CURSO AVANCADO DE ARDUNO MINISTRANTE: GUILHERME MIRANDA

CONFXÃO CHE LARDUNO

COMUNICAÇÃO SERIAL

Sabemos que o processo de comunicação serial utilizando a plataforma arduino é bastante difundida para vários trabalhos e protótipos que utilizam desse microcontrolador.

A comunicação serial nessa plataforma acontece pelos pinos RX, TX e também pelo interface de comunicação USB do Arduino.

O processo de gravação do ATmega328 também acontece por comunicação serial.

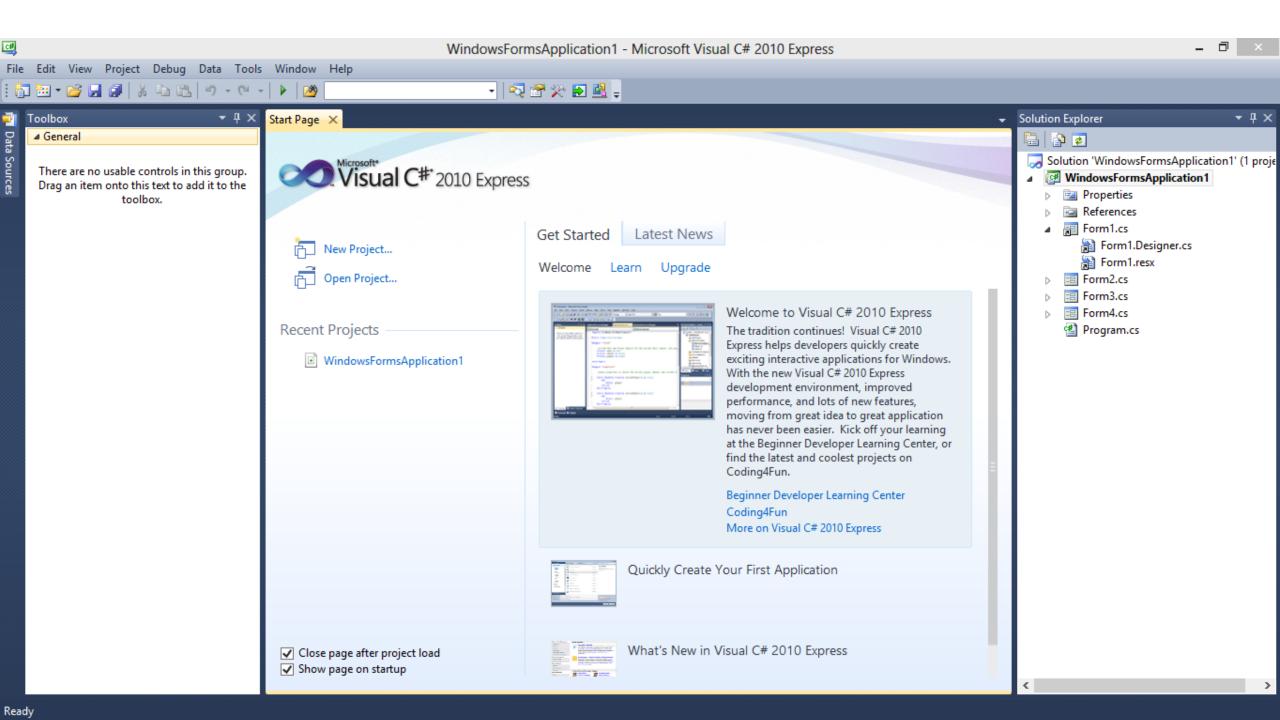
INTRODUÇÃO C#

Definição de Classes e Objetos:

- Classe ou Estrutura é o modelo que especifica o que cada tipo pode fazer.(grupo maior)
- Um objeto é basicamente um bloco de memória que foi atribuído e configurado de acordo com o modelo.(grupo menor)
- Um programa pode criar vários objetos da mesma classe.

FERRAMENTA DE DESENVOLVIMENTO

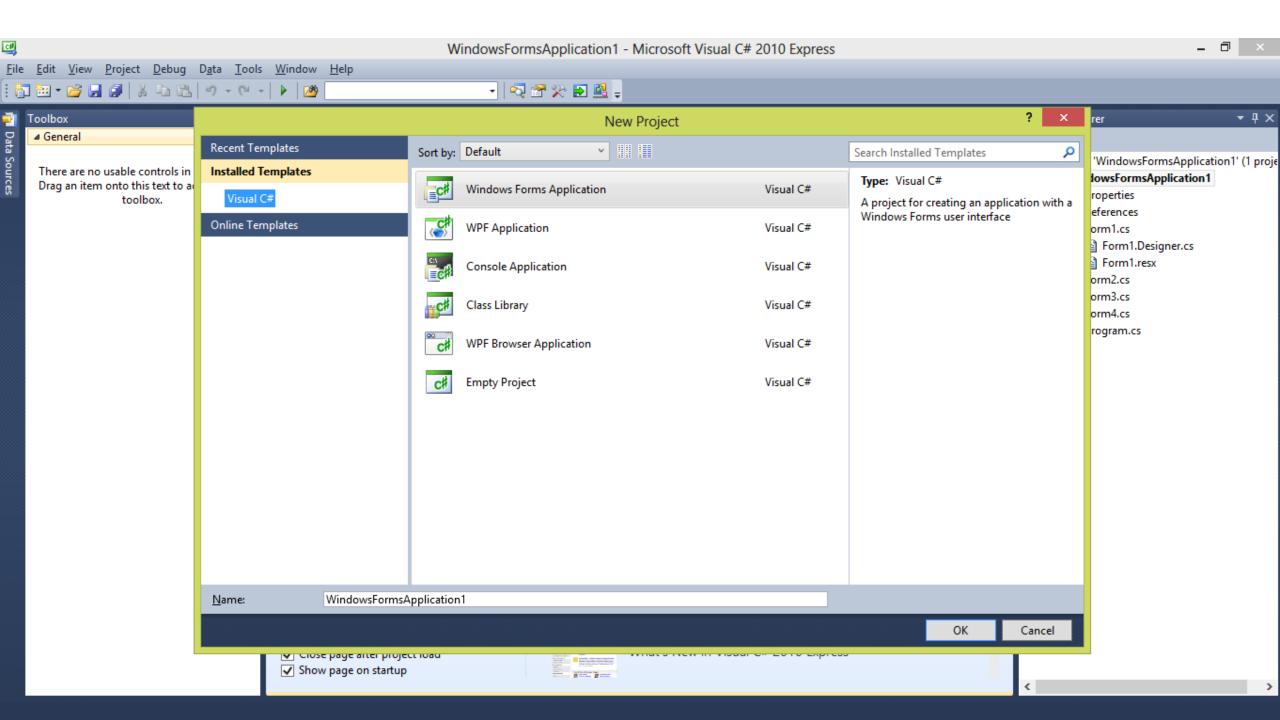
Usaremos a ferramenta Visual Studio C# 2010 para desenvolver o aplicativo

