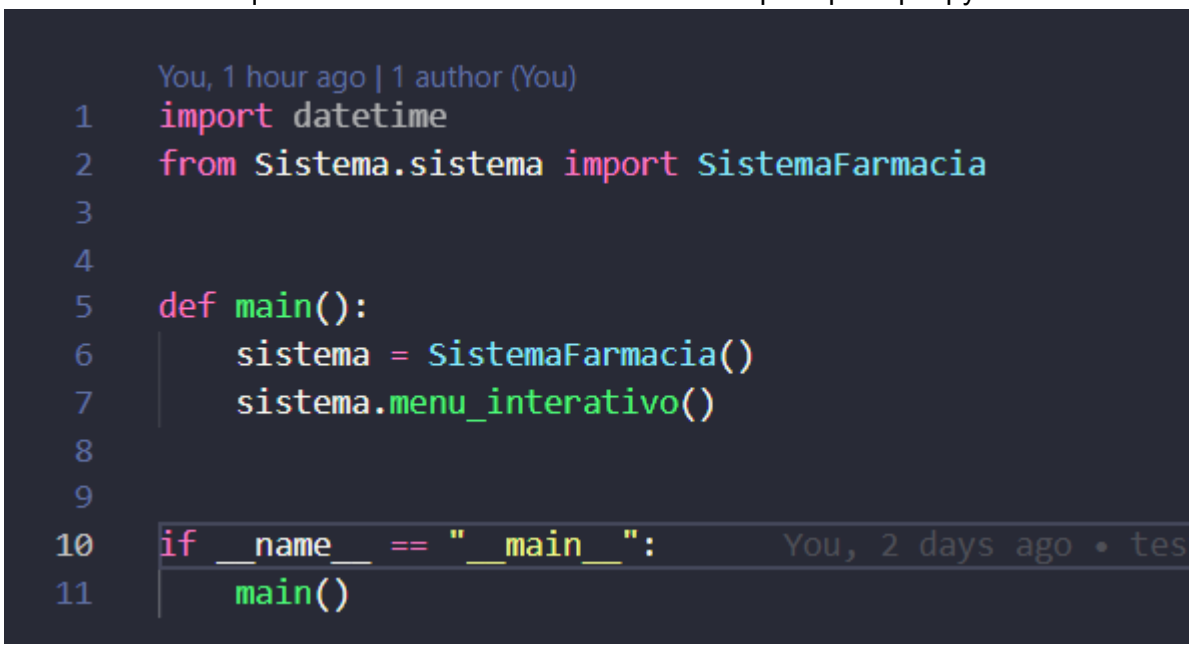
Documentação:

Sistema de Gerenciamento Interno de Hospital;

- Guia de Uso:

Todos os arquivos do sistema são chamados no arquivo principal.py:



```
You, 1 hour ago | 1 author (You)
import datetime
from Sistema.sistema import SistemaFarmacia


def main():
    sistema = SistemaFarmacia()
    sistema.menu_interativo()


if __name__ == "__main__":      You, 2 days ago • tes
    main()
```

OBS.: import datetime já é importada no arquivo sistema, esse, que chama o arquivo usuários que tem todas as class relacionadas aos usuários e suas funções:

```
You, 51 minutes ago | 1 author (You)
import datetime
import abc

# Classe base para os usuários do sistema
You, 23 hours ago | 1 author (You)
class Usuario(abc.ABC): ···

# Classe para Pacientes
You, 23 hours ago | 1 author (You)
class Paciente(Usuario): ···

# Classe para Médicos
You, 51 minutes ago | 1 author (You)
class Medico(Usuario): ···

You, 23 hours ago | 1 author (You)
class Guiche(Usuario): ···

# Classe para Enfermeiros
You, 23 hours ago | 1 author (You)
class Enfermeiro(Usuario): ···

You, 51 minutes ago | 1 author (You)
class AtendenteFarmacia(Usuario): ···
You, 51 minutes ago | 1 author (You)
class Administrador(Usuario):      You, 23 hours ago • Trabalho final
```

