PLACA ARDUÍNO

CIRCUITOS

ARDUÍNO

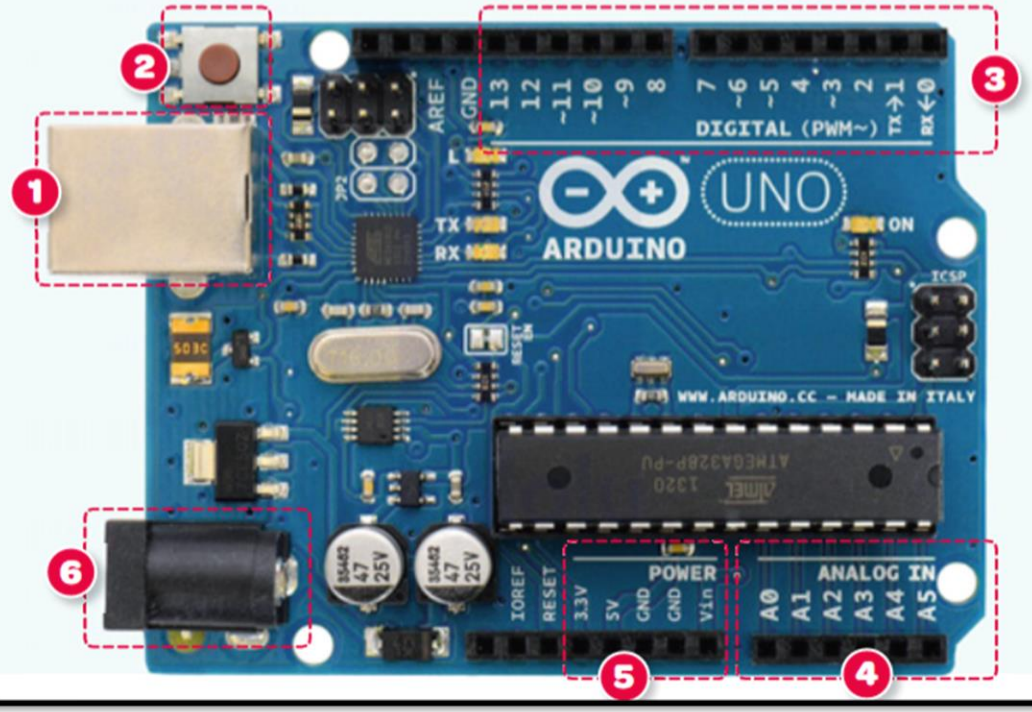
A placa Arduíno permite criar e programar objetos interativos que podem realizar tarefas de forma autónoma ou ser manipulados.

Pode utilizar o Arduíno para criar um jogo ou até para criar um robô com inteligência artificial.

ARDUÍNO

Elementos da placa Arduíno

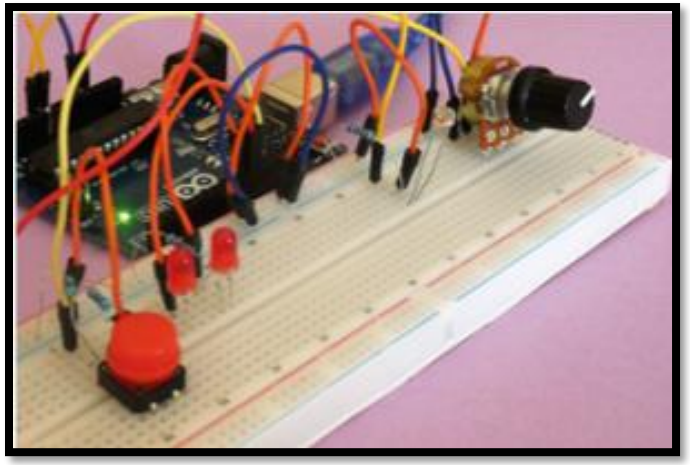
1. **Ligação USB** – Envia dados do computador para a placa (também fornece energia à placa).
2. **Reset** – Reinicia a programação da placa.
3. **Entradas e saídas digitais** – Enviam e recebem dados digitais dos componentes ligados à placa (os pinos 1 e 0, por vezes, não podem ser utilizados). Os pinos que têm o símbolo "~" também podem ser usados como saídas analógicas.
4. **Entradas analógicas** – Recebem dados analógicos dos componentes ligados à placa.
5. **Pinos de alimentação** – Fornecem energia aos componentes (3,3 volts ou 5 volts). Também há dois pinos "GND" (*ground* ou terra) de 0 V.
6. **Alimentação externa** – Alimenta a placa (alternativa à ligação USB). Normalmente são utilizadas pilhas para alimentar a placa.



ARDUÍNO

Entradas vs Saídas

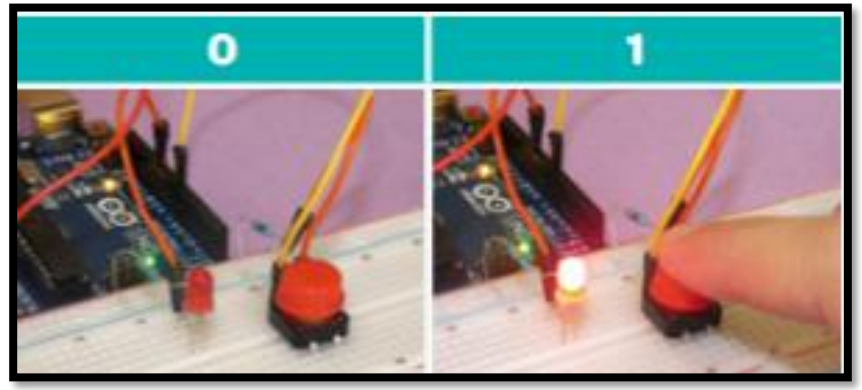
Entradas e saídas, referem-se a componentes que enviam ou recebem dados da placa. Por exemplo, imagine um botão que faz acender um LED. A placa Arduino recebe informação do botão e envia informação para o LED. O botão é uma entrada e o LED é uma saída.



ARDUÍNO

Componentes digitais vs analógicos

Componentes digitais são aqueles que apenas enviam ou recebem os valores 0 ou 1 (desligado/ligado). Por exemplo, o botão é uma entrada digital pois apenas pode enviar um 0 (não pressionado) ou 1 (pressionado). Seguindo a mesma lógica, um LED é uma saída digital pois apenas pode receber 0 (desligado) ou 1 (ligado).

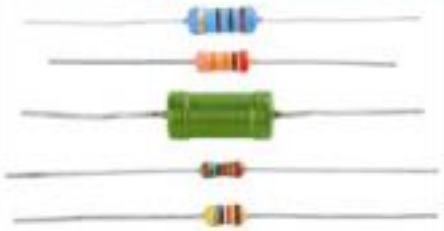


Componentes analógicos são aqueles que permitem enviar ou receber valores de diferentes intensidades.

ARDUÍNO

Criar circuitos elétricos com a placa Arduíno

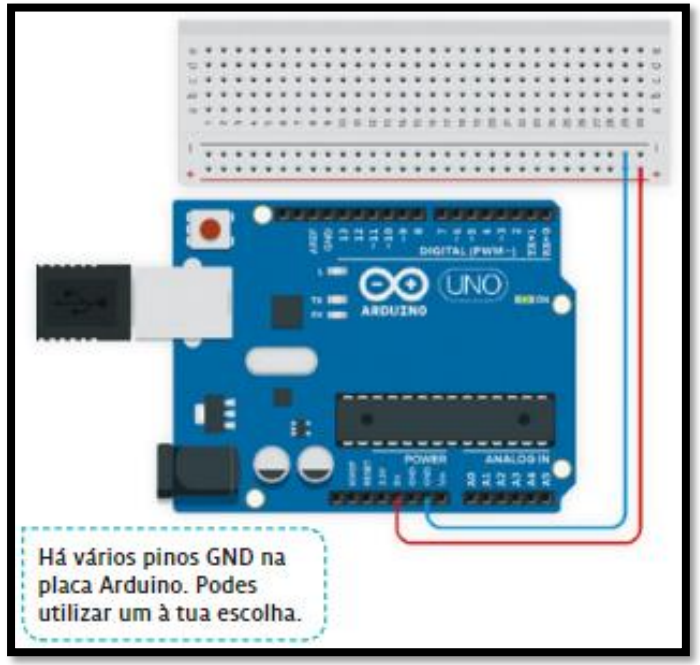
Componentes essenciais para criar circuitos

Placa de ensaio (*breadboard*)	Fios (*jumpers*)	Resistências (*resistors*)
		
Serve para testar os circuitos, permitindo reorganizar as ligações sempre que for necessário.	Ligam a placa Arduino à placa de ensaio e a placa de ensaio a outros componentes eletrónicos, permitindo interligar os diversos componentes. Apesar das diferentes cores, todos os fios são idênticos.	Reduzem a voltagem habitual dos circuitos (5 volts), uma vez que alguns componentes eletrónicos não suportam essa voltagem. Existem diferentes tipos de resistências (que podemos distinguir pelas suas faixas coloridas), mas habitualmente iremos utilizar a resistência de 220 Ω (ohm).

ARDUÍNO

Criar circuitos elétricos com a placa Arduíno

Para começar é necessário ligar a placa Arduíno ao computador (através de um cabo USB) e distribuir a energia pela *breadboard* (ensaio).

