Para criar um claviculário digital que faça o registro de quais chaves cada usuário retirou e/ou repôs, utilizaremos um Raspberry Pi como controlador principal. Este sistema contará com um teclado numérico para a entrada dos IDs dos usuários, um LCD 24x2 via I2C para mostrar informações ao usuário, e armazenará os registros em um arquivo de texto simples ou, preferencialmente, em um banco de dados leve como SQLite. A seguir, detalharemos o código em Python, as recomendações de nomeação e uma descrição para o README.

### Nome do Projeto

- `DigitalKeyCabinet`

### Código Python

Primeiro, é necessário instalar algumas dependências:

```bash

pip install RPi.GPIO

pip install Adafruit\_CharLCD

pip install python-sqlite

```

Agora, vamos ao código:

```python

import RPi.GPIO as GPIO

import time

from Adafruit\_CharLCD import Adafruit\_CharLCD

import sqlite3

from datetime import datetime

# Configuração inicial do LCD via I2C

lcd = Adafruit\_CharLCD(rs=1, en=2, d4=3, d5=4, d6=5, d7=6, cols=24, lines=2)

lcd.begin(24, 2)

# Configuração do teclado numérico

MATRIX = [['1', '2', '3'],

['4', '5', '6'],

['7', '8', '9'],

['\*', '0', '#']]

ROW\_PINS = [7, 8, 9, 10] # altere de acordo com suas conexões

COL\_PINS = [11, 12, 13] # altere de acordo com suas conexões

GPIO.setmode(GPIO.BOARD)

for j in range(3):

GPIO.setup(COL\_PINS[j], GPIO.OUT)

GPIO.output(COL\_PINS[j], 1)

for i in range(4):

GPIO.setup(ROW\_PINS[i], GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

def keypress():

try:

while True:

for j in range(3):

GPIO.output(COL\_PINS[j], 0)

for i in range(4):

if GPIO.input(ROW\_PINS[i]) == 0:

return MATRIX[i][j]

GPIO.output(COL\_PINS[j], 1)

finally:

GPIO.cleanup()

# Função para inicializar o banco de dados

def init\_db():

conn = sqlite3.connect('keycabinet.db')

c = conn.cursor()

c.execute('''CREATE TABLE IF NOT EXISTS keys

(timestamp TEXT, user\_id TEXT, action TEXT)''')

conn.commit()

conn.close()

# Função para registrar a ação no banco de dados

def log\_key\_action(user\_id, action):

conn = sqlite3.connect('keycabinet.db')

c = conn.cursor()

now = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

c.execute("INSERT INTO keys VALUES (?, ?, ?)", (now, user\_id, action))

conn.commit()

conn.close()

init\_db()

# Loop principal

while True:

lcd.clear()

lcd.message('Insira seu ID:')

user\_id = ''

while True:

key = keypress()

if key == '#':

log\_key\_action(user\_id, 'RETRIEVE' if action == '1' else 'RETURN')

lcd.clear()

lcd.message('Chave ' + ('retirada' if action == '1' else 'reposta'))

time.sleep(2)

break

elif key == '\*':

user\_id = ''

lcd.clear()

lcd.message('ID cancelado\nInsira novamente:')

else:

user\_id += key

lcd.clear()

lcd.message(f'ID: {user\_id}\n# Concluir, \* Cancelar')

```

### README.md

```markdown

# DigitalKeyCabinet

## Introdução

Este projeto desenvolve um sistema de claviculário digital usando um Raspberry Pi. O sistema permite que os usuários registrem a retirada e reposição de chaves através de um teclado numérico e confirma a ação em um display LCD 24x2.

## Materiais Necessários

- Raspberry Pi

- Teclado Numérico

- LCD 24x2 com interface I2C

- Cabos de conexão

- Resistores (se necessário para o teclado)

## Propósito do Projeto

Gerenciar de forma eficiente e segura a retirada e a devolução de chaves, registrando cada ação para futura auditoria.

## Prós e Contras

\*\*Prós:\*\*

- Fácil de usar e configurar.

- Registro digital de ações para segurança e auditoria.

\*\*Contras:\*\*

- Requer hardware específico.

- Dependente de energia elétrica.

## Diretrizes Gerais

- Certifique-se de que todas as conexões estejam corretas e seguras.

- Altere os PINs no código conforme a sua montagem.

## Licença

MIT License

```

### Título do Commit

- `Add key management system`

### Descrição do Commit

- `Implemented a digital key management system using Raspberry Pi, numeric keypad, and LCD display for tracking key withdrawals and returns.`

### Imagem Representativa

Vamos criar uma imagem minimalista e isométrica que representa o projeto de programação Python.

```json

{

"prompt": "Create a minimalist, isometric image of a Raspberry Pi connected to an LCD display and a numeric keypad, symbolizing a digital key management system. The design should incorporate elements associated with Python programming, like the Python logo colors or snake, in a subtle, integrated way. The image should convey the idea of digital security and hardware interaction, using soft, welcoming colors and clear, simple lines."

}

```

Acima está a imagem representativa para o seu projeto `DigitalKeyCabinet`. Esta imagem captura a essência do sistema de gerenciamento de chaves digital, combinando elementos de programação Python e a interação segura com hardware. Se precisar de mais assistência ou ajustes no código e na documentação, estou à disposição para ajudar! 