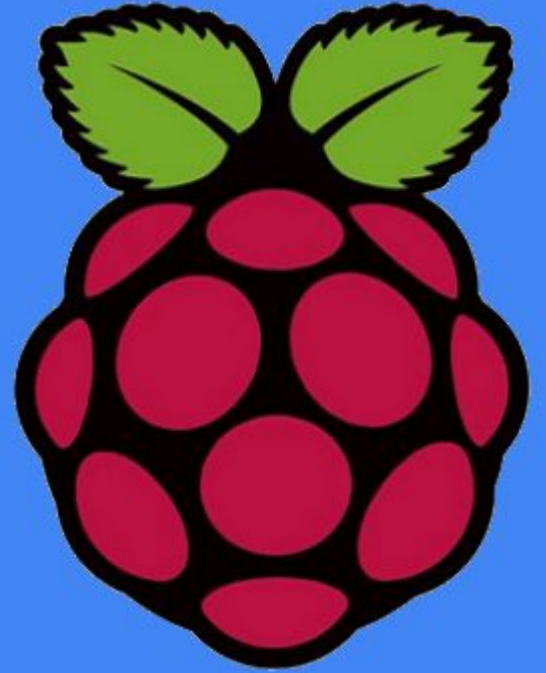


Raspberry Pi



Oque é Raspberry Pi ?!



Raspberry Pi é um computador do tamanho de um cartão de crédito, que se conecta a um monitor de computador ou TV, e usa um teclado e um mouse padrão, desenvolvido no Reino Unido pela Fundação **Raspberry Pi**. Todo o hardware é integrado numa única placa!



Funções

1. Áudio (HDMI \ P2 \ HEADER)
2. Vídeo (HDMI \ HEADER)
3. Câmera
4. USB
5. Ligar \ Desligar
6. GPIO
7. WIFI
8. Bluetooth

Hardware:

Raspberry Pi 3 Model B

Dimensions
85.6mm x 56mm x 21mm

4 x USB 2 Ports

**40 Pin
Extended GPIO**

**Broadcom
BCM2837 64bit
Quad Core CPU
at 1.2GHz,
1GB RAM**

**10/100
LAN Port**

**On Board
Bluetooth 4.1
Wi-Fi**

**3.5mm 4-pole
Composite Video
and Audio
Output Jack**

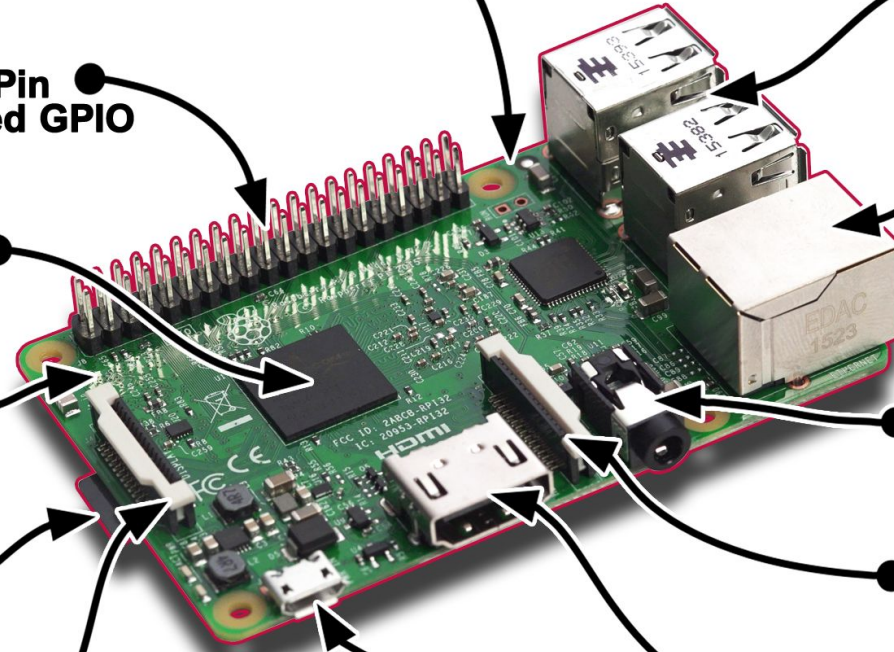
**MicroSD
Card Slot**

CSI Camera Port

DSI Display Port

**Micro USB Power Input.
Upgraded switched
power source that can
handle up to 2.5 Amps**

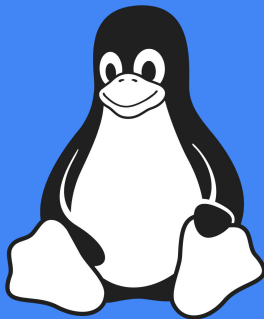
**Full Size HDMI
Video Output**



Software: Linux <3



Oque é linux ?



