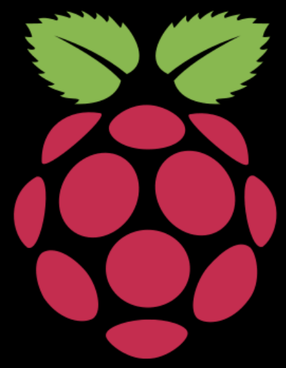


# Projeto 02

## Controle Remoto

Jan K. S. – janks@puc-rio.br

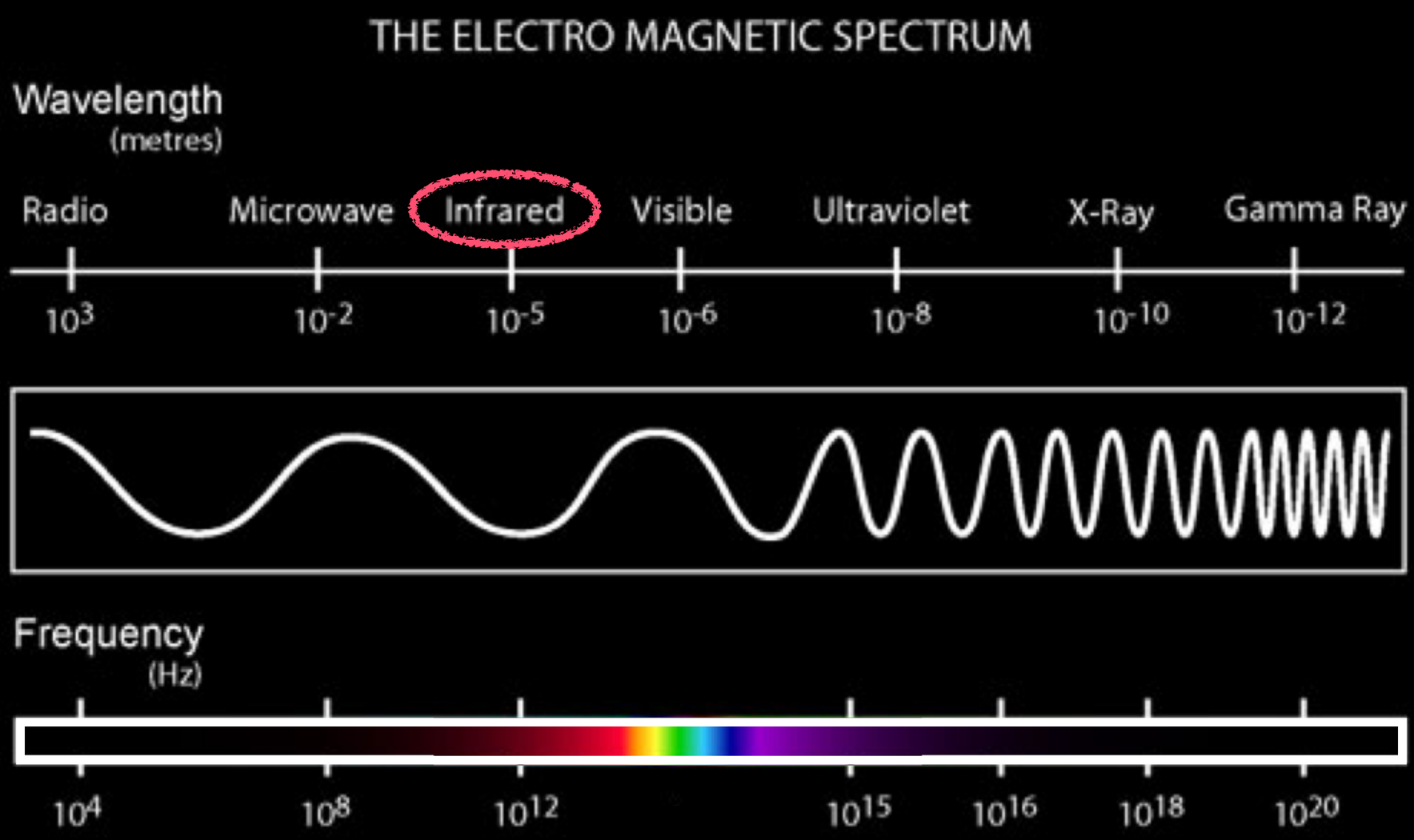
ENG 1419 – Programação de Microcontroladores

# Hardware


Controle Remoto Infravermelho



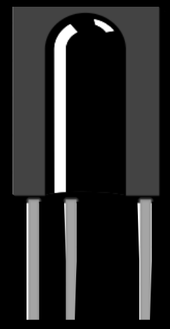

Exemplos de Dispositivos com Controle Remoto Infravermelho



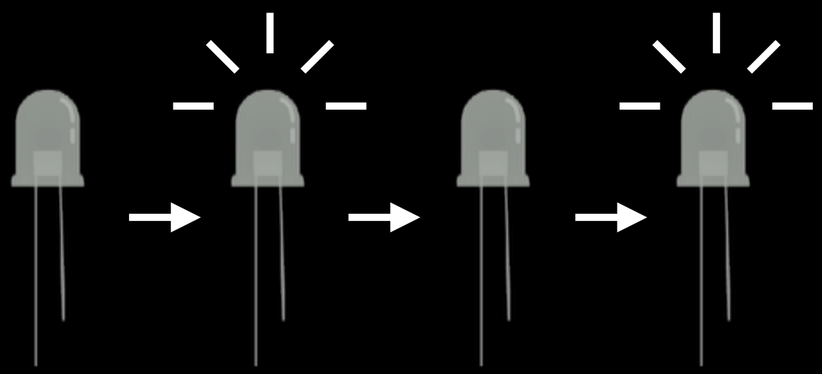

Diferentes Tipos de Onda Eletromagnética de Acordo com a Frequência

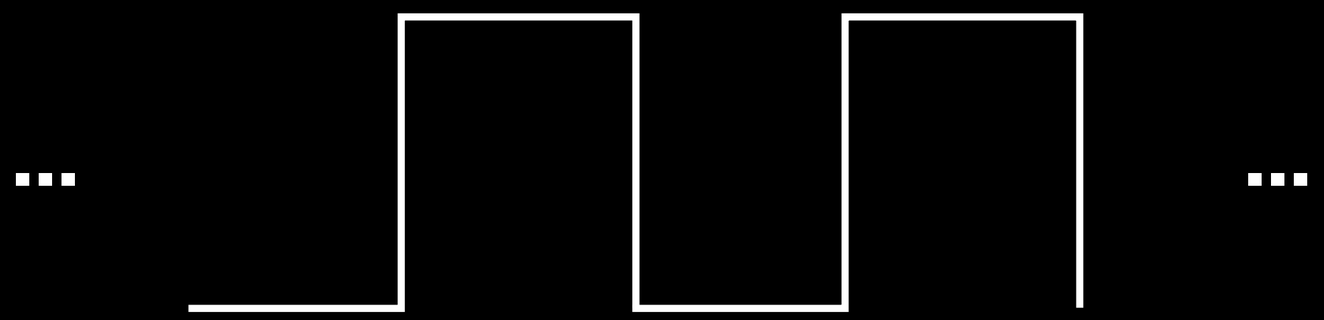
LED Emissor Infravermelho

Receptor Infravermelho



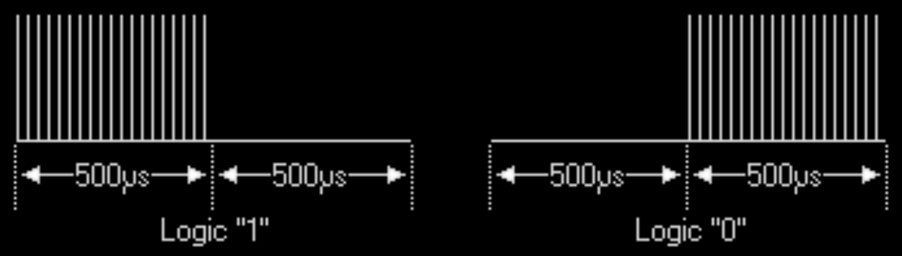
Sinal Transmitido

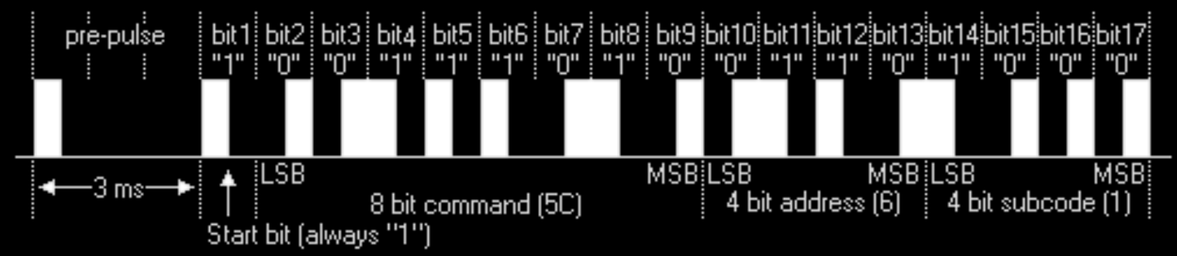




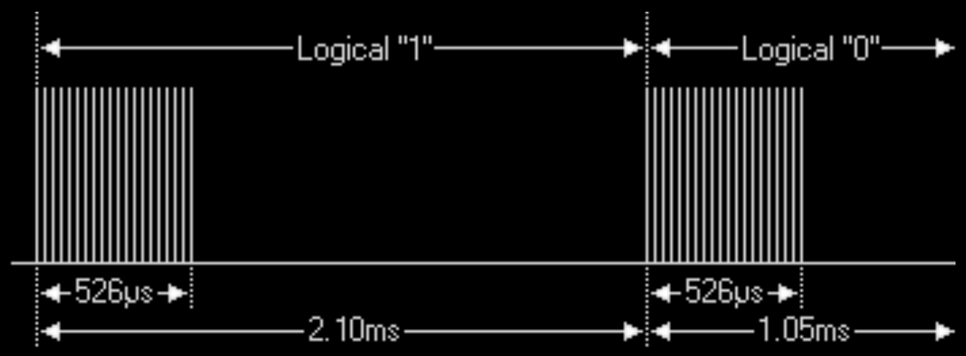
Transmissão de Comandos por Infravermelho

Nokia



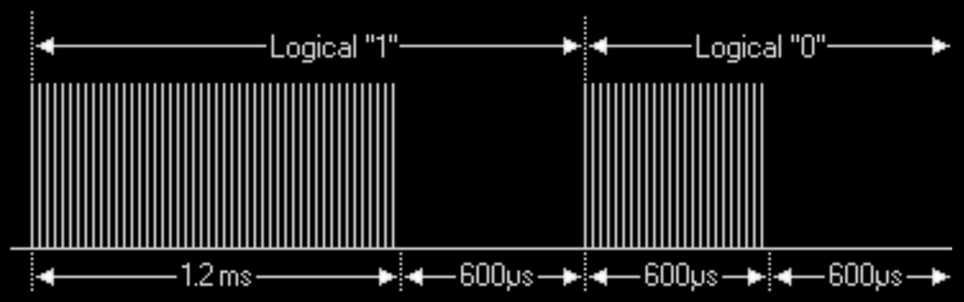



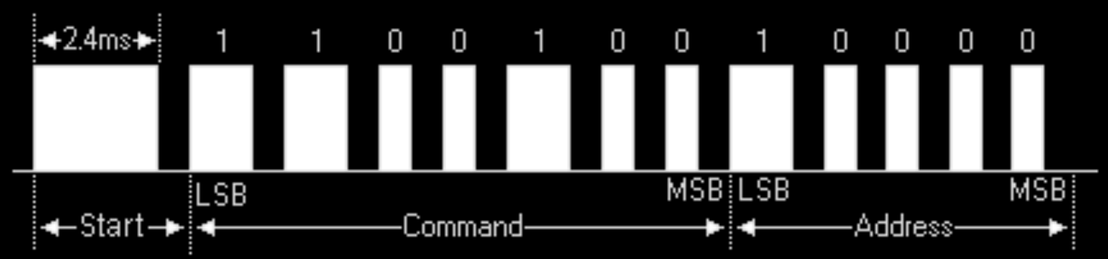

JVC





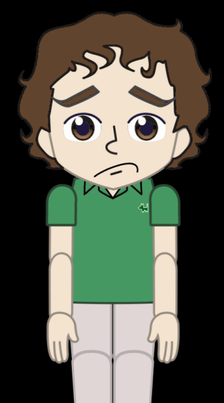
Sony




...

