

## Criando Banco de Memórias

**Mário Leite**

...

Às vezes é preciso reescrever um texto já utilizado em várias ocasiões anteriores; então seria bom ter uma maneira de recuperar esse texto usando algum sistema que permita fazer isto, de maneira simples e eficiente. O programa mostrado abaixo é uma simulação bem simples de como “outorgar memória” a um software que normalmente “esquece” tudo quando é fechado. O programa-exemplo aqui apresentado faz o seguinte

- Na primeira vez: pergunta seu nome e salva no arquivo.
- Na próxima vez que você rodar, ele lembra quem você é sem perguntar de novo.
- Faz o mesmo com preferências (no exemplo, a cor favorita).

Isto significa que ele guarda informações entre execuções; coisa que o ChatGPT (versão grátis) não faz, ainda. Ao executar o programa acontece o seguinte:

1. **Verifica** se o arquivo “ARQ\_MEMO.json” existe.
2. **Se não existe** → cria arquivo vazio → coleta dados → salva.
3. **Se existe** → carrega dados → mostra informações → atualiza → salva.

O programa é **autossuficiente** não precisa que seja criado nenhum arquivo manualmente; sendo assim, ele pode ser aplicado em diversas situações: em programas interativos, em jogos simples, em chatbots, em automação, em prototipagem de sistemas maiores, etc. Neste exemplo são guardados os dados de preferências do usuário para posteriores acessos, criando uma espécie de “memória” em disco que pode ser recuperada mais tarde, e podendo adicionar novas informações.

Foi usado o formato **.json** por ser leve para troca de dados (fácil para humanos lerem e máquinas processarem) e possui as seguintes características:

- **Simplicidade:** Não precisa de banco de dados complexo.
- **Portabilidade:** O arquivo JSON pode ser aberto em qualquer editor de texto.
- **Compatibilidade:** Funciona em qualquer sistema operacional.

Deste modo, o programa é uma maneira de armazenar dados em um repositório, sem precisar se utilizar de bancos de dados proprietários.

A **figura 1** mostra o arquivo “**memoria.json**” na pasta onde o programa foi salvo; a **figura 2** mostra a saída do programa no primeiro acesso do usuário, iniciando com suas preferências; a **figura 3** mostra o conteúdo do arquivo-memória, destacando as preferências iniciais do usuário, a **figura 4** exibe a saída do programa após o segundo acesso, com as preferências memorizadas (as iniciais acrescidas de outras adicionadas) e a **figura 5** mostra o conteúdo do arquivo com as preferências do usuário, atualizadas, após o segundo acesso. Assim, o programa demonstra como uma “memória” em disco pode ser criada num arquivo de formato **.json**, sem a necessidade de usar bancos de dados.

Resumindo: o programa aqui apresentado (“**MemorizaPreferencias**”), codificado em Python, serve para dar continuidade entre execuções de um programa, como se fosse uma “mini memória artificial”; no presente caso “memoriza” certas preferências do usuário.

---

**Nota:** Este programa foi tirado do livro do autor: “**1001 Programas em Python Para Você Aprender Praticando**” - disponível na Amazon.

	Nome	Data de modific...	Tipo	Tamanho
	JogoDaOca.py	02/07/2024 11:13	Arquivo PY	4 KB
	JogoDaVelha.py	18/07/2025 10:58	Arquivo PY	3 KB
	JogoDeDados.py	12/05/2024 12:45	Arquivo PY	2 KB
	JogoDoBichinhoEletronico.py	13/06/2024 13:50	Arquivo PY	6 KB
	JogoDoBicho.py	21/11/2024 13:17	Arquivo PY	2 KB
	JogoDoCacaNiquel.py	12/05/2024 13:02	Arquivo PY	3 KB
	JogoDoNumeroSecreto.py	12/08/2025 15:00	Arquivo PY	2 KB
	JogoDoScrabble.py	19/05/2024 12:46	Arquivo PY	3 KB
	JogoPalavrasCruzadas.py	04/06/2024 12:44	Arquivo PY	1 KB
	memoria.json	29/08/2025 11:34	JSON File	4 KB
	MemorizaPreferencias.py	29/08/2025 11:22	Arquivo PY	2 KB
	MovimentoDeque.py	13/06/2025 11:30	Arquivo PY	2 KB
	MovimentoEmRestaurante.py	26/07/2024 10:50	Arquivo PY	2 KB
	MovimentoFIFO.py	25/11/2024 11:58	Arquivo PY	3 KB
	MovimentoFILO.py	25/11/2024 12:04	Arquivo PY	2 KB
	MovimentoLIFO.py	24/11/2024 12:34	Arquivo PY	2 KB
	MovimentoLILO.py	25/11/2024 11:14	Arquivo PY	3 KB
	NovoCalendario.py	21/07/2025 21:25	Arquivo PY	2 KB
	NumeroPorExtenso.py	22/08/2024 11:29	Arquivo PY	2 KB
	NumerosAleatoriosCorrelacionados.py	09/06/2025 14:48	Arquivo PY	4 KB
	OperacoesComFraçoes.py	07/08/2025 13:11	Arquivo PY	4 KB
	OperacoesDeConjuntos.py	16/07/2025 15:19	Arquivo PY	4 KB

