

Projeto 04

Controle por Notificação – Prática

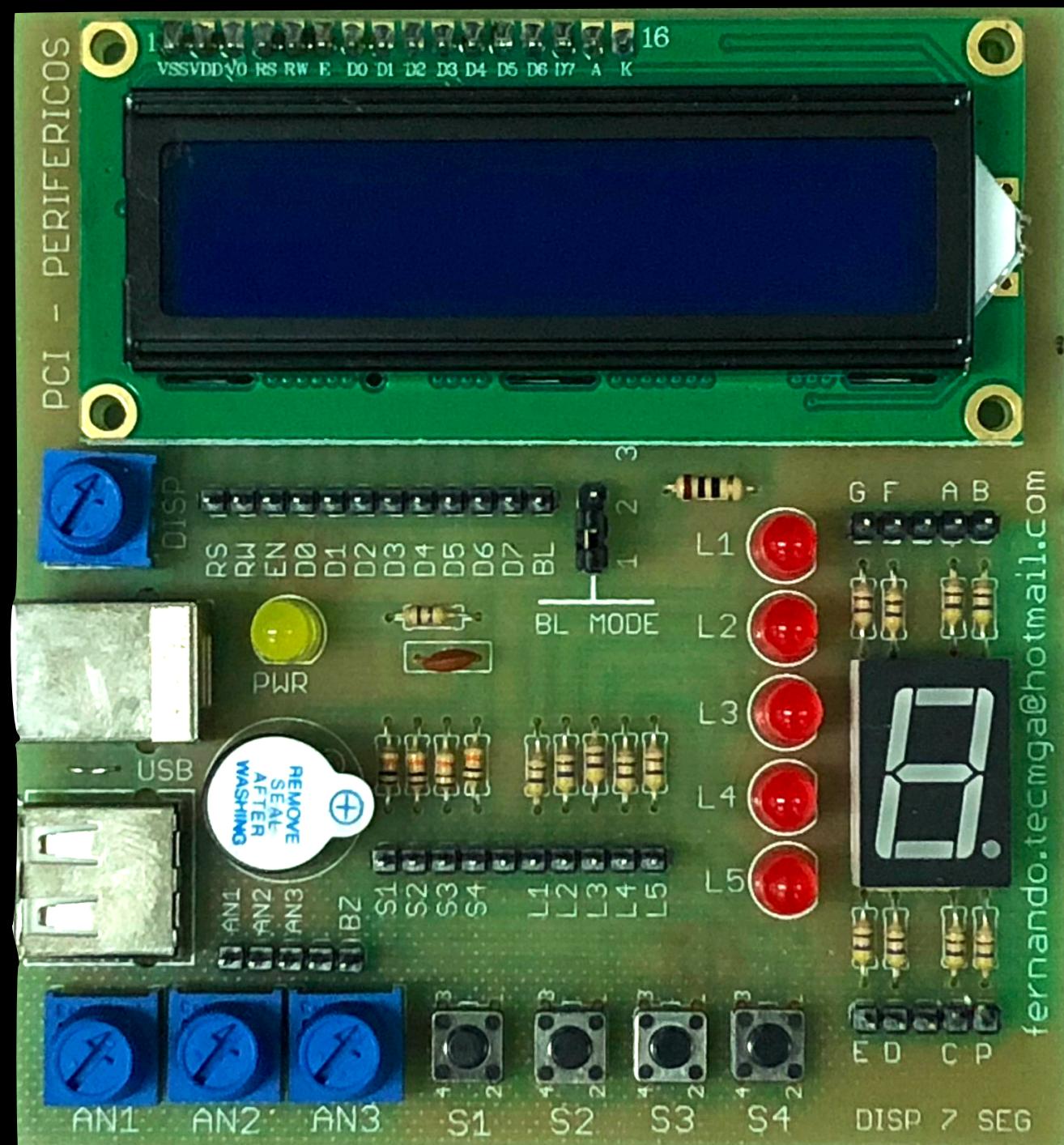
Jan K. S. – janks@puc-rio.br

ENG1419 – Programação de Microcontroladores

Testes Iniciais

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

GPIO 16

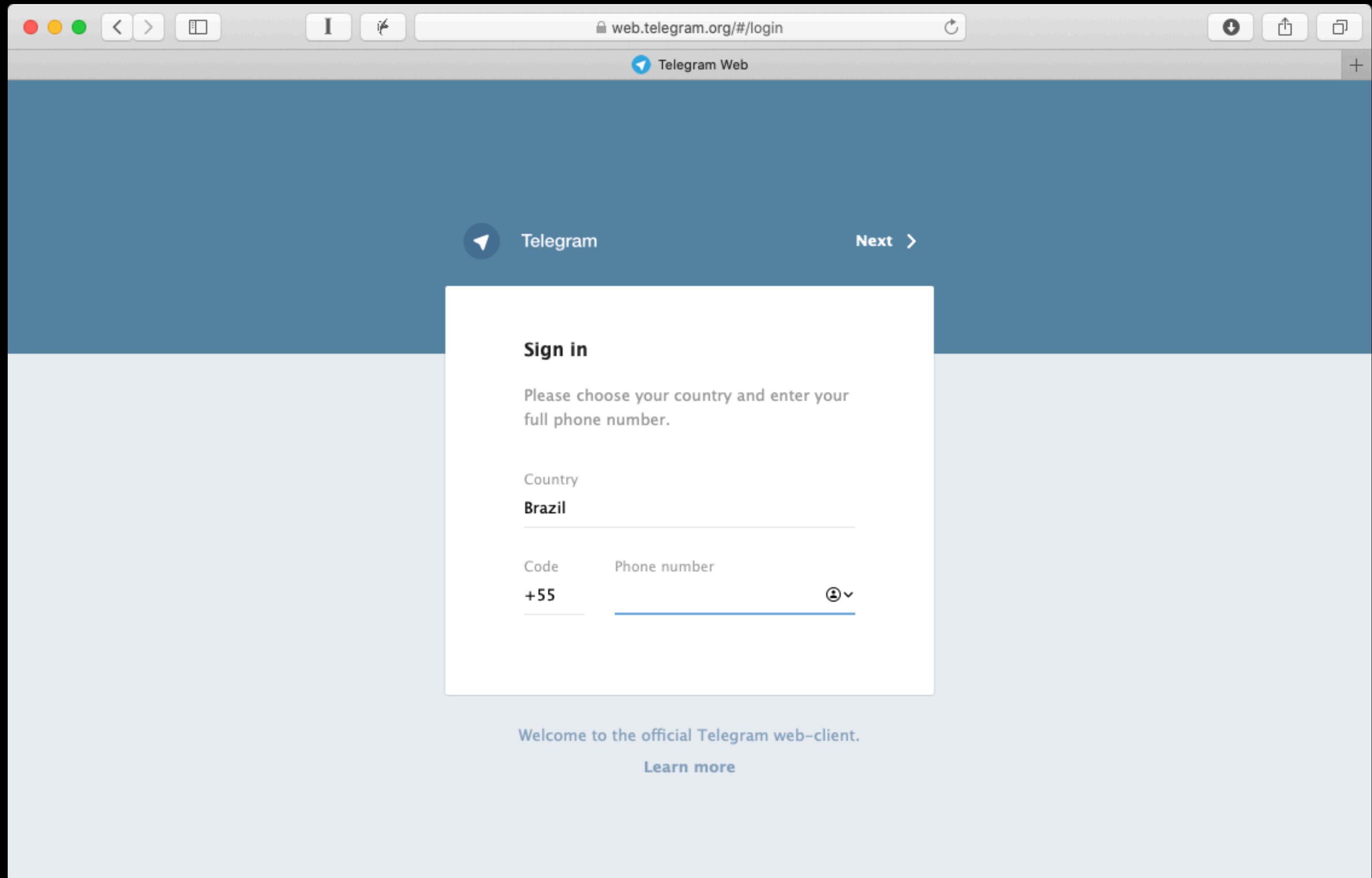


GPIO 11, 12, 13, 14

GPIO 21
GPIO 22
GPIO 23
GPIO 24
GPIO 25

Conexões com as Portas da GPIO

web.telegram.org



Telegram Web

```
>>> endereco = endereco_base + "/sendMessage"
>>> dados_incompletos = {"text": "Olá!"}
>>> resposta = post(endereco, json=dados_incompletos)
>>> print(resposta.text)
{"ok":false,"error_code":400,"description":"Bad
Request: chat_id is empty"}
```

