

PROF. ME.
HÉLIO
ESPERIDIÃO

INTRODUÇÃO AO ARDUINO

Qual a diferença entre Microprocessador e Microcontrolador mesmo?

Ambos realizam algumas operações que são, buscar as instruções da memória e executar estas instruções (operações aritméticas ou lógicas) e o resultado dessas execuções são usados para servir a dispositivos de saída. As instruções eletrônicas representados por um grupo de bits são sempre obtido a partir de sua área de armazenamento, que chamamos de memória.

Então recapitulando, o microprocessador é um circuito integrado responsável pelo processamento de dados, como uma unidade lógica e aritmética, com diversos registradores especiais, mas precisa receber ordens externas e ter outros componentes externos para funcionar.

Já o microcontrolador é um microprocessador, memória RAM, memória ROM, temporizadores, contadores, porta serial, conversores e portas de I/O em um só circuito integrado, ou seja, um microcomputador-de-um-só-chip.

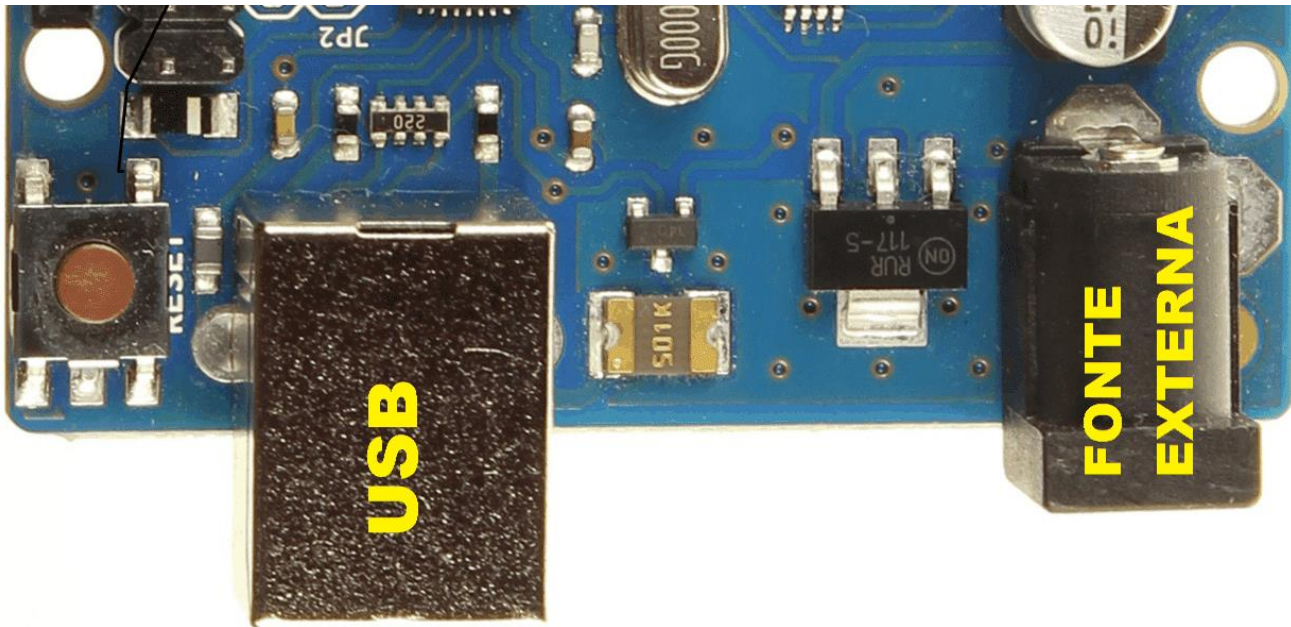
O Arduino é um Microprocessador ou Microcontrolador?

A placa do Arduino tem um microcontrolador, pois dentro dele já existe as memórias e conversores. Mas muitas vezes o espaço de memória interno de um microcontrolador é pequeno, então é adicionado uma expansão, através de um circuito integrado de memória ou um cartão micro SD. Isso não quer dizer que ele não é mais um microcontrolador, ele apenas recebeu um melhoramento de uma função.



