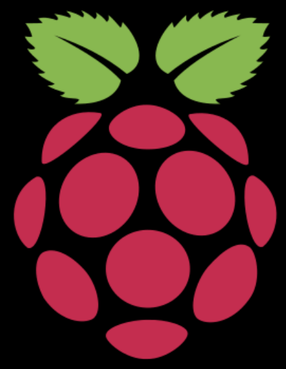


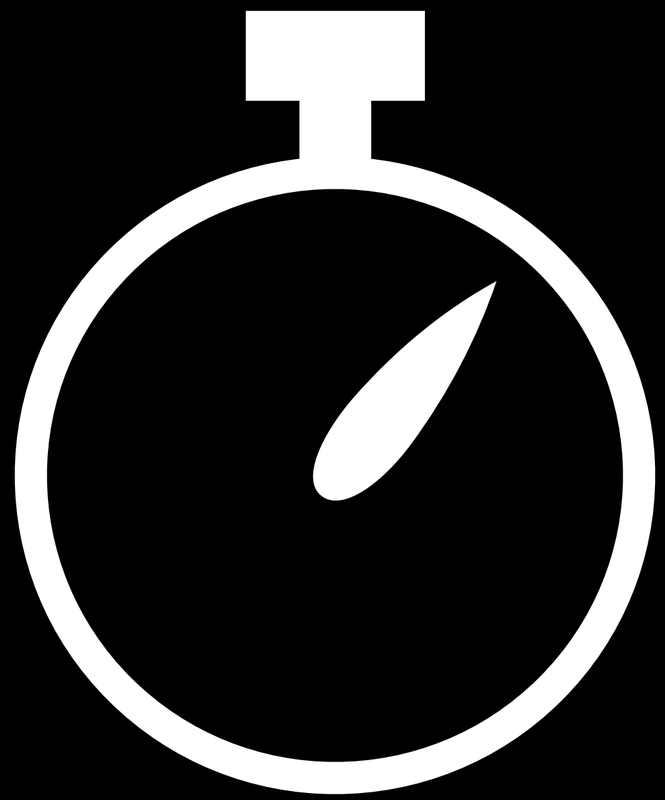
Projeto 05

Controle Automático – Teoria

Jan K. S. – janks@puc-rio.br

ENG1419 – Programação de Microcontroladores

Ferramentas do Python



Timer

Conceito de Timer

```
>>> from threading import Timer
>>> def timer_acabou():
...     print("Acaboouuoo, acabou!")
...
>>> timer = Timer(3.5, timer_acabou)
>>> timer.start()
```

3.5 segundos depois

```
Acaboouuoo, acabou!
```

Ao contrário da `sleep`,
`Timer` não trava a execução!

Timer em Python

```
>>> timer = Timer(3.5, timer_acabou)
>>> timer.start()
>>> timer.cancel()
```

Cancelamento de Timer

```
>>> timer = Timer(3.5, timer_acabou)
>>> timer.start()
>>> timer.cancel()
>>> timer.start()
Traceback (most recent call last):
  ...
RuntimeError: threads can only be started once


# precisamos criar o timer novamente
>>> timer = Timer(3.5, timer_acabou)
>>> timer.start()
```

Cancelamento de Timer

Timer Único

```
daqui a 3.5 segundos,
imprima "Acabou!"
```

3.5 segundos ----------► Acabou!

Timer Recorrente

1 segundo

```
imprima "Repetição"

daqui a 1 segundo,
repita o processo
```

```
cancele o timer!
```

Timer Único x Timer Recorrente

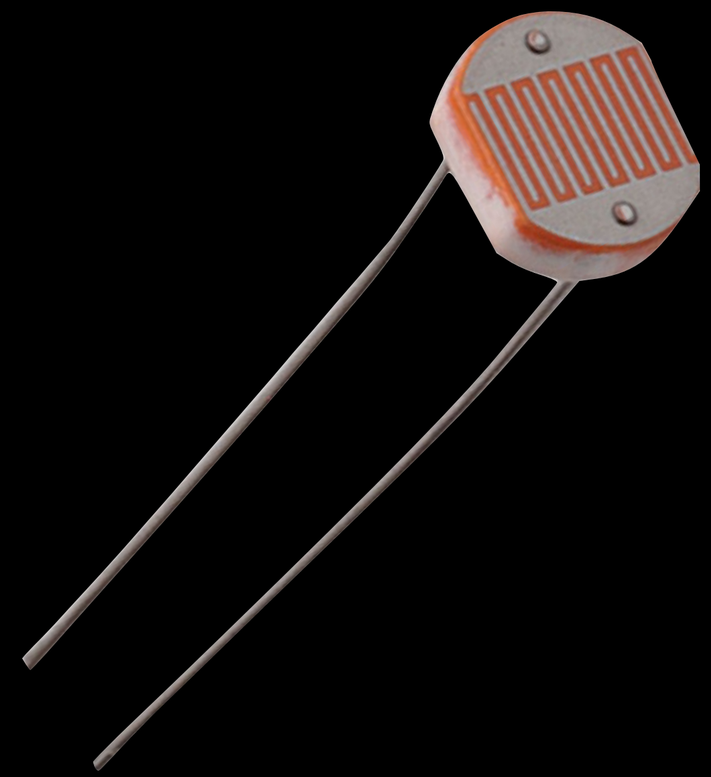
```
>>> from threading import Timer
>>> timer = None
>>> def timer_recorrente():
...     print("Repetição")
...     global timer
...     timer = Timer(1.0, timer_recorrente)
...     timer.start()
...
```

chame depois a própria função!

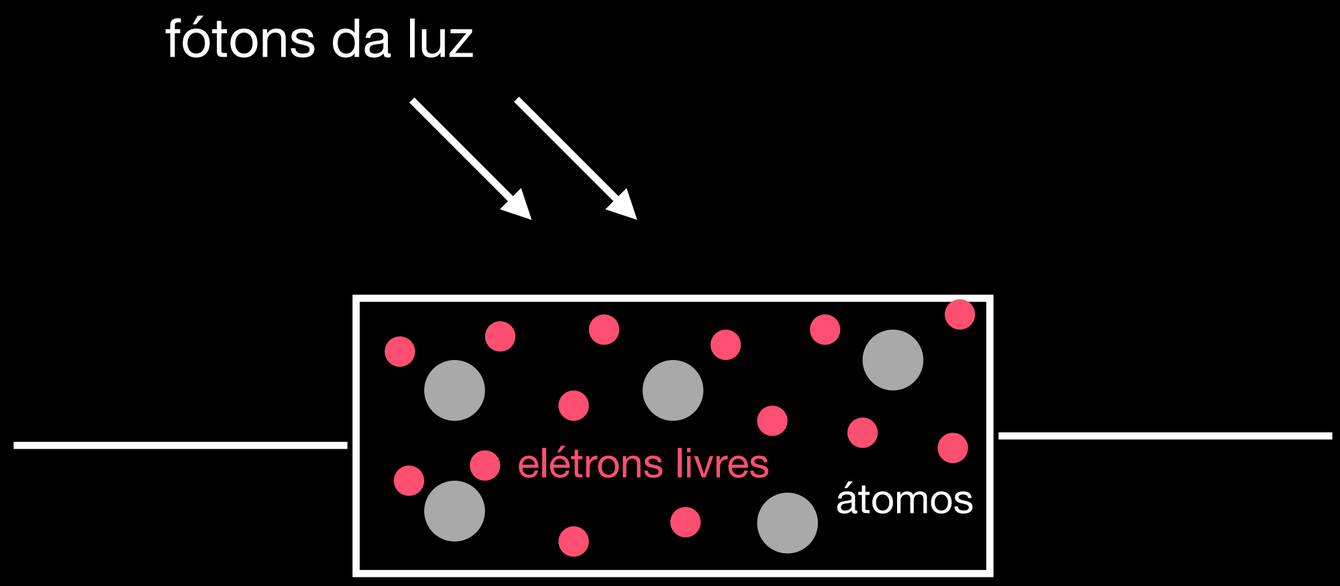
```
>>> timer_recorrente()
>>> def parar_timer():
...     global timer
...     if timer != None:
...         timer.cancel()
...         timer = None
...
```

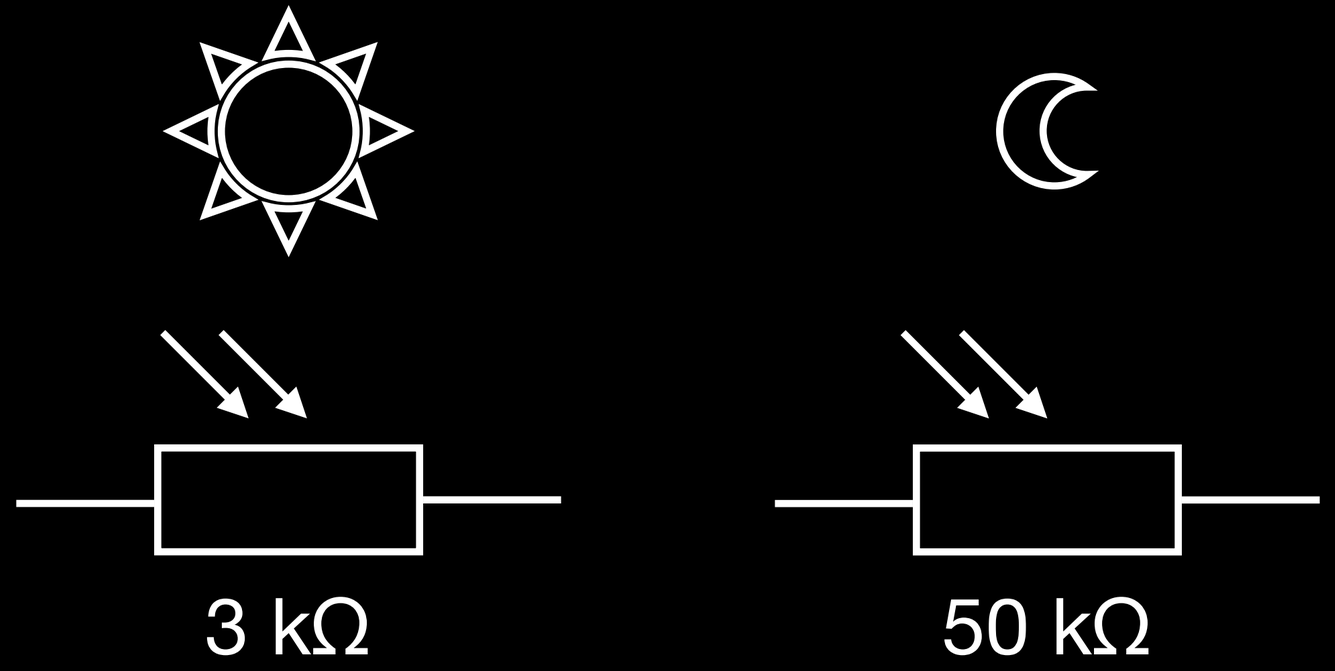
Timer Recorrente e Cancelamento de Timer em Python