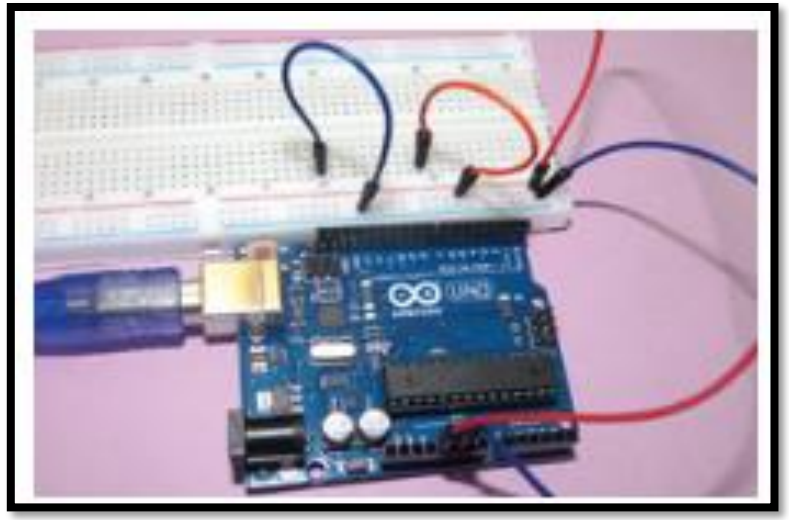


- Ligar um fio desde o pino '5V' até à linha da *breadboard* que tem o '+'.
- Ligar outro um fio desde o pino 'GND' até à linha da *breadboard* que tem o '–'. Isto vai distribuir a carga "positiva" pela linha '+' e carga "negativa" pela linha '–.

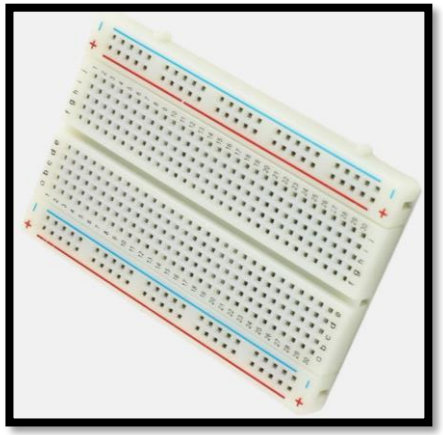
Habitualmente utilizam-se fios com cores quentes para a carga positiva e com cores frias para a carga negativa.

ARDUÍNO

É muito importante perceber como a energia elétrica é distribuída pela *breabdoard*, para que possa ligar todos os componentes corretamente.



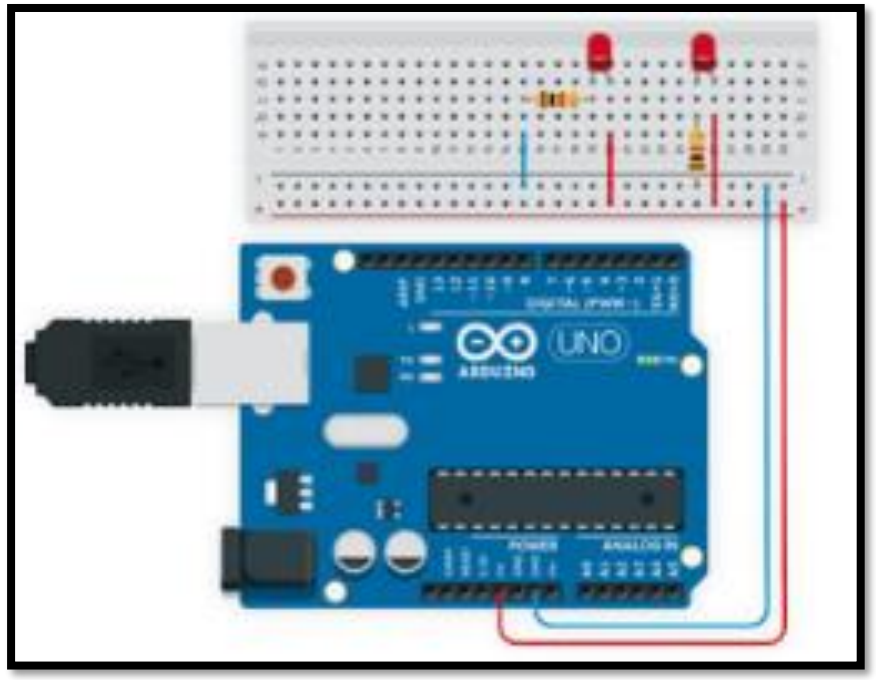
Internamente uma *breabdoard* tem a configuração como pode ver na imagem abaixo.



Se utilizar um jumper (fio) que ligue a linha com carga positiva a uma qualquer coluna da breabdoard, essa coluna também ficará com carga positiva.

ARDUÍNO

Para criar um circuito que permita ligar dois LEDs, usando a placa Arduíno e a breabdoard, pode ter algo como apresentado ao lado.



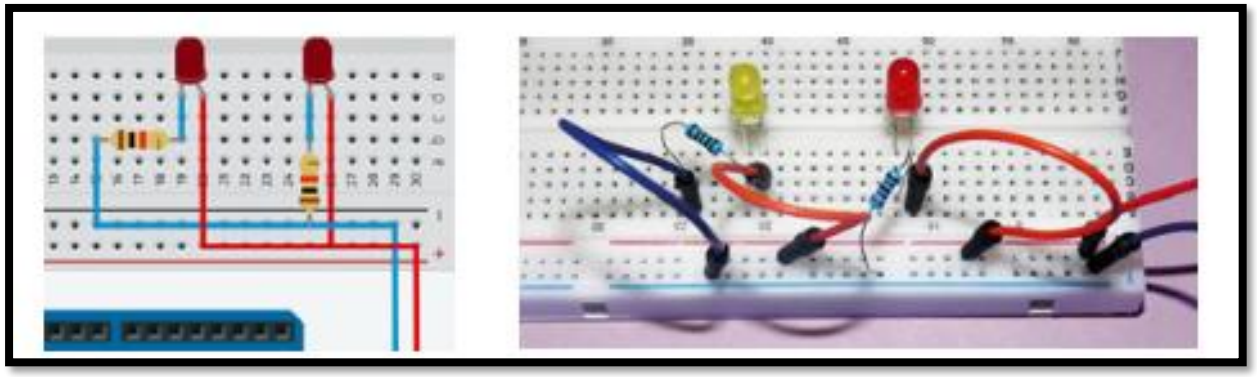
Neste exemplo, o LED fica sempre ligado.
Para o poder ligar ou desligar sempre que quiser ter de ligar o LED à um pino de saída digital, por exemplo, o pino 13.

ARDUÍNO

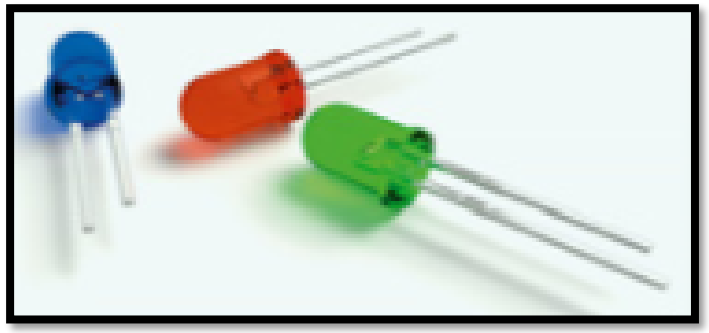
Repara que os dois LEDs estão ligados de forma diferente, ou seja, pode usar diferentes formas de ligar uma das pernas do LED à carga negativa.

Não importa se liga a resistência diretamente à carga negativa ou se usa um fio que ligue uma das pernas da resistência à carga negativa.

O importante é que seja criado um circuito contínuo, como pode ver ao lado.

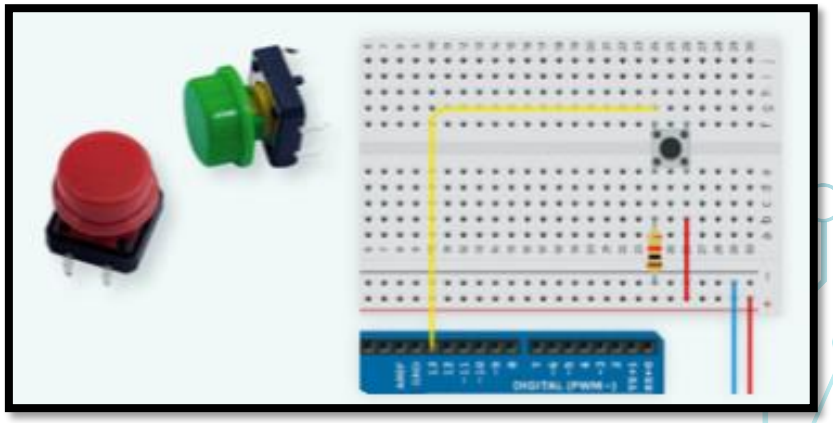


ARDUINO



Ligações de componentes à placa Arduíno

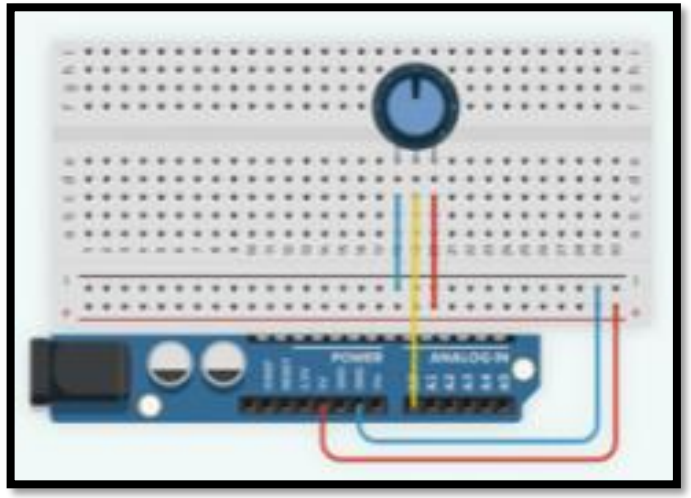
- LED (ligado a saída digital)
  - ✓ Emite luz de uma cor.
  - ✓ Tem um lado positivo (perna mais comprida) e outro negativo (perna mais curta) e deve ficar ligado corretamente, respeitando o lado positivo e negativo.
  - ✓ Não suporta uma voltagem de 5 V, por isso temos de usar uma resistência.
- Botão (ligado a entrada digital)
  - ✓ Pode ser pressionado para enviar um sinal.
  - ✓ O botão envia para o pino escolhido o valor 0 (botão solto) ou 1 (botão pressionado).