Já a sistema tem todos os menus além de alguns usuários e medicamentos já pré-cadastrados:

```python
from Usuarios.usuarios import Medico, Guiche, Enfermeiro, AtendenteFarmacia, Administrador
import os
import platform

You, 33 minutes ago | 1 author (You)
class SistemaFarmacia:
    """Classe principal para gerenciar o sistema da farmácia."""

    def __init__(self):
        self.solicitacoes = []
        self.prescricoes = []
        self.guiche = Guiche("Guiche Principal", "00000000000")
        self.consultas = []
        self.usuario_logado = None

        administrador = Administrador("Admin", "00000000001")
        self.movimentacoes = administrador.movimentacoes
        self.usuarios = administrador.usuarios
        self.usuarios.append(Enfermeiro("Mariana", "98765432100", "COREN5678"))
        atendente = AtendenteFarmacia("Ana", "11223344556")
        self.usuarios.append(atendente)
        self.usuarios.append(Medico("Dr. João", "12312312399", "CRM12345"))
        self.usuarios.append(Guiche("Carlos", "99887766554"))
        self.usuarios.append(administrador)

        # Adicionando um paciente inicial
        self.guiche.registrar_paciente("José da Silva", "11122233344", 45)
        medicamentos_iniciais = {
            "Dipirona": 100,
            "Paracetamol": 80,
            "Amoxicilina": 50
        }
        for medicamento, quantidade in medicamentos_iniciais.items():
            atendente.adicionar_medicamento(medicamento, quantidade)
```

Após o termino de cada ação de determinado usuário como Guiche:
Cadastrou um paciente -> Mandou para o medico -> logout
depois login no medico e continua o ciclo de ação:
Guiche -> Médico -> Enfermeiro -> Atendente -> (ADM) tem acesso a funções de cadastro dos demais e relatório geral do sistema

OBS.: as credenciais para o login é somente o CPF do usuário assim vendo o CPF (isinstance) está instanciado a uma das classes:

```python
else:
    if isinstance(self.usuario_logado, Medico):
        self.menu_medico()
    elif isinstance(self.usuario_logado, Enfermeiro):
        self.menu_enfermeiro()
    elif isinstance(self.usuario_logado, AtendenteFarmacia):
        self.menu_atendente()
    elif isinstance(self.usuario_logado, Guiche):
        self.menu_guiche()
    elif isinstance(self.usuario_logado, Administrador):
```

chamando o menu específico da classe e suas funcionalidades;

Documentação 1.1:

Introdução

Este sistema foi projetado para gerenciar operações de uma farmácia, incluindo cadastro e gestão de usuários, controle de medicamentos e atendimento de pacientes. Ele está organizado em diferentes módulos que se complementam para fornecer funcionalidades completas e seguras.

A estrutura do projeto está dividida em três principais arquivos:

- **principal.py**: Arquivo principal que executa o sistema.
- **Sistema/sistema.py**: Contém a lógica principal do sistema, como menus e funcionalidades globais.
- **Usuarios/usuarios.py**: Define as classes relacionadas aos usuários do sistema.

Arquivo: `principal.py`

Este é o ponto de entrada do sistema. Ele inicializa a classe principal do sistema e chama o menu interativo.

Função Principal:

1. **Importação de Módulos:** Importa a classe `SistemaFarmacia` do módulo `Sistema.sistema`.
2. **Inicialização:** Cria uma instância de `SistemaFarmacia`.
3. **Execução do Menu Interativo:** Inicia o menu principal que gerencia a interação com o usuário.

Arquivo: `Sistema/sistema.py`

Este módulo contém a classe `SistemaFarmacia`, que é o núcleo do sistema. Ele gerencia todas as funcionalidades globais, como controle de usuários, medicamentos, pacientes e relatórios.

