

Projeto 03

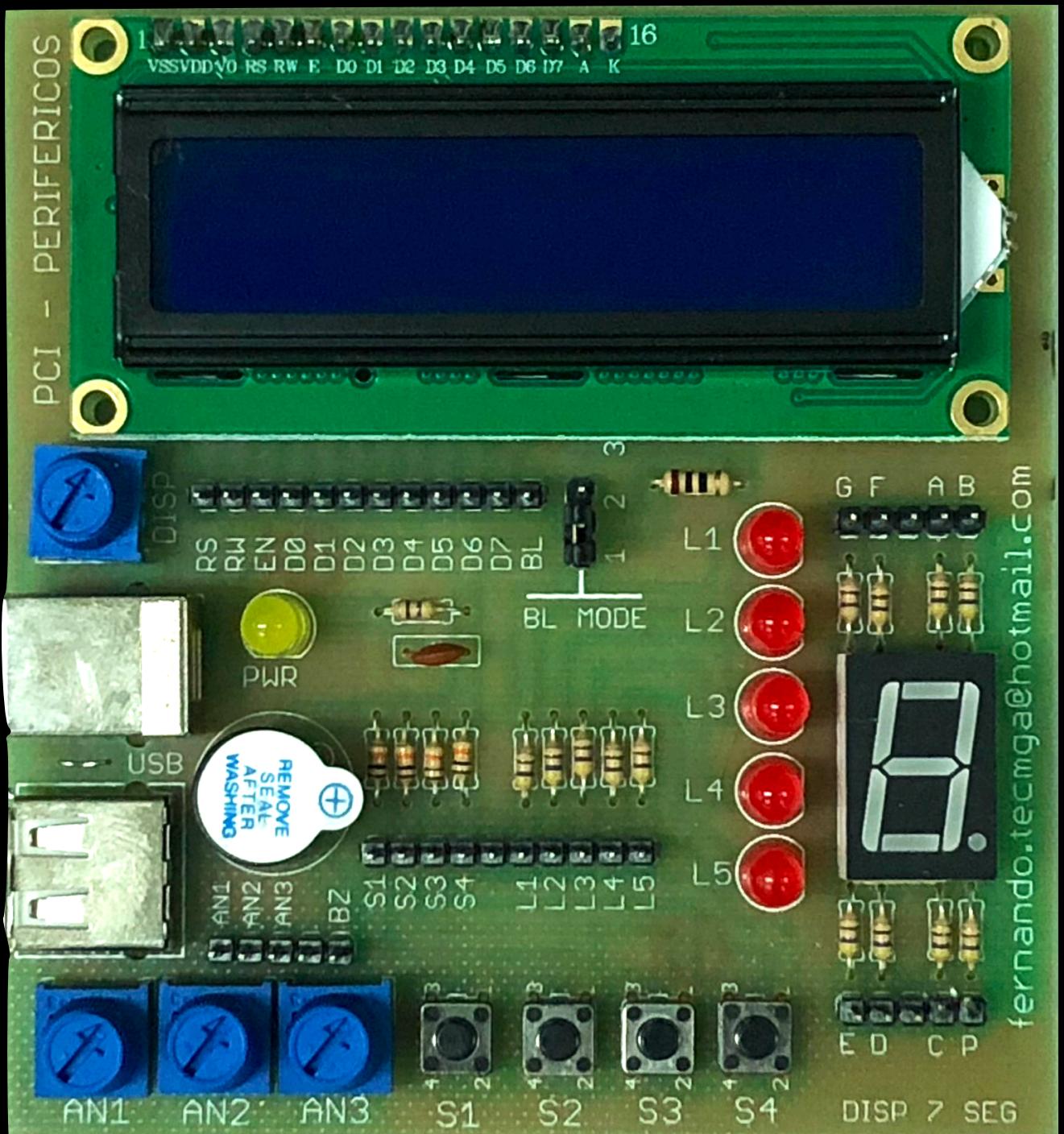
Controle de Acesso – Prática

Jan K. S. – janks@puc-rio.br

ENG1419 – Programação de Microcontroladores

Testes Iniciais

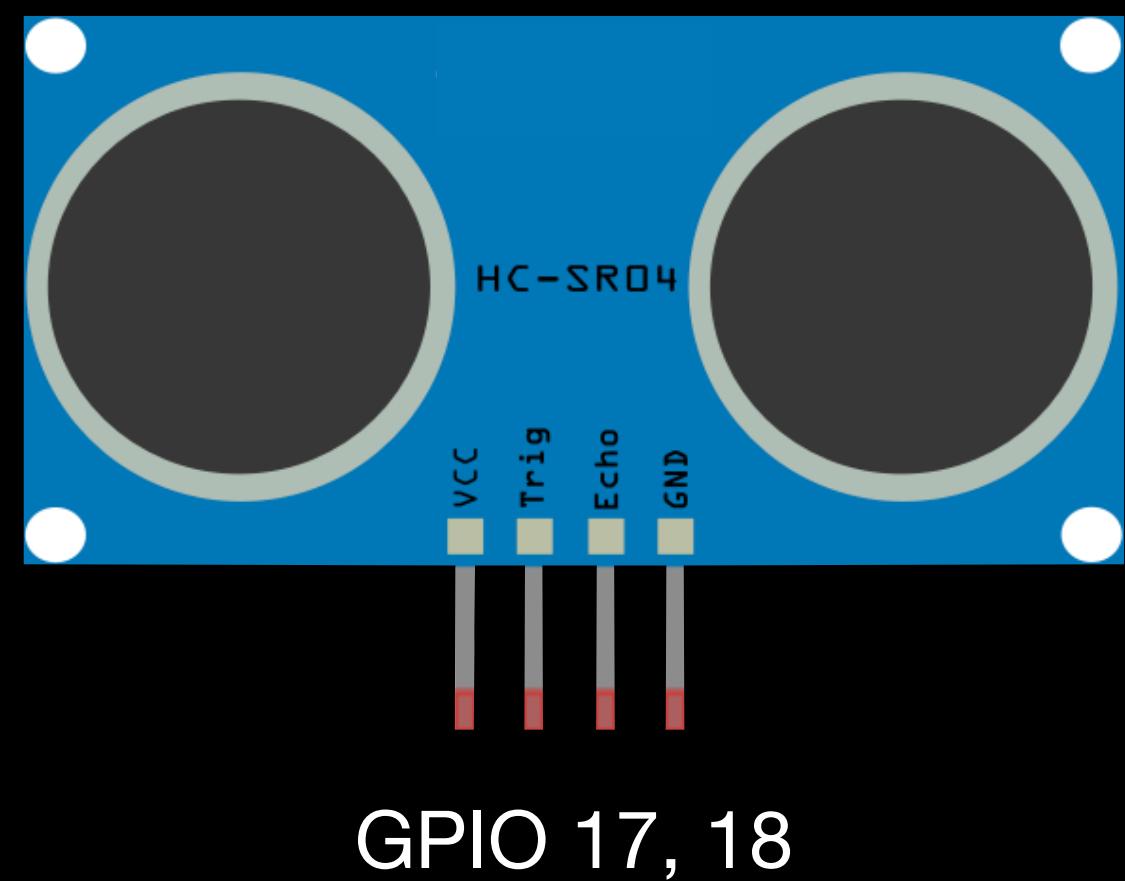
GPIO 2, 3, 4, 5, 6, 7



GPIO 16

GPIO 11, 12, 13, 14

GPIO 21
GPIO 22
GPIO 23
GPIO 24
GPIO 25



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 entries, each with a delete, link, and edit button. The first document's data is:

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

The second document's data is:

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

The third document's data is:

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

The fourth document's data is partially visible:

```
{  
    ...  
}
```

Página para Visualizar os Dados do Banco



Testes Iniciais

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

↪ DICA: use a função beep.

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 o **datetime de agora** 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`

número do apartamento senha

retornar_nome_do_morador("101", "101001")



retorna "Rodrigo"

retornar_nome_do_morador("101", "101002")



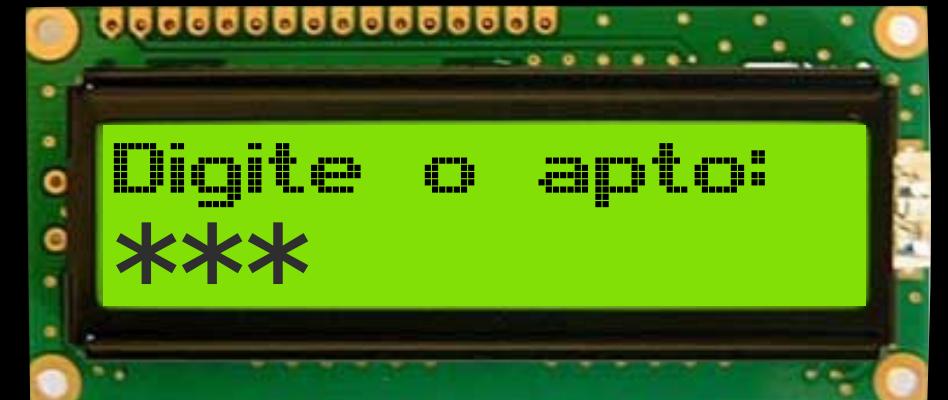
retorna "Gabriela"

