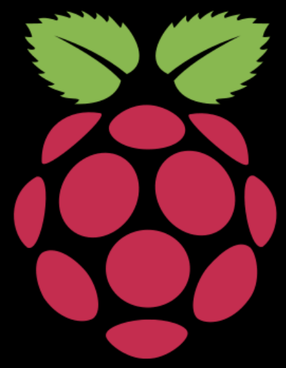


Projeto 05

Controle Automático

Jan K. S. – janks@puc-rio.br

ENG1419 – Programação de Microcontroladores

Revisão do Projeto 04

```
aula@raspberrypi ~ $ fswebcam foto1.jpeg
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
Adjusting resolution from 384x288 to 352x288.
--- Capturing frame...
Captured frame in 0.00 seconds.
--- Processing captured image...
Writing JPEG image to 'foto1.jpeg'.
```

```
aula@raspberrypi ~ $ fswebcam --resolution 640x480 foto2.jpeg
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
--- Capturing frame...
Captured frame in 0.00 seconds.
--- Processing captured image...
Writing JPEG image to 'foto2.jpeg'.
```

Comandos no Terminal para Captura de Fotos

As imagens são geradas na mesma pasta
do arquivo com o código Python.

```
>>> from os import system
>>> comando = "fswebcam foto1.jpeg"
>>> system(comando)
>>> system("fswebcam --resolution 640x480 foto2.jpeg")
```

Chamada de Comandos para Tirar uma Foto

audio1.wav
(baixa qualidade)

audio2.wav
(qualidade de CD)

```
>>> from os import system
>>> system("arecord --duration 3 audio1.wav")
```

```
>>> system("arecord --duration 3 --format cd audio2.wav")
```

Gravação de 3 Segundos de Áudio

```
>>> from subprocess import Popen
>>> aplicativo = None
>>> def iniciar_gravacao():
...     global aplicativo
...     comando = ["arecord", "--duration", "30", "audio.wav"]
...     aplicativo = Popen(comando)
...
```

executa arecord em plano de fundo

```
>>> def parar_gravacao():
...     if aplicativo != None:
...         aplicativo.terminate()
...
>>> iniciar_gravacao()
>>> sleep(4)
>>> parar_gravacao()
```

Interrupção da Gravação de Áudio

```
>>> from os import system
>>> system("lame audio.wav audio.mp3")
>>> system("opusenc audio.wav audio.ogg")
```

Conversão de WAV para MP3 e para OGG em Python

`api.telegram.org/bot<SUA_CHAVE_SECRETA>/<comando>`

+

`dados extras`

Comando	Dados Obrigatórios
`sendMessage`	id do chat, texto
`sendPhoto`	id do chat, arquivo
`sendVoice`	id do chat, arquivo
`getUpdates`	(nenhum)
`getFile`	id do arquivo

Exemplos de Comandos para Bots do Telegram

```
>>> chave = "COLOQUE A SUA CHAVE AQUI!"
>>> id_da_conversa = "COLOQUE O ID DA SUA CONVERSA AQUI!"
>>> endereco_base = "https://api.telegram.org/bot" + chave
```

Configuração Inicial para Usar o Bot

```
>>> from requests import post
>>> endereco = endereco_base + "/sendMessage"
>>> dados = {"chat_id": id_da_conversa, "text": "Oi!"}
>>> resposta = post(endereco, json=dados)
```

Envio de Mensagens de Texto

```
>>> from requests import post
>>> endereco = endereco_base + "/sendPhoto"
>>> dados = {"chat_id": id_da_conversa}
>>> arquivo = {"photo": open("foto.jpeg", "rb")}
>>> resposta = post(endereco, data=dados, files=arquivo)
```

Envio de Fotos

```
>>> from requests import post
>>> endereco = endereco_base + "/sendVoice"
>>> dados = {"chat_id": id_da_conversa}
>>> arquivo = {"voice": open("audio.ogg", "rb")}
>>> resposta = post(endereco, data=dados, files=arquivo)
```

Envio de Áudio

```
{
   "ok":true,
   "result":[
      {
         "update_id":564714813,
         "message":{
            "voice":{
               "file_id":"AwADAQADJAADQzTwRQGVIkswNsWHAg",
               ...
            },
            ...
         }
      },
      {
         "update_id":564714814,
         "message":{
            "photo":[
               {
                  "file_id":"AgADAQAD1qcxG0M08EXPZxFkkC7VB-drDDAABHNCNifaQdcHHhEBAAEC",
                  "file_size":1438,
                  ...
               },
               {
                  "file_id":"AgADAQAD1qcxG0M08EXPZxFkkC7VB-drDDAABAuIfKssEffgIBEBAAEC",
                  "file_size":117149,
                  ...
               },
               ...
            ]
         }
      },
      {
         "update_id":564714815,
         "message":{
            "text":"Texto!",
            ...
         }
      }
   ]
}
```


Dicionário (JSON) com as Atualizações de Mensagens

```
>>> from requests import get
>>> endereco = endereco_base + "/getUpdates"
>>> resposta = get(endereco)
>>> dicionario_da_resposta = resposta.json()
>>> for resultado in dicionario_da_resposta["result"]:
...     mensagem = resultado["message"]
...     if "text" in mensagem:
...         texto = mensagem["text"]
...     elif "voice" in mensagem:
...         id_do_arquivo = mensagem["voice"]["file_id"]
...     elif "photo" in mensagem:
...         foto_mais_resolucao = mensagem["photo"][-1]
...         id_do_arquivo = foto_mais_resolucao["file_id"]
```

Verificação dos Tipos de Mensagens

```
>>> endereco = endereco_base + "/getFile"
>>> dados = {"file_id": id_do_arquivo}
>>> resultado = get(endereco, json=dados)
>>> dicionario = resultado.json()
>>> caminho_do_arquivo = dicionario["result"]["file_path"]
```

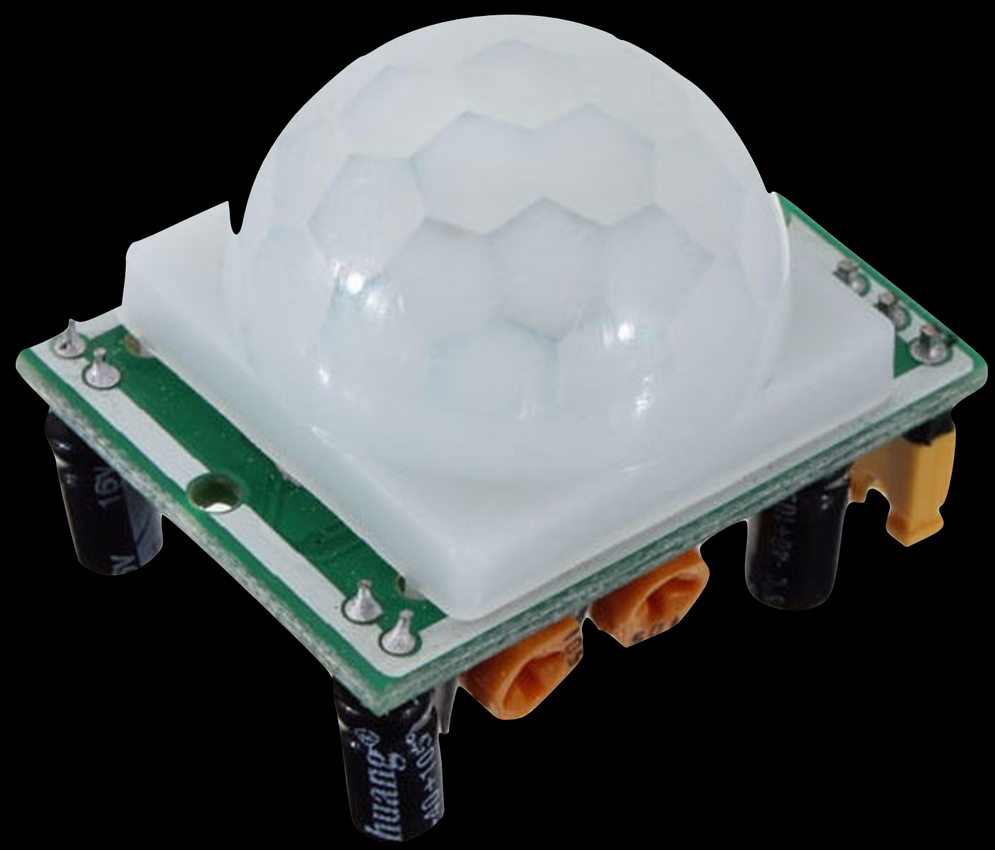
```
>>> from urllib.request import urlretrieve
>>> endereco_do_arquivo = "https://api.telegram.org/file/bot"
+ chave + "/" + caminho_do_arquivo
>>> arquivo_de_destino = "meu_arquivo.ogg"
>>> urlretrieve(endereco_do_arquivo, arquivo_de_destino)
```

Download de um Arquivo de Chat

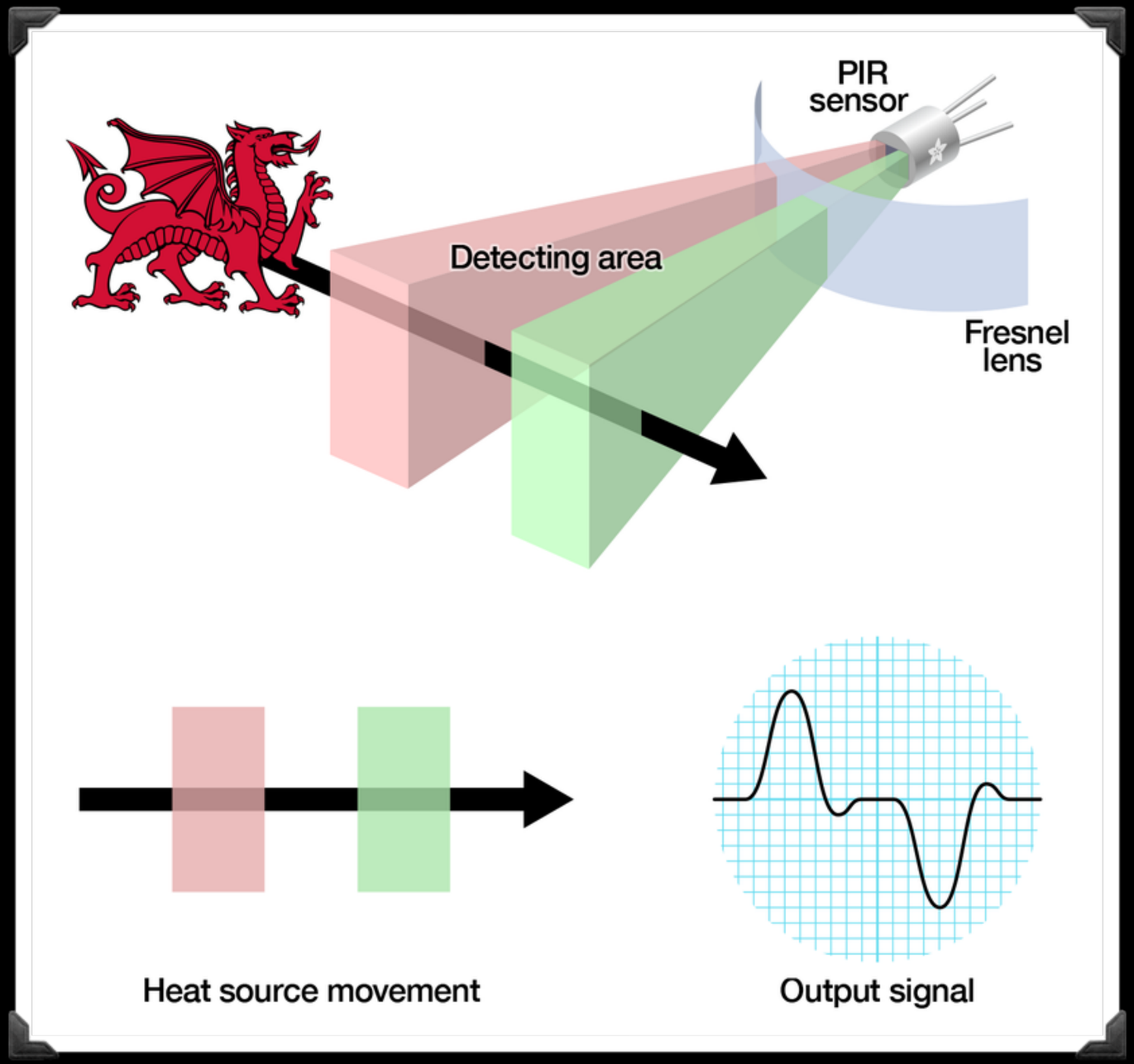
```
>>> proximo_id_de_update = 0
>>> while True:
...     endereco = endereco_base + "/getUpdates"
...     dados = {"offset": proximo_id_de_update}
...     resposta = get(endereco, json=dados)
...     dicionario_da_resposta = resposta.json()
...     for resultado in dicionario_da_resposta["result"]:
...         ...
...         proximo_id_de_update = int(resultado["update_id"]) + 1
```

Busca Contínua de Atualizações de Mensagens

Hardware



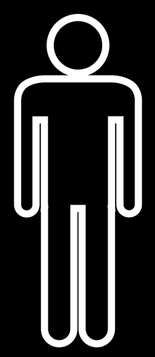
Sensor de Movimento

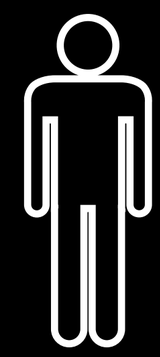


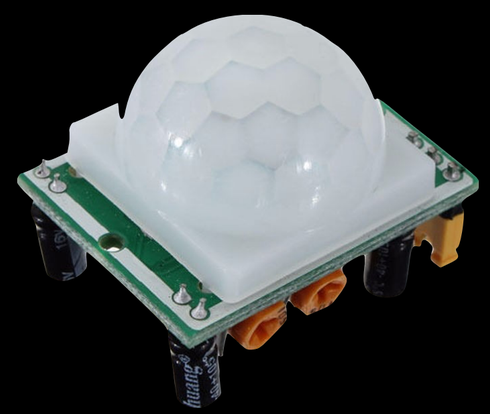
Funcionamento de um Infra-Vermelho Passivo (PIR)

```
>>> from gpiozero import MotionSensor
>>> sensor_de_movimento = MotionSensor(27)
>>> sensor_de_movimento.wait_for_motion()
>>> sensor_de_movimento.wait_for_no_motion()
>>> def movimento_detectado():
...     print("Haja luz!")
...
>>> sensor_de_movimento.when_motion = movimento_detectado
>>> def inercia_detectada():
...     print("Stop! Hammer time!")
...
>>> sensor_de_movimento.when_no_motion = inercia_detectada
```

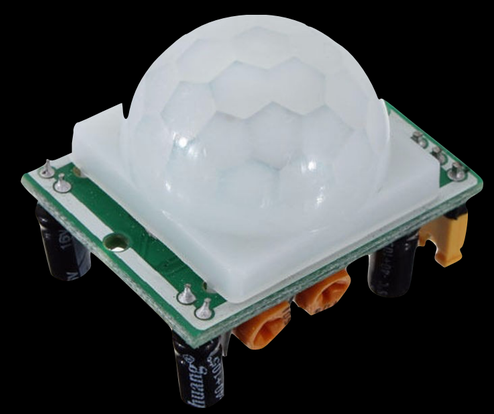
Eventos do Detector de Movimento







≈ 2s



movimento!

ainda em
movimento!

inércia!