Classe: `SistemaFarmacia`

Abaixo estão os principais atributos e métodos desta classe:

Atributos:

- **`solicitacoes`**: Lista de solicitações de medicamentos.
- **`prescricoes`**: Lista de prescrições médicas.
- **`guiche`**: Instância da classe `Guiche`.
- **`consultas`**: Lista de consultas realizadas.
- **`usuario_logado`**: Usuário atualmente logado no sistema.
- **`movimentacoes`**: Lista de movimentações registradas no sistema.
- **`usuarios`**: Lista de todos os usuários cadastrados.

Métodos:

1. **`__init__`**:
    - Inicializa as listas e objetos necessários.
    - Adiciona usuários iniciais (Administrador, Atendente, Médico, Enfermeiro, etc.).
    - Adiciona medicamentos iniciais ao estoque.
2. **`menu_interativo`**:
    - Gerencia a navegação pelo sistema.
    - Redireciona os usuários para menus específicos com base em seu tipo (Médico, Enfermeiro, etc.).
3. **`login`**:
    - Permite que um usuário faça login fornecendo o CPF.
4. **Menus Específicos**:
    - Cada tipo de usuário tem um menu personalizado com opções relevantes às suas responsabilidades:
        - **Médico**: Listar pacientes, realizar consultas, etc.
        - **Enfermeiro**: Solicitar medicamentos.
        - **Atendente**: Gerenciar estoque e atender solicitações.
        - **Guichê**: Registrar e gerenciar pacientes.
        - **Administrador**: Cadastrar novos usuários e visualizar relatórios.
5. **`gerenciar_usuarios`**:
    - Permite ao Administrador cadastrar novos usuários no sistema.

Arquivo: `Usuarios/usuarios.py`

Este módulo contém as definições das classes que representam os diferentes tipos de usuários do sistema. Cada classe possui métodos específicos para suas respectivas responsabilidades.

Classes:

1. `Usuario` (Classe Base)

- Atributos:
  - `nome`: Nome do usuário.
  - `cpf`: CPF do usuário.
- Métodos:
  - `__init__`: Inicializa os atributos da classe base.

2. `Medico`

- Herda de `Usuario`.
- Atributos adicionais:
  - `crm`: Número do registro no Conselho Regional de Medicina.
- Métodos:
  - `realizar_consulta`: Registra a prescrição de medicamentos para um paciente.

3. `Enfermeiro`

- Herda de `Usuario`.
- Atributos adicionais:
  - `coren`: Número do registro no Conselho Regional de Enfermagem.
- Métodos:
  - `solicitar_medicamento`: Solicita medicamentos para pacientes.

4. `AtendenteFarmacia`

- Herda de `Usuario`.
- Métodos:
  - `verificar_estoque`: Confere a quantidade disponível de um medicamento no estoque.
  - `adicionar_medicamento`: Adiciona medicamentos ao estoque.
  - `atender_solicitacao`: Processa solicitações de medicamentos.

5. `Guiche`

- Herda de `Usuario`.
- Métodos:
  - `registrar_paciente`: Adiciona novos pacientes à lista de espera.
  - `listar_pacientes`: Mostra os pacientes aguardando atendimento.

- `enviar_paciente_para_medico`: Direciona um paciente a um médico disponível.

6. `Administrador`

- Herda de `Usuario`.
- Atributos adicionais:
  - `usuarios`: Lista de todos os usuários do sistema.
  - `movimentacoes`: Lista de movimentações no sistema.
- Métodos:
  - `cadastrar_medico`: Registra um novo médico.
  - `cadastrar_enfermeiro`: Registra um novo enfermeiro.
  - `cadastrar_atendente_farmacia`: Registra um novo atendente.
  - `cadastrar_guiche`: Registra um novo guichê.
  - `listar_usuarios`: Mostra todos os usuários cadastrados.

Diagrama de Class: