

# Projeto 03

Controle de Acesso – Prática

Jan K. S. – [janks@puc-rio.br](mailto:janks@puc-rio.br)

ENG1419 – Programação de Microcontroladores

# Testes Iniciais

# localhost:1234

The screenshot shows the adminMongo web interface in a browser window. The address bar displays 'localhost:1234/app/aula/projeto03/moradores/view/1'. The interface has a dark sidebar on the left with the 'adminMongo' logo and navigation options: 'Database Objects' (with 'aula' and 'projeto03' sub-items) and 'Connections' (with 'aula' sub-item). The main area is titled 'Database: projeto03 / Collection - moradores'. Below the title is a breadcrumb trail: 'Home / aula (connection) / projeto03 (database) / moradores (collection)'. A toolbar contains buttons for 'New document', 'Indexes', 'Search', 'Query', and 'Reset'. On the right, there's a 'Docs per page' dropdown set to '10' and a 'Total records: 38' indicator with a 'Delete all' button. The data is displayed as a list of JSON documents. Each document is shown in a light blue box with a dropdown arrow on the right. To the right of each document box are three buttons: 'Delete' (red), 'Link' (blue), and 'Edit' (green).

Document	Actions
<pre>{   "_id": "5b8d7676fffe954d7c8e5bc6",   "nome": "Rodrigo",   "apartamento": "101",   "senha": "101001" }</pre>	Delete Link Edit
<pre>{   "_id": "5b8d7676fffe954d7c8e5bc7",   "nome": "Gabriela",   "apartamento": "101",   "senha": "101002" }</pre>	Delete Link Edit
<pre>{   "_id": "5b8d7676fffe954d7c8e5bc8",   "nome": "Paul McCartney",   "apartamento": "102",   "senha": "102001" }</pre>	Delete Link Edit

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





501 	502 	503 
401 	402 	403 
301 	302 	303 
201 	202 	203 
101 	102 	103 

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"
  },
  ...
]
```

Banco de Dados de Moradores

```
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

Parte 2: Nome do Morador

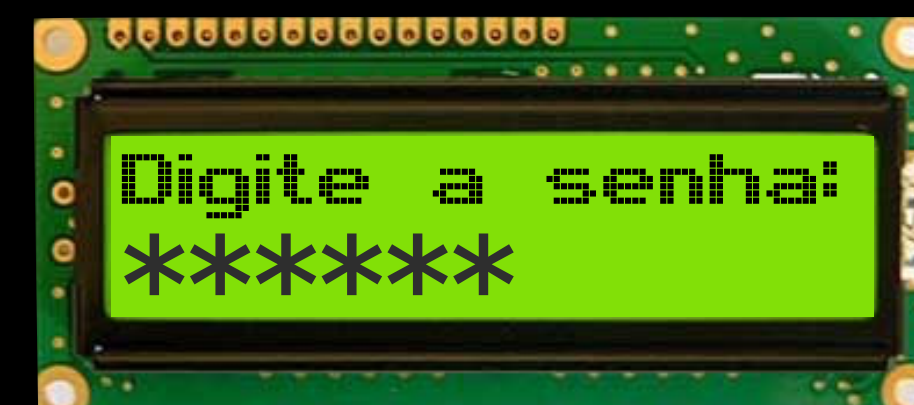
`coletar_digitos("Digite o apto:")`



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



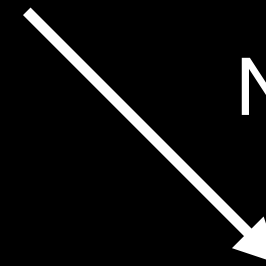
É tecla OK?

