Sistema MediClass

Matheus Marcondes de Oliveira (2022112517)
Graduando em Engenharia de Controle e Automação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
matheusmarcondes@ufmg.br, ORCID 0009-0004-8475-1406

***Abstract*—Este artigo apresenta o MediClass, um sistema de prontuário eletrônico e apoio à decisão clínica implementado em Python, que aplica os pilares da Programação Orientada a Objetos (herança, encapsulamento, abstração e polimorfismo) para modelar entidades como paciente, profissional de saúde, anamnese e diagnóstico. O sistema oferece uma interface de linha de comando para registro de pacientes, coleta de sinais vitais, sugestão de diagnósticos, geração de receituários e declarações, além de persistência de dados em JSON e arquivos texto de histórico clínico. A abordagem modular facilita a manutenção, a extensão e a reutilização de código, servindo como exercício prático dos conceitos vistos em sala de aula.**

Keywords—Programação Orientada a Objetos, Prontuário Eletrônico, Apoio à Decisão Clínica, Python, Interface CLI.

I. Introdução

O MediClass é um sistema desenvolvido para facilitar o registro de prontuários eletrônicos e oferecer suporte à decisão clínica em um ambiente hospitalar. Na prática, cada paciente é identificado unicamente pelo CPF: ao chegar, basta informar esse dado, e o sistema busca o histórico existente ou abre um novo cadastro caso seja a primeira vez. Todas as informações relevantes (dados pessoais, atendimentos anteriores, anamnese e resultados de exames) são armazenadas cronologicamente no histórico.

O acesso ao MediClass é feito por login, e cada tipo de usuário tem permissões específicas: enfermeiros realizam a triagem inicial, médicos conduzem a consulta propriamente dita e técnicos executam e registram exames. Durante a **triagem**, entendida como a avaliação preliminar da condição do paciente, o enfermeiro coleta sinais vitais (frequência cardíaca, pressão arterial e saturação de oxigênio) e classifica o quadro em uma das categorias de sintoma (gastrointestinal, respiratório, cardiovascular, dermatológico, trauma ou outros). Se algum valor de sinal vital estiver fora da faixa considerada segura, o sistema sinaliza prioridade para o paciente. Em seguida, são feitas perguntas de anamnese específicas ao tipo de sintoma escolhido — sem ainda sugerir diagnósticos, permitindo registrar nuances individuais de cada caso.

A **consulta médica** inicia-se quando o médico acessa o registro do paciente e visualiza os dados de triagem. Guiado pela mesma categoria de sintoma registrada na triagem, o profissional é conduzido a um conjunto de perguntas clínicas, cujas respostas levam a hipóteses de diagnóstico e à recomendação de uma lista de exames complementares, que podem ser agendados por um técnico. Ao final, o médico pode gerar dois documentos em formato texto: o **receituário**, com orientações de tratamento, e o **atestado** de comparecimento, ambos contendo informações do paciente, do médico e timestamps de geração.

Por fim, na etapa de **realização de exame**, o técnico seleciona o tipo de procedimento indicado, executa o exame e registra o resultado diretamente no histórico do paciente, fechando o ciclo de atendimento.

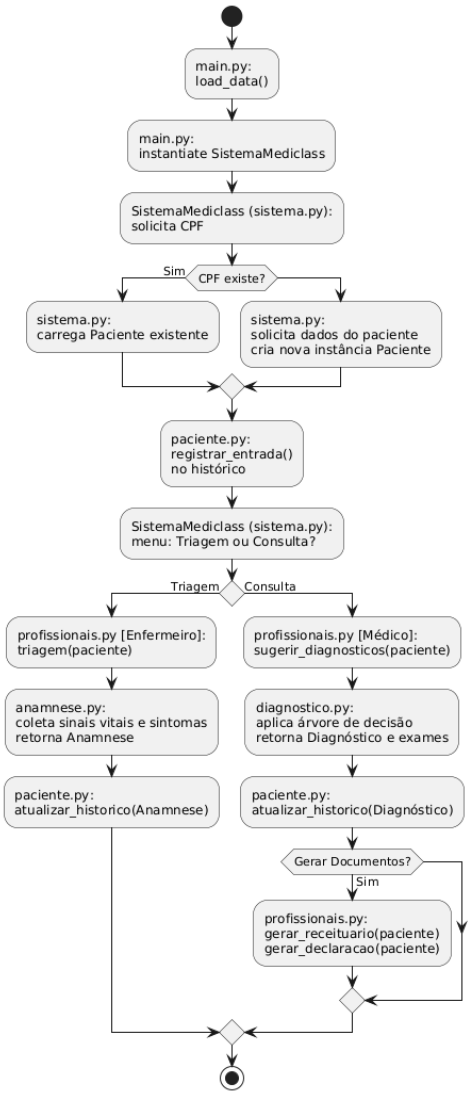


Fig. 1 – Fluxograma de um processo de cadastro, triagem e consulta no sistema

Como podemos ver na figura 1, o MediClass mantém um fluxo claro e integrado, garantindo que cada seja realizada pela equipe adequada e documentada de forma organizada, podendo o histórico do paciente ser exportado, também em formato texto (.txt).