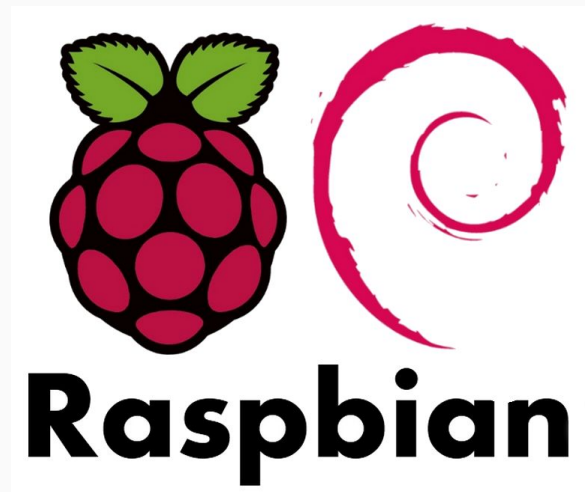
- Linux é um kernel criado por Linus Torvalds em 1991.
- Linus copiou a arquitetura de sistemas UNIX ao escrever o Linux, pois como era estudante não tinha dinheiro para comprar um sistema operacional UNIX.
- O código foi disponibilizado na internet com a esperança de que alguém se interessasse em testar e dar opiniões sobre.
- Hoje o Linux continua sendo totalmente open-source, e é considerado uma das principais tecnologias modernas.



O que são distros linux ?



- Distribuição linux = Kernel linux + um monte de software = SO
- Distribuições linux para PC são normalmente conhecidas como GNU/Linux
- A distribuição oficialmente utilizada no Raspberry Pi é chamada raspbian, uma versão modificada do Debian.
- Atualmente o sistema operacional mais utilizado do mundo é uma distro Linux: Android !!!



Como usar

Ligado em um monitor ou TV !

Vantagens vs Desvantagens

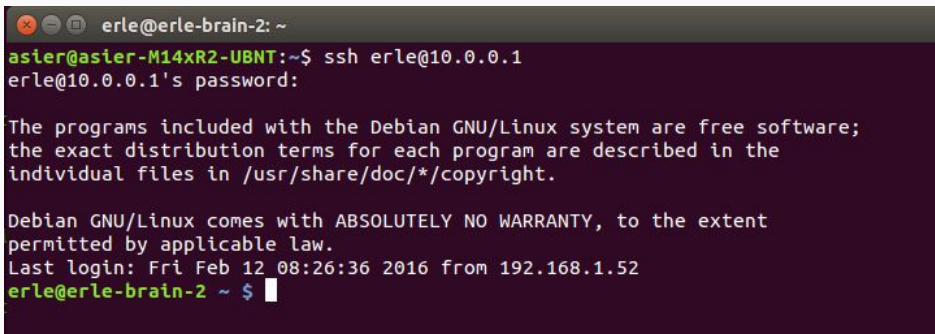
- Intuitivo
- Fácil
- Todas funções disponíveis
- Necessidade de periféricos
- Bagunça !
- Pouco prático



Remoto: SSH

Vantagens vs Desvantagens

- Confiável
- Rápido
- Simples
- Seguro
- Transfere arquivos
- “Difícil” de configurar
- Uso de Terminal
- Interface gráfica ruim
- Necessidade de rede



```
erle@erle-brain-2: ~  
asier@asier-M14xR2-UBNT:~$ ssh erle@10.0.0.1  
erle@10.0.0.1's password:  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Feb 12 08:26:36 2016 from 192.168.1.52  
erle@erle-brain-2 ~ $
```

Remoto: VNC

Vantagens vs Desvantagens

- Simples de configurar
- Acesso remoto
- Possui interface Gráfica
- Pesado
- Necessidade de rede
- Pouco prático