Sim

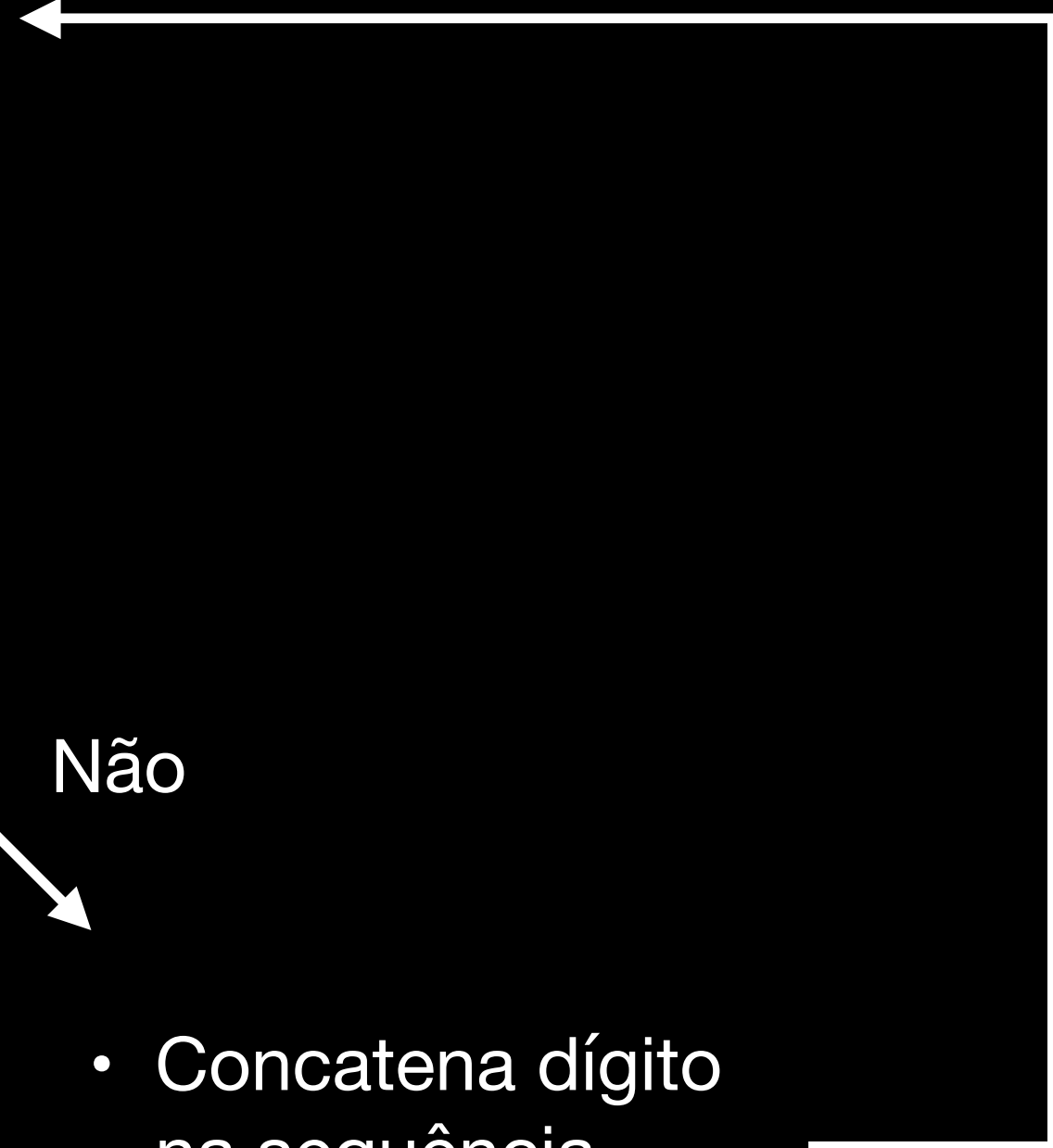


Retorna Sequência  
de Dígitos

Não



- Concatena dígito na sequência
- Exibe asterisco



Lógica de Digitação de Números

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



["KEY\_2"]

Posso tentar  
acessar diretamente  
o caracter do dígito

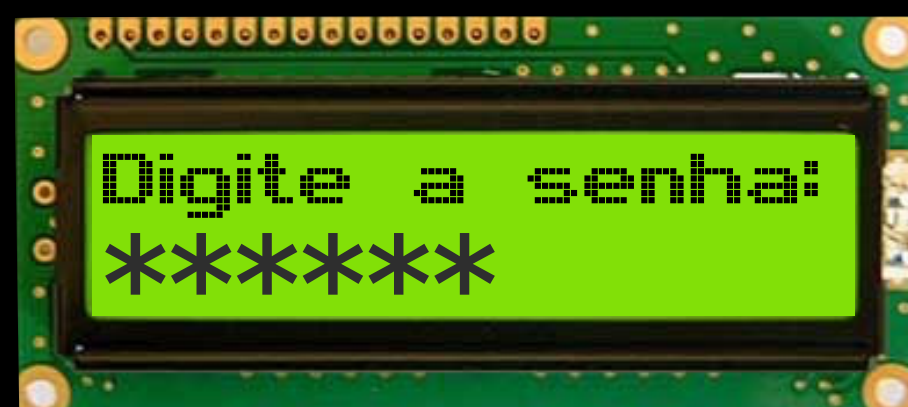


Captura Mais Otimizada da Tecla

```
>>> 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:") )
```

Teste das Funções





1 segundo

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

Thonny - /Users/Jan/Library/Mobile Documents/com~apple~CloudDocs/PUC/Microcontroladores/Projeto...