Hardware

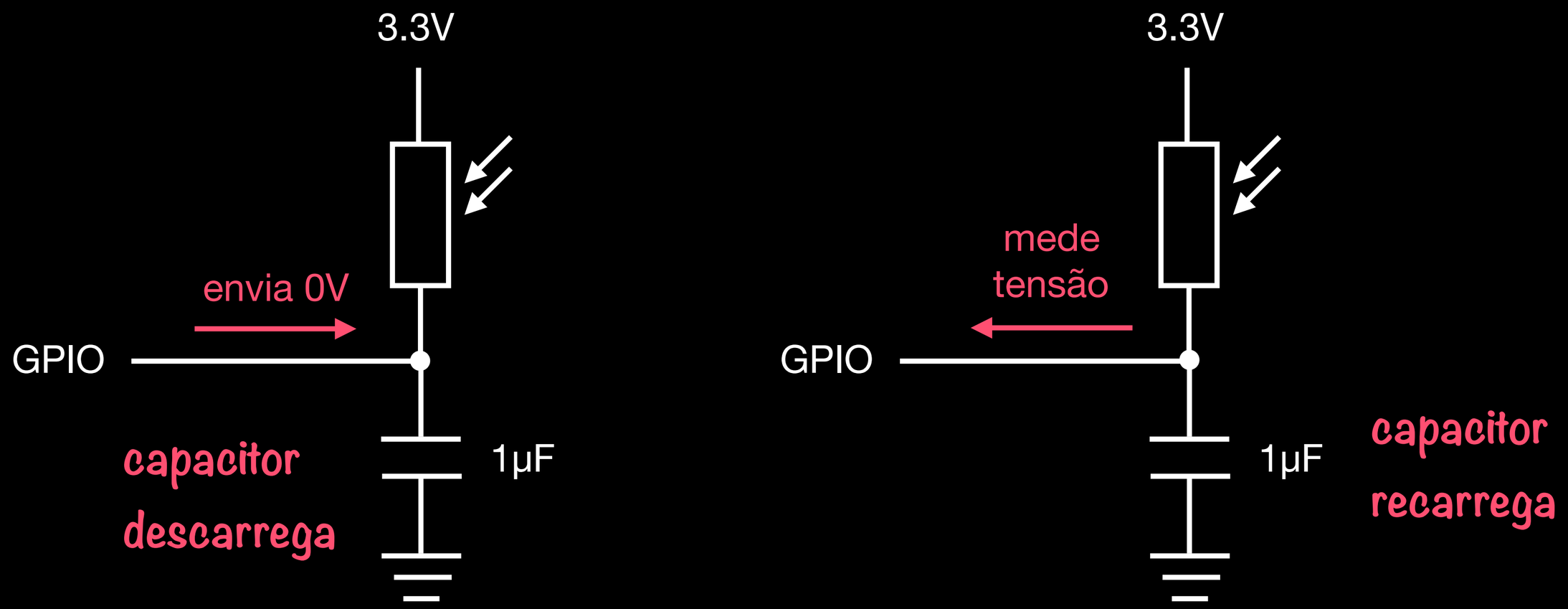


Sensor de Luz

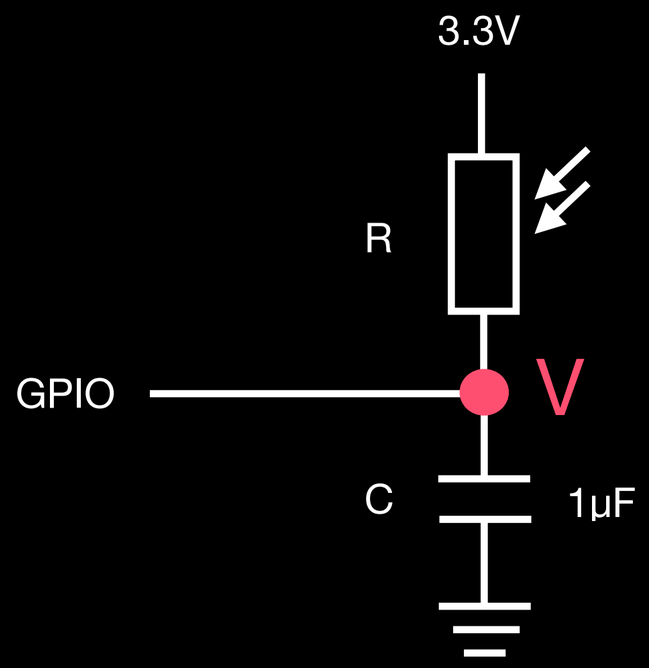




Resistência Dependente de Luz (LDR) ou Fotoresistor

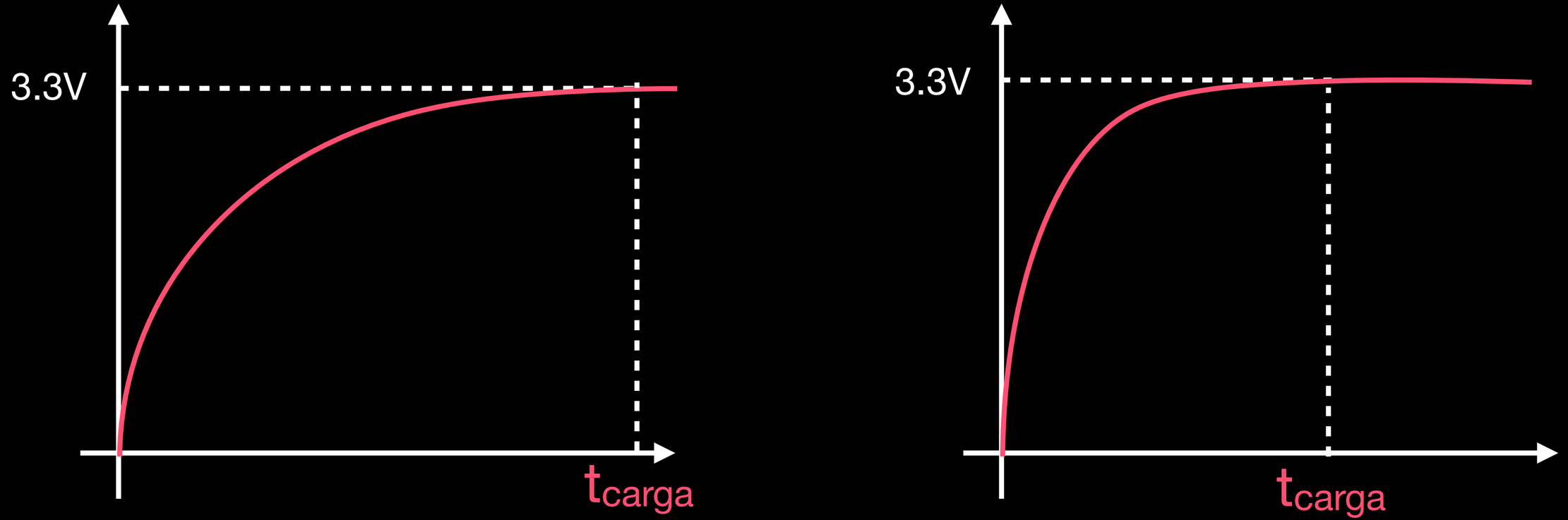


Circuito com Capacitor para Detecção de Luz



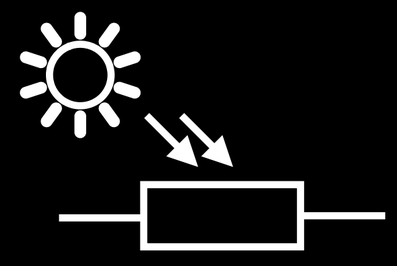
Carregamento sem Luz (R Maior)

Carregamento com Luz (R Menor)

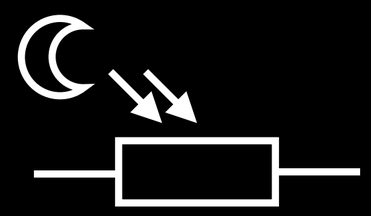


Tempo de Carregamento

```
>>> from gpiozero import LightSensor
>>> sensor_de_luz = LightSensor(8)
>>> sensor_de_luz.value
0.854363
```



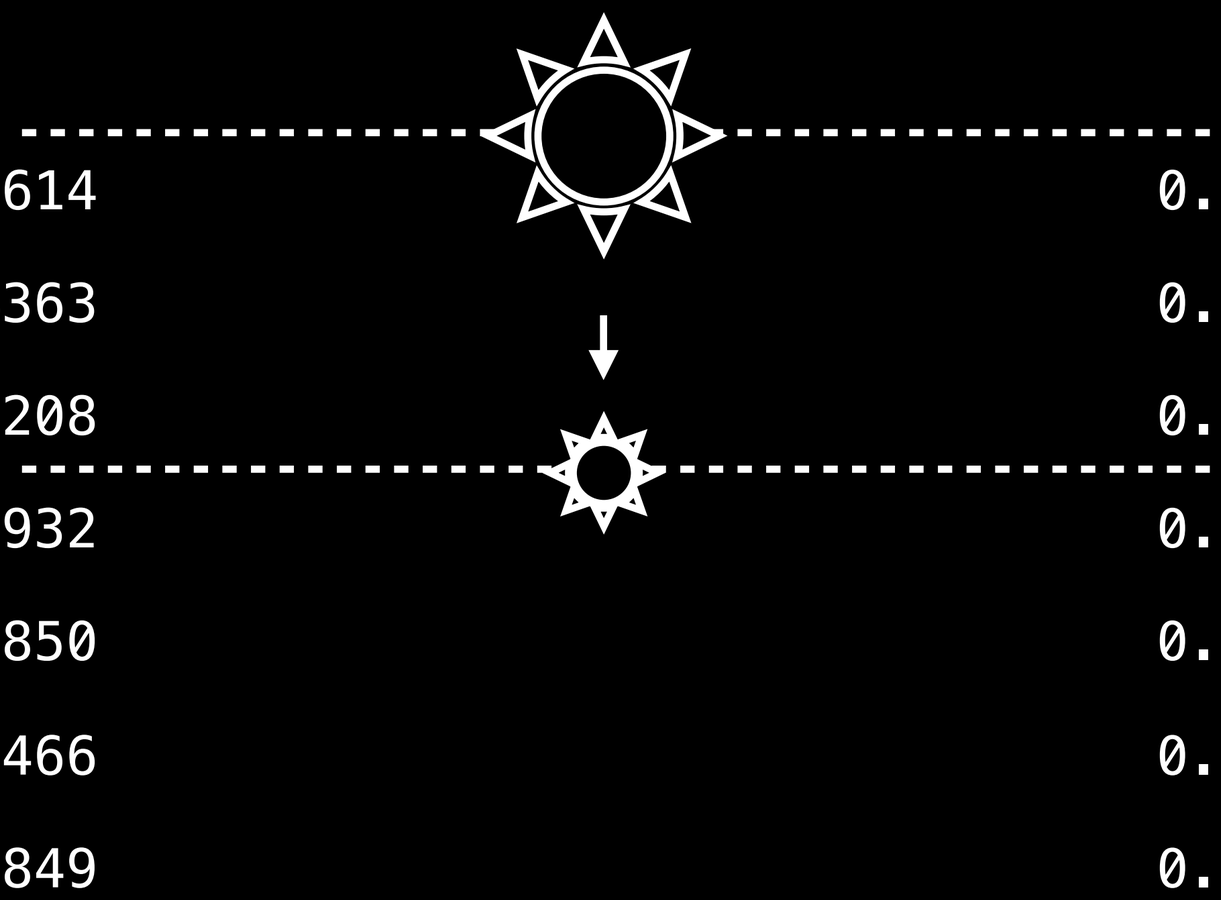
```
>>> sensor_de_luz.value
0.347932
```



Valor de um Sensor de Luz

```
sensor_de_luz = LightSensor(8)
while True:
    print(sensor_de_luz.value)
    sleep(1)
```

```
sensor = LightSensor(8, queue_len=20)
while True:
    print(sensor_de_luz.value)
    sleep(1)
```



Sensibilidade a Mudanças de um Sensor de Luz

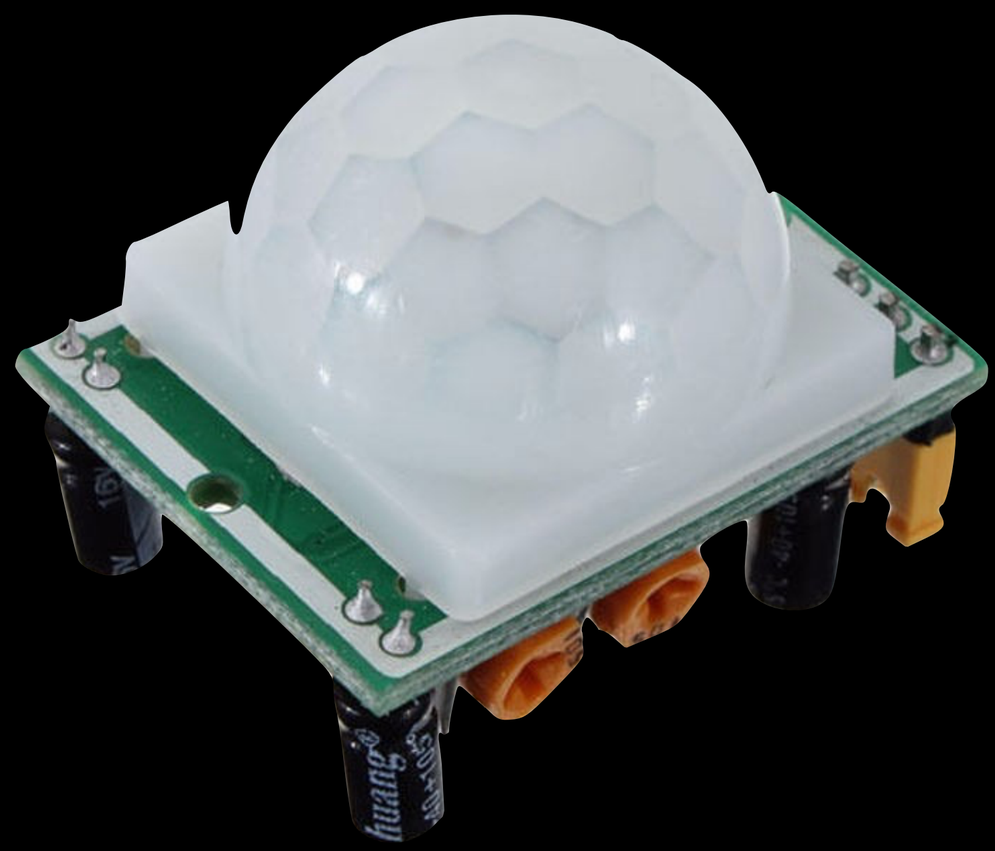
```
>>> from gpiozero import LightSensor

>>> sensor_de_luz = LightSensor(8)

>>> sensor_de_luz.threshold
0.5

>>> sensor_de_luz.value
0.56743

>>> sensor_de_luz.light_detected
True
```

```
>>> sensor_de_luz.threshold = 0.6

>>> sensor_de_luz.value
0.55891

>>> sensor_de_luz.light_detected
False
```

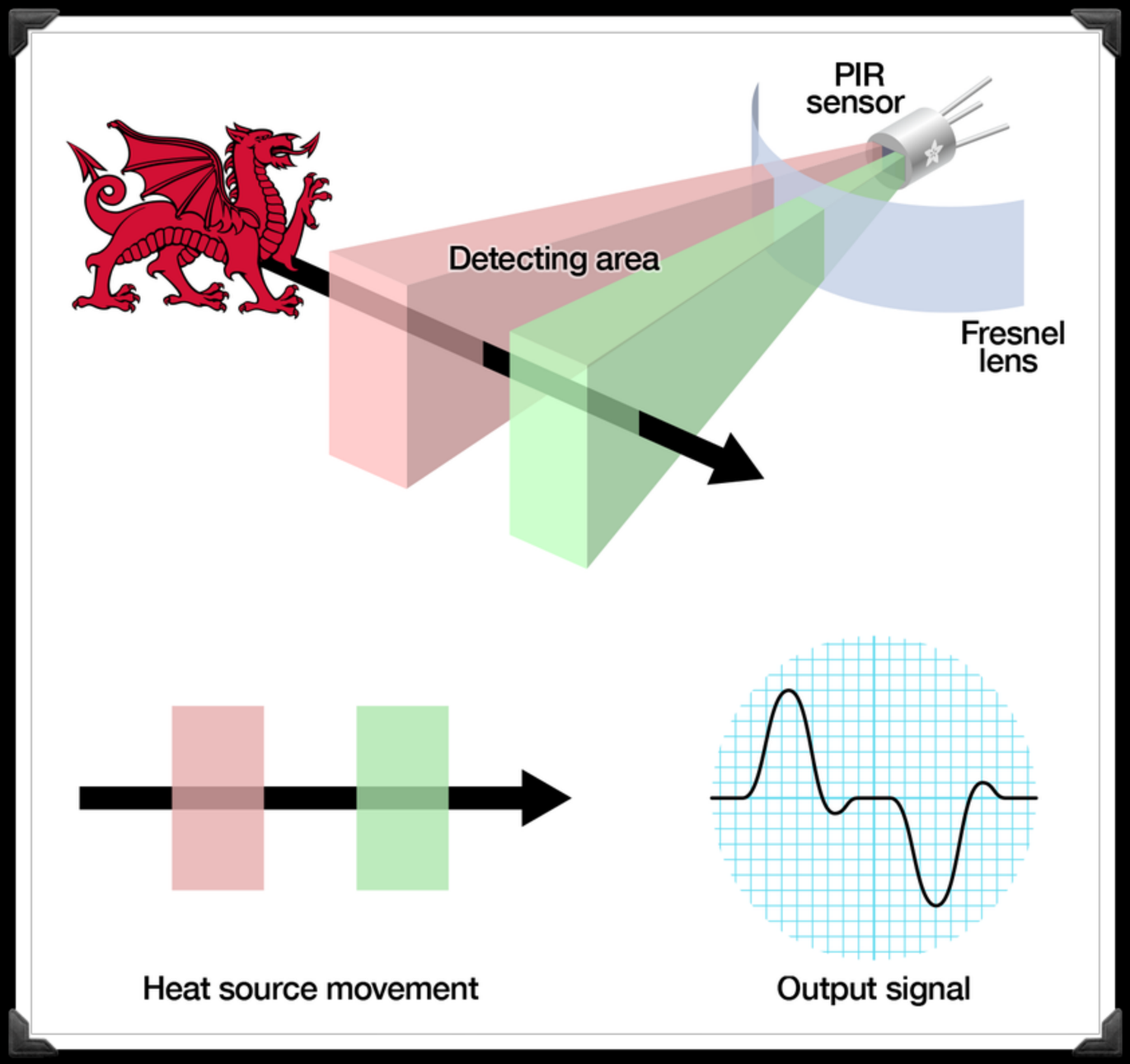
Limiar para Detecção de Luz

```
>>> from gpiozero import LightSensor
>>> sensor_de_luz = LightSensor(8)
>>> sensor_de_luz.threshold = 0.6

>>> def luz_detectada():
...     print("Haja luz!")
...
>>> sensor_de_luz.when_light = luz_detectada

>>> def escuridao_detectada():
...     print("Hello darkness my old friend...")
...
>>> sensor_de_luz.when_dark = escuridao_detectada
```