retornar_nome_do_morador("101", "000000")



retorna None

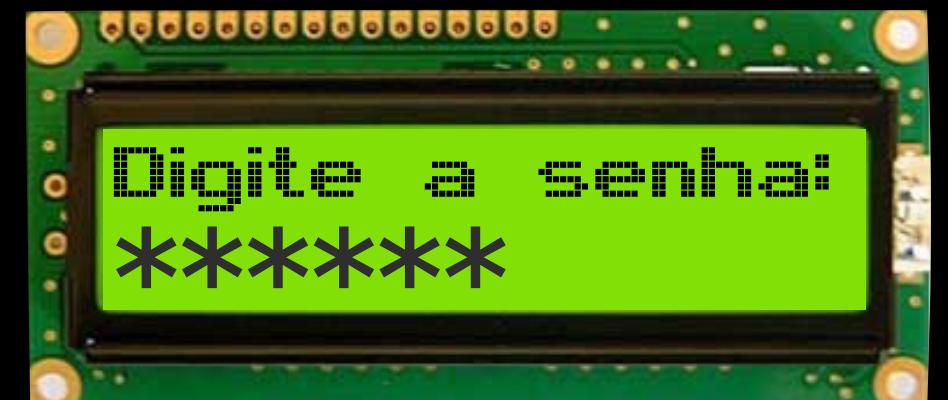
`coletar_digitos("Digite o apto:")`



retorna "101", por exemplo



`coletar_digitos("Digite a senha:")`



retorna "101001", por exemplo





		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