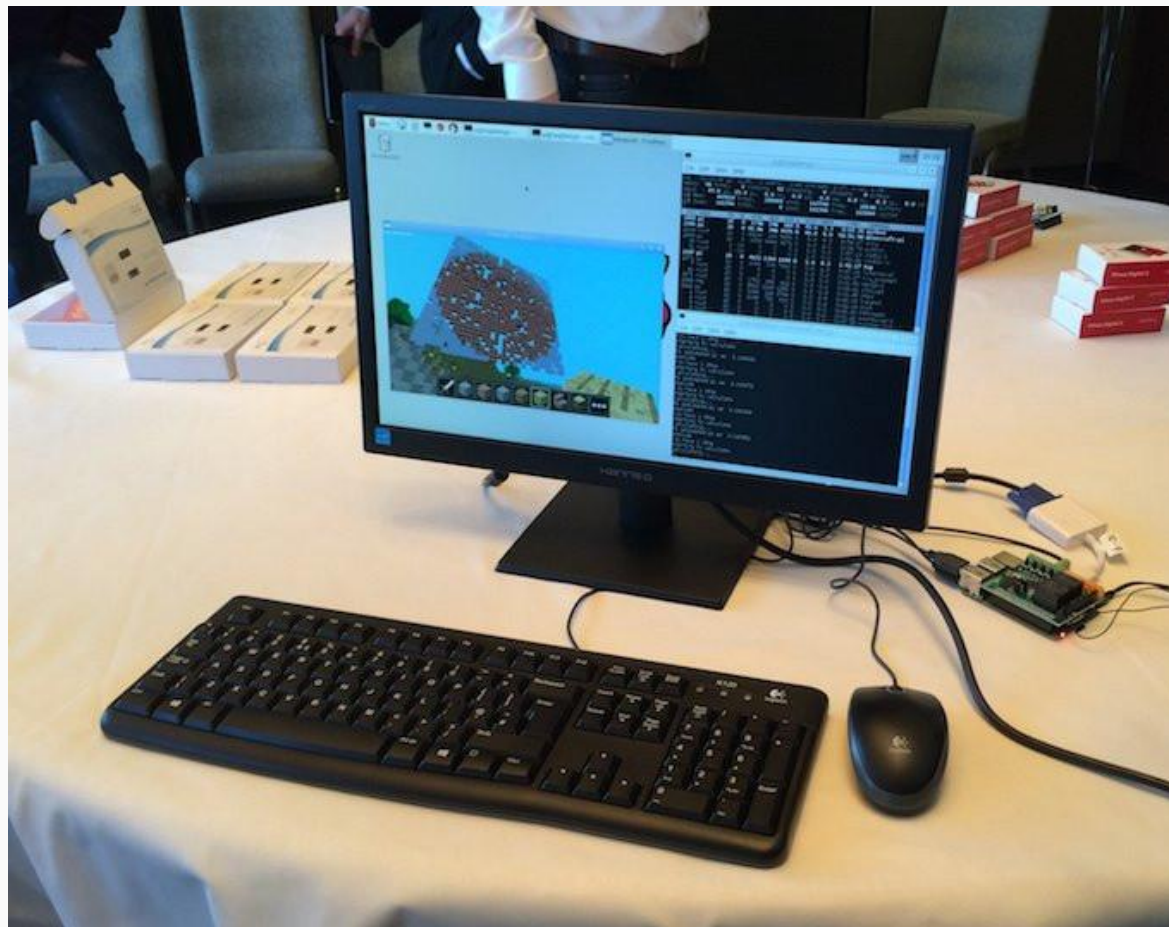
Como NÃO usar

Precauções de uso

- Sempre que **possível**, manusear pelas bordas.
- Cuidado com curtos !
- Não ligar diretamente coisas que 'puxam' muita energia

