

Sistema de Prontuário Eletrônico e Apoio à Decisão Clínica em Python



Universidade Federal de Minas Gerais Escola de Engenharia Departamento de Engenharia Elétrica

ELE078 Programação Orientada a Objetos, Turma TECAD Trabalho Prático 2025/1 Matheus Marcondes de Oliveira (2022112517)

Documentação

- Relatório descritivo
- Registro de desenvolvimento de software
- Código fonte
- Apresentação



Disponível em github.com/matheusmarcondes1/mediclass

Sistema

main.py: carrega dados salvos, instancia o objeto principal (SistemaMediclass) e trata a inicialização de usuários padrão.

sistema.py: reside a classe que gerencia a funcionalidade de login, exibe o menu principal e despacha chamadas para operações de registro e interação com o usuário.

profissionais.py: define a hierarquia de usuários, incluindo métodos para autenticação e roteiro de execução de tarefas clínicas pré-definidas para cada subclasse.

paciente.py está a classe Paciente, responsável por armazenar dados cadastrais, histórico de atendimentos e gerenciamento de arquivos de entrada do histórico

anamnese.py e **diagnostico.py:** modelam, respectivamente, a coleta de sinais vitais e sintomas, e a lógica de sugestão de possíveis diagnósticos

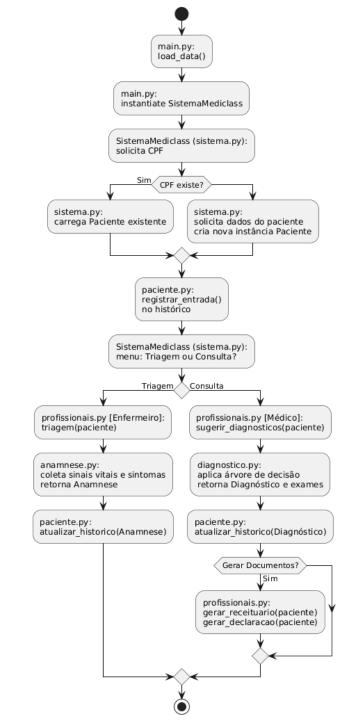
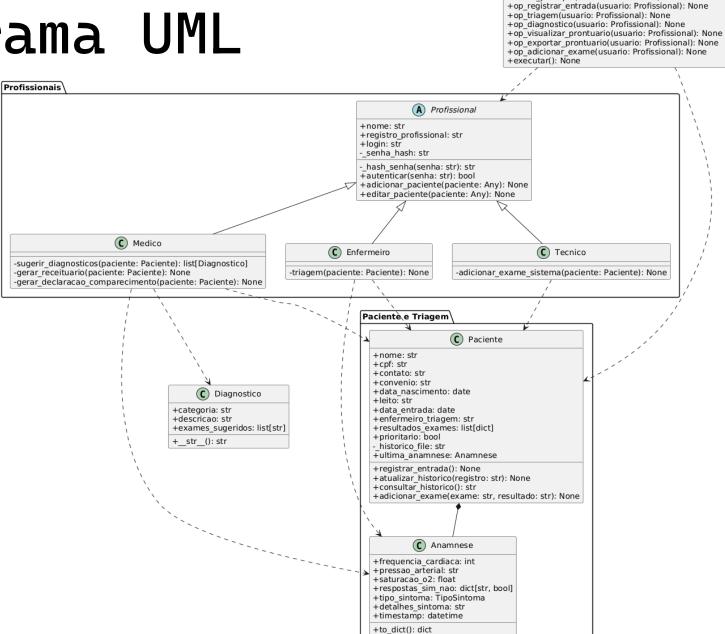


Diagrama UML



C SistemaMediclass

+registrar_usuario(usuario: Profissional): None

+menu_principal(usuario: Profissional): None

+usuarios: dict[str, Profissional] +pacientes: dict[str, Paciente]

+login(): Profissional

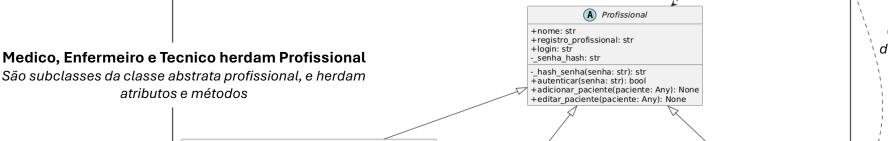


Profissionais

+ usuarios: dict[str, Profissional]
+ pacientes: dict[str, Paciente]
+ registrar_usuario(usuario: Profissional): None
+ login(): Profissional
+ menu_principal(usuario: Profissional): None
+ op_registrar_entrada(usuario: Profissional): None
+ op_triagem(usuario: Profissional): None
+ op_triagem(usuario: Profissional): None
+ op_visualizar_prontuario(usuario: Profissional): None
+ op_exportar_prontuario(usuario: Profissional): None
+ op_adicionar_exame(usuario: Profissional): None
+ executar(): None

