

# ARDUINO & PYTHON COMBINAÇÃO INFALÍVEL!



Tiago Ribeiro Santos

[www.tiagoprogramador.carbonmade.com](http://www.tiagoprogramador.carbonmade.com)

# >> Tiago = who this guy??



- **Programador** Python,Java,C,PHP, Arduino.Desenvolvo projetos com hardware livre para empresas e clientes físicos.
- **Fundador** do Grupo de Arduino no Espírito Santo – [www.arduinoes.com.br](http://www.arduinoes.com.br)
- **Pesquisador** em soluções com Hardware Livre/projetista(IFES/ Eu mesmo rsrs)
- **Monitor de lógica/programação** no IFES – Instituto Federal do ES – Campus Santa Teresa (Acadêmico do Curso Análise de Sistemas)
- **Instrutor de robótica educacional** e atuo como desenvolvedor com interesses em visão computacional,microcontroladores e eletrônica.



# >> Python Language's

- Criada em 1991 por Guido Van Rossum.
- Linguagem simples ,poderosa e bem clara!
- “Batteries included”, ou seja, baterias inclusas..Existe praticamente todas as bibliotecas prontas para programar em Python! Tudo pronto pra você! É somente um “import”
- Uso de indentação para marcar blocos, o que organiza em muito o seu código, adotando “boas práticas” de programação.
- Dá para se fazer quase tudo em Python!! Não tem limitações com a linguagem !

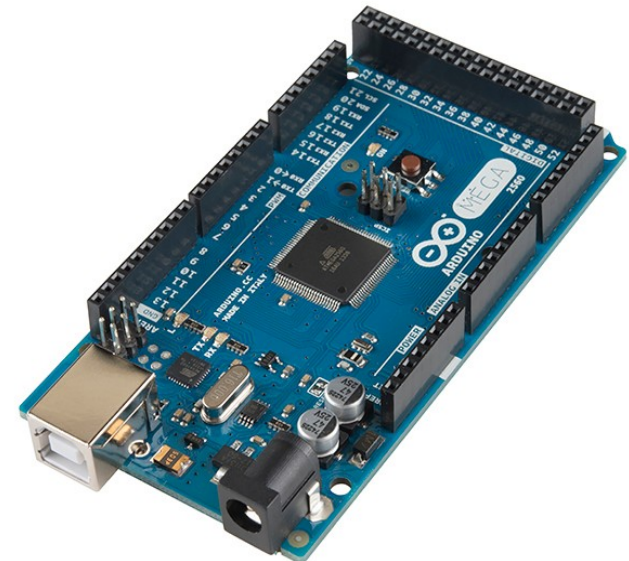
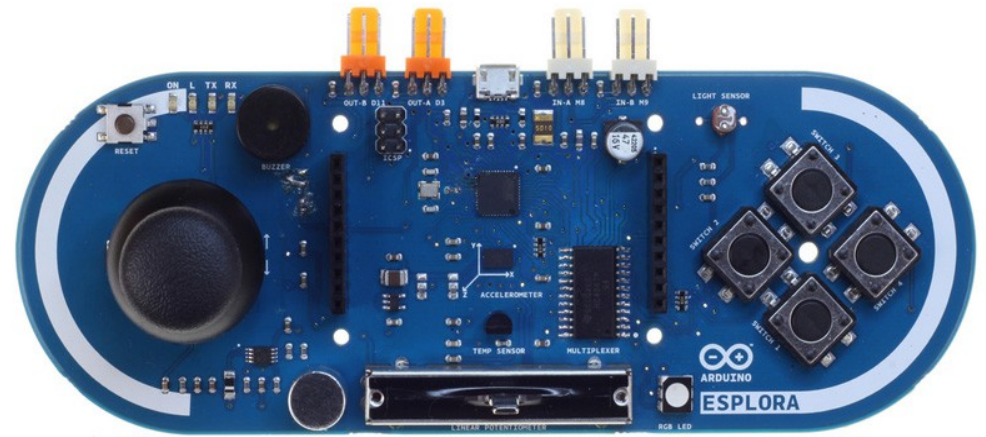
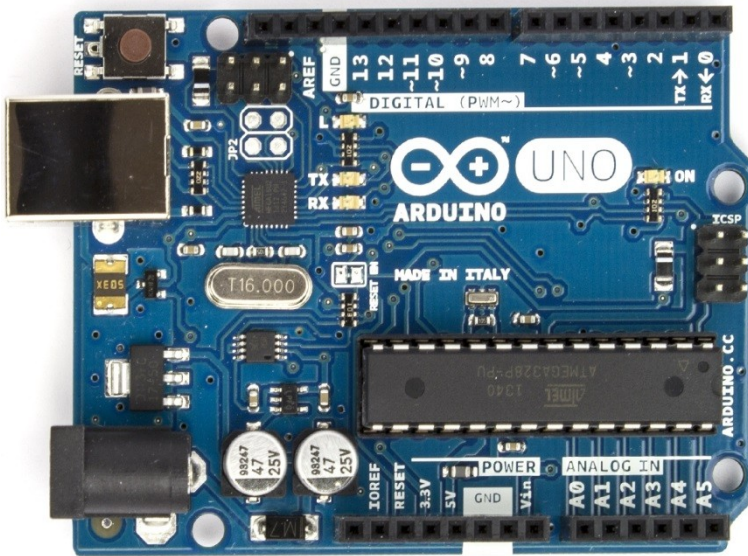