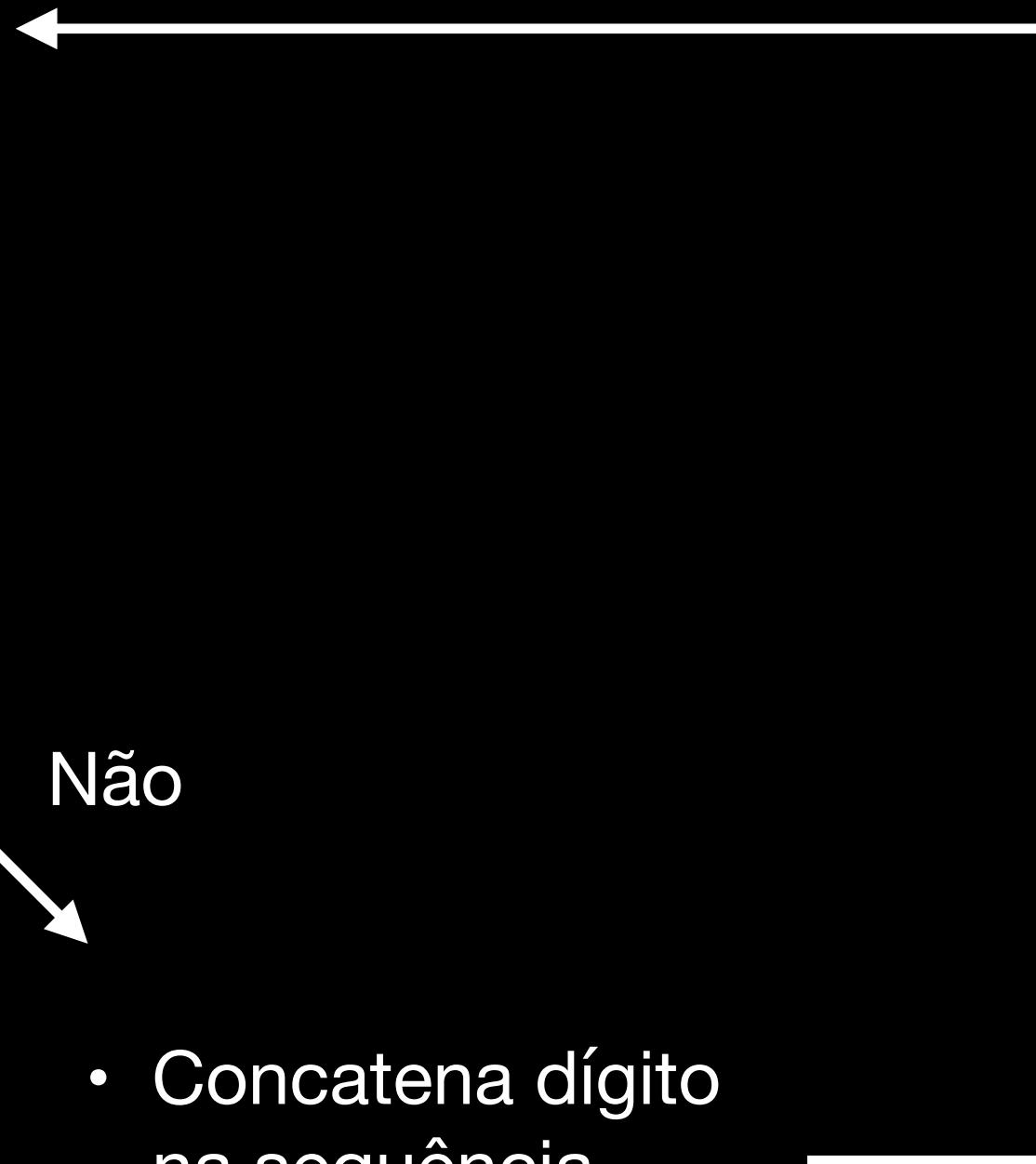
Exibe Mensagem



Inicia string vazia para sequencia de dígitos



Detecta Dígito



Retorna Sequência de Dígitos

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"]:  
    ...
```

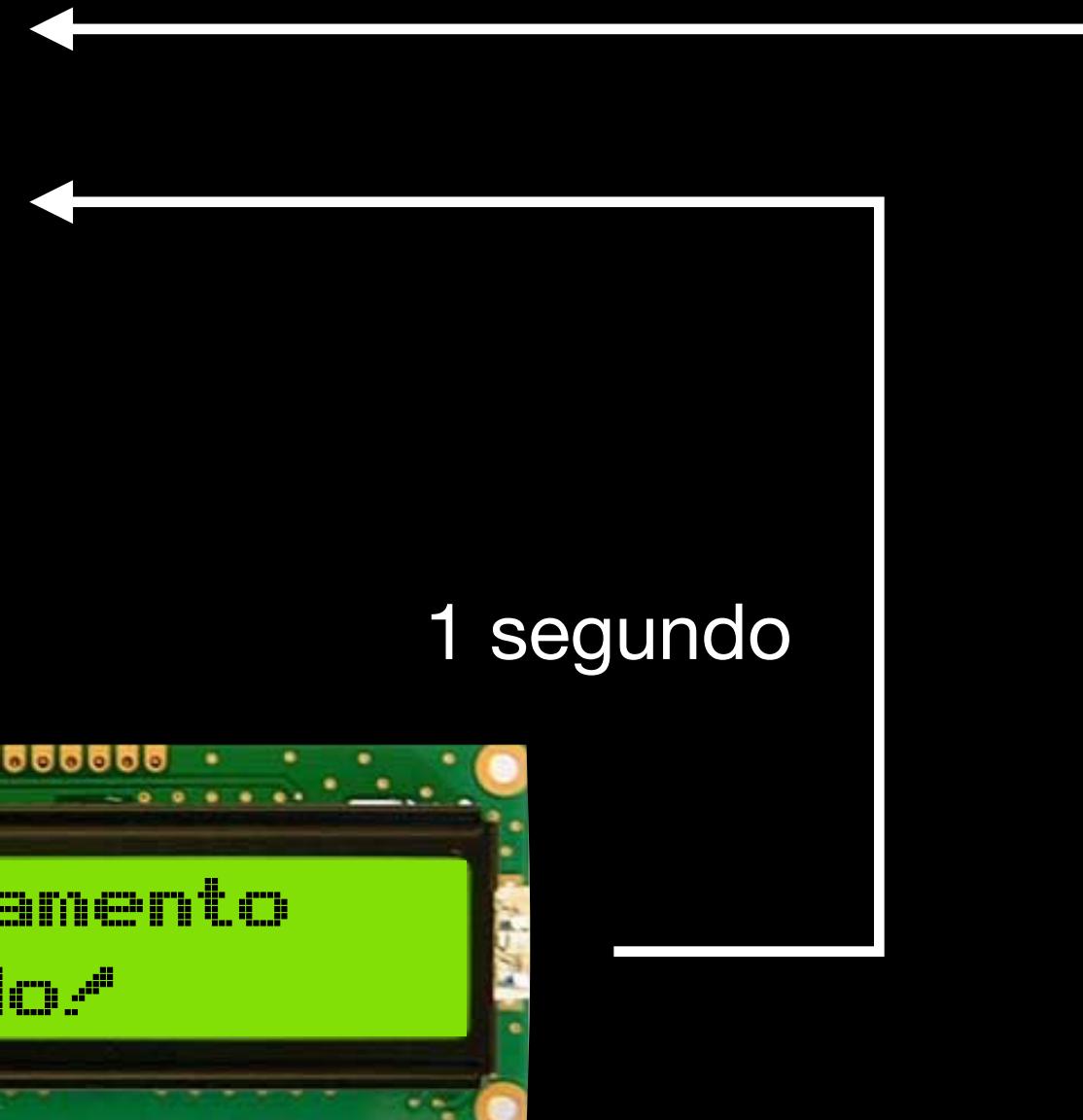


["KEY_2']

Posso tentar
acessar diretamente
o caracter do dígito