(C) Tecnico

-adicionar exame sistema(paciente: Paciente): None



C Enfermeiro

-triagem(paciente: Paciente): None

Sistema depende de Profissionais e Paciente

Classes depende de credenciais do login e dados do paciente para registro e geração de prontuário

Medico depende de Diagnostico

Classes dependem da formatação gerada a partir dos atributos e métodos de Diagnostico

Paciente e Triagem

+nome: str +cpf: str +contato: str

+convenio: str

+data_nascimento: date +leito: str

+data_entrada: date

+enfermeiro_triagem: str +resultados_exames: list[dict]

+prioritario: bool - historico file: str

+ultima_anamnese: Anamnese +registrar entrada(): None

(C) Anamnese

+respostas sim nao: dict[str, bool]

+atualizar_historico(registro: str): None

+consultar_historico(): str

+frequencia_cardiaca: int +pressao arterial: str

+tipo_sintoma: TipoSintoma +detalhes sintoma: str

+saturação o2: float

+timestamp: datetime +to dict(): dict

+adicionar_exame(exame: str, resultado: str): None

(C) Paciente

Medico e Enfermeiro dependem de Anamnese

(C) Diagnostico

+exames_sugeridos: list[str]

+categoria: str

+descricao: str

+ str (): str

(C) Medico

-gerar_declaracao_comparecimento(paciente: Paciente): None

-sugerir diagnosticos(paciente: Paciente): list[Diagnostico]

-gerar receituario(paciente: Paciente): None

Classes dependem do atributo tipo_sintoma informado pelo paciente

Dados iniciais são coletados em Paciente

Classes de Profissional e SistemaMediclass dependem dos dados coletados no método registrar_entrada e armazenadas no histórico_file

Métodos de Consulta, Triagem e Exame são privados

Apenas implementações de determinadas classes podem acessar tais métodos

Paciente

Prontuário exportado do sistema [.txt]

Prontuário de matheus marcondes

CPF: 12345678900

Contato: 31988770000

Convênio: sus

Data de nascimento: 2000-10-10

Leito: 1A

Enfermeiro: Enf. Teste

Prioritário: Sim

Histórico Médico:

Histórico de matheus marcondes (CPF: 12345678900)

Criado em: 2025-07-03 15:36:07

[2025-07-03 15:36:07] Entrada no leito 1A em 2025-07-03

[2025-07-03 15:37:44] Triagem: {'frequencia_cardiaca': 70, 'pressao_arterial': '140/100', 'saturacao_o2': 99,

'respostas_sim_nao': {'tosse': True, 'dispneia': True, 'expectoração': True, 'hemoptise': True, 'dor torácica': True,

'febre': True}, 'tipo_sintoma': 'respiratório', 'detalhes_sintoma': None, 'timestamp': '2025-07-03 15:37:44'}

[2025-07-03 15:37:44] FLAG: Prioridade ativada devido a valores incomuns.

C

Paciente

+nome: str

+cpf: str +contato: str

+convenio: str

+data nascimento: date

+leito: str

+data entrada: date

+enfermeiro_triagem: str

+resultados exames: list[dict]