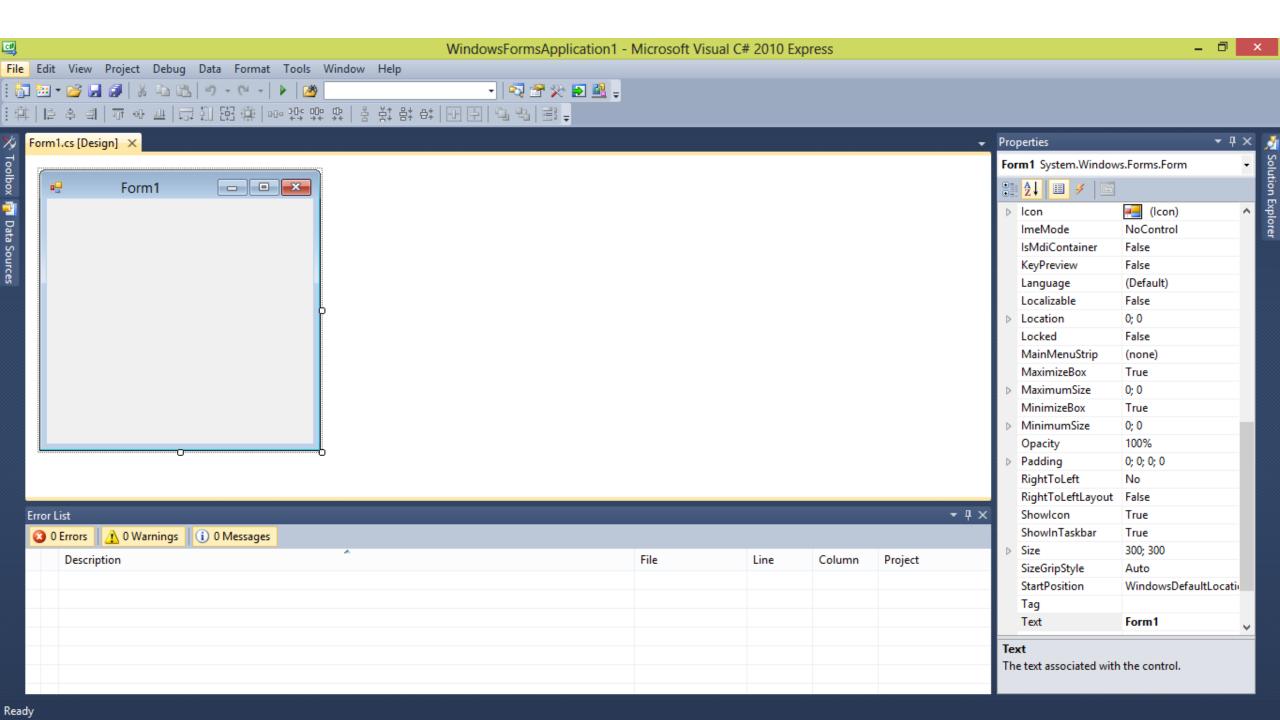


CRIAÇÃO DE UM NOVO PROJETO

Vamos criar um novo Projeto

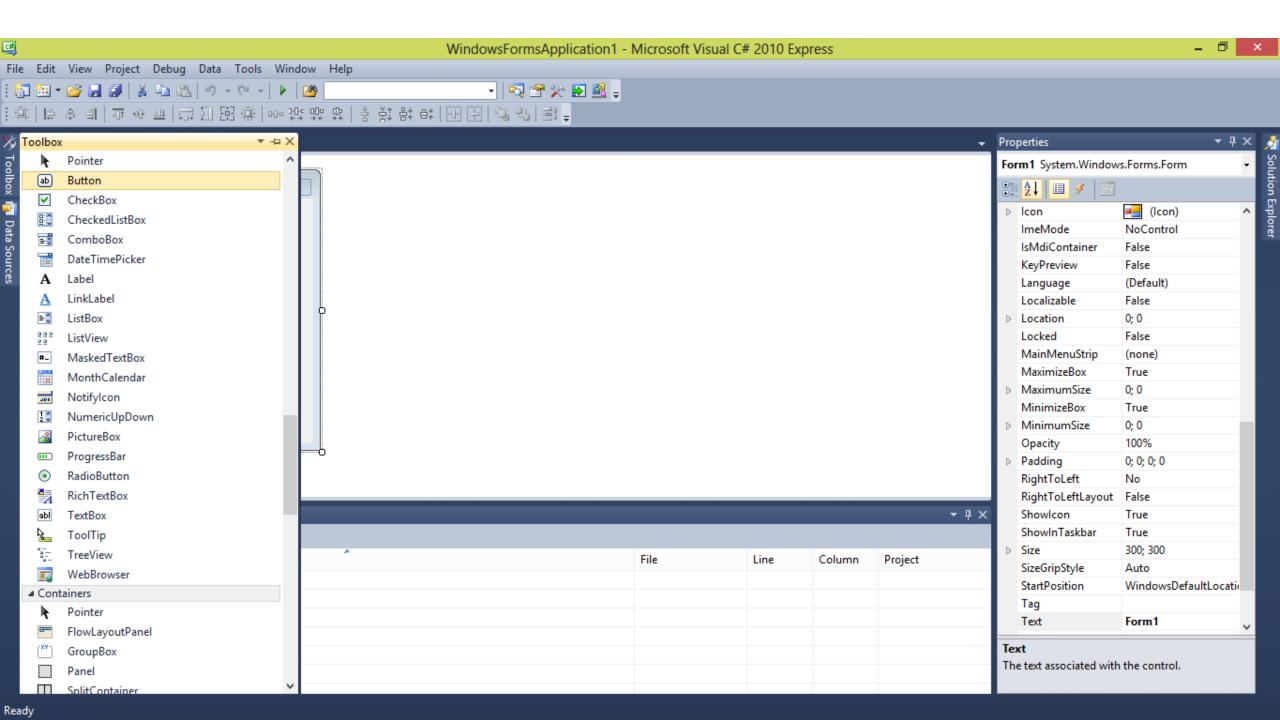
E logo em seguida, vamos escolher uma aplicação do tipo: Windows Forms Application