Eventos de um Sensor de Luz



Sensor de Movimento

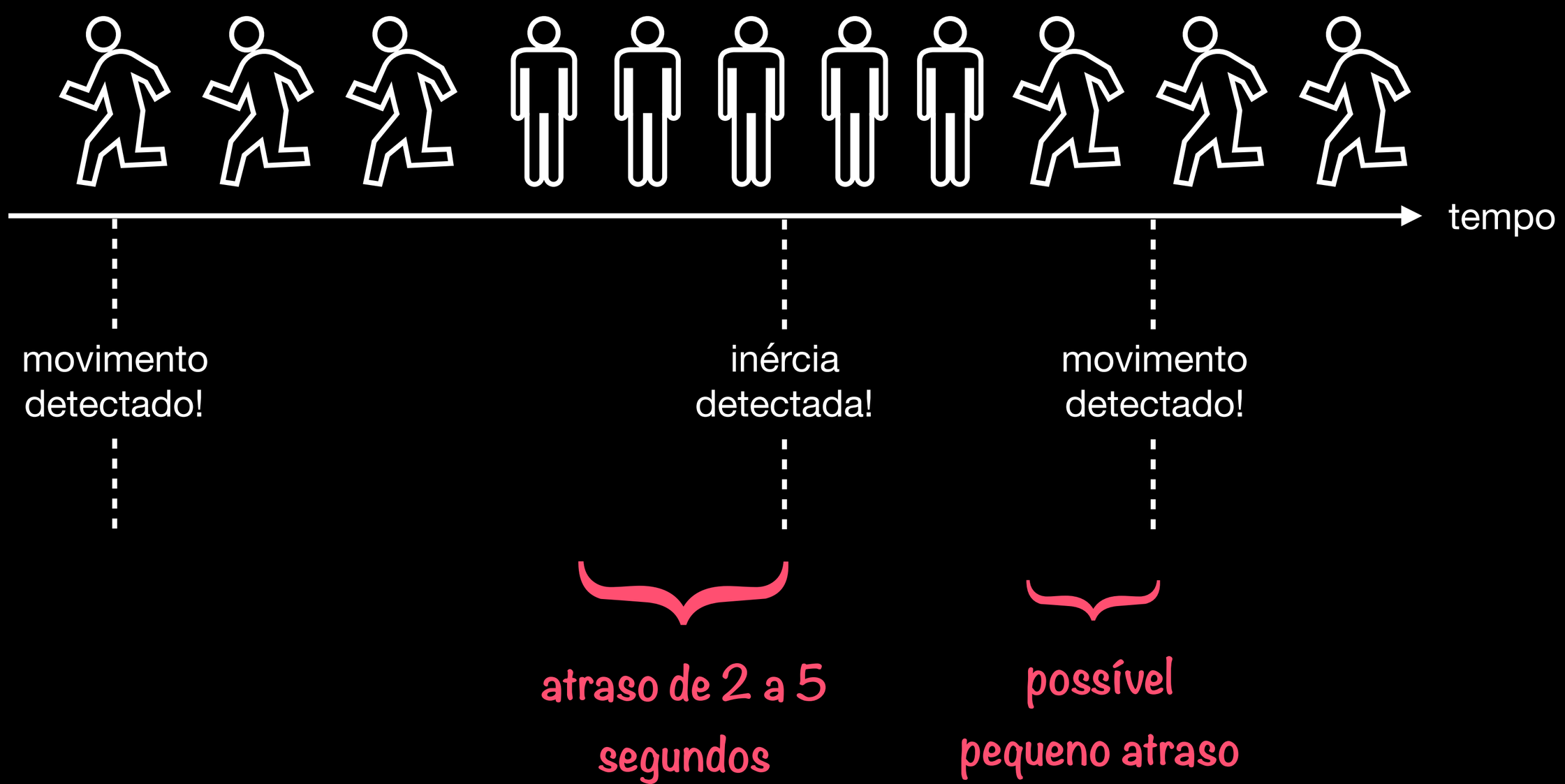


Funcionamento de um Infravermelho Passivo (PIR)

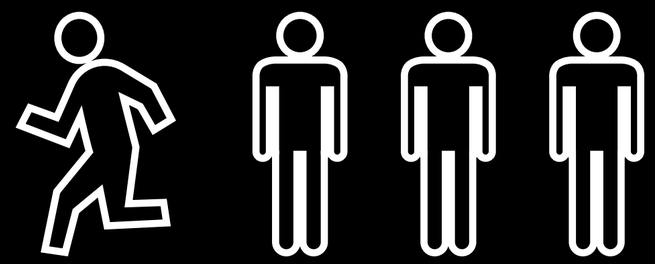
```
>>> from gpiozero import MotionSensor
>>> sensor_de_movimento = MotionSensor(27)
>>> sensor.motion_detected
False
>>> def movimento_detectado():
...     print("Mexe a cadeira, hey!")
...
...
>>> sensor_de_movimento.when_motion = movimento_detectado
>>> def inercia_detectada():
...     print("Stop! Hammer time!")
...
...
>>> sensor_de_movimento.when_no_motion = inercia_detectada
```

Eventos do Detector de Movimento

Atraso de Detecção nos Sensores de Movimento



Atraso de Detecção nos Sensores de Movimento



:

inércia
detectada!

Como faz para
esperar mais tempo?

atraso de 2 a 5
segundos

atraso de mais alguns segundos

ao detectar inércia

↓

inicie um Timer
para esperar mais tempo

ao detectar movimento

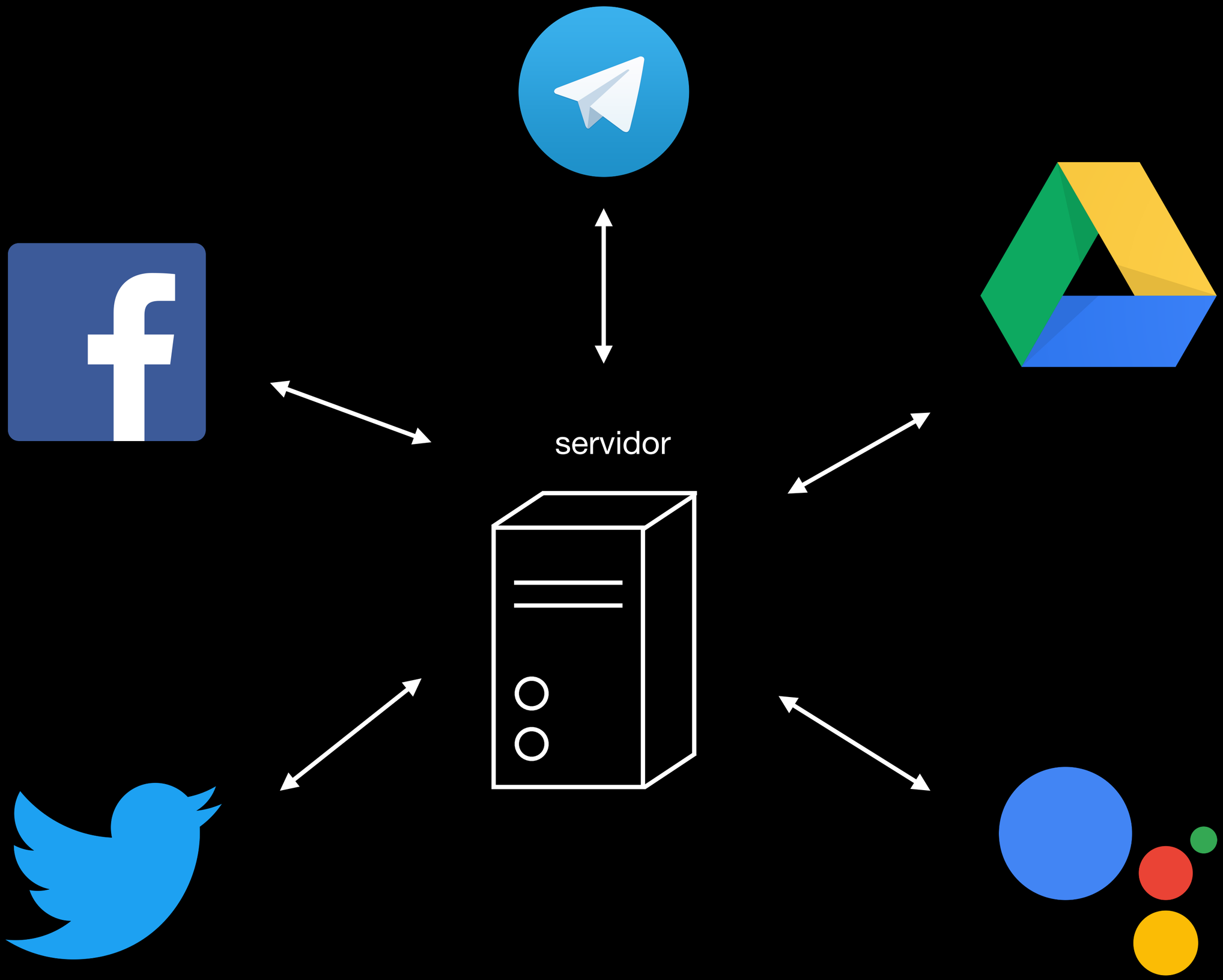
↓

cancele o Timer

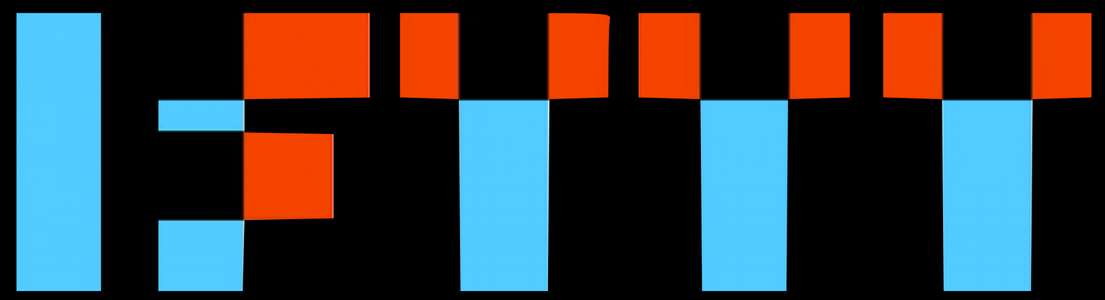
Acréscimo no Tempo de Espera da Inércia

Software

Uso de Softwares dos Projetos Passados



Dificuldade em Programar Várias Comunicações



"applet" → **if this than that**

gatilho

ação

IFTTT: If This Than That

ifttt.com

IFTTT

Sign in

Sign up

A world that works for you

IFTTT is the free way to get all your apps and devices talking to each other. Not everything on the internet plays nice, so we're on a mission to build a more connected world.

