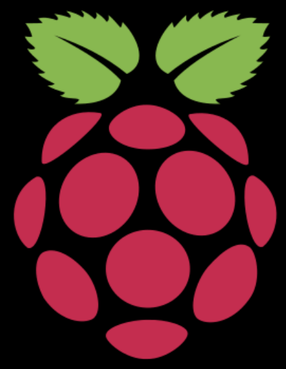


Projeto 02

Controle Remoto – Prática

Jan K. S. – janks@puc-rio.br

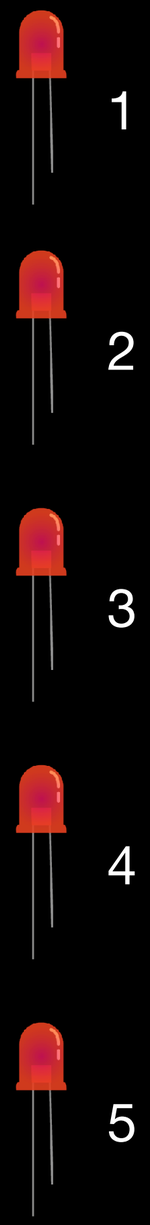
ENG 1419 – Programação de Microcontroladores

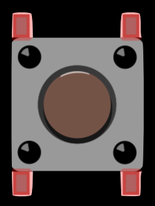
Testes Iniciais

seleciona um LED

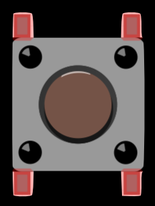
e

acende ou apaga





acende todos os LEDs



apaga todos os LEDs

Controle Remoto de LEDs

① ② ③ ④ ⑤

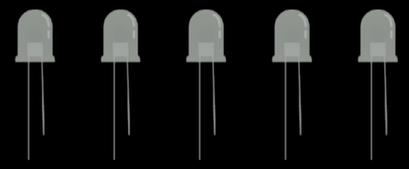
seleciona LED
pelo número