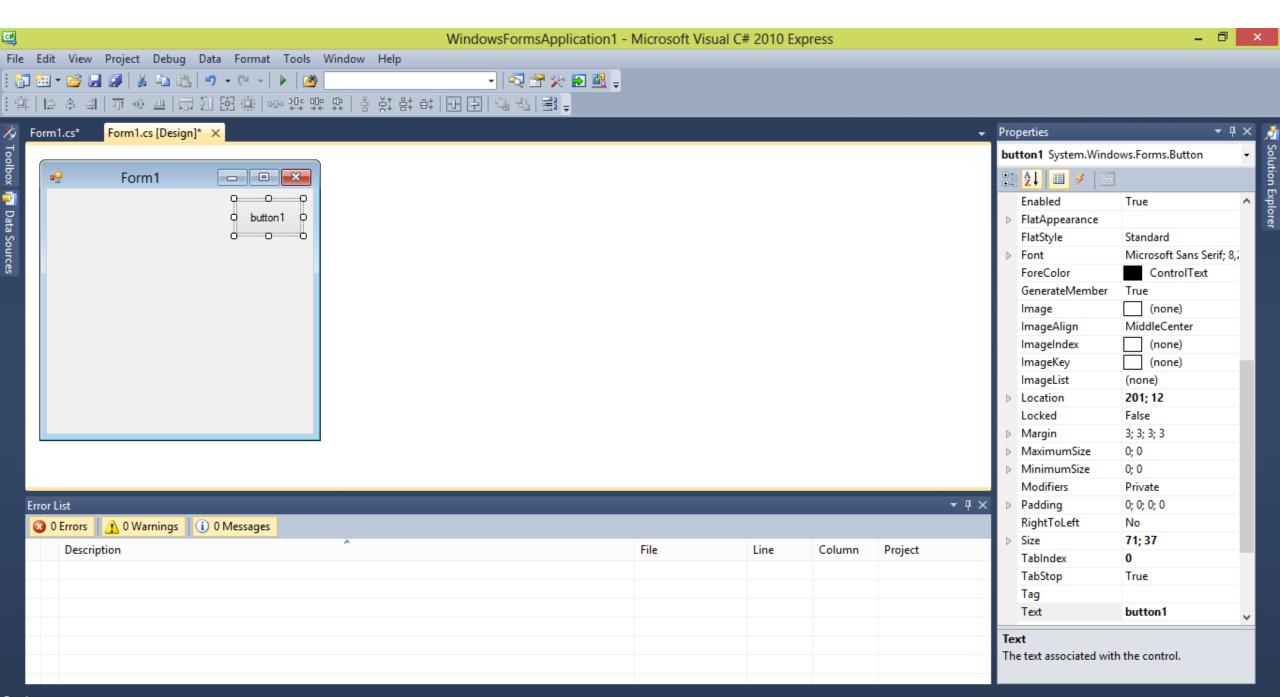




Vamos abrir a Toolbox, dentre todos, escolhemos a opção :buttons.

Ao voltar a tela principal, clicamos e definimos o tamanho e o local de onde vamos posicionar os botões



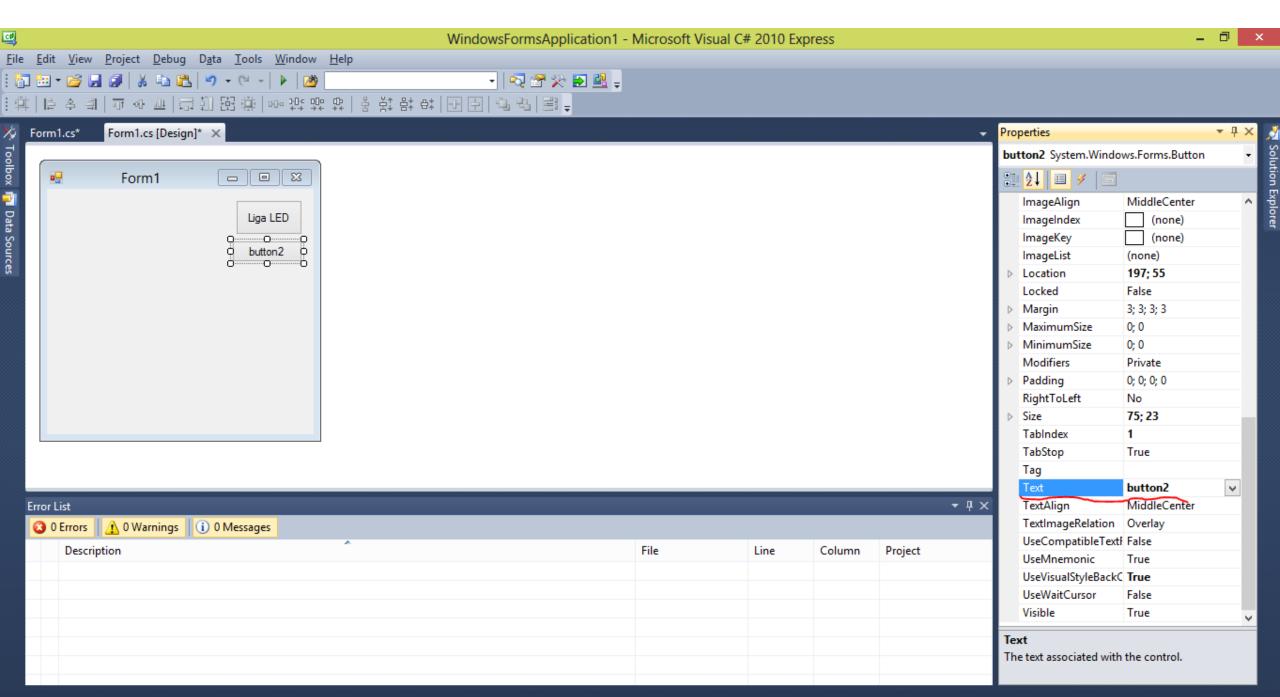


Podemos ver, do lado esquerdo um quadro escrito um quadro escrito "properties"

Nele, conseguimos definir as propriedades de cada objeto de nossa interface

Nesse caso, vamos mudar o texto de dentro do botão Para: "liga Led"

E adicionamos mais um para ser o :"Desliga Led"



Agora, clicando duas vezes algum botão vamos conseguir abrir o código do nosso programa.

