

Projeto 03

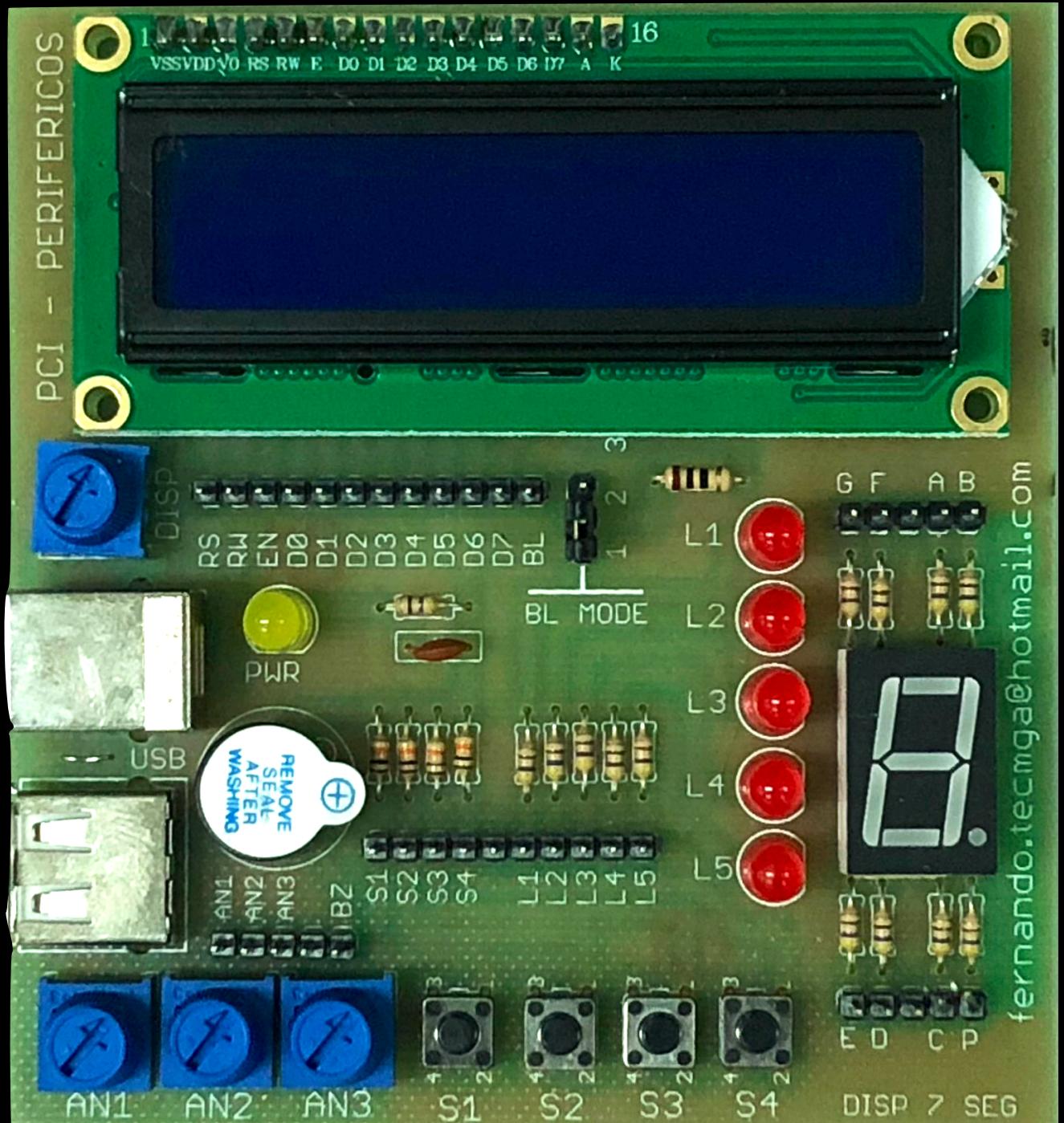
Controle de Acesso – Prática

Jan K. S. – janks@puc-rio.br

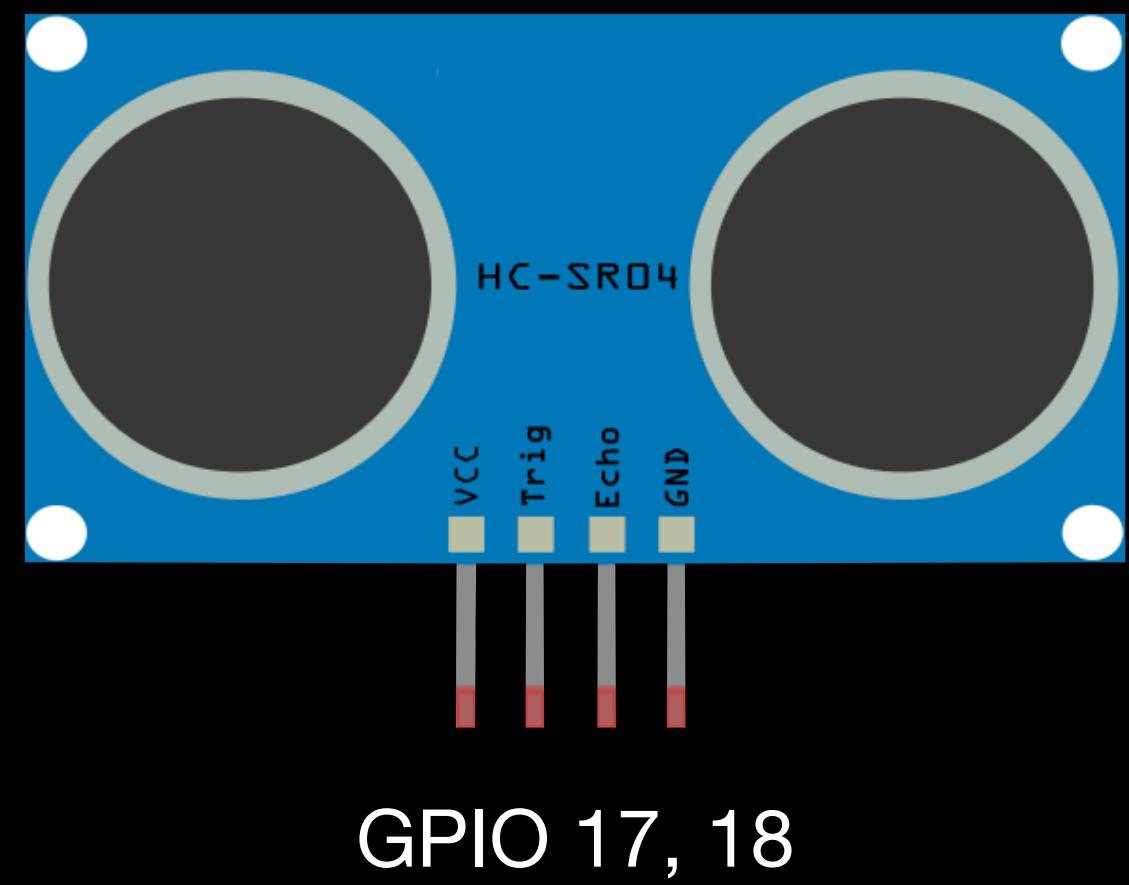
ENG1419 – Programação de Microcontroladores

Testes Iniciais

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



GPIO 21
GPIO 22
GPIO 23
GPIO 24
GPIO 25



GPIO 17, 18

Conexões com as Portas da GPIO

localhost:1234

The screenshot shows the adminMongo interface running on localhost:1234. The title bar indicates the URL is `localhost:1234/app/aula/projeto03/moradores/view/1`. The main title is "Database: projeto03 / Collection - moradores" with an edit icon. Below it, the navigation path is Home / aula (connection) / projeto03 (database) / moradores (collection). The left sidebar shows "Database Objects" with "aula" and "projeto03" selected, and "moradores" expanded. The right panel displays the "moradores" collection with 38 total records. It includes a search bar with "Search" and "Query" buttons, a "Reset" button, a "Docs per page" dropdown set to 10, and a "Delete all" button. Below the search bar, there are four document preview boxes, each with a "Delete" (red), "Link" (light blue), and "Edit" (green) button. The first document shows:

```
{  
    "_id": "5b8d7676ffffe954d7c8e5bc6",  
    "nome": "Rodrigo",  
    "apartamento": "101",  
    "senha": "101001"  
}
```