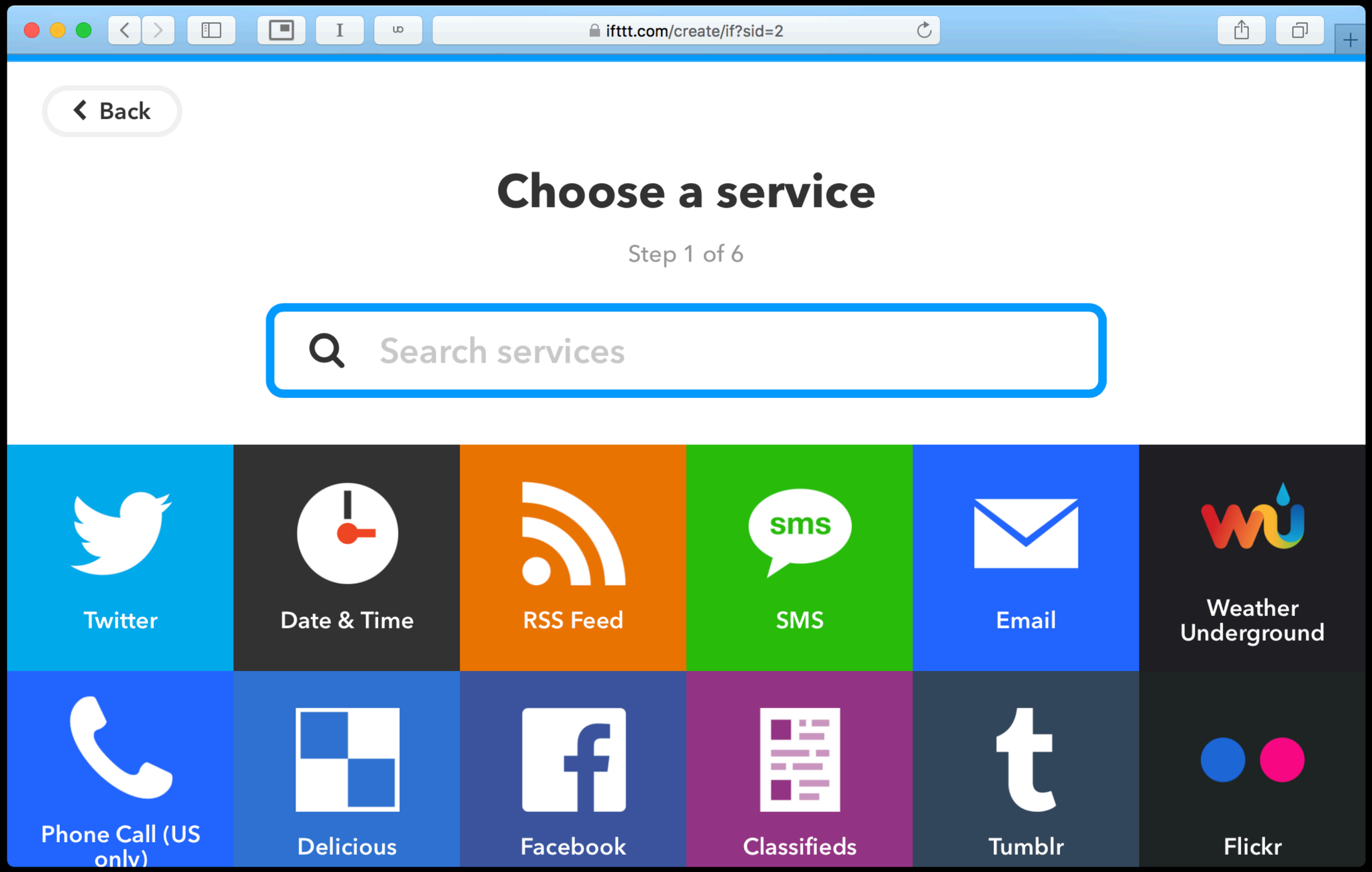
Enter your email

Get started

Continue with Google

Continue with Facebook

Integração entre Serviços com IFTTT



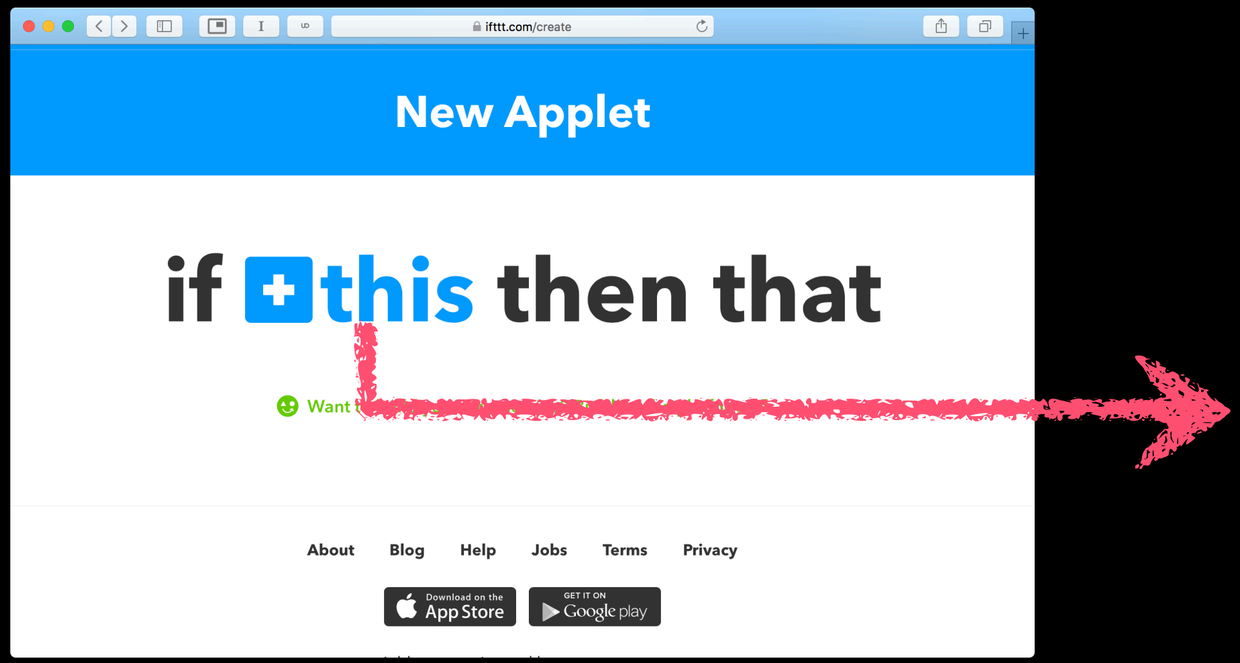
Lista de Serviços Compatíveis

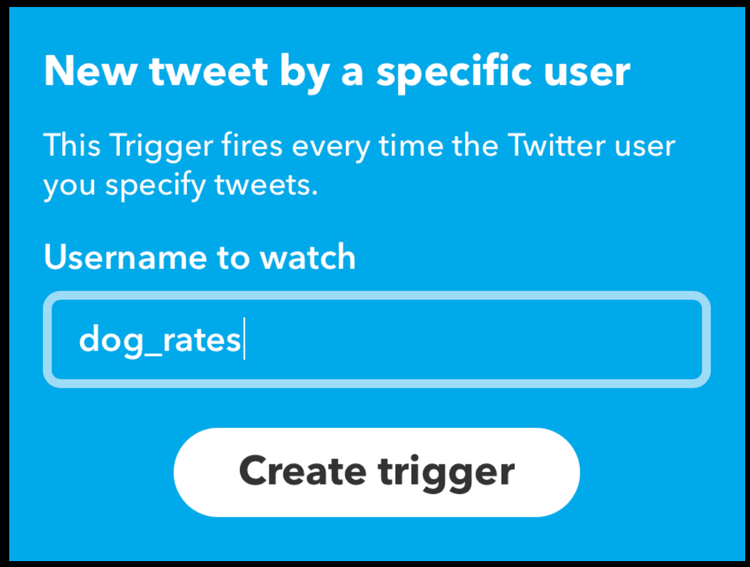
então

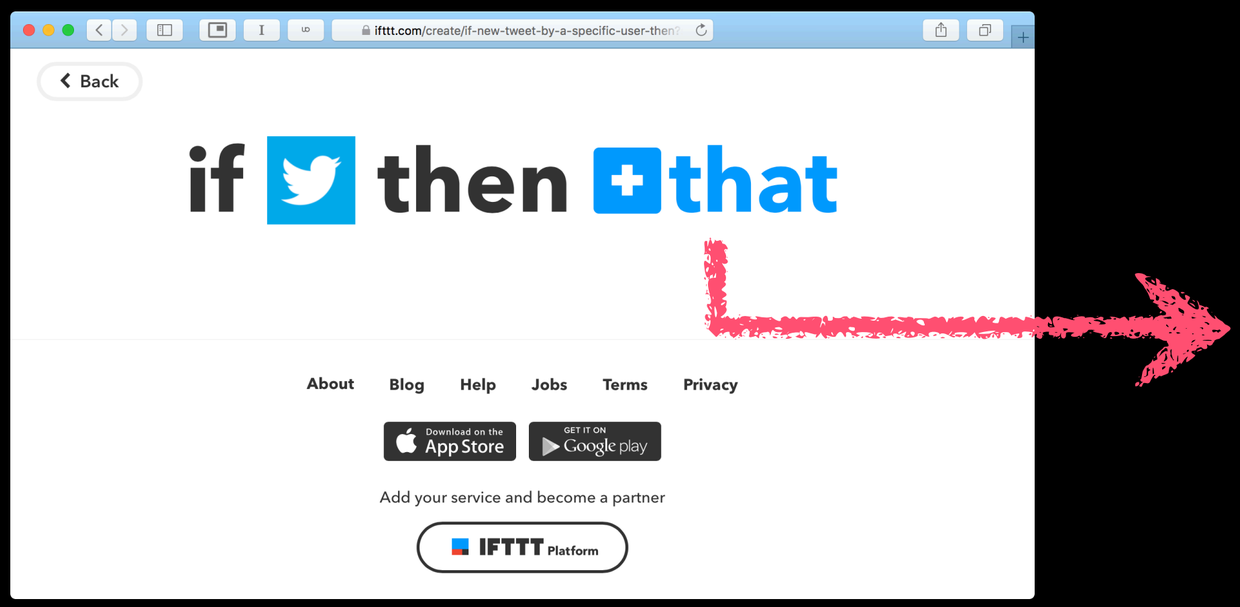
se um usuário tuitar

mande-me um SMS

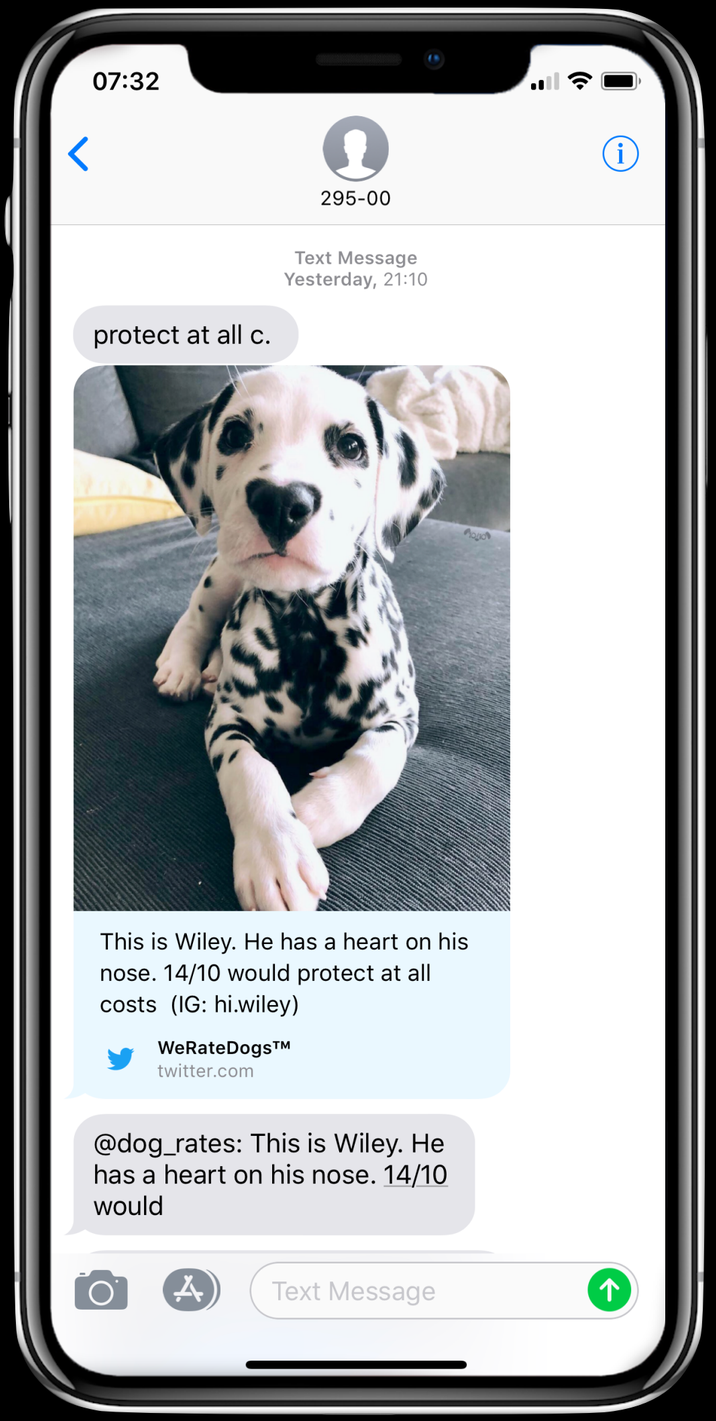
Exemplo de Integração entre Twitter e SMS



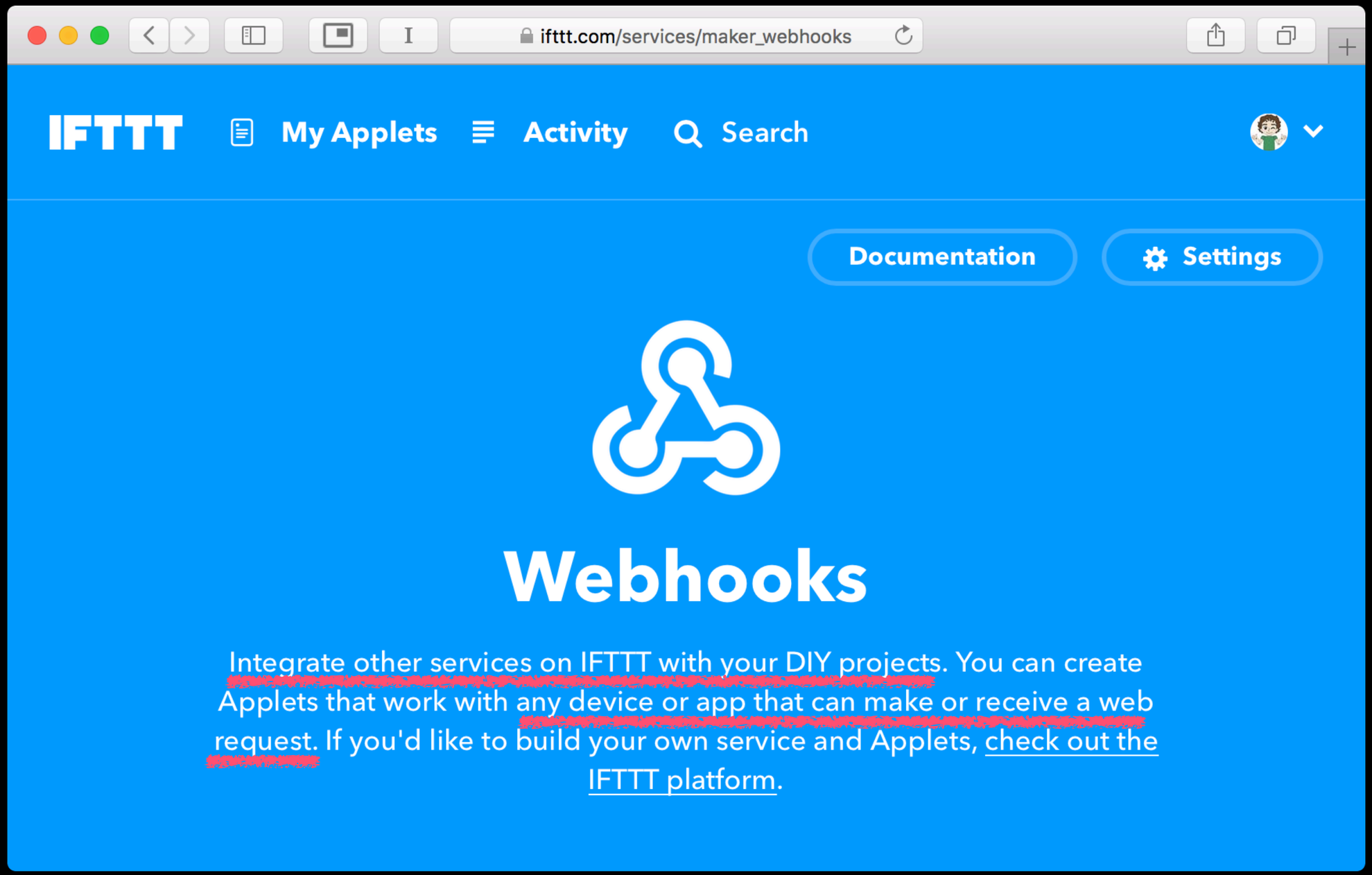




Configuração do "This" e do "That"