ARDUÍNO

Potenciómetro (ligado a entrada analógica)

- Ao rodá-lo, envia valores para a placa Arduíno entre 0 (esquerda) e 1023 (direita).
- Se trocarmos o polo positivo pelo negativo, funciona ao contrário.

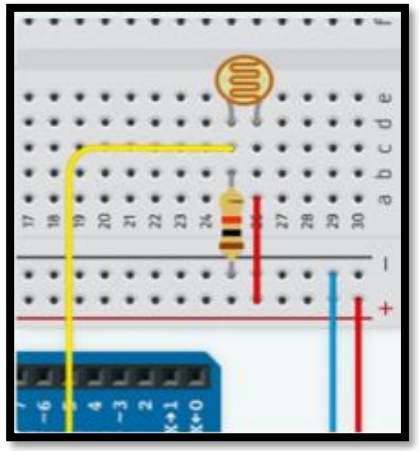


As entradas analógicas são os pinos desde o A0 até ao A5.

ARDUÍNO

Sensor de luz (ligado a entrada analógica)

- Mede a intensidade da luz.
- Ao receber luz, o sensor envia um valor para a placa consoante a intensidade da luz que recebe.
- As duas "pernas" do sensor, ao contrário dos LEDs, podem ser ligadas a qualquer polo.
- Também pode ser chamado fotoresistor.



ARDUÍNO

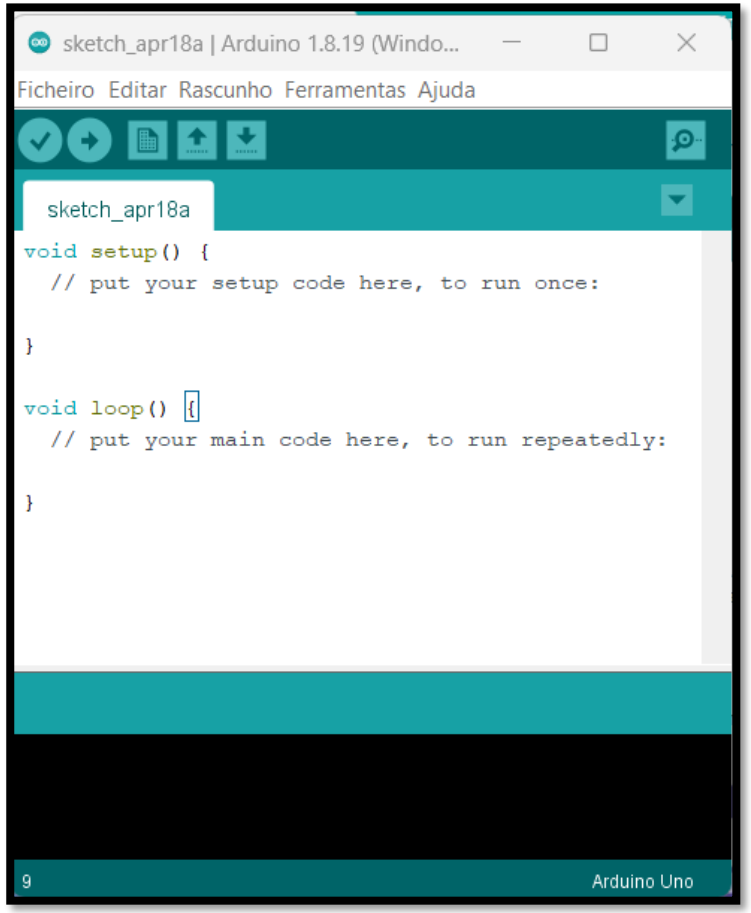
Outros componentes por exemplo:

- Motores (existem vários tipos)
- Sensores de proximidade (ultrassons)
- Sensores de infravermelhos
- Sensores de vibração
- Mostradores digitais
- Monitores LCD
- Buzzer

ARDUÍNO

Programar uma placa Arduíno

- Pode parecer uma tarefa complexa, mas pode aprender a fazê-lo de uma forma simples. A placa é programada através de uma aplicação chamada Arduino IDE. Este software utiliza um formato tradicional de programação, como no exemplo ao lado.
- Felizmente há aplicações que tornam a programação da placa muito mais simples e rápida.Vamos utilizar duas aplicações: a TinkerCADe e a Ardublockly. Ambas permitem programar, de forma simples, utilizando blocos.



ARDUÍNO

Ligações

- Arduino IDE

  www.arduino.cc/en/Main/Software

- TinkerCAD

  https://www.tinkercad.com (fazer registo e criar salas para trabalhar com os alunos)

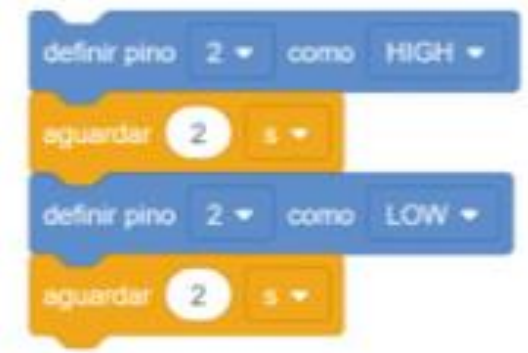
- Ardublockly

  http://www.anpri.pt/anprino/index.php/software/

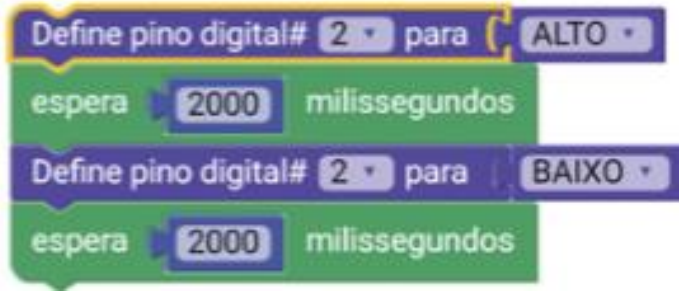
ARDUÍNO

Exemplo dos programas em blocos:

TinkerCAD



Ardublockly



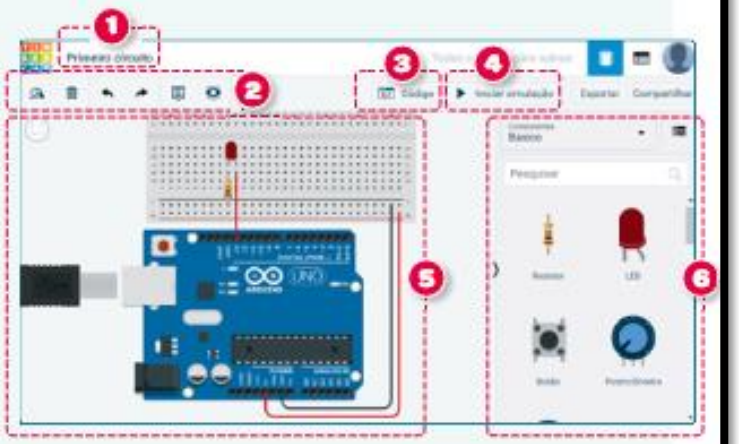
Após realizar a programação através de blocos, estes programas criam automaticamente o código que poderás transferir para a aplicação Arduino IDE.

Independentemente da ferramenta com que criamos o nosso programa, é sempre a Arduino IDE que envia a programação para a placa.

ARDUÍNO

Elementos da área de trabalho do TinkerCAD

1. **Nome do programa** – Permite atribuir um nome ao programa.
2. **Comandos** – Permitem rodar ou eliminar componentes, anular ou refazer ações, comentar e ocultar/mostrar componentes.
3. **Código** – Mostra as categorias, blocos e programação disponíveis.
4. **Iniciar simulação** – Inicia a simulação do circuito e respetiva programação.
5. **Área de simulação** – Área onde é criado e testado o circuito.
6. **Componentes** – Permite selecionar os componentes a utilizar.



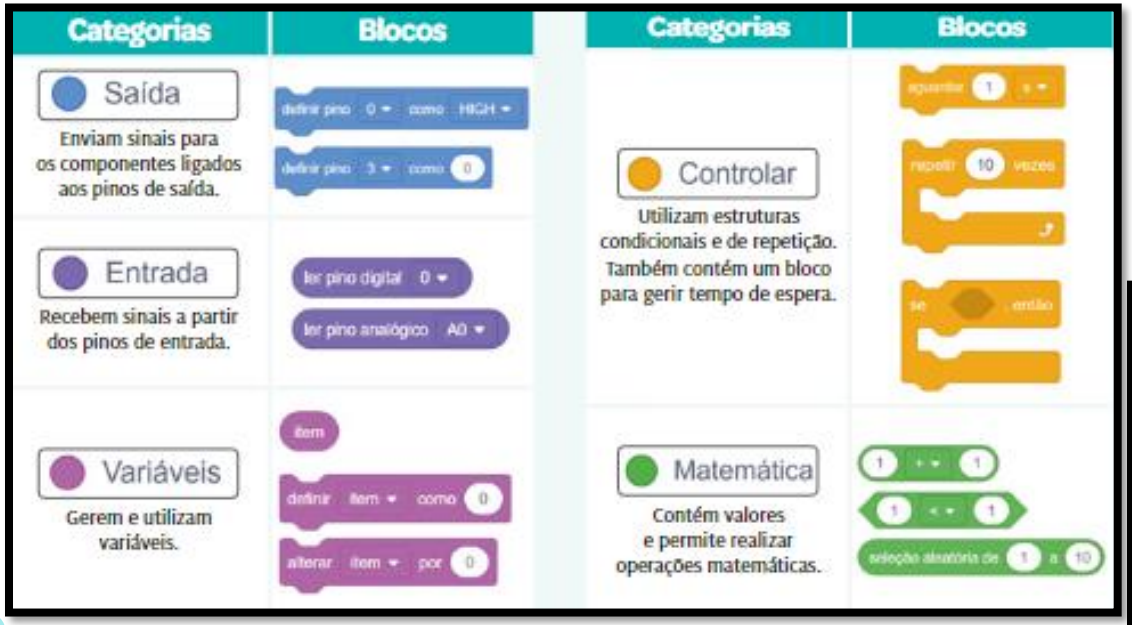
Ao clicar em "Código" surge o seguinte:

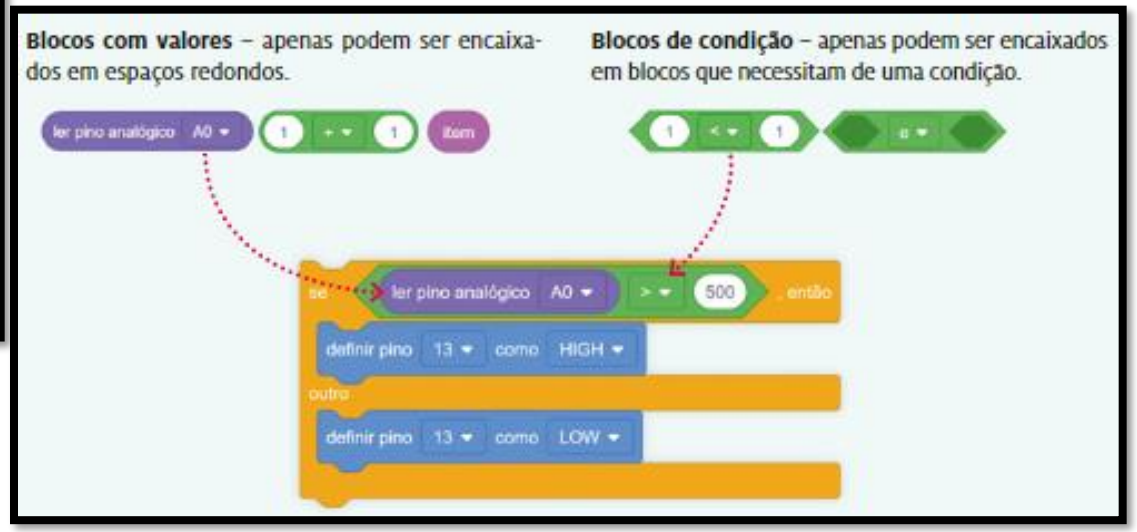
7. **Blocos** – Seleciona o modo de visualização: "Blocos", "Blocos + texto" ou "Texto".
8. **Categorias** – Mostra as categorias e respetivos blocos.
9. **Área de trabalho** – Área onde se programa o circuito. Contém botões para aumentar ou diminuir o tamanho dos blocos.
10. **Código** – Gerado automaticamente (só visível na opção "Blocos + texto").

O código gerado pode ser copiado e colado na aplicação Arduino IDE, para que possamos enviar a programação para uma placa Arduino real.

ARDUÍNO

Categorias de blocos no TinkerCAD





ARDUÍNO

Exemplo: Criação de um primeiro circuito

1 Acesso à aplicação e registo

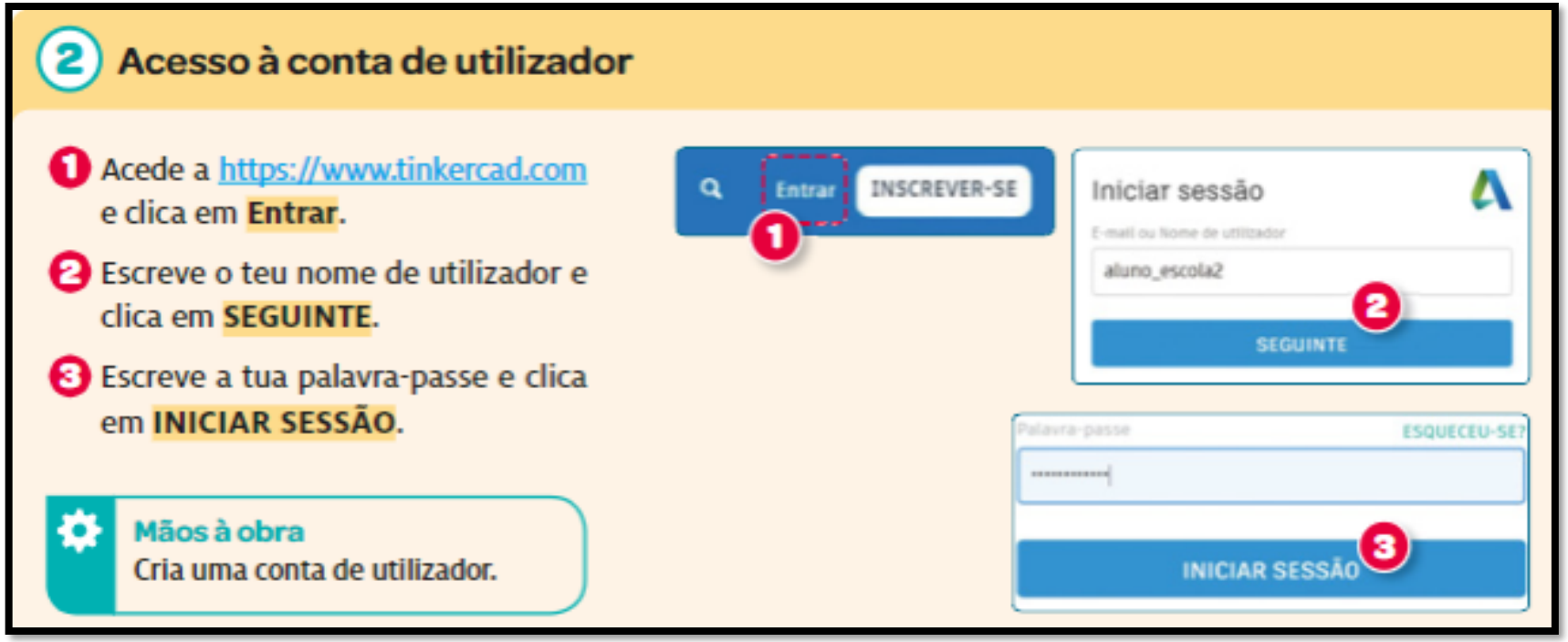
1. Acede a https://www.tinkercad.com.
2. Clica em **INSCREVER-SE**.
3. Preenche o formulário e clica em **SEGUINTE**.
4. Preenche os dados de utilizador e clica em **CRIAR CONTA**.
5. Digita o código de convite que o teu professor te indicar.

Nota: Só com 13 ou mais anos te podes registar em alguns serviços *web*. Se tiveres menos de 13 anos, podes registar-te indicando o *email* do teu encarregado de educação ou do teu professor.

Não te esqueças que...
A tua palavra-passe deve ser apenas do teu conhecimento. Caso te esqueças dela, o teu professor pode alterá-la.

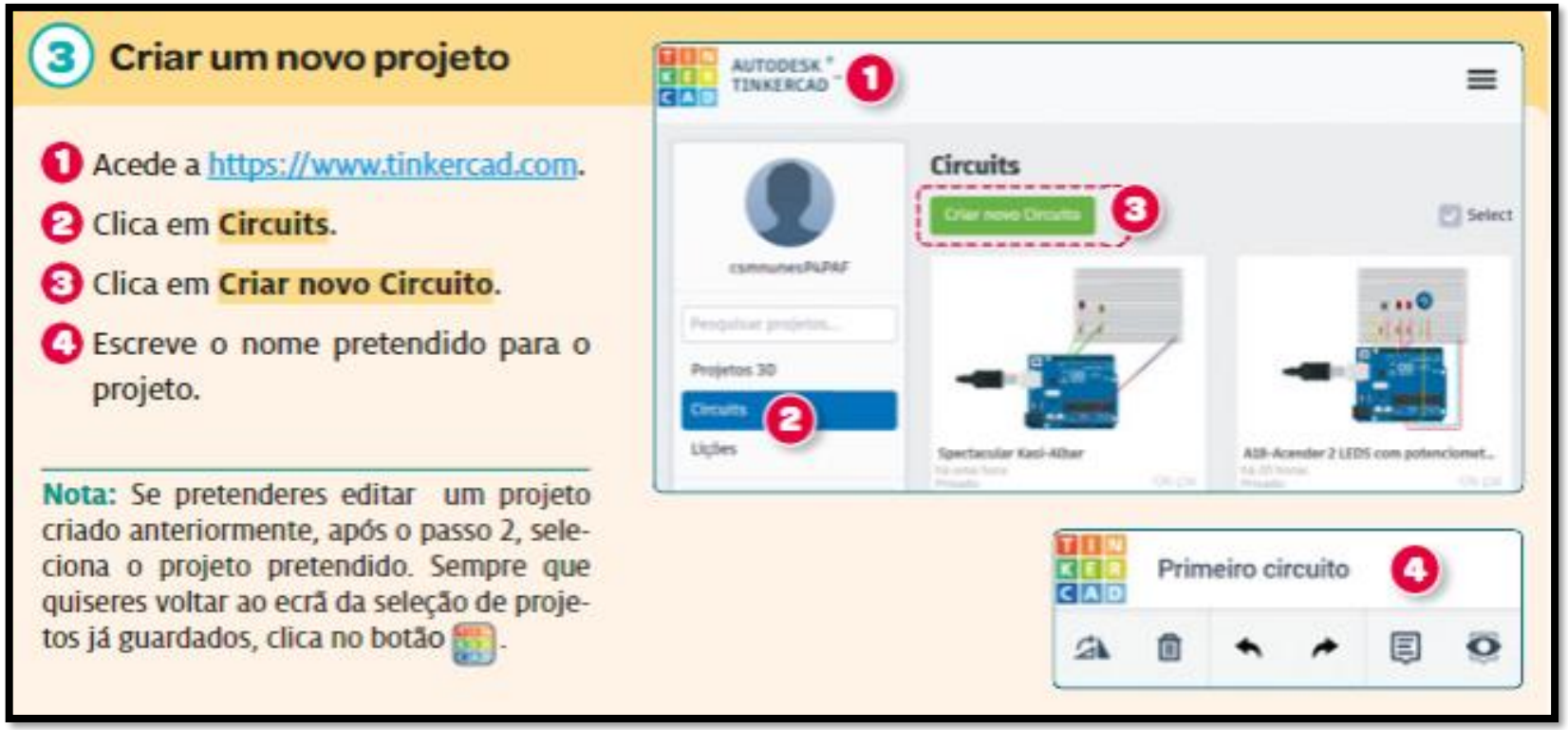
ARDUÍNO

Exemplo: Criação de um primeiro circuito (cont.)