**Figura 1 - O arquivo “memoria.json” na pasta do programa**

```

IDLE Shell 3.13.0
File Edit Shell Debug Options Window Help
Python 3.13.0 (tags/v3.13.0:60403a5, Oct 7 2024, 09:38:07) [MSC v.1941 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/Livros/Livro11/Códigos/Nível 3/Códigos-fonte/MemorizaPreferencias.py

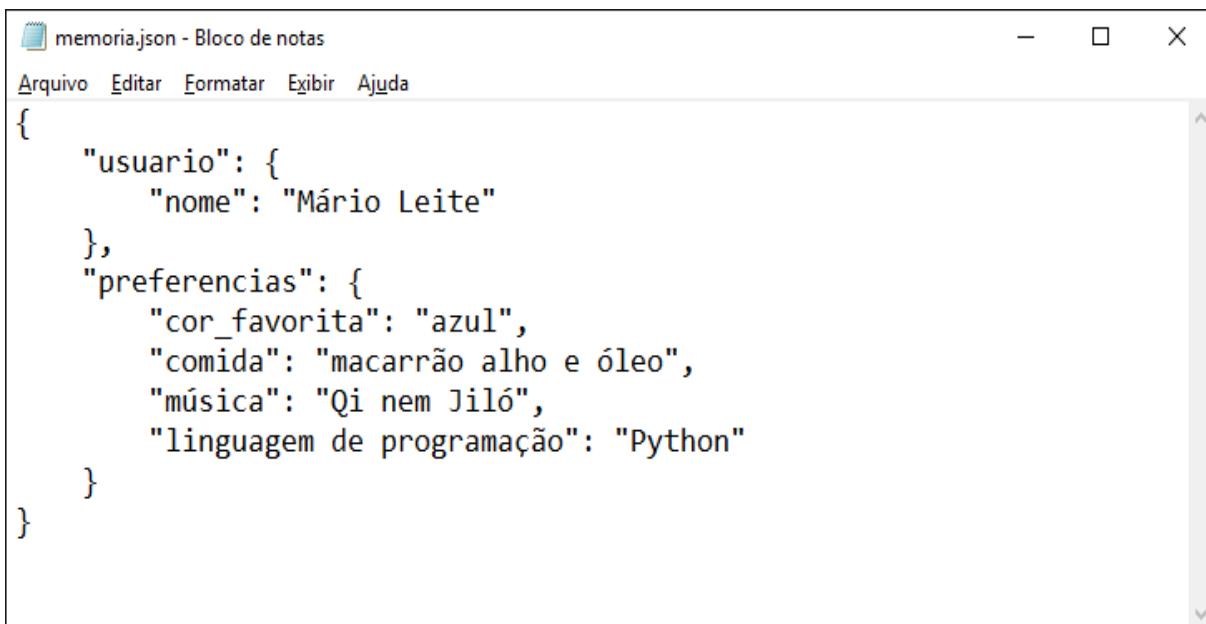
== Simulação de Memória ==
Qual é o seu nome? Mário Leite
Prazer em conhecer, Mário Leite!

Qual é a sua cor favorita? azul
Deseja adicionar outra preferência? (s/n): s
Digite o nome da preferência (ex: comida, música): comida
Digite seu (sua) comida favorita: macarrão alho e óleo
Deseja adicionar outra preferência? (s/n): s
Digite o nome da preferência (ex: comida, música): música
Digite seu (sua) música favorita: Qi nem Jiló
Deseja adicionar outra preferência? (s/n): s
Digite o nome da preferência (ex: comida, música): linguagem de programação
Digite seu (sua) linguagem de programação favorita: Python
Deseja adicionar outra preferência? (s/n): n

Ok, Mário Leite, vou lembrar de suas preferências!
Até a próxima!
>>> | Ln: 24 Col: 0