Diferentes Tipos de Codificação



# LIRC

**Linux Infrared Remote Control**

Software para Emissão e Transmissão de Infravermelho no Linux

## **irsend:** envio de sinais

```
root@raspberrypi:/etc/lirc#
root@raspberrypi:/etc/lirc# irsend LIST lircd.conf ""
irsend: 0000000080bf3bc4 KEY_POWER
irsend: 0000000080bfe11e KEY_0
irsend: 0000000080bf49b6 KEY_1
irsend: 0000000080bfc936 KEY_2
irsend: 0000000080bf33cc KEY_3
irsend: 0000000080bf718e KEY_4
irsend: 0000000080bff10e KEY_5
irsend: 0000000080bf13ec KEY_6
irsend: 0000000080bf51ae KEY_7
irsend: 0000000080bfd12e KEY_8
irsend: 0000000080bf23dc KEY_9
irsend: 0000000080bf738c KEY_OK
irsend: 0000000080bfa35c KEY_EXIT
irsend: 0000000080bf19e6 KEY_MENU
irsend: 0000000080bf5ba4 KEY_EPG
irsend: 0000000080bf41be KEY_BACK
irsend: 0000000080bf39c6 KEY_MUTE
irsend: 0000000080bf8976 KEY_PLAY
irsend: 0000000080bfb14e KEY_STOP
root@raspberrypi:/etc/lirc#
root@raspberrypi:/etc/lirc#
```

## **irrecord**: decodificação de um controle

www.ocinside.de

Datei Bearbeiten Ansicht Terminal Gehe zu Hilfe

```
Press RETURN now to start recording.
.....................................................................
Found const length: 112000
Please keep on pressing buttons like described above.
Creating config file in raw mode.
Now enter the names for the buttons.

Please enter the name for the next button (press <ENTER> to finish recording)
KEY_PAGEUP

Now hold down button "KEY_PAGEUP".
Got it.
Signal length is 23

Please enter the name for the next button (press <ENTER> to finish recording)
KEY_PAGEDOWN

Now hold down button "KEY_PAGEDOWN".
Got it.
Signal length is 23

Please enter the name for the next button (press <ENTER> to finish recording)
```

## **irw:** monitoramento de teclas pressionadas

www.ocinside.de

Datei Bearbeiten Ansicht Terminal Gehe zu Hilfe

```
jp@jp-desktop:~$ irw
0000000000000001 00 KEY_PAGEUP /etc/lirc/own_remote.conf
0000000000000001 01 KEY_PAGEUP /etc/lirc/own_remote.conf
0000000000000002 00 KEY_PAGEDOWN /etc/lirc/own_remote.conf
0000000000000002 00 KEY_PAGEDOWN /etc/lirc/own_remote.conf
0000000000000002 01 KEY_PAGEDOWN /etc/lirc/own_remote.conf
0000000000000003 00 KEY_PLAY /etc/lirc/own_remote.conf
0000000000000003 01 KEY_PLAY /etc/lirc/own_remote.conf
0000000000000004 00 KEY_STOP /etc/lirc/own_remote.conf
0000000000000004 01 KEY_STOP /etc/lirc/own_remote.conf
```

# Exemplos de Ferramentas do LIRC

lircd.conf

```
begin remote

  name  mini
  bits           16
  flags SPACE_ENC|CONST_LENGTH
  eps            30
  aeps          100

  header       9118  4506
  one           600  1650
  zero          600   538
  ptrail        600
  repeat       9124  2230
  pre_data_bits   16
  pre_data       0xFF
  gap          108383
  toggle_bit_mask 0x0

      begin codes
          KEY_1                    0xA25D
          KEY_2                    0x629D
          KEY_3                    0xE21D
          KEY_4                    0x22DD
          KEY_5                    0x02FD
          KEY_6                    0xC23D
          KEY_7                    0xE01F
          KEY_8                    0xA857
          KEY_9                    0x906F
          KEY_0                    0x9867
```

lirc_hardware.conf

```
# /etc/lirc/hardware.conf
#
# Arguments which will be used when launching lircd
LIRCD_ARGS="--uinput"

#Don't start lircmd even if there seems to be a good config file
#START_LIRCMD=false

#Don't start irexec, even if a good config file seems to exist.
#START_IREXEC=false

#Try to load appropriate kernel modules
LOAD_MODULES=true

# Run "lircd --driver=help" for a list of supported drivers.
DRIVER="default"
# usually /dev/lirc0 is the correct setting for systems using udev
DEVICE="/dev/lirc0"
MODULES="lirc_rpi"

# Default configuration files for your hardware if any
LIRCD_CONF=""
LIRCMD_CONF=""
```

lirc_options.conf

```
driver = default
device = /dev/lirc0
```

lircrc

```
begin
  button = KEY_1
  prog = aula
  config = KEY_1
end

begin
  button = KEY_2
  prog = aula
  config = KEY_2
end

begin
  button = KEY_3
  prog = aula
  config = KEY_3
end

begin
  button = KEY_4
  prog = aula
```

...

Diversos Arquivos de Configuração do LIRC

GitHub, Inc. github.com/ChristopherRoger

ChristopherRogers1991 / python-irsend

Code Issues 0 Pull requests 0 Projects 0 Pulse

README.md

# python-irsend

This is a simple wrapper for lirc's irsend.

## Install:

```
pip install py_irsend
```

## Basic Usage:

```
>>> from py_irsend import irsend
>>> irsend.list_remotes()
['lasko_heater', 'lights.conf', 'dynex_tv', 'logitech_z906', 'sabrent_hdmi_switch']
>>> irsend.list_codes('logitech_z906')
['POWER', 'INPUT', 'MUTE', 'LEVEL', 'EFFECT', 'VOLUME_DOWN', 'VOLUME_UP']
>>> irsend.send_once('logitech_z906', ['POWER'])
```

Biblioteca para Emitir Sinais pelo LIRC

```
>>> from py_irsend.irsend import *
>>> list_remotes()
['mini', 'net', 'tomate']
>>> list_codes('mini')
['KEY_1', 'KEY_2', 'KEY_3', 'KEY_4', 'KEY_5', 'KEY_6',
'KEY_7', 'KEY_8', 'KEY_9', 'KEY_0', 'KEY_UP',
'KEY_DOWN', 'KEY_LEFT', 'KEY_RIGHT', 'KEY_OK',]
>>> send_once('mini', ['KEY_1'])
>>> 1
```

ATENÇÃO: segundo parâmetro tem que ser uma LISTA de códigos

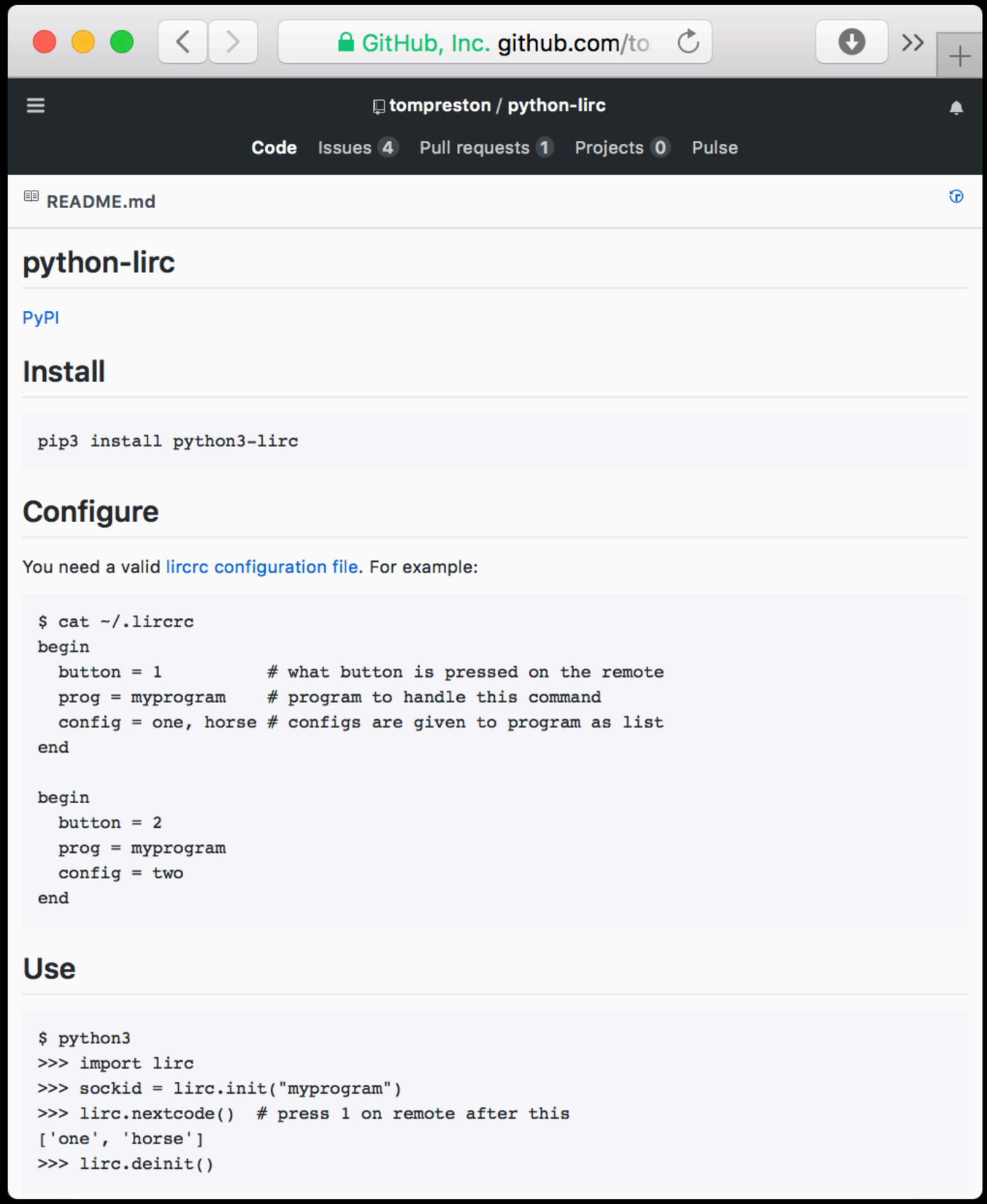
controle remoto "digitou" 1

Comandos para o Emissor Infravermelho

isso pode ser rápido demais para o dispositivo controlado, em alguns casos

```
>>> from py_irsend.irsend import *
>>> send_once('mini', ['KEY_UP', 'KEY_UP', 'KEY_OK'])
>>> from time import sleep
>>> send_once('mini', ['KEY_UP'])
>>> sleep(0.5)
>>> send_once('mini', ['KEY_UP'])
>>> sleep(0.5)
>>> send_once('mini', ['KEY_UP'])
```

Envio de Múltiplos Comandos




Biblioteca para Receber Sinais pelo LIRC

```
>>> from lirc import init, next_code
>>> receptor = init("aula")
>>> next_code()
['KEY_6']
```

"aula" é um nome definido na configuração de comandos

Recebimento de Comandos pelo Emissor Infravermelho

```
>>> from lirc import init, next_code
>>> receptor = init("aula")
>>> while True:
...     codigo = next_code()
...     if codigo == ['KEY_1']:
...         print('Tecla 1 apertada')
...     elif codigo == ['KEY_2']:
...         print('Tecla 2 apertada')
```