Sensor de Movimento

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

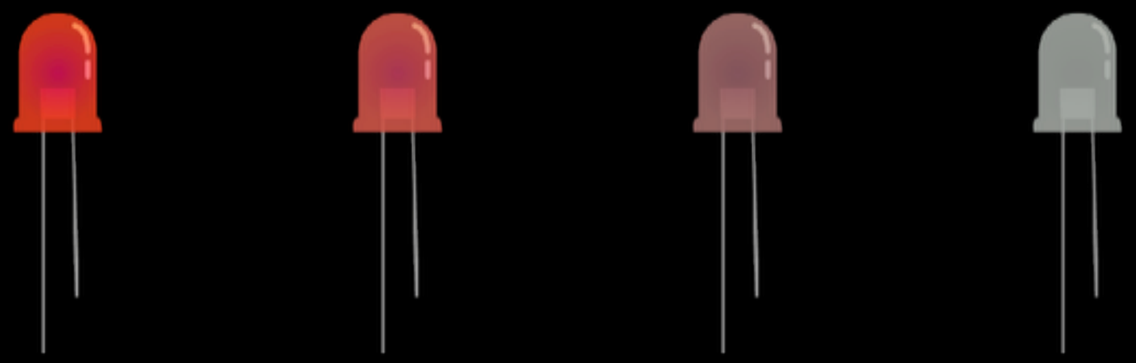
3 segundos e meio depois...

```
Acaboouuoo, acabou!
```

Timer em Python

```
>>> from threading import Timer
>>> timer = None
>>> def timer_recorrente():
...     print("Hey, listen!")
...     global timer
...     timer = Timer(1.0, timer_recorrente)
...     timer.start()
...
>>> timer_recorrente()
>>> def parar_timer():
...     timer.cancel()
...
```

Timer Recorrente e Cancelamento de Timer

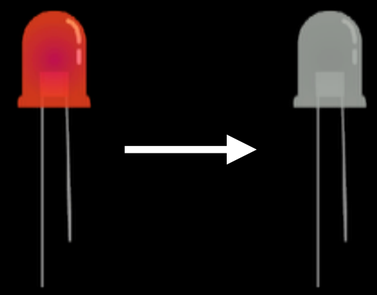


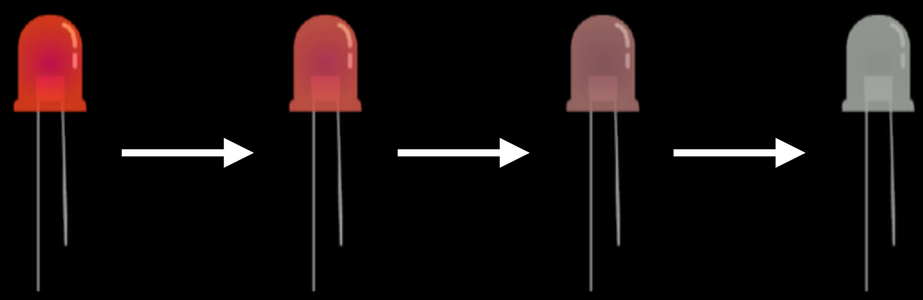
Níveis de Brilho de um LED

```
>>> from gpiozero import PWMLED
>>> ledPWM = PWMLED(21)
>>> ledPWM.on()
>>> ledPWM.value
1.0
>>> ledPWM.off()
>>> ledPWM.value
0.0
>>> ledPWM.value = 0.5
```

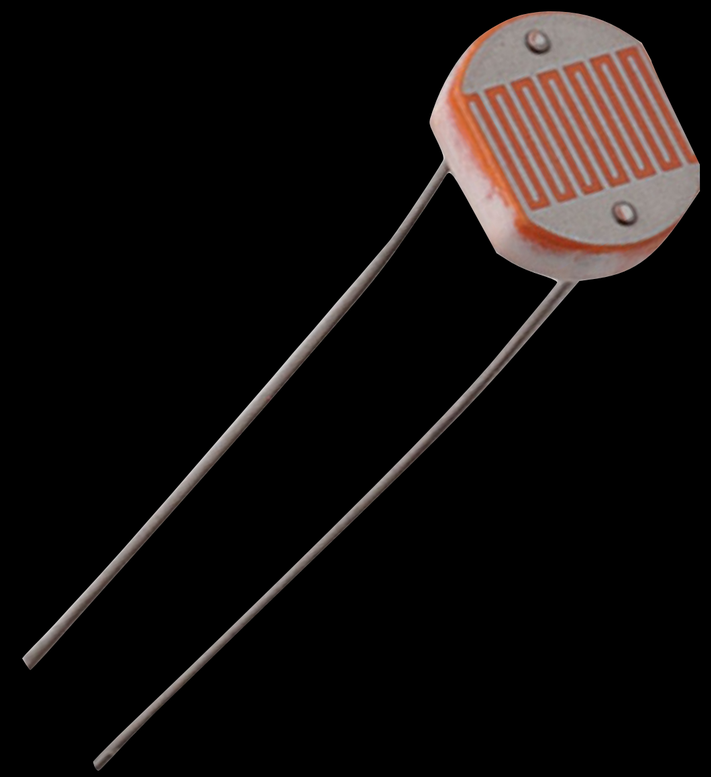
Valor de um LED PWM

```
>>> from gpiozero import PWMLED
>>> ledPWM = PWMLED(21)
>>> ledPWM.blink()
>>> ledPWM.blink(n=3, on_time=1, off_time=0.5)
>>> ledPWM.pulse()
>>> ledPWM.pulse(n=3, fade_in_time=2, fade_out_time=1)
```

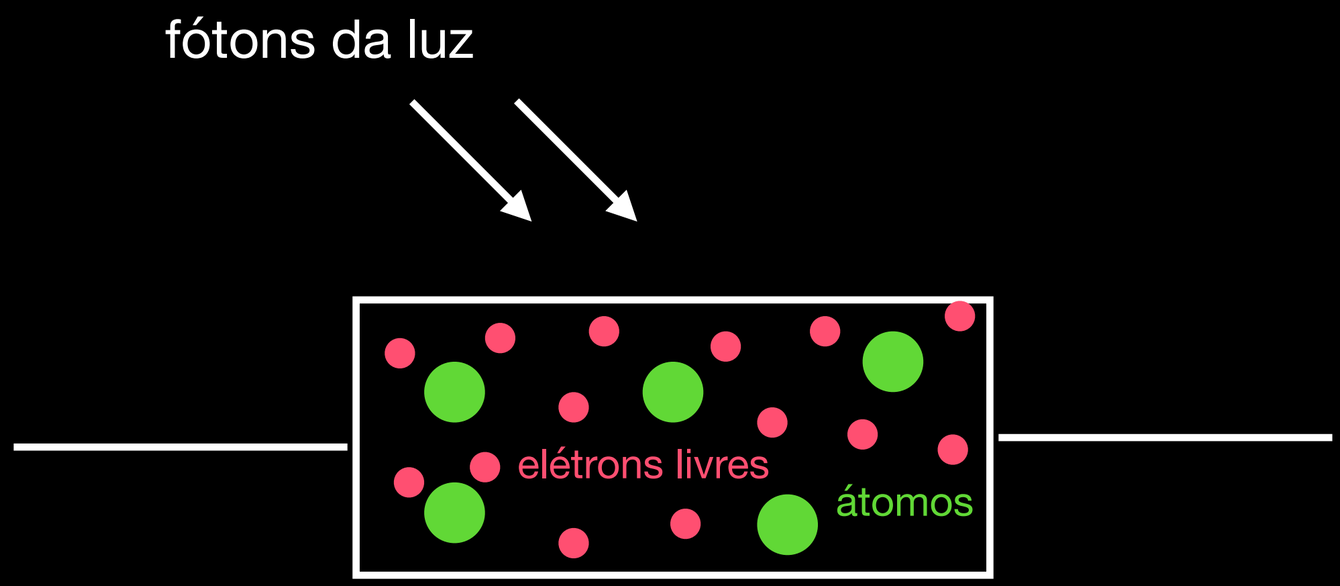


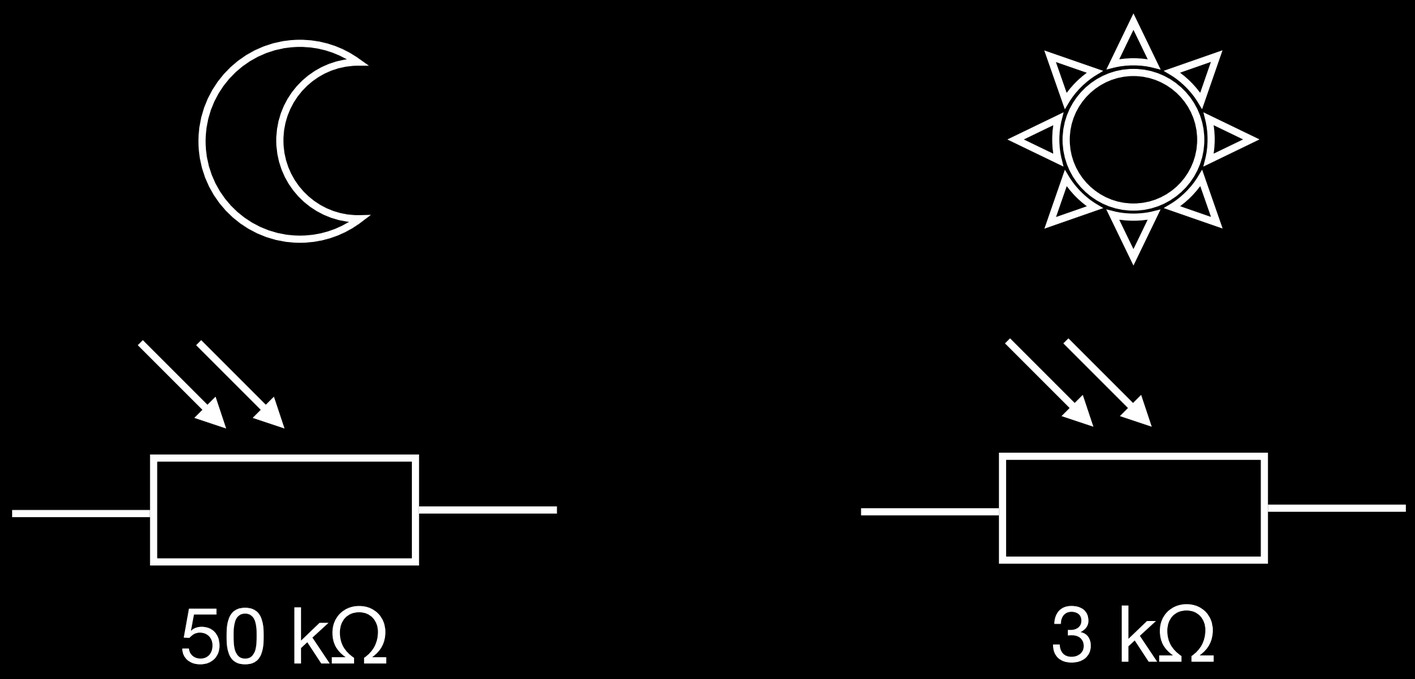


Piscamento vs Pulsação



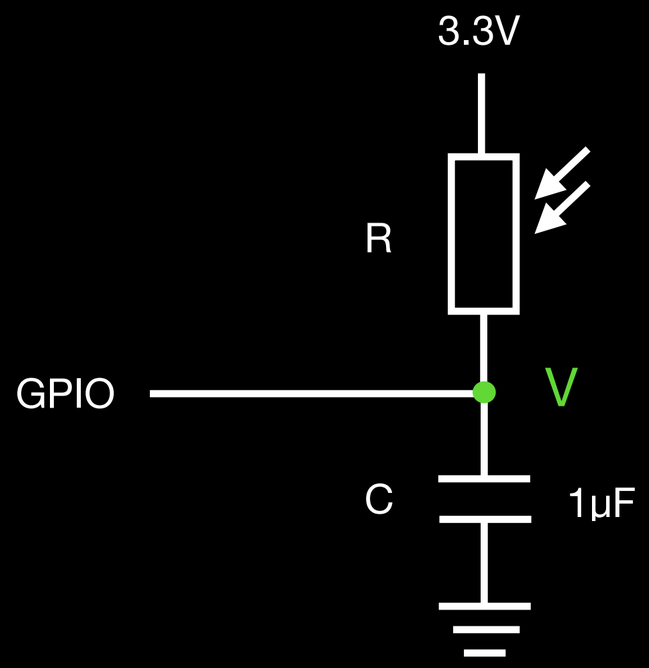
Sensor de Luz





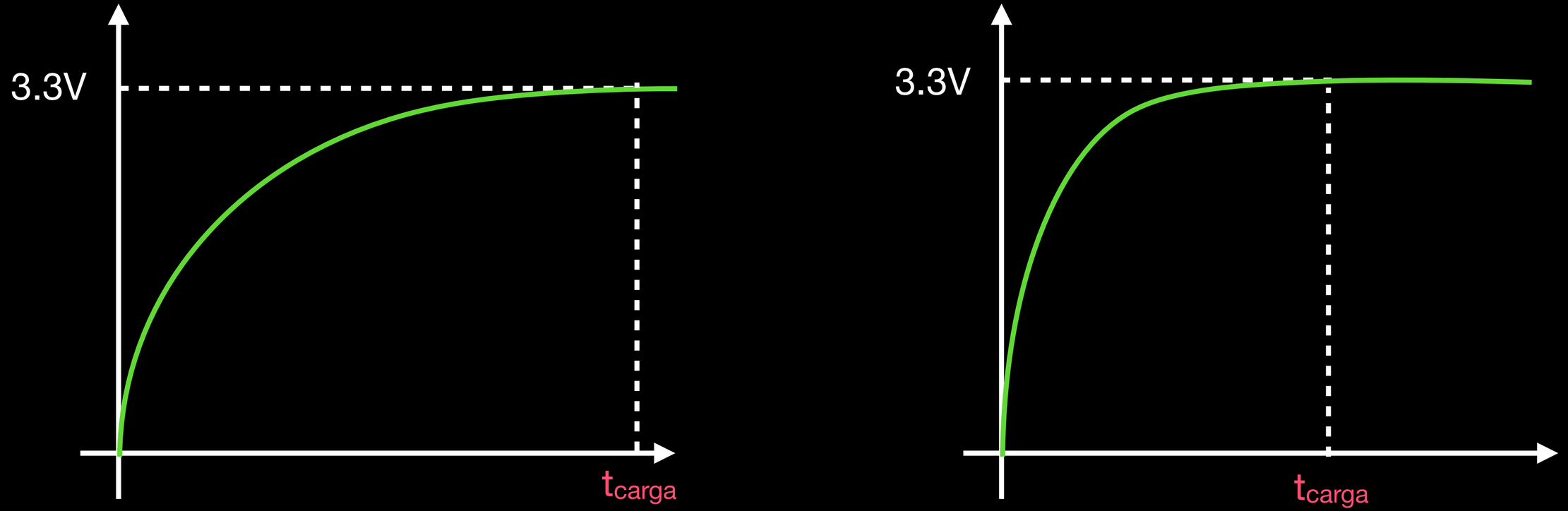
Resistência Dependente de Luz (LDR)

Circuito com Capacitor para Detecção de Luz



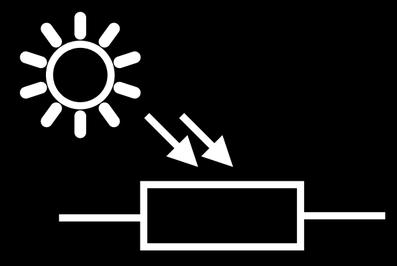
Carregamento sem Luz (R Maior)

Carregamento com Luz (R Menor)

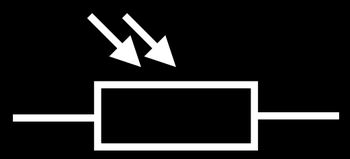


Tempo de Carregamento

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



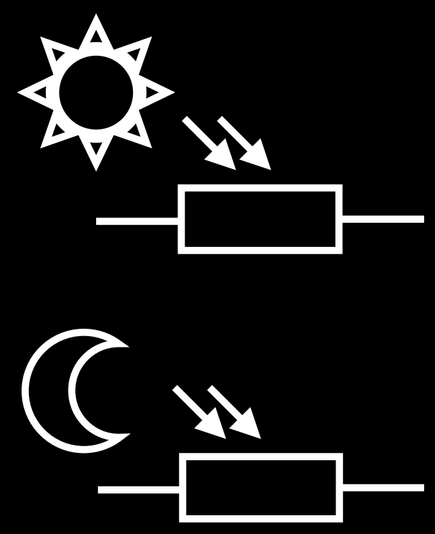
```
>>> sensor_de_luz.value
0.347932
```



Valor de um Sensor de Luz

```
>>> from gpiozero import LightSensor
>>> from time import sleep
>>> sensor_de_luz = LightSensor(8)
>>> for i in range(0, 3):
...     print(sensor_de_luz.value)
...     sleep(1)
...
0.854363
0.854363
0.347932
0.347932
```

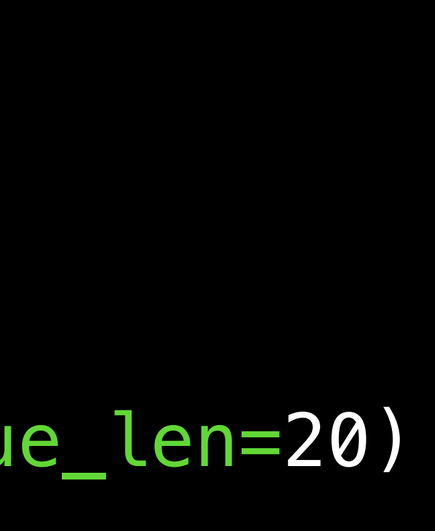
alteração imediata do valor



Captação Gradual de um Sensor de Luz

```
>>> from gpiozero import LightSensor
>>> from time import sleep
>>> sensor_de_luz = LightSensor(8, queue_len=20)
>>> for i in range(0, 3):
...     print(sensor_de_luz.value)
...     sleep(1)
...
0.854363
0.653456
0.498232
0.345463
```

alteração gradual do valor



Captação Gradual de um Sensor de Luz

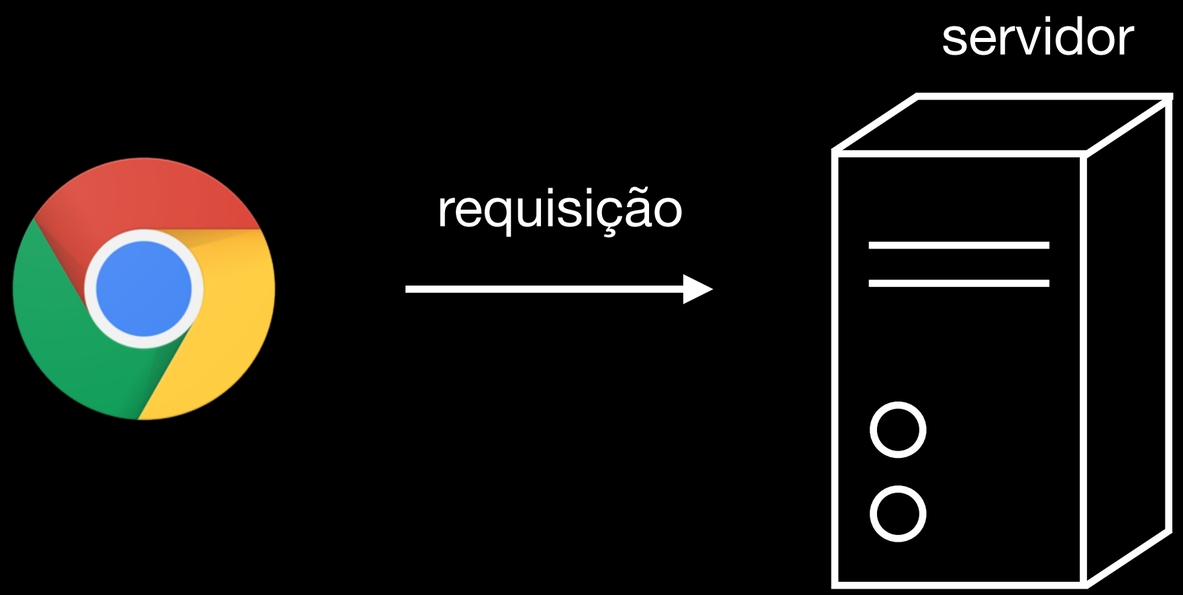
```
>>> from gpiozero import LightSensor
>>> sensor_de_luz = LightSensor(8)
>>> sensor_de_luz.value
0.56743
>>> sensor_de_luz.threshold
0.5
>>> sensor_de_luz.light_detected
True
>>> sensor_de_luz.threshold = 0.6
>>> sensor_de_luz.light_detected
False
```

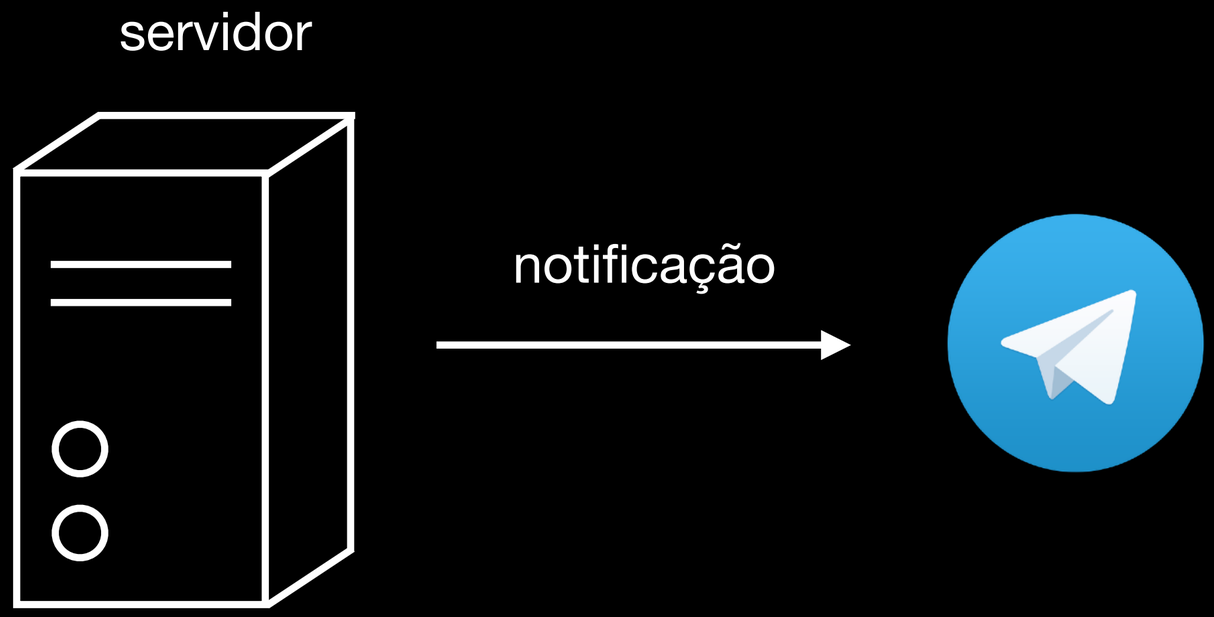
Limiar para Detecção de Luz

```
>>> from gpiozero import LightSensor
>>> sensor_de_luz = LightSensor(8)
>>> sensor_de_luz.wait_for_dark()
>>> sensor_de_luz.wait_for_light()
>>>
>>> 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
```

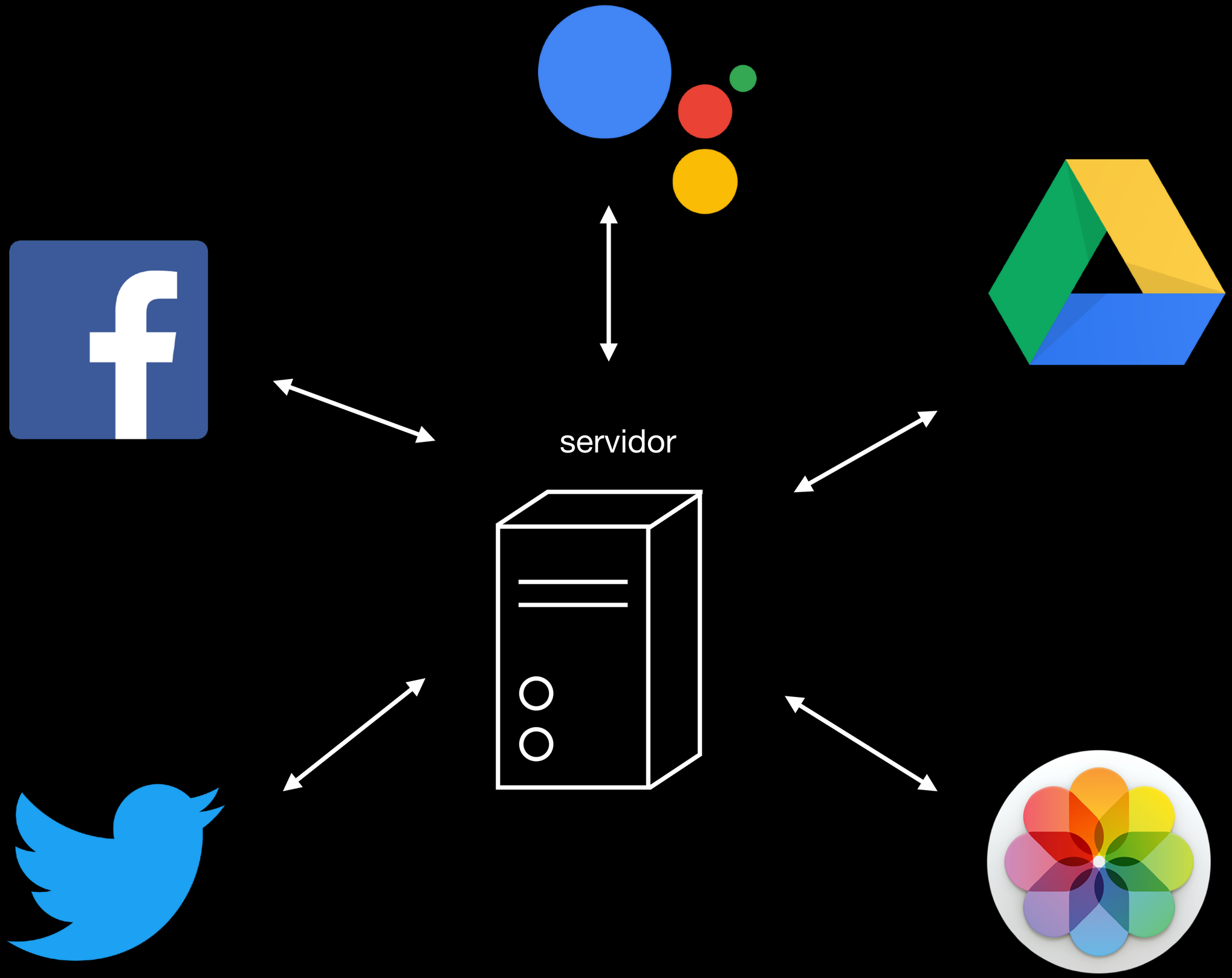
Eventos de um Sensor de Luz

Software





Requisições e Notificações (Vistas em Projetos Anteriores)



Dificuldade em Programar Várias Comunicações

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.

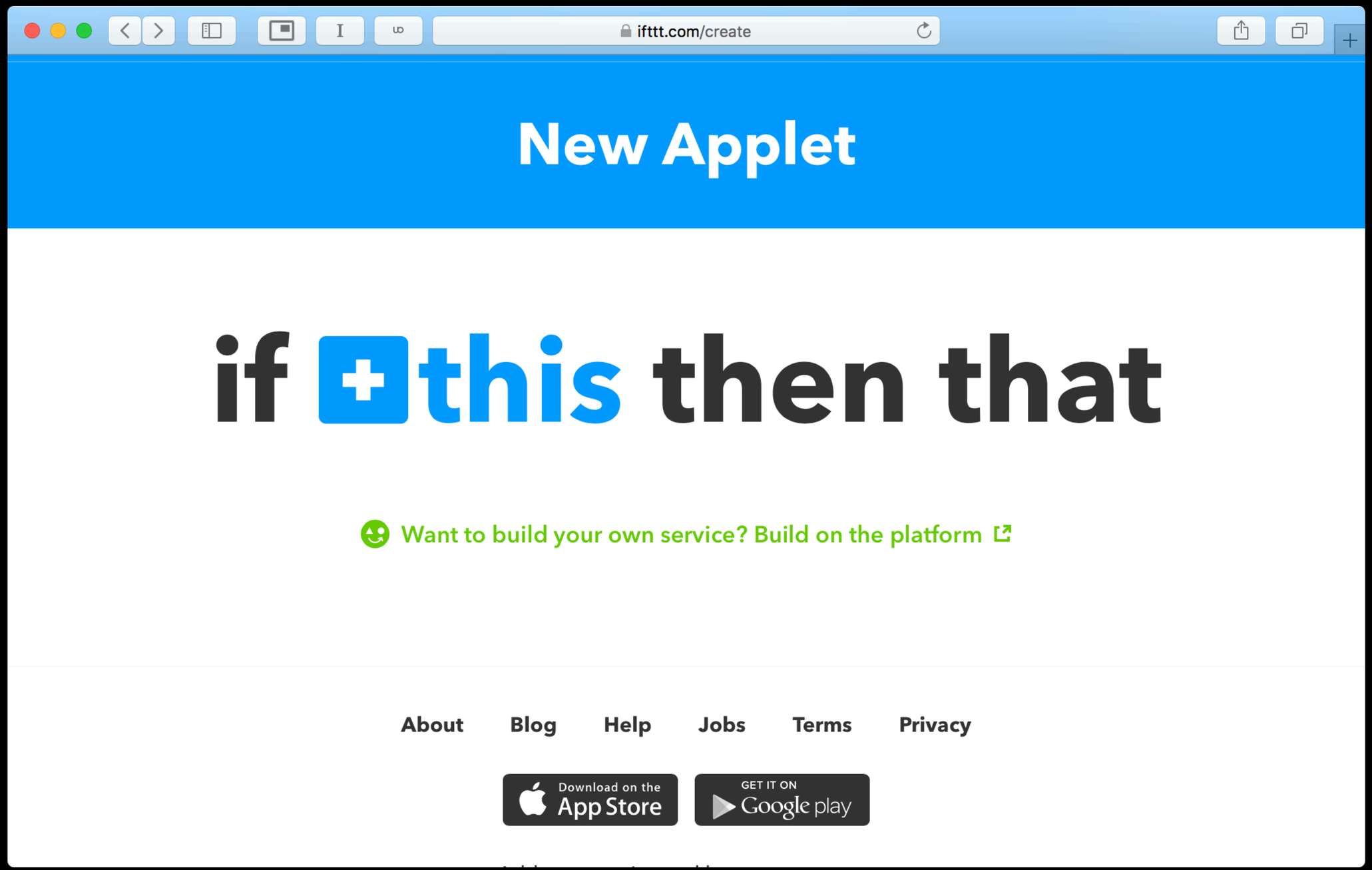
Enter your email

Get started

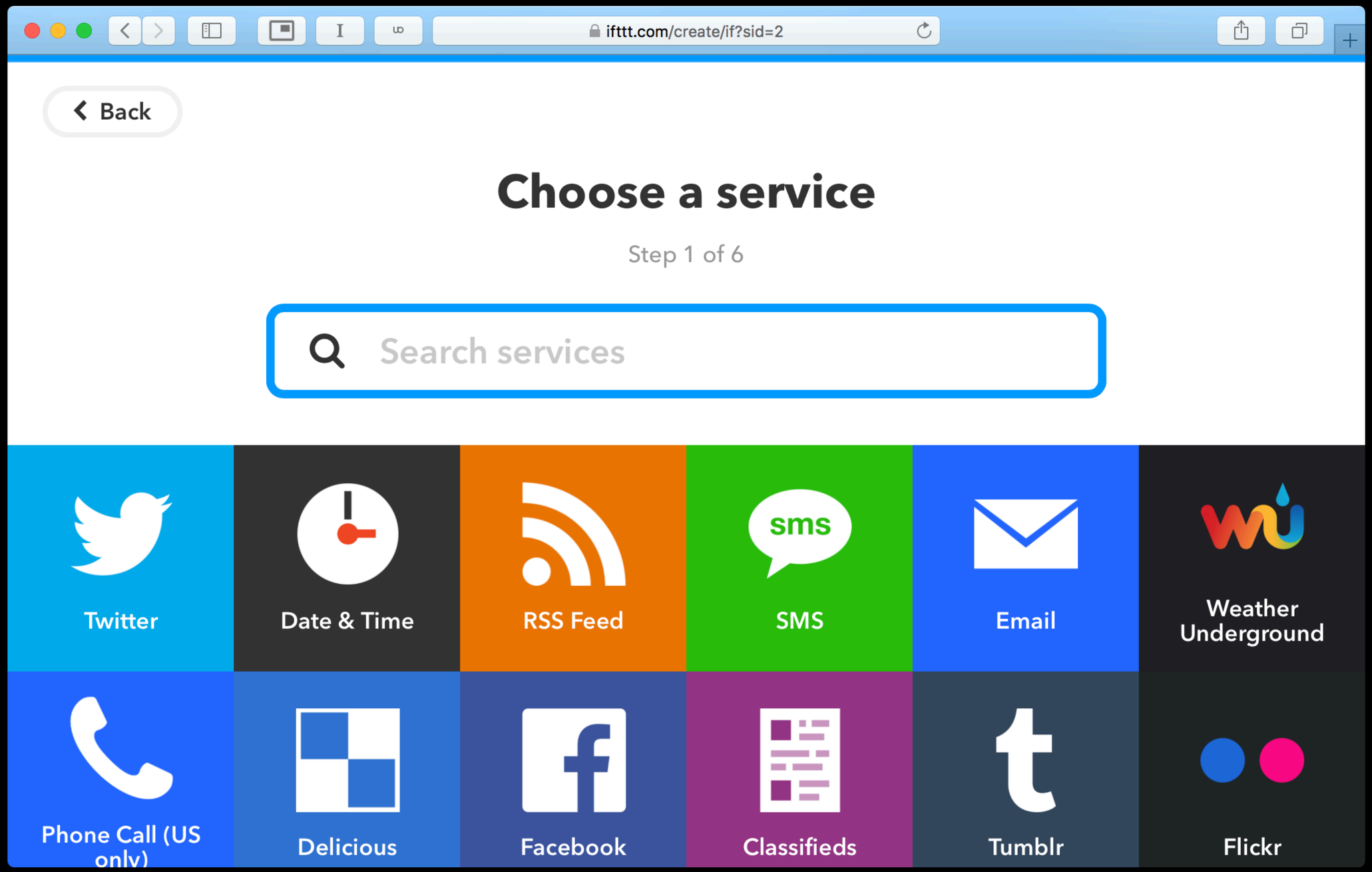
Continue with Google

Continue with Facebook

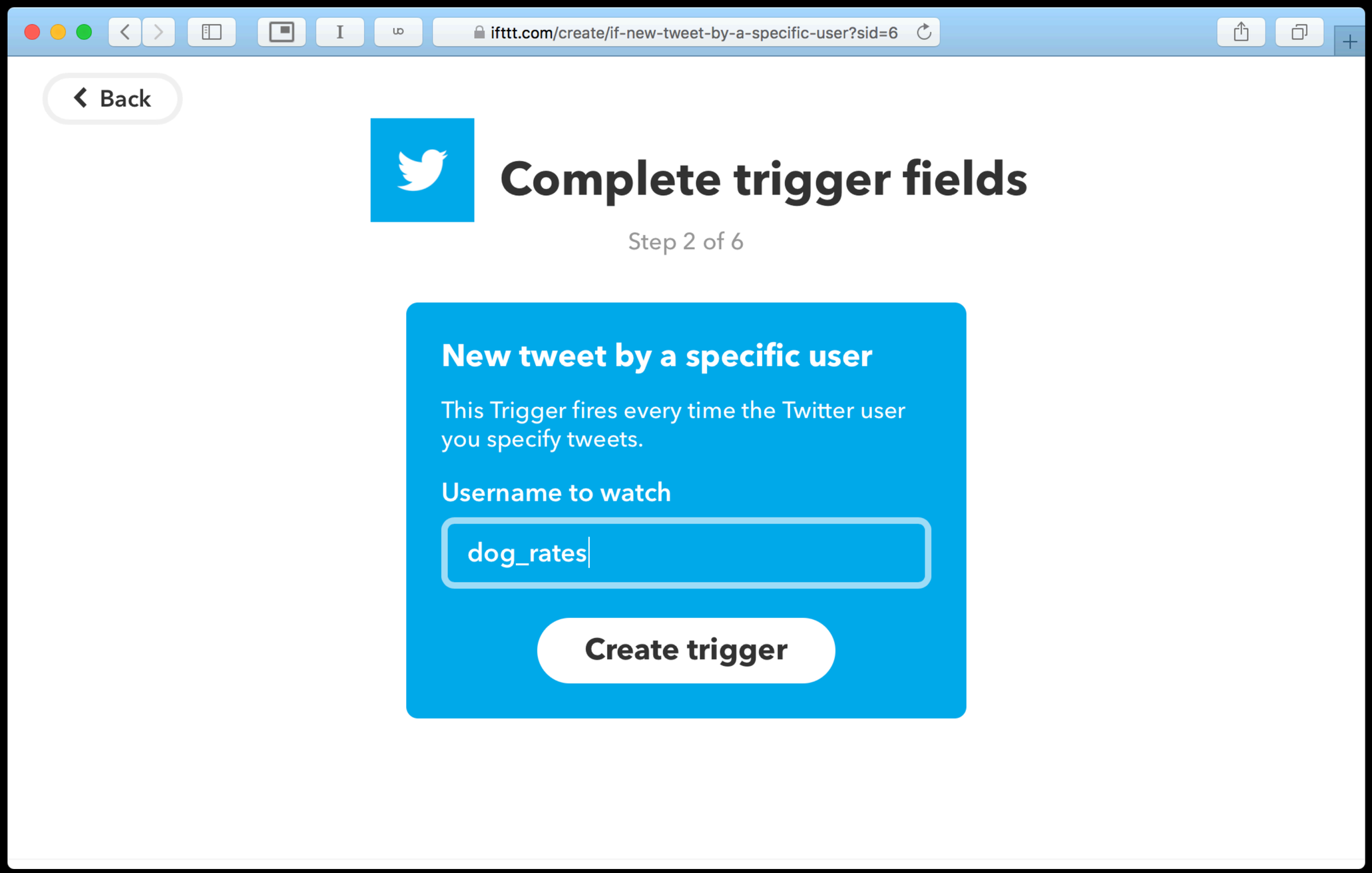
Integração entre Serviços com IFTTT



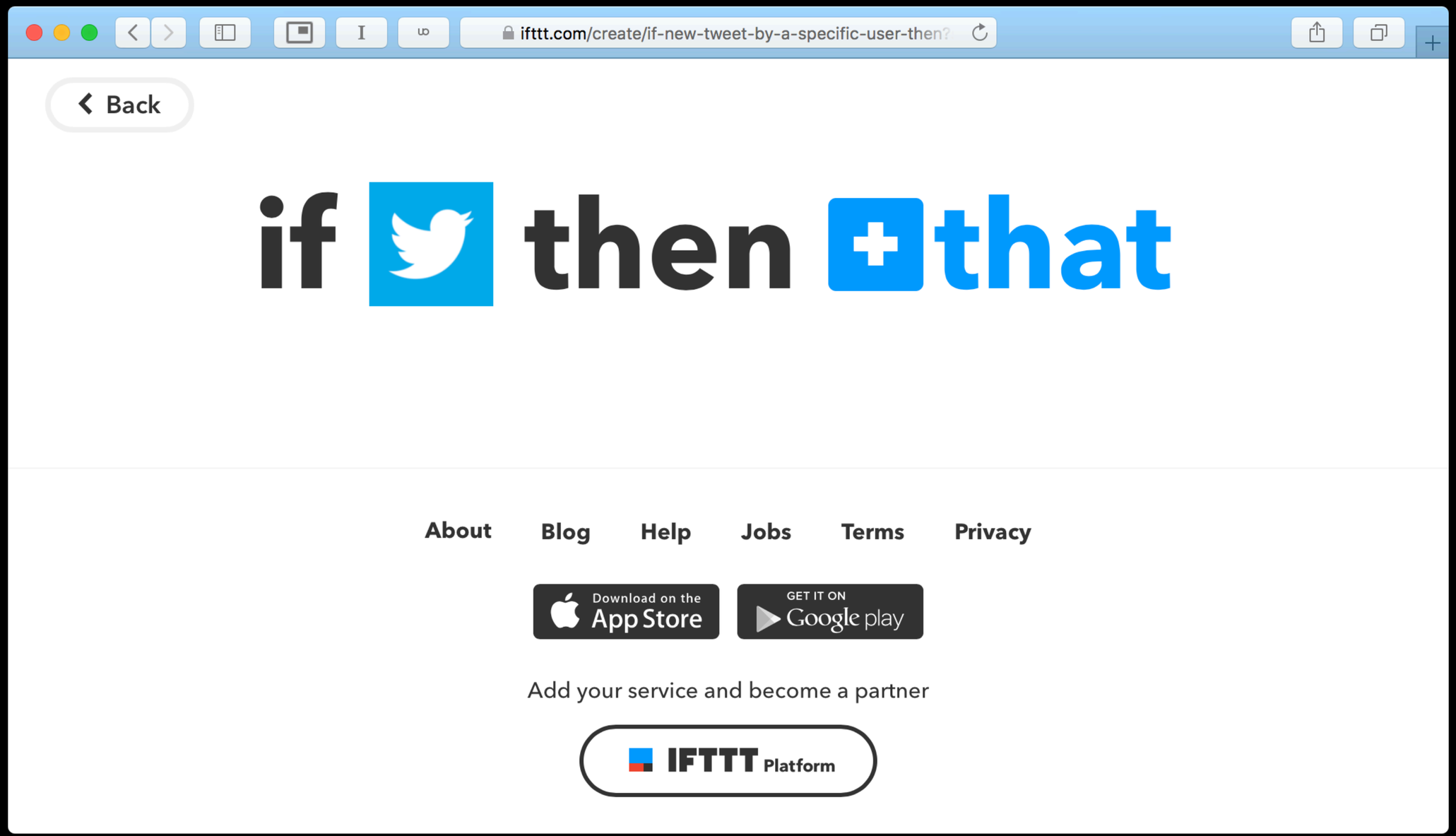
Escolha do Serviço "This"



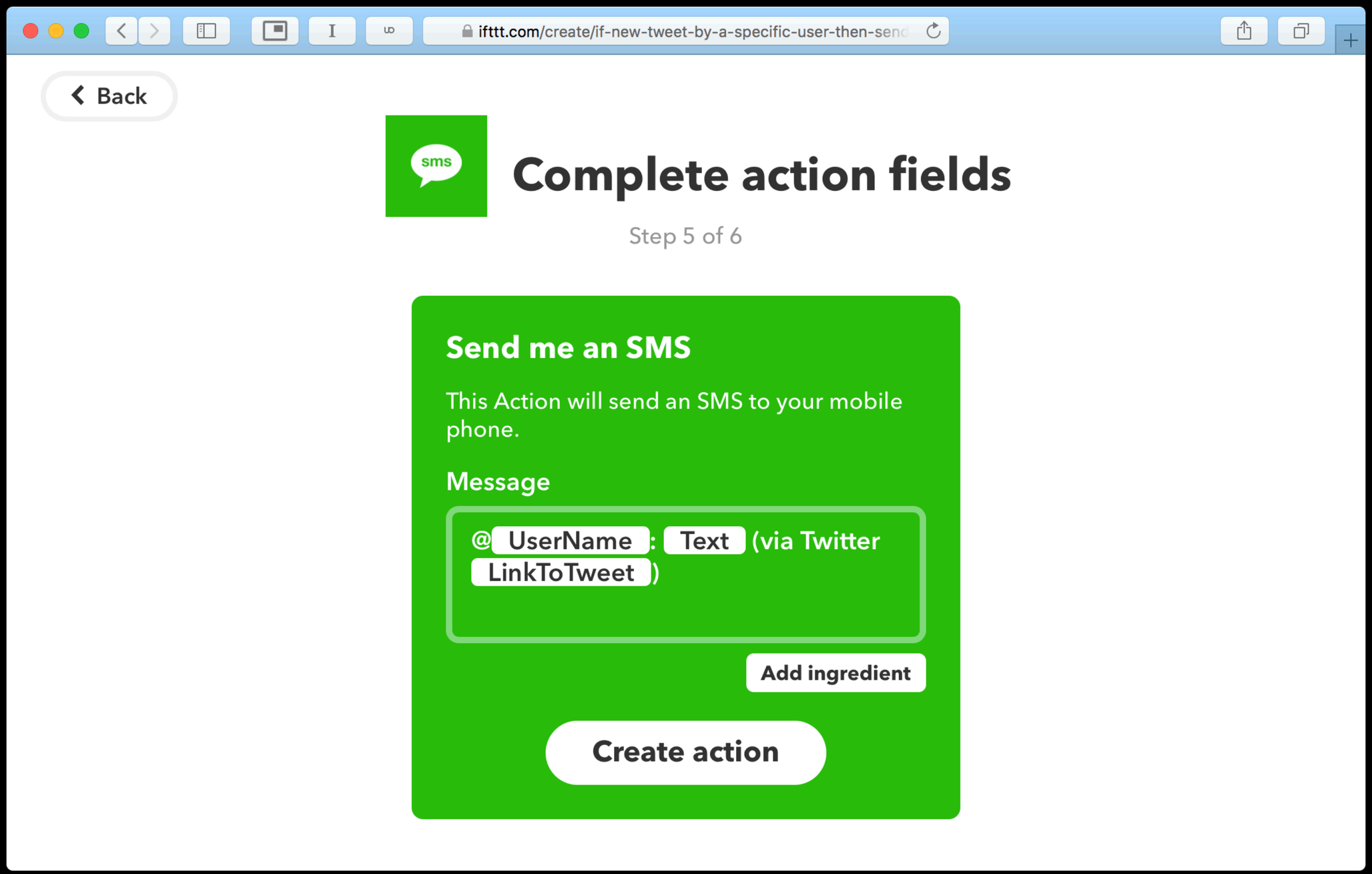
Lista de Serviços Compatíveis



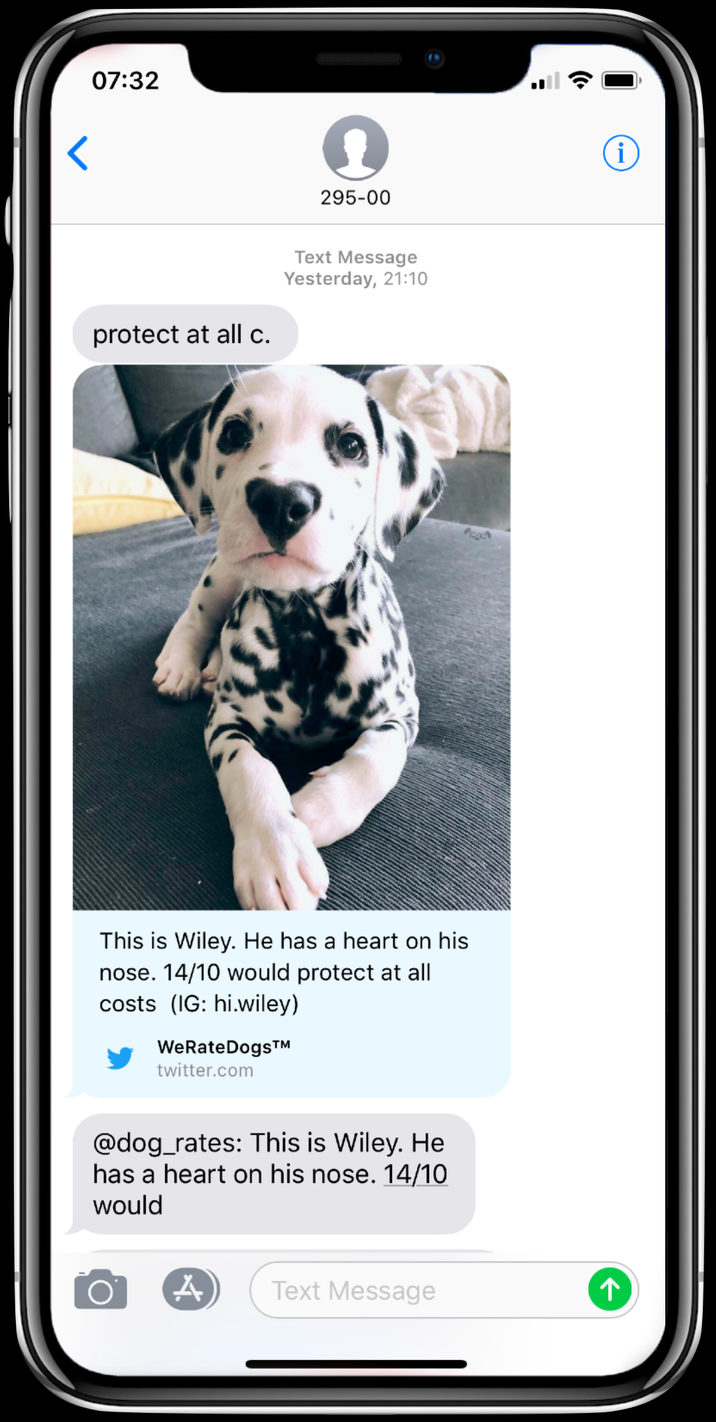
Configuração do Serviço "This"



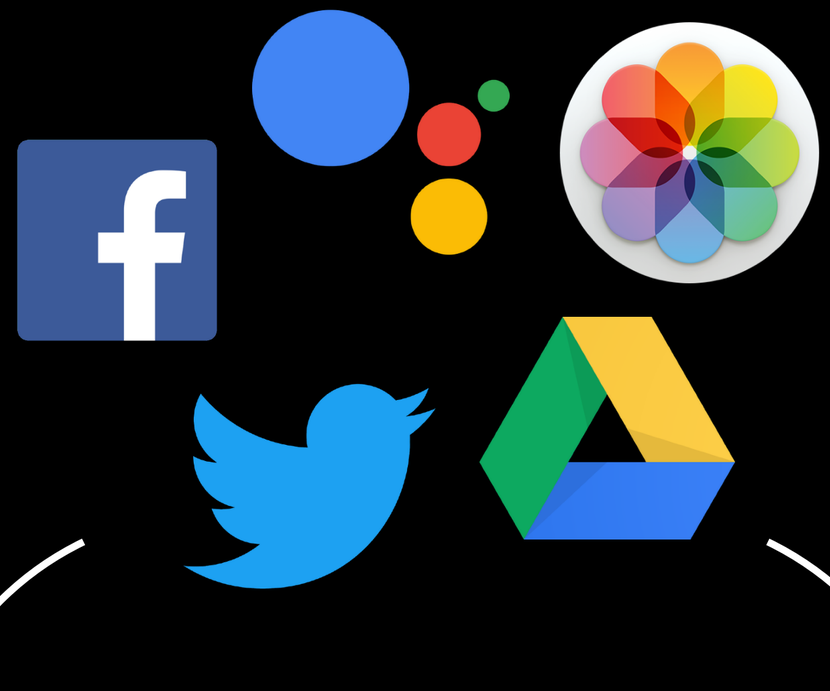
Escolha do Serviço "That"



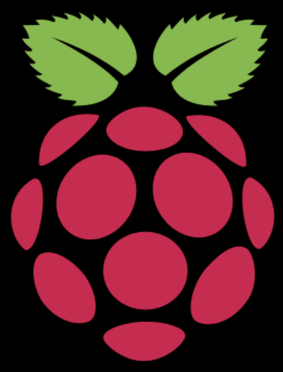
Configuração do Serviço "That"



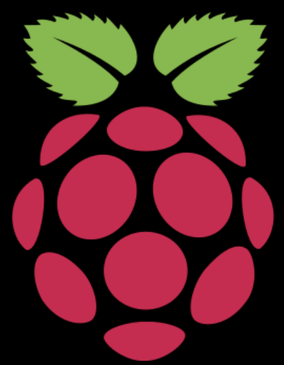
SMS com Tweets de um Perfil



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

ifttt.com/create/if?sid=0

Back

Choose a service

Step 1 of 6

webhooks

Webhooks

About Blog Help Jobs Terms Privacy

Download on the App Store

GET IT ON Google play

Serviço Webhooks como "This"

ifttt.com/create/if-receive-a-web-request?sid=2

Back

Complete trigger fields

Step 2 of 6

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)