Onde usar

Desktop com Raspberry Pi



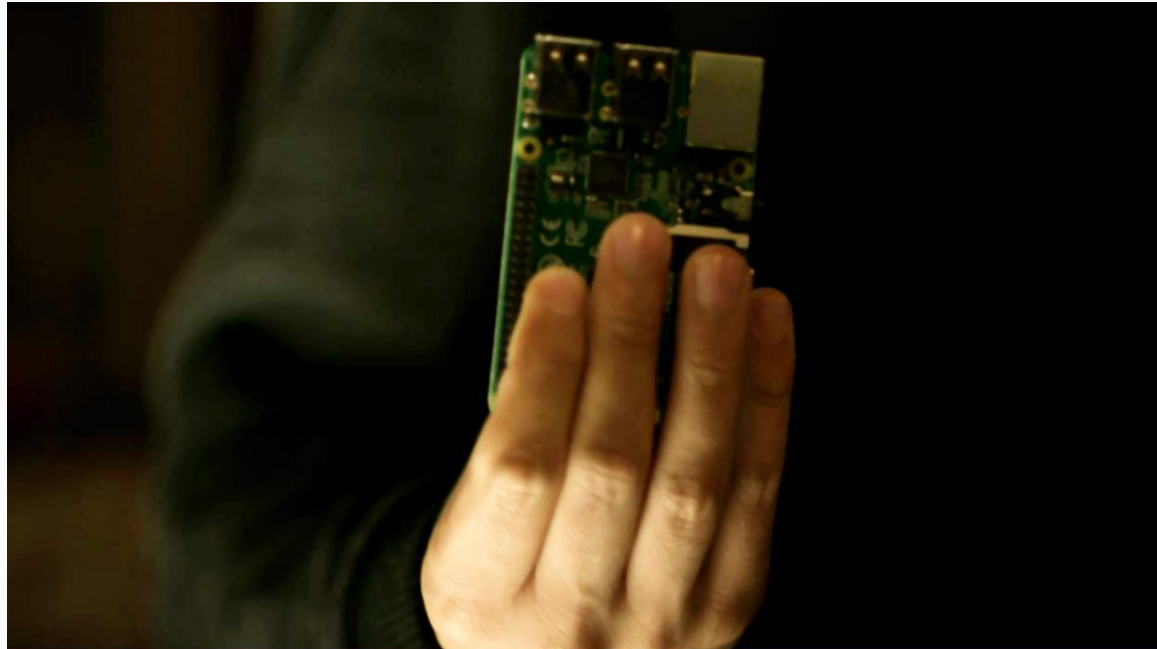
Teclado mecânico com Raspberry Pi



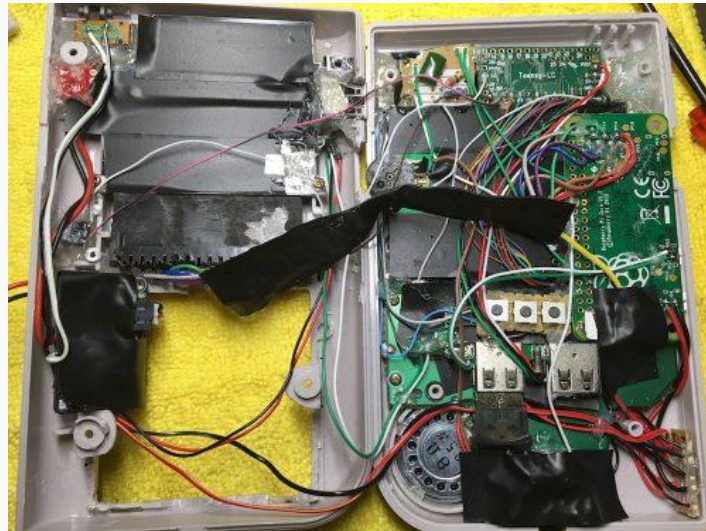
Raquiando com o
Linux/Raspberry Pi



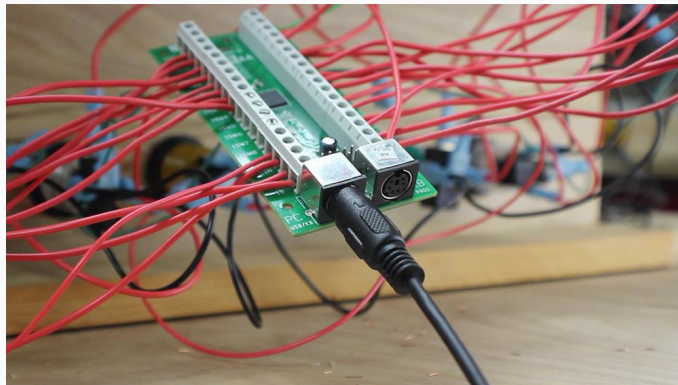
MR. ROBOT



Game Boy Zero



Arcade com Raspberry Pi



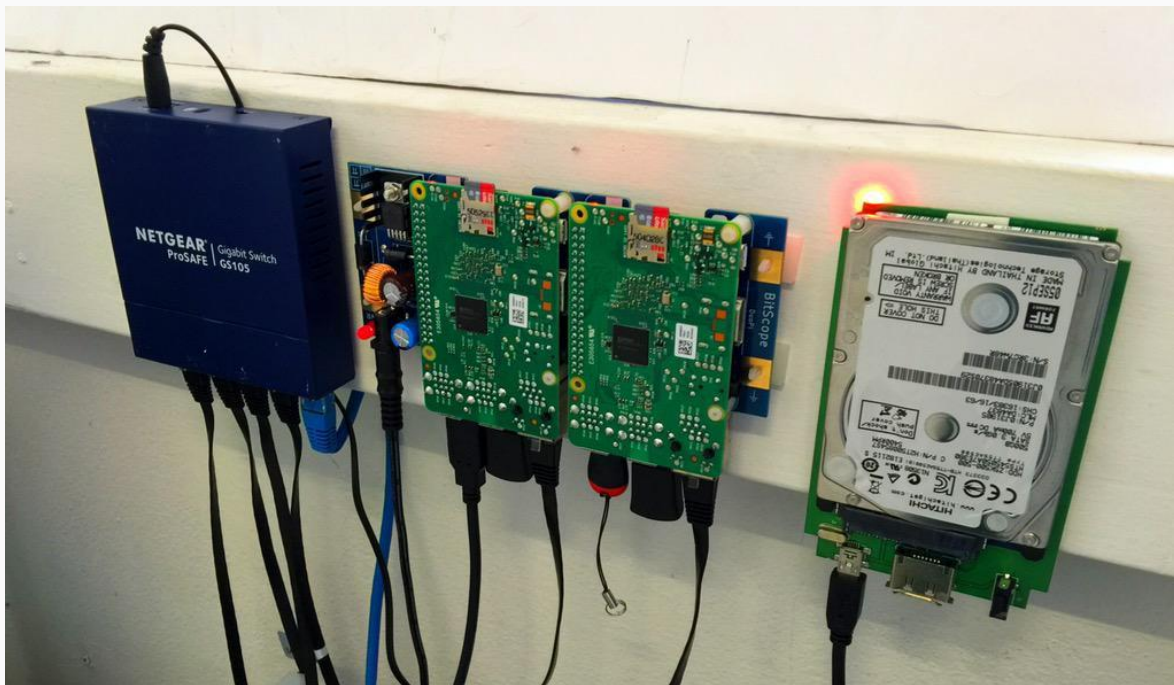
Central de Mídia com Raspberry Pi



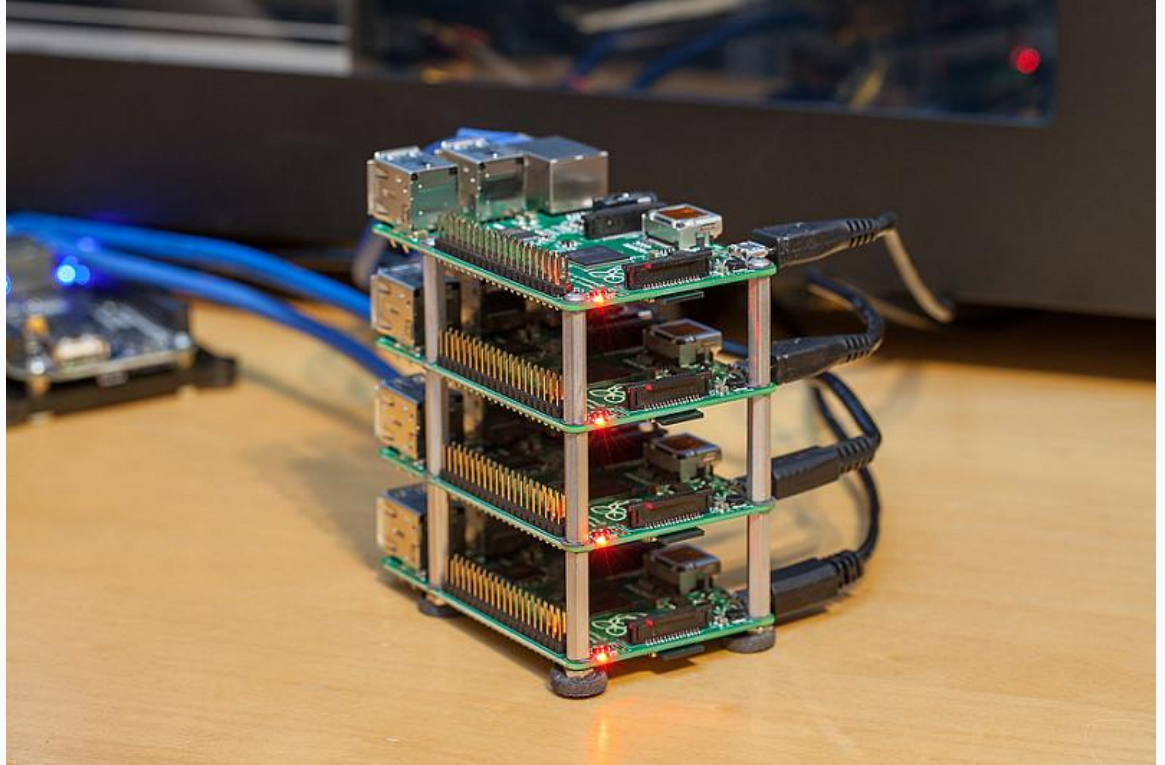
PiBook



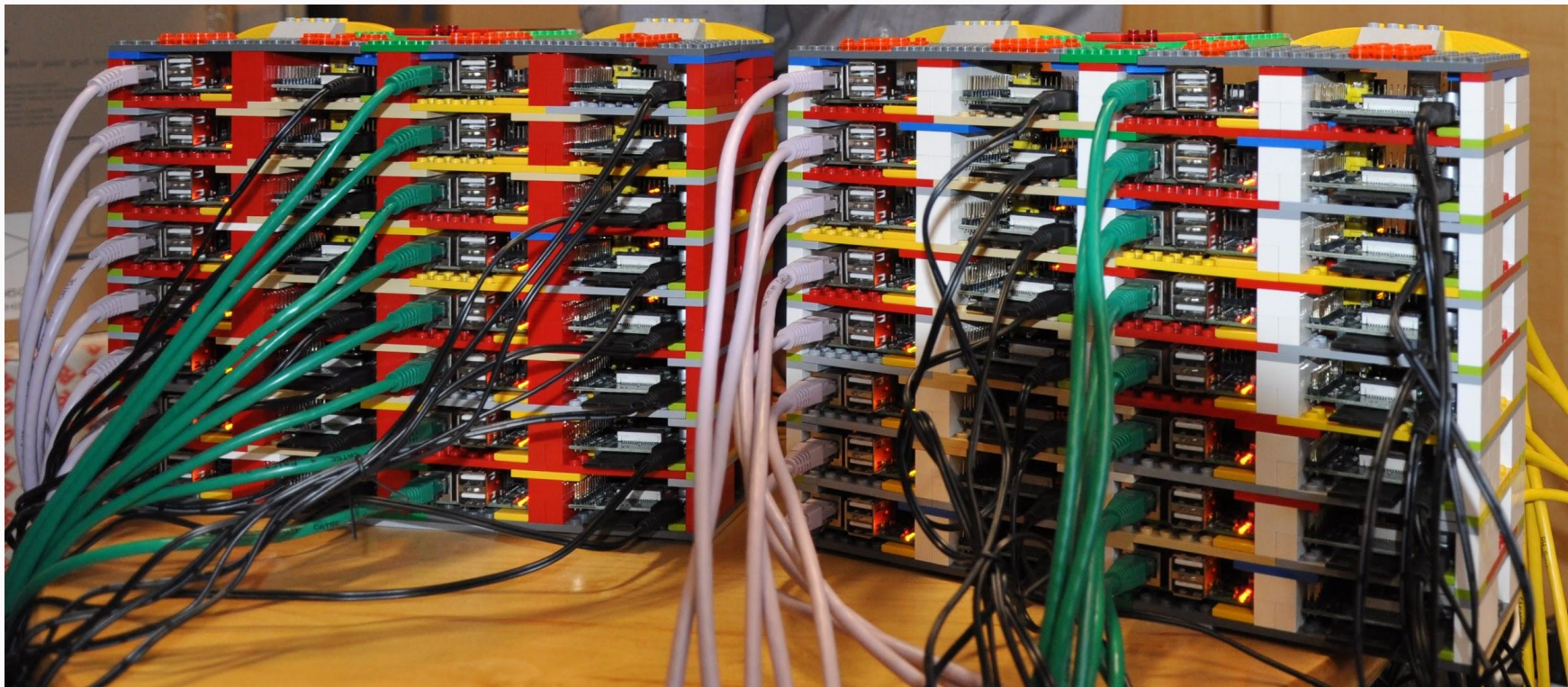
Servidores



Clusters



Cluster lvl 2



Programando o raspberry:

€

~~Java~~

~~Ruby~~

~~Assembly~~

~~PHP~~

~~Prolog~~

~~Portugal~~

~~Haskell~~

Python !!!

C++(arduino) vs Python

```
1 int led = 12;
2
3 void setup(){
4     pinMode(led, OUTPUT);
5 }
6
7 void loop(){
8     digitalWrite(led, HIGH);
9     delay(1000)
10    digitalWrite(led, LOW);
11    delay(1000)
12 }
```



```
1 import RPi.GPIO as GPIO
2 from time import sleep
3
4 GPIO.setmode(GPIO.BCM)
5
6 led = 27
7
8 GPIO.setup(led, GPIO.OUT)
9
10 while True:
11     GPIO.output(led, True)
12     sleep(1)
13     GPIO.output(led, False)
14     sleep(1)
```

Executando códigos: live with IDLE



- Escrever códigos “on the fly”
- Possibilidade de executar um programa sem um arquivo
- Demonstração clara de uma “linguagem de script”

How-to

- Abrir IDLE
- Escrever código linha a linha

Executando códigos: Scripts !

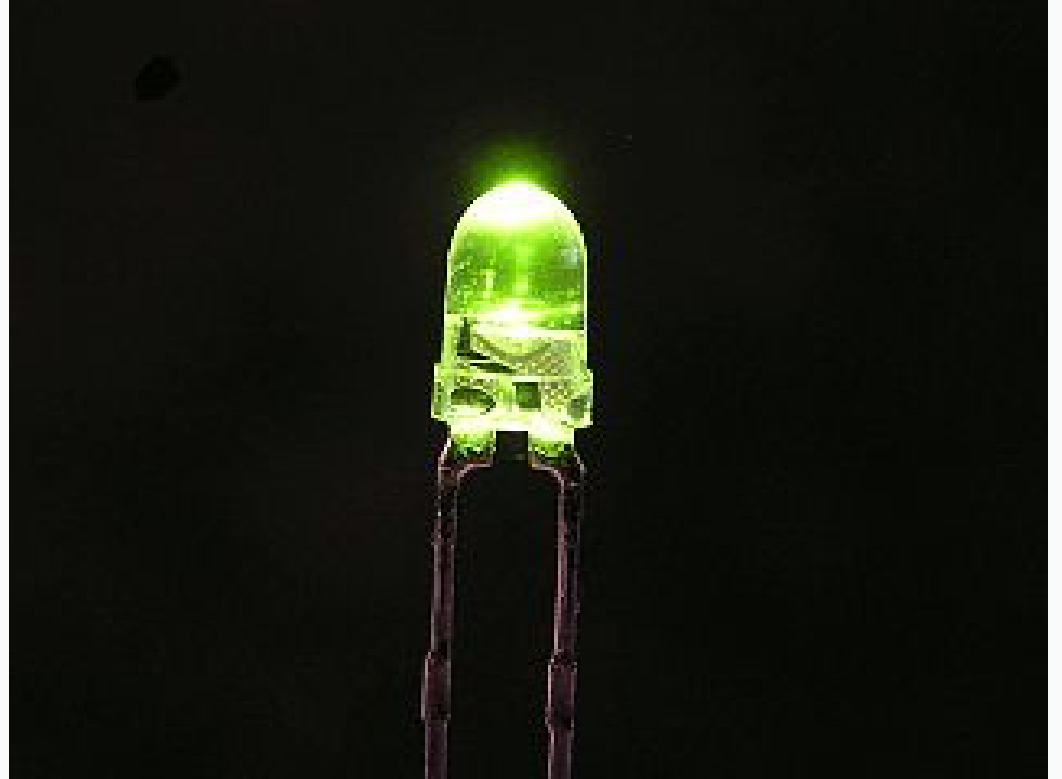


- Possível criar um arquivo .py para executar um script
- Possibilidade de estruturar, organizar e salvar o código
- Necessidade de executar com um interpretador python

How-to

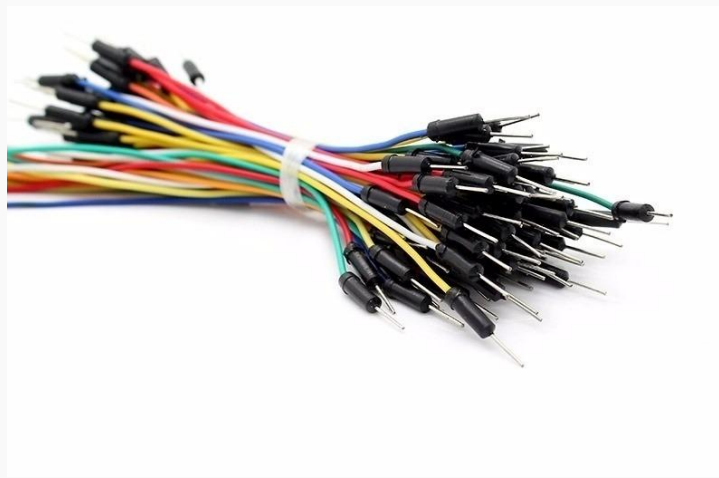
- Escrever o código em um arquivo .py
- Abrir um terminal linux
- Executar o comando: "python nomeDoPrograma.py"

Aula 1 : Blink

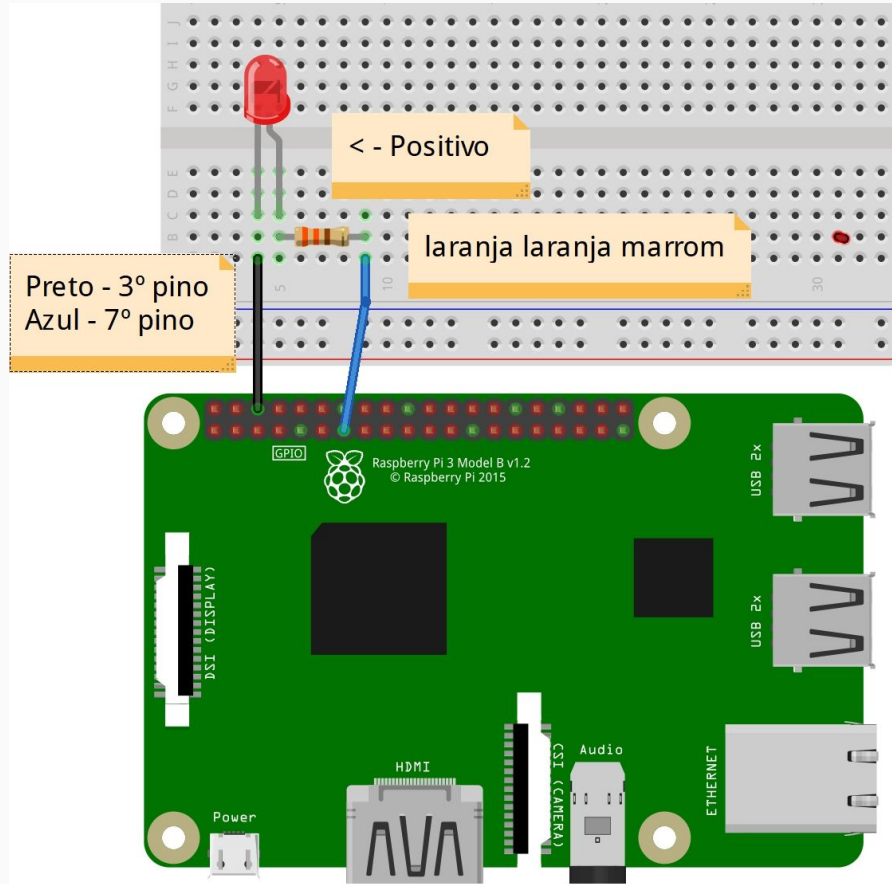


Materials :

- 1 LED
- Jumpers
- Raspberry Pi
- Resistor 330 Ω



Esquemático



| | | | |
|--------------------|----|----|--------------------|
| 3.3V | 1 | 2 | 5V |
| GPIO 2 (I2C1_SDA) | 3 | 4 | 5V |
| GPIO 3 (I2C1_SCL) | 5 | 6 | GND |
| GPIO 4 (GPCLK0) | 7 | 8 | GPIO 14 (UART_TXD) |
| GND | 9 | 10 | GPIO 15 (UART_RXD) |
| GPIO 17 | 11 | 12 | GPIO 18 |
| GPIO 27 | 13 | 14 | GND |
| GPIO 22 | 15 | 16 | GPIO 23 |
| 3.3V | 17 | 18 | GPIO 24 |
| GPIO 10 (SPI_MOSI) | 19 | 20 | GND |
| GPIO 9 (SPI_MISO) | 21 | 22 | GPIO 25 |
| GPIO 11 (SPI_SCLK) | 23 | 24 | GPIO 8 (SPI_CE0) |
| GND | 25 | 26 | GPIO 7 (SPI_CE1) |
| ID_SD | 27 | 28 | ID_SC |
| GPIO 5 | 29 | 30 | GND |
| GPIO 6 | 31 | 32 | GPIO 12 |
| GPIO 13 | 33 | 34 | GND |
| GPIO 19 | 35 | 36 | GPIO 16 |
| GPIO 26 | 37 | 38 | GPIO 20 |
| GND | 39 | 40 | GPIO 21 |

Código

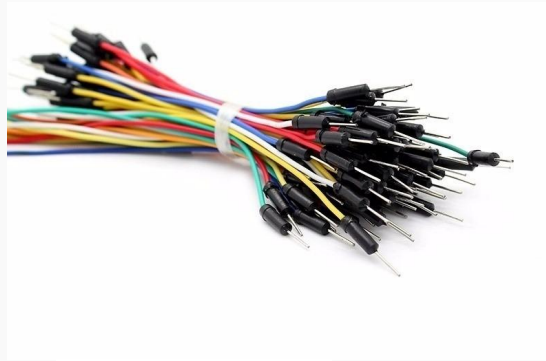
```
1 import RPi.GPIO as GPIO
2 from time import sleep
3
4 GPIO.setmode(GPIO.BCM)
5
6 led = 27
7
8 GPIO.setup(led, GPIO.OUT)
9
10 while True:
11     GPIO.output(led, True)
12     sleep(1)
13     GPIO.output(led, False)
14     sleep(1)
```

Blink com botão

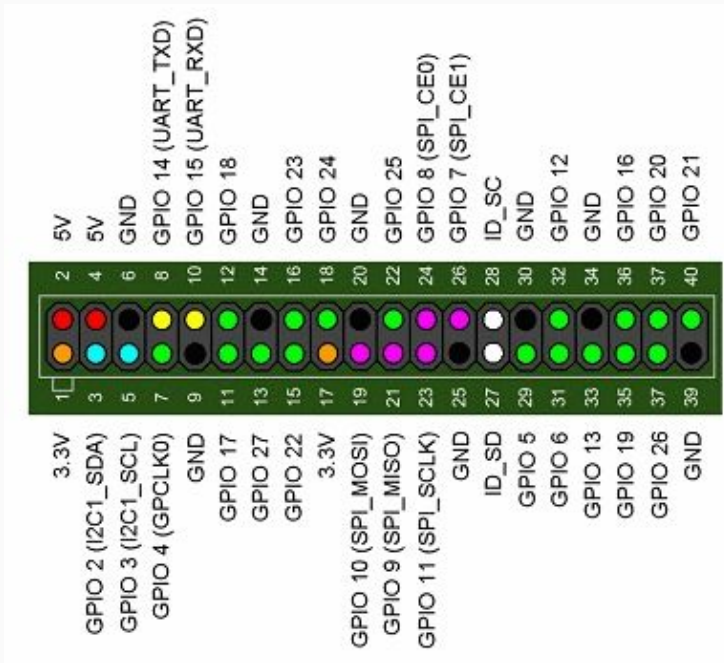
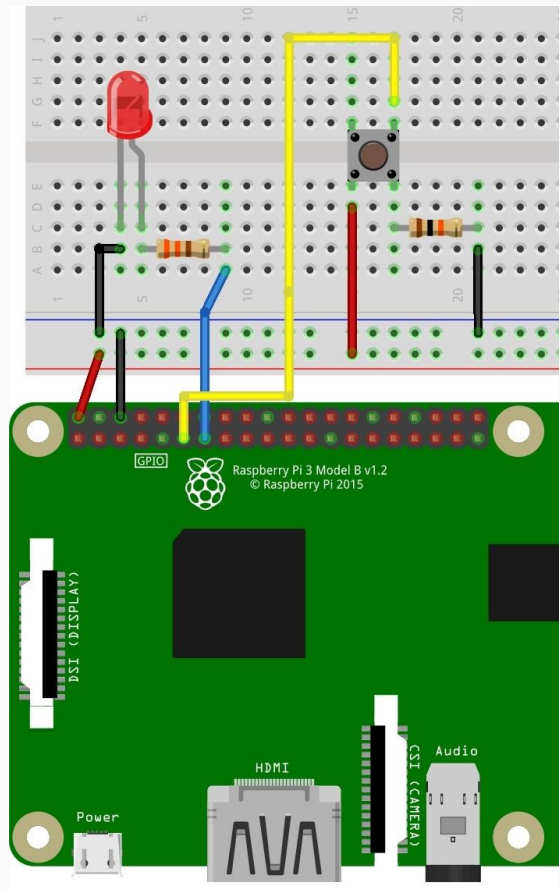


Materiais

- 1 LED
- Jumpers
- Botão
- Resistores (10k Ω e 330 Ω)



Esquemático



Código

```
1 import RPi.GPIO as GPIO
2
3 GPIO.setmode(GPIO.BCM)
4
5 led = 27
6
7 GPIO.setup(led, GPIO.OUT)
8
9 button = 17
10
11 GPIO.setup(button, GPIO.IN)
12
13 GPIO.output(led, False)
14
15 while True:
16     if GPIO.input(button) == 1:
17         GPIO.output(led, True)
18     else:
19         GPIO.output(led, False)
20
21 #Na IDLE lembrar de dar espaço !
22 #CTRL + C para sair do loop !
```