciômetro está conectado
02 int POT = A0;
03 //Pino de saída analógica ao qual o LED está conectado
04 int LED = 9;
05
06 int valorPOT = 0; //Valor lido do potenciômetro
07 int valorLED = 0; //Valor de saída para o PWM (saída analógica)
08
09 void setup () {
10     Serial.begin(9600); //Inicializa a comunicação serial a 9600 bps
11 }
12
    
```

RESOLUÇÃO - ATIVIDADE 07

Regulando o brilho do LED por meio do Potenciômetro

3º
PASSO

CÓDIGO

CLICAR →

Código

Iniciar simulação

Exportar

Compart.

```

13 void loop () {
14     valorPOT = analogRead(POT); //Lê o valor analógico
15     //Mapeia para o intervalo da saída analógica - Função map( )
16     valorLED = map(valorPOT, 0, 1023, 0, 255);
17     //Muda o valor de saída analógica
18     analogWrite(LED, valorLED);
19     //Imprime os resultados no monitor serial
20     Serial.print("Valor do potenciometro = ");
21     Serial.print(valorPOT);
22     Serial.print("\t Valor do LED = ");
23     Serial.println(valorLED);
24     //Aguarda 2 milissegundos antes do próximo loop
25     delay(2);
26 }
    
```

RESOLUÇÃO - ATIVIDADE 07

Regulando o brilho do LED por meio do Potenciômetro

ANALISANDO O CÓDIGO

Linhas 1, 3, 15, 17, 19 e 24. Comentários gerais.

Linha 2. Pino de entrada analógica ao qual o potenciômetro está conectado.

Linha 4. Pino de saída analógica ao qual o LED está conectado

Linha 6. Armazena na variável `valorPOT` o valor lido do potenciômetro.

Linha 7. Armazena na variável `valorLED` o valor de saída para o PWM (saída analógica).

Linha 9. Executa a função `setup()`.

Linha 10. Inicia a comunicação serial entre a placa do Arduino e o computador a 9600 *bits* por segundo;

Linha 11. Término de execução da função `setup()`.

Linha 13. Executa a função `loop()`. É a função principal do programa e executa continuamente enquanto o Arduino estiver ligado.

RESOLUÇÃO - ATIVIDADE 07

Regulando o brilho do LED por meio do Potenciômetro

ANALISANDO O CÓDIGO

Linha 14. Armazena o valor lido no pino analógico e armazena na variável valorPOT.

Linha 16. O código utiliza a função chamada `map()`, que converte o valor lido da entrada analógica, entre 0 e 1023, para um valor entre 0 e 255 (8 bits), que será utilizado para ajustar o brilho do LED. A sintaxe dessa função é a seguinte: `map(valor, de_menor_valor, de_maior_valor, para_menor_valor, para_maior_valor)`. É representado no código: `outputValue = map(sensorValue, 0, 1023, 0, 255);`

O `sensorValue` é lido do potenciômetro e será um valor entre 0 e 1023. Este valor será convertido entre 0 e 255, sendo atribuído à variável LED. A função PWM é similar a uma saída analógica através de uma saída digital. O estado da porta varia do nível lógico 0 para 1, ou 0V para 5V, muito rapidamente, simulando uma tensão entre 0 e 5 volts.

RESOLUÇÃO - ATIVIDADE 07

Regulando o brilho do LED por meio do Potenciômetro

ANALISANDO O CÓDIGO

Linha 18. Muda o valor de saída analógica.

Linhas 20, 21, 22 e 23. Imprime os valores no serial monitor.

Linha 25. Aguarda 2 milissegundos antes do próximo loop.

Linha 26. Término de execução da função `loop()`. Reinicia desde o princípio e assim infinitamente até que o Arduino seja desligado ou o botão `Reset` pressionado.