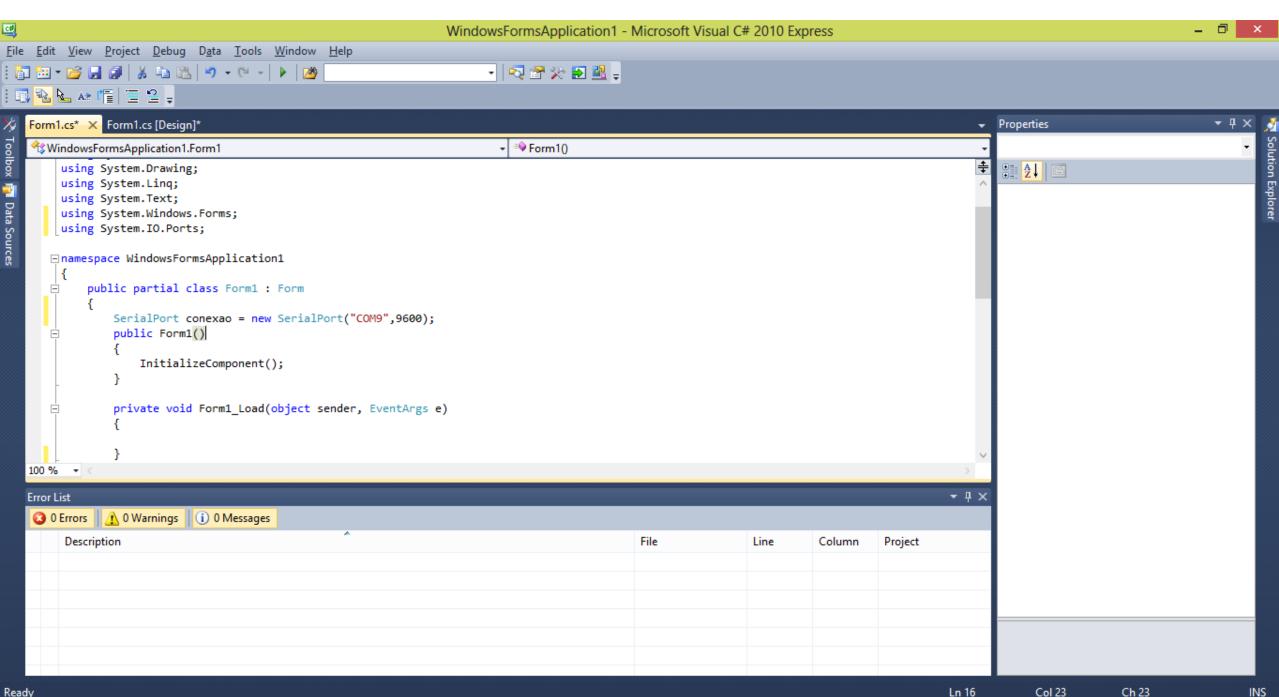
Com o código aberto, vamos adicionar a biblioteca:

using System.IO.Ports;

E dentro do Public Partial Class, vamos criar um Objeto da classe Serial com o nome de conexao:

SerialPort conexao = new SerialPort("COM9",9600);

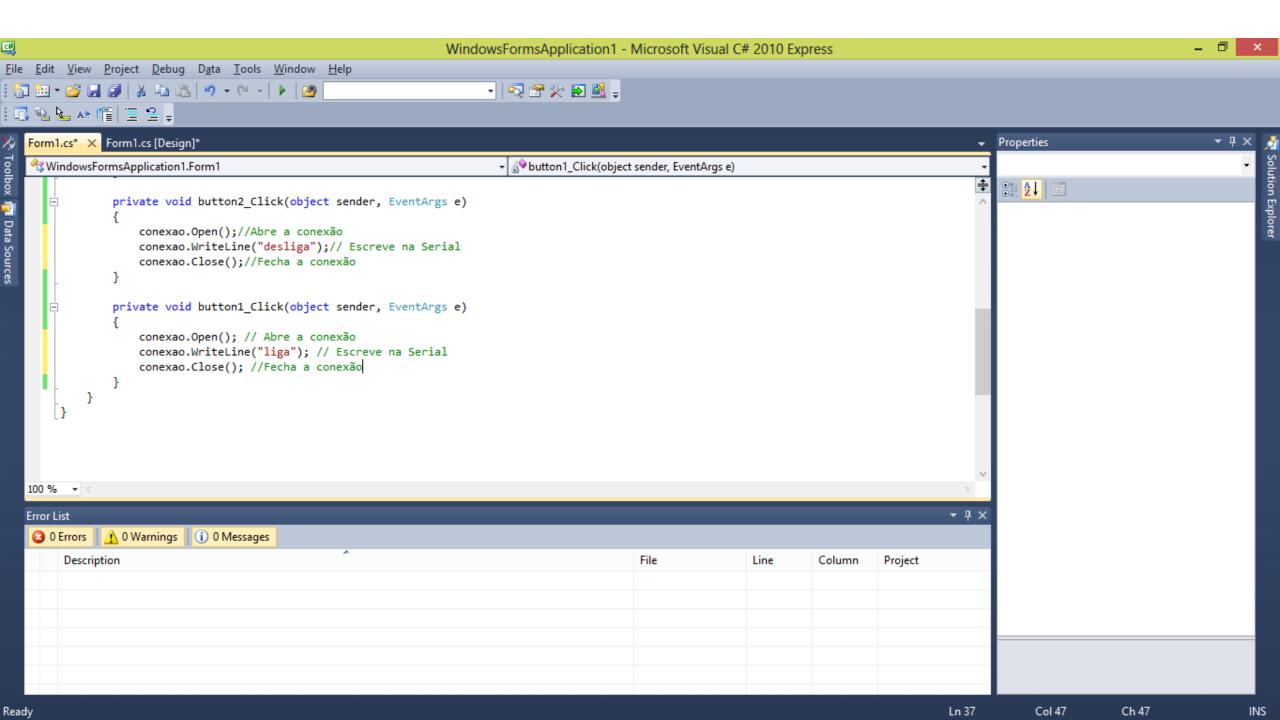
Nesse Objeto, definimos dois parâmetros : A porta, e a velocidade da conexão.