ATmega328P

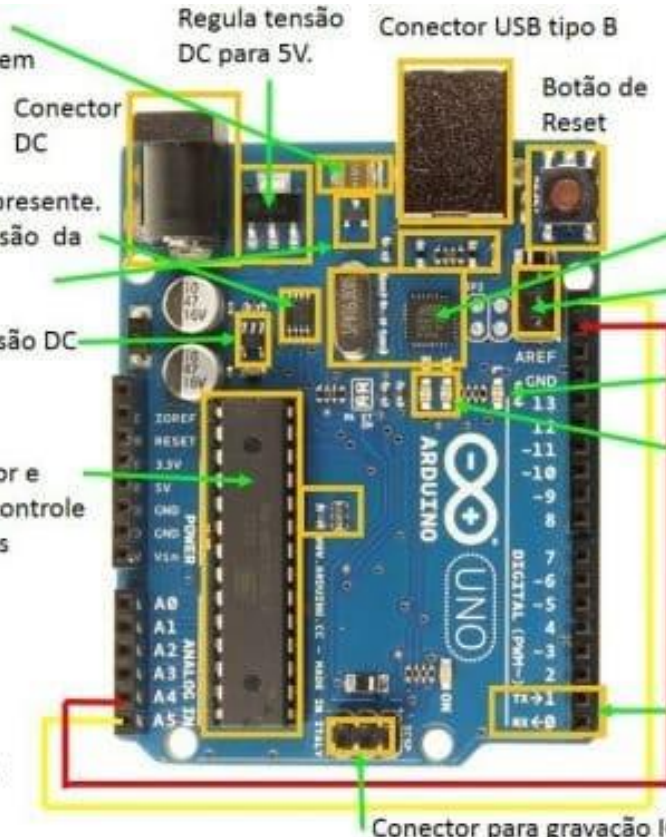
**O Arduino é uma plataforma e
Não um microcontrolador.**



Alimentação da placa Arduino UNO

A ALIMENTAÇÃO EXTERNA DEVE ESTAR ENTRE OS LIMITES 7V. A 12V.

Impede que a USB do computador seja danificada em caso de sobrecorrente. (acima de 500 mA)



Compara se a tensão DC está presente. Se não estiver, deixa que a tensão da USB Alimente o circuito.

Regula a tensão DC para 3,3 V.

Conjunto microcontrolador e cristal, responsável pelo controle e leitura de todos os pinos da placa .