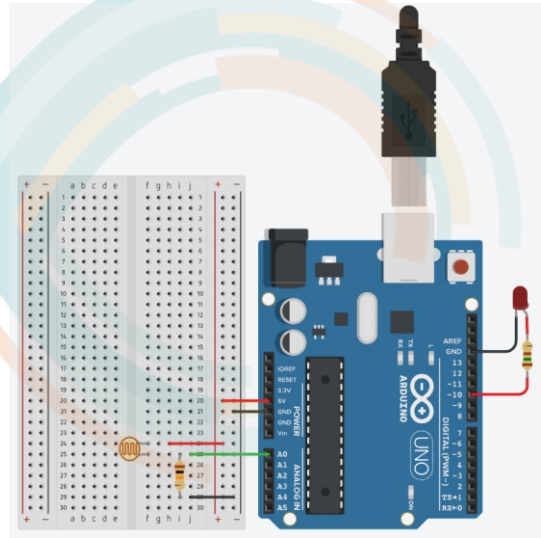
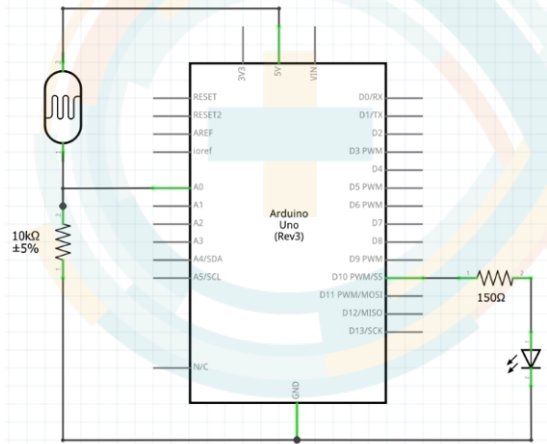
RESOLUÇÃO - ATIVIDADE 09

Acionando um LED por meio do LDR

Objetivo: Ligar / Desligar o LED conforme a intensidade luminosa do ambiente.

2º
PASSO

INTERLIGAR OS COMPONENTES



RESOLUÇÃO - ATIVIDADE 09

Acionando um LED por meio do LDR

ATIVIDADE
EM GRUPO

Objetivo: Ligar / Desligar o LED conforme a intensidade luminosa do ambiente.

3º
PASSO

CÓDIGO

CLICAR →

Código

Iniciar simulação

Exportar

Compart.

```
01 int pinLed = 10;
02 int LDR = A0;
03 int valueLDR = 0;
04
05 void setup() {
06     pinMode(pinLed, OUTPUT);
07 }
08 void loop() {
09     valueLDR = analogRead(LDR);
10     if(valueLDR < 750) {
11         digitalWrite(pinLed, HIGH);
12     }
13     else {
14         digitalWrite(pinLed, LOW);
15     }
16     delay(10);
17 }
```

RESOLUÇÃO - ATIVIDADE 09

Acionando um LED por meio do LDR

ATIVIDADE
EM GRUPO

ANALISANDO O CÓDIGO

Linha 1. Armazena o valor do pino na variável pinLED.

Linha 2. Pino de entrada analógica ao qual o LDR está conectado, ou seja, A0.

Linha 3. Determina uma variável, valueLDR, do tipo inteira para armazenar o valor lido do LDR (valor entre 0 a 1023), que está conectado no pino A0. Inicialmente é atribuído o valor zero.

Linha 5. Executa a função setup(). Possui apenas uma instrução, pinMode(), que diz ao Arduino que desejamos definir o modo de um de seus pinos como saída (OUTPUT).

Linha 6. Inicializando o pino da variável pinLED como saída;

Linha 7. Término de execução da função setup().

RESOLUÇÃO - ATIVIDADE 09

Acionando um LED por meio do LDR

ATIVIDADE
EM GRUPO

ANALISANDO O CÓDIGO

Linha 08. Executa a função `loop()`. Esta é a função principal do programa e executa continuamente enquanto o Arduino estiver ligado.

Linha 09. Armazena o valor lido no pino LDR na variável `valueLDR`.

Linha 10. Compara se (`if`) o valor da variável `valueLDR` com o valor 750. Se for menor, liga o LED, caso contrário mantém o LED desligado.

Linha 11. Atribui nível lógico alto para o pino de saída, ou seja, fornece uma tensão de 5V no pino 10, ligando o LED.

Linha 12. Finaliza a estrutura condicional "se". "Pula" para a instrução `delay`.

Linha 13. Executa a instrução `senão` (`else`) se valor da variável `valueLDR` for maior que 750.

RESOLUÇÃO - ATIVIDADE 09

Acionando um LED por meio do LDR

ATIVIDADE
EM GRUPO

ANALISANDO O CÓDIGO

Linha 14. Atribui nível lógico baixo para o pino de saída, ou seja, fornece uma tensão de 0V no pino 10, desligando o LED.

Linha 15. Finaliza a estrutura condicional "senão".

Linha 16. Essa instrução diz ao Arduino para esperar 10 milissegundos.

Linha 17. Término de execução da função loop(). Reinicia desde o princípio e assim infinitamente até que o Arduino seja desligado ou o botão Reset pressionado.