The second document shows:

```
{  
    "_id": "5b8d7676ffffe954d7c8e5bc7",  
    "nome": "Gabriela",  
    "apartamento": "101",  
    "senha": "101002"  
}
```

The third document shows:

```
{  
    "_id": "5b8d7676ffffe954d7c8e5bc8",  
    "nome": "Paul McCartney",  
    "apartamento": "102",  
    "senha": "102001"  
}
```

The fourth document is partially visible at the bottom.

Página para Visualizar os Dados do Banco



Testes Iniciais

Ao pressionar o botão 1, **toque uma vez** a campainha durante meio segundo.

Pisque 2 vezes o LED 1 sempre que algo se aproximar ou se afastar do sensor a uma distância de 10cm.

↪ **DICA:** use as propriedades `when_in_range` e `when_out_of_range`.

Ao pressionar o botão 2, **meça a distância** e exiba-a no LCD de caracteres em centímetros com uma casa decimal (ex: "6.4 cm").

Também ao pressionar o botão 2, **salve a distância numérica** e a **data/horário** em uma coleção do banco de dados. Verifique os dados inseridos acessando `localhost:1234` no navegador.

Implementação



Controle de Acesso de Moradores

banco "projeto03"
coleção "moradores"



```
[  
  {  
    "nome": "Rodrigo",  
    "apartamento": "101",  
    "senha": "101001"  
  },  
  {  
    "nome": "Gabriela",  
    "apartamento": "101",  
    "senha": "101002"  
  },  
  {  
    "nome": "Paul McCartney",  
    "apartamento": "102",  
    "senha": "102001"  
  },  
  {  
    "nome": "John Lennon",  
    "apartamento": "102",  
    "senha": "102002"  
  },  
  {  
    "nome": "George Harrison",  
    "apartamento": "102",  
    "senha": "102003"  
  },  
  {  
    "nome": "Ringo Star",  
    "apartamento": "102",  
    "senha": "102004"  
  },  
  ...  
]
```