+prioritario: bool - historico file: str

+ultima anamnese: Anamnese

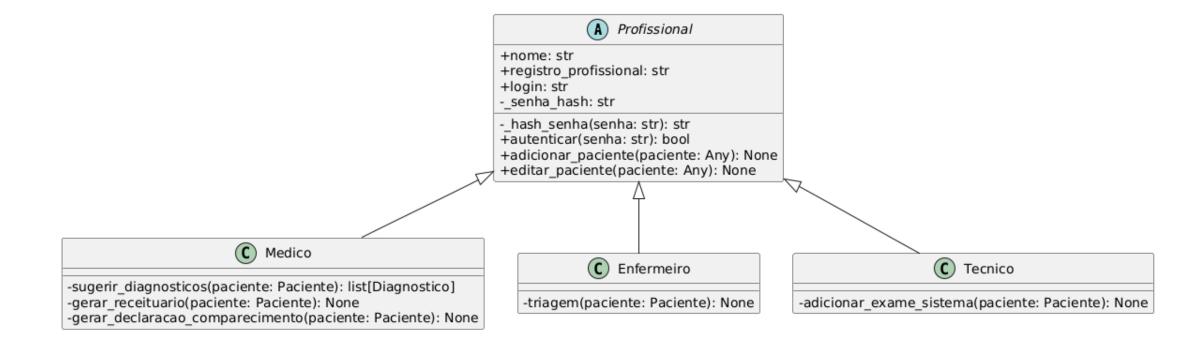
+registrar_entrada(): None

+atualizar historico(registro: str): None

+consultar historico(): str

+adicionar exame(exame: str, resultado: str): None

Profissional



Diagnóstico

Categoria: Respiratório

1. Tosse aguda

2. Dispneia

3. Dor torácica pleurítica

Selecione o subgrupo (1-3): 2

Início súbito com chiado e histórico asmático? (S/N): s

--- Diagnósticos sugeridos ---

Respiratório: Possível exacerbação de asma (Exames sugeridos: Espirometria ou PEFR)



Diagnostico

+categoria: str

+descricao: str

+exames_sugeridos: list[str]

+_str_(): str

Anamnese

Última Anamnese:

frequencia_cardiaca: 70

pressao_arterial: 140/100 # concatenação das variáveis de pressão

saturacao_o2: 99

respostas_sim_nao: {'tosse': True, 'dispneia': True, 'expectoração': True, 'hemoptise': True, 'dor

torácica': True, 'febre': True}

tipo_sintoma: respiratório

detalhes_sintoma: None

timestamp: 2025-07-03 15:37:44



Anamnese

+frequencia cardiaca: int

+pressao_arterial: str

+saturacao_o2: float

+respostas_sim_nao: dict[str, bool]