AVALIAÇÃO DA APRENDIZAGEM

TM

<https://forms.gle/BjA36ZpUnNYMFpeo6>

Ambiente Virtual Tinkercad - para docentes

Acesse o link: www.tinkercad.com

Clique em:
«Fazer login
ou em
INSCREVER-SE»

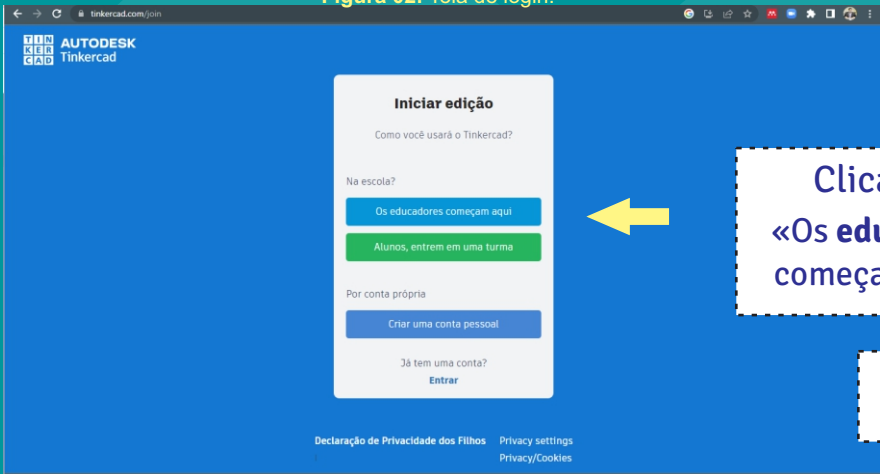
Figura 01: Tela inicial.



Ambiente Virtual Tinkercad

TM

Figura 02: Tela de login.



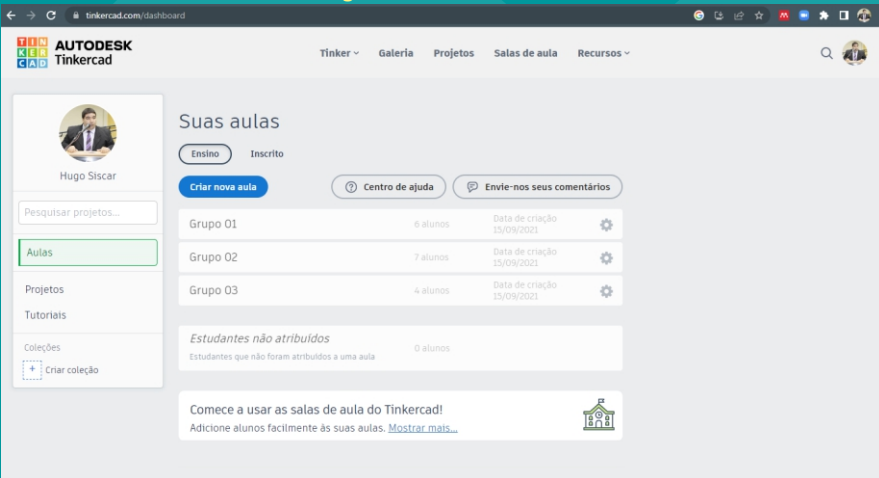
Clicar em
«Os **educadores**
começam aqui»

Seguir as orientações
para criação de conta

Fonte: Tinkercad (com adaptações).

Ambiente Virtual TinkerCad

Figura 03: Criar nova aula.



The screenshot shows the Tinkercad dashboard for user Hugo Siscar. The main navigation bar includes 'Tinker', 'Galeria', 'Projetos', 'Salas de aula', and 'Recursos'. The left sidebar contains 'Pesquisar projetos...', 'Aulas', 'Projetos', 'Tutoriais', and 'Coleções' with a '+ Criar coleção' button. The main content area is titled 'Suas aulas' and features tabs for 'Ensino' and 'Inscrito'. A prominent blue button labeled 'Criar nova aula' is visible. Below it are three classroom groups: 'Grupo 01' (6 alunos), 'Grupo 02' (7 alunos), and 'Grupo 03' (4 alunos), each with a 'Data de criação' of 15/09/2021 and a settings gear icon. A section for 'Estudantes não atribuídos' shows 0 alunos. A bottom banner encourages using classrooms with a 'Mostrar mais...' link.

