

Projeto 02

Controle Remoto – Teoria

Jan K. S. – janks@puc-rio.br

ENG 1419 – Programação de Microcontroladores

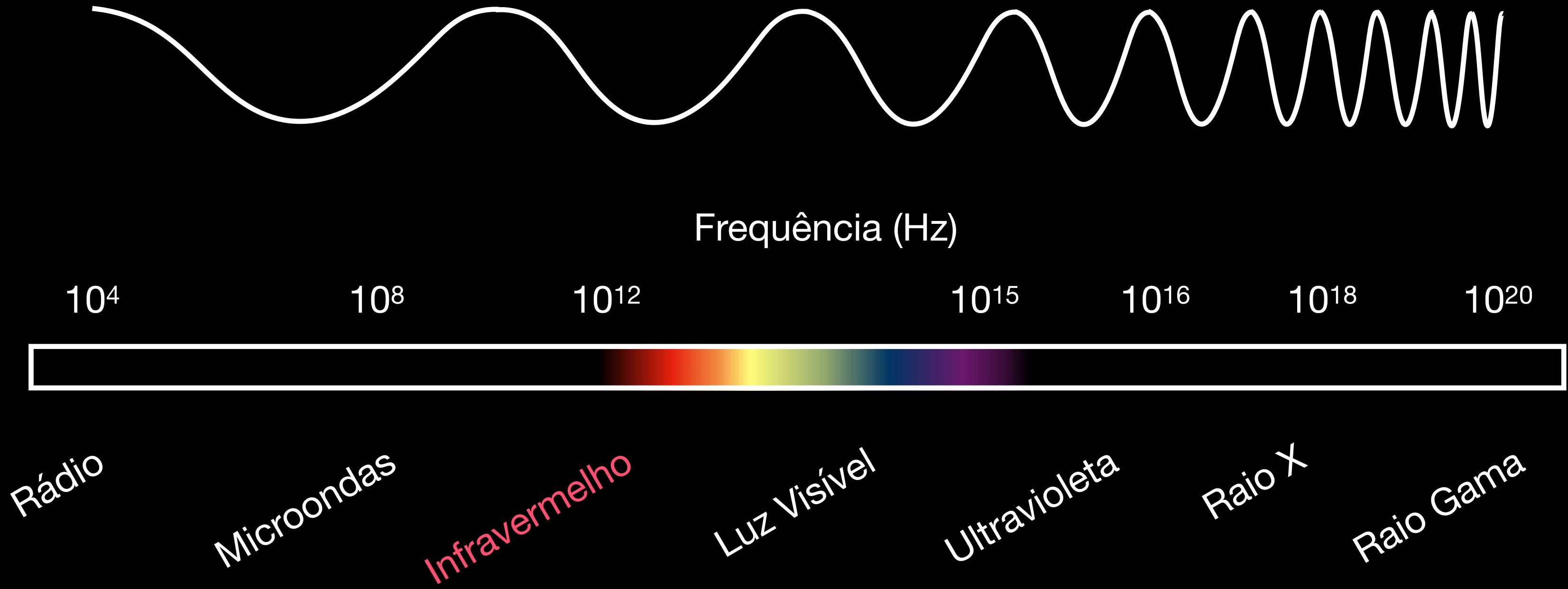
Hardware



Controle Remoto Infravermelho

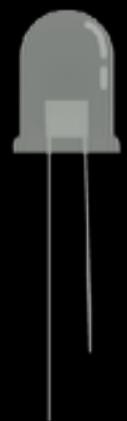


Exemplos de Dispositivos com Controle Remoto Infravermelho

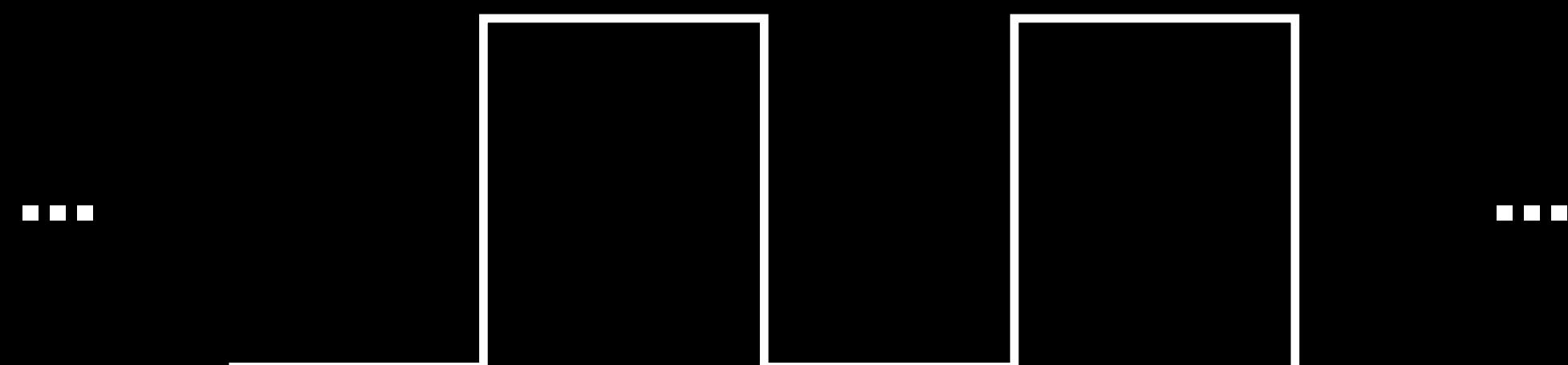
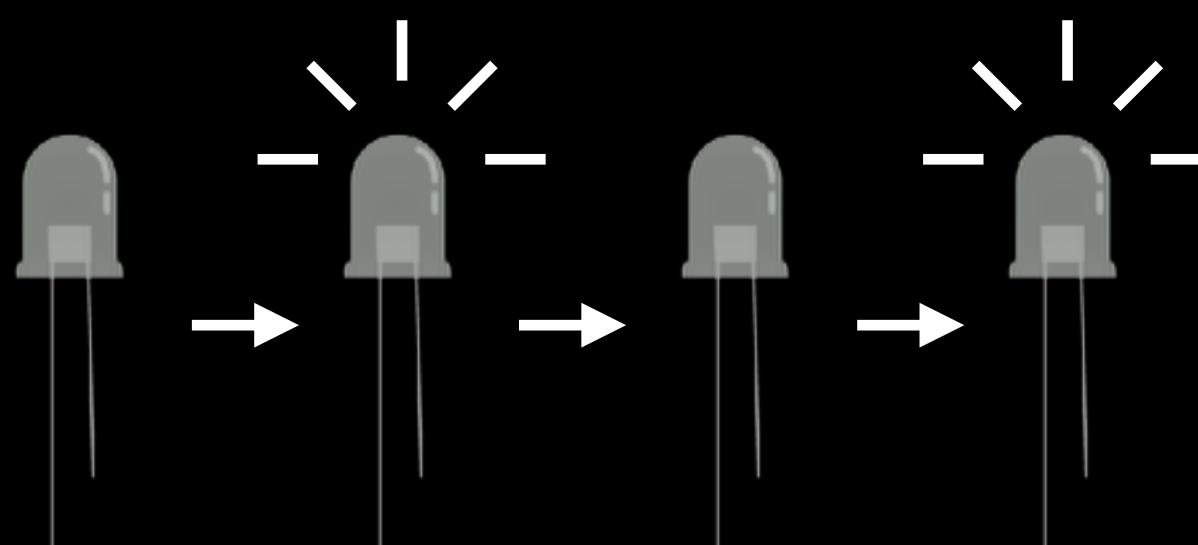
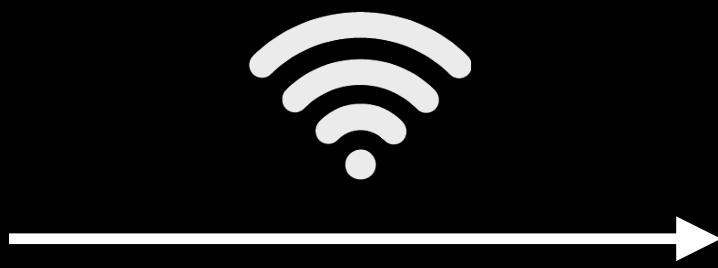
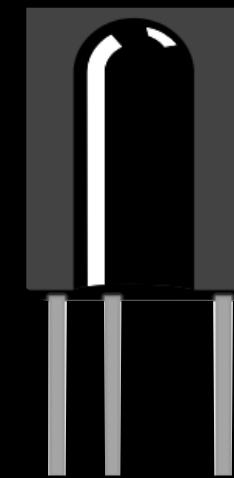


Diferentes Tipos de Onda Eletromagnética

LED Emissor Infravermelho

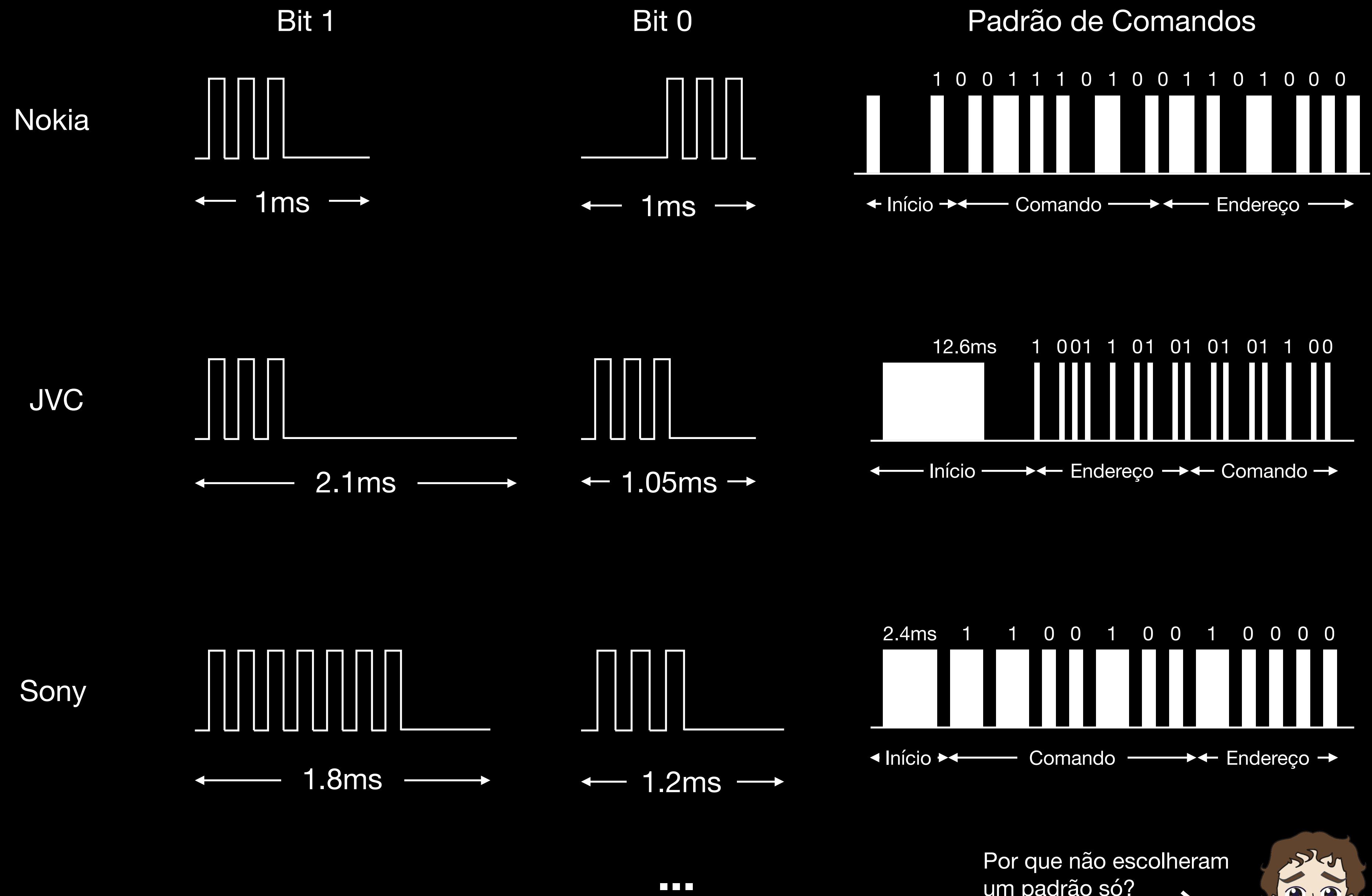


Receptor Infravermelho



Sinal Recebido ao Piscar o LED

Transmissão de Comandos por Infravermelho



Diferentes Tipos de Codificação

Por que não escolheram um padrão só?