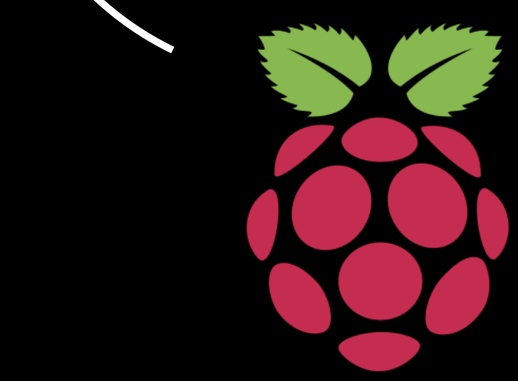


SMS com Tweets de um Perfil

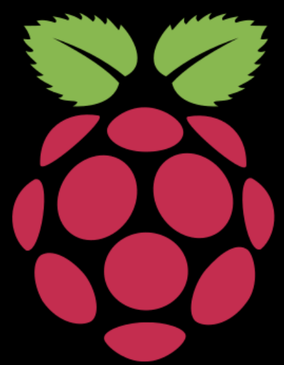


Opção Webhook do IFTTT

if this than that



Integração entre Raspberry Pi e Serviços

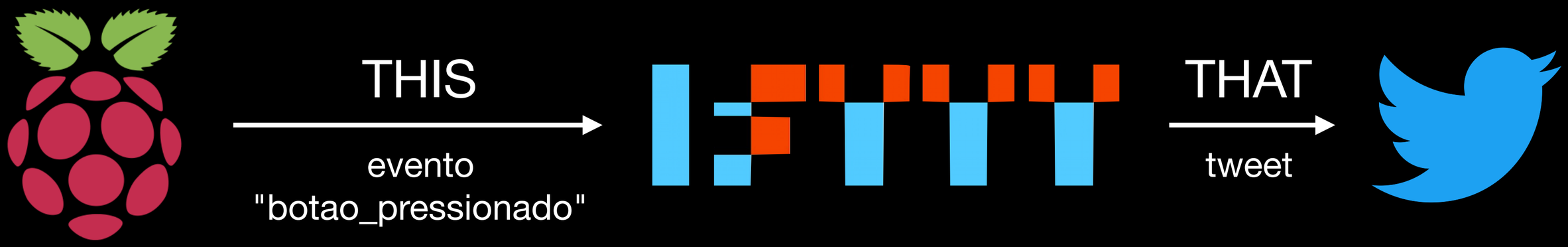


então

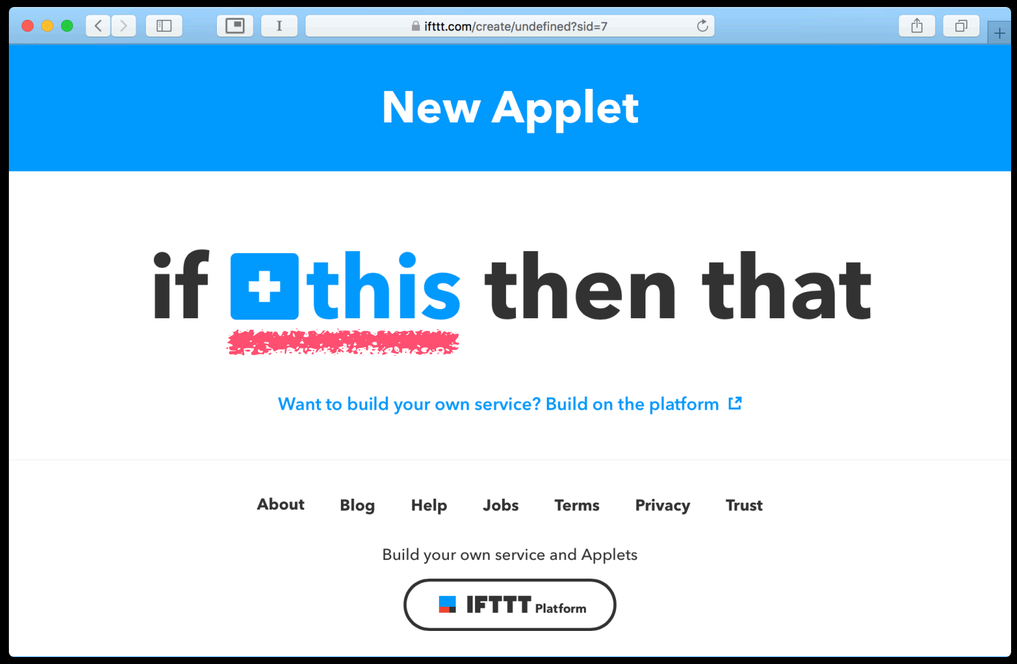
se botão for
pressionado

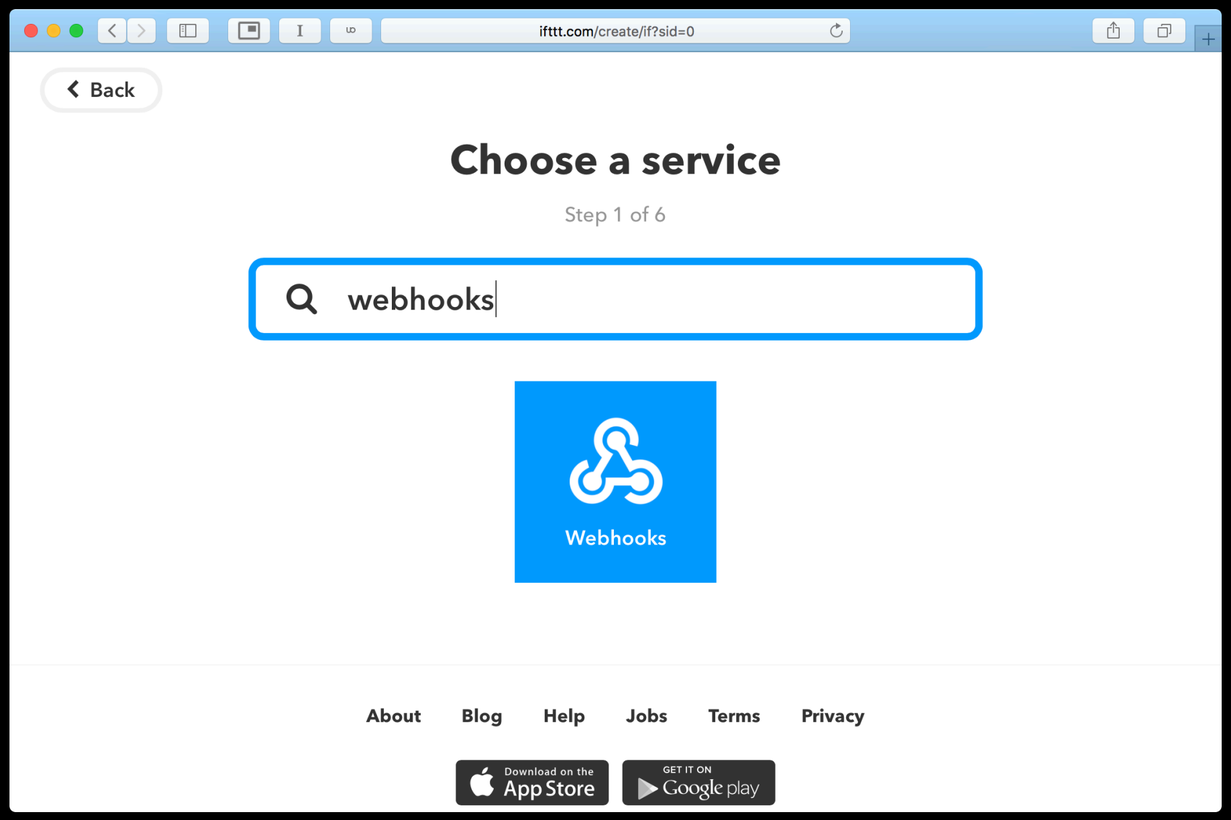
poste um
tweet

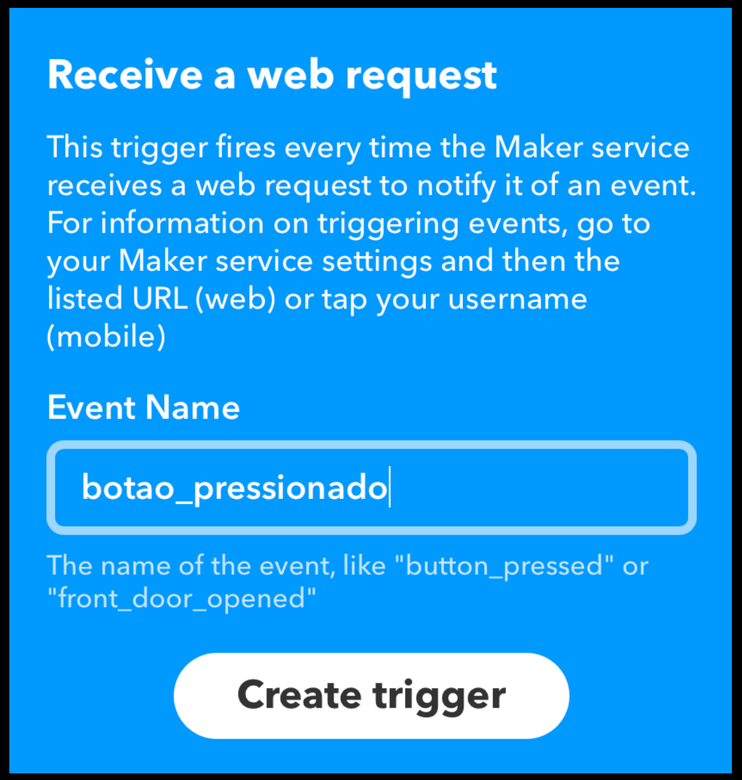
Exemplo de Integração entre Raspberry Pi e Twitter



Envio de Eventos ao IFTTT

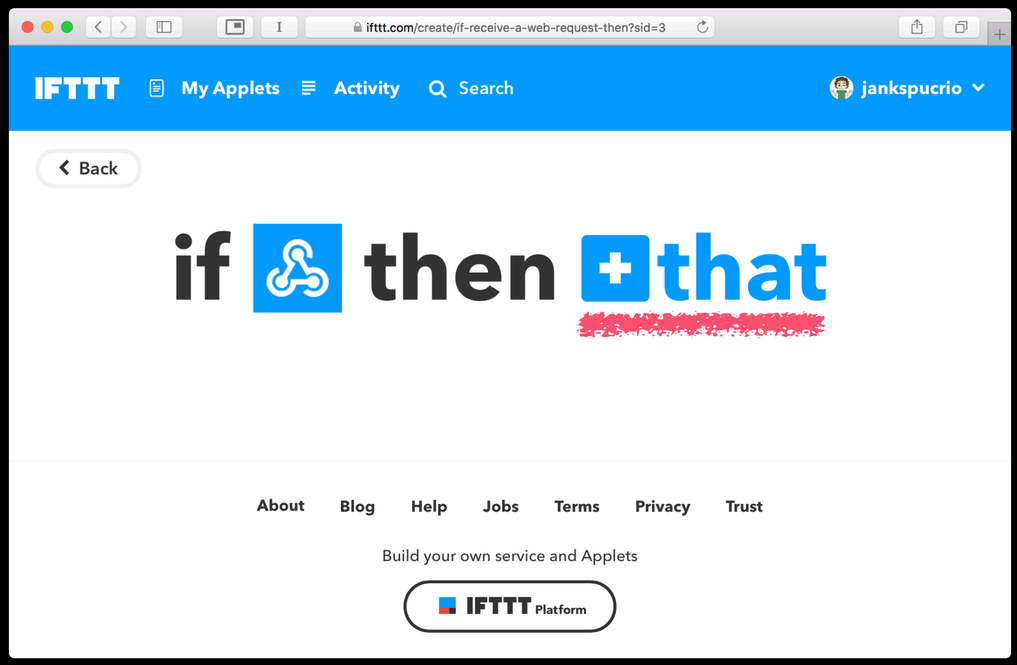


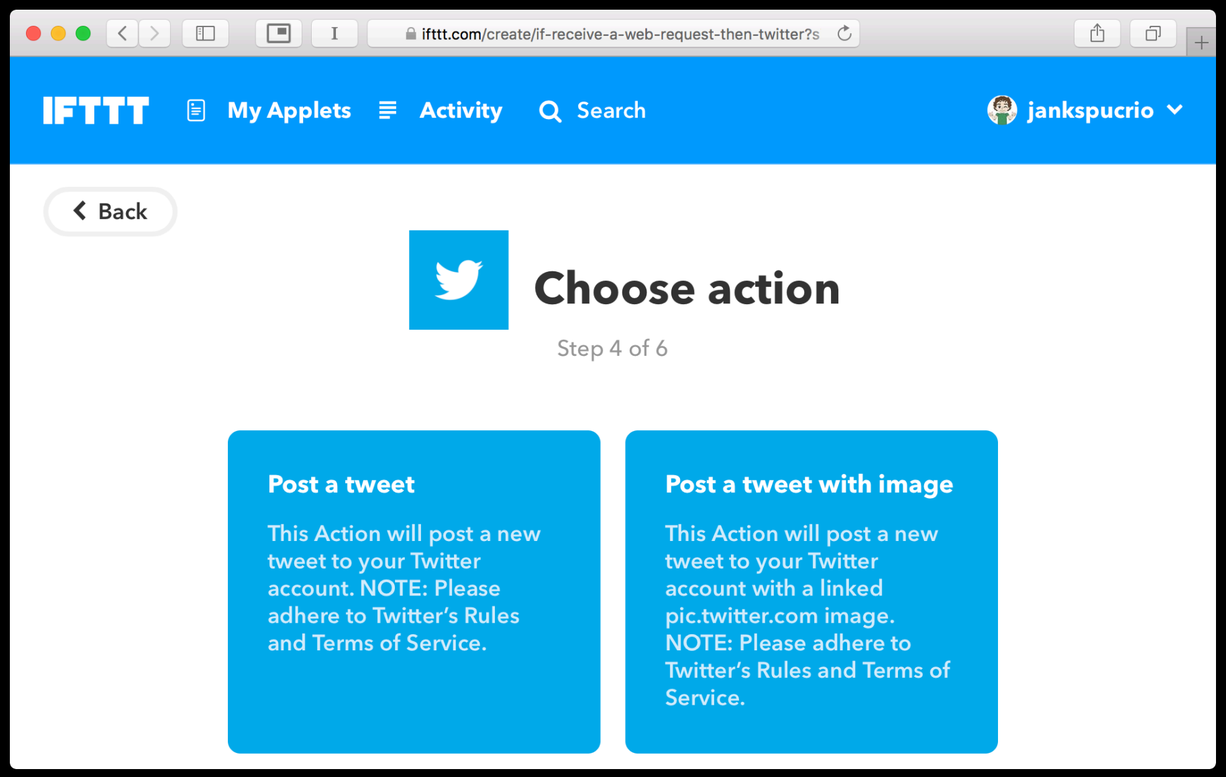


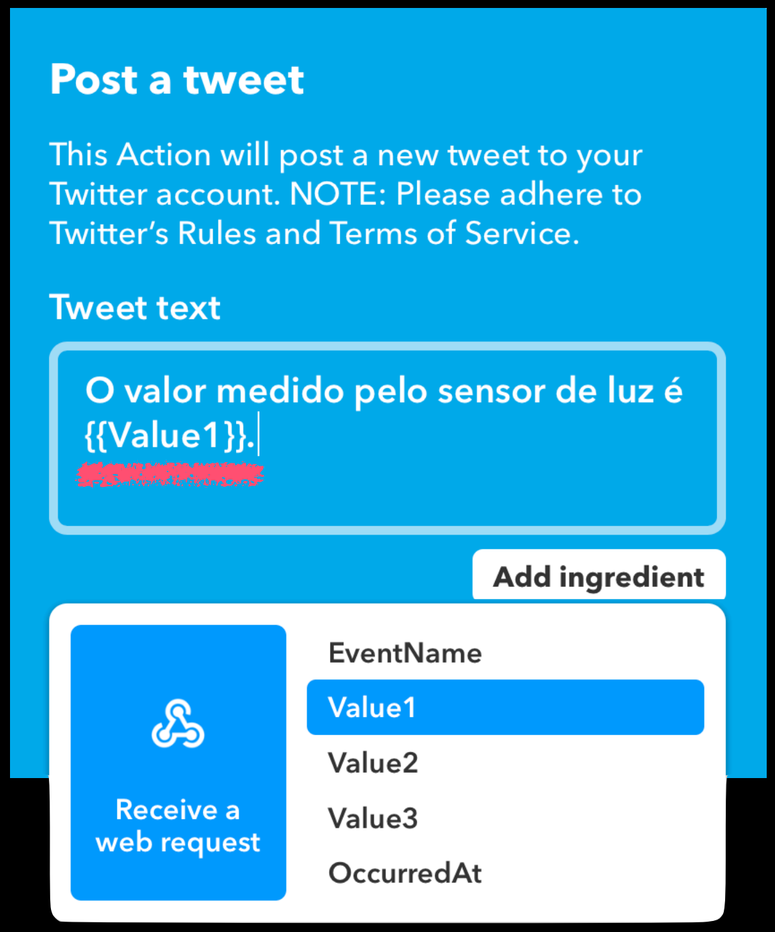


nome do "evento"

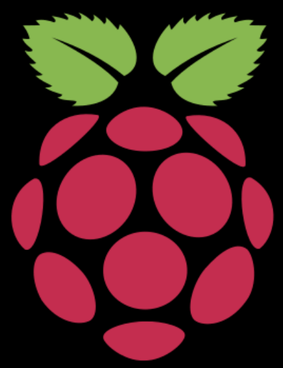
Configuração do "This" como Serviço Webhook





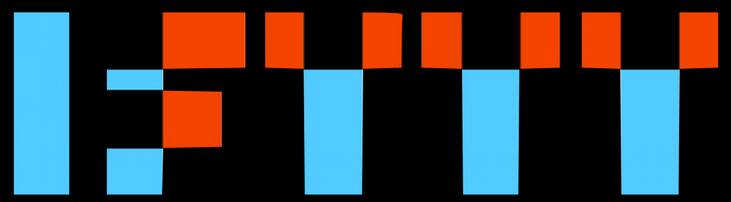


Configuração do "That" como Tweet com Parâmetros



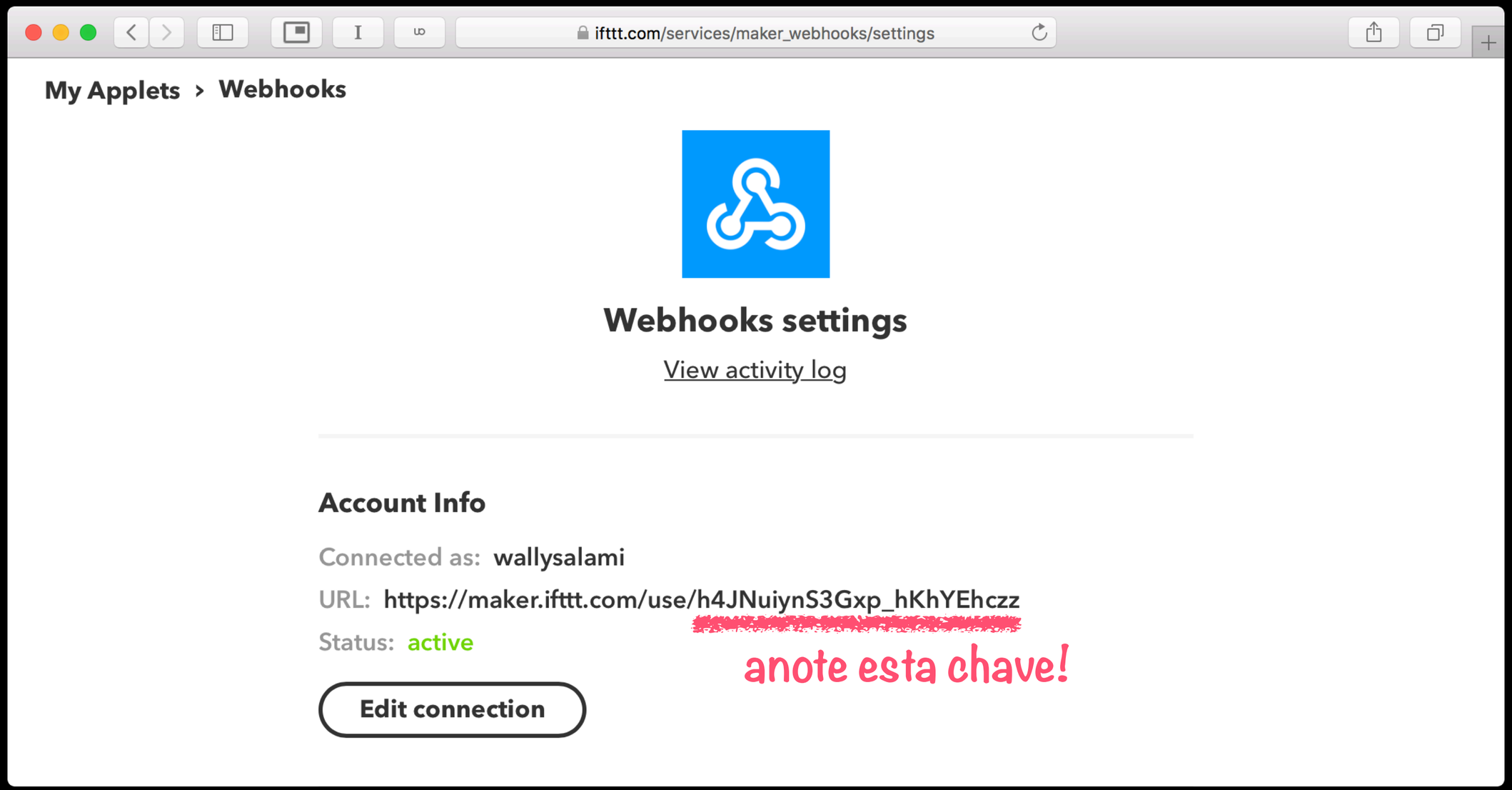
post com dados

post com dados



Envio de Requisições ao IFTTT

https://ifttt.com/services/maker_webhooks/settings



Chave Secreta do IFTTT

```
>>> chave = "COLOQUE SUA CHAVE AQUI"
>>> evento = "botao_pressionado"
>>> endereco = "https://maker.ifttt.com/trigger/" +
evento + "/with/key/"  + chave
```

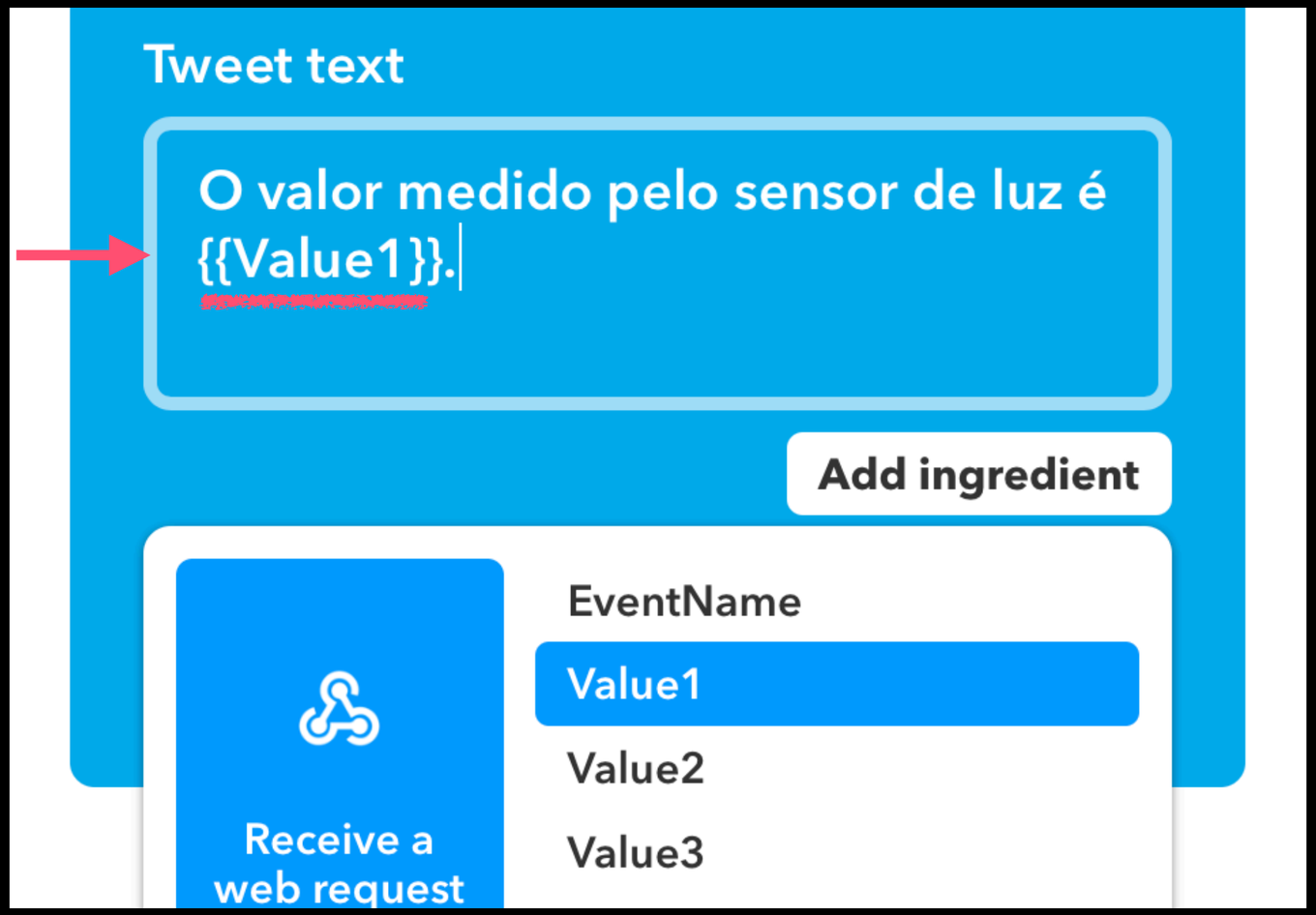
Configuração do Endereço do IFTTT

```
>>> from gpiozero import LightSensor, Button
>>> from requests import post
>>> sensor_de_luz = LightSensor(8)
>>> def botao_pressionado():
...     dados = {"value1": sensor_de_luz.value}
...     resultado = post(endereco, json=dados)
...     print(resultado.text)
...
>>> botao = Button(11)
>>> botao.when_pressed = botao_pressionado
```

Configuração do Sensor de Luz e do Botão

Tweet Gerado pelo Raspberry Pi

"V" MAIÚSCULO



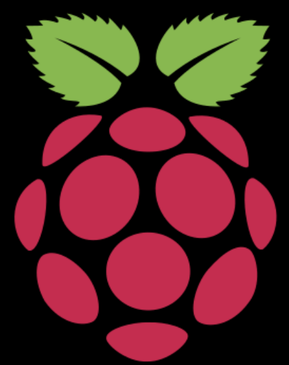
"v" minúsculo

```
dados = {"value1": sensor_de_luz.value}
```

Atenção à Variação da Ortografia de Value1

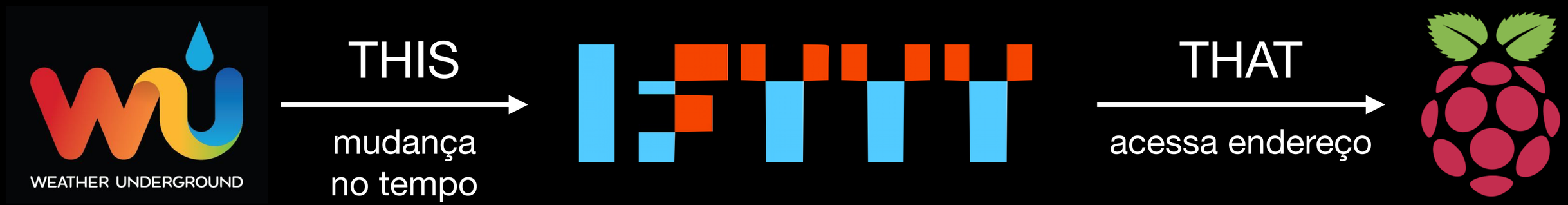
se houver
previsão do tempo

então

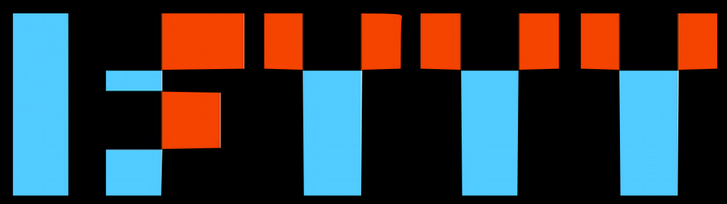


mostre a previsão
no LCD

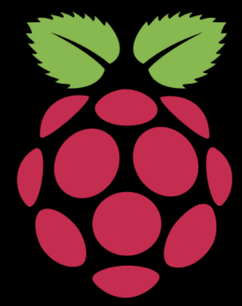
Exemplo de Integração entre Weather Underground e Raspberry Pi