`validar_apartamento("101")`

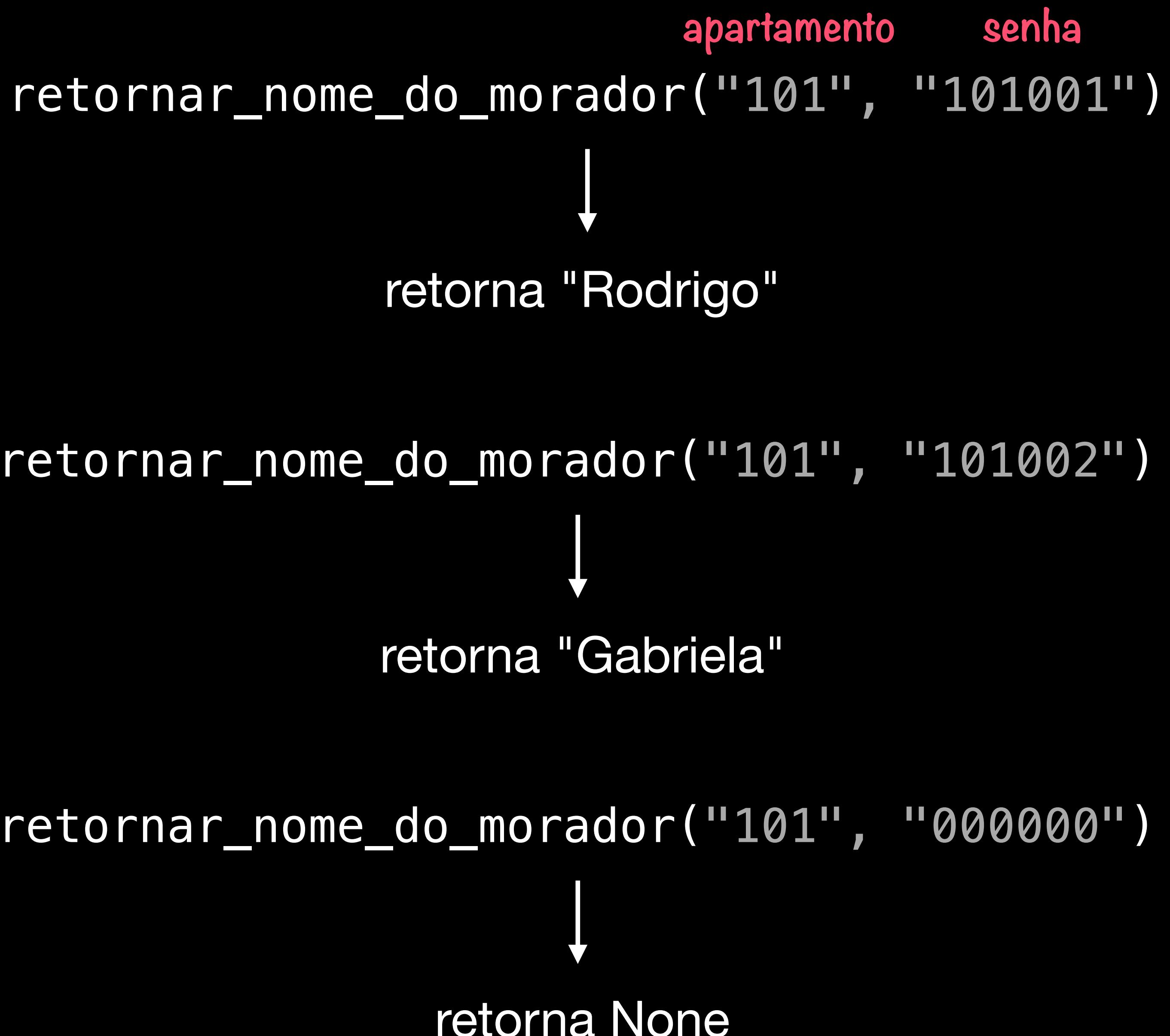


`retorna True`

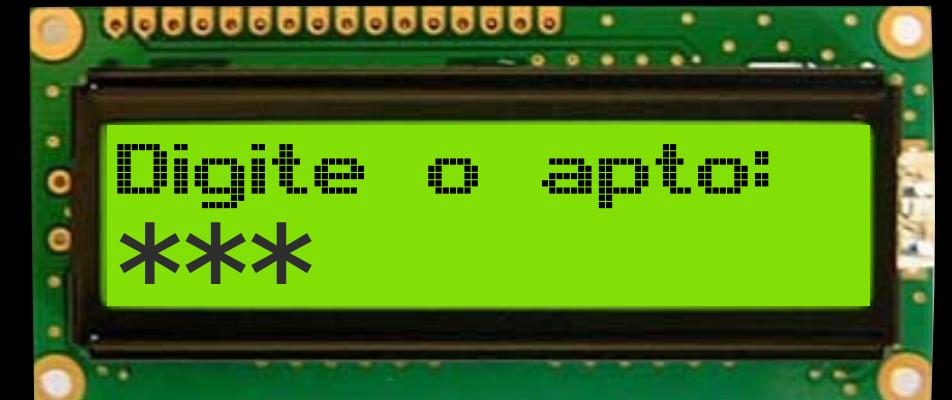
`validar_apartamento("000")`



`retorna False`



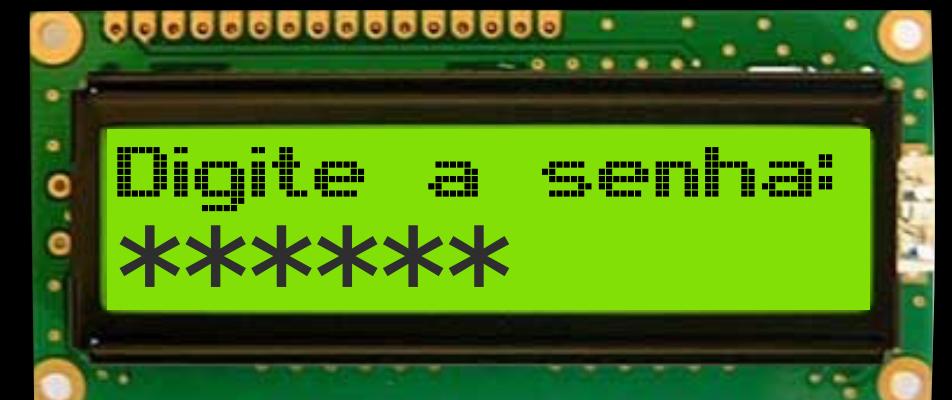
`coletar_digito("Digite o apto:")`



retorna "101", por exemplo



`coletar_digito("Digite a senha:")`



retorna "101001", por exemplo





| | | |
|----------|----------|-----------|
| | | KEY_UP |
| KEY_LEFT | KEY_OK | KEY_RIGHT |
| | KEY_DOWN | |
| | KEY_1 | KEY_2 |
| | KEY_4 | KEY_5 |
| KEY_7 | KEY_8 | KEY_9 |
| | KEY_0 | |

Códigos dos Botões do Mini Controle Remoto

Exibe Mensagem



Detecta Dígito



É tecla OK?

Sim

Retorna Sequência
de Dígitos

Não

- Concatena dígito com sequência
- Exibe asterisco

Lógica de Digitação de Números

```
if codigo == ["KEY_1"]:  
    ...  
  
elif coding == ["KEY_2"]:  
    ...  
  
elif codin == "KEY_3":  
    ...  
  
elif coding == ["KEY_9"]:  
    ...  
  
elif coding == ["KEY_0"]:  
    ...
```



["KEY_2']

Posso tentar
acessar diretamente
o caracter do dígito



The screenshot shows a code editor with a script named `teste.py` and a terminal window below it.

teste.py:

```
10     return False
11
12 def retornar_nome_do_morador(apto, senha):
13     if apto == "101" and senha == "101001":
14         return "Gabriela"
15     else:
16         return False
```

Shell:

```
>>> %Run teste.py
>>> validar_apartamento("101")
True
>>> validar_apartamento("000")
False
>>> retornar_nome_do_morador("101", "101001")
'Gabriela'
>>> retornar_nome_do_morador("101", "000000")