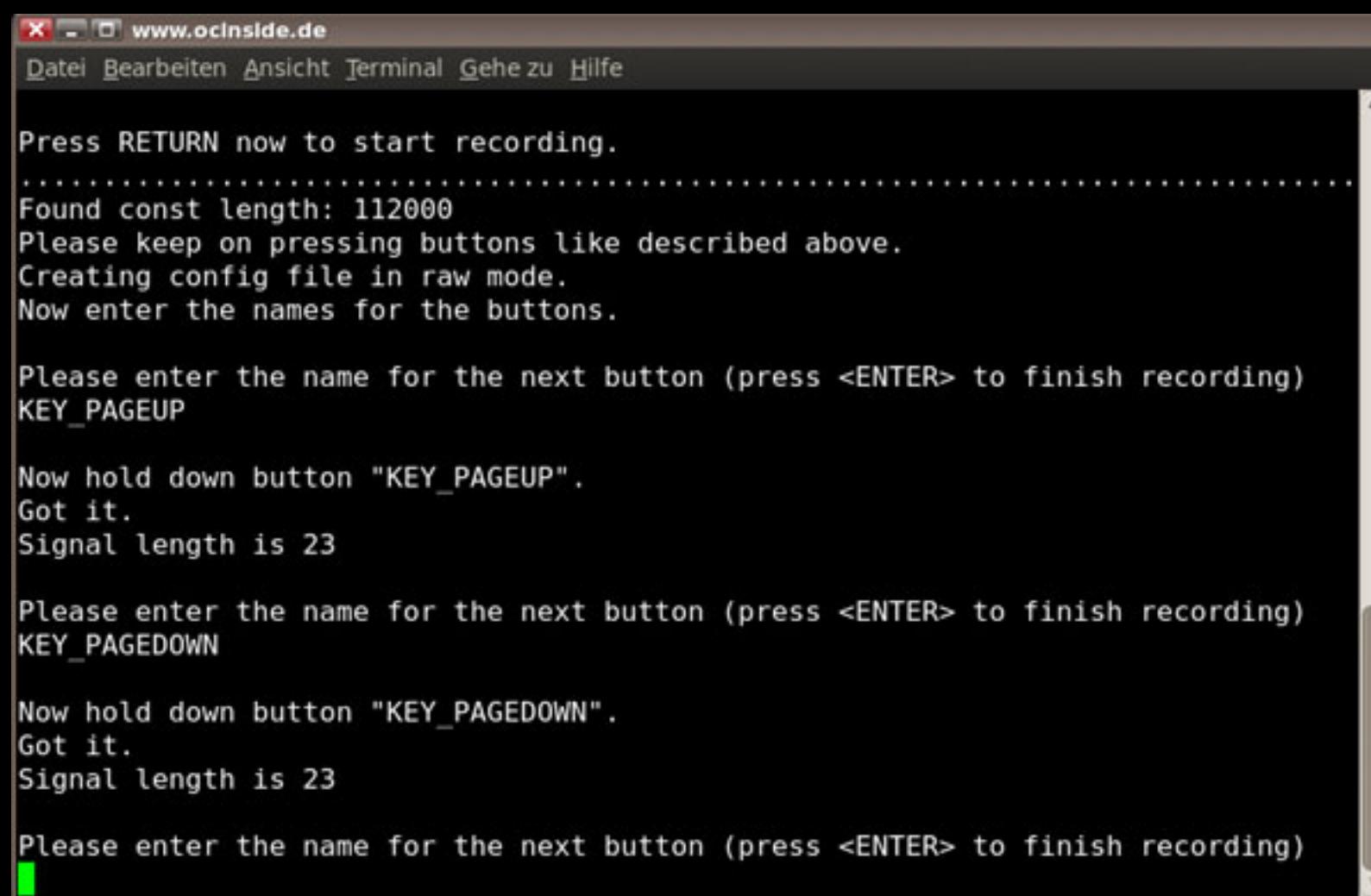
Linux Infrared Remote Control

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: 0000000080bfff10e 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: "aprendizado" sobre 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

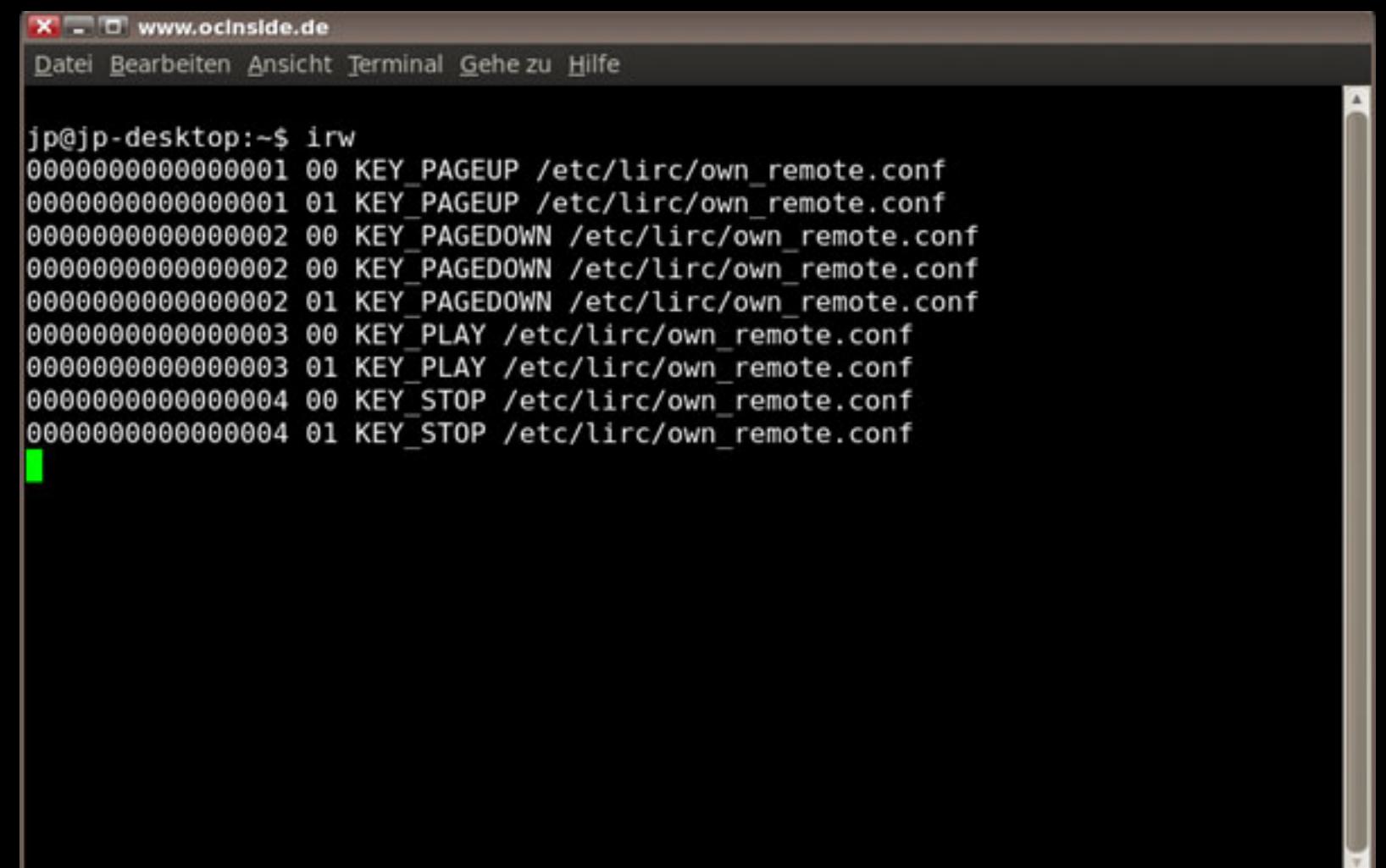
Now hold down button "KEY_PAGEUP".
Got it.
Signal length is 23

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
```

lircd.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=""

```

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
    config = KEY_4
end

```

lirc_options.conf

```

driver = default
device = /dev/lirc0

```

Diversos Arquivos de Configuração do LIRC

The screenshot shows a GitHub repository page for 'ChristopherRogers1991 / python-irsend'. The page includes a navigation bar with icons for file operations, a search bar, and a URL 'GitHub, Inc. github.com/ChristopherRogers1991/python-irsend'. Below the bar are sections for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', and 'Pulse'. A 'README.md' file is listed under the code section. The main content area features a large title 'python-irsend' and a description: 'This is a simple wrapper for [lirc's irsend](#)'. A 'Install:' section contains the command 'pip install py_irsend'. A 'Basic Usage:' section shows Python code examples:

```
>>> 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'])
```

```
>>> from py_irsend.irsend import *
>>> list_remotes()
['mini', 'net', 'aquario']
>>> 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
```

controle remoto "digitou" 1, como
se fosse um teclado



ATENÇÃO: segundo parâmetro
tem que ser uma LISTA de códigos

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

# pode-se também enviar um comando de cada vez,
# caso o dispositivo demore a processar as teclas

>>> from time import sleep
>>> send_once("mini", ["KEY_UP"])
>>> sleep(0.5)
>>> send_once("mini", ["KEY_UP"])
>>> sleep(0.5)
>>> send_once("mini", ["KEY_OK"])
```

The screenshot shows a GitHub repository page for 'tompreston / python-lirc'. The page includes sections for README.md, PyPI, Install, Configure, and Use, each containing code examples.

Install

```
pip3 install python3-lirc
```

Configure

You need a valid [lircrc configuration file](#). For example:

```
$ cat ~/.lircrc
begin
    button = 1          # what button is pressed on the remote
    prog = myprogram   # program to handle this command
    config = one, horse # configs are given to program as list
end

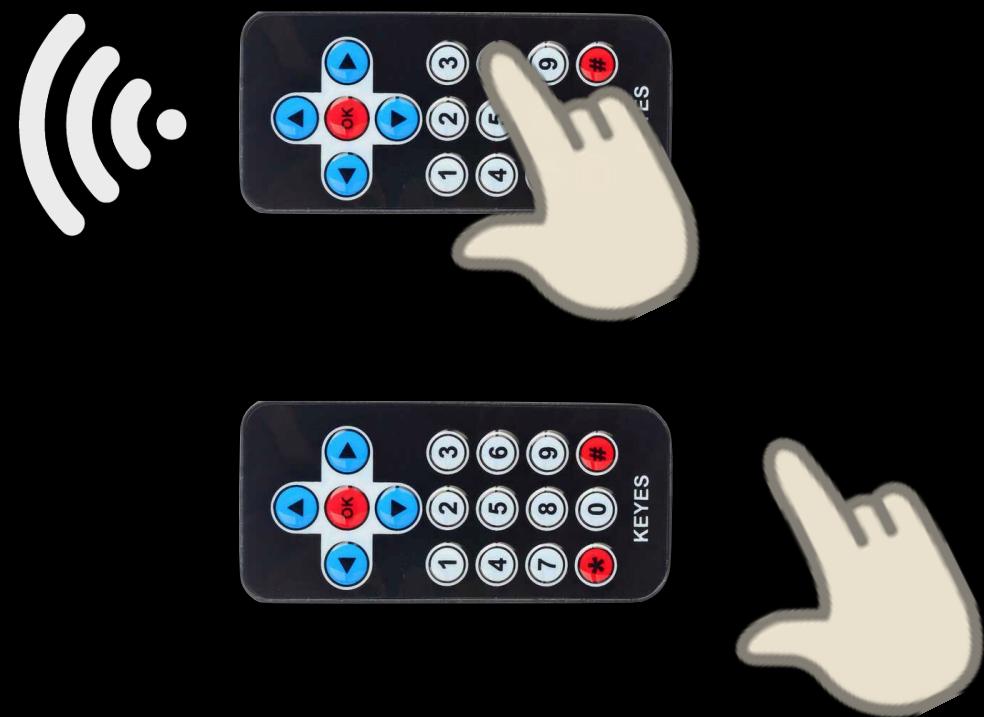
begin
    button = 2
    prog = myprogram
    config = two
end
```

Use

```
$ python3
>>> import lirc
>>> sockid = lirc.init("myprogram")
>>> lirc.nextcode()  # press 1 on remote after this
['one', 'horse']
>>> lirc.deinit()
```