Recebimento de Eventos do IFTTT

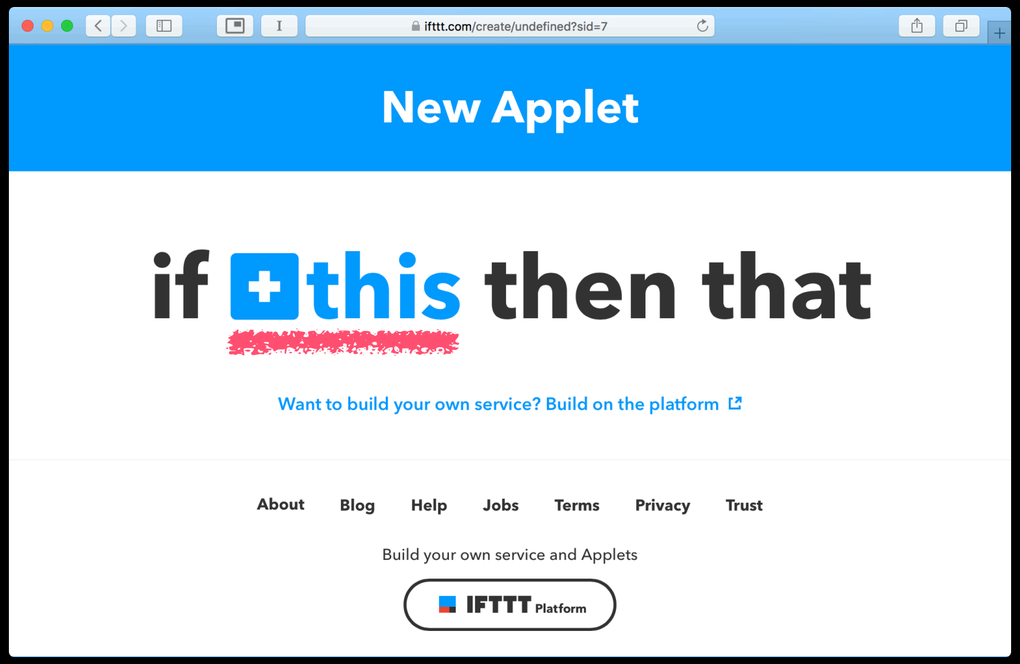


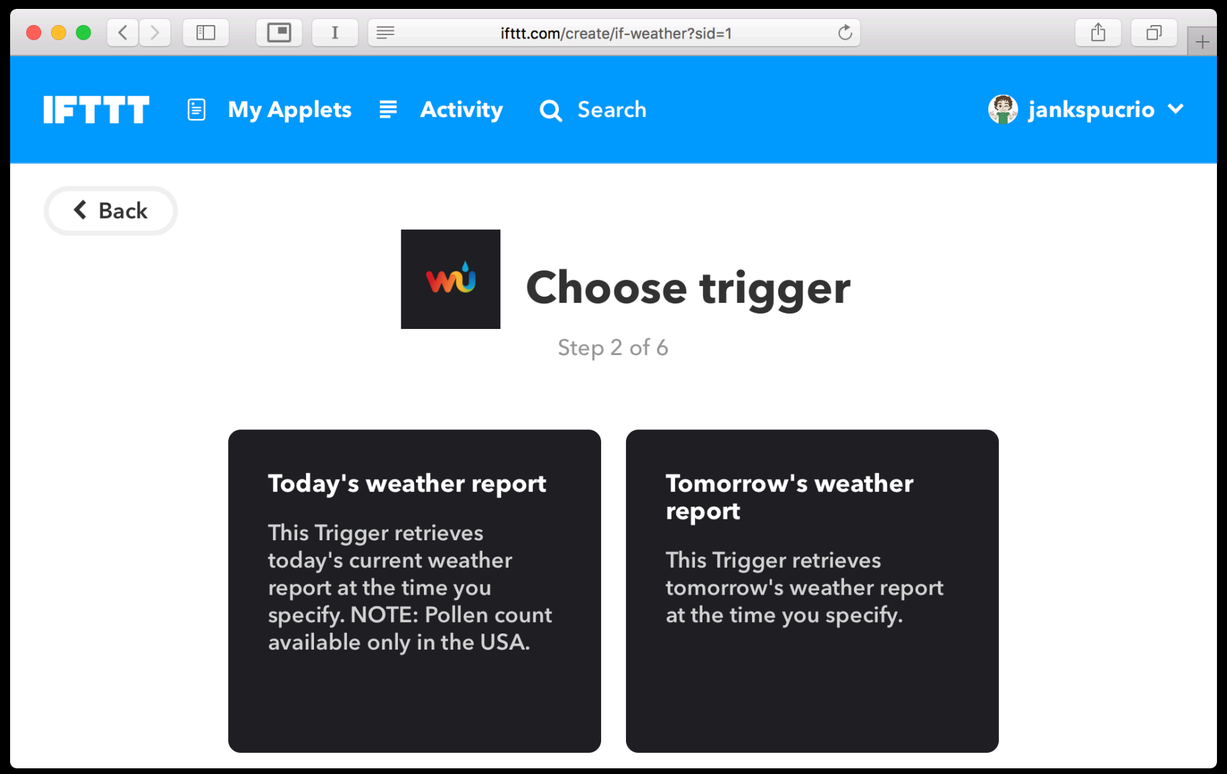
https://.../tempo/rain



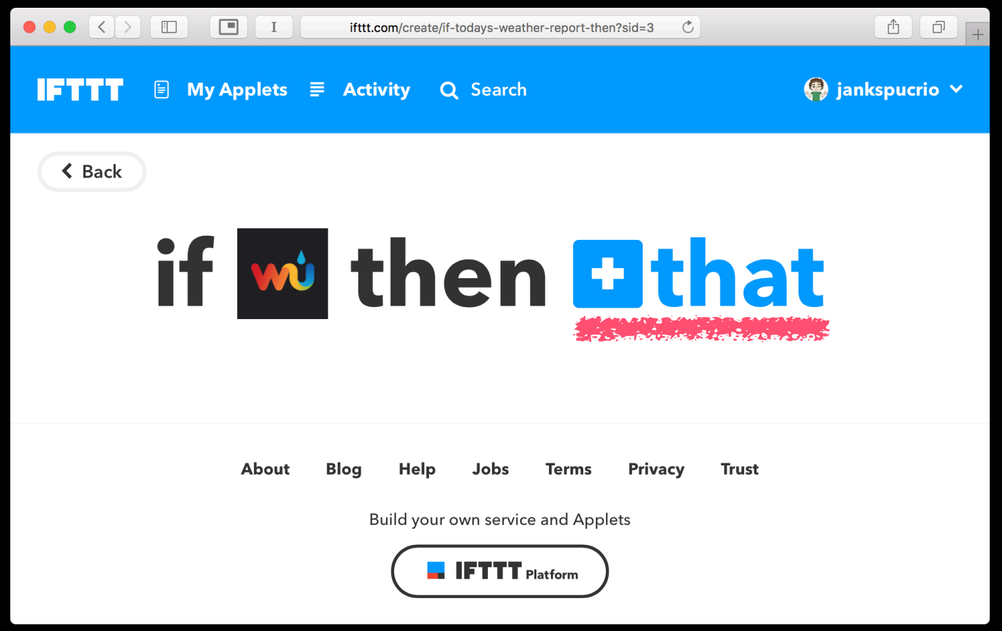
https://.../tempo/sunny

Integração entre Weather Underground e Raspberry Pi via Requisições Web

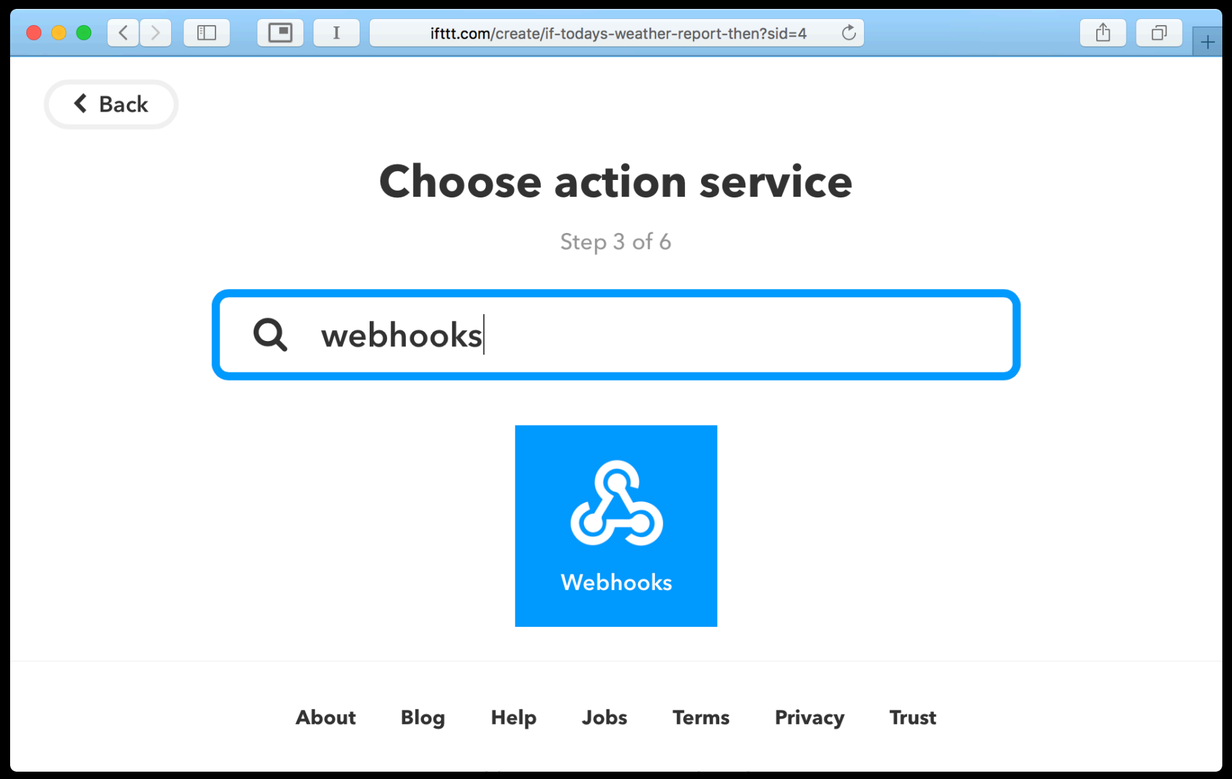


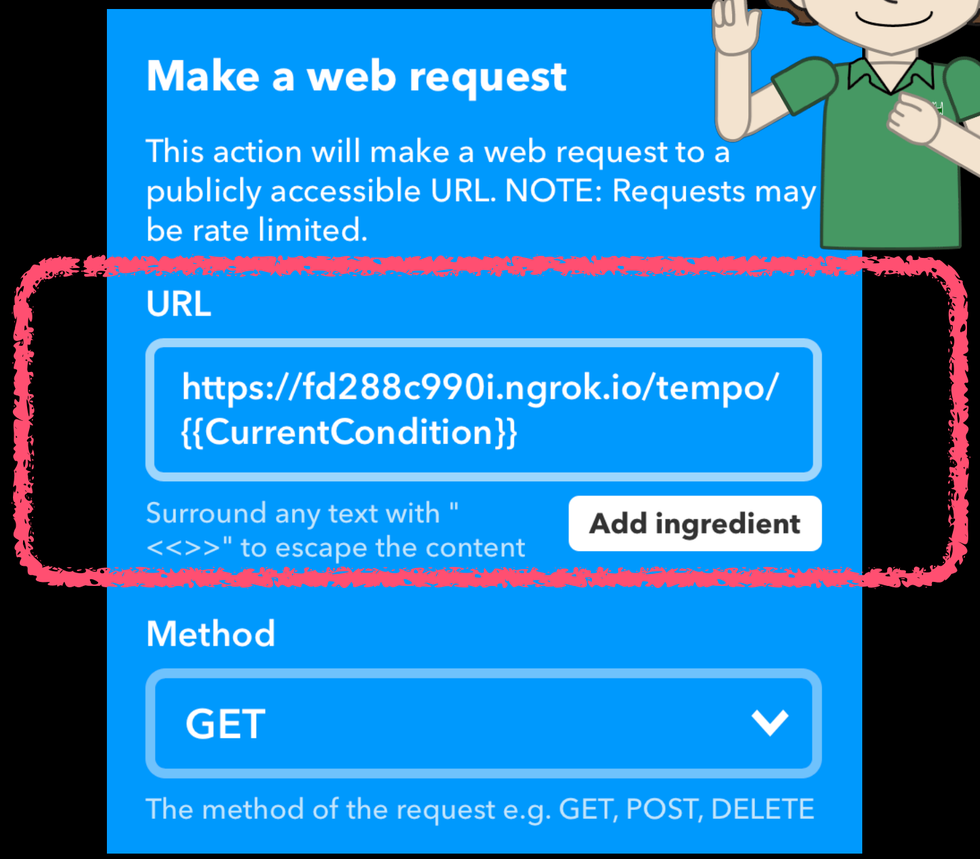


Configuração do Weather Underground como "This"

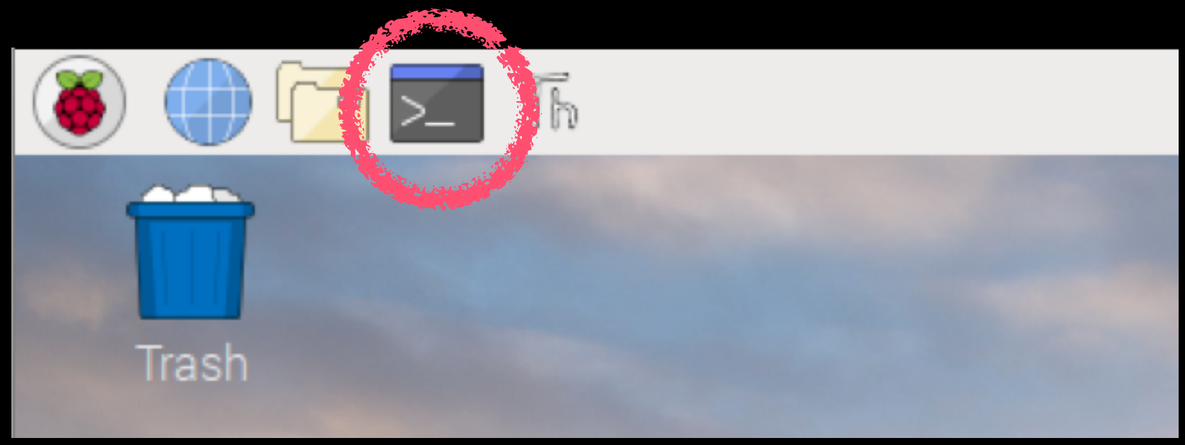


ATENÇÃO: não deixe espaços em branco no endereço!





Configuração do Webhook como "That"




```
aula@raspberrypi ~ $ ngrok http 5000

ngrok by @inconshreveable
(Ctrl+C to quit)

Session Status                online
Session Expires               7 hours, 59 minutes
Version                       2.2.8
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://fd288c990i.ngrok.io -> localhost:5000
Forwarding                    https://fd288c990i.ngrok.io -> localhost:5000

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Criação de um Endereço Público com o Ngrok

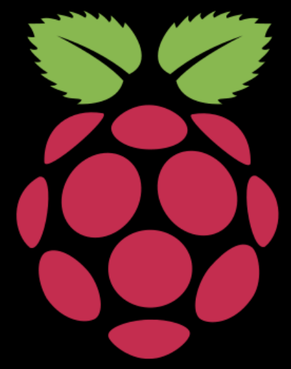
```python
from flask import Flask
from Adafruit_CharLCD import Adafruit_CharLCD
from gpiozero import Buzzer

app = Flask(__name__)
lcd = Adafruit_CharLCD(2, 3, 4, 5, 6, 7, 16, 2)
campainha = Buzzer(16)

@app.route("/tempo/<string:previsao>")
def tempo(previsao):
    lcd.message("Tempo hoje:\n" + previsao)
    if previsao == "Rain":
        campainha.beep(n=5)

    return "A previsão do tempo é: " + previsao

app.run(port=5000)
```

Configuração do Servidor com Página para Previsão do Tempo

Previsão de Tempo no LCD

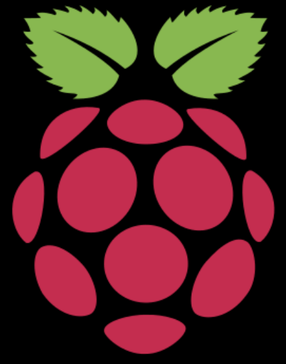
Uber

então



se viagem para
casa for concluída

ligue a luz e a TV
e salve dados no banco



então

se eu ando chegando
muito tarde em casa

agende um horário
para meditar após almoço

Integração com Uber e Agenda

this

Ride completed

This trigger fires when your Uber ride has ended.

Trigger when pickup location is

Anywhere

Be sure to set "Home" and "Work" addresses in the Uber app

and drop off location is

Home

Be sure to set "Home" and "Work" addresses in the Uber app

Create trigger

that

Make a web request

This action will make a web request to a publicly accessible URL. NOTE: Requests may be rate limited.

URL

https://fd288c990i.ngrok.io/chegue i_em_casa/ VehicleLicensePlate

Surround any text with "<<>>" to escape the content

Add ingredient

Method

GET

The method of the request e.g. GET, POST, DELETE

Primeira Parte: Integração com Uber

this

Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

Event Name

preciso_descansar

The name of the event, like "button_pressed" or "front_door_opened"

Create trigger

that

Create a detailed event

This action will create a detailed event in your Google Calendar.

Which calendar?

Teste

Start time

tomorrow at 13PM

Ex. 10AM, Next Monday at 3PM.

Add ingredient

End time

tomorrow at 14PM

Add ingredient

Segunda Parte: Integração com Google Calendar

```python
from flask import Flask

app = Flask(__name__)

@app.route("/cheguei_em_casa/<string:placa_do_carro>")
def querida_cheguei(placa_do_carro):
    ligar_aparelhos()
    registrar_chegada(placa_do_carro)

    if trabalhando_muito():
        providenciar_descanso()

    return "Cheguei em casa!"

app.run(port=5000)
```

Combinação de Todas as Ações

```python
from gpiozero import LED
from py_irsend.irsend import send_once

luz = LED(21)

def ligar_aparelhos():
    luz.on()
    send_once("TV", ["KEY_POWER"])
    sleep(6)  # espera a TV ligar
    send_once("TV", ["KEY_5", "KEY_4", "KEY_0"])
```

Função para Ligar Luz e TV