Grupo	Alunos	Data de criação
Grupo 01	6 alunos	15/09/2021
Grupo 02	7 alunos	15/09/2021
Grupo 03	4 alunos	15/09/2021

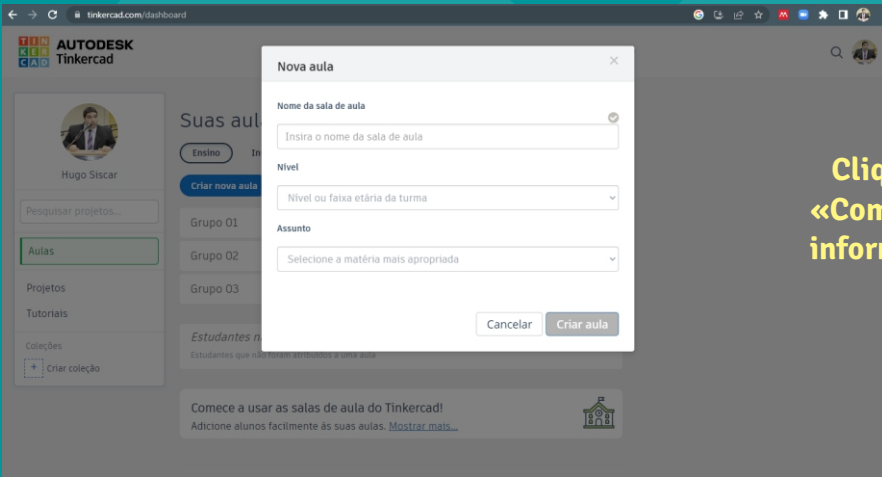
Estudantes não atribuídos
Estudantes que não foram atribuídos a uma aula | 0 alunos

Clique em:
«Criar nova aula»

Ambiente Virtual Tinkercad

TM

Figura 04: Editando classe.



Clique em:
«Complete as
informações»

Ambiente Virtual Tinkercad

TM

Figura 05: Tela das classes.

The screenshot displays the Tinkercad interface for a teacher's class management page. The header includes the Autodesk Tinkercad logo and navigation links: Tinker, Galeria, Projetos, Salas de aula, and Recursos. The user profile for Hugo Siscar is visible on the left. The main content area is titled 'Suas aulas' and features two tabs: 'Ensino' (selected) and 'Inscrito'. Below the tabs are buttons for 'Criar nova aula', 'Centro de ajuda', and 'Envie-nos seus comentários'. A table lists the classes:

Nome da Classe	Alunos	Data de criação	Ações
Classe modelo	0 alunos	17/11/2022	[Configurar]
Grupo 01	6 alunos	15/09/2021	[Configurar]
Grupo 02	7 alunos	15/09/2021	[Configurar]
Grupo 03	4 alunos	15/09/2021	[Configurar]

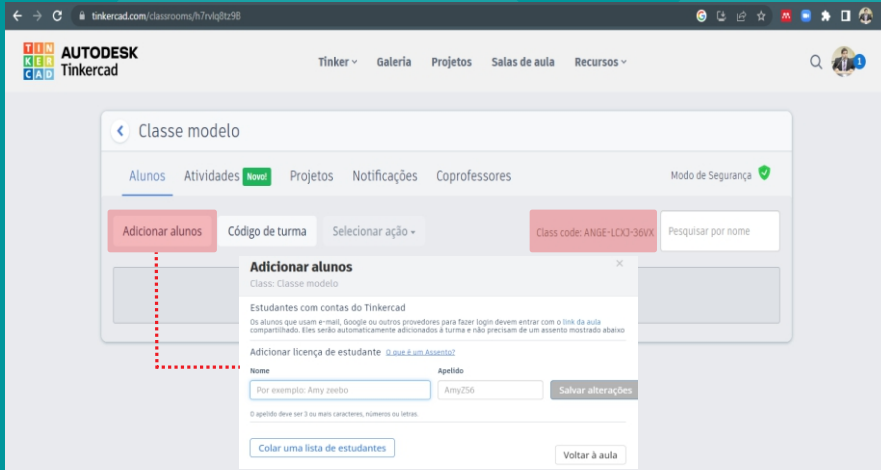
Below the table, there is a section for 'Estudantes não atribuídos' (0 alunos) and a promotional banner for using Tinkercad classrooms.

Clique em:
«Clique no nome da aula para editar a classe»

Fonte: Tinkercad (com adaptações).

Ambiente Virtual Tinkercad

Figura 06: Editar alunos.