Biblioteca para Receber Sinais pelo LIRC

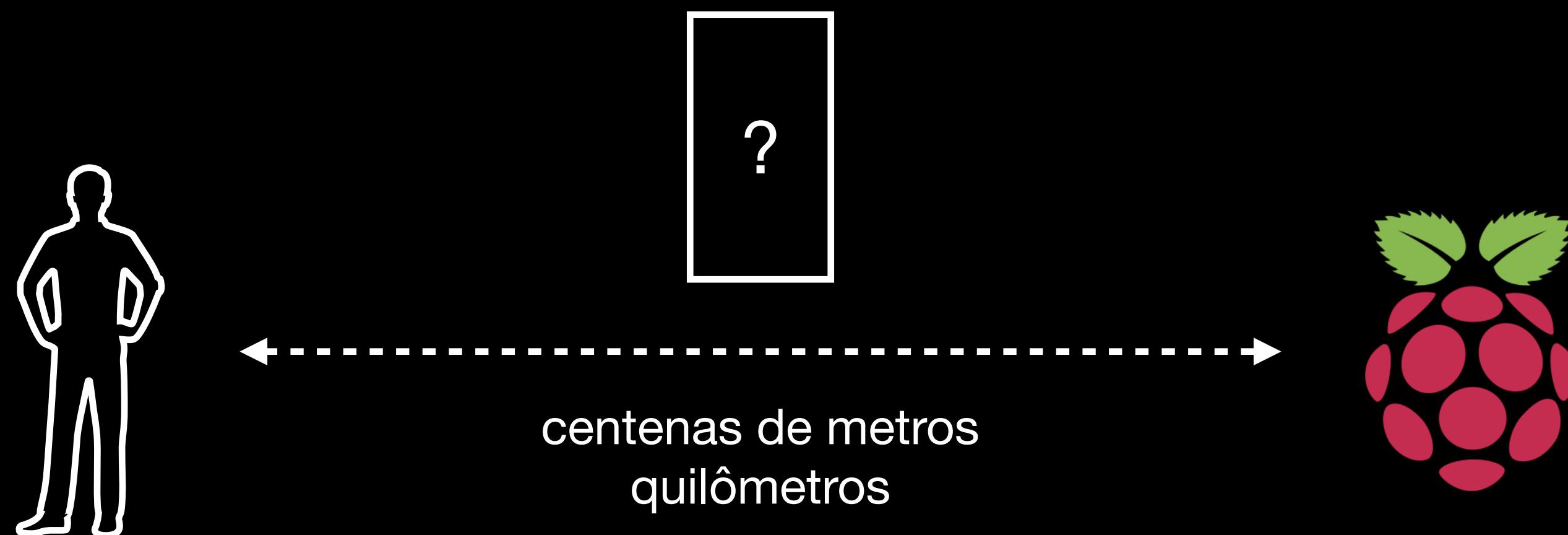
```
>>> from lirc import init, nextcode  
# "aula" é um nome definido na configuração do LIRC  
>>> init("aula", blocking=False)  
>>> nextcode()  
['KEY_6']  
>>> nextcode()  
[]
```



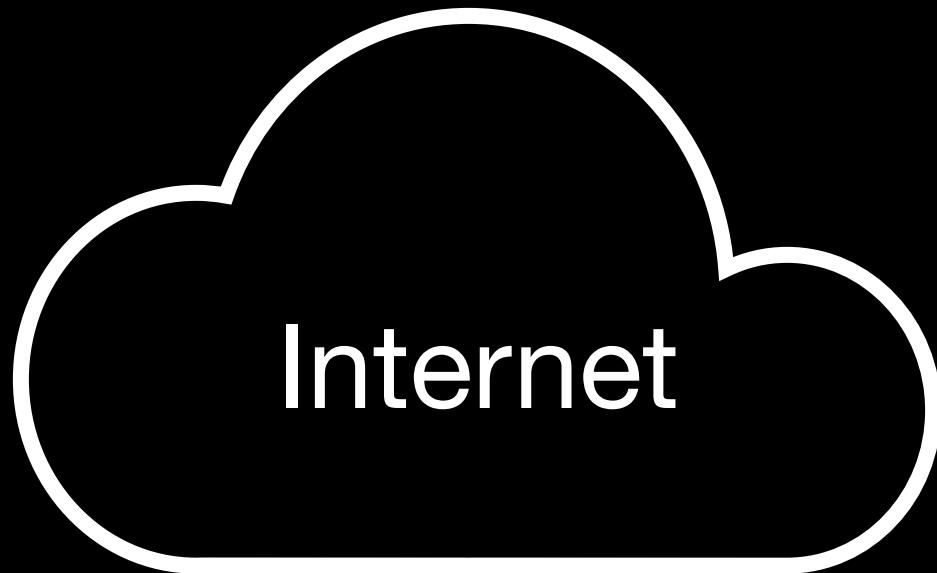
Recebimento de Comandos pelo Emissor Infravermelho

```
>>> from time import sleep
>>> from lirc import init, nextcode
>>> receptor = init("aula", blocking=False)
>>> while True:
...
    lista_com_codigo = nextcode()
...
    if lista_com_codigo != []:
...
        codigo = lista_com_codigo[0]
...
        if codigo == "KEY_LEFT":
...
            print("Tecla esquerda apertada")
...
        elif codigo == "KEY_2":
...
            print("Tecla 2 apertada")
...
        sleep(0.1)
```

Software

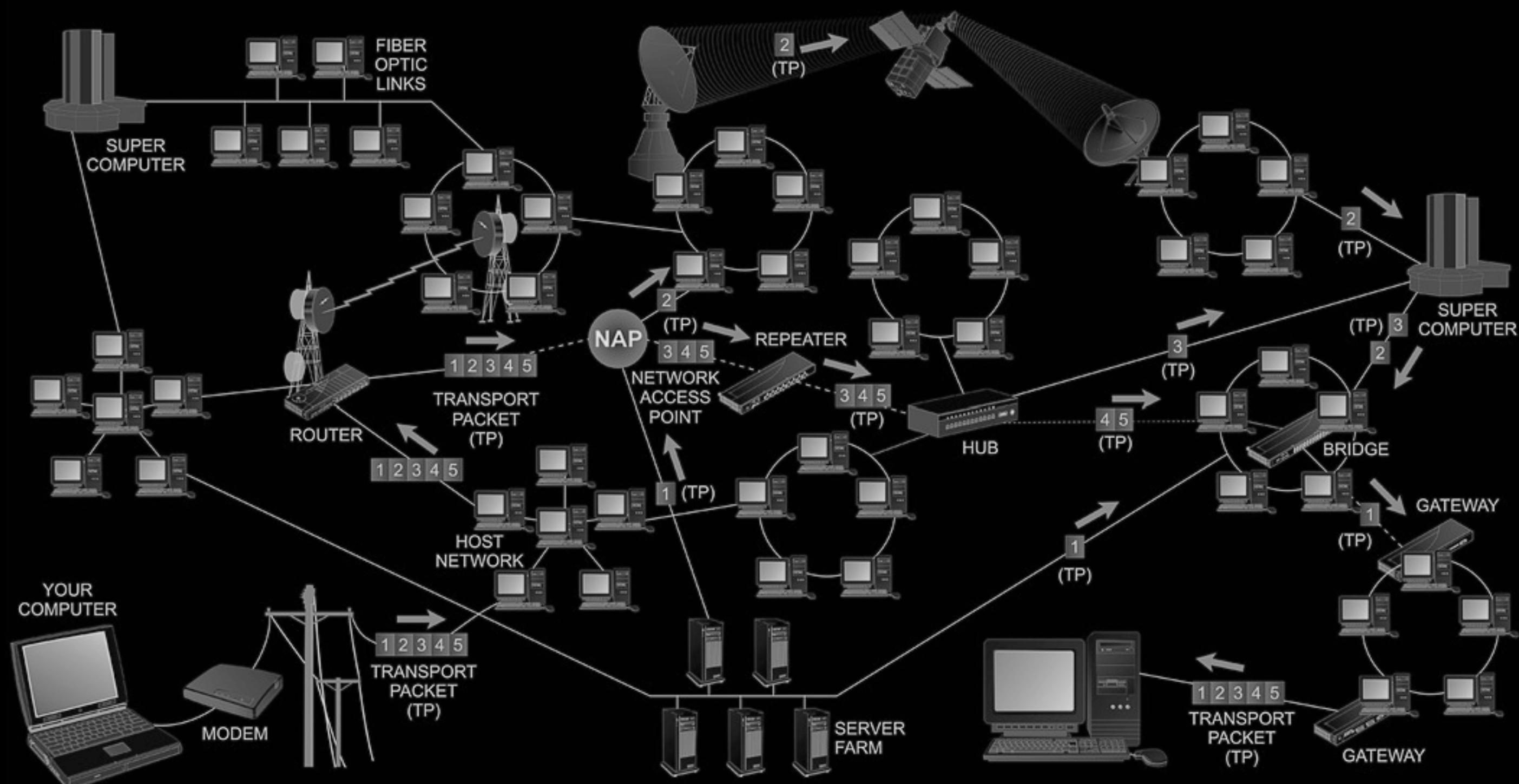


Comunicação Remota a Longas Distâncias

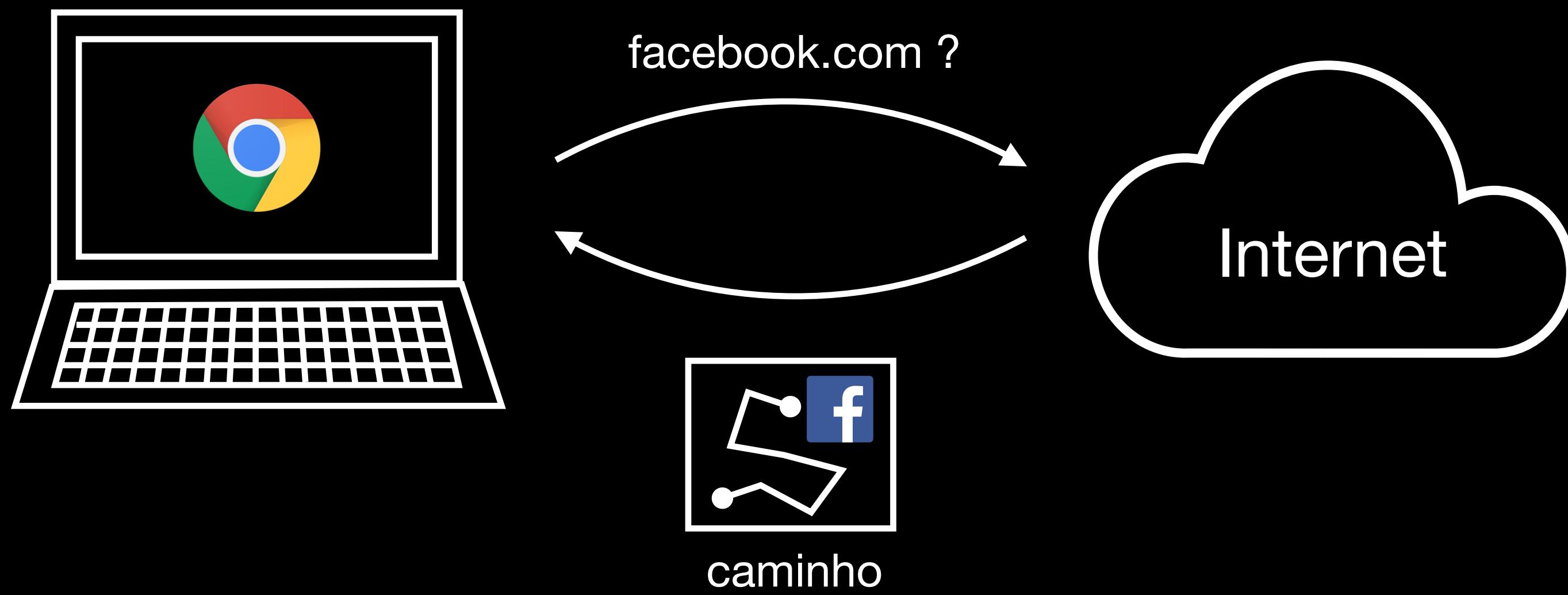


qualquer distância!

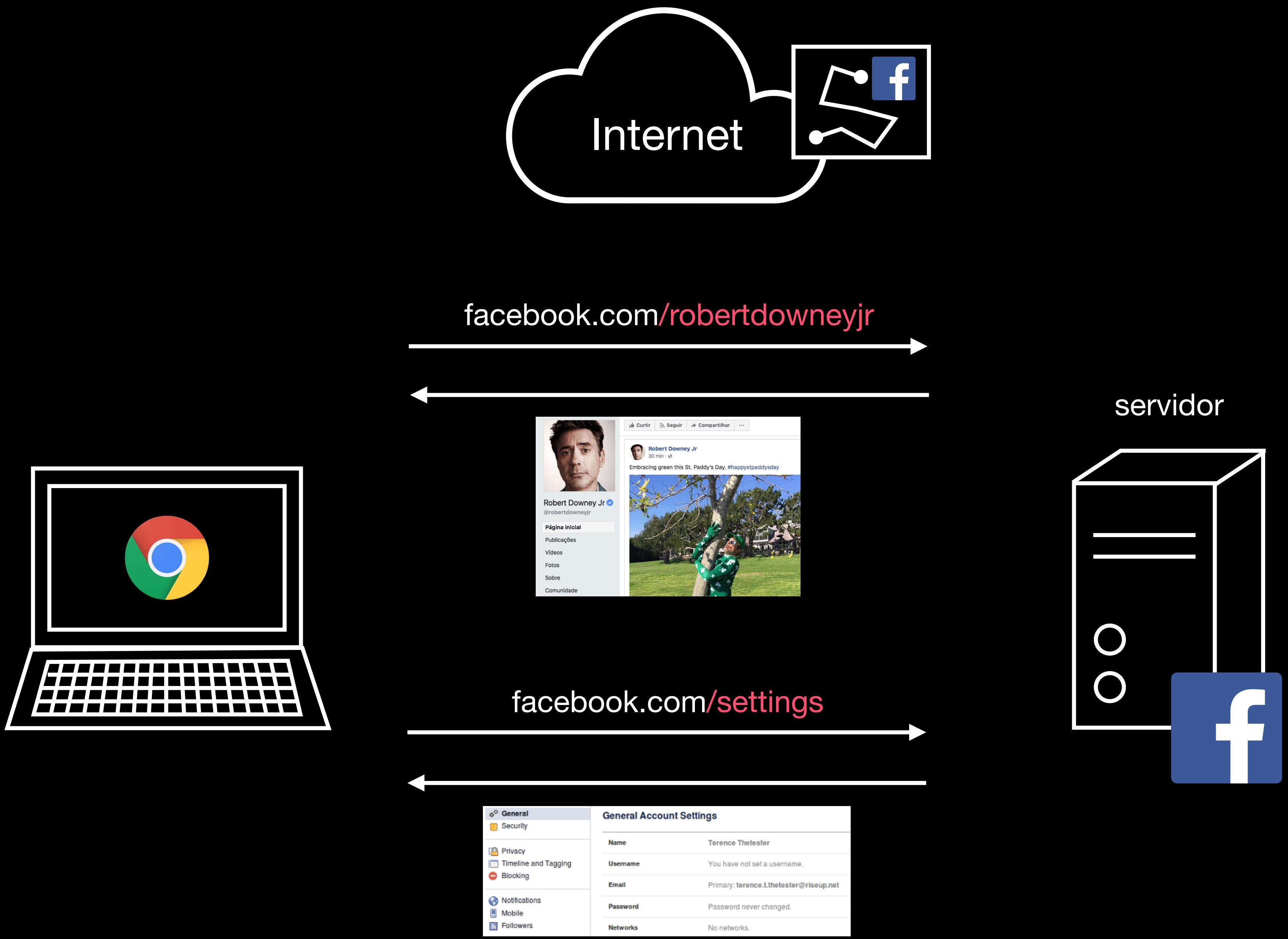
Mecanismo Mágico de Conexão Remota



Rede de Computadores na Internet

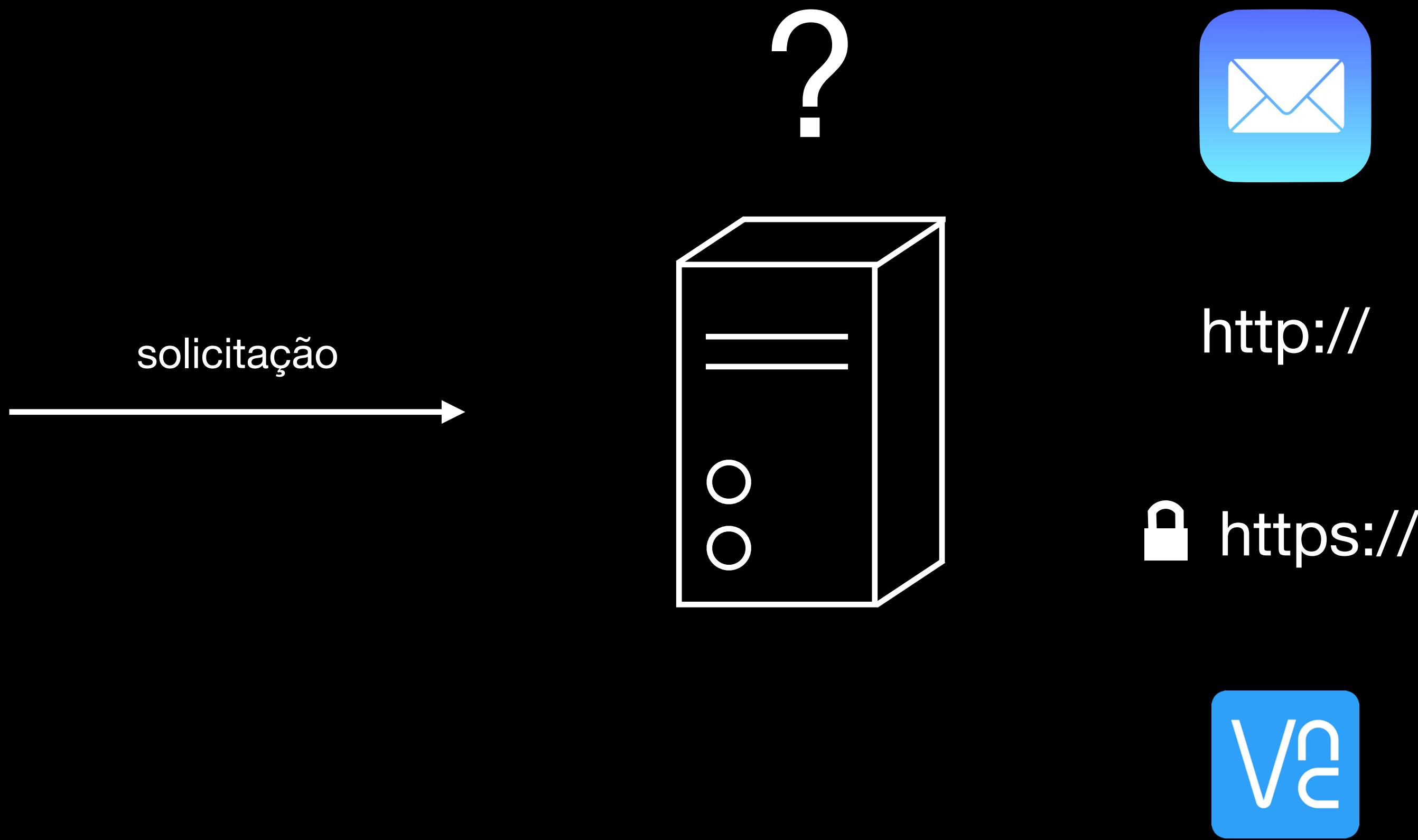


Conexão pela Internet a um Site

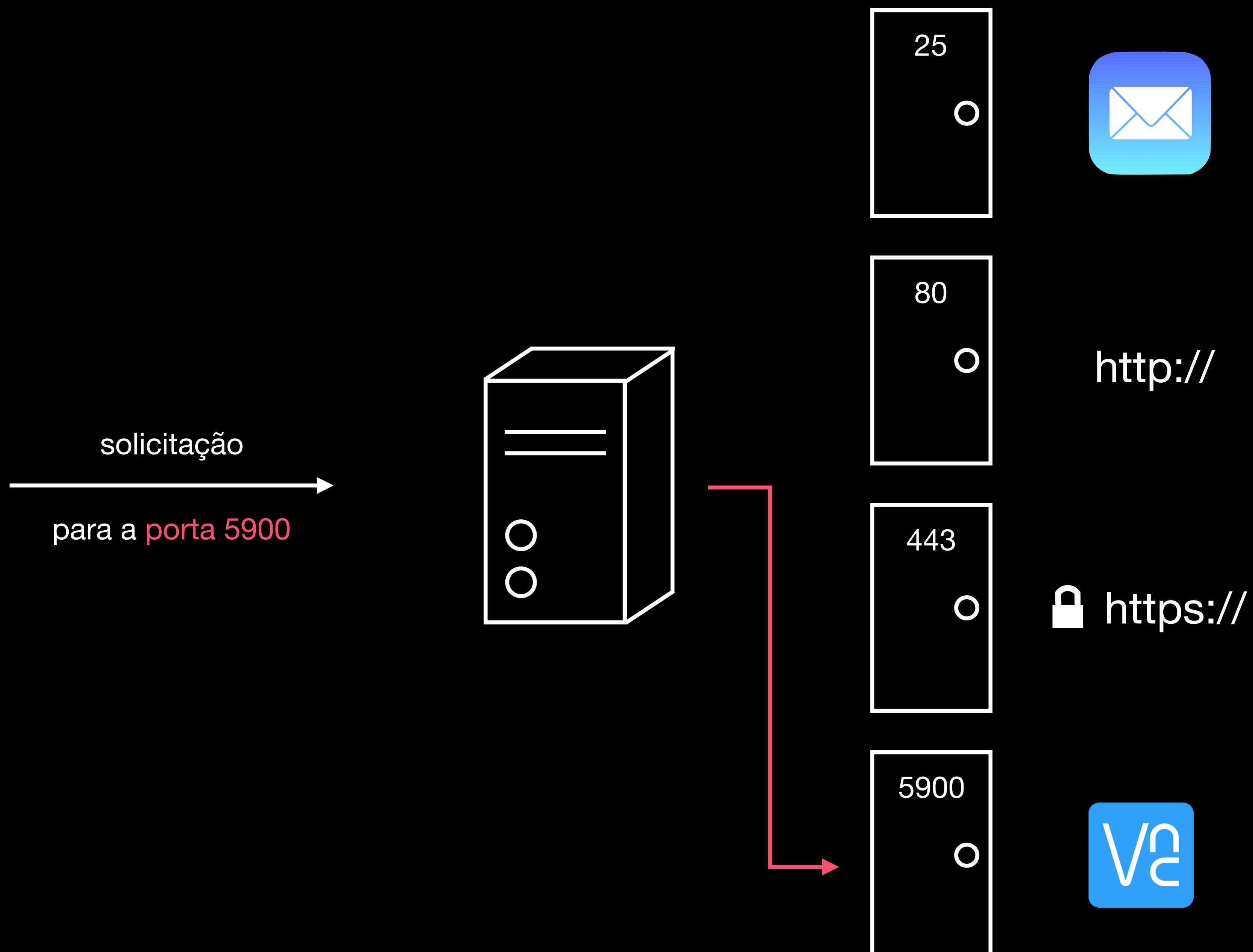


Solicitação de Páginas para Servidores da Internet

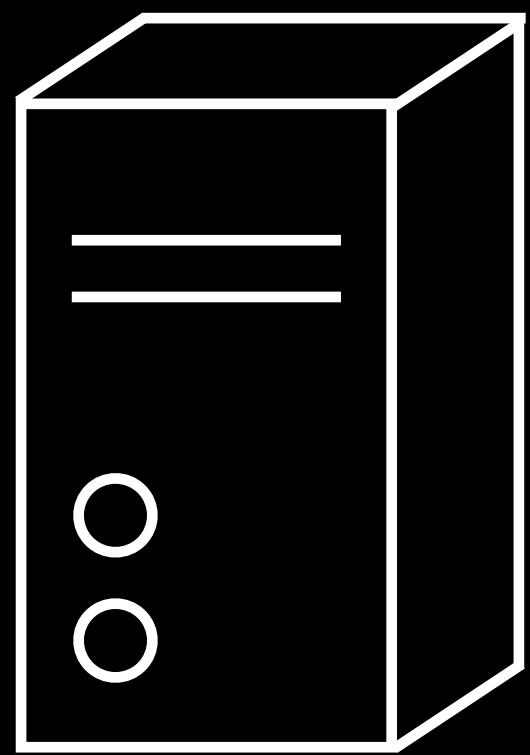
programas rodando
dentro do servidor



Múltiplos Aplicativos Rodando em um Servidor



Comunicação por Portas



=



Exemplos de Computadores para Criar um Servidor



Flask

web development,
one drop at a time

Framework Flask

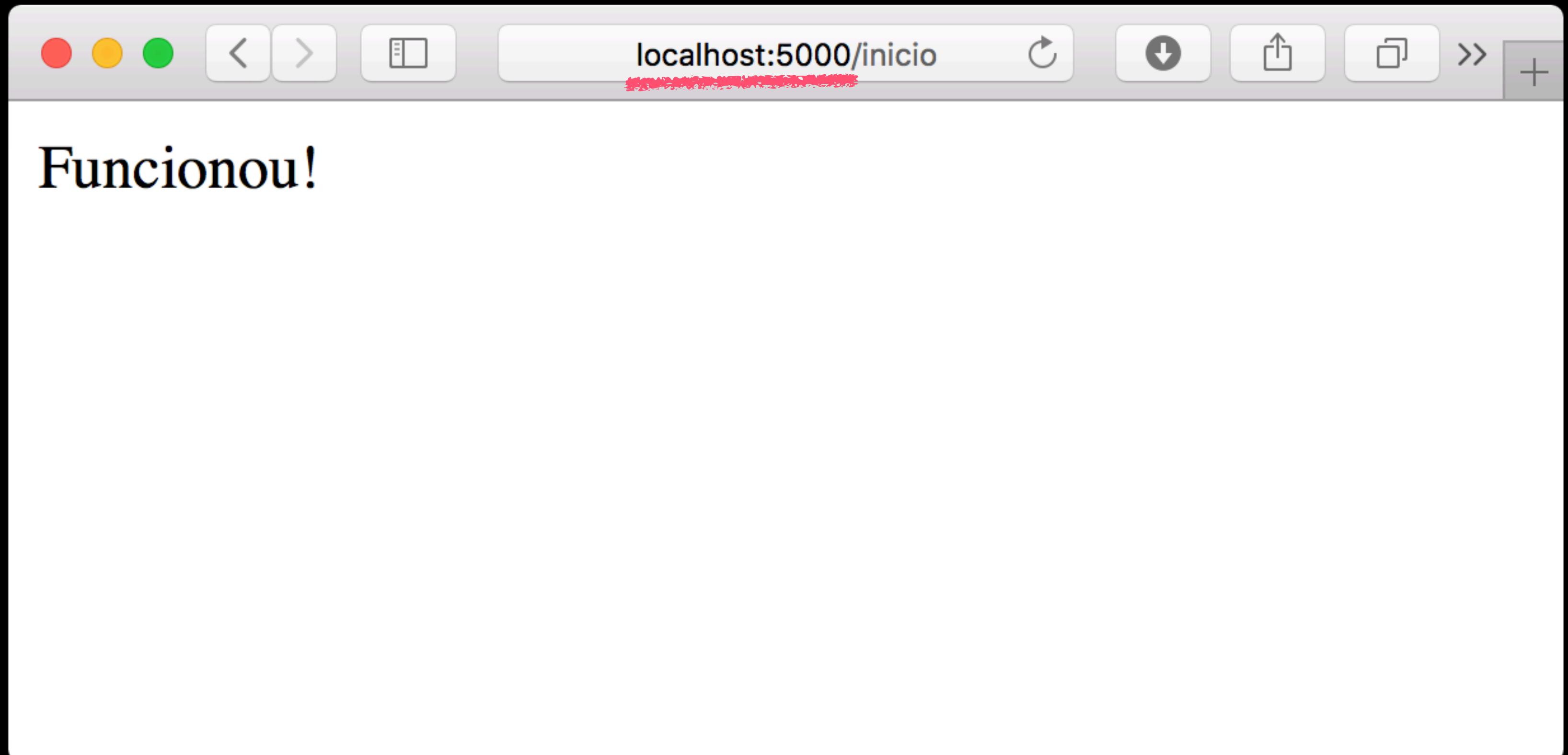
```
from flask import Flask

# initialize servidor com o nome deste arquivo
app = Flask(__name__)

# chame esta função ao acessar a página "inicio"
@app.route("/inicio")
def funcao_da_pagina_inicio():
    return "Funcionou!"

# rode o servidor na porta 5000
app.run(port=5000)
```

procure a página do nosso próprio computador na porta 5000



Carregamento da Página no Navegador

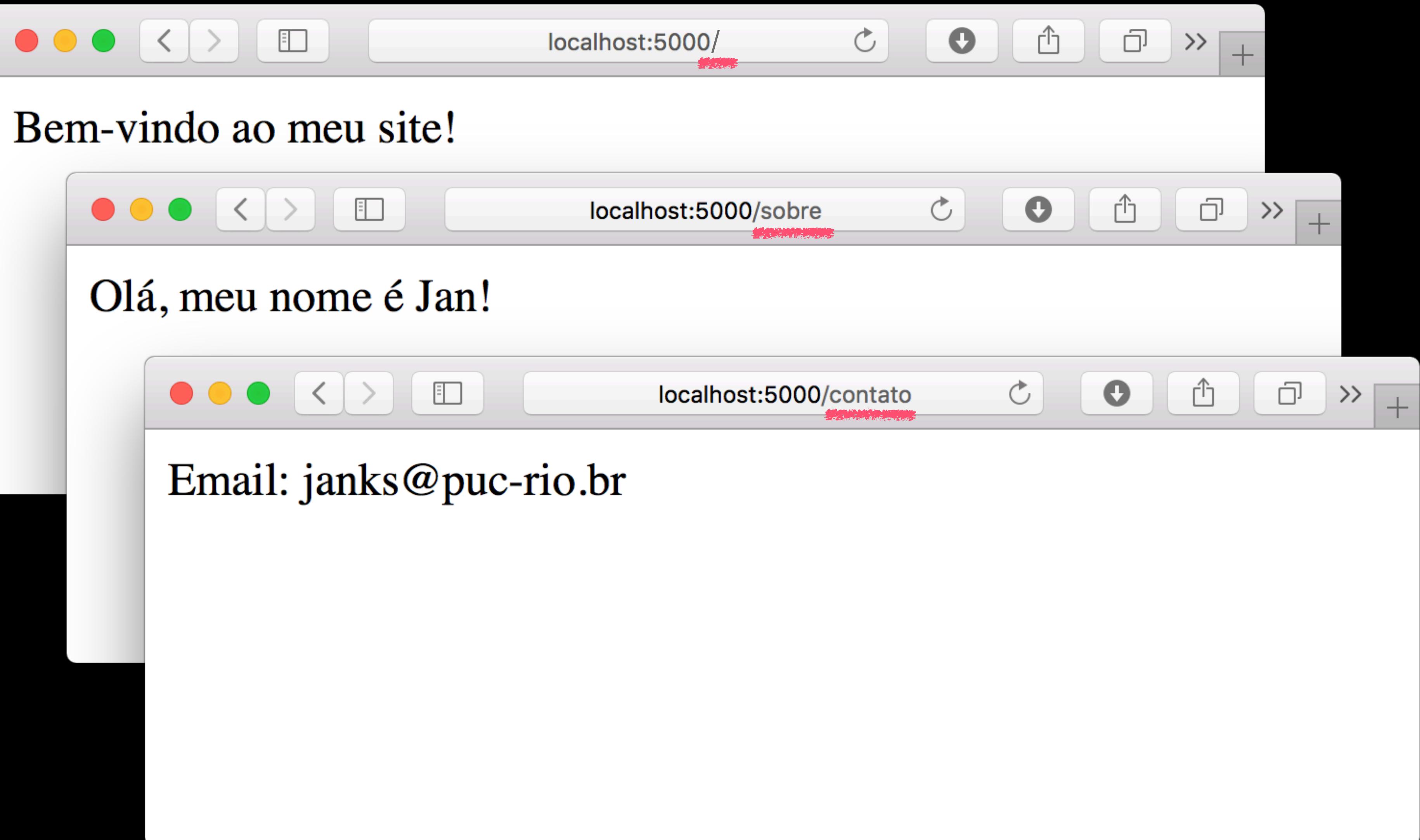
```
from flask import Flask
app = Flask(__name__)

@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)
```



Carregamento das Páginas no Navegador

```
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)
```