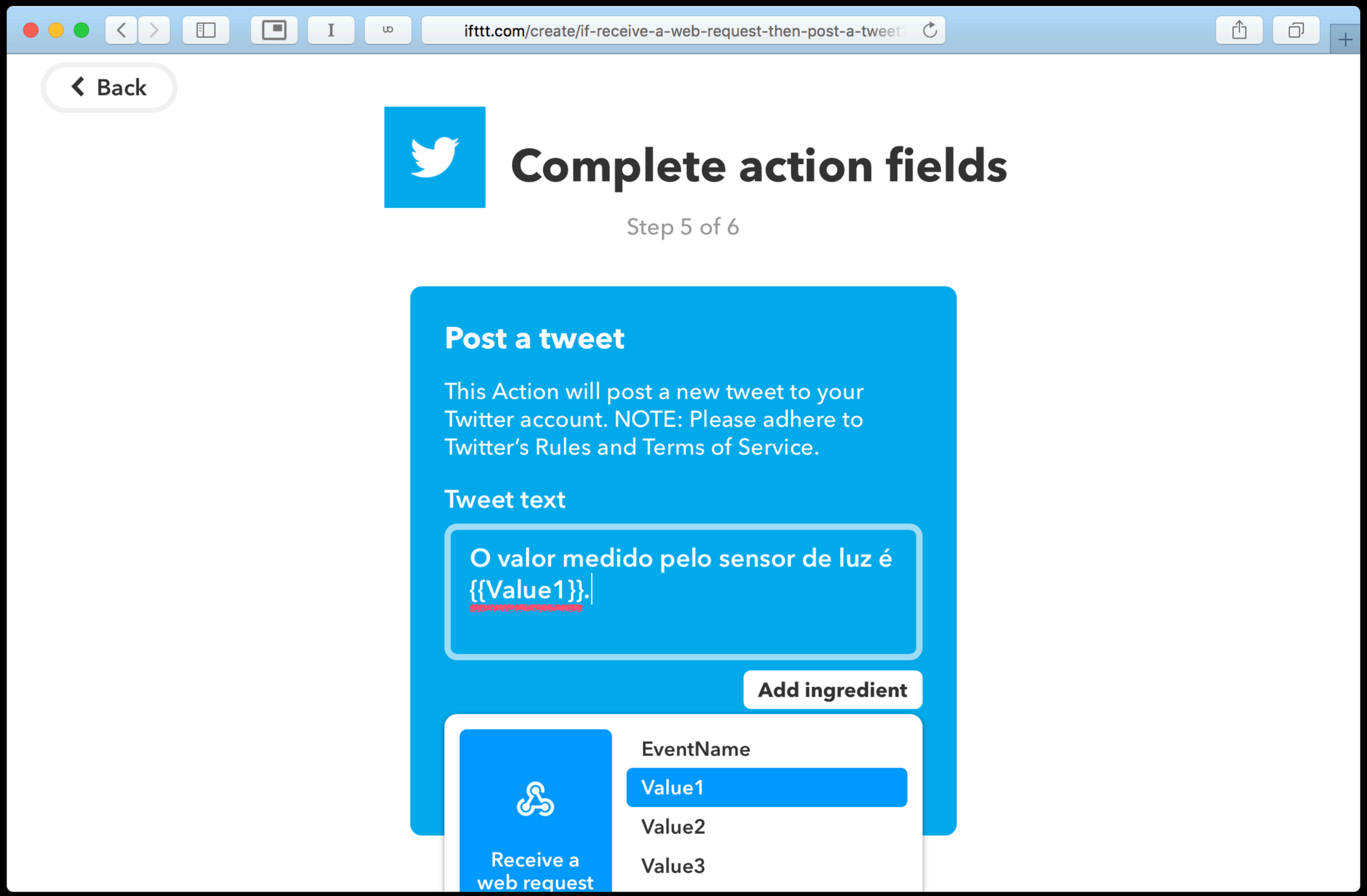
Event Name

botao_pressionado

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

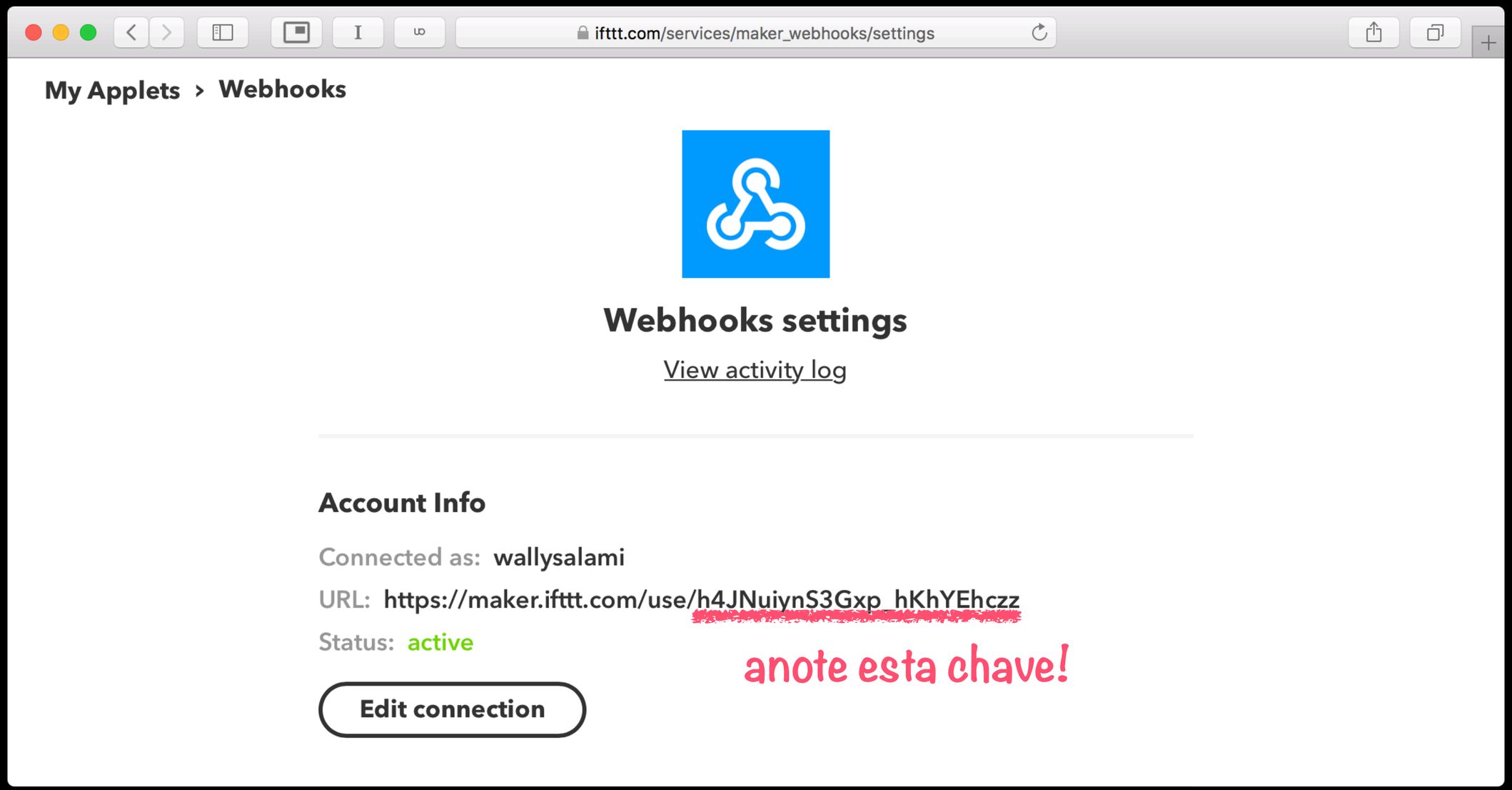
Create trigger

Configuração de um Evento do Webhook



Configuração do "That" como Tweet com Parâmetros

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
```

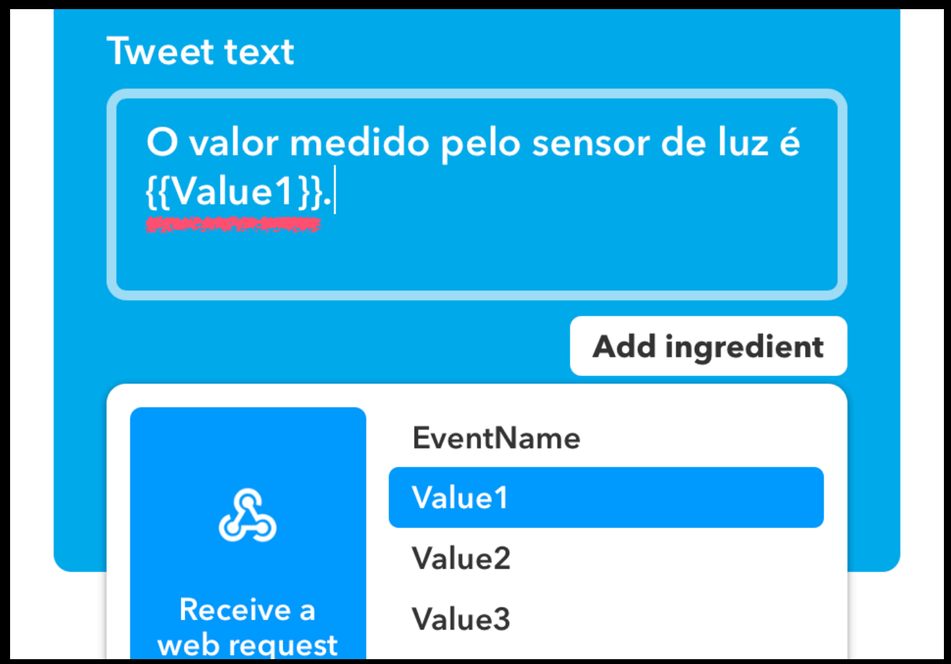
Configuração do Endereço do IFTTT

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

Configuração do Sensor de Luz e do Botão

Tweet Gerado pelo Raspberry Pi

"V" MAIÚSCULO ⟶



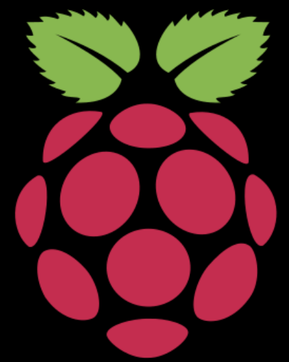
"v" minúsculo ⟵

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

Atenção à Variação da Ortografia de Value1

se houver
previsão do tempo

então

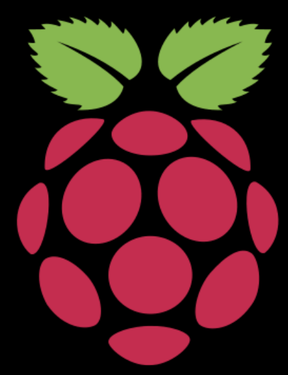


mostre a previsão
no LCD

Exemplo de Integração entre Weather Underground e Raspberry Pi

https://.../tempo/rain

https://.../tempo/sunny



Integração entre Weather Underground e Raspberry Pi via Requisições Web

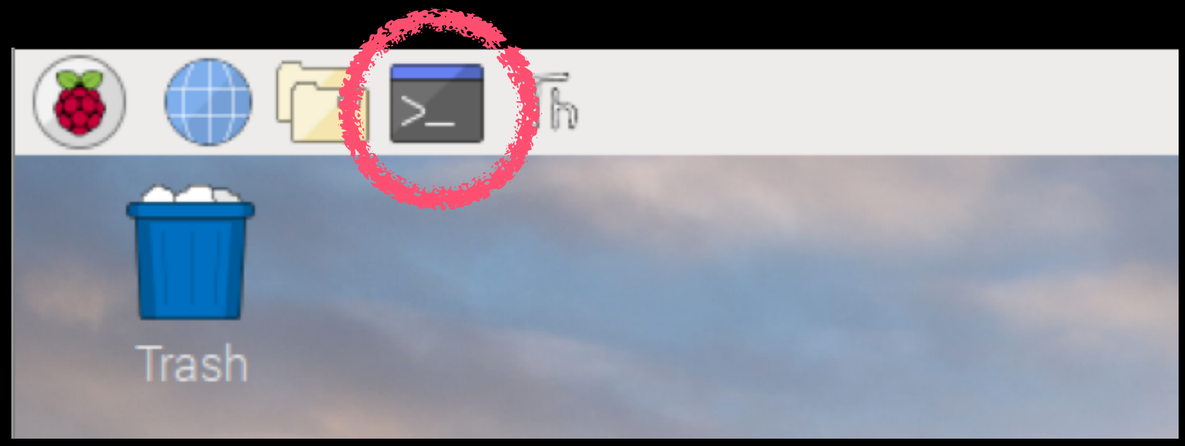
```python
from flask import Flask
from Adafruit_CharLCD import Adafruit_CharLCD
from gpiozero import Buzzer

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=10)
    return "A previsão do tempo é: " + previsao

app.runserver(port=5000, debug=False)
```

Configuração do Servidor com Página para Previsão do Tempo



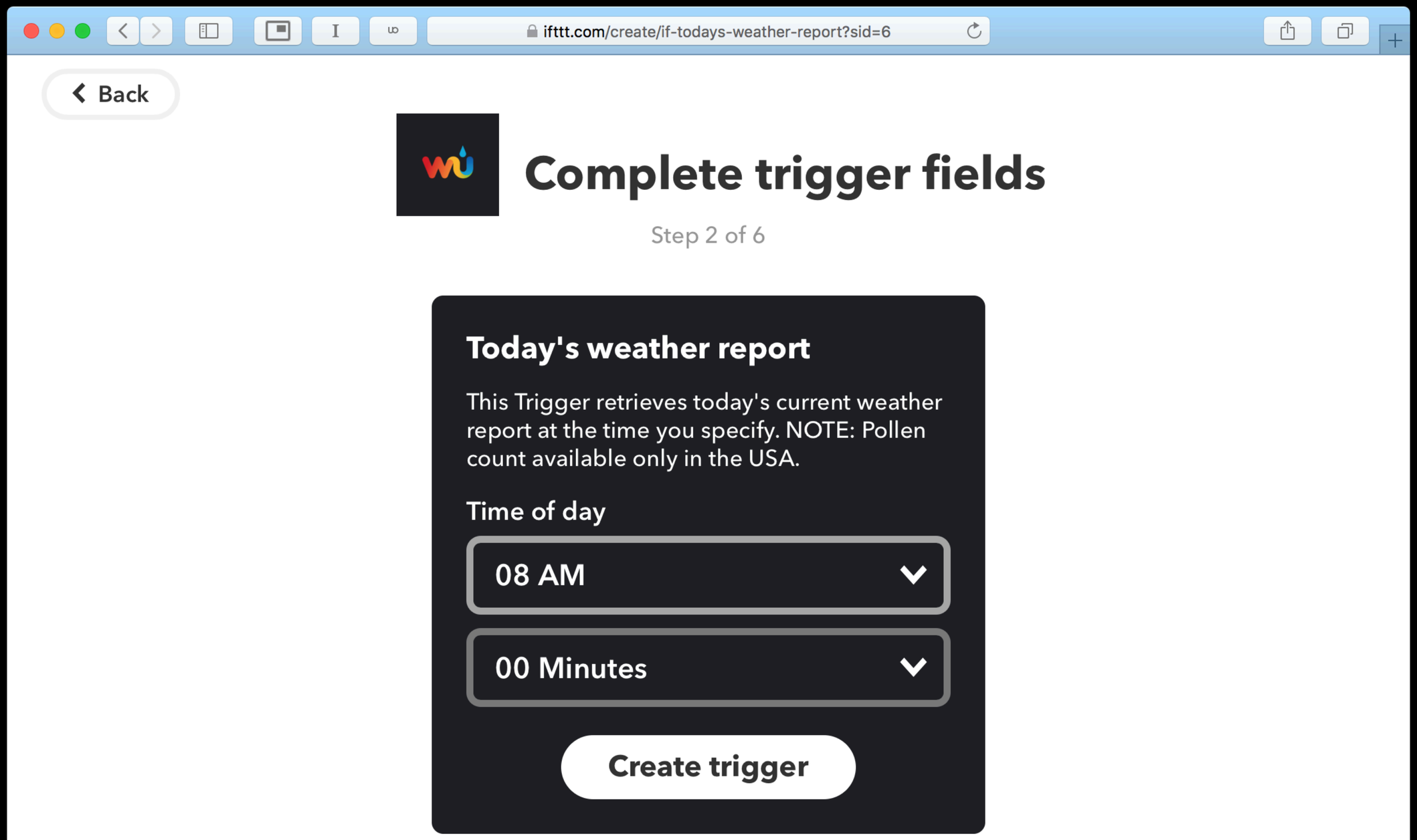

```
aula@raspberrypi ~ $ ngrok http 5000

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



Configuração do Weather Underground como "This"

ifttt.com/create/if-todays-weather-report-then-make-a-web-rec

Back

Complete action fields

Step 5 of 6

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/tempo/{{CurrentCondition}}

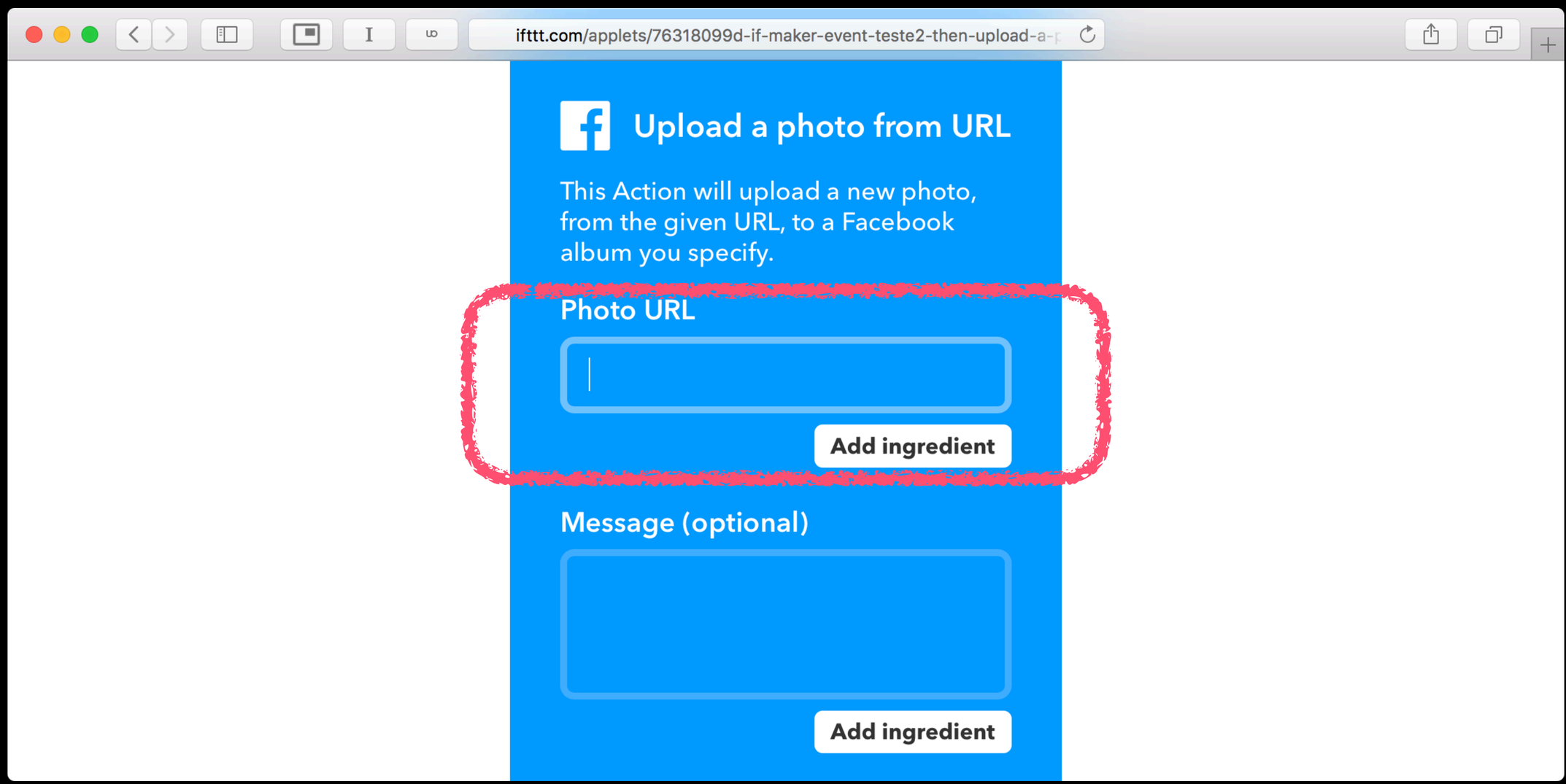
Surround any text with "<<<" and ">>>" to escape the content

Add ingredient

Method

GET

Configuração do Webhook como "That"



URL para Envio de Arquivos

localhost:5000/foto1234.jpeg ⟶

foto1234.jpeg

```
from flask import Flask, send_from_directory

app = Flask(__name__)

@app.route("/<string:nome_da_foto>")
def foto(nome_da_foto):
    return send_from_directory("pasta", nome_da_foto)

app.runserver(port=5000, debug=False)
```

Requisição de um Arquivo

Resumo da Opera

Funcionalidade	Comandos

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

ARecord
acessar documentação
	`from os import system`
	`system("arecord --duration 5 --format cd audio.wav")`
	`from subprocess import POpen`
	`aplicativo = POpen("arecord audio.wav") • 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!"}`
	`post(endereco, json=dados)`
	`endereco = base + "/sendPhoto" • endereco = base + "/sendVoice"`
	`arquivos = {"photo": open("foto.jpg", "rb")}`
	`post(endereco, data=data, files=arquivos)`
	`resultado = get(base + "/getUpdates", json=dados)`

Resumo para o Projeto 04

Funcionalidade	Comandos

Datas e Horas
acessar documentação
	`from datetime import datetime, timedelta`
	`tempo = datetime(2018, 3, 28, 15, 35, 12) • datetime.now()`
	`novo_tempo = tempo + timedelta(months=4)`
	`texto_com_tempo_formatado = tempo.strftime("%H:%M:%S")`

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

Sensor de Distância
acessar documentação
	`from gpiozero import DistanceSensor`
	`sensor = DistanceSensor(trigger=17, echo=18)`
	`sensor.distance • sensor.threshold_distance`
	`sensor.wait_for_in_range() • sensor.wait_for_out_of_range()`
	`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_one(dados) • colecao.insert_many(lista_de_dados)`
	`busca = {"chave": valor} • documento = colecao.find_one(busca)`
	`documentos = colecao.find_many(busca) • list(documentos)`
	`colecao.find_many(busca, sort=ASCENDING)`

Resumo para o Projeto 03

Funcionalidade	Comandos

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

Receptor Infravermelho	`from lirc import init, nextcode`
	`receptor = 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=True)`

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(porta_da_GPIO)`
	`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(porta_da_GPIO)`
	`botao.is_pressed • led.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.duration • player.pt_step(-1)`
	`player.metadata["Title"] • player.metadata["Artist"]`
	`player.volume = 70 • player.speed = 2`

Resumo para o Projeto 01

Funcionalidade	Comandos
Entrada / Saída	`x = input("Digite um número: ") • print("Resultado: ", x)`
Listas acessar documentação	`lista = [1, 2, 3] • lista2 = ["texto", [0, 0], 5];` `lista[0] • lista.append(elemento) • lista.remove(indice)`
Dicionários acessar documentação	`dicionario = {"chave 1": 42, "chave 2": [1, 2, 3]}` `dicionario["chave 1"] • dicionario["chave 3"] = "Olá!"`
Textos	`x = "aspas duplas" • y = 'aspas simplas' • x + "\n" + y`

Condicionais

```
if x == 0:
    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 i in [1, 2, 3]:
    ...
```

```
for i in range(1, 4):
    ...
```

```
while x > 1:
    ...
```

Criação de Funções

```
def funcao1(x):
    return x + 2
```

```
def funcao2(x, y, z):
    ...
```

```
def funcao3():
    ...
```