II. MÉTODOS

A. Ferramentas e Tecnologias Utilizadas

O MediClass foi inteiramente desenvolvido em **Python 3**, aproveitando sua portabilidade e vasta biblioteca padrão. Para garantir a persistência de dados de pacientes e usuários, utiliza-se o módulo ***json***, que lê e grava em um arquivo único (mediclass_data.json) todo o estado do sistema. As credenciais dos profissionais são protegidas com **hashes SHA-256** gerados pelo módulo ***hashlib***, assegurando que senhas nunca sejam armazenadas em texto puro. As categorias de sintomas e tipos de exame são padronizadas via **enums** do pacote *enum*, enquanto o registro de horários e *timestamps* (presentes em entrada de pacientes, anamneses e geração de documentos) é tratado pelo módulo ***datetime***. Por fim, toda a interação com o usuário ocorre por meio de uma **interface de linha de comando** (CLI), construída sobre input e print, com verificação de entradas em alguns pontos específicos.

B. Estrutura dos arquivos e módulos

A arquitetura do sistema está dividida em seis módulos Python, cada um com responsabilidade bem definida. O arquivo **main.py** atua como ponto de partida: carrega dados salvos, instancia o objeto principal (SistemaMediclass) e trata a inicialização de usuários padrão. Em **sistema.py** reside a classe que gerencia a funcionalidade de login, exibe o menu principal e despacha chamadas para operações de registro, triagem, diagnóstico, exportação e exames. O módulo **profissionais.py** define a hierarquia de profissionais de saúde — a superclasse abstrata Profissional e as subclasses Medico, Enfermeiro e Tecnico — incluindo métodos para autenticação e roteiro de execução de tarefas clínicas pré-definidas para cada subclasse (que inclui a triagem, consulta, realização de exames e geração de documentos). Em **paciente.py** está a classe Paciente, responsável por armazenar dados cadastrais, histórico de atendimentos e gerenciamento de arquivos de entrada do histórico. Os arquivos **anamnese.py** e **diagnostico.py** modelam, respectivamente, a coleta de sinais vitais e sintomas, e a lógica de sugestão de possíveis diagnósticos, seguindo árvores de decisão definidas.

C. Modelagem Orientada a Objetos

O design do MediClass explora a programação orientada a objetos para manter o código coeso, extensível e manutenível. A **herança** aparece na criação de Profissional e suas três especializações, permitindo reaproveitar lógica comum de autenticação e abstração de interface. O **encapsulamento** é assegurado pela definição de atributos protegidos (por exemplo, _senha_hash em Profissional e _historico_file em Paciente, além das funções Enfermeiro.triagem(), Medico.sugerir_diagnosticos() e Tecnico.adicionar_exame_sistema(), que são acessíveis apenas por essas determinadas classes), evitando acessos indevidos. Graças ao **polimorfismo**, o fluxo de execução em SistemaMediclass trata de forma uniforme chamadas de triagem(), sugerir_diagnosticos() e adicionar_exame(), mesmo que cada subclasse implemente sua própria versão. Por fim, a **abstração** aparece na separação clara entre entidades de domínio (Paciente, Anamnese, Diagnóstico) e a interface de usuário, garantindo que alterações internas não impactem a forma como o usuário interage com o sistema.

D. Testes e validação

Para validar o comportamento esperado, foram realizados **testes de integração** diretamente na interface de linha de comando, cobrindo cenários cruciais: autenticações válidas e rejeitadas, registro de pacientes novos e recorrentes, triagens com e sem alertas de risco, execução de consultas médicas com sugestão de diagnósticos e exportação de arquivos de prescrição e atestado, além do despacho de exames para técnicos. Cada etapa foi verificada quanto à correta atualização do histórico, geração de documentos com cabeçalhos e timestamps adequados e aplicação das regras de permissão por tipo de profissional, garantindo a confiabilidade e a robustez do fluxo de atendimento completo.