03b\_implementacao.py x

```
1 # importação de bibliotecas
2 from extra.redefinir_banco import redefinir_banco
3 from pymongo import MongoClient
4
5
6 # a linha abaixo apaga todo o banco e reinsere os moradores
7 redefinir_banco()
8
9 # parâmetros iniciais do banco
10 cliente = MongoClient("localhost", 27017)
11 banco = cliente["projeto03"]
12 colecao = banco["moradores"]
13
```

Shell

```
>>>
```

Comando para Recriar o Banco de Dados





## Aperfeiçoamento

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 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  
-----▶



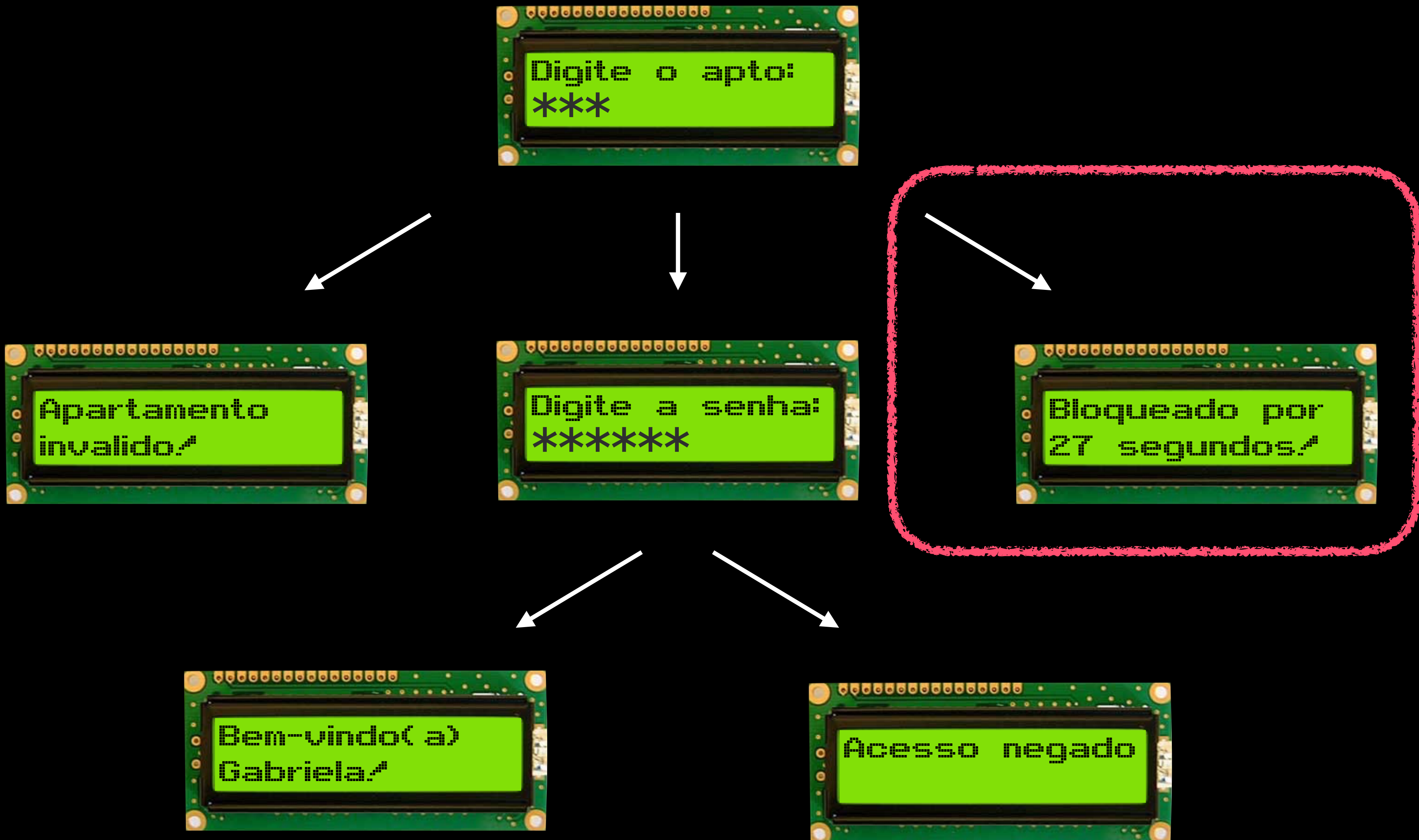
03d\_desafio.py

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	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	bloqueado!	--	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 15:06:00	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
}
```

Nova Coleção com Informações de Tentativas Incorretas





## 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.

Possíveis Melhorias



## Possíveis Melhorias

Como pedir o apartamento somente com a proximidade + depois que a porta for fechada?

Como interromper a coleta de dígitos caso o usuário fique muito tempo sem digitar nada?



[janks.link/micro/projeto03.zip](https://janks.link/micro/projeto03.zip)

Material do Projeto 03