Conjunto microcontrolador e cristal que faz a interface USB com o computador

Conector para gravação ICSP, do ATMEGA16U2

Led conectado ao pino 13 do arduino

Leds de status da comunicação serial Entre placa e computador

Os sinais em amarelo e vermelho Indicam dois pinos que estão em curto

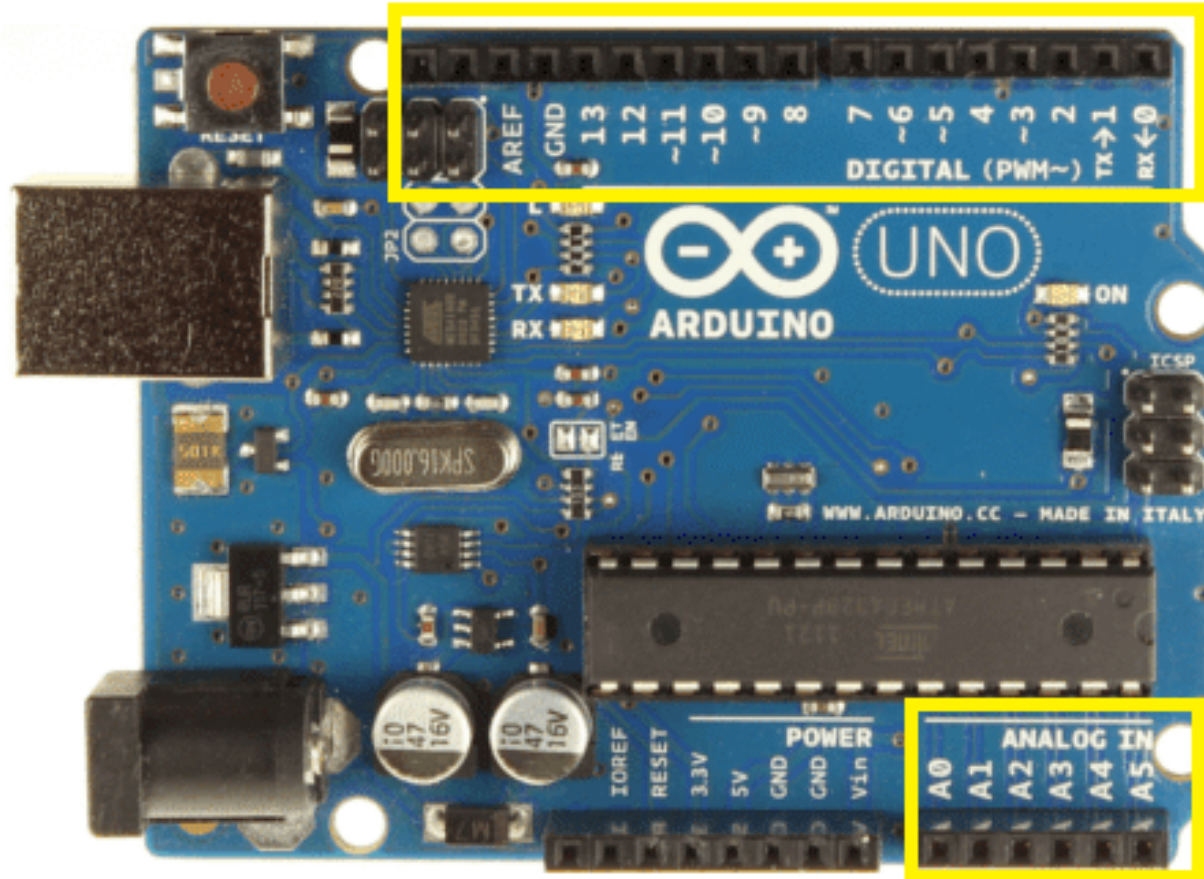
Caso utilize esses sinais no projeto, tome cuidado pois estão conectados ao outro microcontrolador para gravação

Conector para gravação ICSP do ATMEGA328



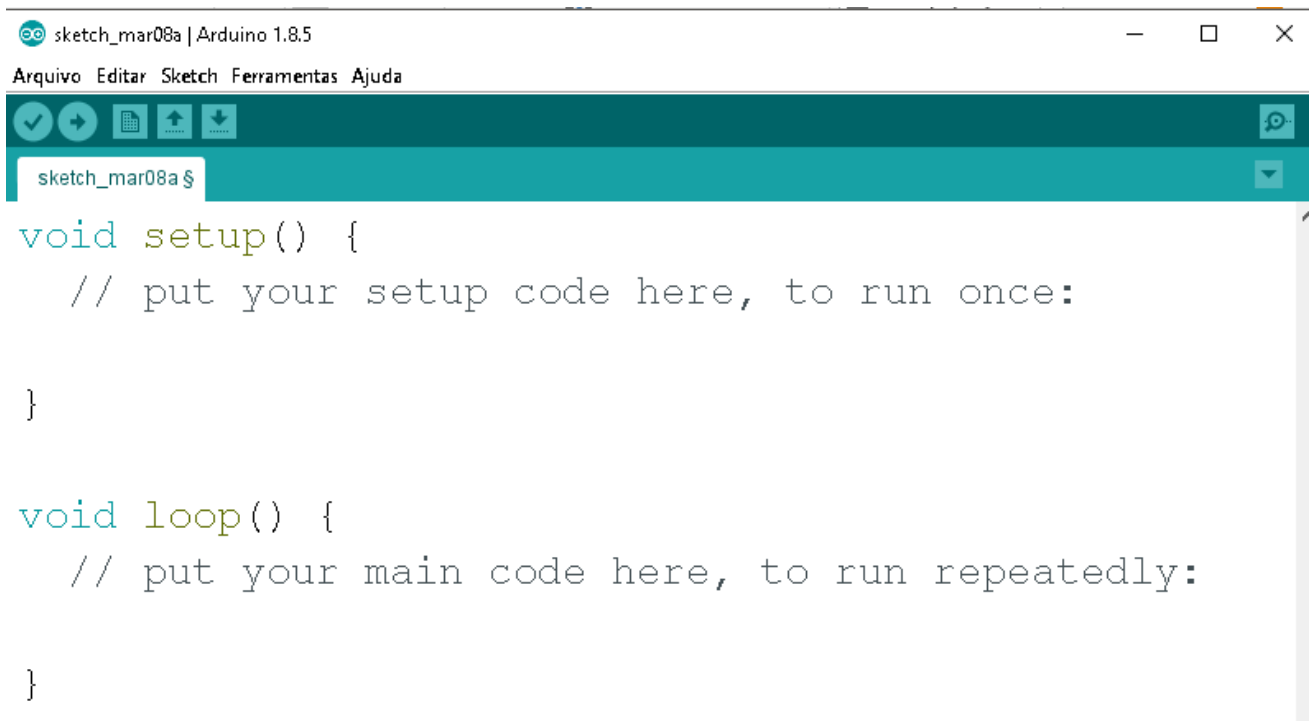
Principais portas

Entradas / Saídas Digitais



Entradas Analógicas

Entradas e saídas do Arduino UNO



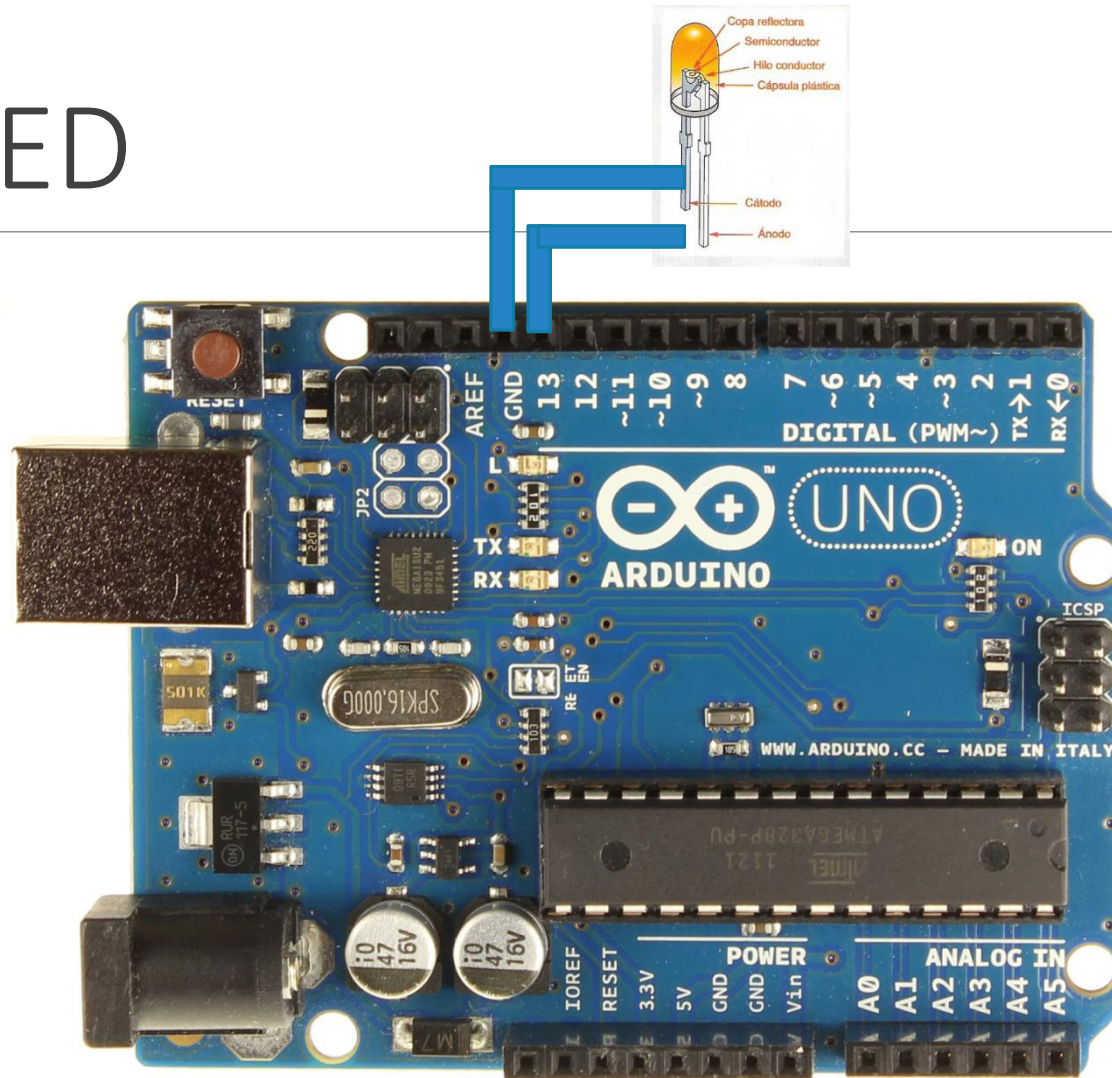
The image shows a screenshot of the Arduino IDE interface. The window title is "sketch_mar08a | Arduino 1.8.5". The menu bar includes "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". The toolbar contains icons for a checkmark, a right arrow, a document, an upload arrow, a download arrow, and a play button. The main editor area shows the following code:

```
sketch_mar08a $  
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

Ambiente de desenvolvimento

Ligar um LED

SEMPRE ligue o LED através de um resistor (tipicamente 470ohms).




```
sketch_mar08a | Arduino 1.8.5
Arquivo Editar Sketch Ferramentas Ajuda
sketch_mar08a $
int led = 13;

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

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

Determine que o pino led será de saída

Envia sinal alto para a porta 13

Dorme 1000 milisegundos

Envia sinal baixo para a porta 13

Dorme 1000 milisegundos

Programação no Arduino.

A linguagem de programação utilizado pelo sistema Arduino é a linguagem c.

boolean: valor verdadeiro (true) ou falso (false)	char: um caractere	byte: um byte, ou sequência de 8 bits	int: número inteiro de 16 bits com sinal (-32768 a 32767)
unsigned int: número inteiro de 16 bits sem sinal (0 a 65535)	long: número inteiro de 16 bits com sinal (-2147483648 a 2147483647)	unsigned long: número inteiro de 16 bits sem sinal (0 a 4294967295)	float: número real de precisão simples (ponto flutuante)
	double: número real de precisão dupla (ponto flutuante)	string: sequência de caracteres	

Tipo de Dado

$+$: adição ("mais")

$-$: subtração ("menos")

$*$: multiplicação ("vezes")

$/$: divisão ("dividido por")

Operadores
aritméticos:

&&: conjunção ("e")

||: disjunção ("ou")

==: igualdade ("igual a")

!=: desigualdade ("diferente de")

!: negação ("não")

>: "maior que"

<: "menor que"

>=: "maior ou igual a"

<=: "menor ou igual a"

Operadores
lógicos:

While

O while é uma estrutura que executa um conjunto de comandos repetidas vezes enquanto uma determinada condição for verdadeira. While em inglês quer dizer "enquanto", e pronuncia-se "uái-ou". Ele segue o seguinte formato:

```
while(condição) {  
    ...  
}
```

sketch_mar08a \$

```
int i = 0;
void loop() {
    digitalWrite(led, HIGH);
    delay(1000);
    digitalWrite(led, LOW);
    delay(1000);
    i++;
    if(i == 3) {
        delay(5000);
        i = 0;
    }
}
```

if

ANALISE O CÓDIGO E DETERMINE
SUA SAÍDA

```
int led = 13;
```

```
void setup() {
```

```
    pinMode(led, OUTPUT);
```

```
}
```

```
void loop() {
```

```
    for(i = 0; i < 3; i++) {
```

```
        digitalWrite(led, HIGH);
```

```
        delay(1000);
```

```
        digitalWrite(led, LOW);
```

For

Entradas Analógicas

analogRead(Pino)

Lê entrada analógica 0-5V transformando em 10 bit's

Pinos analógicos podem ser usados como porta digitais usando a função `pinMode()`, quando usado como porta analógica não necessitam de configuração.