III. RESULTADOS

A. Funcionalidades implementadas

O MediClass provê um conjunto completo de operações clínicas via CLI: **autenticação de profissionais** (médicos, enfermeiros e técnicos), **registro e recuperação de prontuários por CPF**, **realização de triagem de enfermagem** (coleta de sinais vitais e anamnese segmentada por tipo de sintoma), **execução de consultas médicas com sugestão de diagnósticos** baseados em árvore de decisão, **geração de prescrições e atestados** em arquivos de texto padronizados e **despacho de exames** para técnicos. Cada etapa está devidamente controlada por permissões, garantindo que apenas o perfil adequado execute a operação correspondente.

B. Fluxo e persistência de dados

Os dados transitam entre três camadas principais: **JSON** (atendendo a uma recomendação técnica da disciplina), para armazenamento de longo prazo (usuários e pacientes), **objetos em memória** durante a execução do sistema e **arquivos texto individuais** (histórico clínico, receituários e atestados). Ao registrar a entrada, o paciente recebe timestamp e a informação é gravada no histórico. Os métodos de triagem, consulta e exames atualizam esse histórico tanto em memória quanto em disco, assegurando que todas as ações fiquem registradas cronologicamente.

C. Arquitetura

A arquitetura do sistema foi concebida levando em consideração os princípios da programação orientada a objetos. Sendo assim, na forma como foi construído, projetos de expansão são tarefas intuitivas.

Como prova da fácil expansão, novas funcionalidades foram desenhadas para implementação: a migração para banco de dados relacional, a adição de uma interface gráfica ou web, a incorporação de relatórios gerenciais automáticos e a inclusão de métricas via uma classe Gerente para monitoramento de produtividade (avaliando chamadas dos métodos) por profissional. Essas melhorias confortam a proposta inicial de facilitar a manutenção e a escalabilidade.

D. Demonstração

Nesta seção, será apresentada uma demonstração passo a passo do fluxo inicial de uso do MediClass, mostrando as interações através do prompt e representando as ações.

1) *Processo de autenticação*

```
Mensagens do sistema          Entradas do usuário
Login:
                                                enf
Senha:
                                              senha
```

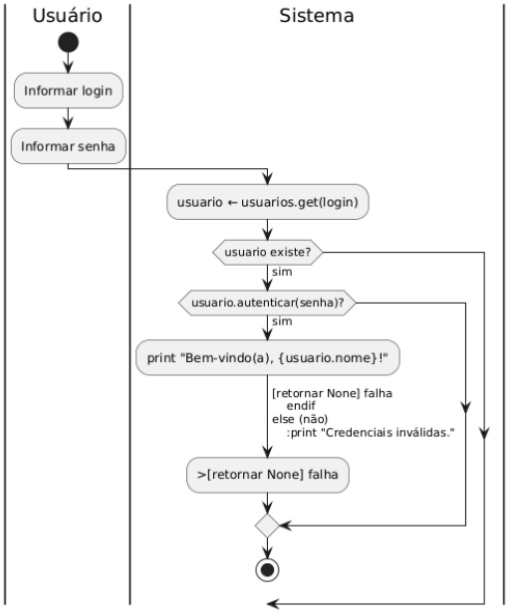


Fig. 2 – Fluxograma do processo de autenticação

O processo de autenticação inicia-se quando o usuário informa seu login e senha pelo prompt; em seguida, o sistema busca o objeto Profissional correspondente ao login fornecido, calcula o hash SHA-256 da senha digitada e o compara ao hash armazenado; se o usuário existir e os hashes coincidirem, o sistema exibe uma mensagem de boas-vindas e retorna o objeto Profissional autenticado, permitindo acesso ao menu principal com as permissões adequadas; caso contrário, informa "Credenciais inválidas" e encerra ou reinicia o fluxo de login.

```
Mensagens de falha
Credenciais inválidas.
Encerrando sistema
```

```
Mensagens de sucesso
Bem-vindo(a), Enf. Teste!

--- Menu Principal ---
1. Registrar entrada de paciente
2. Realizar triagem (enfermeiro)
3. Realizar consulta (médico)
4. Visualizar prontuário
5. Exportar prontuário (.txt)
6. Adicionar exame (técnico)
0. Logout
Escolha uma opção:
```

2) *Registrar entrada de paciente*

```
Mensagens do sistema          Entradas do usuário
CPF do paciente:
                                       12345678900
Paciente não encontrado.
Cadastrando novo.
Nome completo:
                                  Matheus de Oliveira
Contato:
                                       31900000000
Convênio:
                                               SUS
Data de nascimento (YYYY-MM-DDD):
                                        2000-01-01
Leito:
                                                1A
Entrada registrada.
```