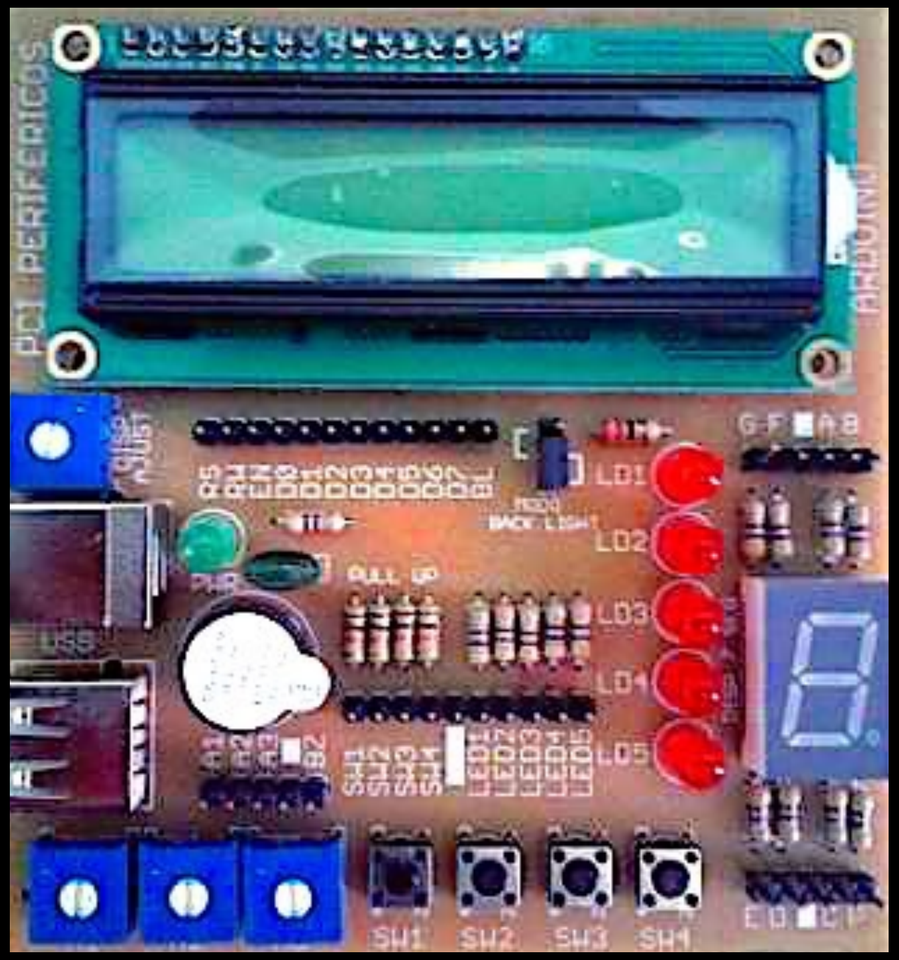
Prática

janks.link/micro/projeto05.zip

Material do Projeto 05

Testes Iniciais

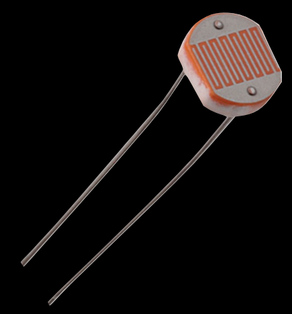
GPIO 2, 3, 4, 5, 6, 7



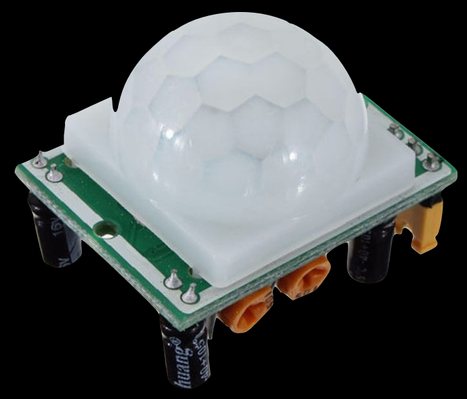
GPIO 16

GPIO 11, 12, 13, 14

GPIO 21
GPIO 22
GPIO 23
GPIO 24
GPIO 25



GPIO 8



GPIO 27

Conexões com as Portas da GPIO

Testes Iniciais

Faça com que a intensidade de brilho do LED 1 varie de acordo com a luz captada no sensor.
↪ DICA: use um **`While True`**.

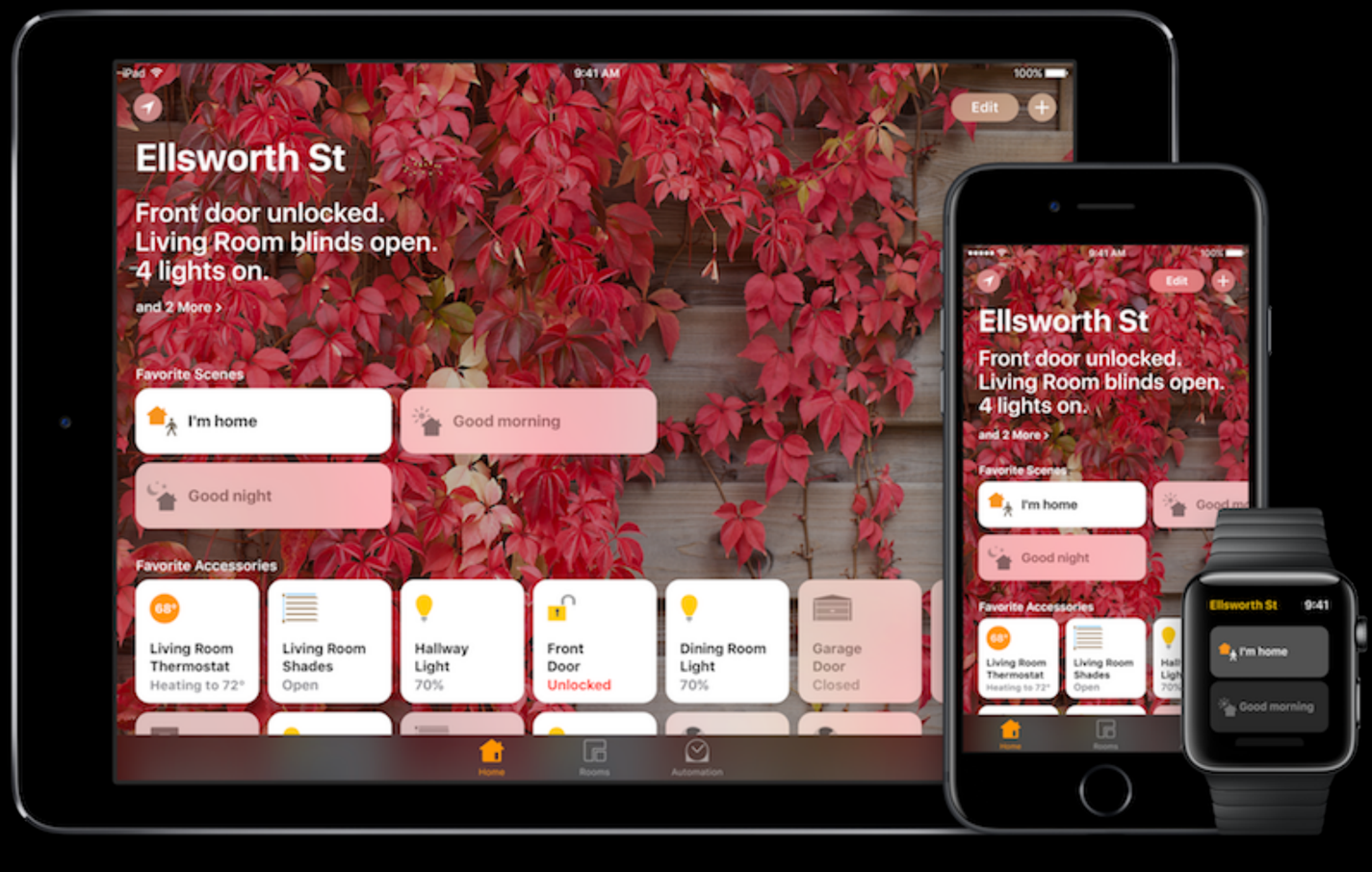
Acenda o LED 2 ao detectar um movimento, e apague-o ao detectar a inércia.

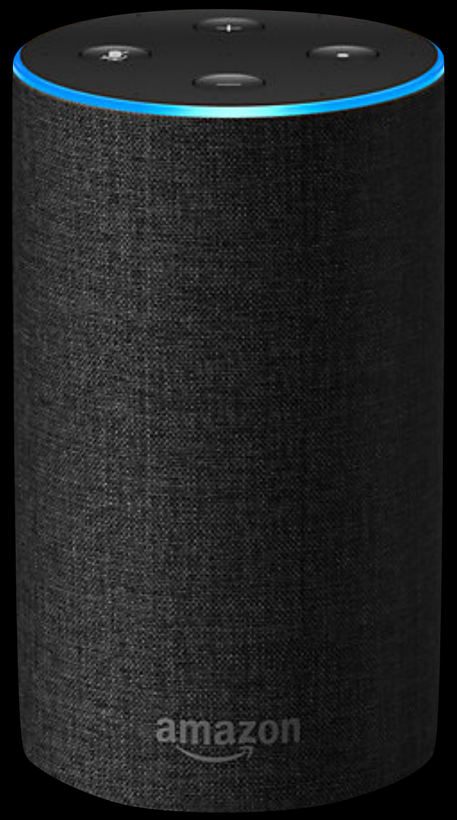
Acenda o LED 3 ao detectar um movimento e apague-o somente se não houver movimento por 8 segundos.
↪ DICA: use um **`Timer`** em caso de inércia e cancele-o se houver movimento.

Configure a integração do IFTTT com o Google Sheets.

Ao apertar o botão 1, adicione uma linha com um texto qualquer em uma planilha do Google Sheets.

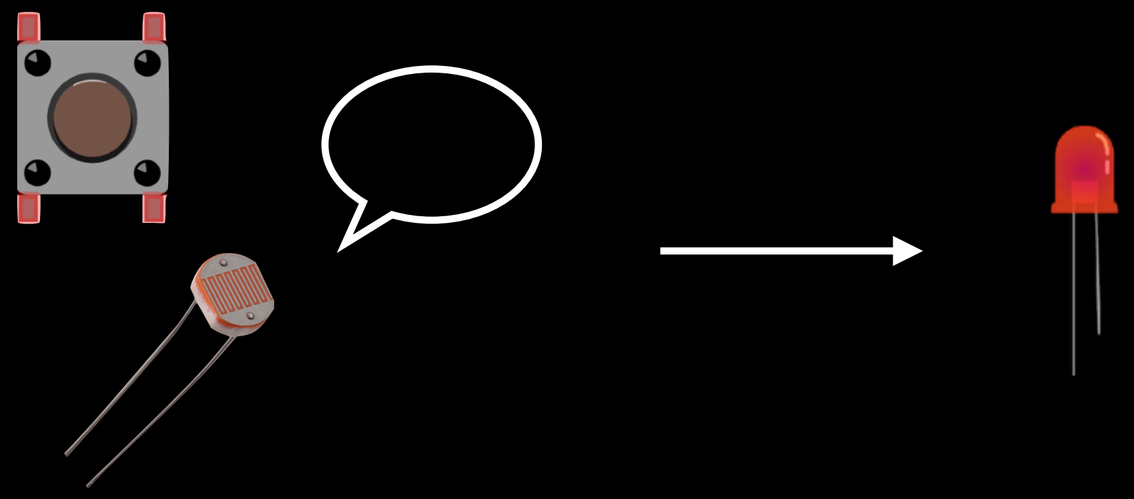
Implementação


Ellsworth St
Front door unlocked.
Living Room blinds open.
4 lights on.
and 2 More ›
Edit
Favorite Scenes
I'm home
Good morning
Good night
Favorite Accessories
Living Room Thermostat
Heating to 72°
Living Room Shades
Open
Hallway Light
70%
Front Door
Unlocked
Dining Room Light
70%
Garage Door
Closed
Home
Rooms
Automation


amazon


PHILIPS

Smart Home



docs.google.com/spreadsheets/d/1lH

Luzes

File Edit View Insert Format Data Tools Add-ons Help All ...

SHARE

100% $ % .0 .00 123 Arial 10

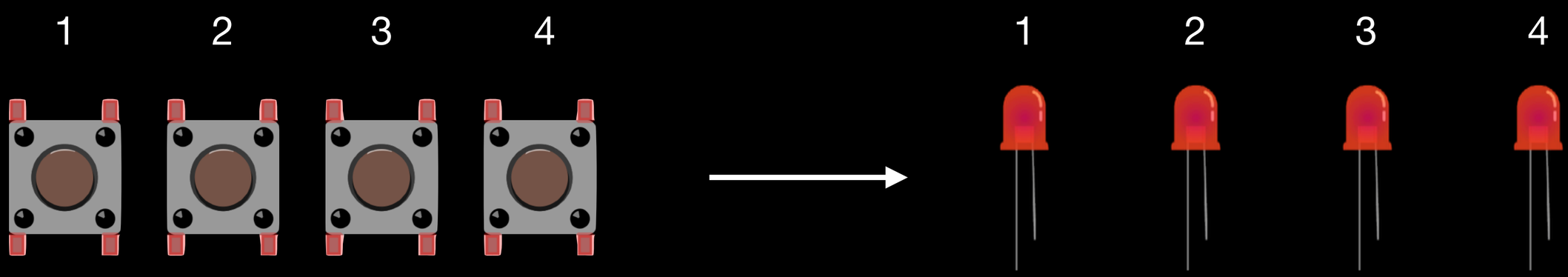
fx

	A	B	C	D	E	F
1	Luz 1	aceso	April 8, 2018 at 06:30PM			
2	Luz 1	apagado	April 8, 2018 at 06:30PM			
3	Luz 2	aceso	April 8, 2018 at 06:31PM			
4	Luz 2	apagado	April 8, 2018 at 06:31PM			
5	Luz 3	aceso	April 8, 2018 at 06:31PM			
6	Luz 1	aceso	April 8, 2018 at 06:31PM			
7	Luz 2	aceso	April 8, 2018 at 06:31PM			
8	Luz 1	aceso	April 8, 2018 at 08:34PM			
9	Luz 2	aceso	April 8, 2018 at 08:34PM			
10	Luz 2	apagado	April 8, 2018 at 08:34PM			
11	Luz 1	apagado	April 8, 2018 at 08:34PM			
12	Luz 5	aceso	April 8, 2018 at 08:35PM			
13	Luz 5	apagado	April 8, 2018 at 08:36PM			
14	Luz 2	aceso	April 8, 2018 at 08:39PM			
15	Luz 2	apagado	April 8, 2018 at 08:39PM			