Funções Matemáticas e de tempo

`delay(t)` - O programa tem uma pausa de t milissegundos (1000 milissegundos = 1 segundo)

`delayMicroseconds(t)` - O programa tem uma pausa de t microssegundos

`millis()` - Retorna o tempo, em milissegundos, desde que o programa começou a rodar

`randomSeed(referência)` - Gera uma referência para o primeiro número aleatório (Função setup)

`random(min,max)` - Gera um valor pseudo aleatório int entre min e max (a função acima é necessária)

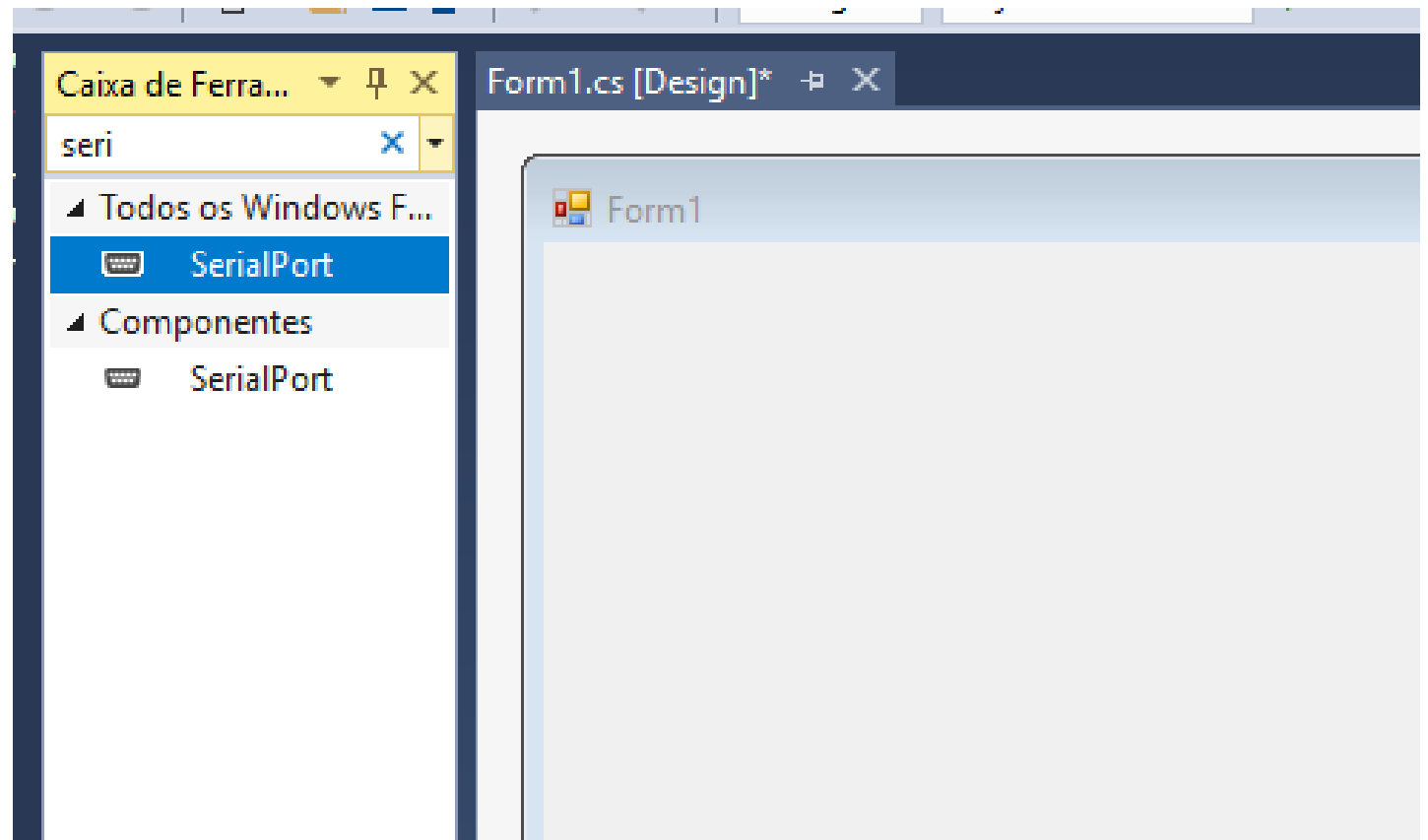
`abs(x)` - Retorna o módulo (valor absoluto) do número real passado como parâmetro