```
>>> print( validar_apartamento("102") )  
>>> print( validar_apartamento("000") )  
>>> print( retornar_nome_do_morador("102", "102001") )  
>>> print( retornar_nome_do_morador("102", "00") )  
>>> print( coletar_digitos("Digite o apto:") )  
>>> print( coletar_digitos("Digite a senha:") )
```



1 segundo



1 segundo

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 chamando com alguns valores de apartamento e dando print.

↪ DICA: use a `find_one`.

Crie a função `retornar_nome_do_morador` que receba o texto com o número do apartamento e o texto com 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: dentro da função, use o exemplo do `while True` com `nextcode` do controle remoto 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 o outro `while True` no final do script.

Aperfeiçoamento



03b_implementacao.py

cópia
----->



03c_aperfeicoamento.py

Cópia do Código da Implementação para o Aperfeiçoamento

The screenshot shows the Thonny Python IDE interface. The top bar displays the path: Thonny - /Users/Jan/Library/Mobile Documents/com~apple~CloudDocs/PUC/Microcontroladores/Projeto... The main window contains a code editor with the file 03b_implementacao.py open. The code imports the redefinir_banco function from extra.redefinir_banco and MongoClient from pymongo. It then calls redefinir_banco() to clear and recreate the database. The code editor has line numbers on the left and syntax highlighting. A red box highlights the line 'redefinir_banco()'. Below the code editor is a shell window labeled 'Shell' containing the prompt '>>>'. The status bar at the bottom shows the message 'Python 3.7.3 (v3.7.3:ef4ec6e, Mar 25 2019, 16:57:53) [Clang 4.0.1 (tags/RELEASE_401/final)] on darwin'. The title bar also includes standard OS X window controls.