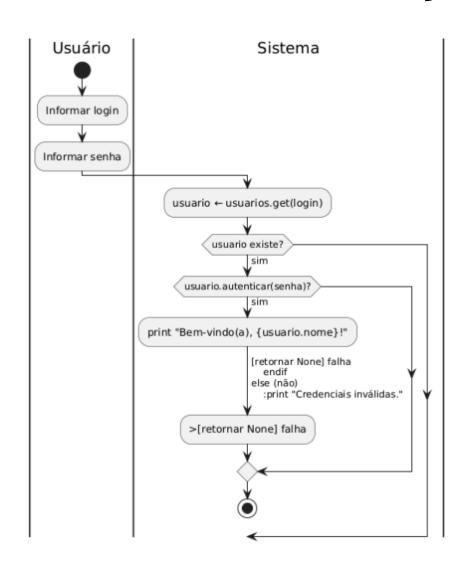
+tipo_sintoma: TipoSintoma

+detalhes_sintoma: str

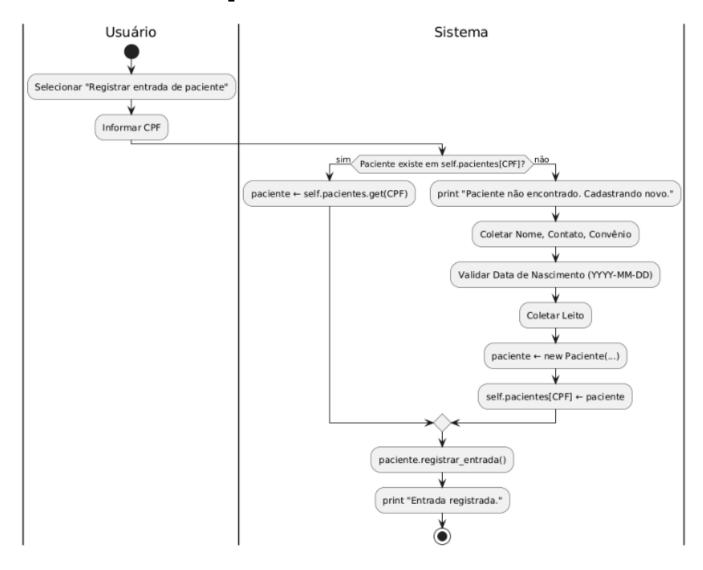
+timestamp: datetime

+to dict(): dict

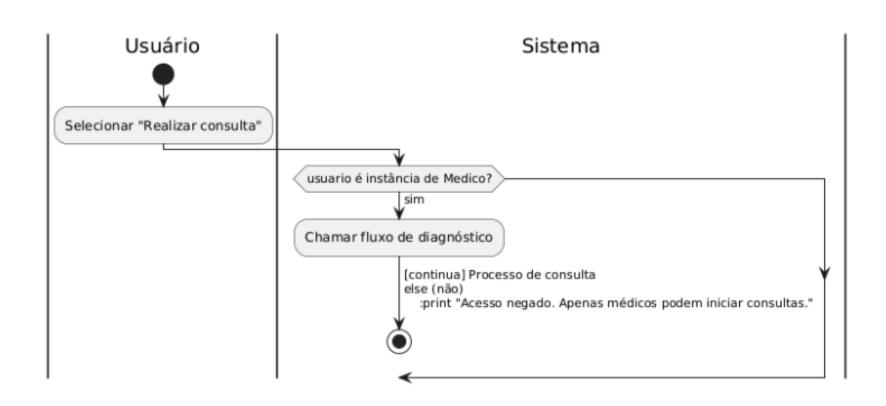
Processo de autenticação



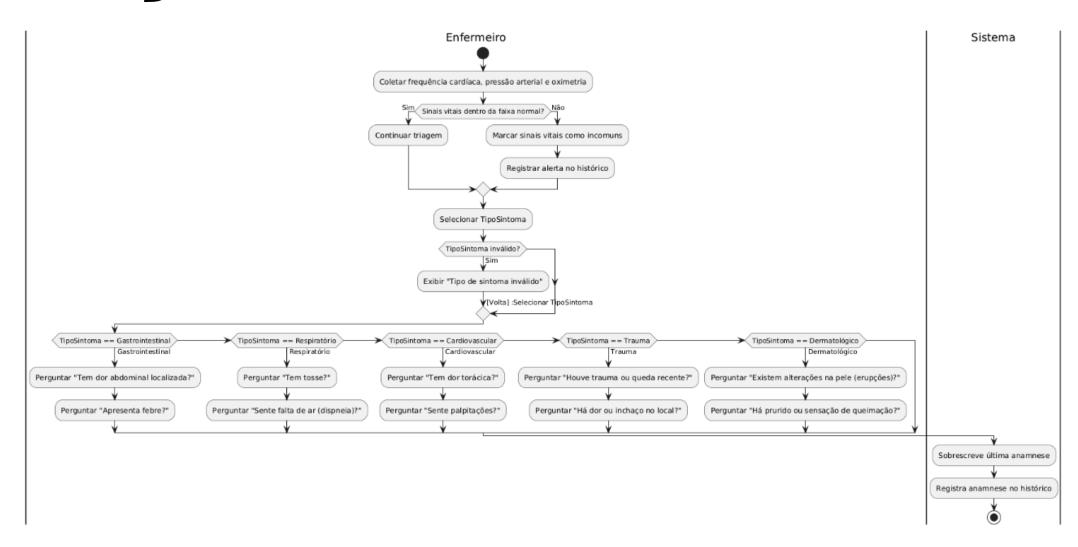
Registro de paciente



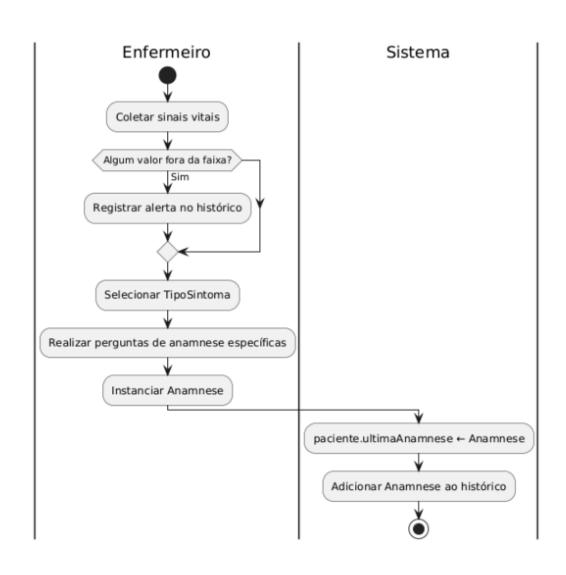
Verificação de classe para execução de métodos protegidos