Ln 16 Col 23 Ch 23 Ready

PROGRAMANDO O BOTÃO

```
Dentro da função Button vamos escrever:
conexao.open();
conexao.write("Desliga"); ou conexao.write("Liga"); Depende de qual botão estiver Programando.
conexao.close;
```



O CÓDIGO DO ARDUINO:



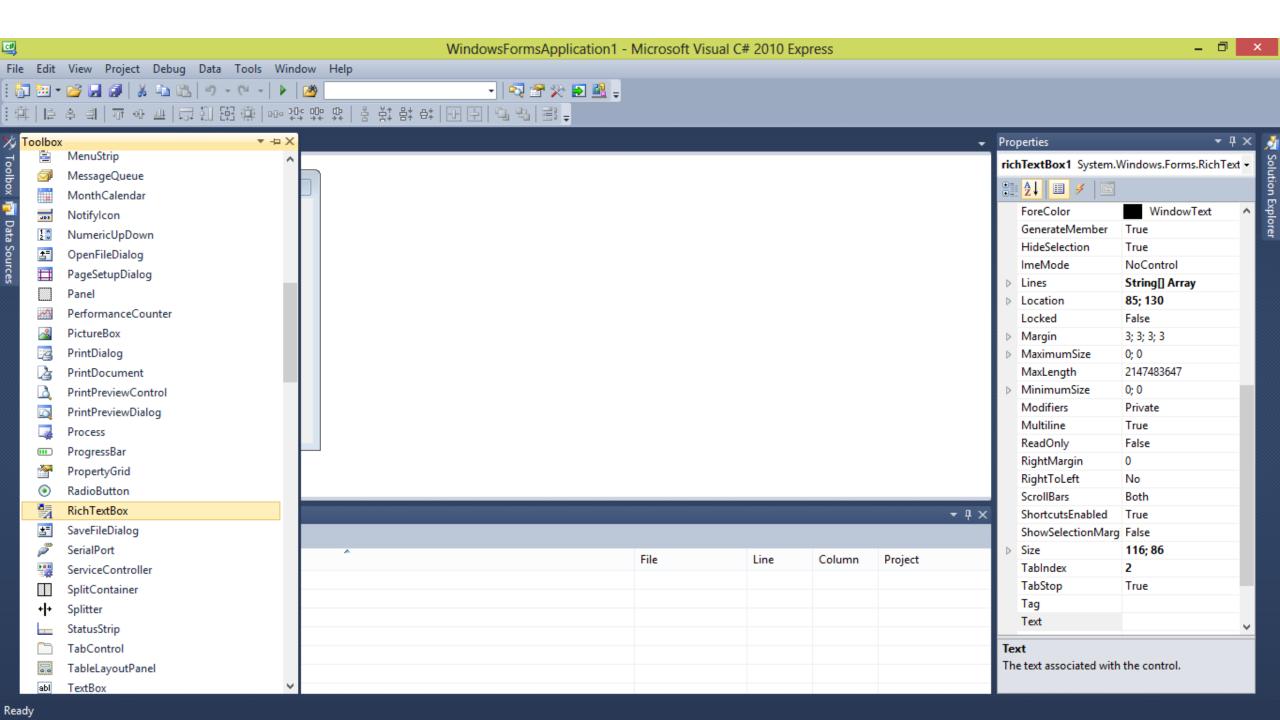
MODO DE VISUALIZAÇÃO DE TEXTO UTILIZANDO C#