ARDUÍNO

Exemplo: Criação de um primeiro circuito (cont.)

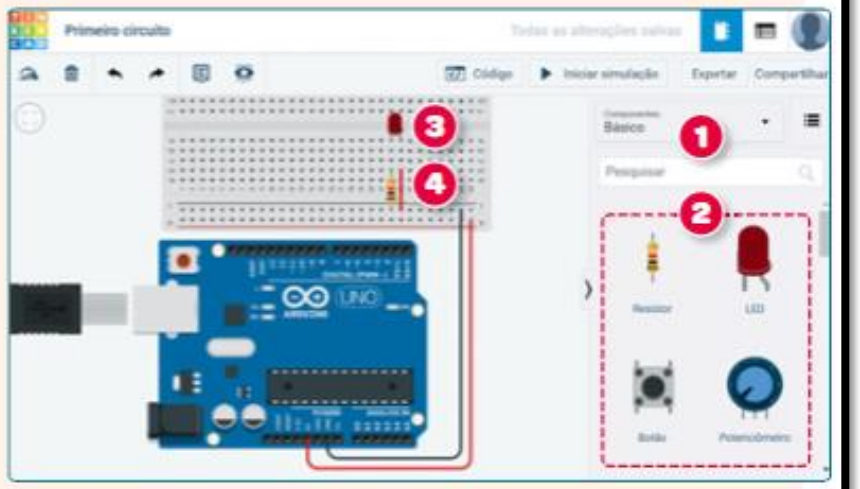


ARDUÍNO

Exemplo: Criação de um primeiro circuito (cont.)

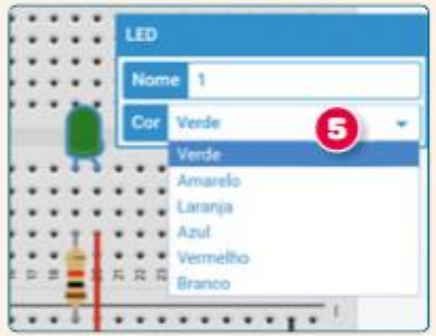
4 Criar um circuito

1. Seleciona o tipo de componentes.
2. Clica no componente pretendido.
3. Arrasta o componente para a sua posição.
4. Cria as ligações em falta, clicando e arrastando o rato entre dois orifícios.
5. (Opcional) Clica no componente (ou fio) e altera o seu nome e cor.



Nota: Se escolheres o tipo de componente "Disparadores Arduino", irás encontrar alguns pequenos circuitos já preparados, como os que podes ver abaixo.

Mãos à obra
Acede à tua conta e experimenta criar um novo circuito.



ARDUÍNO

Exemplo: Criação de um primeiro circuito (cont.)

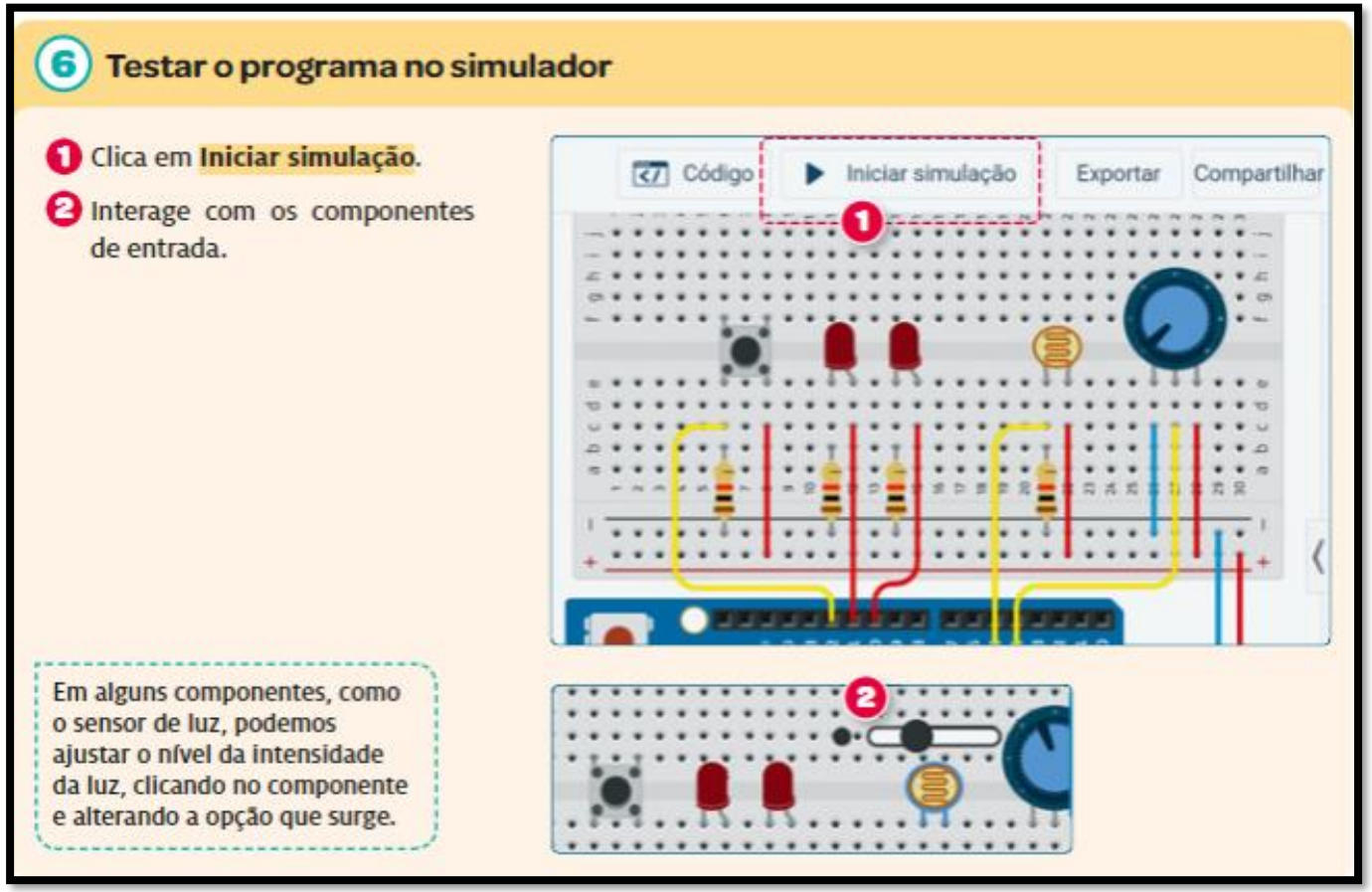
5 Escolher os blocos pretendidos

1. Clica em **Código**.
2. Seleciona a categoria que tem o bloco pretendido.
3. Arrasta o bloco para o local pretendido.
4. Altera opções/valores do bloco.
5. Repete os passos anteriores para os restantes blocos.

Nota: Podes clicar com o botão direito em cima de um bloco para aceder a opções como duplicar ou remover blocos. Também podes eliminar blocos movendo-os para cima das categorias ou para o caixote do lixo.

ARDUÍNO

Exemplo: Criação de um primeiro circuito (cont.)



ARDUÍNO

Exemplo: Criação de um primeiro circuito (cont.)

7 Enviar programa para a placa Arduino

Para enviar um programa para uma placa Arduino, tens de utilizar obrigatoriamente a aplicação Arduino IDE. Instala e abre a aplicação para enviares o código para a placa.

1. Seleciona a opção **Blocos + texto**.
2. Seleciona o código gerado automaticamente.
3. Copia o texto.
4. Abre a aplicação Arduino IDE, cola o código e envia-o para a placa Arduino clicando no comando (“enviar”).

Nota: Depois de enviado para a placa Arduino, o código é executado repetidamente num ciclo infinito.