>>> coletar_digitos("Digite o apto:")
'202'
>>> |
```

Teste das Funções na Tela de Comandos



Digite o apto:

Digite a senha:

Apartamento
invalido:/

Bem-vindo(a)
Gabriela:/

Acesso negado

Parte 4: Controle de Acesso Completo



Implementação

Crie a função `validar_apartamento` que receba o texto com o número do apartamento. Retorne True ou False caso ele exista ou não. Teste a função.

↪ DICA: use a função `find_one`.

Crie a função `retornar_nome_do_morador` que receba o texto com o número do apartamento e a senha. Retorne o nome do morador se a combinação existir no banco ou None caso contrário. Teste a função.

Crie a função `coletar_digitos` que receba uma mensagem, imprima-a no LCD, peça os dígitos do controle remoto (seguindo o layout do slide anterior) e retorne o texto com os dígitos teclados. Teste a função.

↪ DICA: use um `while True` e o acesso a um caracter da string com `texto[posicao]`.

Use as três funções acima para `continuamente solicitar` o número do apartamento, validar esse número, solicitar a senha e validar o acesso.

↪ DICA: use um outro `while True`.

Aperfeiçoamento



03b_implementacao.py

cópia
----->



03c_aperfeicoamento.py

Cópia do Código da Implementação para o Aperfeiçoamento

The screenshot shows the Thonny IDE interface. The top bar displays the title "Thonny - /Users/Jan/Library/Mobile Documents/com~apple~CloudDocs/PUC/Microcontroladores/Projeto...". Below the title is a toolbar with various icons. The main window contains a code editor titled "03b_implementacao.py" and a shell window below it.