DICA: Mensagens de Erro do Telegram (Caso Algo Não Esteja Funcionando)



Testes Iniciais

Ao apertar o botão 1, exiba o texto "Gravando..." no LCD, **grave 5 segundos de áudio** e apague o texto. Abra o arquivo manualmente para ouvir a gravação.
↪ DICA: chame a arecord com a **system**.

Ao apertar o botão 2, tire **5 fotos com a webcam**, com 2 segundos de intervalo entre elas. Abra os 5 arquivos manualmente para visualizar as imagens.
↪ DICA: use um for, variando o nome do arquivo.

Pisque 1 vez rapidamente o LED 1 após capturar cada foto no item anterior.

Configure o seu bot no Telegram, anotando a chave secreta. Em seguida, inicie uma conversa com ele pelo aplicativo. Por fim, obtenha o id do chat pelo navegador.
↪ DICA: veja os slides 36, 37 e 38 da teoria. Use a versão web do aplicativo em web.telegram.org.

Ao apertar o botão 3, **envie uma mensagem de texto** qualquer para a conversa com o bot.
↪ DICA: imprima o **resposta.text** do post.

Implementação



Visitantes?

Controle de Acesso para Visitantes



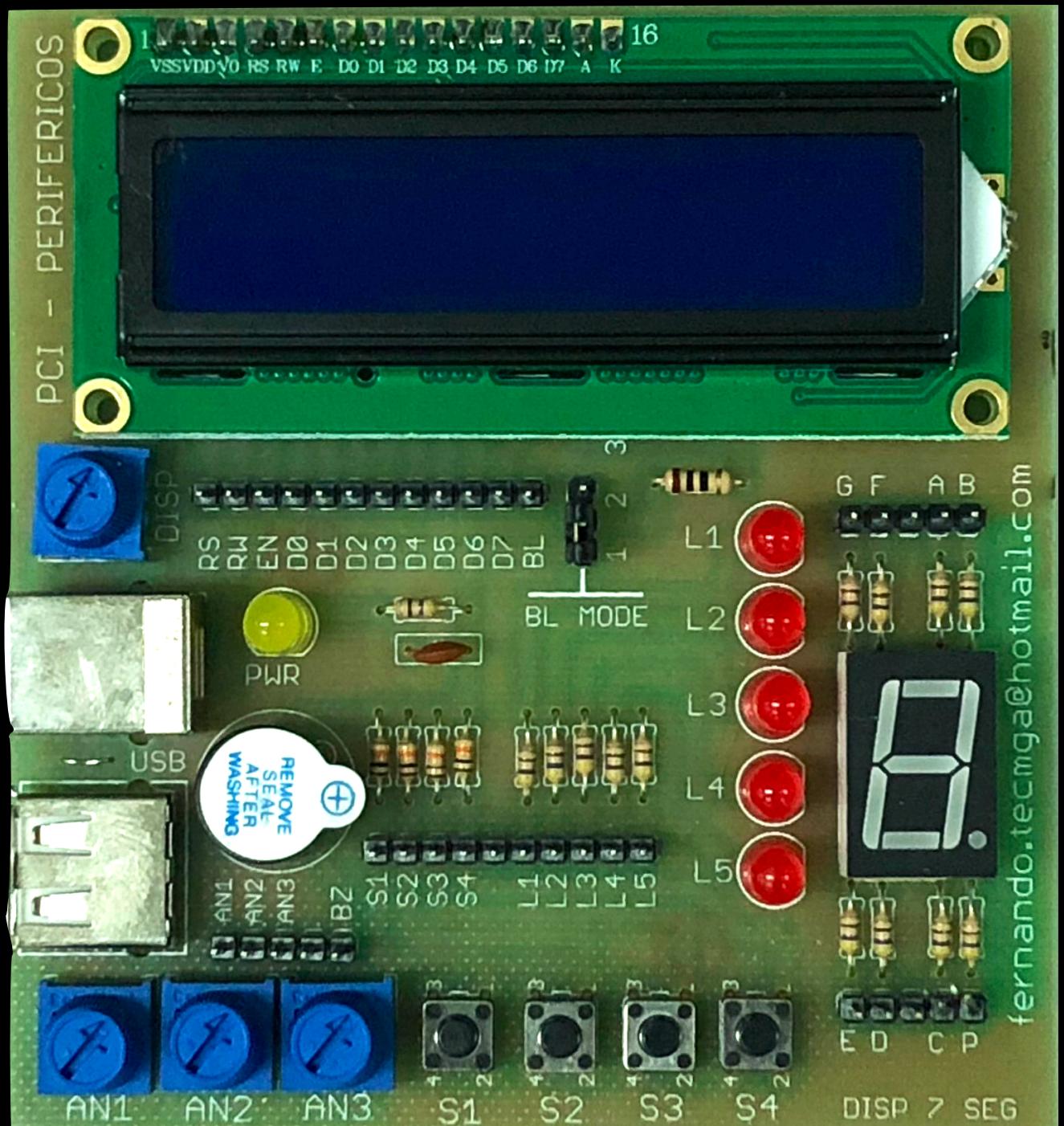
notificação com foto

comando

