



# TÍTULO: SISTEMA DE AGENDAMENTO PARA CLÍNICAS DE PSICOTERAPIA

Autores: Daniel da Silva Alves e  
Renato Alves Júnior

Disciplina: INF0063 – AED1

# Problema e Justificativa

**Dificuldades em gerenciar agendamentos manualmente.**

**Justificativa: Melhora a organização, reduz erros de agendamento e aumenta a produtividade.**

# Objetivo do Sistema

**Criar um sistema de agendamento especializado em clínicas de psicoterapia. Funcionalidades: Cadastro de pacientes, verificação de conflitos de horário e persistência de dados.**

# Manipulação de Datas e Horários

Uso da biblioteca datetime para gerenciamento confiável. Prevenção de conflitos de horários com verificações automáticas.

# Metodologia

**Planejamento:** Definição de requisitos e ferramentas (Python, JSON, datetime). **Desenvolvimento:** Cadastro de pacientes (armazenados em JSON). **Agendamento e verificação de conflitos.** Persistência de dados após cada operação. **Testes:** Simulações com múltiplos pacientes e cenários reais.

# Conceito de TAD

**Definição:** Coleção de dados e operações que podem ser realizadas sobre eles.

**No sistema:**

**Lista:** Para armazenar agendamentos cronológicos.

**Dicionário:** Para guardar dados dos pacientes com acesso por chave.

# Trecho de Código - Funções

## Função Listar Pacientes

```
45 # Função para listar os pacientes com todos os detalhes
46 def listar_pacientes(dados):
47     print("\nLista de Pacientes:")
48     if not dados["pacientes"]:
49         print("Nenhum paciente cadastrado.")
50     else:
51         for i, paciente in enumerate(dados["pacientes"], start=1):
52             print(f"{i}. Nome: {paciente['nome']}")
53             print(f"    Nome Social: {paciente['nome_social']}")
54             print(f"    Telefone: {paciente['telefone']}")
55             print(f"    Celular: {paciente['celular']}")
56             print(f"    Contato de Emergência: {paciente['contato_emergencia']}")
57             print(f"    Telefone de Emergência: {paciente['telefone_emergencia']}")
58             print(f"    Data da Próxima Consulta: {paciente['data_proxima_consulta']}")
59             print(f"    Valor da Consulta: {paciente['valor_consulta']}")
60             print(f"    Histórico de Consultas: {', '.join(paciente['historico_consultas']) if paciente['historico_consultas'] else 'Nenhum registro.'}")
61             print("-" * 40)
```

**Explicação:** O dicionário armazena pacientes e suas informações como chave-valor. A lista pacientes organiza todos os registros.

# Função - Cadastro de Pacientes

```
# Função para listar os pacientes com todos os detalhes
def listar_pacientes(dados):
    print("\nLista de Pacientes:")
    if not dados["pacientes"]:
        print("Nenhum paciente cadastrado.")
    else:
        for i, paciente in enumerate(dados["pacientes"], start=1):
            print(f"{i}. Nome: {paciente['nome']}")
            print(f"    Nome Social: {paciente['nome_social']}")
            print(f"    Telefone: {paciente['telefone']}")
            print(f"    Celular: {paciente['celular']}")
            print(f"    Contato de Emergência: {paciente['contato_emergencia']}")
            print(f"    Telefone de Emergência: {paciente['telefone_emergencia']}")
            print(f"    Data da Próxima Consulta: {paciente['data_proxima_consulta']}")
            print(f"    Valor da Consulta: {paciente['valor_consulta']}")
            print(f"    Histórico de Consultas: {' , '.join(paciente['historico_consultas']) if paciente['historico_consultas'] else 'Nenhum registro.'}")
            print("-" * 40)
```

**Explicação:** Permite que o usuário cadastre um novo paciente e após a coleta dos dados utiliza a função 'salvar\_dados()'