Recebimento Contínuo de Comandos

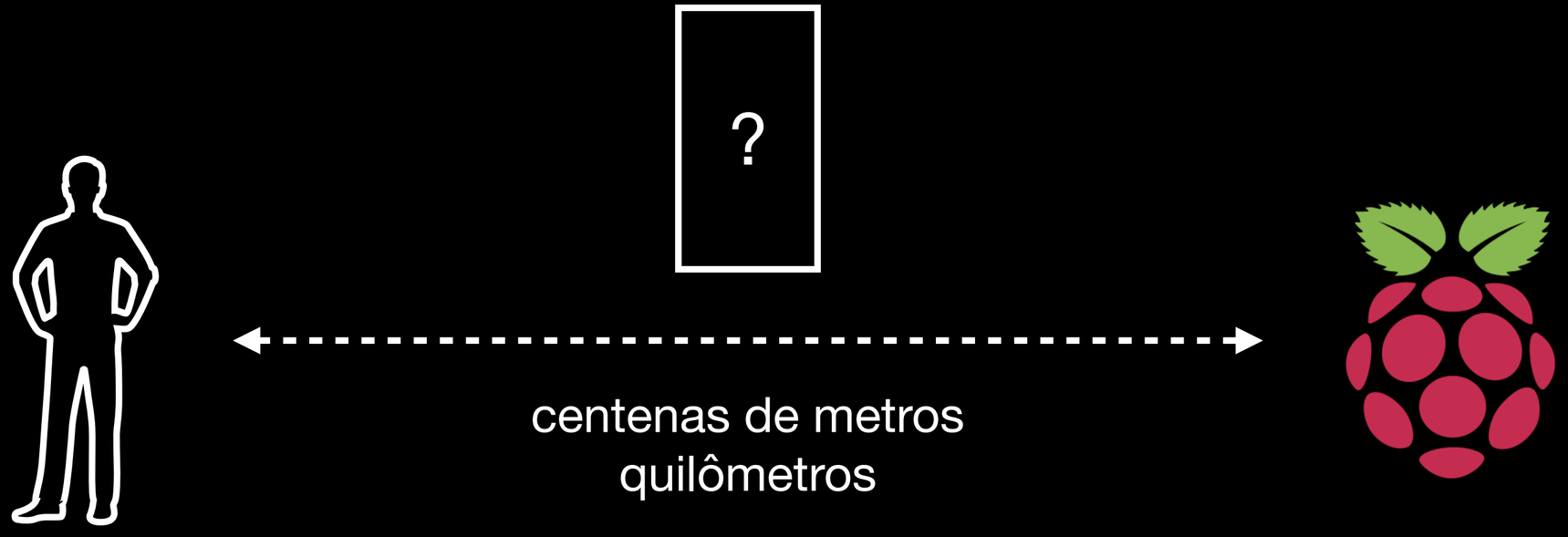
```
>>> from lirc import init, next_code
>>> from gpiozero import LED, Button
>>> led = LED(21)
>>> button = Button(11)
>>> button.when_pressed = led.toggle
>>> receptor = init("aula")
>>> while True:
...     codigo = next_code()
...     if codigo == ['KEY_1']:
...         print('Tecla 1 apertada')
...     elif codigo == ['KEY_2']:
...         print('Tecla 2 apertada')
```

when_pressed nunca vai executar, pois next_code bloqueia qualquer outra chamada enquanto não recebe um sinal

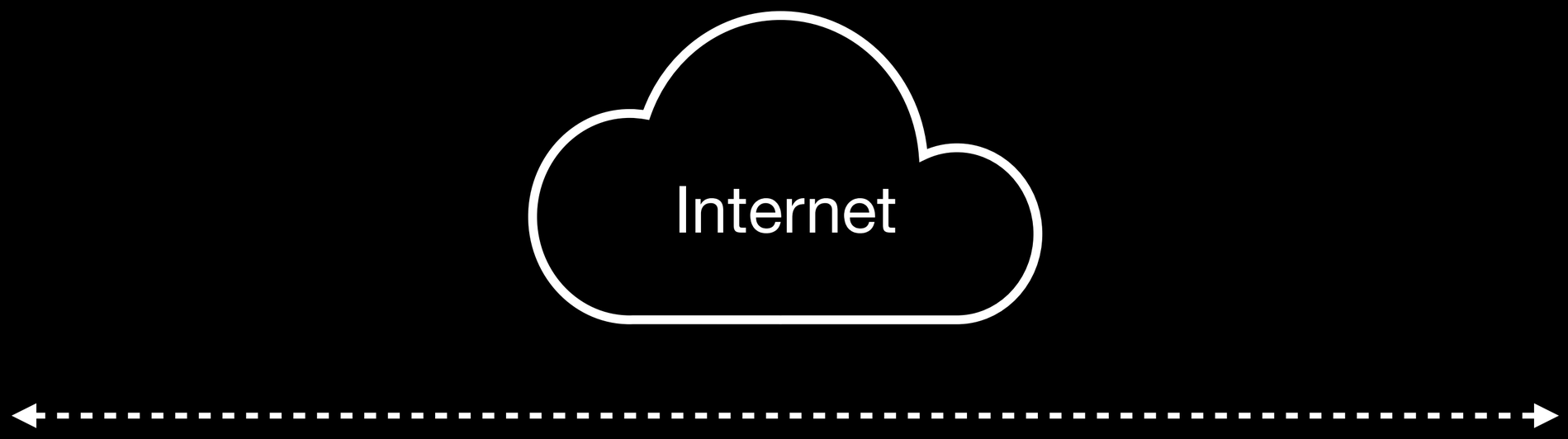
```
>>> from lirc import init, next_code
>>> from time import sleep
>>> from gpiozero import LED, Button
>>> led = LED(21)
>>> button = Button(11)
>>> button.when_pressed = led.toggle
>>> receptor = init("aula", blocking=False)
>>> while True:
...     codigo = next_code()
...     if codigo == ['KEY_1']:
...         print('Tecla 1 apertada')
...     elif codigo == ['KEY_2']:
...         print('Tecla 2 apertada')
...     sleep(0.1)
```

Recebimento Contínuo de Comandos

# Software

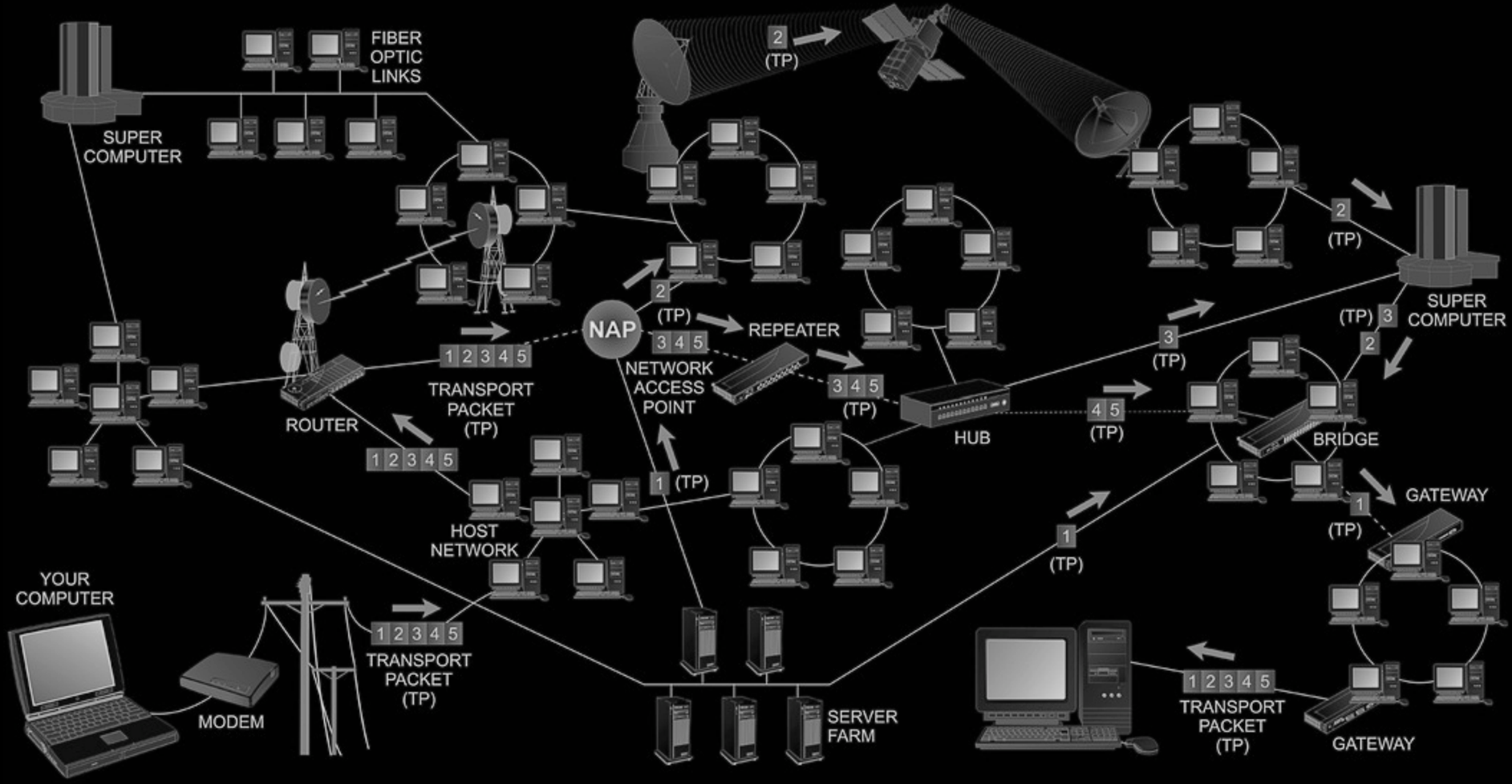




Comunicação Remota a Longas Distâncias

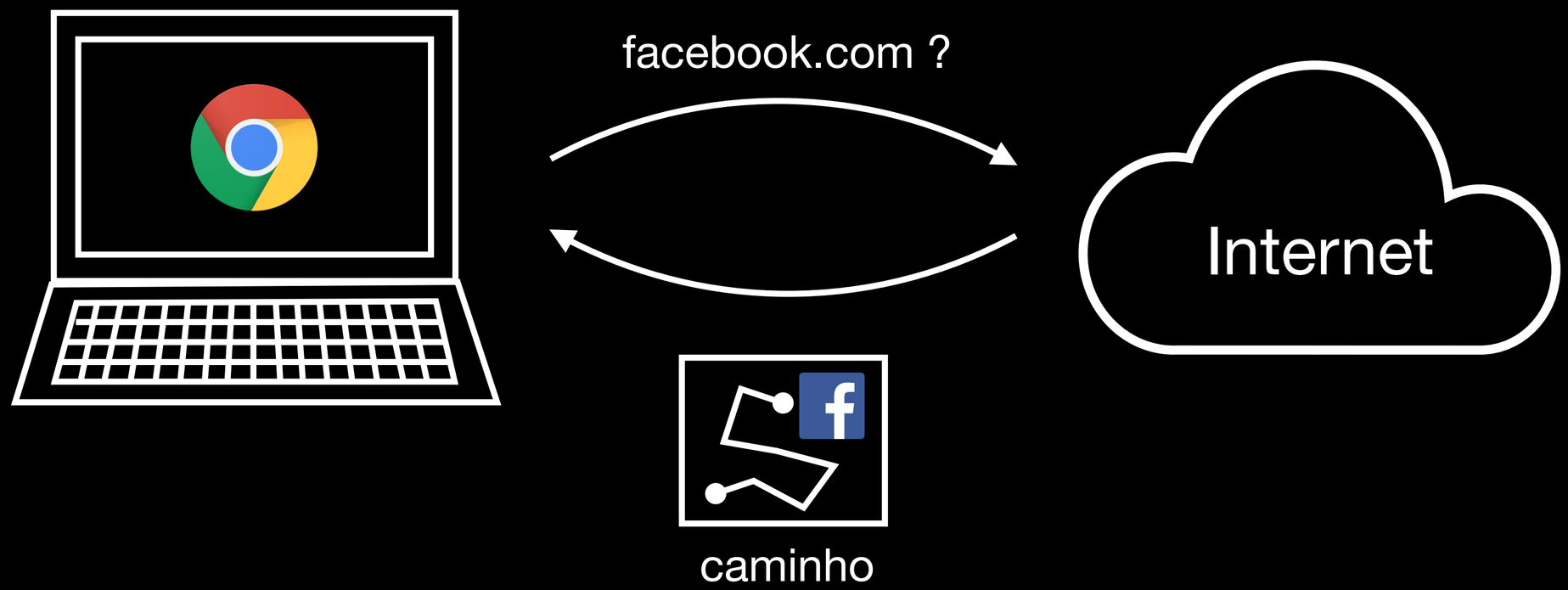



qualquer distância!

Mecanismo Mágico de Conexão Remota



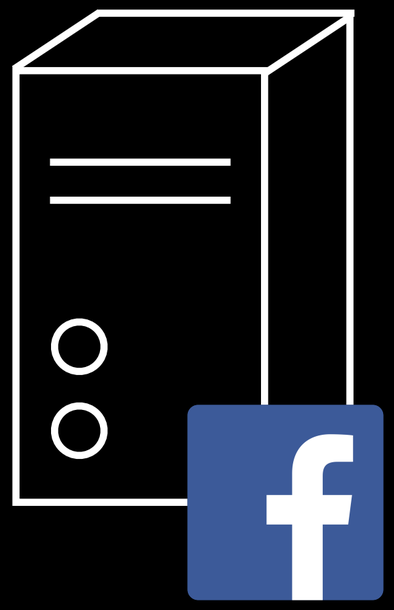

Rede de Computadores na Internet



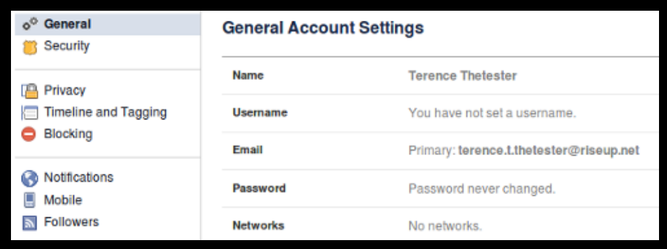

Conexão pela Internet a um Site


facebook.com/robertdowneyjr


servidor



facebook.com/settings




Solicitação de Páginas para Servidores da Internet

programas rodando
dentro do servidor

?

solicitação

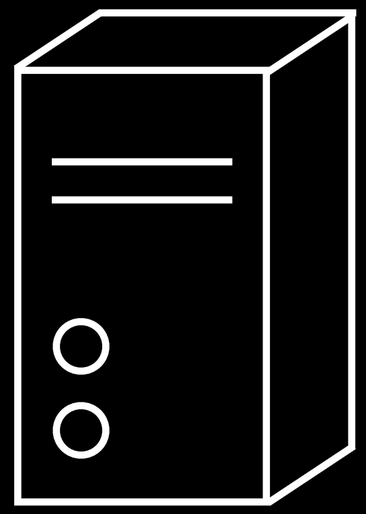
http://

https://

Múltiplos Aplicativos Rodando em um Servidor

solicitação

para a porta 5900



25

80

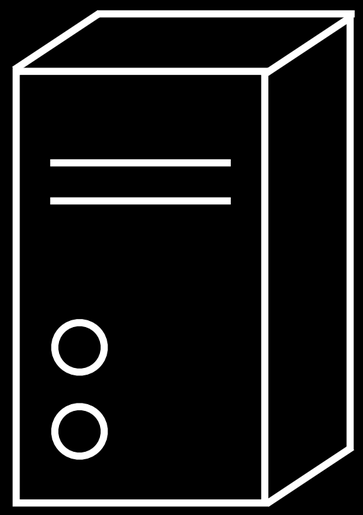
443

5900

http://

https://

Comunicação por Portas



=

Exemplos de Computadores para Criar um Servidor

# Flask

web development,
one drop at a time

Framework Flask

```python
from flask import Flask
app = Flask(__name__)

@app.route("/inicio")
def funcao_da_pagina_inicio():
    return "Funcionou!"

app.run(port=5000, debug=True)
```

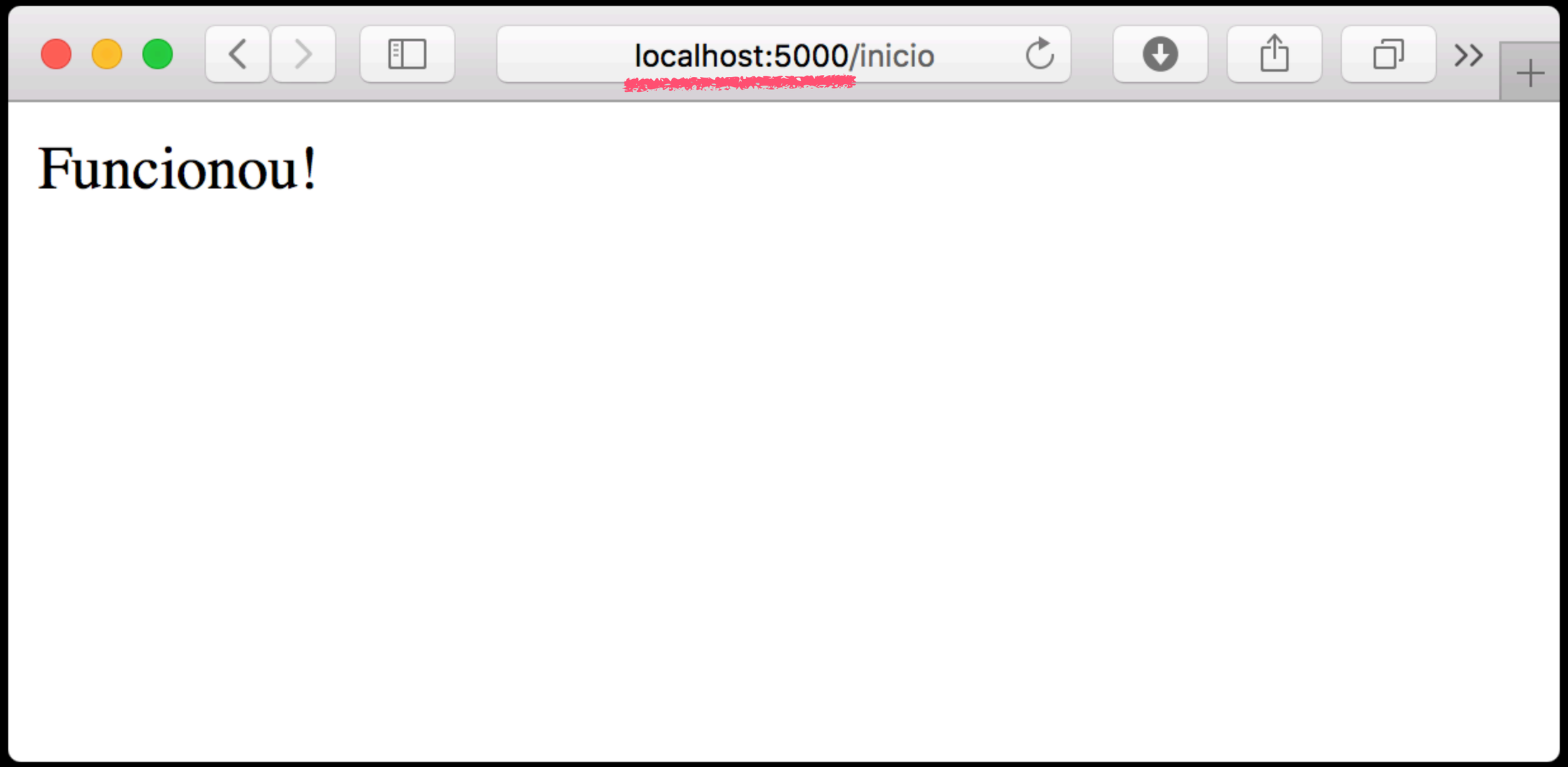
inicialize o aplicativo do servidor com um nome referente ao nosso script atual

rode esta função ao acessar a página "início"

rode o servidor na porta 5000 e exiba possíveis erros do código

Código Base de um Servidor em Flask

procure a página do nosso próprio computador na porta 5000




Carregamento da Página no Navegador

```python
from flask import Flask
app = Flask(__name__)

@app.route("/")
def mostrar_pagina_principal():
    return "Bem-vindo ao meu site!"

@app.route("/sobre")
def mostrar_pagina_sobre():
    return "Olá, meu nome é Jan!"

@app.route("/contato")
def mostrar_pagina_de_contato():
    return "Email: janks@puc-rio.br"

app.run(port=5000, debug=True)
```

Código de um Servidor com Várias Páginas

localhost:5000/

Bem-vindo ao meu site!

localhost:5000/sobre

Olá, meu nome é Jan!

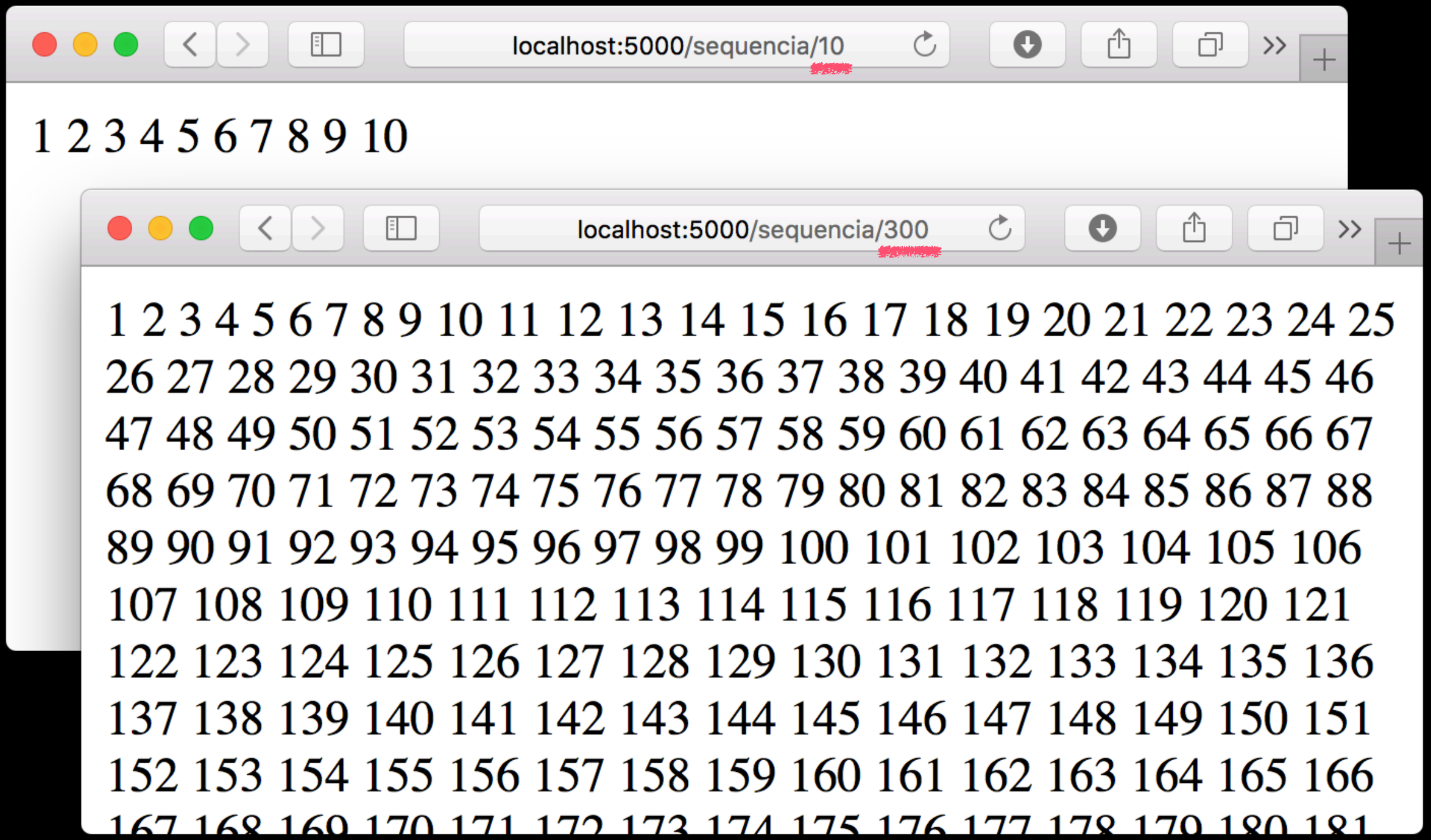
localhost:5000/contato

Email: janks@puc-rio.br

Carregamento das Páginas no Navegador

```python
from flask import Flask
app = Flask(__name__)

@app.route("/sequencia/<int:numero_final>")
def imprimir_sequencia(numero_final):
    texto_da_contagem = ""
    for numero in range(1, numero_final + 1):
        texto_da_contagem += str(numero) + " "
    return texto_da_contagem

app.run(port=5000, debug=True)
```