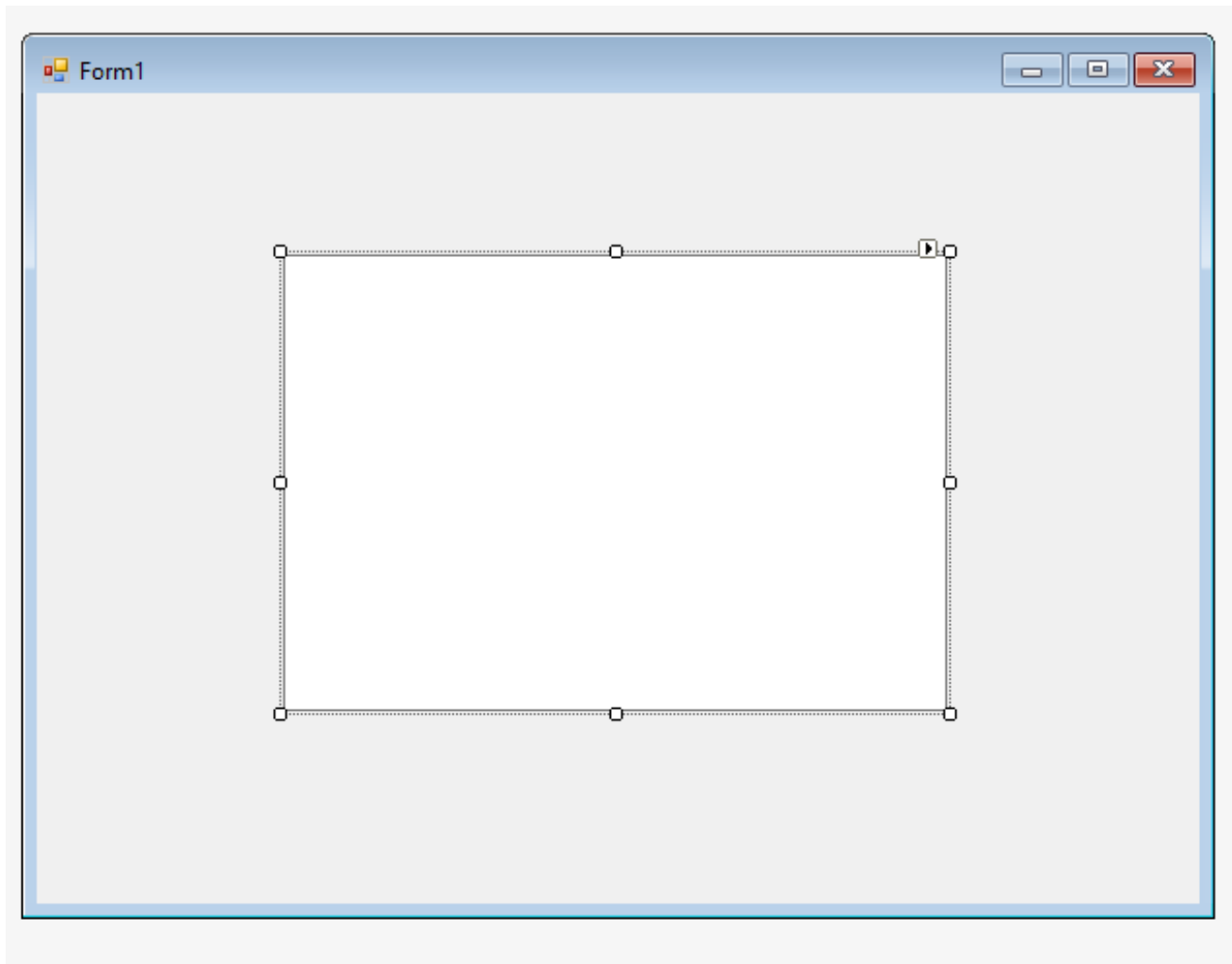
`sin(x)` - Retorna o seno de x(rad)