The image displays two separate browser windows side-by-side, both showing a sequence of integers from 1 to 300. The top window has a URL of `localhost:5000/sequencia/10`, and the bottom window has a URL of `localhost:5000/sequencia/300`. Both windows have a standard OS X-style title bar with red, yellow, and green buttons, and a toolbar with back, forward, and other navigation icons.

Top Window (localhost:5000/sequencia/10):

```
1 2 3 4 5 6 7 8 9 10
```

Bottom Window (localhost:5000/sequencia/300):

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25  
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46  
47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67  
68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88  
89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106  
107 108 109 110 111 112 113 114 115 116 117 118 119 120 121  
122 123 124 125 126 127 128 129 130 131 132 133 134 135 136  
137 138 139 140 141 142 143 144 145 146 147 148 149 150 151  
152 153 154 155 156 157 158 159 160 161 162 163 164 165 166  
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181
```

Parâmetro dentro da Rota da Página

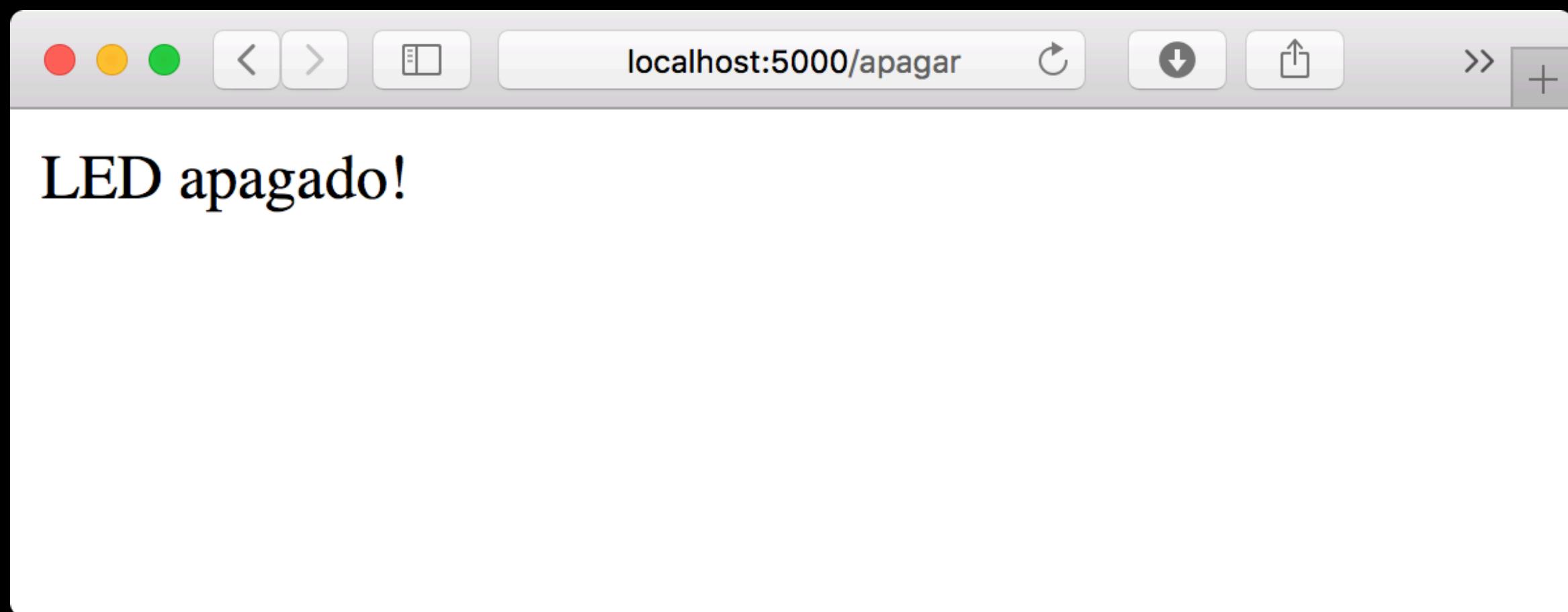
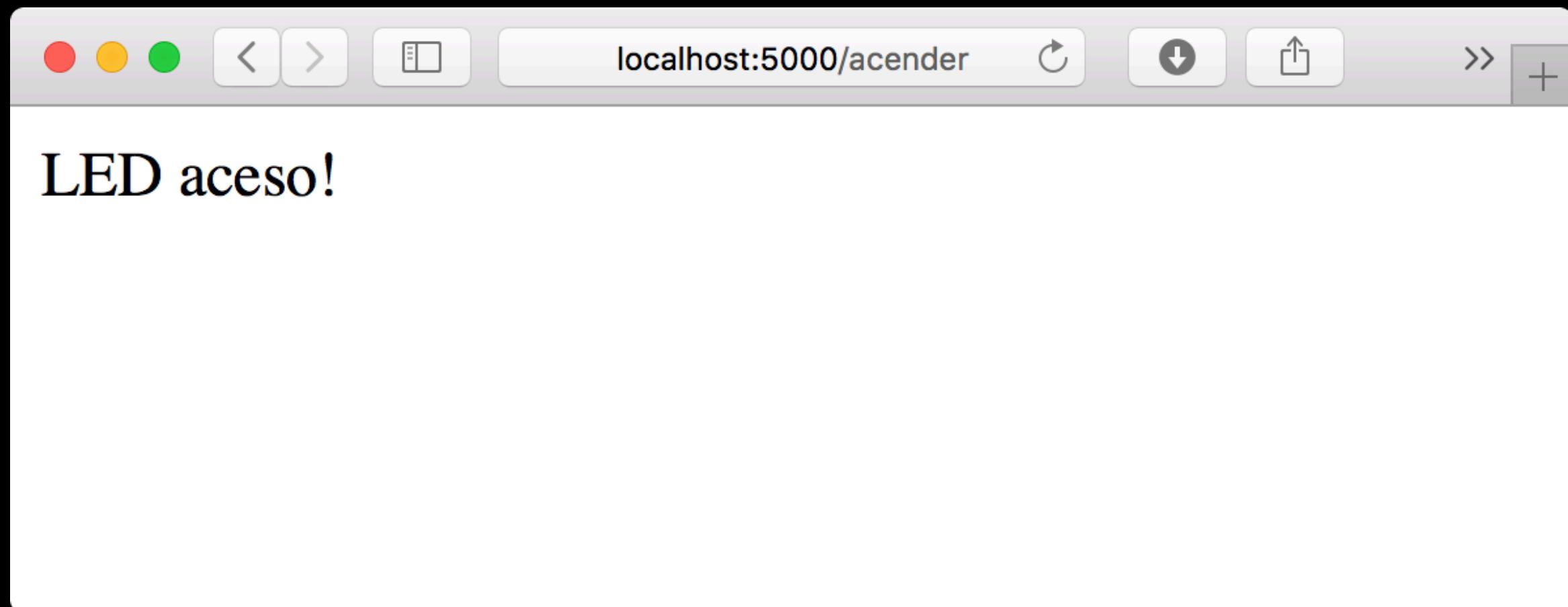
```
from gpiozero import LED  
from flask import Flask  
app = Flask(__name__)  
led = LED(21)
```