# >> Arduino

- Criado em 2005 por Massimo Banzi e Casey Reas.
- Microcontrolador de baixo custo (mais barato do que ir comer pizza rsrssr xD)
- Podemos fazer diversos projetos com Arduino desde um código para acender LEDs até um sistema avançado de automatização residencial .
- Comunidade Open-Source Hardware Livre forte!
- Linguagem simples de programação baseada em C (Wiring)

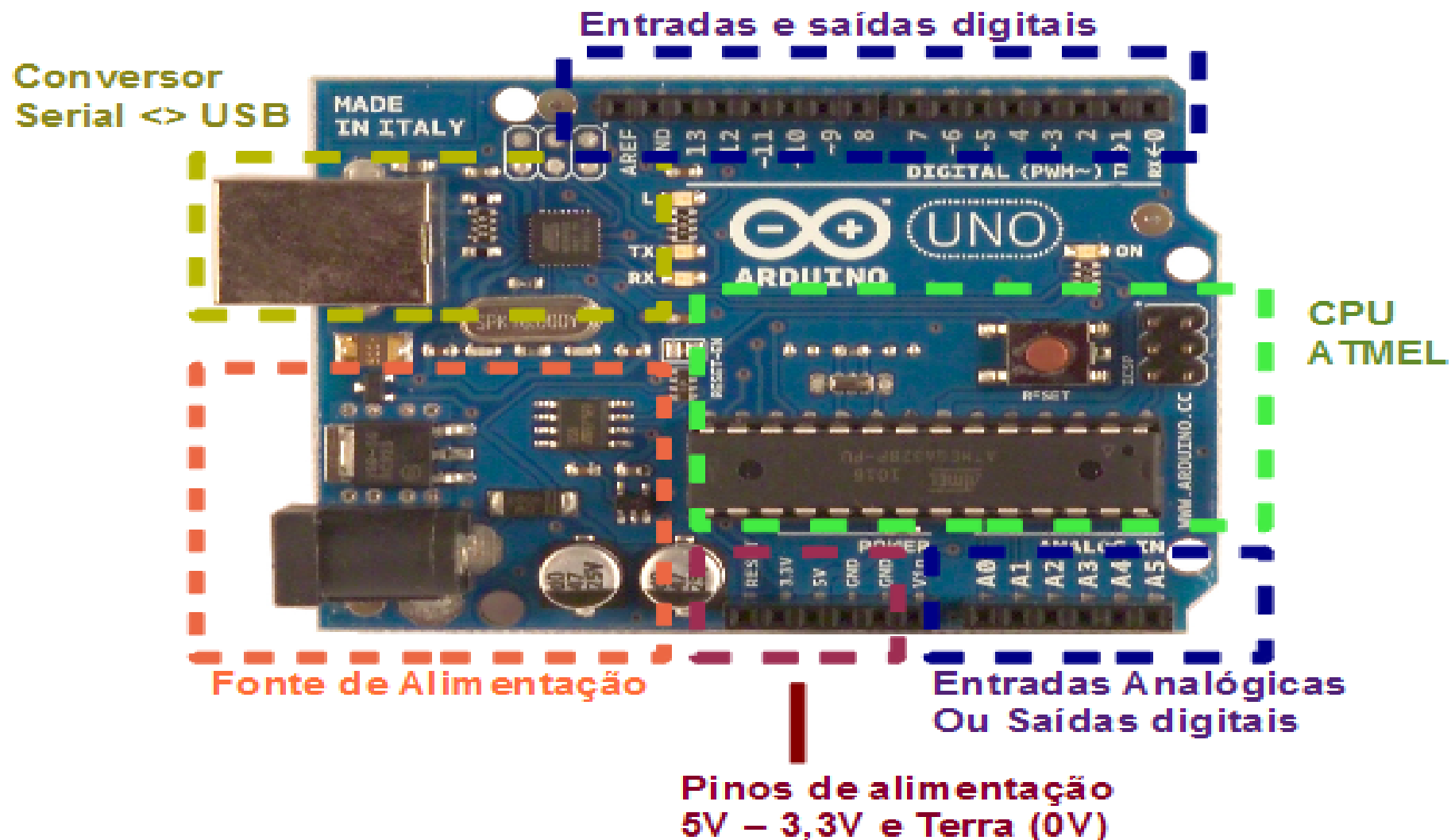


# >> Models of Arduino Hardware



# >> Hardware Arduino

## Blocos identificados de uma placa Arduino Uno



# >> Arduino + Python



- Fácil e intuitiva
- Código Aberto
- Produtividade Alta



## >> PySerial

- Biblioteca criada em 2006 que permite a comunicação do Python com dispositivos eletrônicos via porta Serial.
- A biblioteca é necessária para a comunicação do Python com o Arduino.
- How do install a PySerial on your System(LINUX) :

**\$ sudo pip install pyserial**

**OU**

**\$ sudo apt-get install python**



# >> Test a connection PortSerial

- Utilizaremos nosso Arduino UNO para criar uma conexão com Python.
- Digite no terminal Linux para Listar as portas Serial :  
**# dmesg | grep tty**

Se aparecer : **ttyACM0 : USB ACM device (Porta Arduino)**

- Agora abra o interpretador Python no Terminal e rode os seguintes comandos :

```
>> import serial
```

```
>> conexao = serial.Serial("/dev/ttyACM0",9600)
```

```
>> conexao.isOpen() #Retorna TRUE ou FALSE
```

# >> Test\_SerialPort.py (Part I)

```
# -*-coding:utf-8 -* # Codificação UTF-8
```

```
import serial # importar a biblioteca Serial
```

```
porta = "/dev/ttyACM0" # Endereço PortaSerial
```

```
conexao = serial.Serial(porta,9600) #Abre PortaSerial
```

```
verifica_porta = conexao.isOpen() #Retorna TRUE  
OR FALSE para ConexaoSerial
```

## >> Test\_SerialPort.py (Part II)

```
def verifica_porta():
```

```
if verifica_porta == True:  
    print("Porta esta aberta!")
```

```
elif verifica_porta == False :  
    print("PortaSerial está fechada!")
```

```
else:  
    print("PortaSerial está desconectada!")
```

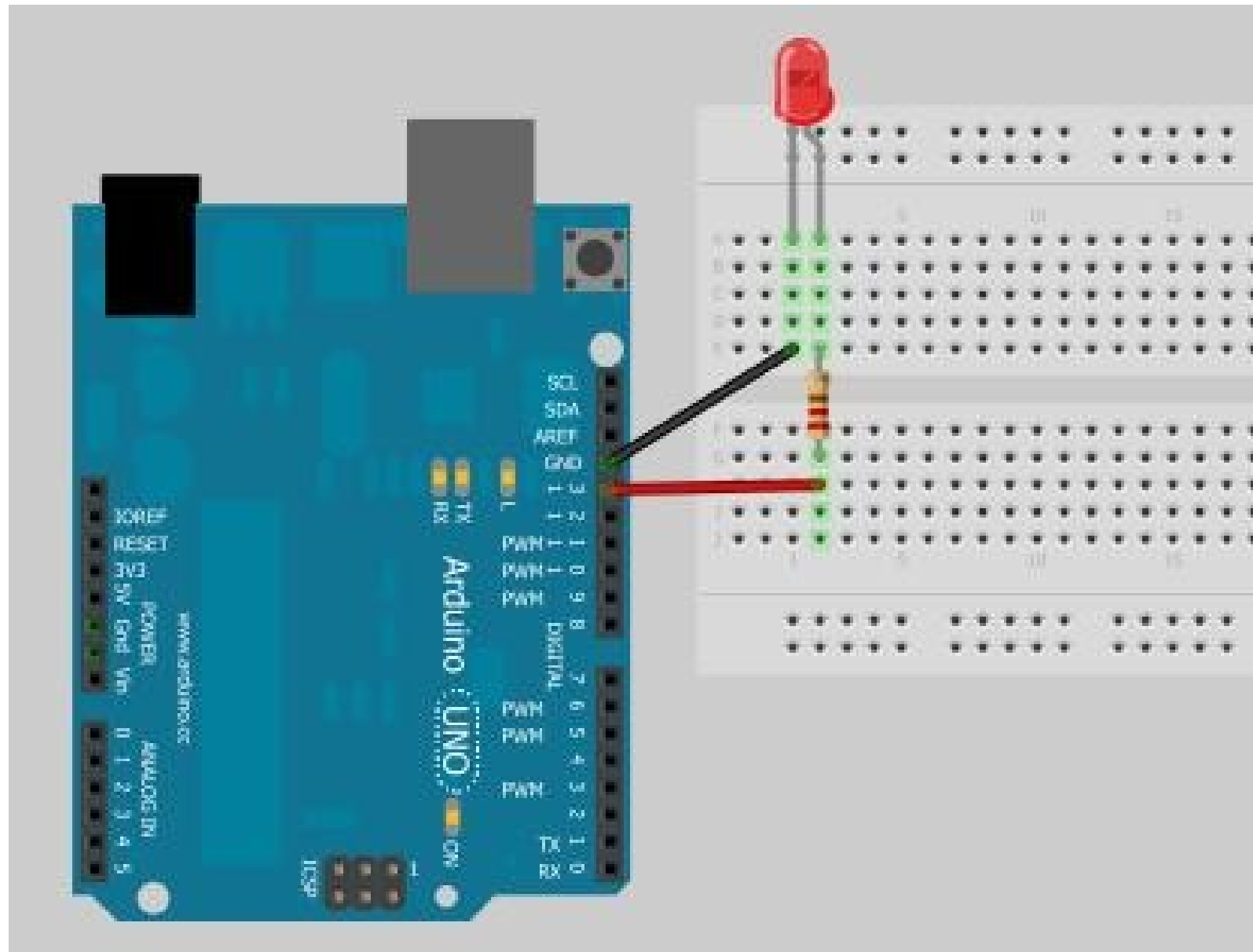
```
verifica_porta()    #Chama a função Verificadora
```

# >> Arduino IDE Bridge for Python

- Para que a biblioteca **PySerial** reconheça as portas digitais ou analógicas de nosso Arduino que serão utilizadas para se comunicar com o Python, devemos escrever um sketch (programa) para nosso Hardware Livre usando o **Arduino IDE**.

```
Arquivo Editar Sketch Ferramentas Ajuda
sketch_oct05b
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

# >> Build a LED Circuit



# >> Sketch Arduino

```
#define portaLed 13

void setup() {
    Serial.begin(9600);
    pinMode(portaLed,OUTPUT);
}

void loop() {
    char valor_recebido ='0';
    while(Serial.available() > 0 ){
```



# >> Sketch Arduino

```
valor_recebido = Serial.read();  
if(valor_recebido == '1'){  
    digitalWrite(portaLed,HIGH);  
}  
else if(valor_recebido == '2'){  
    digitalWrite(portaLed,LOW);  
}  
}  
}
```

# >> Python IDE for a pyCode!

- Escolher uma boa IDE Python pode ser bastante útil e interessante para facilitar uma indentação da linguagem e por usarmos menos tempo e esforço no desenvolvimento.
- A **Python Brasil** indica uma lista dos melhores IDE Python para trabalhar. Visite o link abaixo e instale a ferramenta de acordo com sua preferência.

<https://wiki.python.org.br/IdesPython>

# >> PythonCode for Arduino

- Logo após a instalação de nossa IDE Python , vamos agora trabalhar em nosso código para controlar nosso Led que já está ligado ao Arduino.
- Sempre iremos utilizar o sistema de codificação **UTF-8** em nossos códigos Python para usarmos caracteres de nosso idioma nativo (Português) no programa. Na primeira linha do nosso programa colocaremos a instrução:

```
>> # -*-coding:utf-8 -*
```

# >> acende\_led.py (Part I)

```
# -*-coding:utf-8 -*-
```

```
import serial
```

```
conexao = serial.Serial("/dev/ttyACM0",9600)
```

```
logico = True
```

```
def ativaLed(logico):
```

```
    while logico:
```

```
        print("")
```

```
        print("Digite 1 para Ligar LED , 2 para  
Desligar LED ou 0 para sair:")
```

```
        opcao = (str)(raw_input("Digite sua opção  
escolhida : "))
```

## >> acende\_led.py (Part II)

```
if opcao == '1':  
    conexao.write('1'.encode())  
    print("LED1 LIGADO!")  
if opcao == '2':  
    conexao.write('2'.encode())  
    print("LED DESLIGADO!")  
if opcao == '0':  
    print("Bye!!!Volte sempre!")  
    exit()  
if (not(opcao == '1' or opcao == '2' or opcao == '0')):  
    print("Opção invalida!Tente novamente!")  
  
ativaLed(logico)
```

# >> Tkinter – A Python Interface GUI

- O **Tkinter** é uma biblioteca Python que acompanha a instalação padrão e permite desenvolver interfaces gráficas.
- A maioria das instalações Linux já vem com a biblioteca instalada, com exceção de pouquíssimas distribuições, que necessita de instalação de módulo.
- Para testarmos nossa biblioteca Tkinter, digitamos os seguintes comandos abaixo :

```
>> import Tkinter  
>> Tkinter._test()
```



# >> Janela\_GraficaArduino.py(Part I)

```
import serial
```

```
import time
```

```
from Tkinter import *
```

```
conexao = serial.Serial('/dev/ttyACM0',9600)
```

```
def ligar_led():
```

```
    conexao.write('1'.encode())
```

## >> Janela\_GraficaArduino.py(Part II)

```
def desligar_led():
```

```
    conexao.write('2'.encode())
```

```
janela = Tk()
```

```
janela.title("BLINK LED (Liga ou Desliga)")
```

```
janela.geometry("400x130")
```

```
botaoLigar = Button(janela,text="Ligar  
LED",command=ligar_led)
```

```
>> Janela_GraficaArduino.py(Part III)
```

```
botaoDesliga = Button(janela,text="Desligar  
LED",command=desligar_led)
```

```
botaoLigar.pack()
```

```
botaoDesliga.pack()
```

```
janela.mainloop()
```

# >> References, Tutorials..

- **Tkinter(Interface Gráfica Python)**

[devmedia.com.br/tkinter-interfaces-graficas-em-python/33956](http://devmedia.com.br/tkinter-interfaces-graficas-em-python/33956)

<http://www.tkdocks.com/tutorial/install.html>

- **PySerial (Comunicação Python/Arduino)**

<https://pythonhosted.org/pyserial/>

- **Apostila/E-Book Python and Arduino:**

[https://github.com/ismartSa/eBook-1/blob/master/Python%20Programming%20for%20Arduino%20\(2015\).pdf](https://github.com/ismartSa/eBook-1/blob/master/Python%20Programming%20for%20Arduino%20(2015).pdf)



# Empresa Especializada em Arduino/Raspberry/Robótica

- **Vida de Silício (Parceria) com Descontos bem bacanas!**

**<http://www.vidadesilicio.com.br/user/arduinoes>**

- **Cupom Promocional para as Compras com ótimos descontos!!**

Digitem o código Promocional : **“ARDUINO-123”**

Boas compras! Recomendo esta empresa para todos os fãs de  
Arduino/Robótica/Raspberry.

# >> Thanks!!

- My Facebook :  
<https://www.facebook.com/tiago.arduinoRobotica>
- My GitHub:  
<https://github.com/tiglinux>
- My Website:  
<https://tiagoprogramador.carbonmade.com/>

Email :

[tiago.programador@hotmail.com](mailto:tiago.programador@hotmail.com)  
[gnu\\_tigolinux@openmailbox.org](mailto:gnu_tigolinux@openmailbox.org)