```

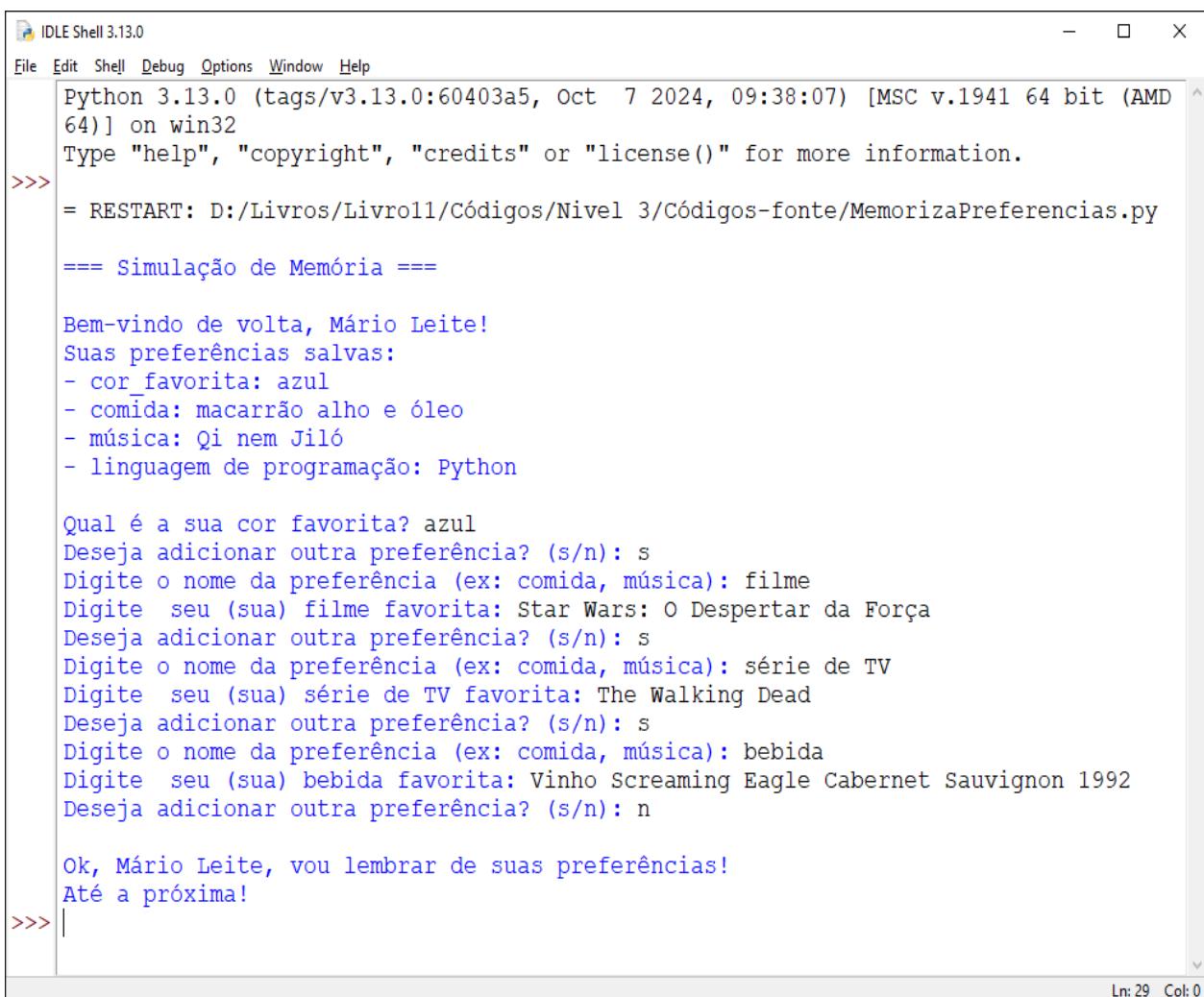
**Figura 2 - Saída do programa “AcessaArquivoMemoria”: primeiro acesso**



A screenshot of a Windows Notepad window titled "memoria.json - Bloco de notas". The window contains the following JSON code:

```
{  
    "usuario": {  
        "nome": "Mário Leite"  
    },  
    "preferencias": {  
        "cor_favorita": "azul",  
        "comida": "macarrão alho e óleo",  
        "música": "Qi nem Jiló",  
        "linguagem de programação": "Python"  
    }  
}
```