The code editor contains the following Python script:

```
1 # importação de bibliotecas
2 from extra.redefinir_banco import redefinir_banco
3 from pymongo import MongoClient
4
5 # a linha abaixo apaga todo o banco e reinsere os moradores
6 redefinir_banco()
7
8 # parâmetros iniciais do banco
9 cliente = MongoClient("localhost", 27017)
10 banco = cliente["projeto03"]
11 colecao = banco["moradores"]
```

The line `redefinir_banco()` is highlighted with a red rectangular box.

The shell window below the code editor has the tab "Shell" selected and contains the text ">>>".

Comando para Recriar o Banco de Dados

Ao pressionar as teclas do controle remoto, **emita um "beep" rápido**. Ao final de um acesso inválido, emita um outro efeito sonoro.

Comece a pedir o número do apartamento apenas **se alguém se aproximar a pelo menos 20 cm do sensor**.



Aperfeiçoamento

Salve cada tentativa de acesso em uma outra coleção, com o número do apartamento, a data/hora da tentativa e o nome do morador – não inclua esse campo nome caso a senha esteja incorreta.

Ao apertar o Botão 1, peça o número do apartamento. Em seguida, **imprima no terminal** o nome e a data/hora de **cada tentativa de acesso** a ele, com data/hora em ordem decrescente

↪ DICA: use a função `strftime`.

25/04 (05:37): Rodrigo

23/04 (22:39): Gabriela

23/04 (13:43): SENHA INCORRETA

...