```
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"]
12
13
```

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.



Aperfeiçoamento

Comece a pedir o número do apartamento apenas **se alguém se aproximar a pelo menos 10 cm do sensor**.

↪ DICA: use a `when_in_range`, removendo o código anterior do `while True` do final do script e colocando numa função.

Salve cada tentativa de acesso em uma outra coleção, com o número do apartamento, o `datetime` da tentativa e o nome do morador – não inclua esse campo nome caso a senha esteja incorreta. Teste e verifique o banco de dados.

↪ DICA: use a `insert`.

Ao apertar o Botão 1, peça o número do apartamento. Em seguida, **busque no banco as tentativas** naquele apartamento, com **`datetime` em ordem decrescente**, e imprima-as no terminal (Shell) o nome e a data/hora, no formato abaixo.

↪ DICA: use a `find` e a `strftime`.

25/04 (05:37): Rodrigo

23/04 (22:39): Gabriela

23/04 (13:43): SENHA INCORRETA

...

Desafio Extra



03c_aperfeicoamento.py

cópia
----->



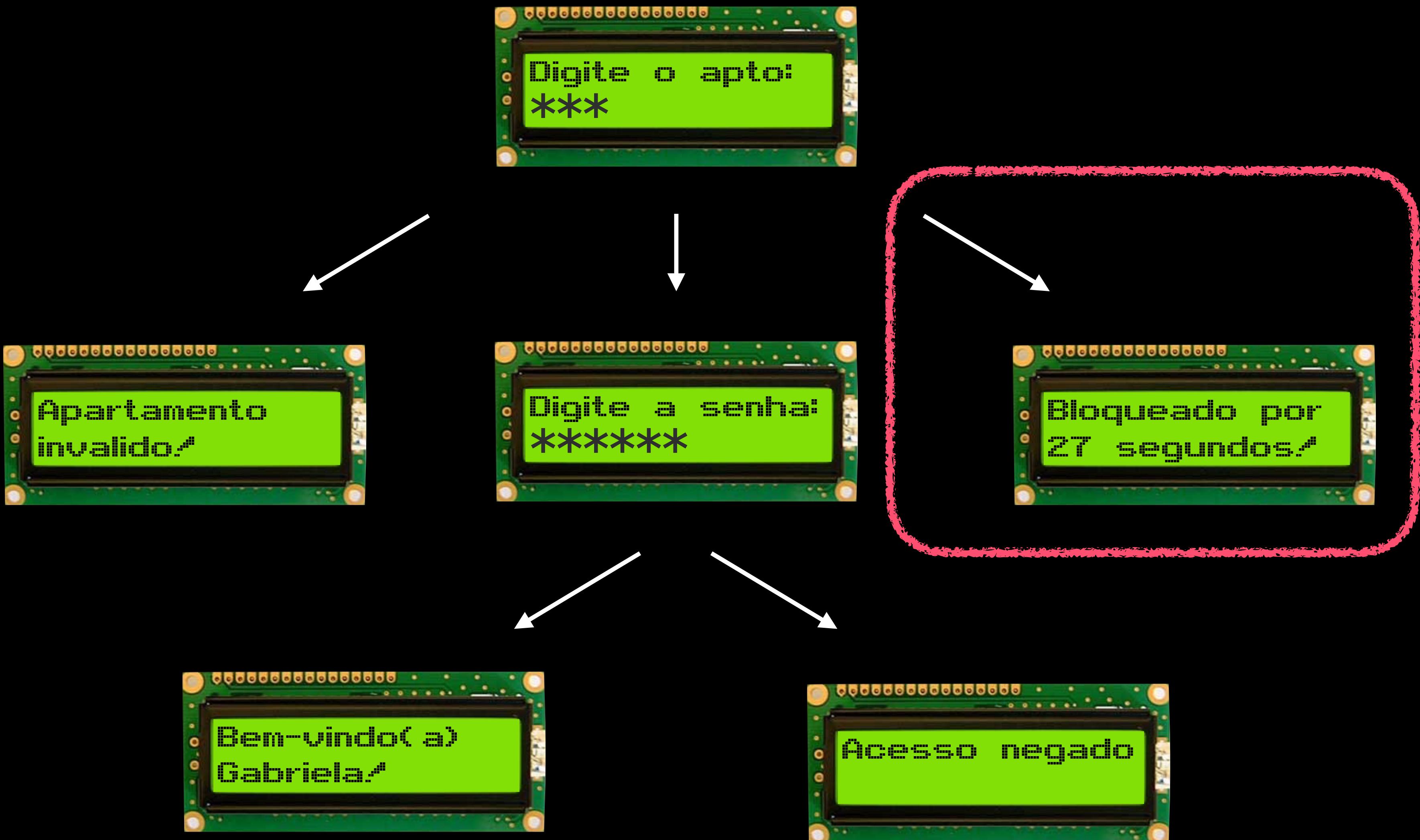
03d_desafio.py

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	1 ^a incorreta	--	--
03/09, às 14:59:30	2 ^a incorreta	--	--
03/09, às 15:00:00	3 ^a 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	4 ^a incorreta	30 segundos	até 03/09, às 15:01:00
03/09, às 15:05:00	5 ^a incorreta	45 segundos	até 03/09, às 15:05:45
03/09, às 15:06:00	correta	(reiniciada)	
03/09, às 17:00:00	1 ^a incorreta	--	
06/09, às 22:00:00	2 ^a incorreta	--	
16/11, às 23:00:00	3 ^a 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

Na função `retornar_nome_do_morador`, **registre as informações de tentativas incorretas numa nova coleção**, 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