```
@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)
```

ATENÇÃO: não esqueça de retornar
um texto no final das funções



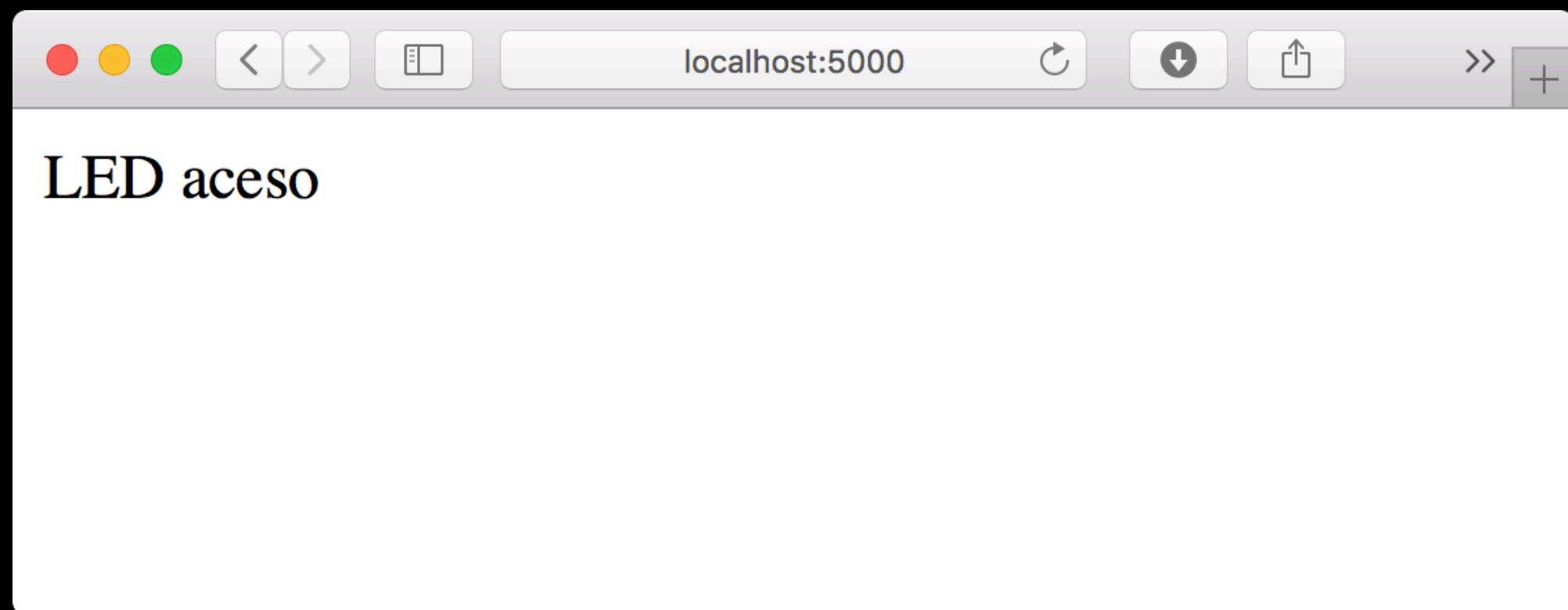
Controle de LED pelas Páginas no Navegador

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

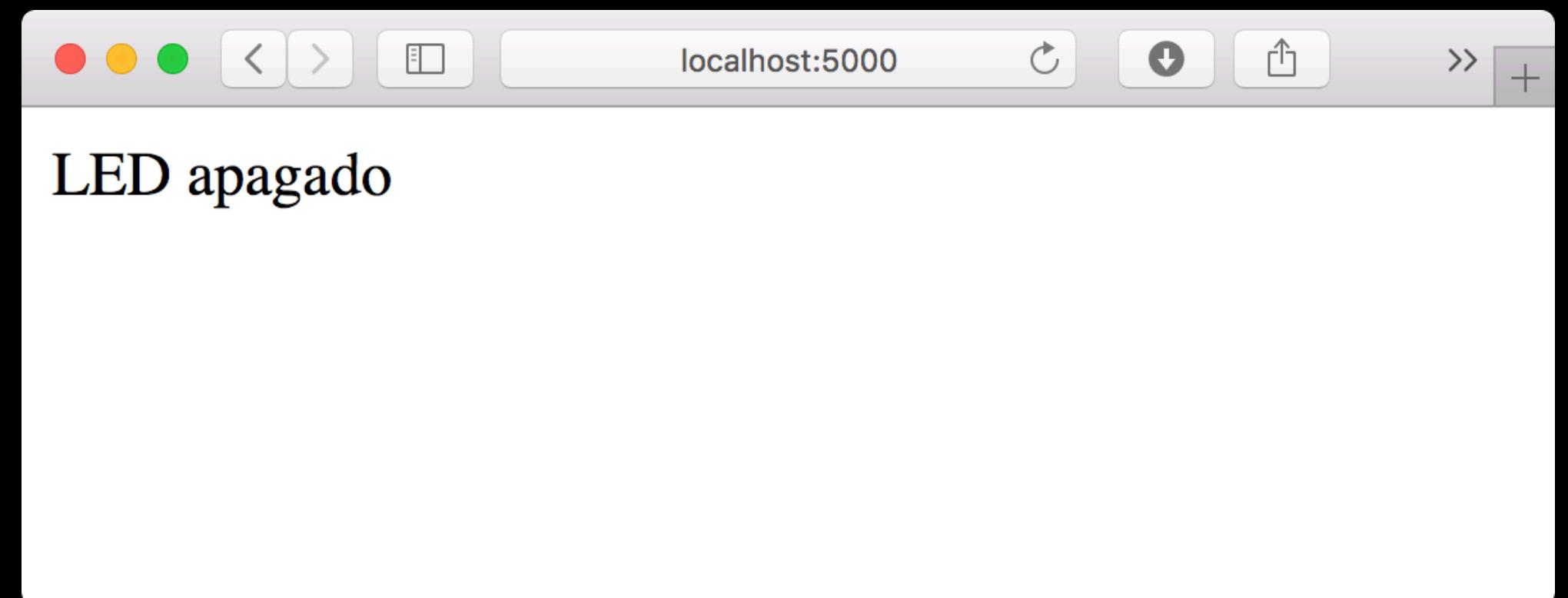
@app.route("/")
def pagina_principal():
    if led.is_lit:
        return "LED aceso"
    else:
        return "LED apagado"

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

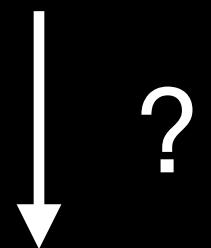
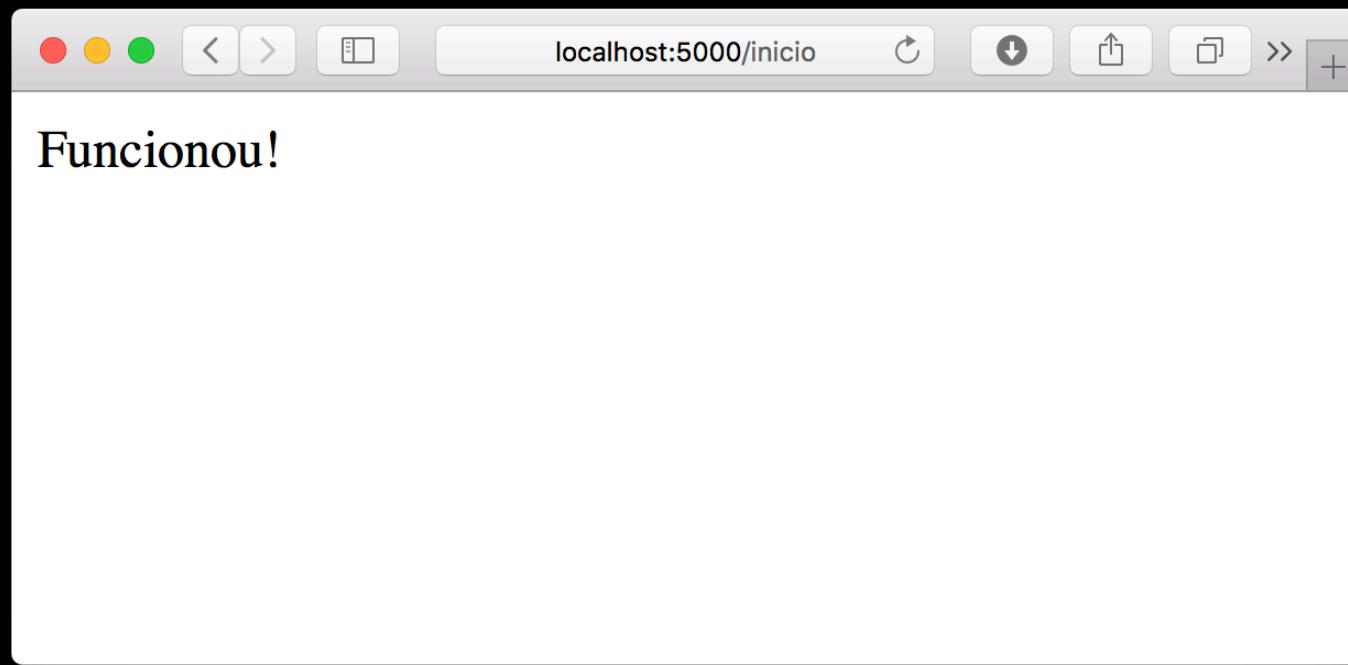
app.run(port=5000)
```



localhost:5000/alterar → redireciona para /



Redirecionamento de Página



A screenshot of the Facebook login page at www.facebook.com. The page features a blue header with the 'facebook' logo. Below the header, there's a message in Portuguese: 'O Facebook ajuda você a se conectar e compartilhar com as pessoas que fazem parte da sua vida.' To the right, there's a form for creating a new account, titled 'Abra uma conta'. It includes fields for 'Nome' (Name), 'Sobrenome' (Last Name), 'Celular ou email' (Cell phone or email), and 'Nova senha' (New password). There are also dropdown menus for 'Data de nascimento' (Birth date) set to '19 Mar 1993', and gender options 'Feminino' (Female) and 'Masculino' (Male). A note at the bottom explains that by clicking 'Criar conta', the user agrees to Facebook's terms and policies. A green 'Criar conta' (Create account) button is at the bottom right.