# Funções - Carregar e salvar dados no arquivo JSON

```
# Função para carregar dados de um arquivo JSON
def carregar_dados():
    try:
        with open("dados.json", "r") as arquivo:
            return json.load(arquivo)
    except FileNotFoundError:
        return {"pacientes": []}

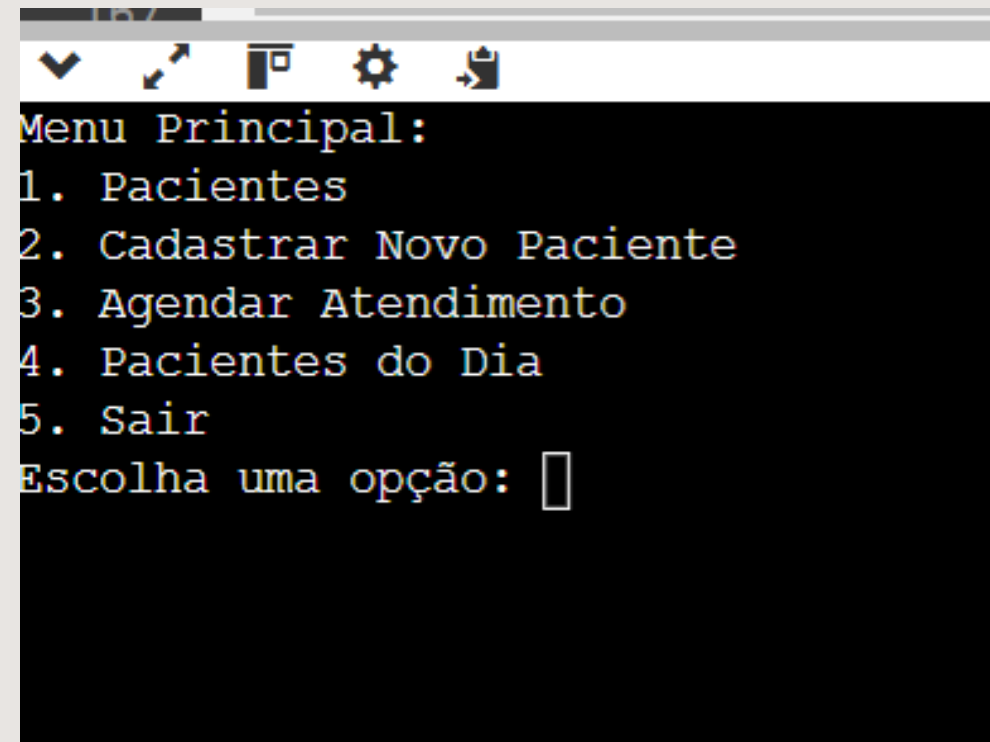
# Função para salvar dados em um arquivo JSON
def salvar_dados(dados):
    with open("dados.json", "w") as arquivo:
        json.dump(dados, arquivo, indent=4)
```

Explicação: Carregar os dados dos pacientes a partir de um arquivo JSON e Salvar o arquivos em um arquivo JSON.

Se o arquivo não for encontrado, retorna um dicionário com uma chave "pacientes" contendo uma lista vazia.