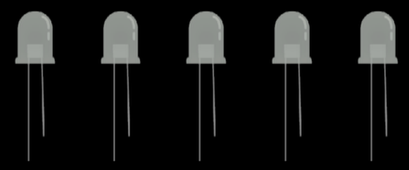
▲ ▼

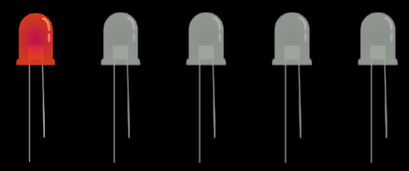
vai para próximo LED
ou anterior

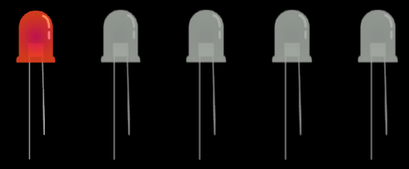
OK

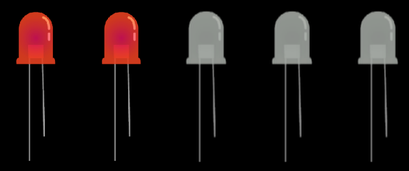
apaga/acende
LED selecionado

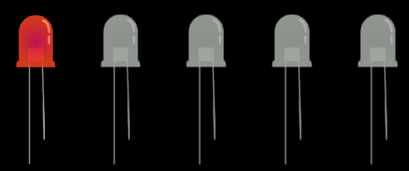












Controle Remoto de LEDs

KEY_UP

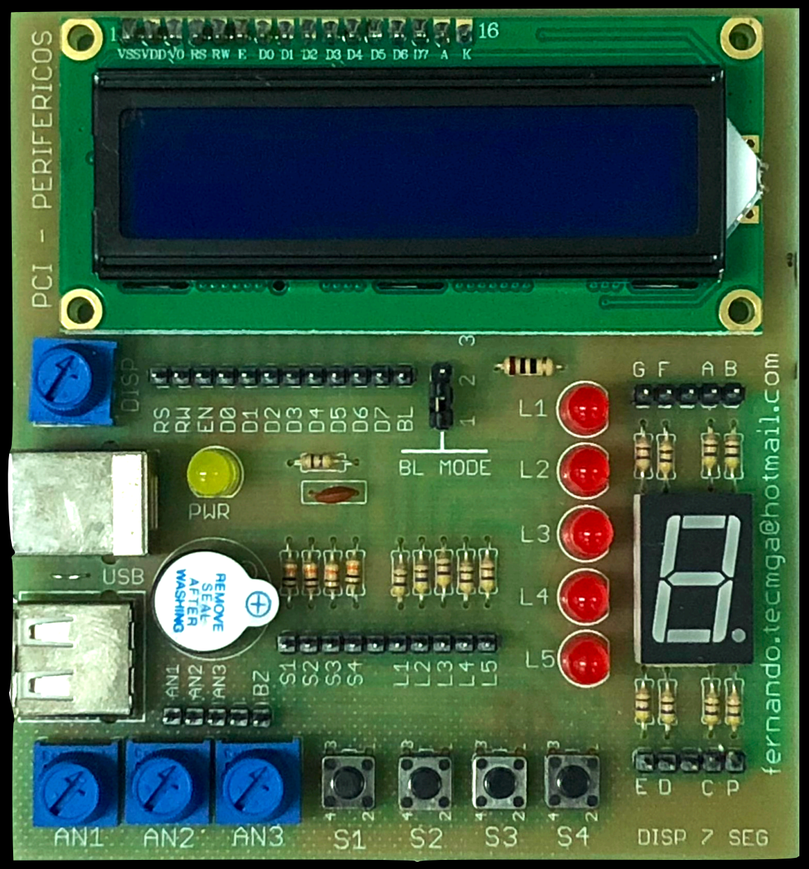
KEY_LEFT KEY_OK KEY_RIGHT

KEY_DOWN

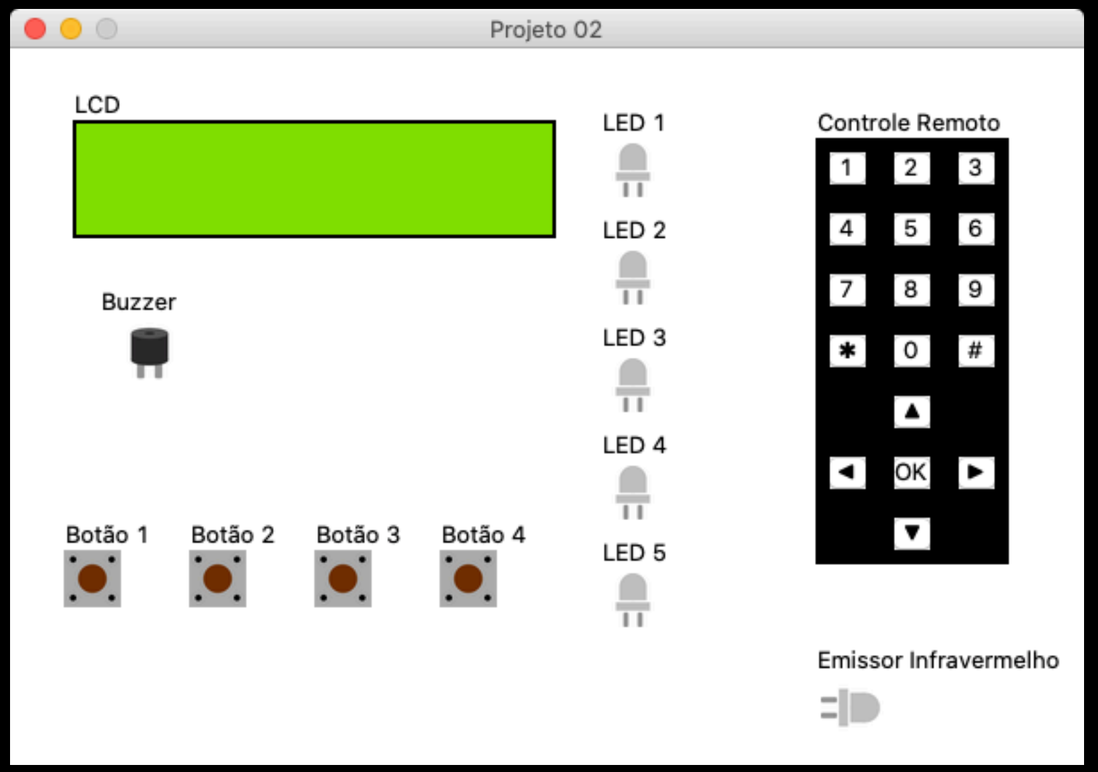
KEY_1	KEY_2	KEY_3
KEY_4	KEY_5	KEY_6
KEY_7	KEY_8	KEY_9
	KEY_0	

Códigos das Teclas do Mini Controle Remoto

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



GPIO 21
GPIO 22
GPIO 23
GPIO 24
GPIO 25



GPIO 11, 12, 13, 14

```
leds = [LED(21), LED(22), LED(23), LED(24), LED(25)]
```

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

```
from extra.aula import rodar

@rodar
def programa():
```

escrevam daqui para baixo!

```
    # importação de bibliotecas
    from gpiozero import LED
    from time import sleep

    # definição de funções

    # criação de componentes
    led = LED(21)

    # loop infinito
    while True:

        sleep(0.2)
```

Programa Dentro de uma Função

```python
from extra.aula import rodar

@rodar
def programa():
    global x        # colocar do lado de fora também!
    x = 2

    def aumenta_x():
        global x  # alterar uma variável global
        x = x + 2
        return x


while True:   # obrigatório no final da função!

    sleep(0.2)
```

Pequenas Mudanças para o Circuito Virtual

Testes Iniciais

Ao apertar o Botão 1 da placa, acenda todos os LEDs. Ao apertão o Botão 2 da placa, apague todos os LEDs.
↳ DICA: use a propriedade when_pressed do botão e crie funções que percorram a lista de LEDs.

Ao apertar uma tecla numérica, "selecione" um dos LEDs, mostrando o valor no LCD de caracteres.
↳ DICA: use o código do recebimento de teclas com o `next_code` dentro do `while True`.

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

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

Implementação

Controle Remoto de uma TV Digital

Conversor Digital Aquário

controle "aquario"

KEY_POWER KEY_MUTE

KEY_1 KEY_2 KEY_3

KEY_4 KEY_5 KEY_6

KEY_7 KEY_8 KEY_9

KEY_0

KEY_MENU KEY_EXIT

KEY_CHANNELUP

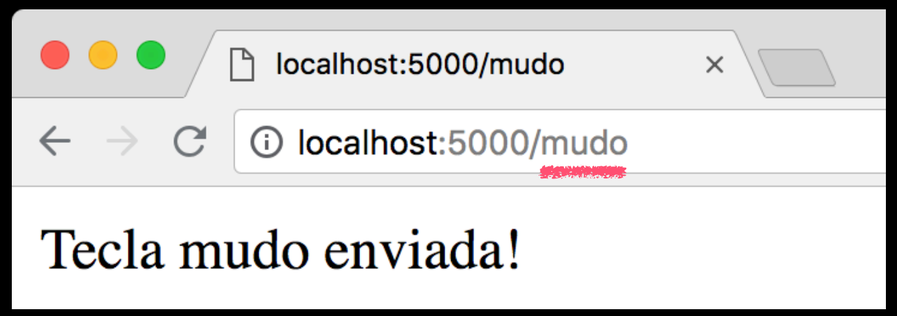
KEY_VOLUMEDOWN KEY_OK KEY_VOLUMEUP

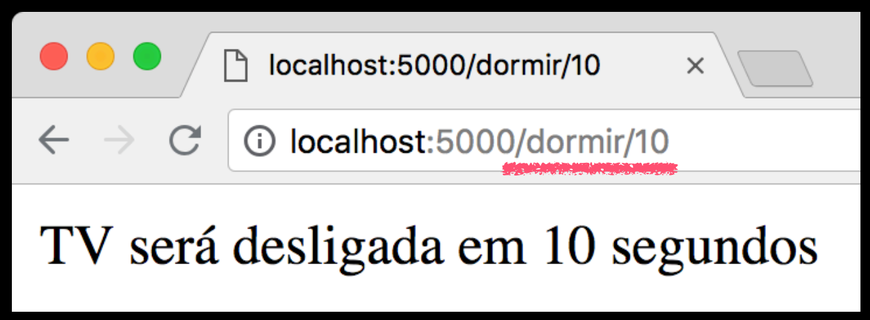
KEY_CHANNELDOWN

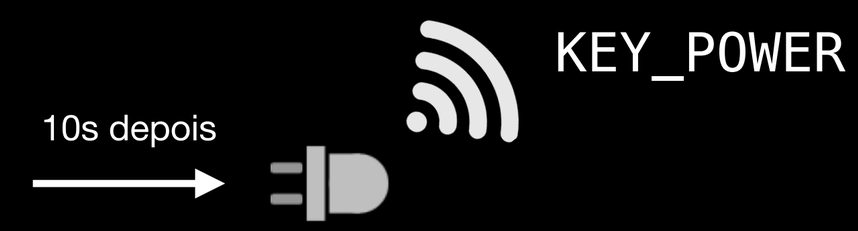
Código das Teclas Usadas

02.01	TV Brasil HD
02.02	NBR
02.03	TV Escola
02.04	Canal Saúde
02.31	TV Brasil 1seg
04.01	Globo HD
04.31	Globo 1seg
06.01	Rede TV! HD
06.31	Rede TV! 1seg
07.01	Band HD
07.31	Band 1seg
09.01	CNT HD
09.31	CNT Movel
11.01	SBT HD
11.31	SBT 1seg
12.01	NGT HDTV
12.31	NGT 1seg
13.01	Record HD
13.31	Record 1seg
14.01	RCI 1seg
14.31	RCI Movel
20.10	Canção Nova HD
20.31	Canção Nova 1 seg
...	

Código dos Canais Brasileiros







Páginas para Funções e Canais

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

```
send_once(...) ⟶
```

Shell

```
KEY_POWER of remote "aquario" transmitted!
KEY_CHANNELUP of remote "aquario" transmitted!
```

Implementação

Crie um servidor com páginas para simular o botão Power, aumentar o volume, diminuir o volume e acionar a função mudo. Teste essas página no navegador.
↪ DICA: use a `send_once`. Lembre-se que toda a função de página deve retornar um texto.

Crie uma página que mude para o canal, recebendo um parâmetro tipo "string" com o texto dos dígitos numéricos (ex: "0431").
↪ DICA: use um for para acessar cada caracter da string, e faça concatenação para enviar o código.

Crie uma página com um parâmetro numérico N no endereço que desligue a TV após N segundos.
↪ DICA: não use a função `sleep`! Pesquise sobre o objeto `Timer` do Python, da biblioteca threading.