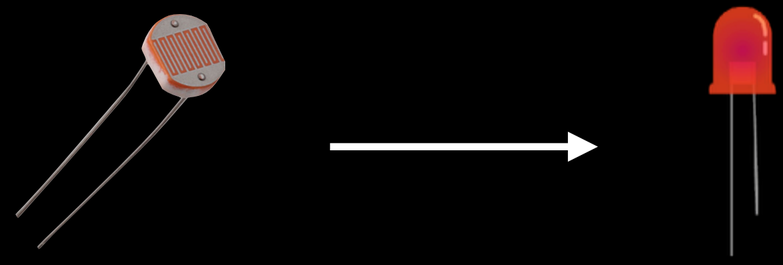
Sheet1

Smart Home Caseiro com Raspberry Pi

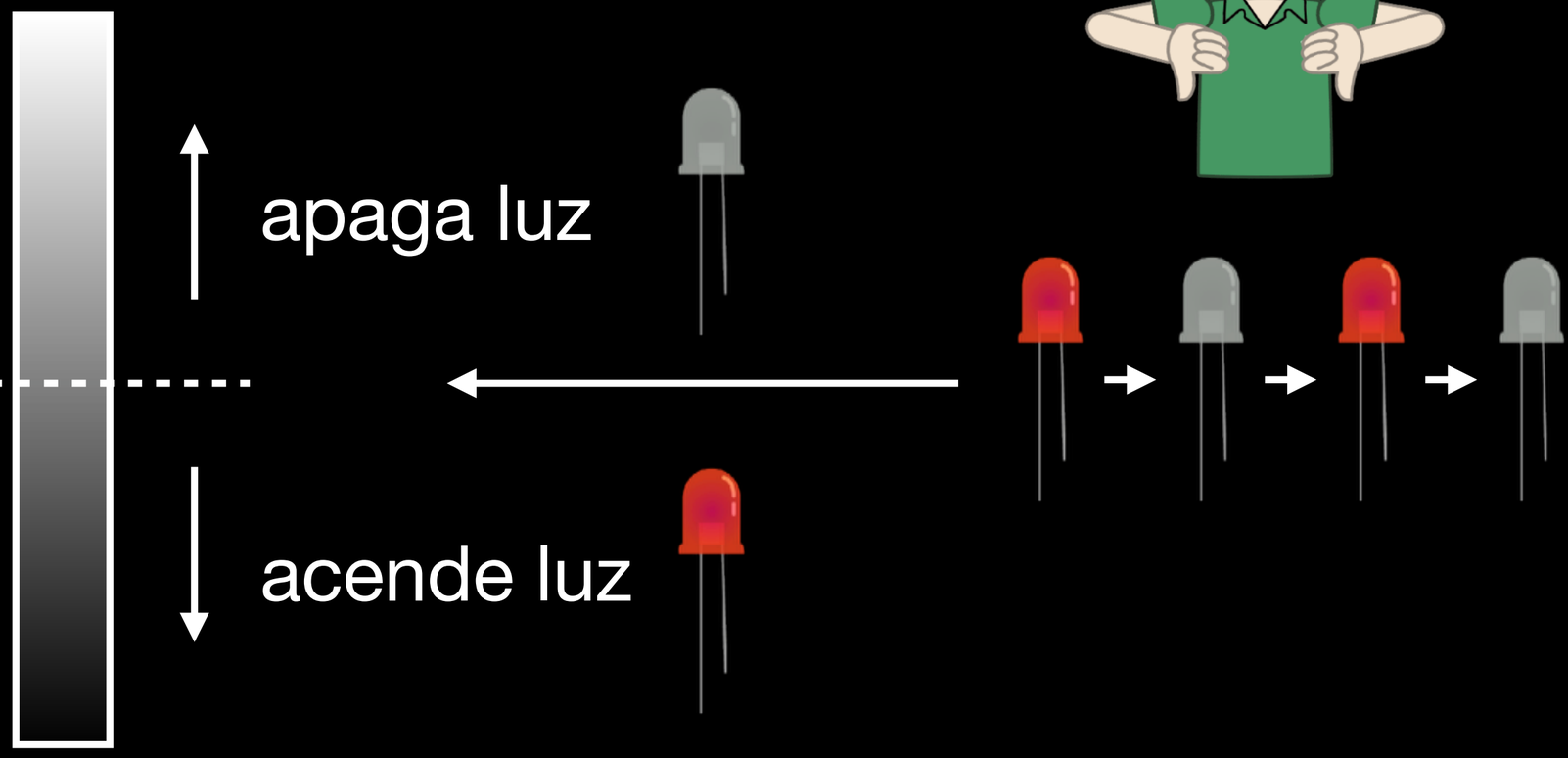
aperte para ligar, aperte de novo para desligar



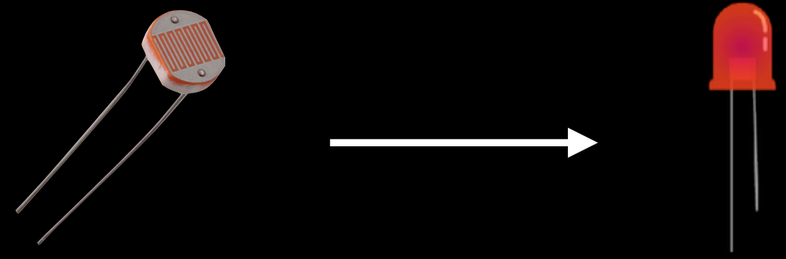
Parte 1: Controle de Luzes por Botão

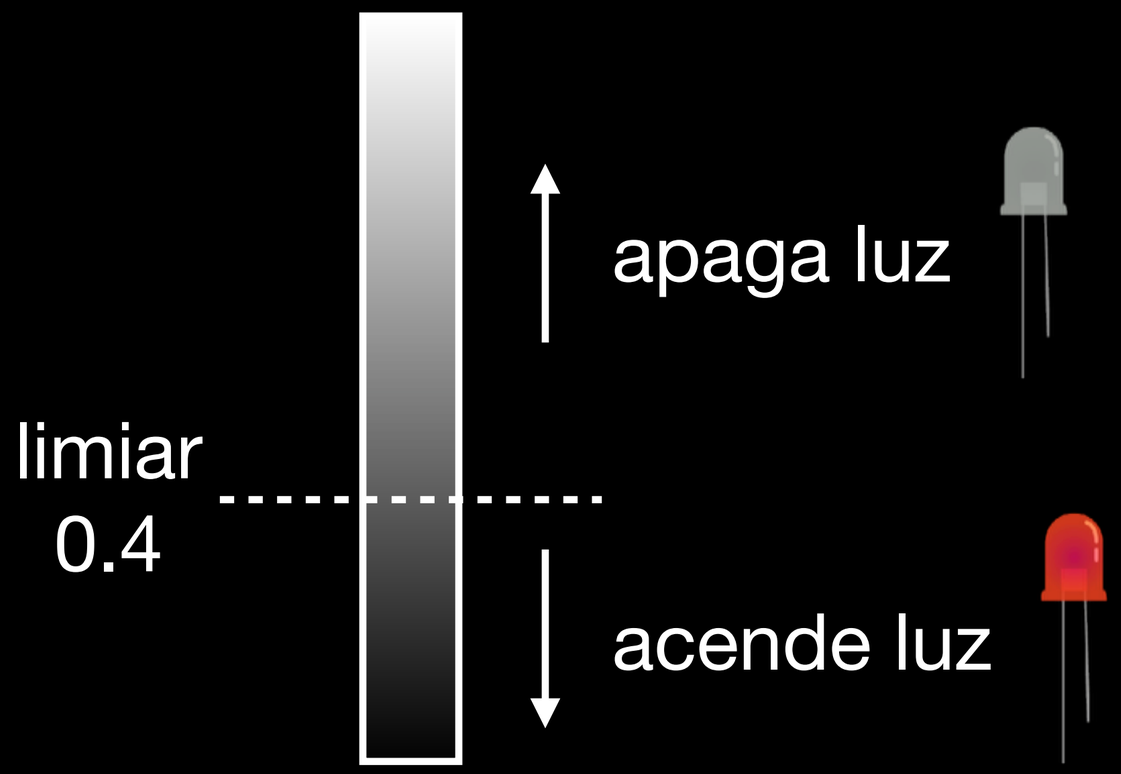


Luz pode ficar piscando em alguns casos de luz...

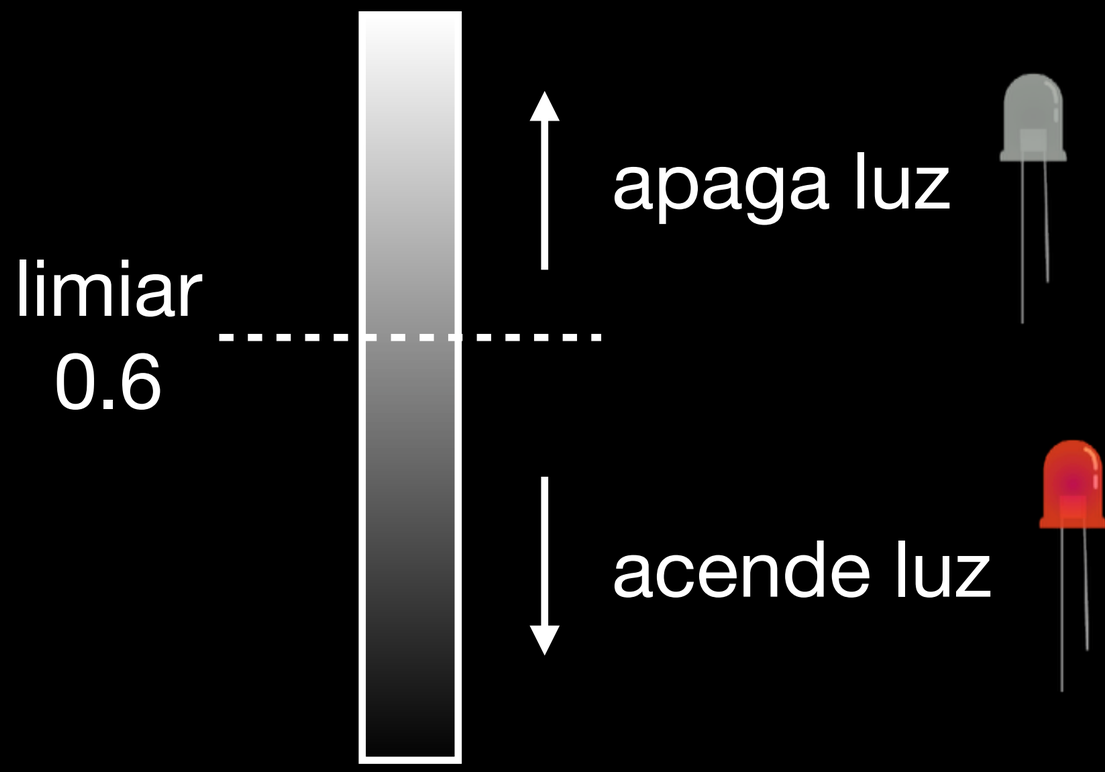


Parte 2: Controle de Luzes por Sensor de Luz com 1 Limiar





se luz estiver apagada



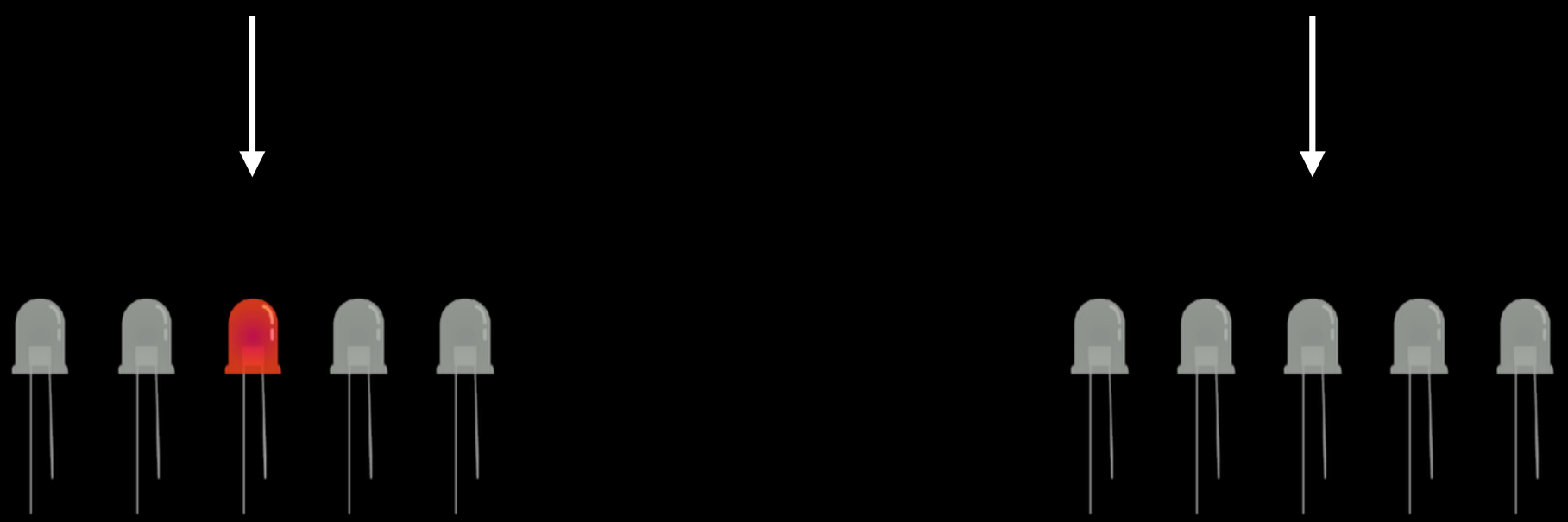
se luz estiver acesa

turn on light number 3

turn off light number 3

http://.../luz/3/on

http://.../luz/3/off



Parte 3: Controle de Luzes por Voz (com Listagem em uma Planilha)

Implementação

Implemente a função **`salvar_na_planilha`** que receba o número do LED e o seu estado, gerando uma nova linha na planilha (como foi ilustrado). Teste a função manualmente.
↳ DICA: crie uma lista de LEDs no script.

Ao apertar os botões 1, 2, 3 e 4, alterne o estado dos LEDs 1, 2, 3 e 4, respectivamente. Depois, atualize a planilha.

Ao detectar escuridão, acenda o LED 5. Ao detectar luz, apague-o. Exiba o valor do sensor de luz no LCD. Por fim, atualize a planilha.
↳ DICA: varie o threshold como explicado.

Crie um **servidor** com uma página que receba o número do LED e o estado desejado (on/off), e altere o LED solicitado. Acesse a página em localhost:5000.
↳ DICA: use os parâmetros `int` e `string` na rota. Teste inicialmente com localhost:5000, depois use o Ngrok.

Integre o **Google Assistant** com o IFTTT para controlar as luzes por voz pelo app. E, mais uma vez, não esqueça de atualizar a planilha.

```
...

print("Enviando linha para planilha...")

...

print("Planilha enviada!")
```

```
...

resposta post(endereco, json=dados)
print(resposta.text)

...
```

DICA: Acompanhamento das Etapas com Print

Aperfeiçoamento

Email


```
Alguém entrou na casa!<br>
<img src='https://.../foto15:25:37.jpeg'><br>
<img src='https://.../foto15:25:40.jpeg'><br>
<img src='https://.../foto15:25:43.jpeg'><br>
...
```

Email com Fotos ao Detectar Movimento

Aperfeiçoamento

Ao detectar movimento, inicie um Timer que capture uma foto a cada 3 segundos; ao detectar inércia, cancele o timer. Salve as fotos com a hora no nome na pasta "fotos".
↪ DICA: veja o slide a seguir.

Crie um servidor com uma página que receba o nome da foto e a retorne. Teste a página no navegador com o localhost:5000.

Em caso de inércia, envie um email com as fotos caso haja pelo menos 3 delas.
↪ DICA: crie uma lista com o nome das fotos tiradas desde a última inércia. Gere o texto do email com tags HTML, conforme exemplificado.

```
>>> from datetime import datetime
>>> agora = datetime.now()
>>> hora = agora.strftime("%H:%M:%S")
>>> nome_do_arquivo = "foto-" + hora + ".jpeg"
>>> nome_do_arquivo
foto-15:25:37.jpeg
```

DICA: Nome da Foto com Data e Hora