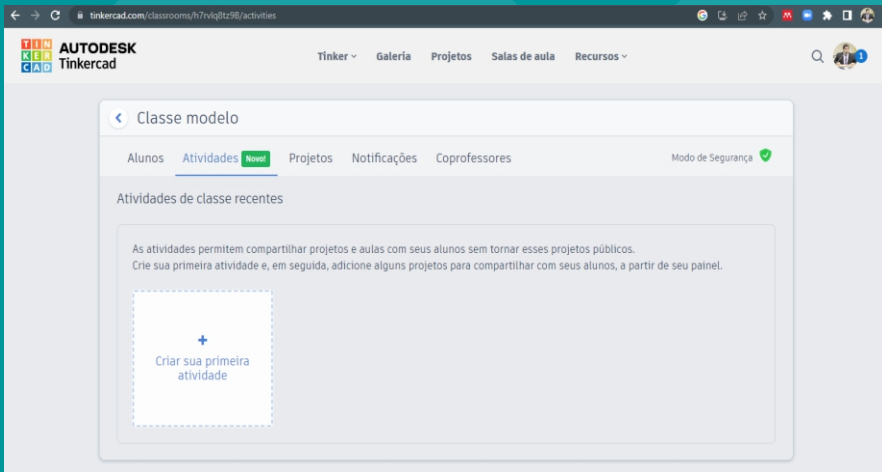


Você pode adicionar os estudantes indicando um «Nome» e um «Apelido» ou informando o código da classe para o estudante

Fonte: Tinkercad (com adaptações).

Ambiente Virtual Tinkercad

Figura 07: Criando atividades.



As atividades que você criar serão compartilhadas com os estudantes daquela classe.

Fonte: Tinkercad.

TEORIA HISTÓRICO-CULTURAL

CONCEITOS ESPONTÂNEOS E CIENTÍFICOS

TM

“[...] os conceitos iniciais que foram construídos na criança ao longo de sua vida no contexto de seu ambiente social (Vygotski chamou esses conceitos de "diários" ou "espontâneos", espontâneos na medida em que são formados independentemente de qualquer processo especialmente voltado para desenvolver seu controle) são agora deslocados para um novo processo, para nova relação especialmente cognitiva com o mundo, e assim nesse processo os conceitos da criança são transformados e sua estrutura muda. Durante o desenvolvimento da consciência na criança o entendimento das bases de um sistema científico de conceitos assume agora a direção do processo” (VIGOTSKI, 2007, p. 163).

TEORIA HISTÓRICO-CULTURAL

MEDIAÇÃO e INTERAÇÃO

TM

A **mediação** é um conceito central na teoria de Vygotsky. Para ele:

[...] "o uso de meios artificiais - a transição para a atividade mediada - muda fundamentalmente, todas as operações psicológicas, assim como o uso de instrumentos amplia de forma ilimitada a gama de atividades em cujo interior as novas funções psicológicas podem operar. Nesse contexto, podemos usar o termo função psicológica superior, ou comportamento superior com referência à combinação entre o instrumento e o signo na atividade psicológica" (VIGOTSKY, 2007, p. 56).

Outro ponto importante da concepção vygostskyana é sobre a **interação social** e o papel que desempenha no desenvolvimento das funções psicológicas superiores.

TEORIA HISTÓRICO-CULTURAL

IMITAÇÃO

TM

[...] "quando afirmamos que a criança age por **imitação** isto não quer dizer que ela olhe outra pessoa nos olhos e imite. Se eu vi alguma coisa hoje, e faço a mesma coisa amanhã, eu faço por imitação. Quando em casa uma criança resolve problemas depois de ter visto a amostra em sala de aula, ela continua a agir em **colaboração**, embora nesse momento o professor não esteja" (VIGOTSKY, 2001, p. 242).

TEORIA HISTÓRICO-CULTURAL

ZONA DE DESENVOLVIMENTO PROXIMAL - ZDP

TM



[...] distância entre o nível de desenvolvimento real, que se costuma determinar através da solução independente de problemas, e o nível de desenvolvimento potencial, determinado através da solução de problemas sobre a orientação de um adulto ou em colaboração com companheiros mais capazes" (VIGOTSKY, 2007, p. 97).

REFERÊNCIAS

CULKIN, J.; HAGAN, E. **Aprenda eletrônica com Arduino: um guia ilustrado de eletrônica para iniciantes**. São Paulo: Novatec Editora, 2018.

EVANS, M.; NOBLE, J.; HOCHENBAUM, J. **Arduino em Ação**. São Paulo: Novatec Editora, 2013.

MCRBERTS. M. **Arduino básico**. São Paulo: Novatec Editora, 2011.

SCOPACASA, V. Introdução à tecnologia de LED. **Revista Lume Arquitetura – Especial LA_PRO**, São Paulo, 1 ed., p. 5-11, nov. 2004.

VIGOTSKI, L. S. **A construção do pensamento e da linguagem**. 2. ed. São Paulo: Martins Fontes, 2009.

VIGOTSKI, L. S. **A formação social da mente**. Tradução de José Cipolla Neto et al. 7. ed. São Paulo: Martins Fontes, 2007.

YAMAMOTO, K. ; FUKU, L. F. Física para o ensino médio. 4. ed. São Paulo: Saraiva, 2016. v. 3.