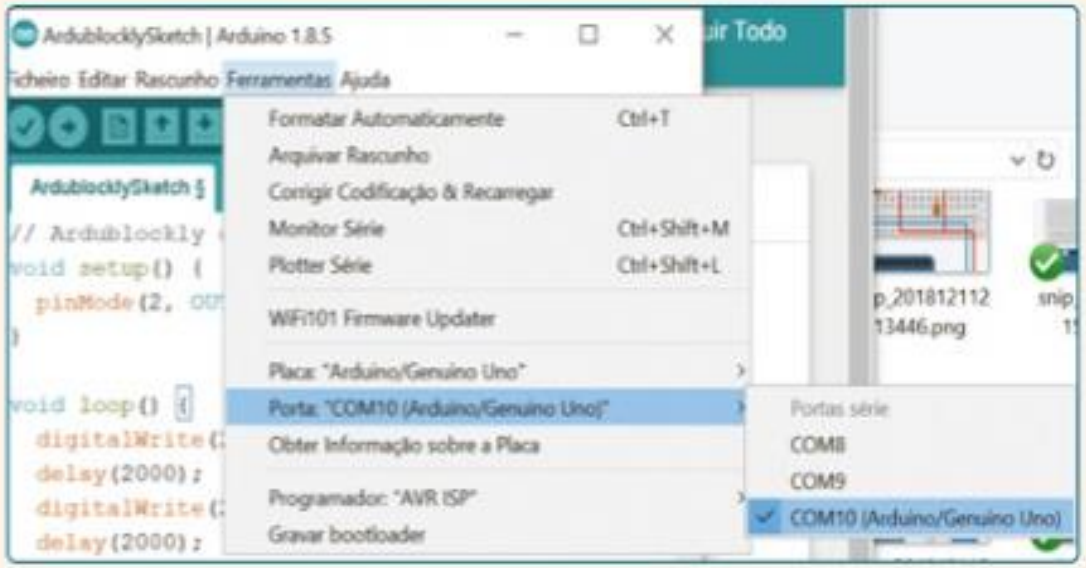
ARDUÍNO

Exemplo: Criação de um primeiro circuito (conclusão)

Depois de enviado com sucesso, na parte de baixo da aplicação deverá surgir o seguinte:

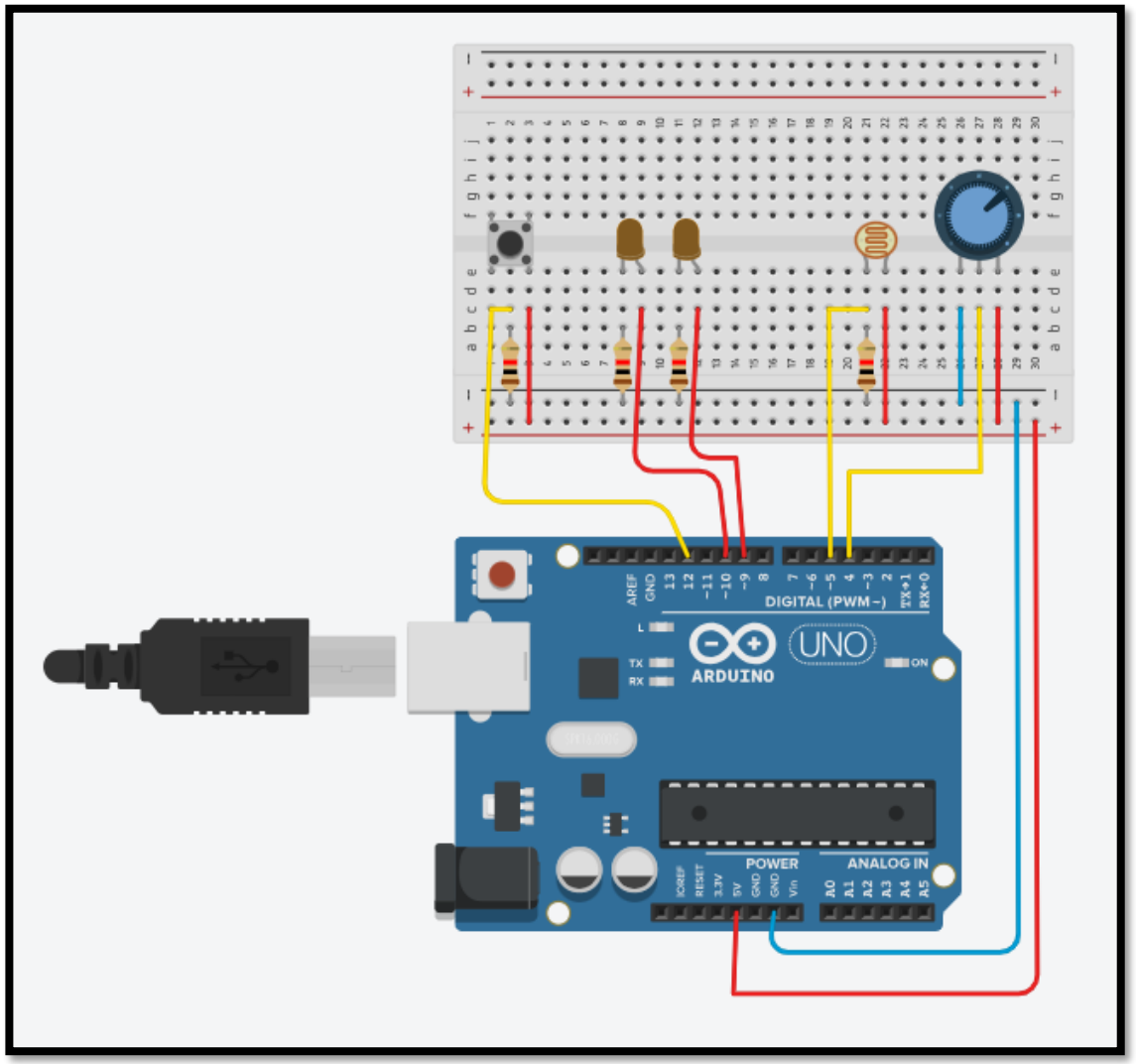
Se não acontecer, verifica se a Arduino está ligada ao computador e se a informação da ligação à placa está correta.

Em **Ferramentas**, nas opções **Placa** e **Porta**, deverá aparecer o modelo da placa Arduino que estás a usar (como na imagem ao lado).