COMUNICAÇÃO SERIAL C#

Componente Serial Port

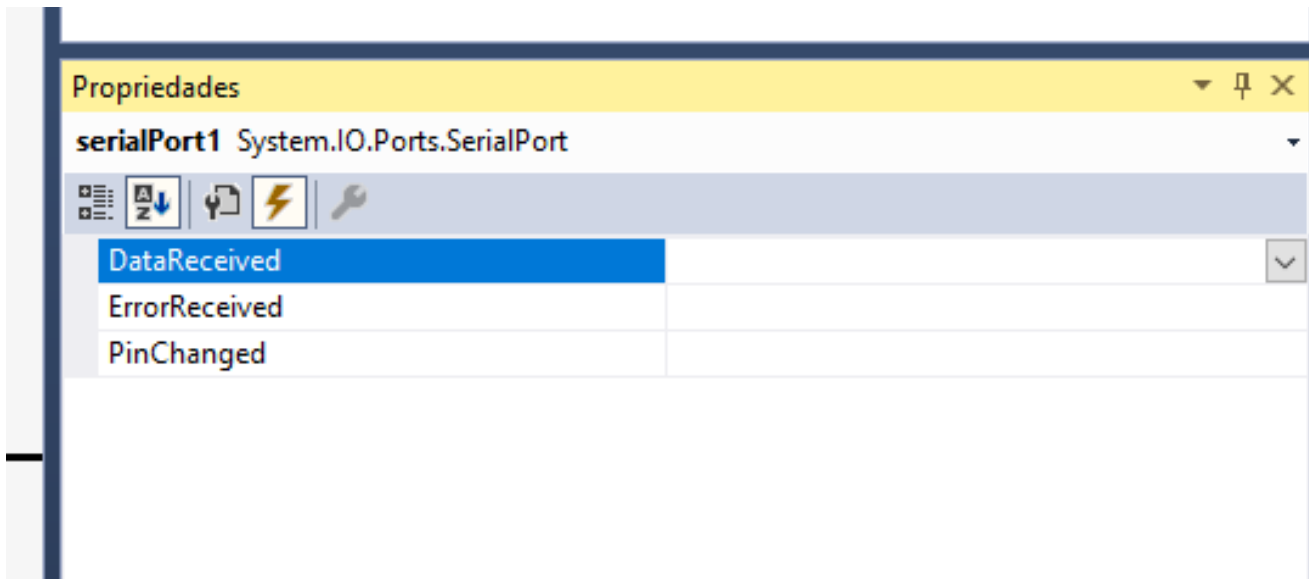
Permite que seja possível a
comunicação entre o Arduino e o C#.





Utilize uma
Caixa de texto
que permita
múltiplas linhas

Evento do componente Serial Port



```
private void serialPort1_DataReceived(object sender, System.IO.Ports.SerialPortDataReceivedEventArgs e)
{
    this.dados = serialPort1.ReadLine();
    this.existeDado = true;
}
```

```
private void Form1_Load(object sender, EventArgs e)
{
    serialPort1.BaudRate = 9600;
    serialPort1.PortName = "COM4";
    serialPort1.Open();
}
```

```
private void timer1_Tick(object sender, EventArgs e)
{
    if (this.existeDado) {
        this.existeDado = false;
        textBox1.Text += dados;
    }
}
```

```
public partial class Form1 : Form
{
    public String dados;
    public Boolean existeDado=false;
    public Form1()
    {
        InitializeComponent();
    }
}
```

```
int LED=13;
void setup() {
  pinMode(LED,13);
  Serial.begin(9600);
}
void loop() {
  digitalWrite(LED,HIGH);
  delay(1000);
  Serial.write("1\n");
  digitalWrite(LED,LOW);
  delay(1000);
  Serial.write("0\n");
}
```

HIGH, LOW


```
int TRIG=13;
int ECHO=12;
void setup() {
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
  Serial.begin(9600);
}
void loop() {
  digitalWrite(TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  unsigned long duration= pulseIn(ECHO, HIGH);
  int distance= duration/29/2;
  Serial.println(distance);
  delay(500);
}
```

Calculando a
distância.