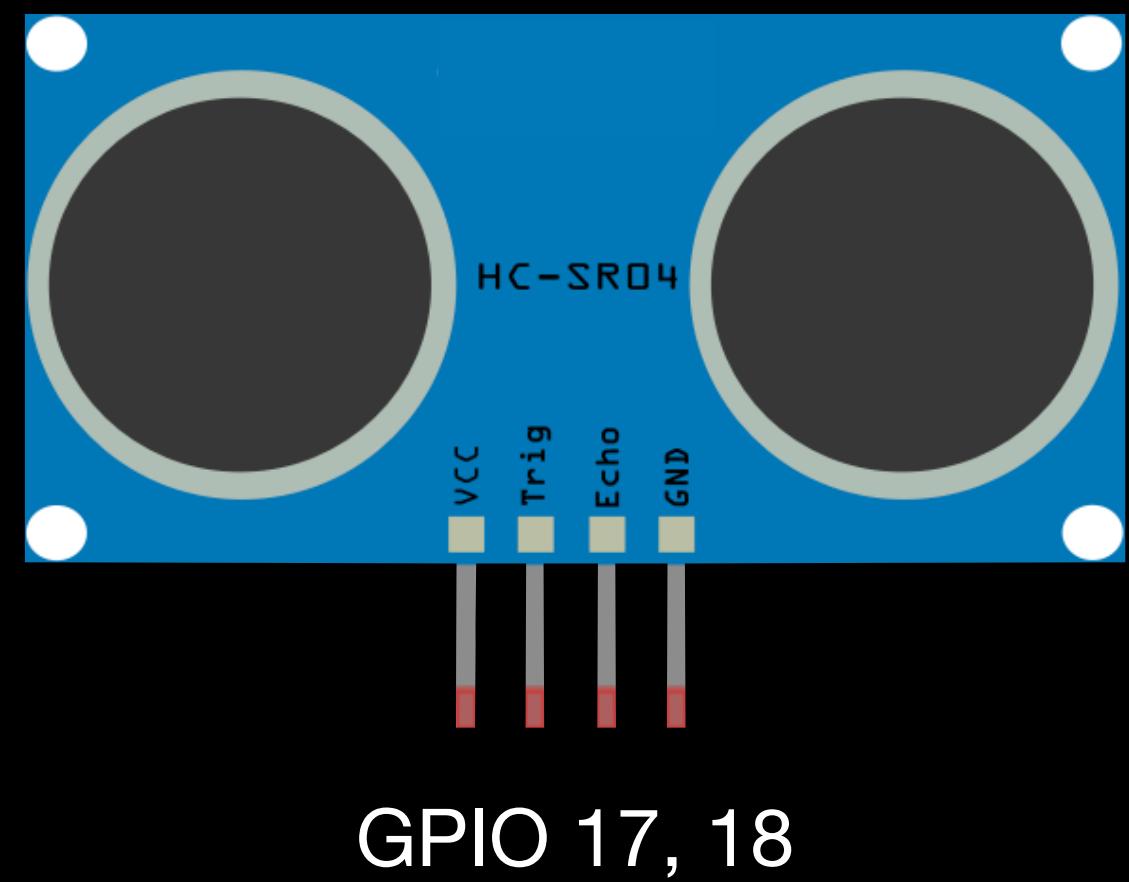


Controle de Acesso por Notificação

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



GPIO 21
GPIO 22
GPIO 23
GPIO 24
GPIO 25



Conexões com as Portas da GPIO

...

```
print("Enviando foto...")
```

...

```
print("Foto enviada!")
```

...

```
print("Enviando mensagem de texto...")
```

...

DICA: Acompanhamento das Etapas com Print



Implementação

Ao apertar o Botão 1, **ligue a campainha**. Ao soltá-lo, **desligue a campainha**.

Logo após desligar a campainha, envie uma **mensagem para o chat** avisando que alguém está na porta.

↪ DICA: coloque prints para antes e depois do envio.

Logo após a mensagem anterior, **envie também uma foto** tirada pela webcam, para mostrar quem está na porta.

Ao receber uma mensagem "**Abrir**" do usuário pelo chat, acenda o LED 1 (representando a abertura da porta). Ao receber "**Alarme**", toque a campainha 5 vezes rapidamente.

↪ DICA: veja a receita do Telegram com o while True para ficar verificando novas mensagens.

Ao apertar o Botão 2, **apague o LED 1** (representando o fechamento da porta).