```python
from pymongo import MongoClient
from datetime import datetime

cliente = MongoClient("localhost", 27017)
banco = cliente["casa"]
colecao = banco["viagens"]

def registrar_chegada(placa_do_carro):
    agora = datetime.now()
    dados = {"chegada": agora, "placa": placa_do_carro}
    if agora.hour > 9 and agora.hour < 20:
        dados["tarde"] = False
    else:
        dados["tarde"] = True  # chegou muito tarde

    colecao.insert(dados)
```

Função para Salvar Dados da Viagem no Banco de Dados

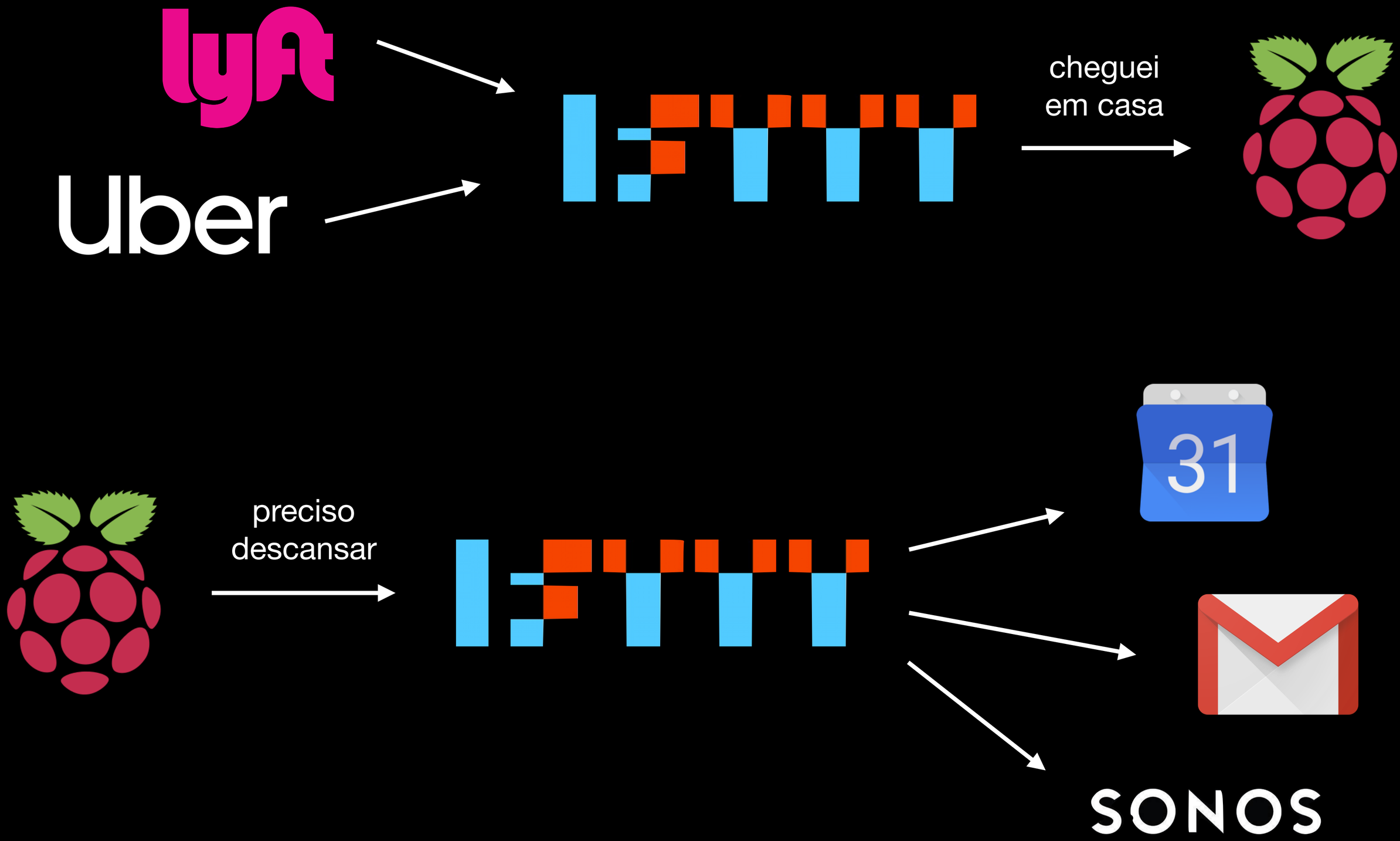
```
from pymongo import MongoClient, DESCENDING
from datetime import datetime

cliente = MongoClient("localhost", 27017)
banco = cliente["casa"]
colecao = banco["viagens"]

def trabalhando_muito():
    agora = datetime.now()
    ha_seis_dias = agora - timedelta(days=6)
    busca = {"tarde": True, "chegada": {"$gt": ha_seis_dias}}
    viagens_tardias = list( colecao.find(busca) )
    total_de_viagens_tardias = len(viagens_tardias)

    if total_de_viagens_tardias > 4:
        return True
    else:
        return False
```

Função para Buscar Dados sobre as Viagens

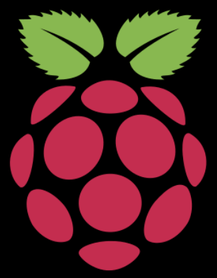
```
from requests import post

chave = "COLOQUE SUA CHAVE AQUI"
evento = "preciso_descansar"
endereco = "https://maker.ifttt.com/trigger/" + evento
+ "/with/key/"  + chave
```

```
def providenciar_descanso():
    dados = {}
    resultado = post(endereco, json=dados)
```

Função para Enviar Pedido ao IFTTT



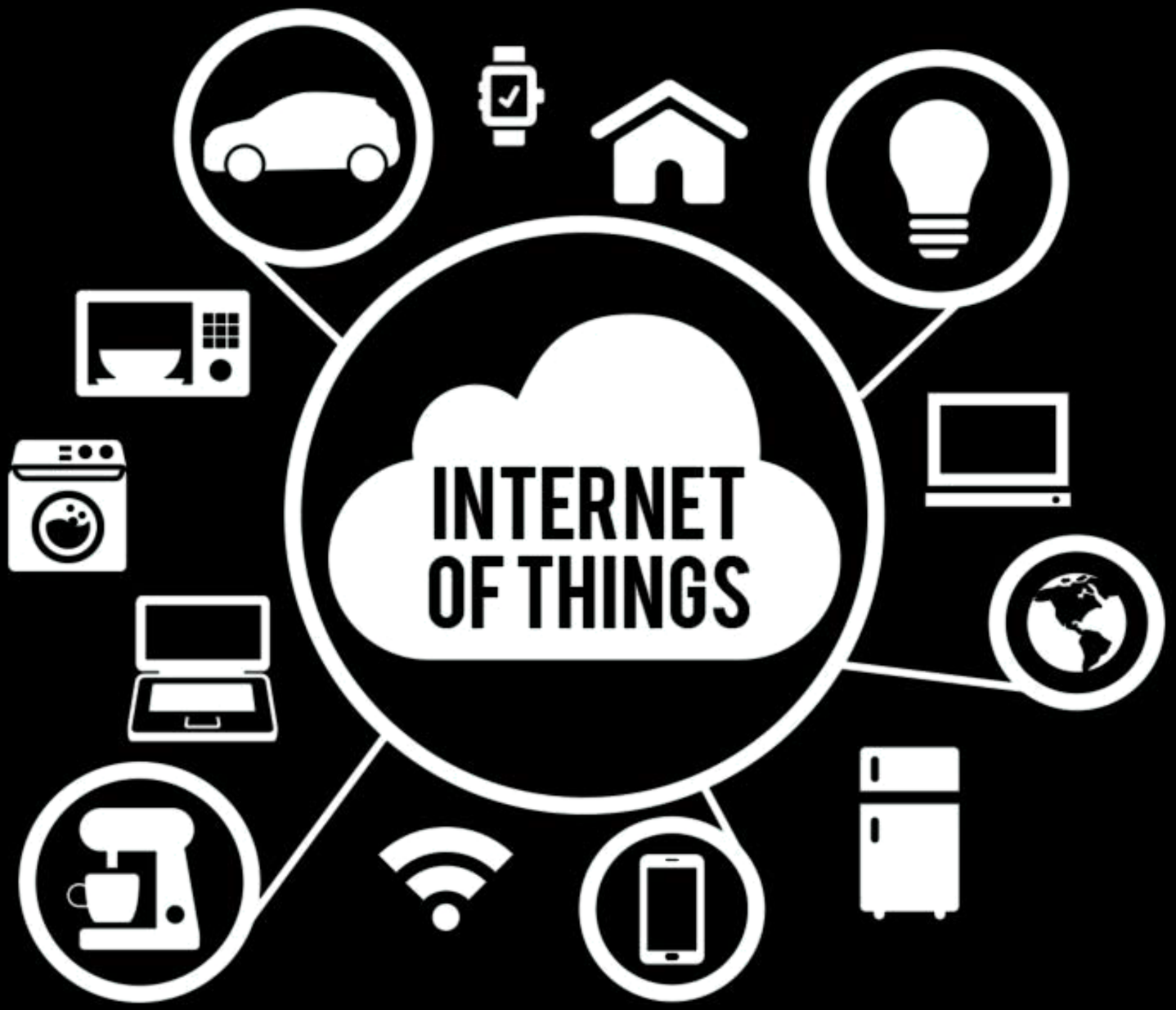
Novas Integrações Aproveitando o Mesmo Código



led.on()

led.on()

Controle de Liga/Desliga na Vida Real



Internet das Coisas (IoT)

Resumo da Opera

Funcionalidade	Comandos

Timer
acessar documentação
	`from threading import Timer`
	`timer = Timer(3.5, funcao_para_quando_terminar)`
	`timer.start()`
	`timer.cancel()`

Sensor de Luz
acessar documentação
	`from gpiozero import LightSensor`
	`sensor = LightSensor(8)`
	`intensidade_de_luz = sensor.value`
	`sensor.threshold = 0.6 • sensor.light_detected`
	`sensor.when_dark = funcao • sensor.when_light = funcao`

Sensor de Movimento
acessar documentação
	`from gpiozero import MotionSensor`
	`sensor = MotionSensor(27)`
	`tem_movimento = sensor.motion_detected`
	`sensor.when_motion = funcao • sensor.when_no_motion = funcao`

IFTTT
acessar documentação
	`url = "https://maker.ifttt.com/trigger/evento/with/key/CHAVE"`
	`dados = {"value1": 2, "value2": -9.3, "value3": "olá"}`
	`resultado = post(url, json=dados)`

Resumo para o Projeto 05

Funcionalidade	Comandos
FSWebcam acessar documentação	`from os import system` `system("fswebcam --resolution 640x480 --skip 10 foto.jpg")`

ARecord
acessar documentação
	`from os import system • from subprocess import Popen`
	`system("arecord --duration 3 --format cd audio.wav")`
	`comando = ["arecord", "--duration", "30", "audio.wav"]`
	`aplicativo = Popen(comando) • aplicativo.terminate()`

Lame acessar documentação	`from os import system` `system("lame audio.wav audio.mp3")`
OpusEnc acessar documentação	`from os import system` `system("opusenc audio.wav audio.ogg")`

Telegram
acessar documentação
	`from requests import post, get`
	`base = "https://api.telegram.org/bot" + chave`
	`endereco = base + "/sendMessage"`
	`dados = {"chat_id": id_da_conversa, "text": "Olá!"}`
	`resposta = post(endereco, json=dados)`
	`print(resposta.text) • dicionario = resposta.json()`
	`endereco = base + "/sendPhoto" • endereco = base + "/sendVoice"`
	`arquivo = {"photo": open("foto.jpg", "rb")}`
	`resposta = post(endereco, data=dados, files=arquivo)`
	`dados = {"offset": proximo_id_de_update}`
	`resposta = get(base + "/getUpdates", json=dados)`
	`from urllib.request import urlretrieve`
	`urlretrieve(endereco_do_arquivo, nome_do_arquivo_baixado)`

Funcionalidade	Comandos

Datas e Horários
acessar documentação
	`from datetime import datetime, timedelta`
	`tempo = datetime(2018, 3, 28, 15, 35, 12) • datetime.now()`
	`intervalo = timedelta(months=4) • tempo2 = tempo + intervalo`
	`tempo2 > tempo • intervalo.seconds • tempo.strftime("%H:%M")`

Campainha
acessar documentação
	`from gpiozero import Buzzer • buzzer = Buzzer(16)`
	`buzzer.on() • buzzer.off() • buzzer.toggle()`
	`buzzer.is_active • buzzer.beep()`
	`buzzer.beep(n=4, on_time=0.5, off_time=2)`

Sensor de Distância
acessar documentação
	`from gpiozero import DistanceSensor`
	`sensor = DistanceSensor(trigger=17, echo=18)`
	`sensor.distance • sensor.threshold_distance`
	`s.when_in_range = funcao • s.when_out_of_range = funcao`

MongoDB
acessar documentação
	`from pymongo import MongoClient, ASCENDING, DESCENDING`
	`cliente = MongoClient("localhost", 27017)`
	`banco = cliente["nome"] • colecao = banco["nome"]`
	`dados = {"nome": "Jan K. S.", "idade": 32}`
	`colecao.insert(dados) • colecao.insert([dados1, dados2])`
	`busca = {"chave1": valor1, "chave2": {"$gt": valor2}}`
	`ordenacao = [ ["idade", DESCENDING] ]`
	`documento = colecao.find_one(busca, sort=ordenacao)`
	`documentos = list( colecao.find(busca, sort=ordenacao) )`

Resumo para o Projeto 03

Funcionalidade	Comandos

Emissor Infravermelho	`from py_irsend.irsend import *`
	`controles = list_remotes() • codigos = list_codes("mini")`
	`send_once("mini", ["KEY_1", "KEY_2"])`

Receptor Infravermelho	`from lirc import init, nextcode`
	`init("aula", blocking=False)`
	`codigo = nextcode() • codigo == ["KEY_1"]`

Servidor Flask
acessar documentação
	`from flask import Flask`
	`app = Flask(__name__)`
	
	`@app.route("/")`
	`def mostrar_inicio():`
	`  return "Bem-vindo!"`
	
	`@app.route("/contato")`
	`def mostrar_contato():`
	`  return "janks@puc-rio.br"`
	
	`@app.route("/numero/<int:x>")`
	`def mostrar_numero(x):`
	`  return "x = " + str(x)`
	
	`return redirect("/outrapagina")`
	
	`return render_template("index.html")`
	
	`app.run(port=5000, debug=False)`

HTML	`<p>Parágrafo</p>`
	`<img src="site.com/imagem">`
	`<a href="/pagina">Link</a>`
	
	`<ul>`
	`  <li>Item 1 da Lista</li>`
	`  <li>Item 2 da Lista</li>`
	`</ul>`

Ngrok	abrir Terminal → `ngrok http 5000`

Resumo para o Projeto 02

Funcionalidade	Comandos

LED
acessar documentação
	`from gpiozero import LED • led = LED(21)`
	`led.on() • led.off() • led.toggle() • led.is_lit`
	`led.blink() • led.blink(n=4, on_time=0.5, off_time=2)`

Botão
acessar documentação
	`from gpiozero import Button • botao = Button(11)`
	`botao.is_pressed • botao.wait_for_press()`
	`botao.when_pressed = funcao`
	`botao.when_released = funcao`
	`botao.when_held = funcao`

LCD
acessar documentação
	`from Adafruit_CharLCD import Adafruit_CharLCD`
	`lcd = Adafruit_CharLCD(2, 3, 4, 5, 6, 7, 16, 2)`
	`lcd.message("Texto 1\nTexto 2")`
	`lcd.clear()`

MPlayer	`from mplayer import Player • player = Player()`
	`player.loadfile("Musica.mp3") • player.loadlist("lista.txt")`
	`player.pause() • player.paused • player.quit()`
	`player.time_pos = 2 • player.length • player.pt_step(-1)`
	`player.metadata["Title"] • player.metadata["Artist"]`
	`player.volume = 70 • player.speed = 2`

Resumo para o Projeto 01

Funcionalidade	Comandos
Funções	`x = input("Digite um número: ") • print("Resultado: ", x)` `from time import sleep • sleep(0.5)`

Listas
acessar documentação
	`lista = [1, 2, 3] • lista2 = ["texto", [0, 0], 5]`
	`lista[0] • lista[1:3] • total_de_elementos = len(lista)`
	`lista.append(novo_elemento) • lista.remove(indice)`

Dicionários acessar documentação	`dicionario = {"chave 1": 42, "chave 2": [1, 2, 3]}` `dicionario["chave 1"] • dicionario["chave 3"] = "Olá!"`

Textos (Strings)
acessar documentação
	`texto = "olá!" • texto[0] • texto[1:4] • len(texto)`
	`texto + "\n" • texto + str(numero) • "x = %.2f" % numero`
	`2 + int("11") • 4 / float("23.5")`

Condicionais	`if x == 0 and z < 4:`
	`  y = 4`
	
	`if x not in [1, 2]:`
	`  y = 4`
	`else:`
	`  y = 0`
	
	`if x != 0:`
	`  y = 4`
	`elif x >= 0:`
	`  y = 3`
	`else:`
	`  y = 0`

Repetições	`for elem in lista:`
	`  ...`
	
	`for i in range(1, 4):`
	`  ...`
	
	`while x > 1:`
	`  ...`

Criação de Funções	`def funcao1(x):`
	`  return x + 2`
	
	`def funcao2(x, y, z):`
	`  ...`
	
	`x = 2`
	`def funcao3():`
	`  global x`