Desafio Extra



03c_aperfeicoamento.py

cópia
-----►



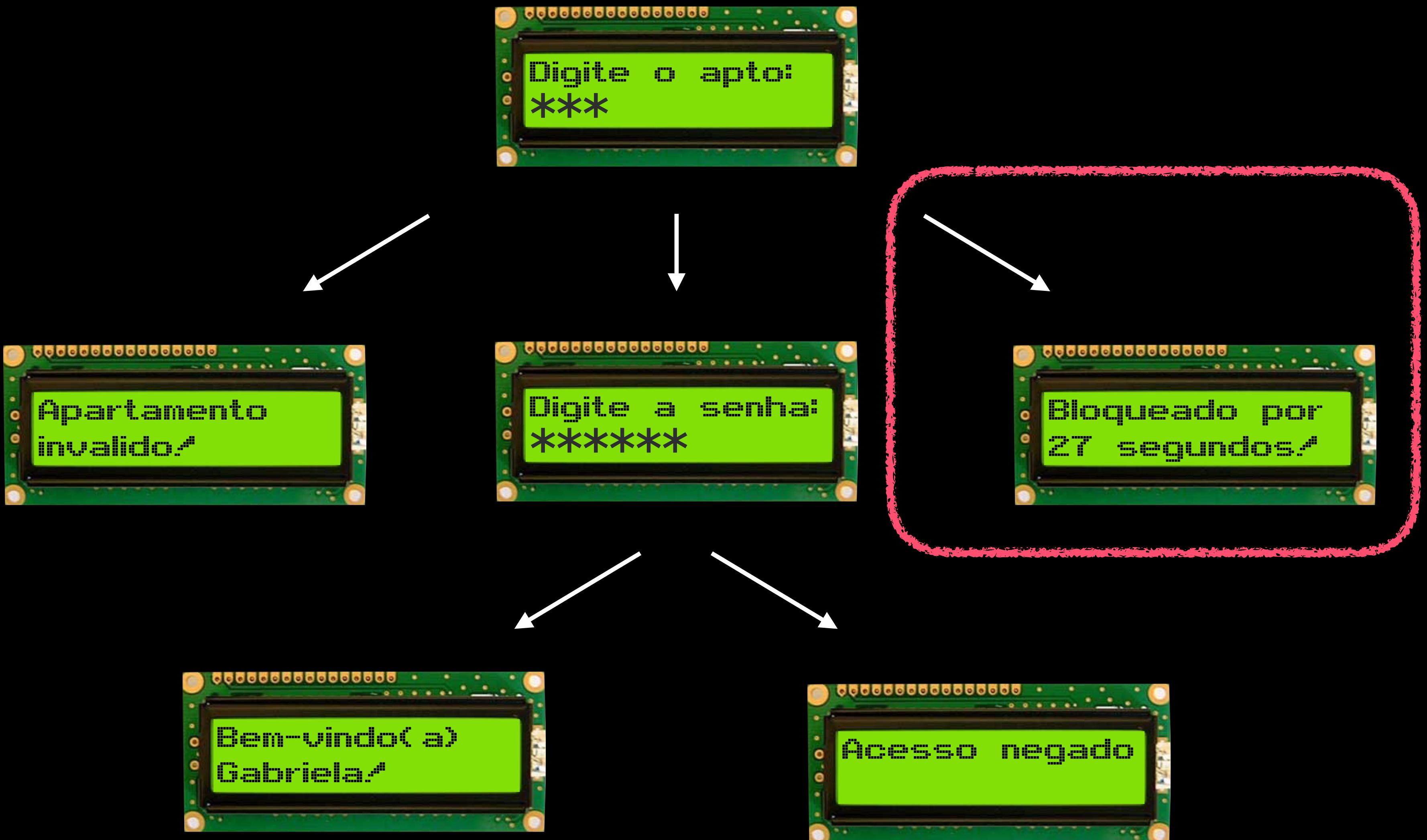
03d_desafio.py

Cópia do Código do Aperfeiçoamento para o Desafio

muitas tentativas incorretas



Bloqueio Após Tentativas Incorretas



Controle de Acesso com Bloqueio Temporário

| Data/Horário | Tentativa | Penalidade | Bloqueio |
|--------------------|--------------|--------------|------------------------|
| 03/09, às 14:59:00 | 1ª incorreta | -- | -- |
| 03/09, às 14:59:30 | 2ª incorreta | -- | -- |
| 03/09, às 15:00:00 | 3ª incorreta | 15 segundos | até 03/09, às 15:00:15 |
| 03/09, às 15:00:10 | blockeado! | -- | até 03/09, às 15:00:15 |
| 03/09, às 15:00:30 | 4ª incorreta | 30 segundos | até 03/09, às 15:01:00 |
| 03/09, às 15:05:00 | 5ª incorreta | 45 segundos | até 03/09, às 15:05:45 |
| 03/09, às 15:05:40 | correta | (reiniciada) | |
| 03/09, às 17:00:00 | 1ª incorreta | -- | |
| 06/09, às 22:00:00 | 2ª incorreta | -- | |
| 16/11, às 23:00:00 | 3ª incorreta | 15 segundos | até 16/11, às 23:00:15 |
| | | ... | |

Exemplo de Aumento do Tempo Bloqueio para um Apartamento

```
{  
    "data": datetime(2018, 8, 20, 15, 30, 45),  
    "apartamento": "101",  
    "tentativas incorretas": 3  
}  
  
{  
    "data": datetime(2018, 8, 20, 15, 31, 17),  
    "apartamento": "101",  
    "tentativas incorretas": 4  
}  
  
{  
    "data": datetime(2018, 8, 20, 15, 30, 45),  
    "apartamento": "202",  
    "tentativas incorretas": 0  
}
```



Desafio Extra

Registre as informações de tentativas incorretas na função `retornar_nome_do_morador`, preenchendo adequadamente os campos para o caso de senha correta ou incorreta.
↪ DICA: no caso de senha incorreta, carregue o dado mais recente para o apartamento em questão.

Na função `validar_apartamento`, verifique o dado mais recente de tentativas incorretas para o apartamento fornecido. Se for maior que 2, calcule o número de segundos restantes do bloqueio. Se esse tempo for maior que 0, retorne-o em vez do True/False de antes.

Verifique o retorno da `validar_apartamento` e, caso seja numérico, exiba a mensagem de bloqueio (com o tempo restante em segundos) em vez de pedir a senha.



janks.link/micro/projeto03.zip

Material do Projeto 03