Aperfeiçoamento



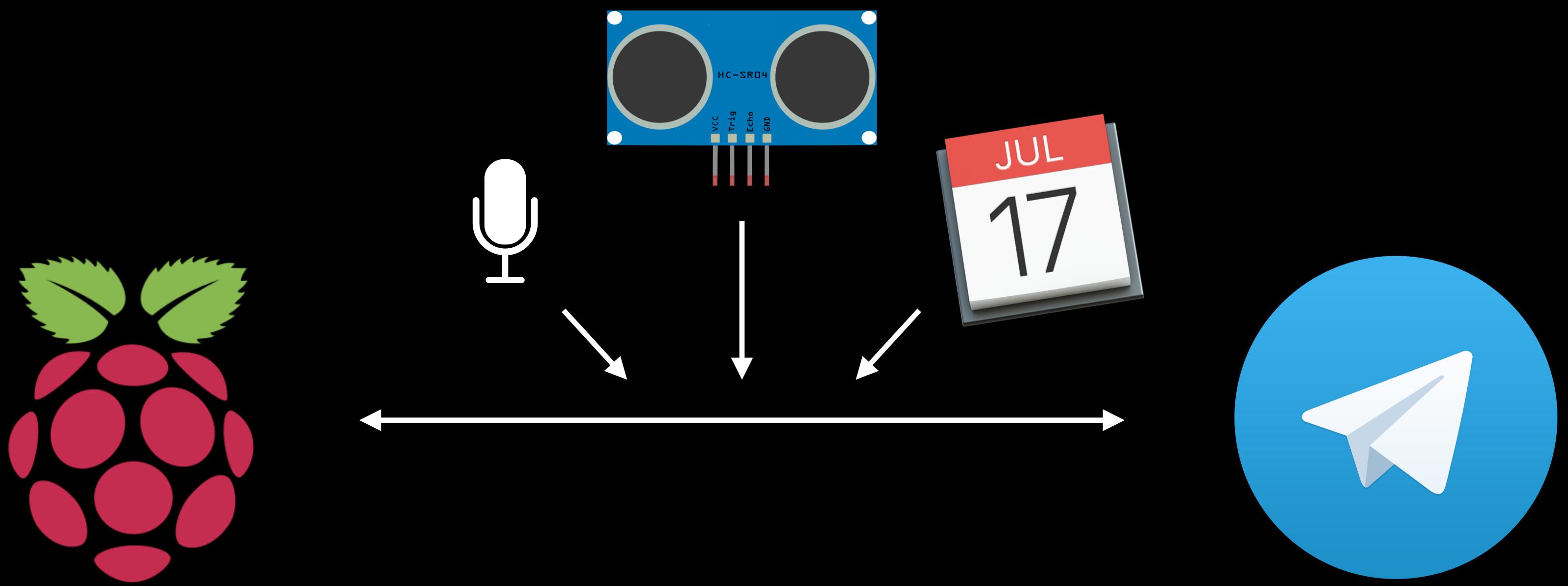
04b_implementacao.py

cópia
----->

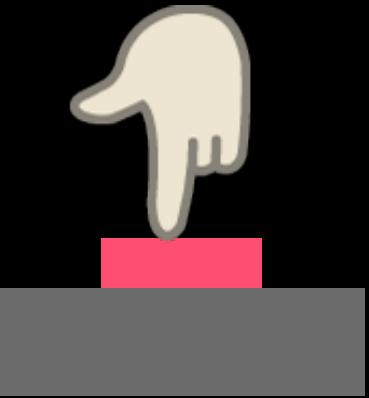
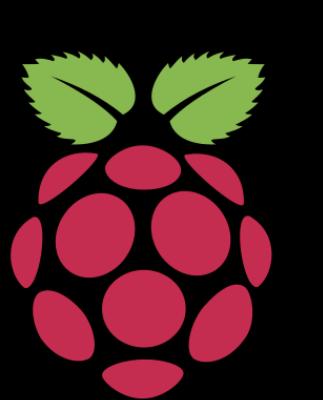


04c_aperfeicoamento.py

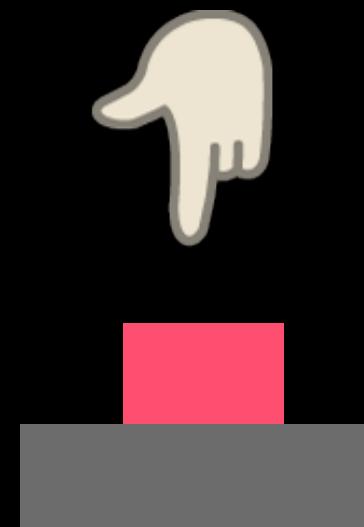