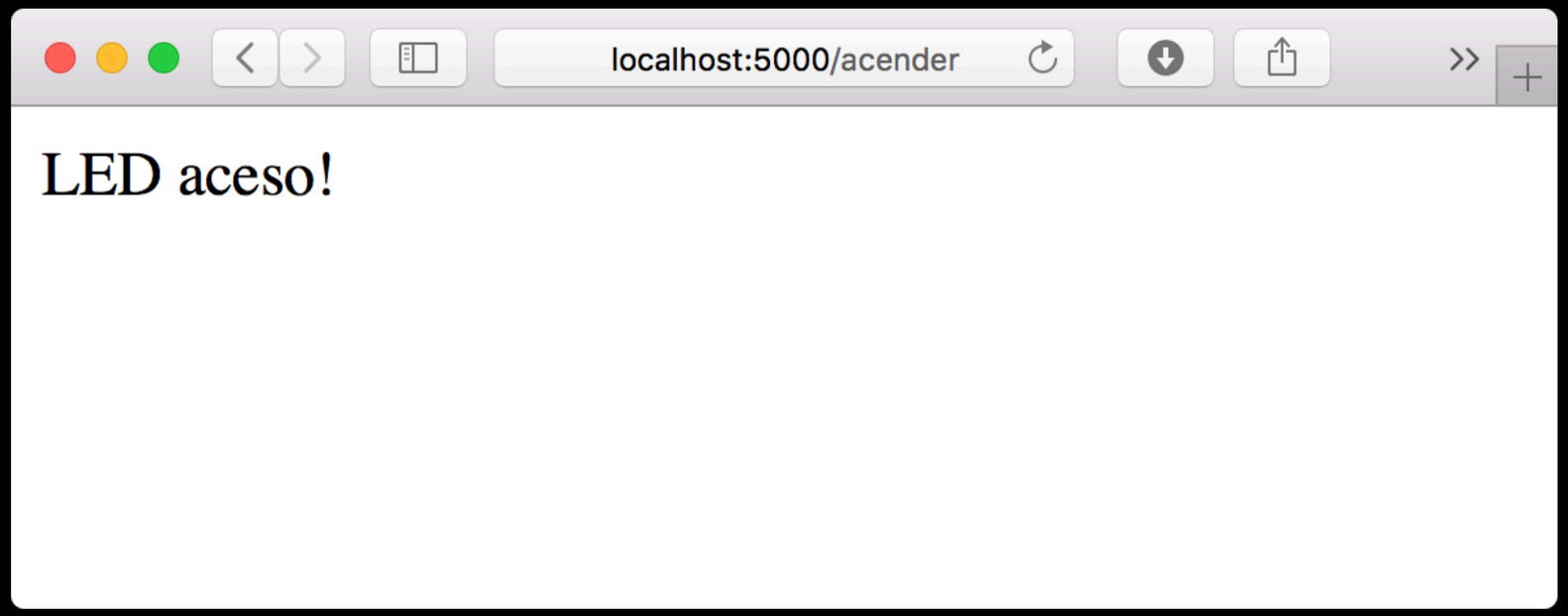
Parâmetro dentro da Rota da Página

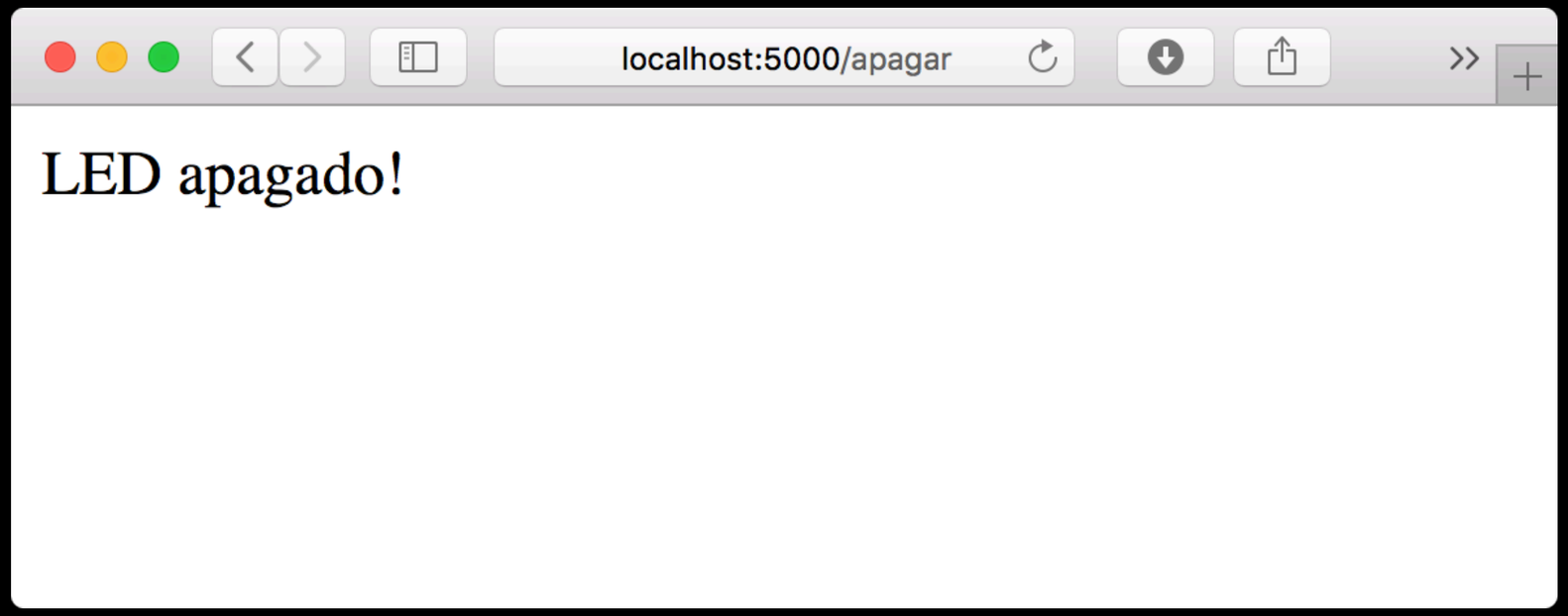



Parâmetro dentro da Rota da Página

```python
from gpiozero import LED
from flask import Flask
app = Flask(__name__)
led = LED(21)

@app.route("/acender")
def acender_led():
    led.on()
    return "LED aceso!"

@app.route("/apagar")
def apagar_led():
    led.off()
    return "LED apagado!"

app.run(port=5000, debug=True)
```

Exemplo de Controle de LEDs pelo Servidor

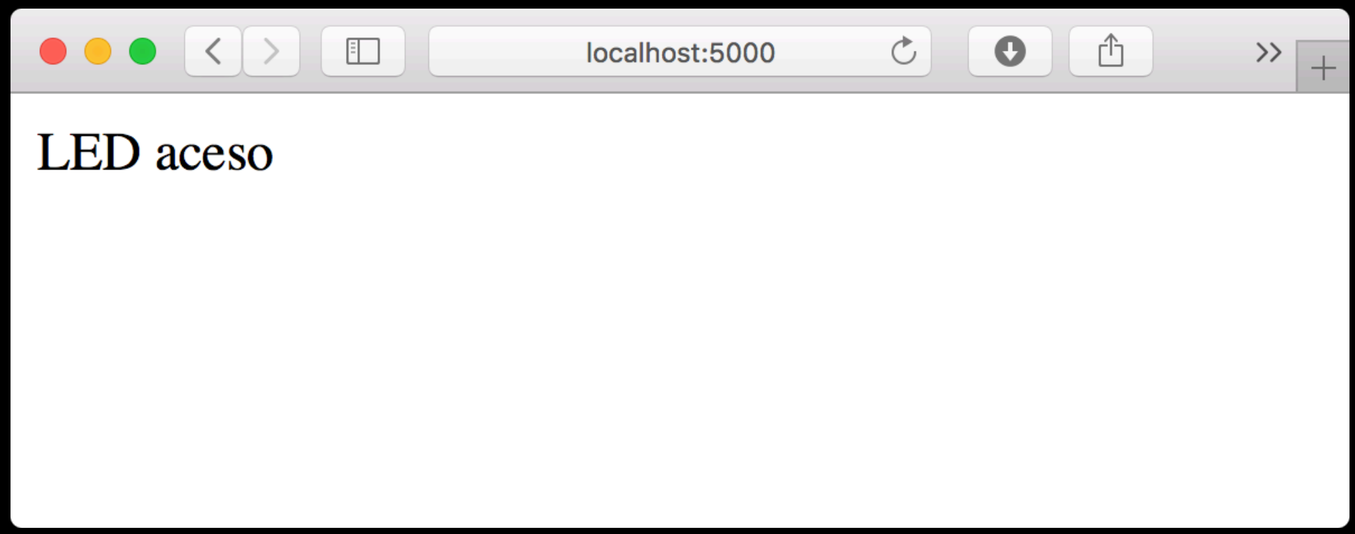


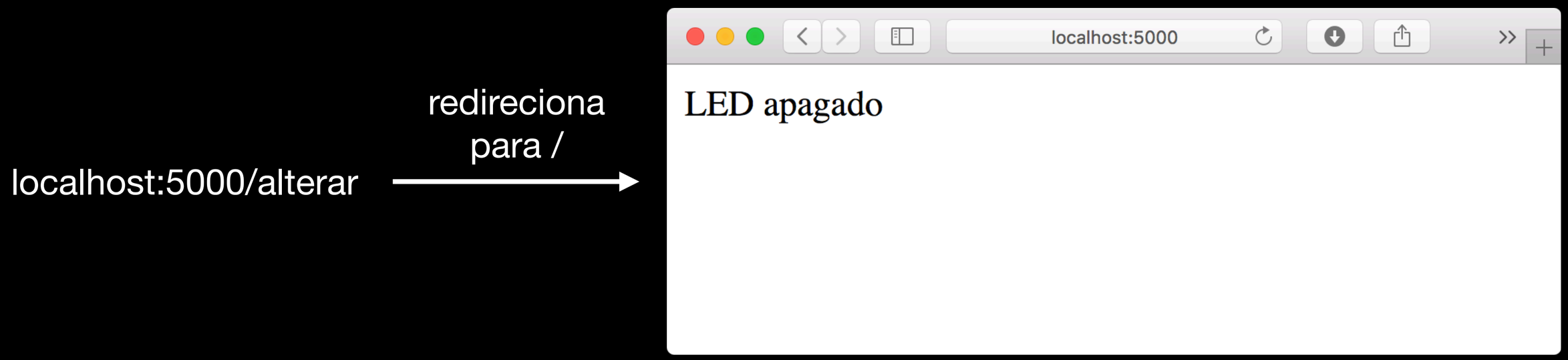




Controle de LED pelas Páginas no Navegador

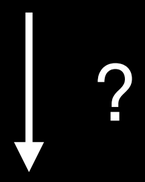
```python
from gpiozero import LED
from flask import Flask, redirect
app = Flask(__name__)
led = LED(21)

@app.route("/")
def pagina_principal():
    if led.is_lit:
        return "LED aceso"
    else:
        return "LED apagado"

@app.route("/alterar")
def alterar():
    led.toggle()
    return redirect("/")

app.run(port=5000, debug=True)
```

Redirecionamento de Página




Redirecionamento de Página

?


# Geração de Páginas com Formatação


Tecnologias para Gerar Páginas na Web

```
150         <!-- Column 2 / Sidebar -->
151         <div class="grid_4">
152
153             <h4>Catagories</h4>
154             <ul class="sidebar">
155                 <li><a href="">So who are we?</a></li>
156                 <li><a href="">Philosophy</a></li>
157                 <li><a href="">History</a></li>
158                 <li><a href="">Jobs</a></li>
159                 <li><a href="">Staff</a></li>
160                 <li><a href="">Clients</a></li>
161             </ul>
162
163         <h4>Latest News</h4>
164         <ul class="sidebar">
165             <li><a href="">Chickens rule the world</a></li>
166             <li><a href="">Chuck Bartowski is back!</a></li>
167             <li><a href="">Aurelius for sale</a></li>
168             <li><a href="">Themeforest goes bust</a></li>
169         </ul>
170
171             <h4>Archives</h4>
172             <ul class="sidebar">
173                 <li><a href="">January 2010</a></li>
174                 <li><a href="">December 2009</a></li>
175                 <li><a href="">Novemeber 2009</a></li>
```

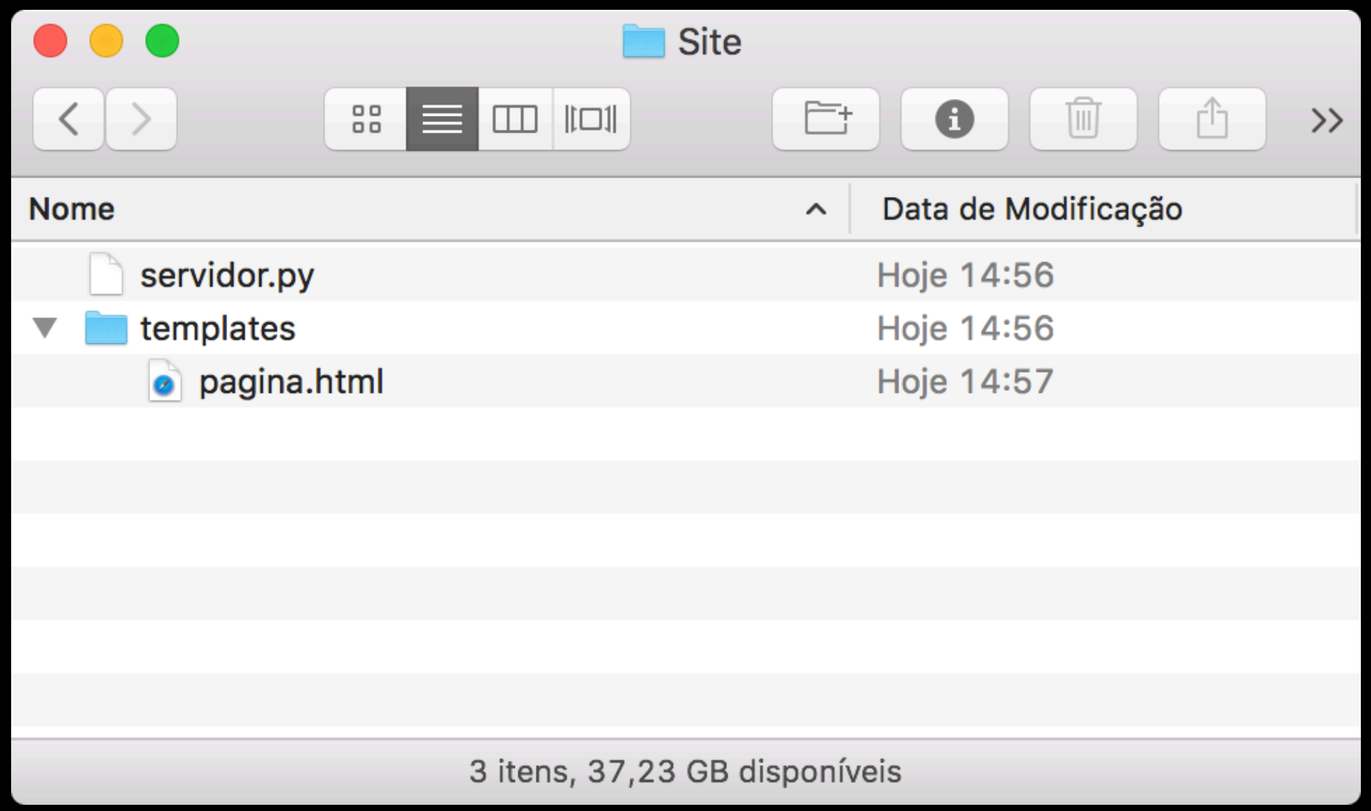
Exemplo de Código Fonte em HTML

```
<elemento>
    conteúdo do elemento
</elemento>
```

Elemento HTML como um Tag

| Código | Resultado |
| --- | --- |
| `<p>Parágrafo 1</p>`<br>`<p>Parágrafo 2</p>` | Parágrafo 1<br>Parágrafo 2 |
| `<strong>Negrito</strong>` | **Negrito** |
| `<img src='https://www.apple.com/ac/flags/1/images/br/32.png'>` |  |
| `<ul>`<br>`  <li>Item 1</li>`<br>`  <li>Item 2</li>`<br>`  <li>Item 3</li>`<br>`</ul>` | • Item 1<br>• Item 2<br>• Item 3 |
| `<a href="/pagina">`<br>`  Link para a Página`<br>`</a>` | Link para a Página |
| `Linha 1 <br> Linha 2` | Linha 1<br>Linha 2 |

Exemplos de Elementos do HTML



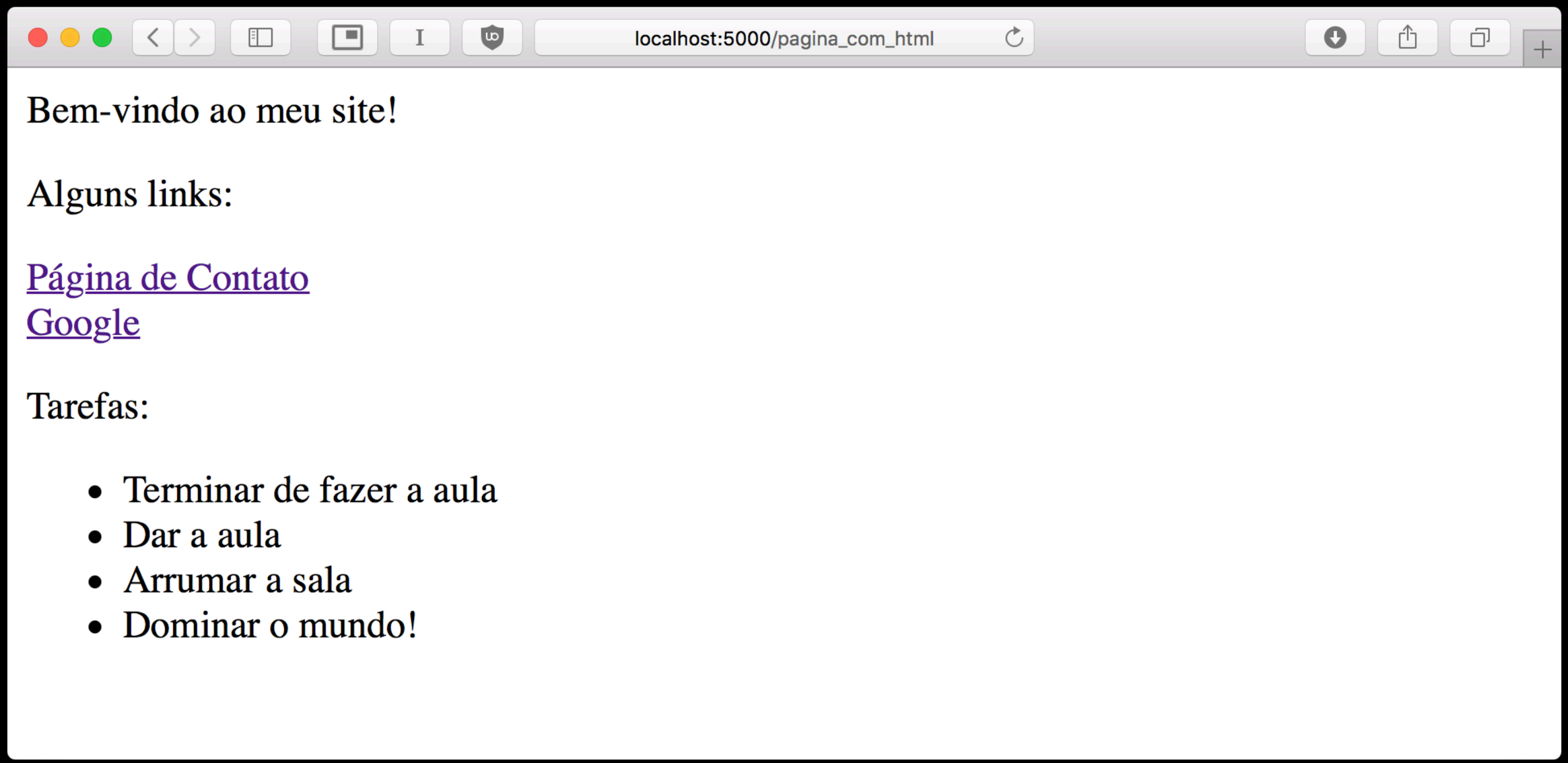

# Organização de Arquivos com o Flask

```python
from flask import Flask, render_template
app = Flask(__name__)

@app.route("/pagina_de_texto")
def mostrar_pagina_de_texto():
    return "Texto, texto, texto..."

@app.route("/pagina_com_html")
def mostrar_pagina_com_html():
    return render_template("pagina.html")

app.run(port=5000, debug=True)
```

Renderização de Página

```html
<p>Bem-vindo ao meu site!</p>

<p>Alguns links:</p>

<a href="/contato">Página de Contato</a>
<br>
<a href="https://www.google.com">Google</a>

<p>Tarefas:</p>
<ul>
    <li>Terminar de fazer a aula</li>
    <li>Dar a aula</li>
    <li>Arrumar a sala</li>
    <li>Dominar o mundo!</li>
</ul>
```

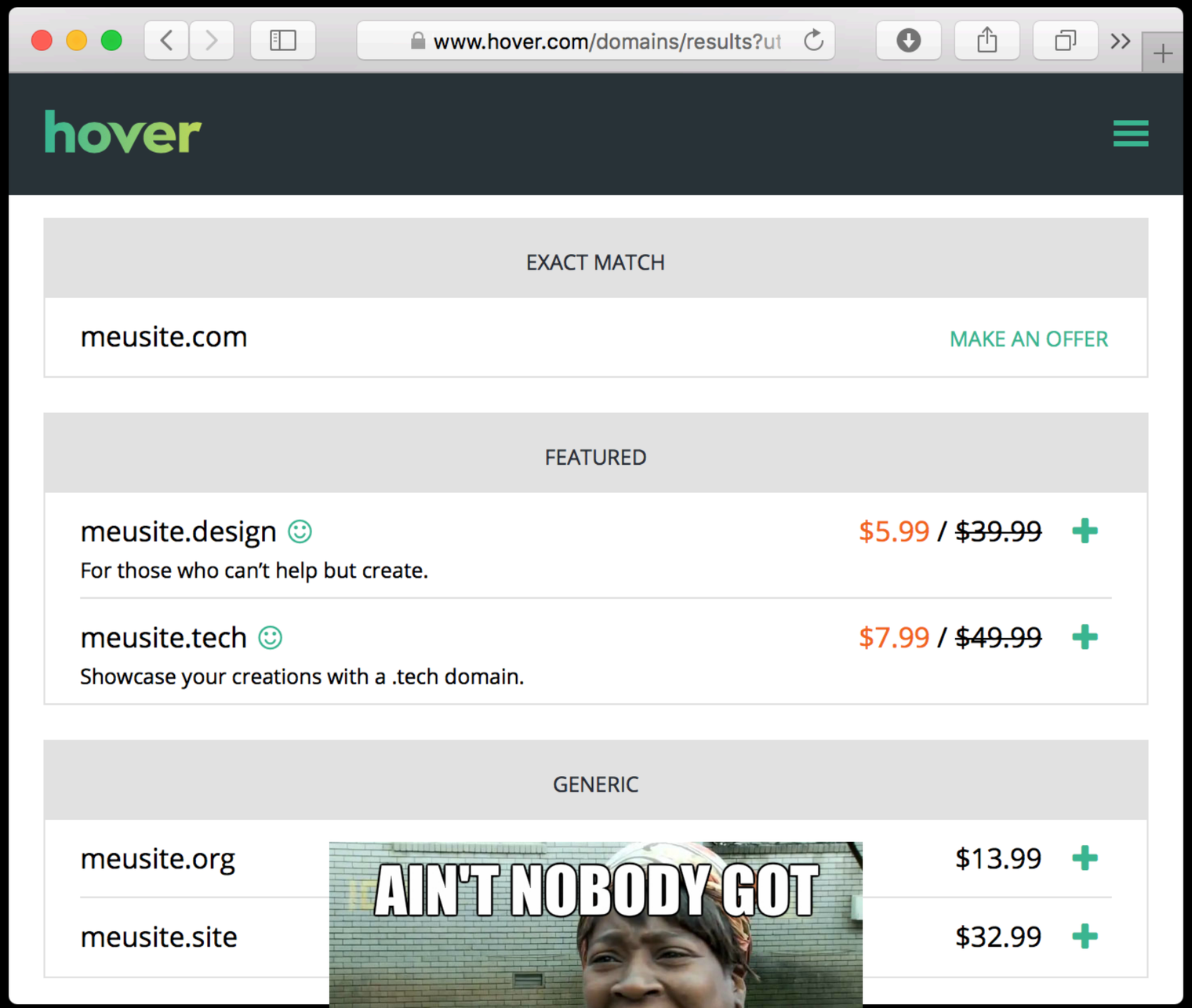
Exemplo de Página Simples




Página Renderizada

Como fazer para o site ficar
acessível de fora do computador?


Acesso do Site sem Usar Localhost?



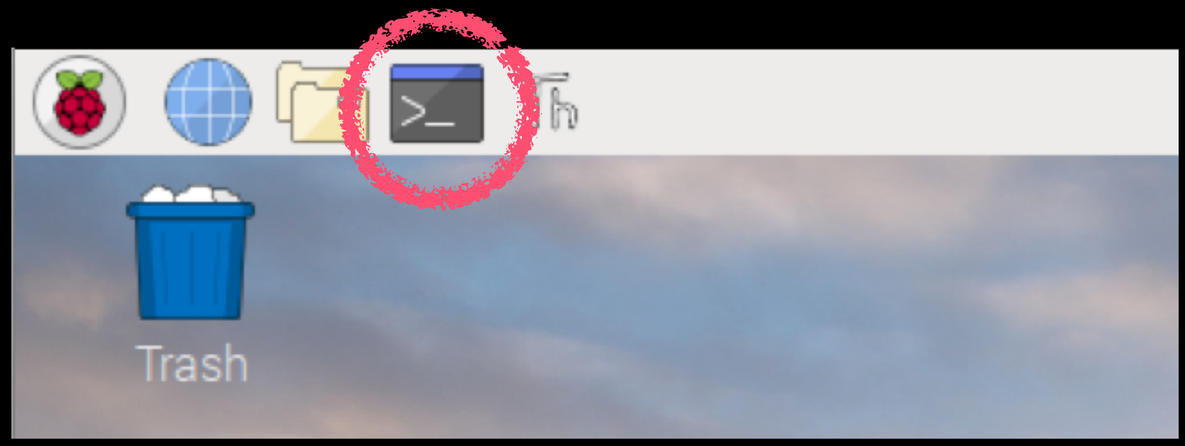




Solução Demorada: Compra + Configuração de um Domínio

# ngrok

"ngrok exposes local servers behind NATs and firewalls to the public internet over secure tunnels."




```
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://21f58849.ngrok.io -> localhost:5000
Forwarding                    https://21f58849.ngrok.io -> localhost:5000

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

Executando o Ngrok no Terminal


Acessando o Site pelo Endereço do Ngrok

# Resumo da Opera

| Funcionalidade | Comandos |
|---|---|
| LED | `from gpiozero import LED • led = LED(porta_da_GPIO)`<br>`led.on() • led.off() • led.toggle() • led.is_lit`<br>`led.blink() • led.blink(n=4, on_time=0.5, off_time=2)`<br>mais funções na documentação oficial |
| Botão | `from gpiozero import Button • botao = Button(porta_da_GPIO)`<br>`botao.is_pressed • led.wait_for_press()`<br>`botao.when_pressed = funcao`<br>`botao.when_held = funcao • botao.when_released = funcao`<br>mais funções na documentação oficial |
| LCD | `from Adafruit_CharLCD import Adafruit_CharLCD`<br>`lcd = Adafruit_CharLCD(2, 3, 4, 5, 6, 7, 16, 2)`<br>`lcd.message("Texto 1\nTexto 2") • lcd.clear()`<br>mais funções no exemplo do repositório oficial |
| MPlayer | `from mplayer import Player • player = Player()`<br>`player.loadfile("Musica.mp3") • player.loadlist("lista.txt")`<br>`player.pause() • player.paused • player.quit()`<br>`player.time_pos = 2 • player.duration • player.pt_step(-1)`<br>`player.metadata["Title"] • player.metadata["Artist"]`<br>`player.volume = 70 • player.speed = 2` |

Resumo para o Projeto 01

| Funcionalidade | Comandos |
|---|---|
| Entrada / Saída | `x = input('Digite um número: ') • print('Resultado: ', x)` |
| Listas | `lista = [1, 2, 3] • lista2 = ['texto', [0, 0], 5];`<br>`lista[0] • lista.append(elemento) • lista.remove(indice)`<br>mais funções na documentação oficial |
| Dicionários | `dicionario = {'chave 1': 42, 'chave 2': [1, 2, 3], ...}`<br>`dicionario['chave 1'] • dicionario['chave 3'] = 'Olá!'`<br>mais funções na documentação oficial |
| Textos | `x = 'texto' • y = "outro texto" • 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):
    ...
```

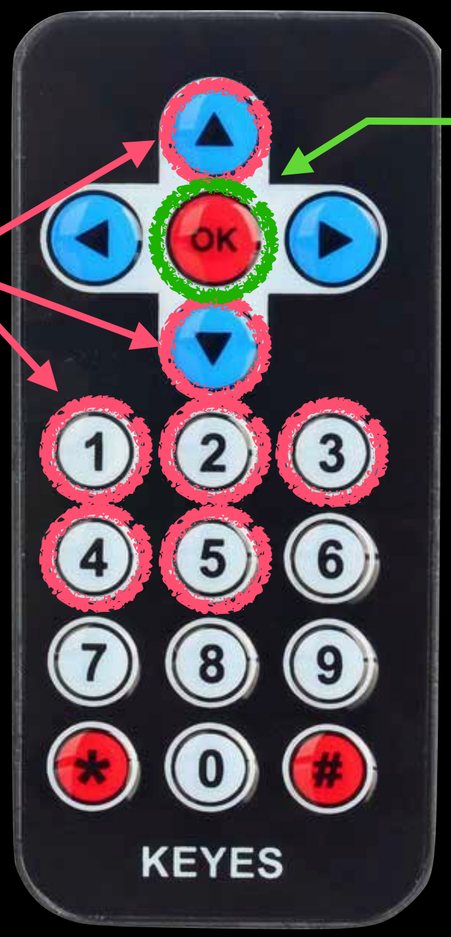
```
def funcao3():
    ...
```

# Prática


janks.link/micro/projeto02.zip

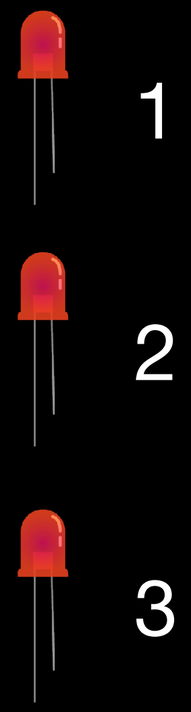
Material do Projeto 02

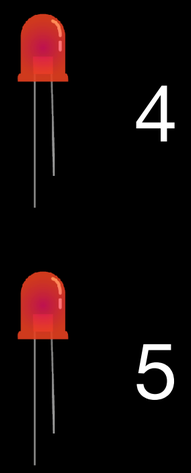
# Testes Iniciais


Seleciona um LED

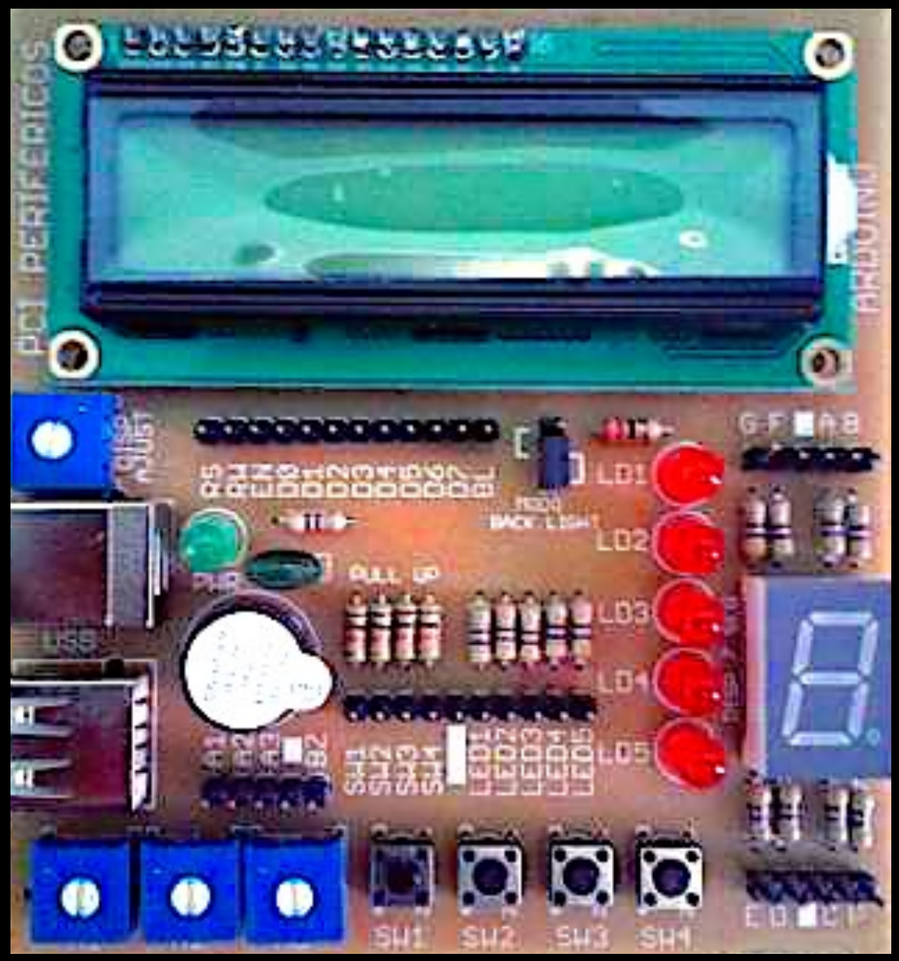
Acende/Apaga




Controle Remoto de LEDs

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




GPIO 21
GPIO 22
GPIO 23
GPIO 24
GPIO 25

GPIO 11, 12, 13, 14

Conexões dos Botões, LEDs e LCD de Caracters com as Portas da GPIO

"mini"


KEY_UP

KEY_LEFT KEY_OK KEY_RIGHT

KEY_DOWN

| | | |
|---|---|---|
| KEY_1 | KEY_2 | KEY_3 |
| KEY_4 | KEY_5 | KEY_6 |
| KEY_7 | KEY_8 | KEY_9 |
| | KEY_0 | |

Códigos dos Botões do Mini Controle Remoto

Testes Iniciais

Ao apertar um botão numérico, "selecione" um LED de 1 a 5, mostrando o valor no LCD de caracteres.
↳ DICA: use o `while True` com um `if/elif`

Ao apertar a tecla OK, mude o estado (aceso/apagado) do LED selecionado.
↳ DICA: use uma variável para guardar o número do LED selecionado.

Ao apertar as setas para cima/baixo, altere o LED selecionado.

Ao apertar um dos botões da placa, apague todos os LEDs.
↳ DICA: use o `blocking=False` da função `init`.

# Implementação


Controle Remoto de uma TV Digital

"tomate"


KEY_MUTE KEY_POWER

KEY_1 KEY_2 KEY_3 KEY_VOLUMEUP

KEY_4 KEY_5 KEY_6 KEY_VOLUMEDOWN

KEY_7 KEY_8 KEY_9

KEY_0

KEY_EXIT

KEY_UP

KEY_LEFT KEY_ENTER KEY_RIGHT

KEY_DOWN

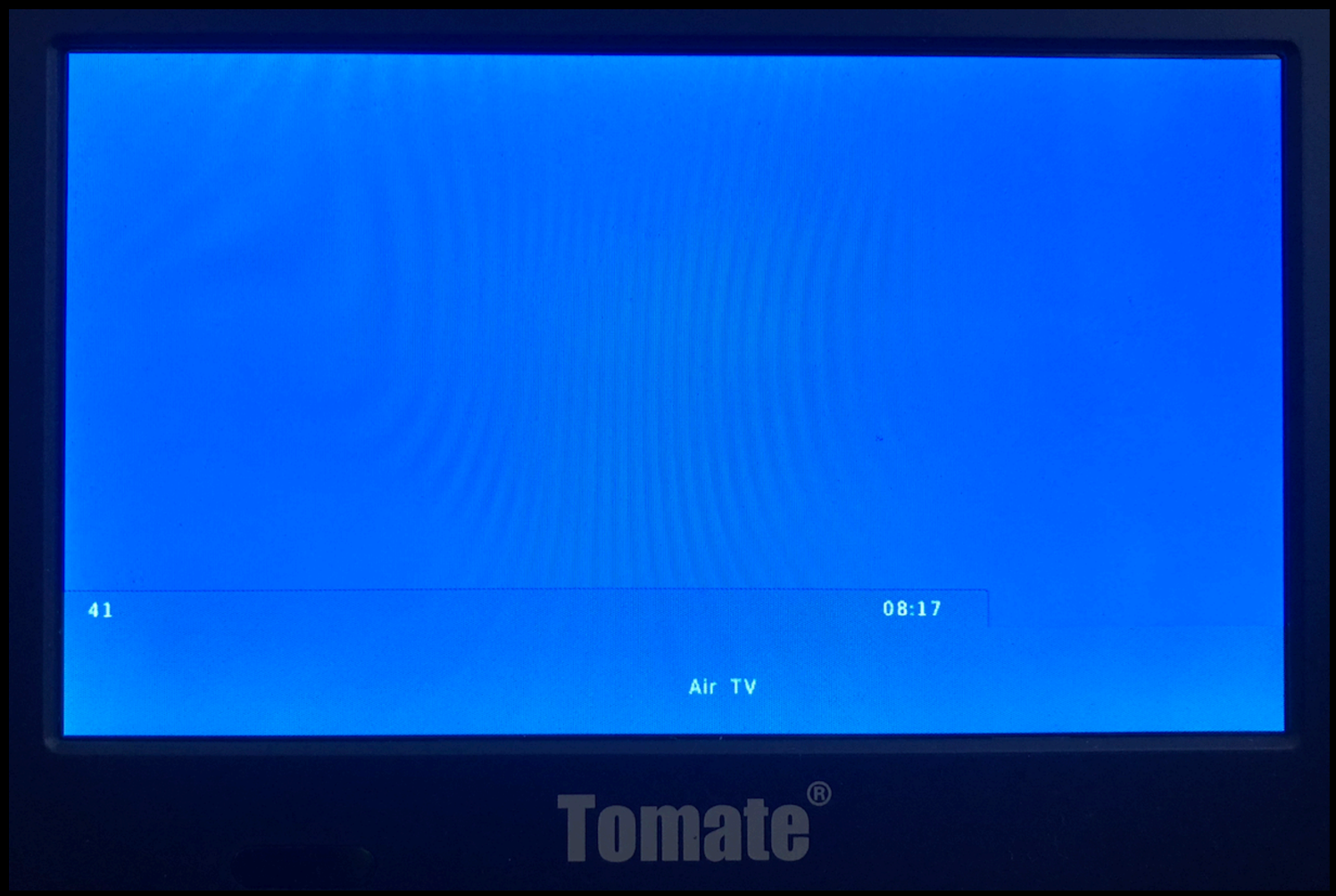
KEY_LIST

Código das Teclas Usadas



| | |
|---|---|
| 2.1 | TV Brasil HD |
| 2.2 | NBR |
| 2.3 | TV Escola |
| 2.4 | Canal Saúde |
| 2.31 | TV Brasil 1seg |
| 4.1 | Globo HD |
| 4.31 | Globo 1seg |
| 7.1 | Band HD |
| 7.31 | Band 1seg |
| 9.1 | CNT HD |
| 9.31 | CNT 1seg |
| 11.1 | SBT HD |
| 11.31 | SBT 1seg |
| 13.1 | Record HD |
| 13.31 | Record 1seg |
| 14.1 | RCI HD |
| 14.31 | RCI 1seg |
| 16.1 | RBI HD |
| 16.31 | RBI 1 seg |
| 20.1 | Canção Nova HD |
| 20.31 | Canção Nova 1 seg |
| 32.1 | TV Universo HD |
| 34.1 | Rede Vida HD |
| 34.31 | Rede Vida 1 seg |

Troca de Canais pelos Botões Numéricos




Tentativa Frustrada de Selecionar um Canal pelo Teclado Numérico


| | |
|---|---|
| 2.1 | TV Brasil HD |
| 2.2 | NBR |
| 2.3 | TV Escola |
| 2.4 | Canal Saúde |
| 2.31 | TV Brasil 1seg |
| 4.1 | Globo HD |
| 4.31 | Globo 1seg |
| 7.1 | Band HD |
| 7.31 | Band 1seg |
| 9.1 | CNT HD |
| 9.31 | CNT 1seg |
| 11.1 | SBT HD |
| 11.31 | SBT 1seg |
| 13.1 | Record HD |
| 13.31 | Record 1seg |
| 14.1 | RCI HD |
| 14.31 | RCI 1seg |
| 16.1 | RBI HD |
| 16.31 | RBI 1 seg |
| 20.1 | Canção Nova HD |
| 20.31 | Canção Nova 1 seg |
| 32.1 | TV Universo HD |
| 34.1 | Rede Vida HD |
| 34.31 | Rede Vida 1 seg |

Troca de Canais pela Listagem

ir para um canal "azul" qualquer
(2 dígitos + ENTER)

esperar um pouco

exibir lista, descer até o canal desejado, ENTER, EXIT

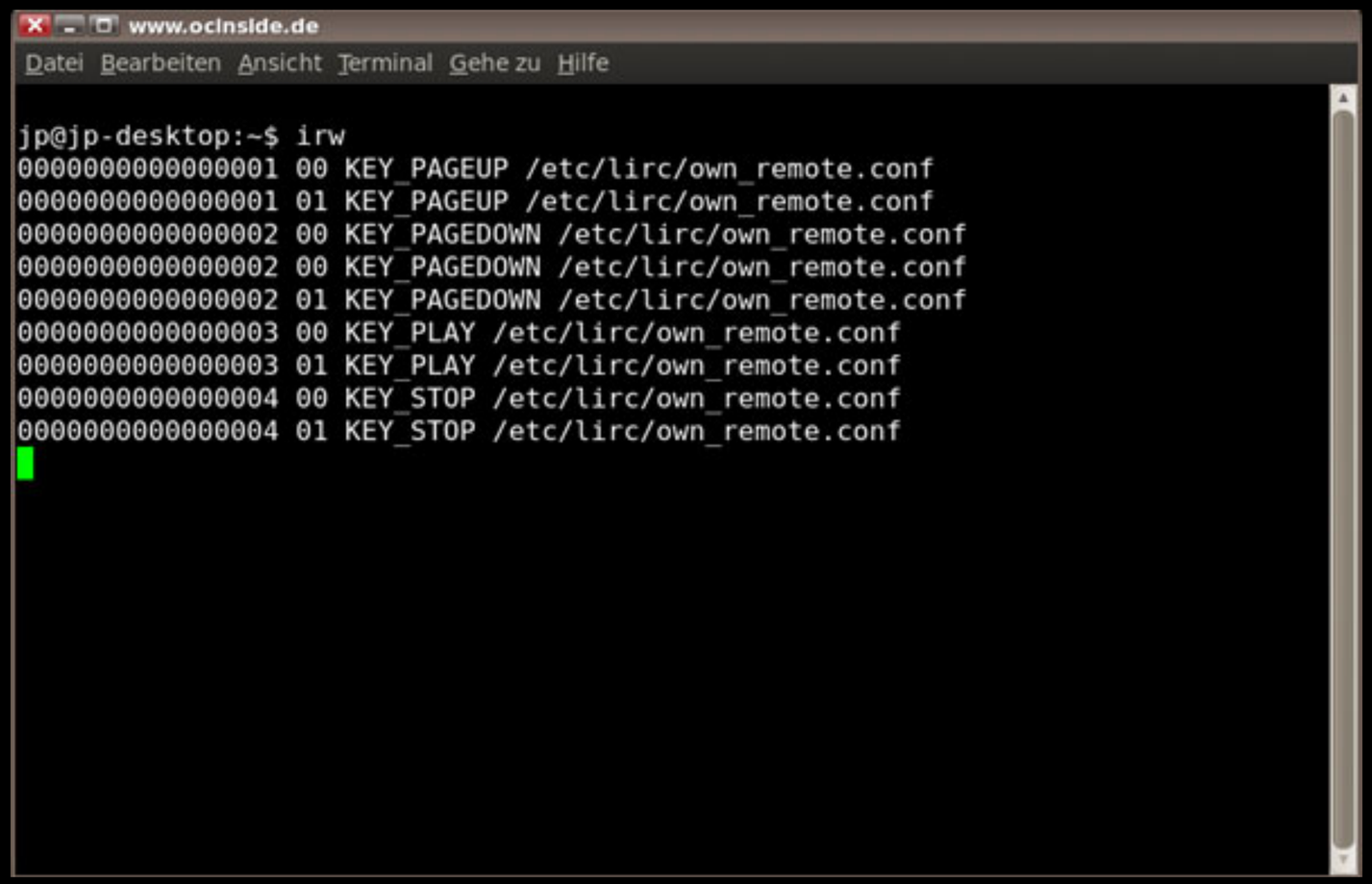
Algoritmo "Gambiarra" para Troca de Canais

Implementação

Crie um servidor com páginas para ligar/desligar a TV, aumentar o volume, diminuir o volume e acionar a função mudo.

Crie uma página com um parâmetro numérico N que desligue a TV após N segundos.
↪ DICA: não use a função `sleep`! Pesquise sobre o objeto `Timer` do Python.

Crie uma página com um parâmetro numérico N no endereço e mude para o N-ésimo canal da lista.
↪ DICA: use a "gambiarra" da listagem.




Monitoramento Opcional com o Comando irw no Terminal

# Aperfeiçoamento


Troca de Canais pelo Celular

Aperfeiçoamento

Crie uma página principal com links para os canais (pelo menos 5) e as outras funcionalidade (volume, dormir, etc).

Redirecione as páginas da Implementação de volta para a página principal.

Use o ngrok para controlar a TV pelo celular.

Otimize o tamanho da página para a tela do celular.
↳ DICA: pesquise no Google sobre a tag HTML `meta viewport`