Aperfeiçoamento

02b_implementacao.py

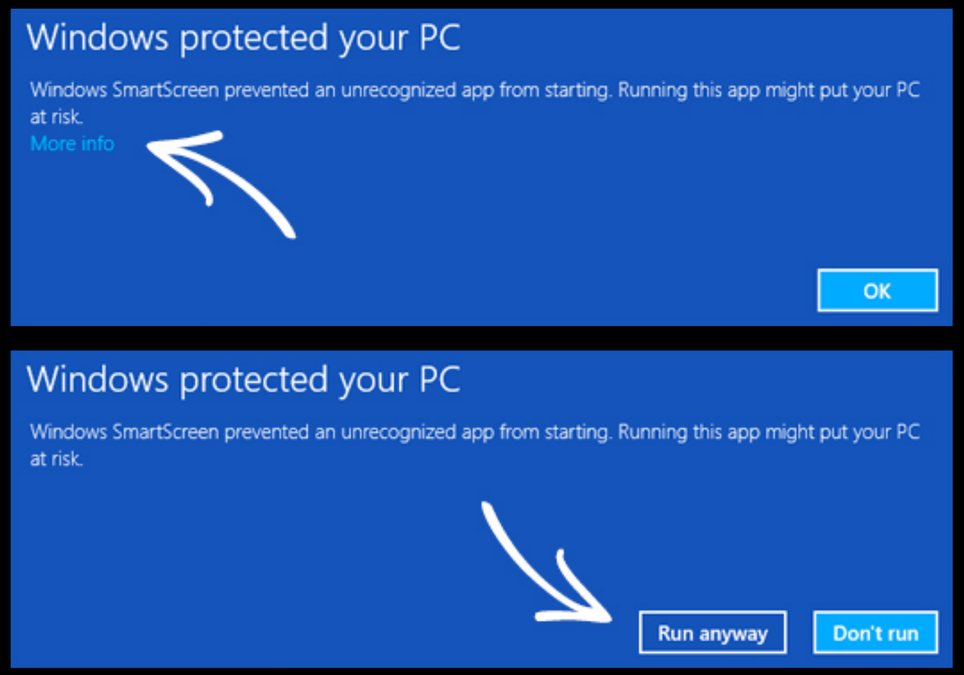
cópia

02c_aperfeicoamento.py

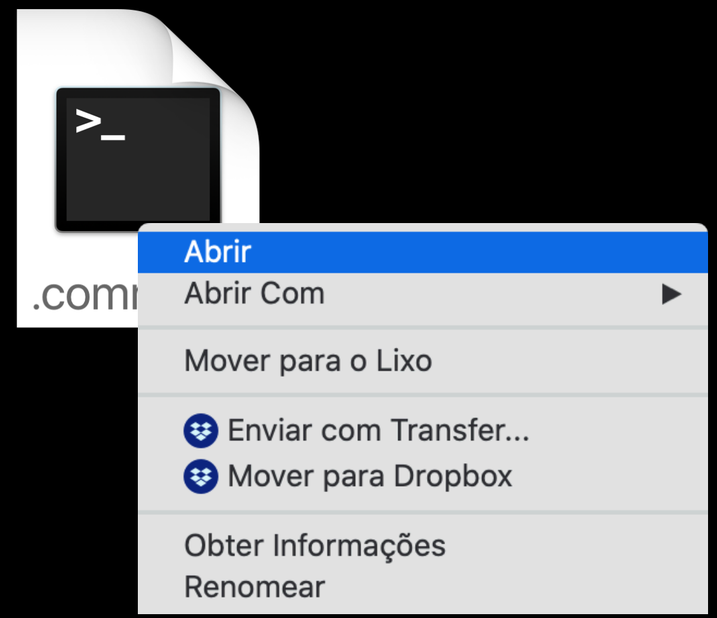
Cópia do Código da Implementação para o Aperfeiçoamento

Troca de Canais pelo Celular

rodar_ngrok.bat



rodar_ngrok.command



Execução do NGrok no Windows e no Mac

Aperfeiçoamento

Crie uma página principal com links para os canais (pelo menos 5, exibindo o nome deles) e as outras funcionalidade (volume, mudo, power, dormir, etc).

Redirecione as páginas de funções e canais de volta para a página principal.

Use o ngrok para controlar a TV pelo celular.
↳ DICA: use o arquivo rodar_ngrok.bat (Windows) ou rodar_ngrok.command (Mac → clicar com botão direito > Abrir).

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

Desafio Extra

página.html

```
<ul>
   link para canal 1
   link para canal 2
   link para canal 3
   link para canal 4
   ...
</ul>
```

canais.json

```
[
    {
        "nome": "Globo",
        "código": "0431"
    },
    {
        "nome": "Band",
        "código": "0731"
    },
    ...
]
```

página.html

```
<ul>
   para cada canal:
      gera link do canal
</ul>
```

Geração da Lista de Canais a Partir de um JSON

Desafio Extra

Crie um arquivo JSON com uma lista de dicionários de canais (cada um com chaves "nome" e "código").

Carregue o arquivo JSON ao iniciar o servidor, gerando uma lista de dicionários dos canais.

Passe a lista de dicionários para o template ao renderizar página principal.
↳ DICA: pesquise no Google como passar parâmetros para um template **em Flask**.

Gere os links de canais no template HTML a partir da lista.
↳ DICA: pesquise no Google como criar um loop dentro de um template HTML **em Flask**.

janks.link/micro/projeto02.zip

Material do Projeto 02