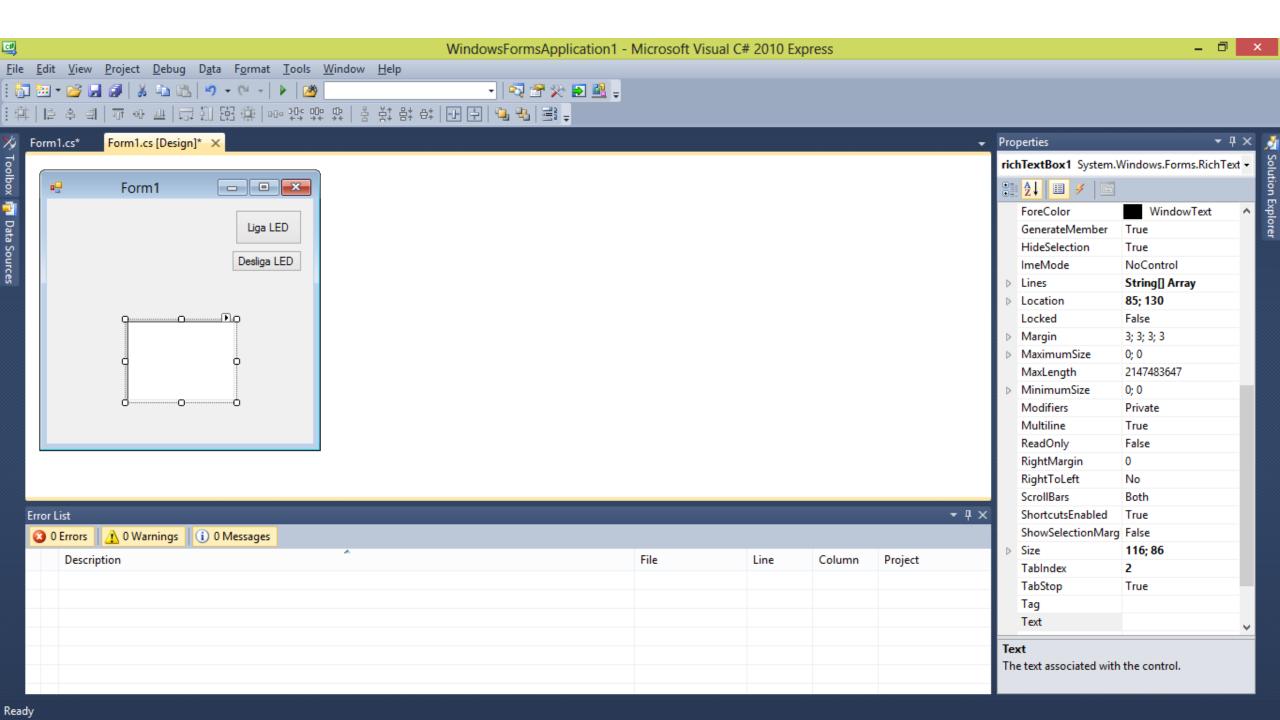
ADICIONANDO UM VISUALIZADOR DE TEXTO

Agora, vamos adicionar uma Richtextbox para visualizarmos o que é escrito na porta Serial.

Novamente, abrimos a Toolbox, e agora escolhemos a opção RichTextbox.

Após isso, voltamos a tela principal, e disponibilizamos a Richtextbox





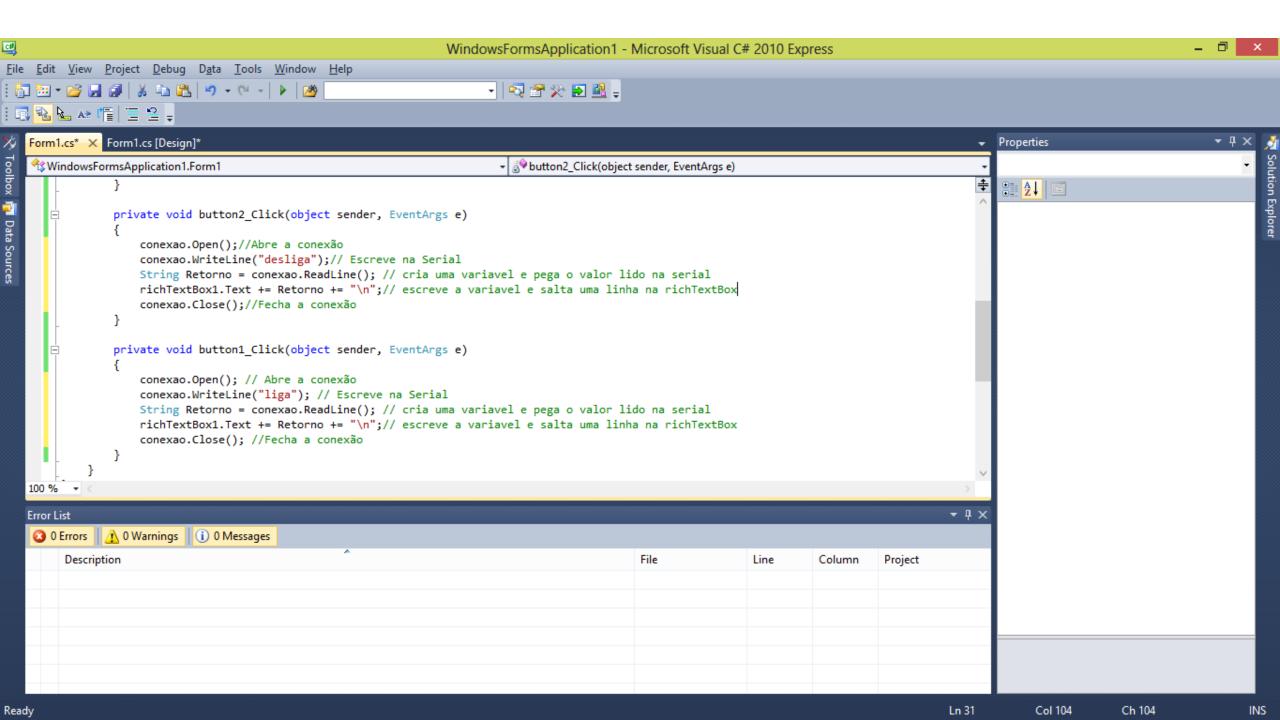
PROGRAMAÇÃO ESCRITA

Agora, vamos programar a escrita na Richtextbox.

Primeiro, em cada botão, vamos declarar uma String

Depois, igualar a string a conexao. Readline (leitura da ultima linha do nosso objeto conexao)

Após isso, para escrever usamos : richtextbox.Text += string (isso nos concatenamos a String anterior ao texto que já tinha na richtextbox.



CONEXÃO HTML, PHP E SERIAL

Pelo fato de que a porta serial de um computador fica armazenado em um arquivo, podemos manipular isso com várias programações.

E é possível criar uma aplicação para controlar o arduino através de uma pagina web.

Para isso, vamos utilizar programação HTML, PHP, e um servidor Web chamado XAMPP.

Primeiro passo, instalar o XAMPP

Logo em seguida, abrir o painel de controle do XAMPP e ligar as aplicações do APACHE e do banco de dados MySQL.

Após isso, vamos ao local onde instalamos o servidor, abrir a sua pasta e procurar por uma pasta Htdocs.

Essa pasta é onde o servidor mantem os arquivos dos sites

Nesta pasta vamos criar nosso site e nossas conexões.

Agora, dentro da pasta vamos criar uma nova pasta com nome Mini Curso Avançado

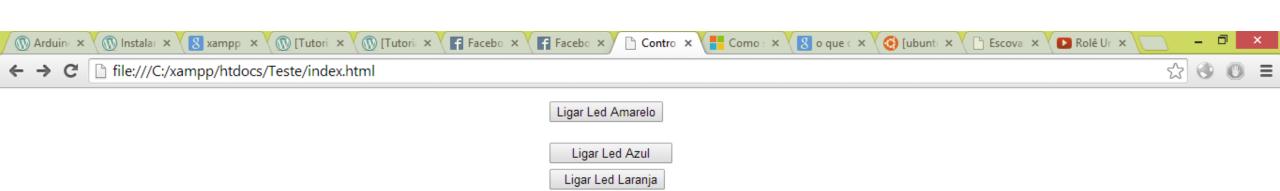
Dentro dela, vamos criar dois arquivos de texto no bloco de notas. Um com o nome de : index.html e outro com o nome de arduino.php





```
<u>A</u>rquivo <u>E</u>ditar <u>F</u>ormatar <u>Exibir</u> Aj<u>u</u>da
k?php
$port = fopen("/dev/ttyUSB0", "w");
if ($_POST['estado']=="Amarelo")
    echo "Ligou o Amarelo";
    fwrite($port, "a");
    if ($_POST['estado']=="Azul")
        echo "Ligou o Azul";
        fwrite($port, "z");
        if ($_POST['estado']=="Laranja")
             echo "Ligou o Laranja";
             fwrite($port, "1");
             if ($_POST['estado']=="Verde")
                 echo "Ligou o Verde";
                 fwrite($port, "v");
fclose($port);
```