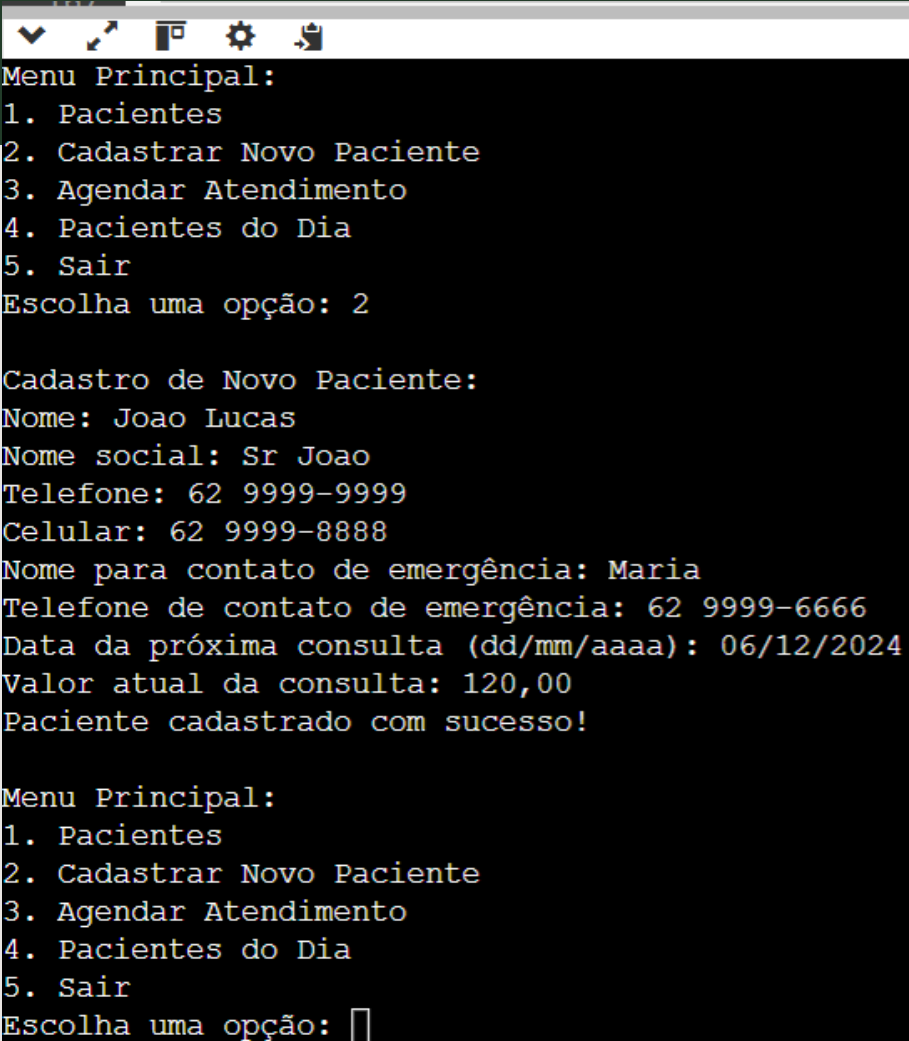
# Prints do Sistema

-> Menu Inicial do Sistema



# Prints do Sistema

-> Cadastrando Paciente



```
Menu Principal:
1. Pacientes
2. Cadastrar Novo Paciente
3. Agendar Atendimento
4. Pacientes do Dia
5. Sair
Escolha uma opção: 2

Cadastro de Novo Paciente:
Nome: Joao Lucas
Nome social: Sr Joao
Telefone: 62 9999-9999
Celular: 62 9999-8888
Nome para contato de emergência: Maria
Telefone de contato de emergência: 62 9999-6666
Data da próxima consulta (dd/mm/aaaa): 06/12/2024
Valor atual da consulta: 120,00
Paciente cadastrado com sucesso!

Menu Principal:
1. Pacientes
2. Cadastrar Novo Paciente
3. Agendar Atendimento
4. Pacientes do Dia
5. Sair
Escolha uma opção: □
```

# Prints do Sistema

-> Listando/Consultando  
Pacientes cadastrados

```
Menu Principal:
1. Pacientes
2. Cadastrar Novo Paciente
3. Agendar Atendimento
4. Pacientes do Dia
5. Sair
Escolha uma opção: 1

Menu de Pacientes:
1. Listar Pacientes
2. Editar Paciente
3. Voltar
Escolha uma opção: 1

Lista de Pacientes:
1. Nome: Joao Lucas
   Nome Social: Sr Joao
   Telefone: 62 9999-9999
   Celular: 62 9999-8888
   Contato de Emergência: Maria
   Telefone de Emergência: 62 9999-6666
   Data da Próxima Consulta: 06/12/2024
   Valor da Consulta: 120,00
   Histórico de Consultas: Nenhum registro.
-----

Menu de Pacientes:
1. Listar Pacientes
2. Editar Paciente
3. Voltar
Escolha uma opção: 
```

# Prints do Sistema

-> Agendando Consulta

Menu Principal:

1. Pacientes
2. Cadastrar Novo Paciente
3. Agendar Atendimento
4. Pacientes do Dia
5. Sair

Escolha uma opção: 3

Lista de Pacientes:

1. Joao Lucas

Selecione o número do paciente para agendar consulta: 1

Digite a data da consulta (dd/mm/aaaa): 06/12/2024

Digite o horário da consulta (hh:mm): 10:00

Consulta agendada com sucesso!

Menu Principal:

1. Pacientes
2. Cadastrar Novo Paciente
3. Agendar Atendimento
4. Pacientes do Dia
5. Sair

Escolha uma opção:

# Prints do Sistema

-> Como os dados são salvos no JSON.

```
main.py  dados.json
1 {
2   "pacientes": [
3     {
4       "nome": "Joao Lucas",
5       "nome_social": "Sr Joao",
6       "telefone": "62 9999-9999",
7       "celular": "62 9999-8888",
8       "contato_emergencia": "Maria ",
9       "telefone_emergencia": "62 9999-6666",
10      "data_proxima_consulta": "06/12/2024",
11      "historico_consultas": [
12        "06/12/2024 \u00e0s 10:00"
13      ],
14      "valor_consulta": "120,00"
15    }
16  ]
17 }
```

# Resultados

**Cadastro de pacientes eficiente e persistente. Prevenção de conflitos de horário com verificações automáticas. Interface simples para usuários com pouca experiência em tecnologia.**

# Conclusão

**Impacto:** Organização e eficiência aumentadas para clínicas de pequeno/médio porte.

**Futuro:** Notificações automáticas por e-mail. Relatórios detalhados de consultas, inclusão da função valida CPF e cadastro usando CPF.



# Referências

**Barlow et al., 2020 - Psychotherapy and the new technologies**

**Ferreira, A. L., 2018 - Sistemas de gestão para clínicas de psicoterapia**

**Souza, R. P.; Lima, M. T., 2019 - Soluções tecnológicas para o setor de saúde mental.**