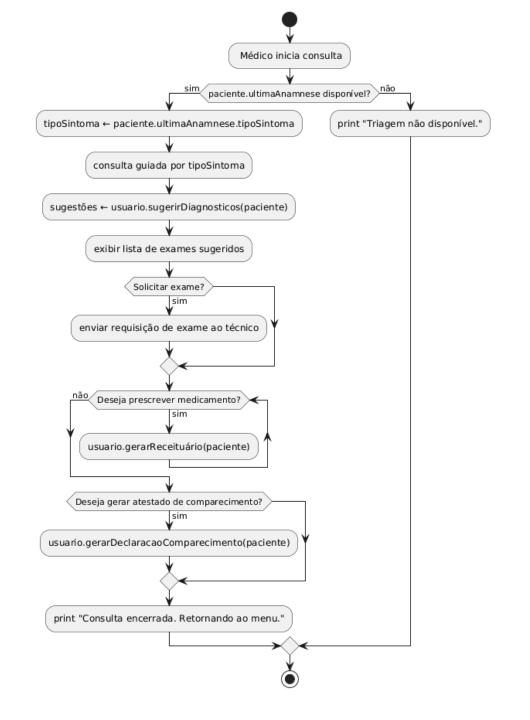
Triagem



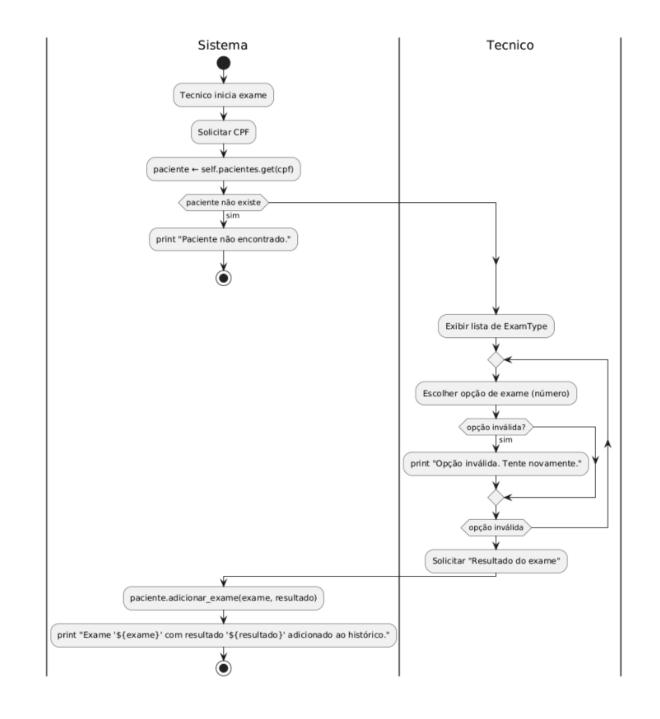
Triagem



Consulta



Exame



Herança

Superclasse Profissional reúne atributos e métodos comuns (nome, registro, login, hash de senha, contratos para cadastro/edição de pacientes) profissionais.

Enfermeiro herda autenticação e adiciona triagem(paciente: Paciente) para coletar sinais vitais e gerar Anamnese.

Medico herda e adiciona sugerir_diagnosticos(paciente: Paciente) além de geração de receituário e declaração (produz objetos Diagnostico).

Tecnico herda e implementa adicionar_exame_sistema(paciente: Paciente) para registrar resultados de exames profissionais.

Polimorfismo

Na classe **SistemaMediclass** (sistema.py), o método genérico login() retorna sempre um Profissional.

```
user = sistema.login()
# Polimorfismo: trata o retorno como Profissional,
# mas cada subclasse implementa seu próprio comportamento.
if hasattr(user, 'triagem'):
    user.triagem(paciente) # Enf. executa triagem
elif hasattr(user, 'sugerir_diagnosticos'):
    user.sugerir_diagnosticos(paciente) # Med. faz diagnóstico
```

Abstração e encapsulamento

Módulo Anamnese: define a classe Anamnese, que encapsula tudo que é relevante à triagem (freq. cardíaca, pressão, oximetria, perguntas e respostas, tipo de sintoma, timestamp) e expõe apenas to_dict() para uso externo anamnese.

Classe Paciente: mantém seu histórico médico em arquivo privado (_historico_file) e fornece métodos públicos como registrar_entrada(), atualizar_historico() e consultar_historico() para manipulação.

Classe Abstrata Profissional: oculta sua senha real em _senha_hash e só permite autenticação via autenticar(), impedindo acesso direto aos atributos profissionais.