?>



Ligar Led Verde

COMINICACIAO 12C

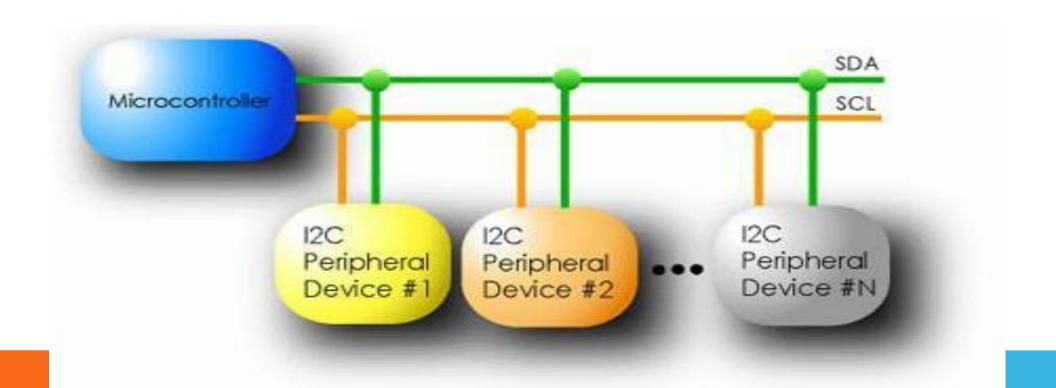
Esse novo barramento foi batizado pela empresa Holandesa (Philips) como Inter IC ou I2C e possibilita a utilização de grande quantidade de componentes padronizados, os quais podem realizar diversas funções, além de possibilitar a troca eficaz de informações entre eles. Esse novo barramento obteve uma grande aceitação do mercado, tornando-se um dos mais utilizados e isso foi determinante para a criação da versão 2.0, lançada em 1998.

O conceito do barramento I2C é facilitar a integração de circuitos de caráter final de aplicação como por exemplo sensores e conversores, com um sistema de controle, de modo que eles possam trabalhar com seus sinais de maneira direta.

O barramento I2C

Uma característica bastante interessante deste barramento é a possibilidade de utilizar, em um mesmo sistema, componentes de tecnologias construtivas diferentes sem que haja incompatibilidade e nem conflitos na comunicação.

No I2C a transmissão da informação entre os dispositivos é feita através de 2 fios (Serial Data DAS e Serial Clock SCL).



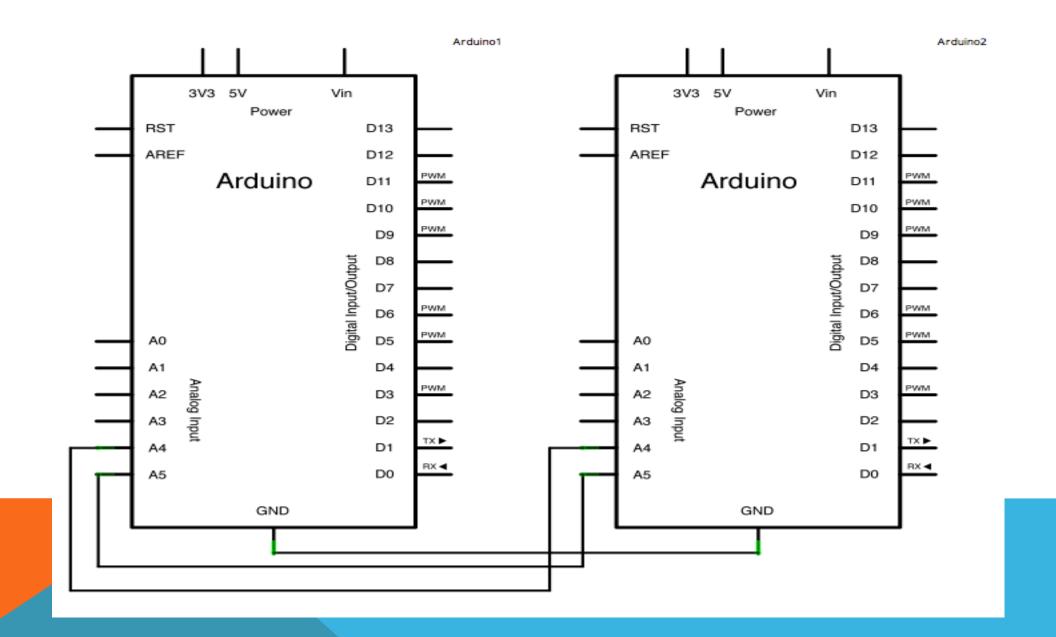
Os dispositivos ligados em Inter IC possuem um endereço fixo (cada componente recebe um endereço específico), e podemos configurá-los para receber ou transmitir dados; dessa maneira eles podem ser classificados de várias formas, como: mestres (MASTER), escravos (SLAVE), entre outras.

Dois conceitos importantes no protocolo, são os relativos ao equipamento mestre e ao escravo

Define-se como mestre, o equipamento responsável pôr toda a iniciativa do processo de comunicação, isto é, uma troca de dados entre os equipamentos, seja em qualquer dos dois sentidos, é iniciado e finalizado pelo mestre.

Define-se como escravo, o equipamento que, no processo de comunicação, só realiza alguma função, sob requisição e controle do mestre.

Um equipamento pode ser mestre ou escravo dependendo somente de sua aplicação no processo. Isto quer dizer que para um determinado processo, um equipamento pode estar configurado como mestre, sendo que, em outro processo o mesmo equipamento pode estar configurado como escravo.