**Figura 3 - Conteúdo do arquivo após o primeiro acesso**



A screenshot of an IDLE Shell 3.13.0 window. The window title is "IDLE Shell 3.13.0". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell area displays the following text:

```
Python 3.13.0 (tags/v3.13.0:60403a5, Oct  7 2024, 09:38:07) [MSC v.1941 64 bit (AMD  
64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> = RESTART: D:/Livros/Livro11/Códigos/Nível 3/Códigos-fonte/MemorizaPreferencias.py  
  
==== Simulação de Memória ====  
  
Bem-vindo de volta, Mário Leite!  
Suas preferências salvas:  
- cor_favorita: azul  
- comida: macarrão alho e óleo  
- música: Qi nem Jiló  
- linguagem de programação: Python  
  
Qual é a sua cor favorita? azul  
Deseja adicionar outra preferência? (s/n): s  
Digite o nome da preferência (ex: comida, música): filme  
Digite seu (sua) filme favorita: Star Wars: O Despertar da Força  
Deseja adicionar outra preferência? (s/n): s  
Digite o nome da preferência (ex: comida, música): série de TV  
Digite seu (sua) série de TV favorita: The Walking Dead  
Deseja adicionar outra preferência? (s/n): s  
Digite o nome da preferência (ex: comida, música): bebida  
Digite seu (sua) bebida favorita: Vinho Screaming Eagle Cabernet Sauvignon 1992  
Deseja adicionar outra preferência? (s/n): n  
  
Ok, Mário Leite, vou lembrar de suas preferências!  
Até a próxima!  
>>> |
```

**Figura 4 - Saída do programa “AcessaArquivoMemoria”: segundo acesso**

```
{  
    "usuario": {  
        "nome": "Mário Leite"  
    },  
    "preferencias": {  
        "cor_favorita": "azul",  
        "comida": "macarrão alho e óleo",  
        "música": "Qi nem Jiló",  
        "linguagem de programação": "Python",  
        "filme": "Star Wars: O Despertar da Força",  
        "série de TV": "The Walking Dead",  
        "bebida": "Vinho Screaming Eagle Cabernet Sauvignon 1992"  
    }  
}
```

**Figura 5 - Conteúdo do atualizado com as preferências adicionadas**

```
'''  
MemorizaPreferencias.py  
-----  
Este programa demonstra um conceito fundamental de persistência de dados em  
em Python, usando JSON para armazenar informações de forma simples e eficiente.  
É ideal para projetos iniciais em Python ou para implementar  
-----  
memória em sistemas interativos.  
'''  
  
import os  
import json  
  
ARQ_MEMO = "memoria.json" #define o arquivo para armazenar o texto  
  
#Função para carregar a memória  
def CarregarMemoria():  
    try:  
        if(os.path.exists(ARQ_MEMO)):  
            with(open(ARQ_MEMO, "r", encoding="utf-8") as f):  
                return json.load(f)  
    except Exception as e:  
        print(f"Erro ao carregar memória: {e}")  
    return {}  
#-----  
#Função para salvar a memória  
def SalvarMemoria(memoria):  
    try:  
        with(open(ARQ_MEMO, "w", encoding="utf-8") as f):  
            json.dump(memoria, f, ensure_ascii=False, indent=4)  
    except Exception as e:  
        print(f"Erro ao salvar memória: {e}")  
#-----
```

```

#Função principal
def main():
    memoria = CarregarMemoria()
    print()
    print("== Simulação de Memória ==")

    #Verifica se já tem as informações do usuário
    if("usuario" in memoria):
        nome = memoria["usuario"].get("nome")
        print(f"Bem-vindo de volta, {nome}!")

    #Mostra informações salvas anteriores
    if("preferencias" in memoria):
        print("Suas preferências salvas:")
        for key, valor in memoria["preferencias"].items():
            print(f"- {key}: {valor}")

    else:
        #Primeira execução - coleta informações básicas
        memoria["usuario"] = {}
        nome = input("Qual é o seu nome? ")
        memoria["usuario"]["nome"] = nome
        SalvarMemoria(memoria)
        print(f"Prazer em conhecer, {nome}!")

    #Coleta ou atualiza preferências
    if("preferencias" not in memoria):
        memoria["preferencias"] = {}

    cor = input("Qual é a sua cor favorita? ")
    memoria["preferencias"]["cor_favorita"] = cor

    #Opção para adicionar mais preferências
    while (True):
        nova_pref = input("Deseja adicionar outra preferência? (s/n): ").lower()
        if(nova_pref != 's'):
            break

        chave = input("Digite o nome da preferência (ex: comida, música): ")
        valor = input(f"Digite seu (sua) {chave} favorita: ")
        memoria["preferencias"][chave] = valor

    SalvarMemoria(memoria) #salva o(s) texto(s) no arquivo

    print(f"Ok, {memoria['usuario']['nome']}, vou lembrar de suas preferências!")
    print("Até a próxima!")

#=====
#Programa principal
if(__name__ == "__main__"):
    main()
#Fim do programa "MemorizaPreferencias" -----

```