Cópia do Código da Implementação para o Aperfeiçoamento



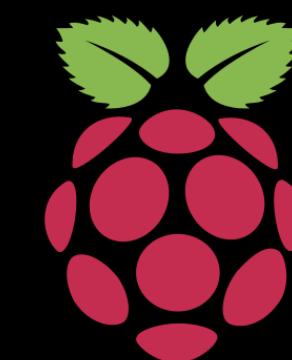
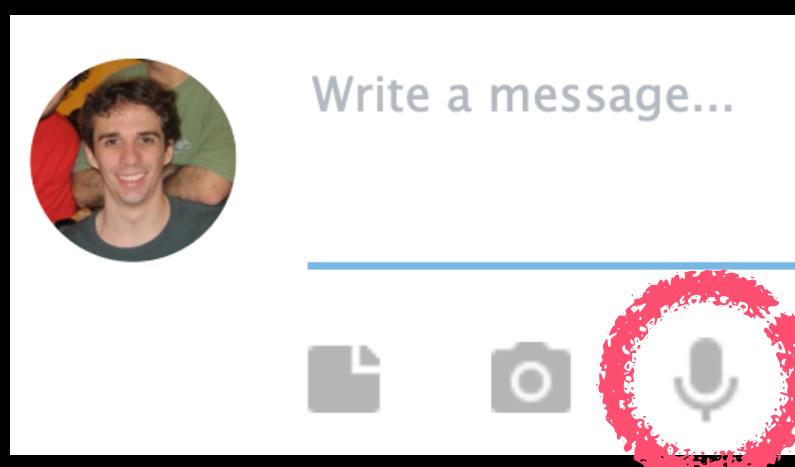
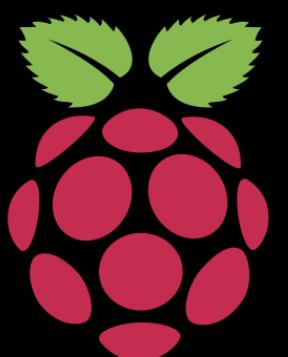
Inclusão de Áudio, Data/Horário e Distância na Comunicação