SINTAXE 12C

Wire.begin(end) - inicia o processo, semelhante a Serial.begin.

<u>Wire.requestFrom</u>(end,qnt) – manda o pedido ao slave seguido da quantidade de bytes.

<u>Wire.beginTransmission</u>(end) – inicia a transmissão no endereço.

Wire.endTransmission(end) -finaliza a transmissão no endereço.

Wire.write("...") - Escreve na transmissão I2C.

<u>Wire.available()</u> – Verifica a disponibilidade, semelhante ao Serial.available.

Wire.read() - Le a transmissão;

<u>Wire.onReceive()</u> – Registra a função quando o slave recebe uma ordem do master.

Wire.onRequest() - Registra um função quando o slave recebe um pedido do master.

CÓDIGO MASTER

File Edit Sketch Tools Help



```
sketch_jun25a§
#include <Wire.h>
void setup()
 Wire begin():
 Serial.begin(9600);
void loop()
char op = Serial.read();
if(op == 'a')
 Wire beginTransmission(2):
 Wire.write('h');
 Wire.endTransmission();
 Wire.requestFrom(2, 6);
 while(Wire.available())
    char c = Wire.read();
   Serial print(c):
if(op == 'b')
<
```

Done uploading.

Binary sketch size: 4.098 bytes (of a 32.256 byte maximum)



sketch_jun25a §

```
Wire beginTransmission(2):
 Wire.write('h');
 Wire.endTransmission();
 Wire.requestFrom(2, 6);
 while(Wire.available())
   char c = Wire.read();
   Serial.print(c);
}
if(op == 'b')
 Wire beginTransmission(2):
 Wire.write('f');
 Wire.endTransmission();
 Wire.requestFrom(2, 6);
 while(Wire.available())
   char d = Wire.read();
   Serial.print(d);
}
 delay(500);
```

Done uploading.

Binary sketch size: 4.098 bytes (of a 32.256 byte maximum)

CÓDIGO SLAVE

File Edit Sketch Tools Help

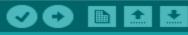
sketch_jun25b §

```
#include <Wire.h>
char n[6];
int estado =0;
void setup()
  pinMode(13,0UTPUT);
  digitalWrite(13,LOW);
  Wire.begin(2);
  Wire.onReceive(requestEvent);
 Wire.onRequest(requestEvent2);
  for(int i = 0;i<5;i++)</pre>
  n[i]=i;
void loop()
  delay(100);
void requestEvent(int howmany)
  while(Wire.available())
    char op = Wire.read();
    if(op =='h')
<
```

Done uploading.

Binary sketch size: 2.828 bytes (of a 32.256 byte maximum)

File Edit Sketch Tools Help



```
sketch_jun25b§
    char op = Wire.read();
    if(op =='h')
      digitalWrite(13,HIGH);
      estado = 1;
     if(op =='f')
      digitalWrite(13,LOW);
      estado = 0;
void requestEvent2()
  if(estado == 1)
 Wire.write("LedOn ");
 else
Wire.write("LedOFF");
}
<
```

Done uploading.

Binary sketch size: 2.828 bytes (of a 32.256 byte maximum)

BLUETOOTH

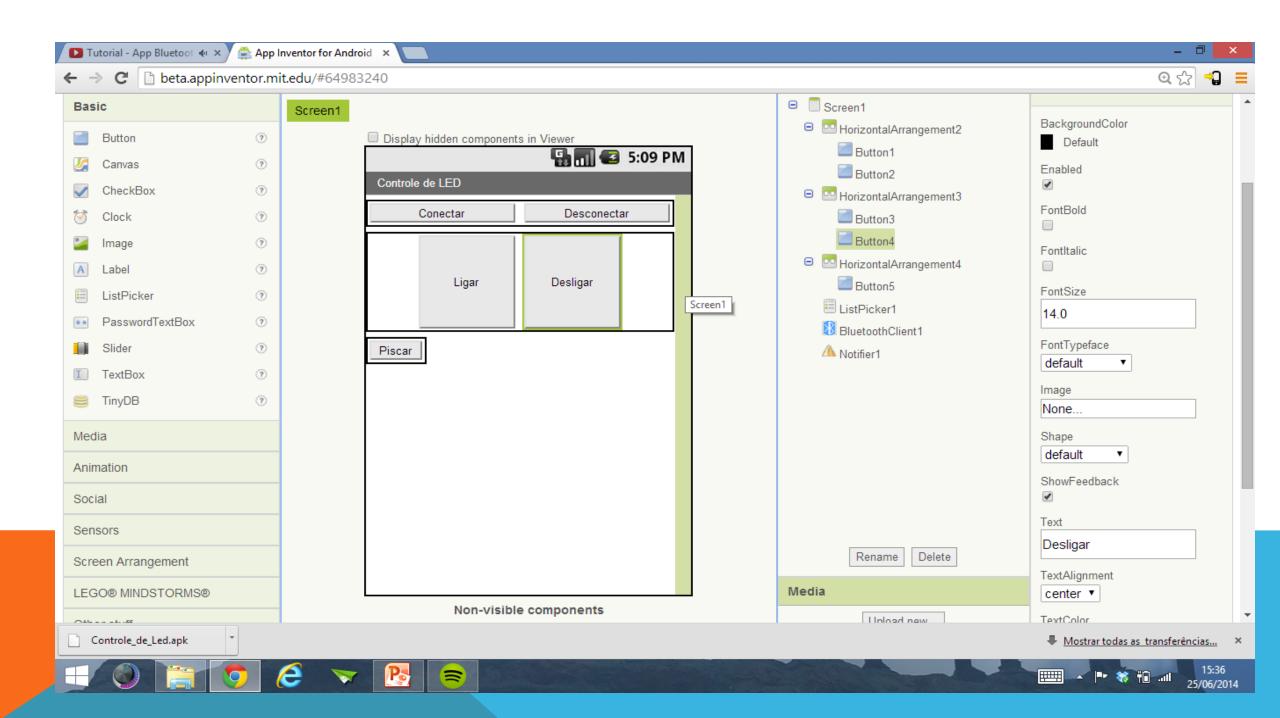
Funciona como comunicação Serial

Rx e Tx

DESENVOLVENDO UM APLICATIVO

MIT Appinventor

Criar Botões e itens



New emulator