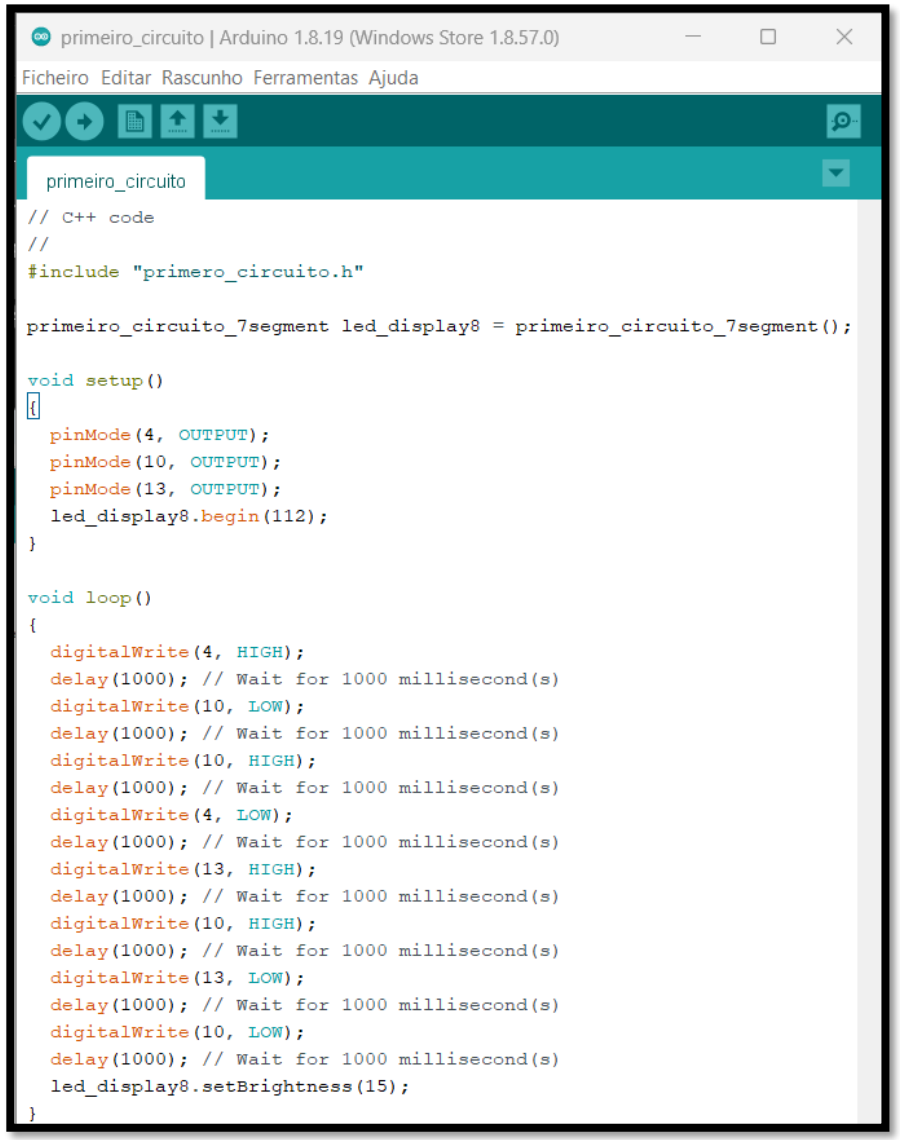
ARDUÍNO

Primeiro Circuito



ARDUÍNO

Primeiro Circuito

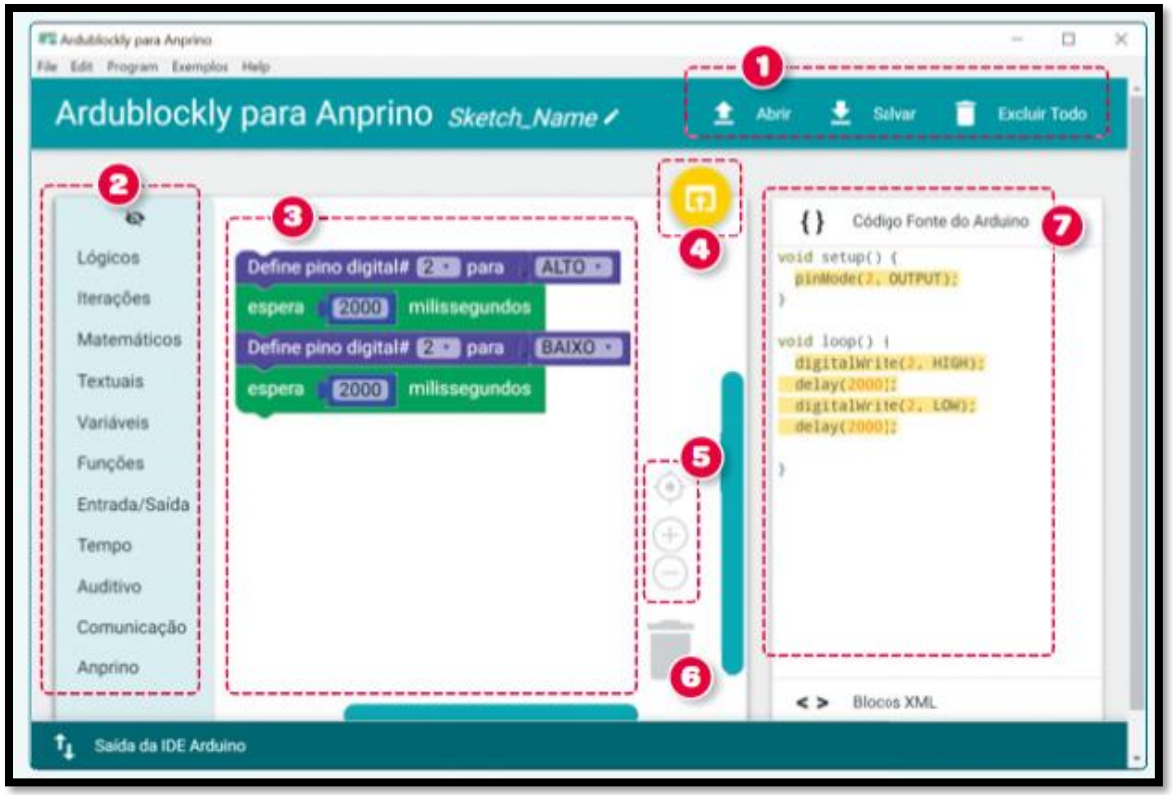


ARDUÍNO

Programar Arduino com Ardublockly

Elementos da área de trabalho do Ardublockly

1. **Comandos** – Permitem abrir um *sketch* guardado anteriormente (Abrir), guardar o *sketch* (Salvar) e eliminar o *sketch* atual (Excluir Todo).
2. **Categorias** – Mostra todos os blocos necessários para criar um *sketch*.
3. **Área de trabalho** – Área onde se programa o nosso circuito.
4. **Comandos de código** – Permite verificar o código do *sketch*, enviar o código para a Arduino IDE e enviar o código diretamente para a placa Arduino.
5. **Zoom** – Permite centrar, aumentar ou diminuir o tamanho dos blocos.
6. **Lixo** – Permite eliminar um ou mais blocos, arrastando-os para o ícone. Também os podes eliminar arrastando-os para as categorias.
7. **Código** – Mostra o código a utilizar na Arduino IDE. Podemos copiar e colar manualmente este código para a Arduino IDE.



ARDUÍNO

Categorias de blocos no Ardublockly

Categorias	Blocos
Lógicos Verificam se uma condição é verdadeira ou falsa.	se / faça; =
Iterações Criam ciclos de repetição, com um número definido de repetições ou com base numa condição.	repita 10 vez / faça; repita enquanto / faça
Matemática Permitem realizar operações matemáticas.	0; +

Categorias	Blocos
Entrada/saída Enviam e recebem sinais aos componentes ligados à placa.	Define pino digital# 1 para ALTO; Ler pino digital# 2; Define pino analógico# 3 para; Ler pino analogico# A0
Tempo Definem tempos de espera e verificam quanto tempo passou. **Nota:** 1000 ms = 1 segundo	espera 1000 milissegundos; Tempo decorrido (milissegundos)
Variáveis Gerem e utilizam variáveis.	item; definir item para

ARDUÍNO

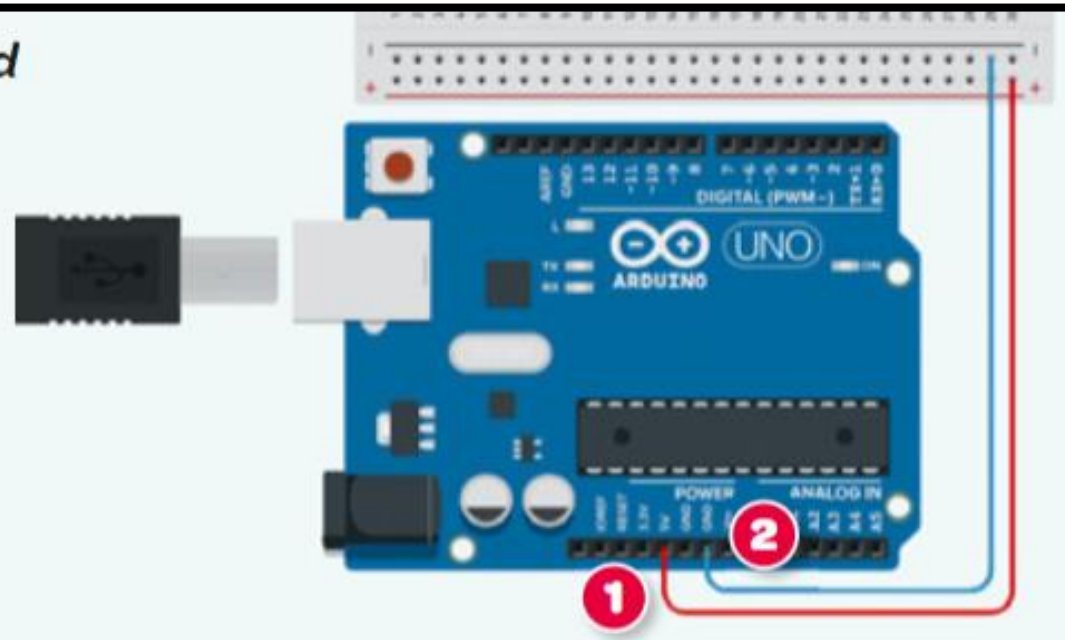
Categorias de blocos no Ardublockly

ARDUÍNO

Exemplo: Criar e programar um circuito - sirene do carro de polícia

Ligar a placa Arduino à *breadboard*

1. Liga um fio, desde o pino "5V" (carga positiva) até à linha que tem o "+" na *breadboard*.
2. Liga um fio desde o pino "GND" (carga negativa) até à linha que tem o "–" na *breadboard*.

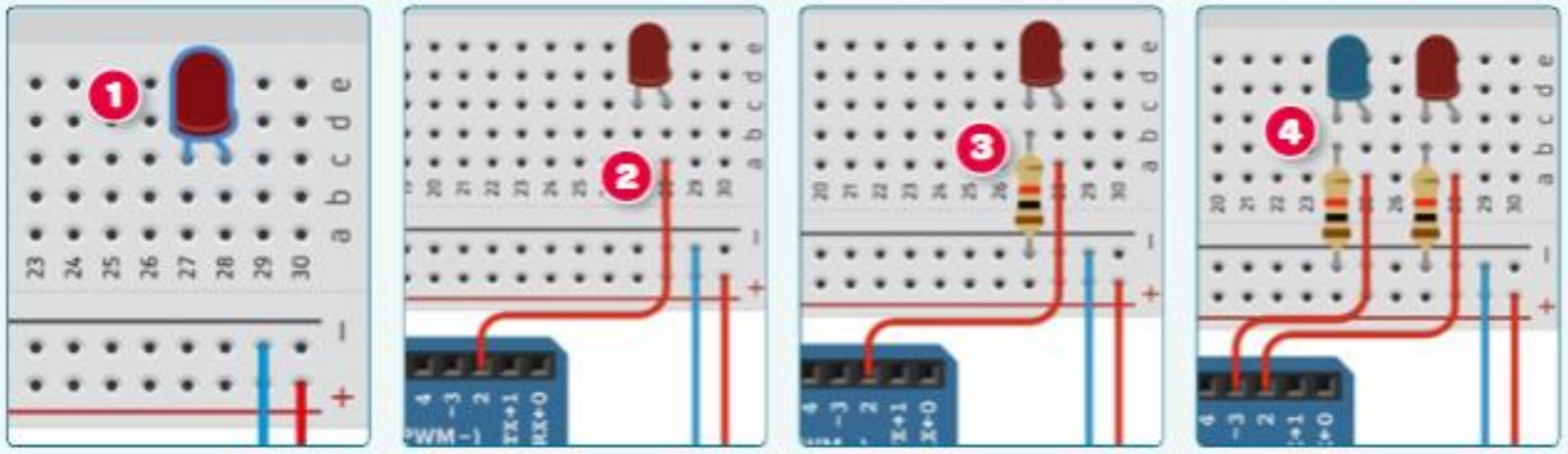


ARDUÍNO

Exemplo: Criar e programar um circuito - sirene da polícia (cont.)

Criar o circuito com um LED vermelho e outro azul

1. Coloca o primeiro LED.
2. Liga um fio entre um dos pinos de saída (por exemplo, o 2) e a coluna onde está a perna do LED com o polo positivo (a perna mais comprida).
3. Liga uma resistência de 220 Ω entre a linha da carga negativa e a coluna onde está a perna do LED com o polo negativo.
4. Repete os passos anteriores para ligar o segundo LED, mas liga o fio ao pino 3 (em vez de ao pino 2).



ARDUÍNO

Exemplo: Criar e programar um circuito - sirene da polícia (cont.)