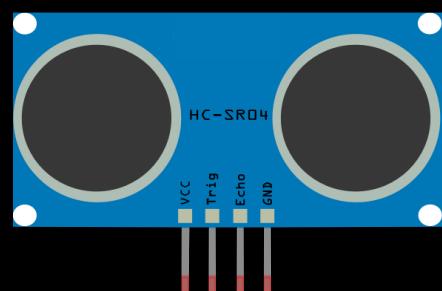
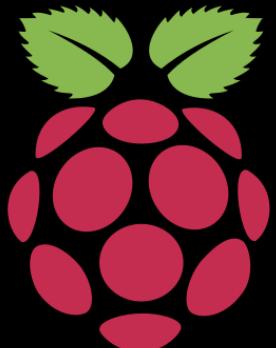
começa a gravar



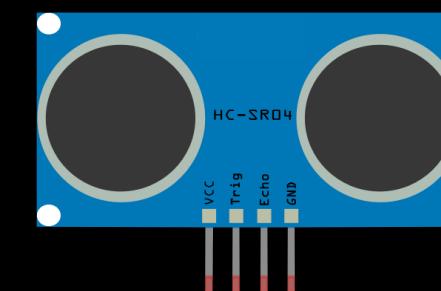
para de gravar,
converte para OGG e envia



baixa e toca no MPlayer



salva instante
de aproximação



verifica se intervalo
desde o instante salvo é
maior que 10 segundos

Etapas do Aperfeiçoamento



Aperfeiçoamento

Ao pressionar o Botão 3, comece a **gravar um áudio**. Ao soltar, **finalize a gravação**, converta o arquivo para OGG e envie-o para o chat.
↪ DICA: use a Popen com o arecord para gravar e depois a system com o opusenc para converter para ogg.

Ao receber um arquivo de áudio do chat, baixe-o e toque-o no Raspberry Pi.
↪ DICA: use o urlretrieve e o mplayer.

Caso a pessoa tenha **ficado ao menos 10 segundos em frente à porta e saído**, envie uma mensagem de alerta "Pessoa saiu".

↪ DICA: use as propriedades when_in_range e when_out_of_range, uma variável global de datetime para o momento de aproximação e a .total_seconds().