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="">November 2009</a></li>
```

Exemplo de Código Fonte em HTML

tag HTML

<elemento>

conteúdo do elemento

</elemento>

Elemento HTML como um Tag

Código	Resultado
<pre><p>Parágrafo 1</p> <p>Parágrafo 2</p></pre>	Parágrafo 1 Parágrafo 2
<pre>Negrito</pre>	Negrito
<pre></pre>	
<pre> Item 1 Item 2 Item 3 </pre>	<ul style="list-style-type: none"> • Item 1 • Item 2 • Item 3
<pre> Link para a Página </pre>	Link para a Página
Linha 1 Linha 2	Linha 1 Linha 2

```
from flask import Flask  
app = Flask(__name__)  
  
@app.route("/pagina_com_html")  
def mostrar_pagina_com_html():  
    return """  


Bem-vindo ao meu site!



Alguns links:



Página de Contato



Google



Tarefas:



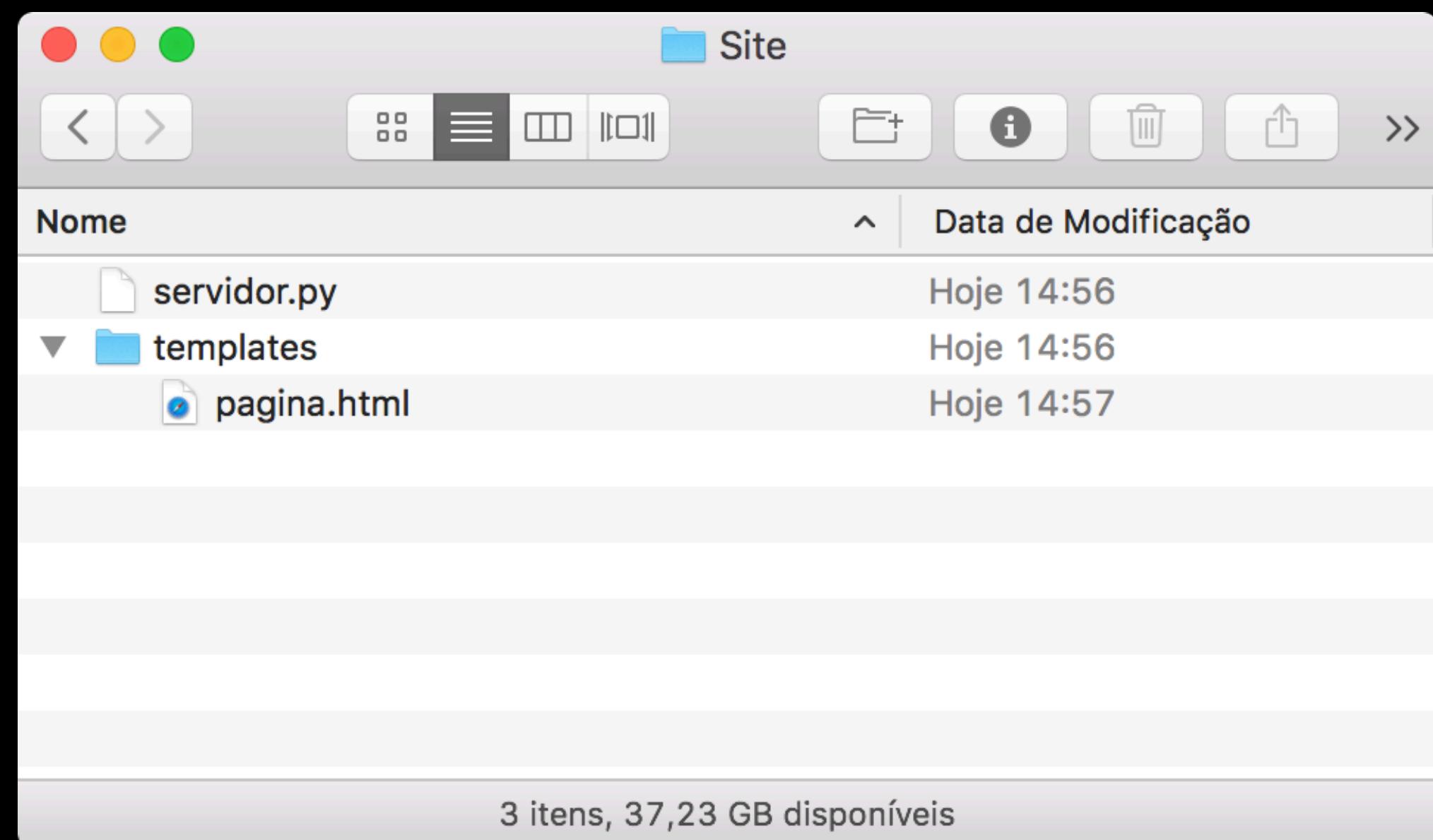
- Terminar de fazer a aula
- Dar a aula
- Arrumar a sala
- Dominar o mundo!

  
....  
  
app.run(port=5000)
```

O HTML ocupa muito espaço,
e confunde o código Python



Inclusão de Código HTML dentro do Código Python?



Organização de Arquivos com o Flask

```
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)
```

pagina.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 >

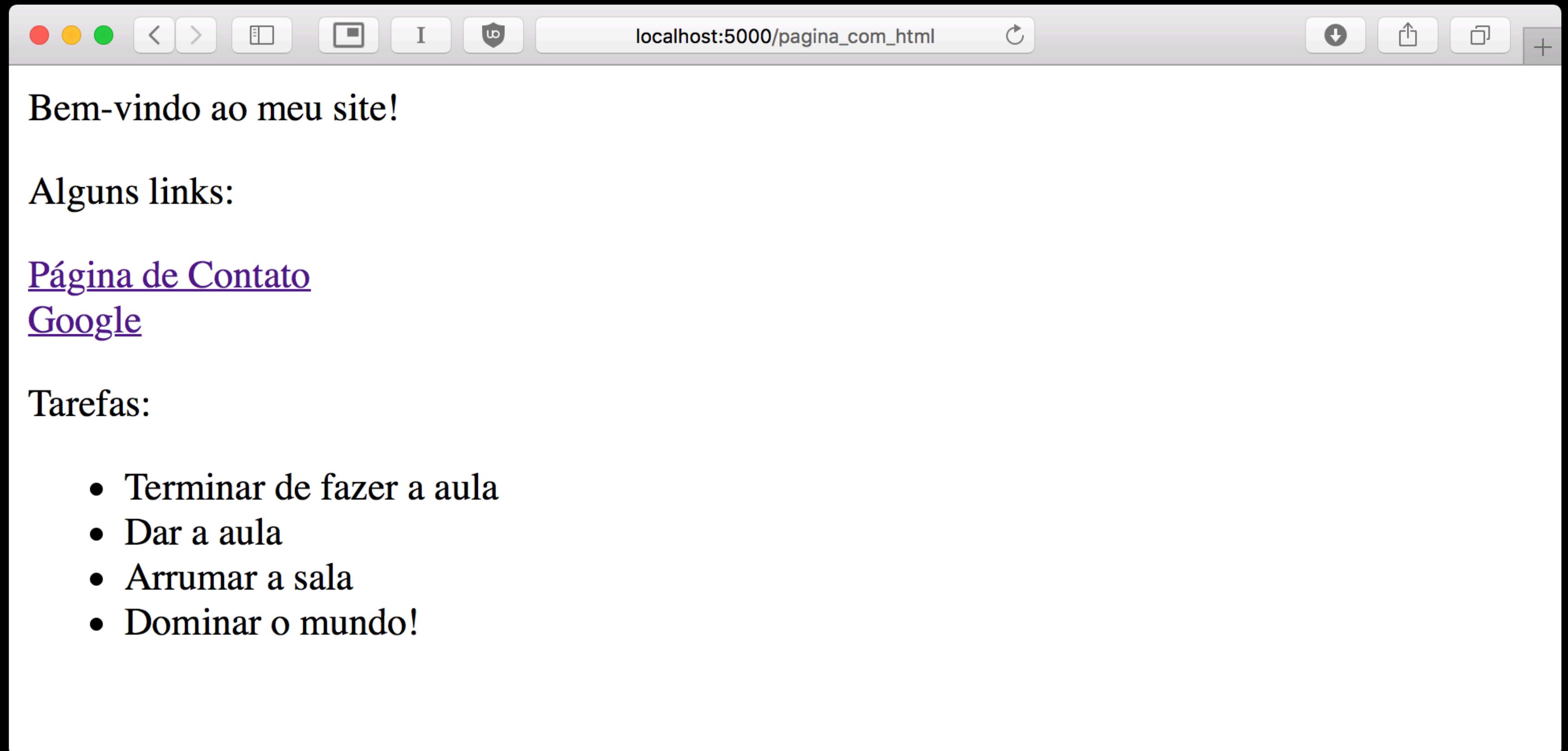
< li > Terminar de fazer a aula </ li >

< li > Dar a aula </ li >

< li > Arrumar a sala </ li >

< li > Dominar o mundo! </ li >

< / ul >



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



Acesso do Site sem Usar Localhost?

Screenshot of the Hover website showing domain search results for "meusite".

The page header shows the URL www.hover.com/domains/results?ut.

EXACT MATCH

meusite.com [MAKE AN OFFER](#)

FEATURED

meusite.design 😊 **\$5.99 / \$39.99** [+](#)
For those who can't help but create.

meusite.tech 😊 **\$7.99 / \$49.99** [+](#)
Showcase your creations with a .tech domain.

GENERIC

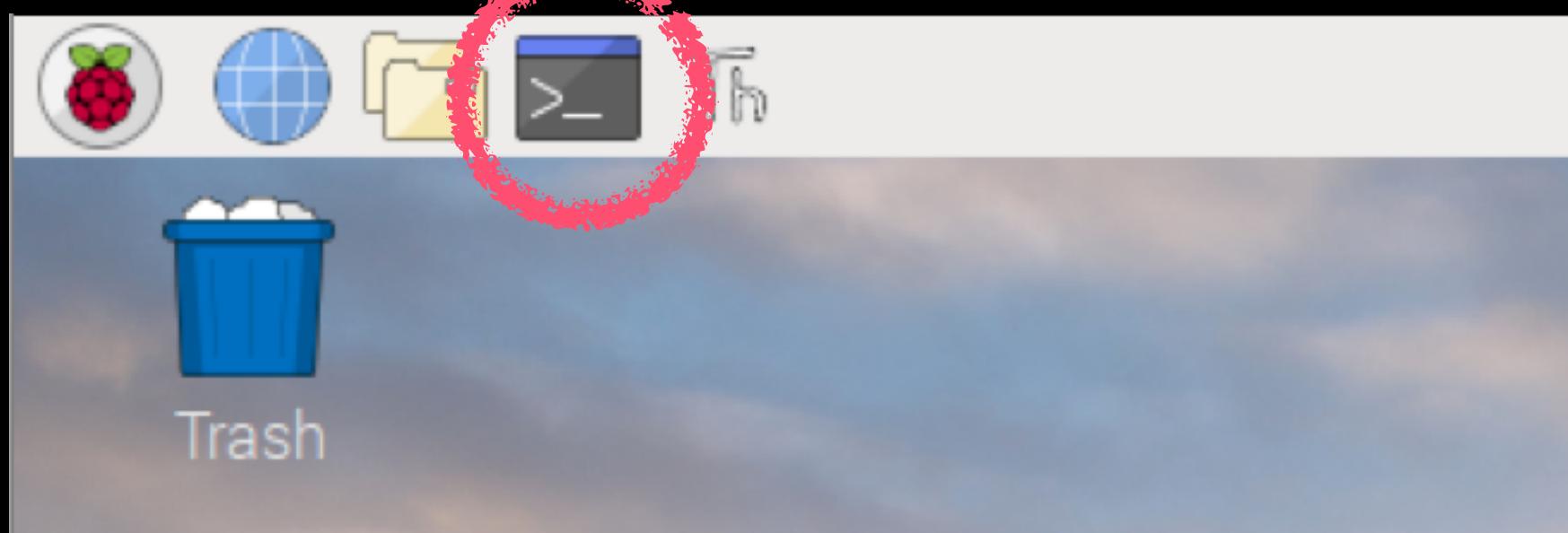
meusite.org  **\$13.99** [+](#)
meusite.site **\$32.99** [+](#)

Solução Demorada: Compra + Configuração de um Domínio

ngrok

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

Solução Rápida: Ngrok

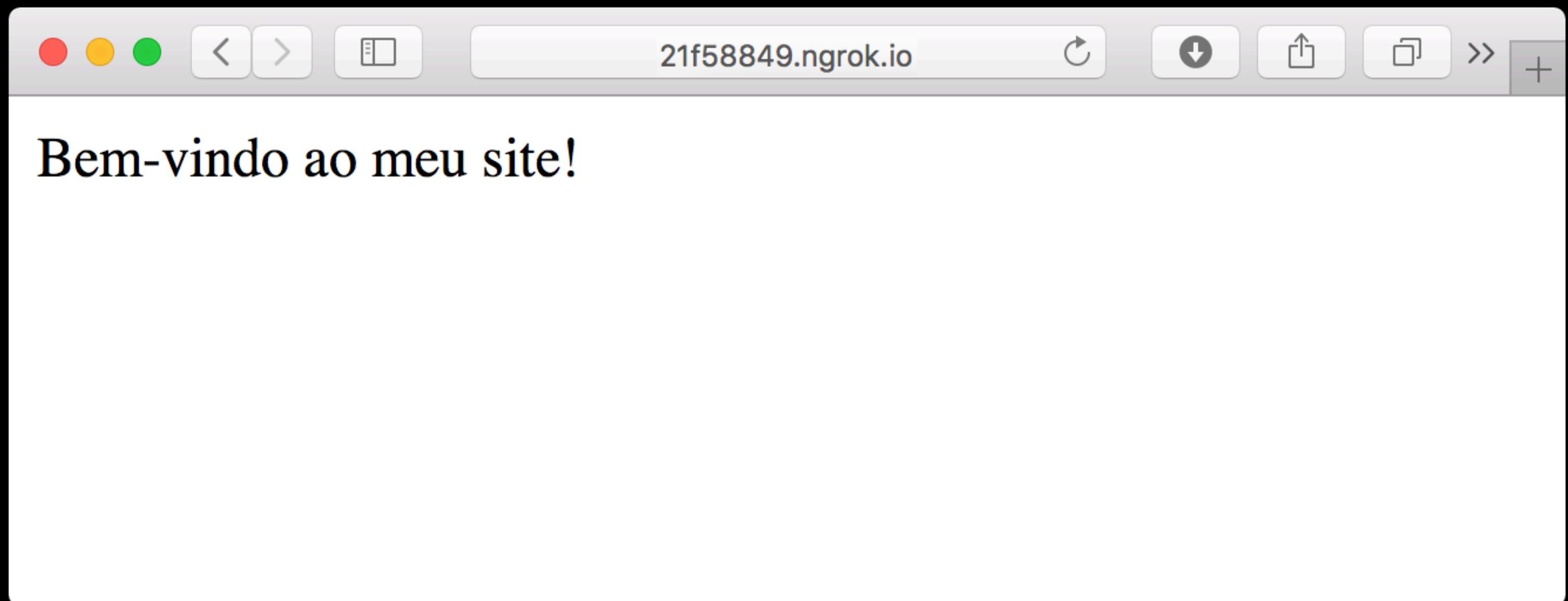