Programar o circuito no Ardublocky

1. Liga o LED vermelho (pino 2) e desliga o LED azul (pino 3).
2. Espera 500 milissegundos (meio segundo).
3. Desliga o LED vermelho (pino 2) e liga o LED azul (pino 3).
4. Espera 500 milissegundos.

Nota: Um programa em Arduino repete para sempre a programação.

ARDUÍNO

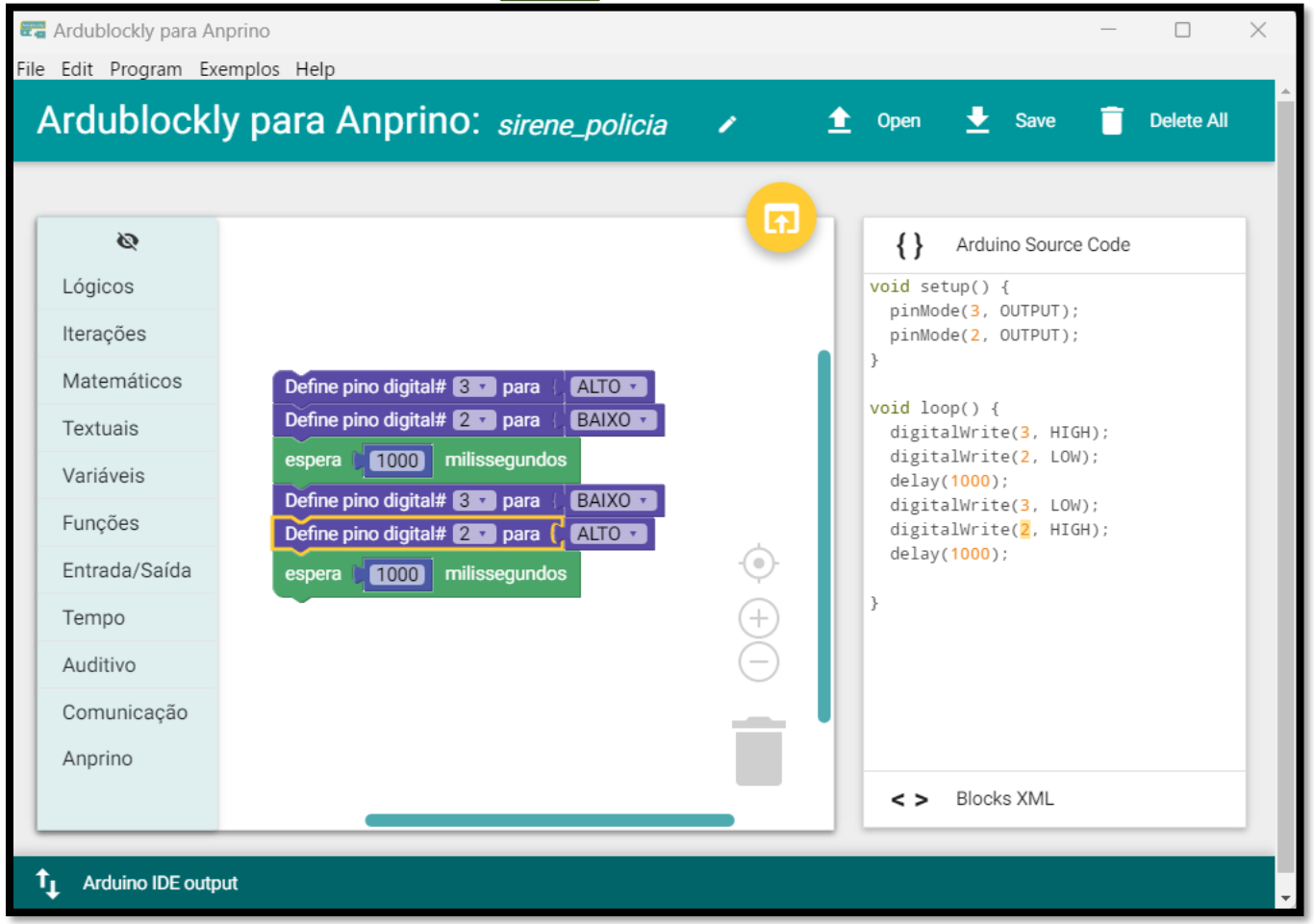
Exemplo: Criar e programar um circuito - sirene da polícia (cont.)

Enviar o programa para a placa Arduíno:

1. No Ardublockly, clica no comando para abrir o Arduíno IDE com o programa, caso contrário, abra e copie o código.
2. Na Arduíno IDE, clica no comando para transferir o código para a placa Arduíno e verificar se a programação está correta.

ARDUÍNO

1


Ardublockly para Anprino
File Edit Program Exemplos Help
Ardublockly para Anprino: sirene_policia
Open
Save
Delete All
Lógicos
Iterações
Matemáticos
Textuais
Variáveis
Funções
Entrada/Saída
Tempo
Auditivo
Comunicação
Anprino
Define pino digital# 3 para ALTO
Define pino digital# 2 para BAIXO
espera 1000 milissegundos
Define pino digital# 3 para BAIXO
Define pino digital# 2 para ALTO
espera 1000 milissegundos
Arduino Source Code
void setup() {
pinMode(3, OUTPUT);
pinMode(2, OUTPUT);
}
void loop() {
digitalWrite(3, HIGH);
digitalWrite(2, LOW);
delay(1000);
digitalWrite(3, LOW);
digitalWrite(2, HIGH);
delay(1000);
}
Blocks XML
Arduino IDE output

2


sirene_policia | Arduino 1.8.19 (Windo...
Ficheiro Editar Rascunho Ferramentas Ajuda
sirene_policia
void setup() {
pinMode(3, OUTPUT);
pinMode(2, OUTPUT);
}
void loop() {
digitalWrite(3, HIGH);
digitalWrite(2, LOW);
delay(1000);
digitalWrite(3, LOW);
digitalWrite(2, HIGH);
delay(1000);
}
Carregamento completo
O rascunho usa 972 bytes (3%) do espaço de armazena
Variáveis globais usam 9 bytes (0%) de memória dinâ
14
Arduino Uno em COM5

ARDUINO

A linguagem de programação do Arduino pode ser dividida em três partes principais: funções, valores (variáveis e constantes) e estruturas.

Funções

Servem para controlar a placa Arduino e realizar programação.

Entradas e Saídas Digitais	Entradas e Saídas Avançadas
digitalRead() digitalWrite() pinMode()	noTone() pulseIn() pulseInLong() shiftIn() shiftOut() tone()
Entradas e Saídas Analógicas	
analogRead() analogReference() analogWrite()	

ARDUINO

Funções

Funções Trigonométricas	Funções Matemáticas	Funções Temporizadoras	Comunicação	USB
cos() sin() tan()	abs() constrain() map() max() min() pow() sq() sqrt()	delay() delayMicroseconds() micros() millis()	Serial Stream	Keyboard Mouse

ARDUINO

Funções

Números Aleatórios random() randomSeed()	**Apenas Zero, Due e Família MKR** analogReadResolution() analogWriteResolution()	**Interrupções Externas** attachInterrupt() detachInterrupt()
Bits e Bytes bit() bitClear() bitRead() bitSet() bitWrite() highByte() lowByte()	**Caracteres** isAlpha() isAlphaNumeric() isAscii() isControl() isDigit() isGraph() isHexadecimalDigit() isLowerCase() isPrintable() isPunct() isSpace() isUpperCase() isWhitespace()	**Interrupções** interrupts() noInterrupts()

ARDUINO

Variáveis

Tipos de dados e constantes da linguagem Arduino

Constantes	Conversão
HIGH | LOW INPUT | OUTPUT | INPUT_PULLUP LED_BUILTIN true | false Constantes de Ponto Flutuante Constantes Inteiras	byte() char() float() int() long() word()
Escopo de Variáveis e Qualificadores const escopo static volatile	**Utilitários** PROGMEM sizeof()

ARDUINO

Variáveis

Tipos de dados e constantes da linguagem Arduino

Tipos de Dados

bool
boolean
byte
char
double
float
int
long
short
size_t
string
String()
unsigned char
unsigned int
unsigned long
vetor
void
word

ARDUINO

Estruturas

Os elementos da linguagem Arduino (C++).

Sketch	**Operadores para Ponteiros**	**Operadores Boleanos**
loop() setup()	& (referência) * (desreferência)	! (NÃO lógico) && (E lógico) || (OU lógico)
Operadores Aritméticos	**Operadores de Comparação**	**Operadores Bitwise**
% (resto) * (multiplicação) + (adição) - (subtração) / (divisão) = (operador de atribuição)	!= (diferente de) < (menor que) <= (menor que ou igual a) == (igual a) > (maior que) >= (maior que ou igual a)	& (E) << (deslocamento à esquerda) >> (deslocamento à direita) ^ (OU EXCLUSIVO) | (OU) ~ (NÃO)

ARDUINO

Estruturas

Estruturas de Controle

break
continue
do...while
else
for
goto
if
return
switch...case
while

Outros Elementos da Sintaxe

#define (define)
#include (include)
/* */ (comentário em bloco)
// (comentário)
; (ponto e vírgula)
{} (chaves)

Operadores de Atribuição Composta

%= (compound remainder)
&= (atribuição por e)
*= (atribuição por multiplicação)
++ (incremento)
+= (atribuição por adição)
-- (decremento)
-= (atribuição por subtração)
/= (atribuição por divisão)
^= (atribuição por ou exclusivo)
|= (atribuição por ou)

SÍNTESE

- O que é uma placa Arduíno?
- Como programar o Arduíno?

Esta proposta educativa foi traduzida e/ou adaptada dos projetos, manual digital 6.º Login e Arduíno Education

Link de projetos no TinKerCAD
https://www.tinkercad.com/projects/Building-a-Simple-Electronic-Piano-Using-Tinkercad