Desafio Extra



04c_aperfeicoamento.py

cópia
-----►



04d_desafio.py

Cópia do Código do Aperfeiçoamento para o Desafio

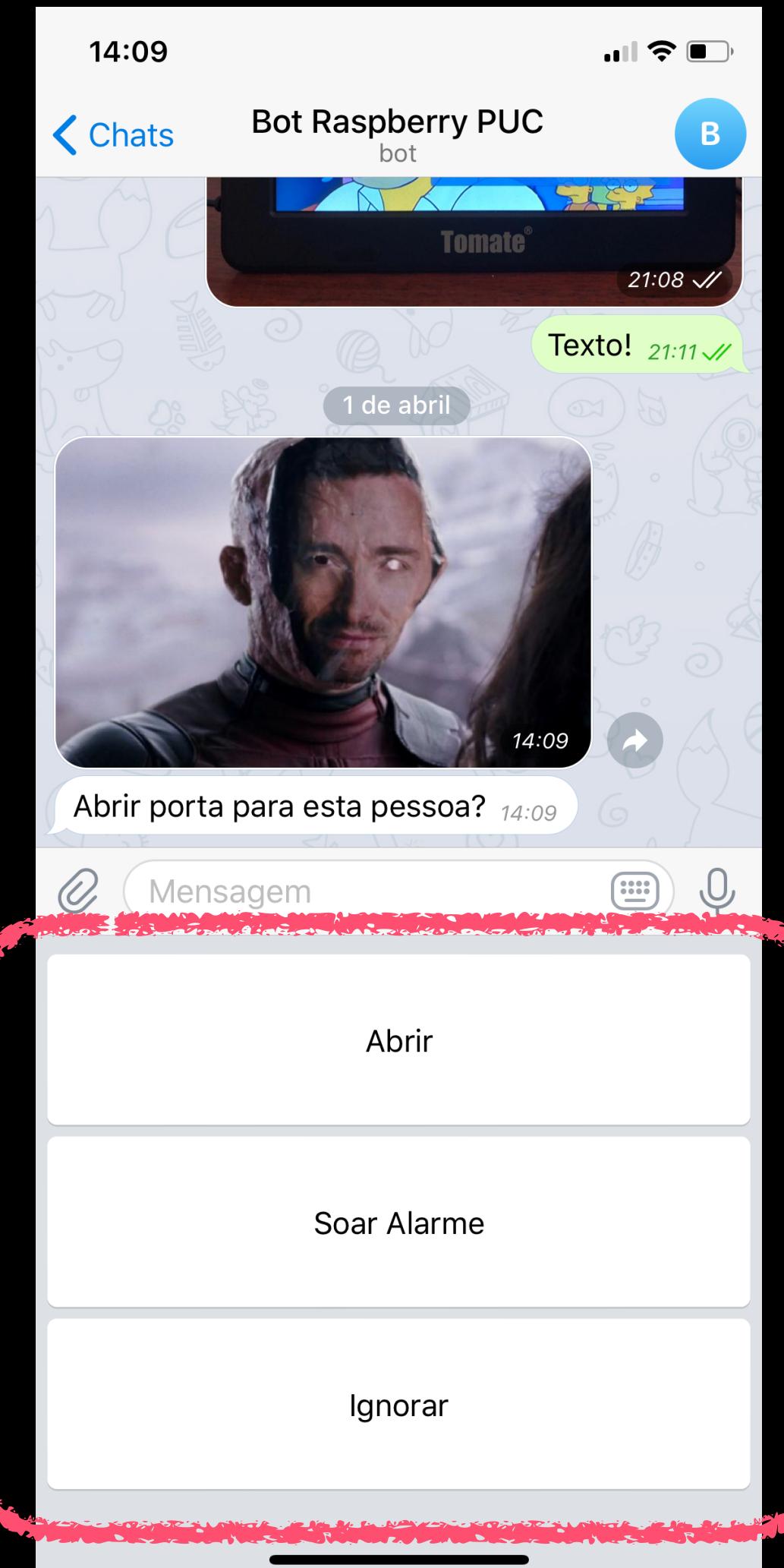
alerta piscante e sonoro



mensagem rolando



e



Teclado de Respostas Pré-Definidas e Outras Mensagens Recebidas no LCD



Desafio Extra

Ao receber um texto que não seja "Abrir", "Soar Alarme" ou "Ignorar", pisque "Mensagem Recebida" **duas vezes** no LCD, **soando a campainha** duas vezes (junto com as piscadas). Em seguida, **role a mensagem recebida** uma vez no display, **removendo os acentos** antes.

↪ DICA: você pode usar a função sleep. Pesquise no Google como remover os acentos de uma string em Python.

Ao enviar o texto de alerta após apertar o Botão 1, configure **opções pré-definidas de resposta** para "Abrir", "Soar Alarme" e "Ignorar". A opção de alarme deve soar a campainha.

↪ DICA: consulte a documentação do Telegram em core.telegram.org/bots/api#sendmessage e envie um dicionário como parâmetro extra nos dados da sendMessage.



janks.link/micro/projeto04.zip

Material do Projeto 04