```
aula@raspberrypi ~ $ ngrok http 5000
```

```
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

```

from flask import Flask
app = Flask(__name__)

@app.route("/")
def funcao_da_pagina_inicio():
    return "x = " + 2

app.run(port=5000)

```

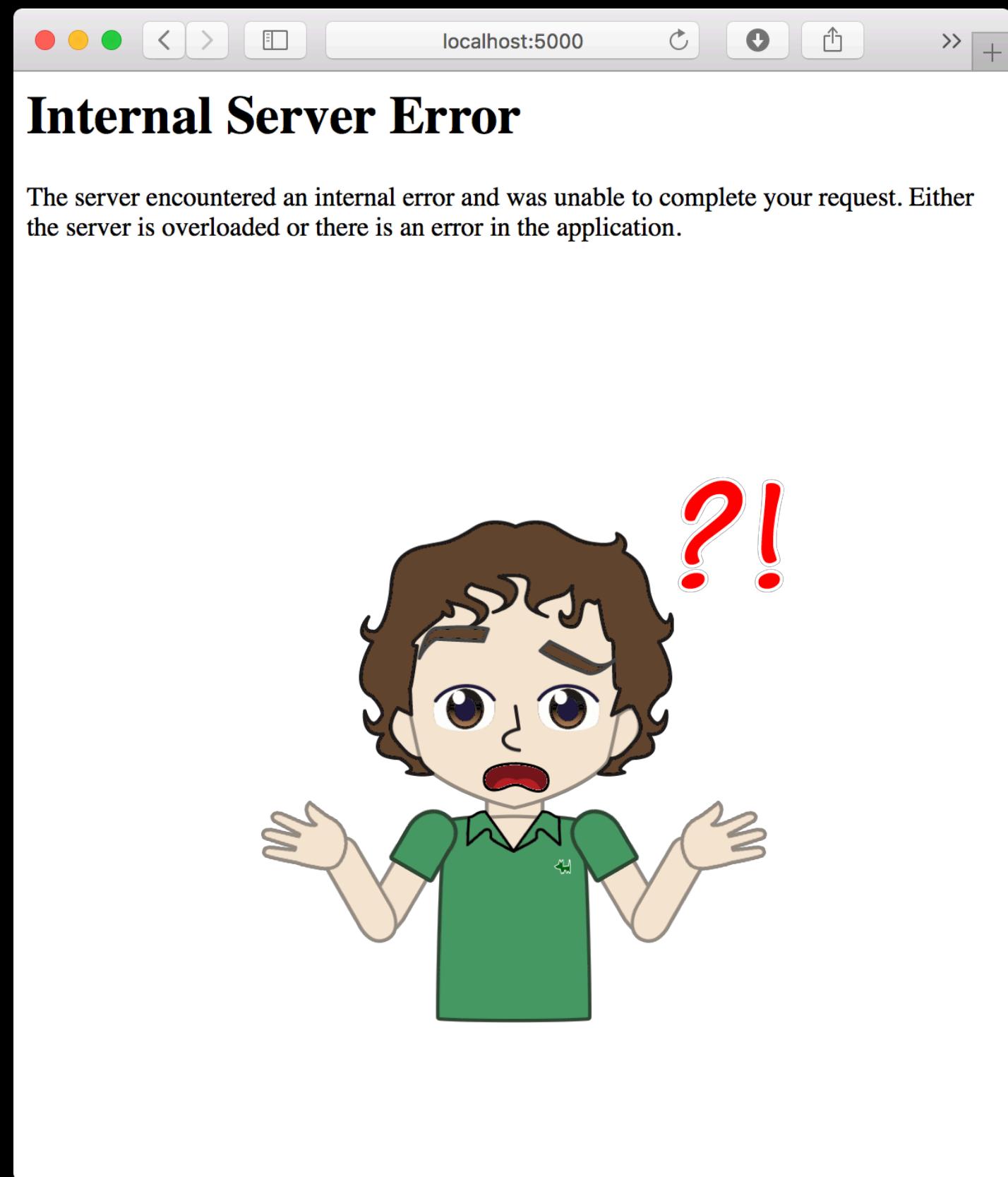
```

from flask import Flask
app = Flask(__name__)

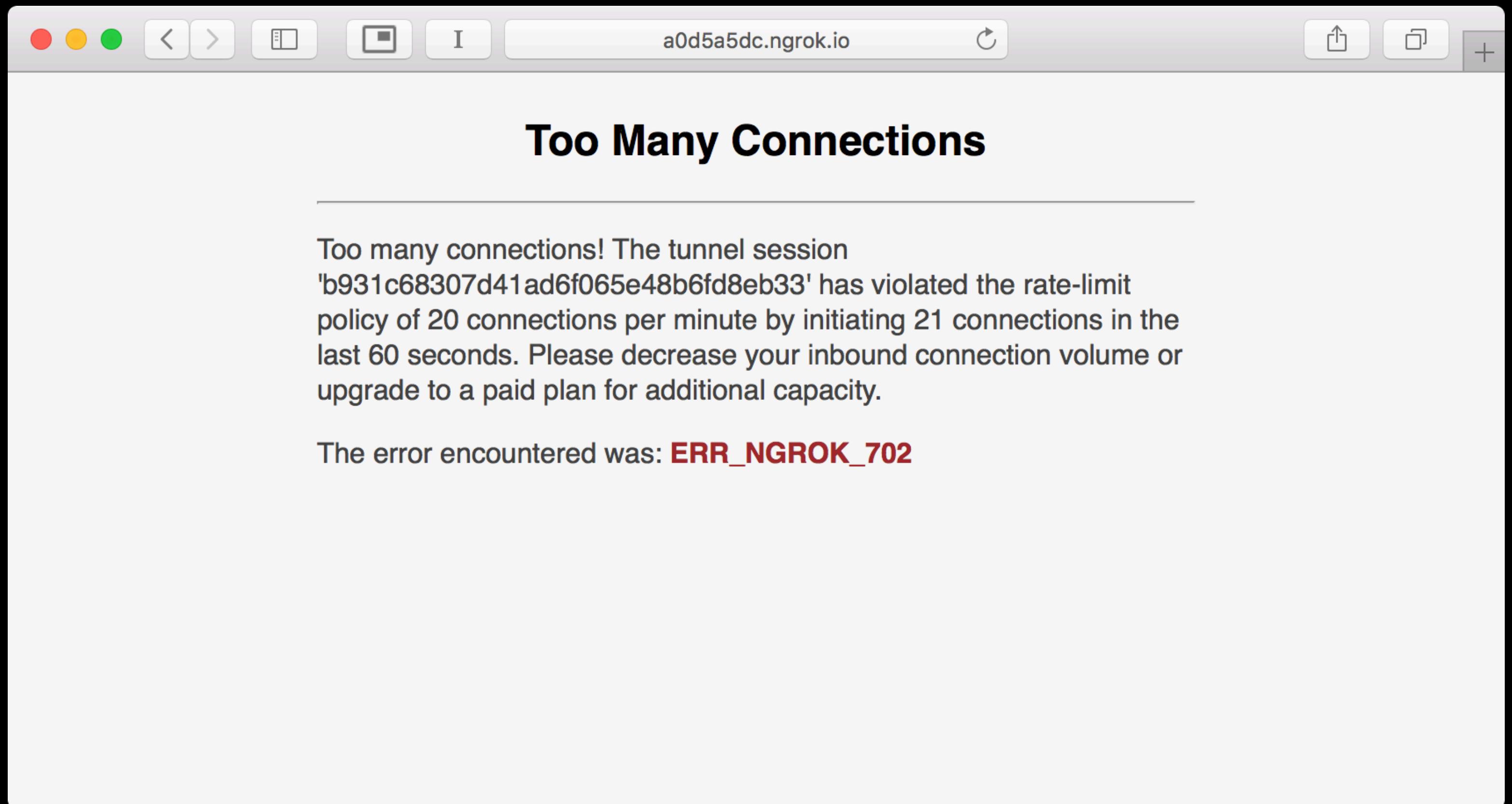
@app.route("/")
def funcao_da_pagina_inicio():
    > return "x = " + 2

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

```



Modo "Debug" para Encontrar Erros



Limite de Acessos Usando o Ngrok

Resumo da Ópera

Funcionalidade

Comandos

Envio de Infravermelho

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

Recebimento de 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>

<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

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
```

Funcionalidade	Comandos
Funções	<pre>x = input("Digite um número: ") • print("Resultado: ", x) from time import sleep • sleep(0.5)</pre>
Listas acessar documentação	<pre>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)</pre>
Dicionários acessar documentação	<pre>dicionario = {"chave 1": 42, "chave 2": [1, 2, 3]} dicionario["chave 1"] • dicionario["chave 3"] = "Olá!"</pre>
Textos (Strings) acessar documentação	<pre>texto = "olá!" • texto[0] • texto[1:4] • len(texto) texto + "\n" • texto + str(numero) • "x = %.2f" % numero 2 + int("11") • 4 / float("23.5")</pre>
Condicionais	<pre>if x != 0: if x not in [1, 2]: y = 4 else: y = 0 elif x >= 0: y = 3 else: y = 0</pre>
Repetições	<pre>for elem in lista: ... for i in range(1, 4): ... while x > 1: ...</pre>
Criação de Funções	<pre>def funcao1(x): return x + 2 def funcao2(x, y, z): ... def funcao3(): global x</pre>