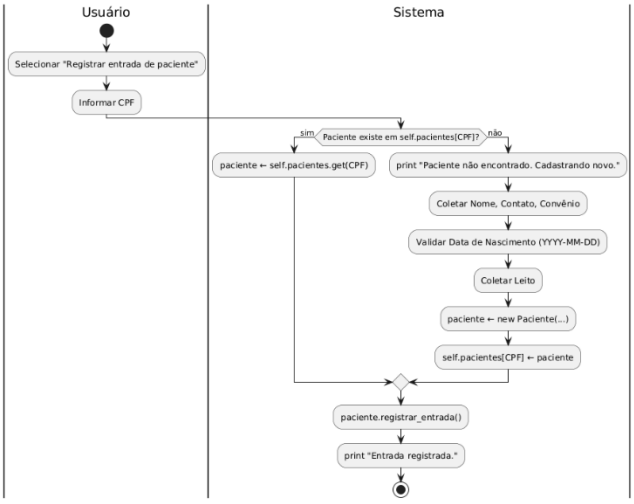


Fig. 3 – Fluxograma do processo de registro de paciente

No Sistema MediClass, quando um usuário autenticado escolhe "Registrar entrada de paciente", o método op_registrar_entrada da classe SistemaMediclass é chamado, podendo ser acessado por qualquer objeto Profissional que tenha feito login; o sistema então solicita o CPF do paciente e faz uma busca em memória no dicionário self.pacientes via get(cpf); se o CPF já estiver registrado, o objeto Paciente existente é recuperado; se não for encontrado, exibe-se "Paciente não encontrado. Cadastrando novo.", coleta-se nome, contato, convênio, data de nascimento e leito, instancia-se um novo Paciente (incluindo o nome do enfermeiro responsável) e armazena-se no dicionário; por fim, independentemente de novo ou existente, chama-se paciente.registrar_entrada() para gravar a data de entrada no histórico e imprime-se "Entrada registrada.".

3) *Tentativa de acesso indevido a método protegido*

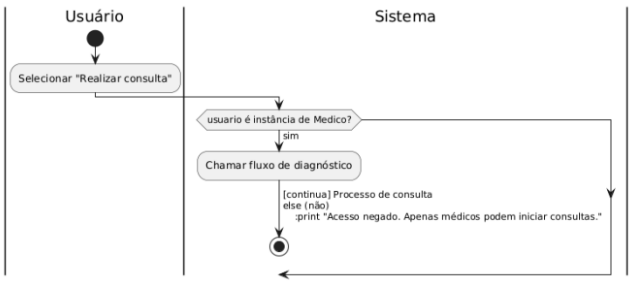


Fig. 4 – Fluxograma do processo de verificação da classe para execução de métodos protegidos

O acesso indevido a um método protegido, como tentar iniciar uma consulta quando quem está logado é um enfermeiro, segue este fluxo: o usuário seleciona a opção

"Realizar consulta", disparando a chamada a SistemaMediclass.op_diagnostico(usuario); dentro desse método, logo no início há uma verificação de tipo (isinstance(usuario, Medico)); se o objeto usuario não for uma instância de Medico (por exemplo, um Enfermeiro), o sistema imprime

```
v
Acesso negado.
Apenas médicos podem iniciar consultas.
```

e retorna imediatamente ao menu principal.

4) ***Triagem***

Simularemos uma triagem de paciente que apresenta pressão arterial sistólica incomum, (e portanto, esperamos que uma flag de prioridade seja ativada), e vários sintomas gastrointestinais.

```
Mensagens do sistema                Entradas do usuário

CPF do paciente
                                              12345678900
Frequência cardíaca (40-200 bpm):
                                                       90
Pressão arterial sistólica (70-250 mmHg):
                                                      140
Pressão sistólica incomum! Ativando prioridade.
Pressão arterial diastólica (40-150 mmHg):
                                                       80
Oximetria de pulso (85-100%):
                                                       99
Selecione o tipo de sintoma:
1. gastrointestinal
2. respiratório
3. cardiovascular
4. trauma
5. dermatológico
6. outros
                                                        1
Náuseas ou vômitos (S/N):
                                                        S
Diarreia (S/N):
                                                        S
Constipação (S/N):
                                                        S
Dor abdominal (S/N):
                                                        S
Perda de peso inexplicada (S/N):
                                                        S
Febre (S/N):
                                                        S
Triagem concluída.
```

Quando um enfermeiro autenticado seleciona "Realizar triagem", o método SistemaMediclass.op_triagem(usuario) é invocado, verifica-se que usuario é instância de Enfermeiro (caso contrário imprime "Acesso negado. Apenas enfermeiros podem realizar triagem." e aborta), solicita-se o CPF e faz-se uma busca em self.pacientes (se não encontrado imprime "Paciente não encontrado." e encerra) e, se houver registro, chama-se Enfermeiro.triagem(paciente), onde são coletados sinais vitais (frequência cardíaca, pressão arterial sistólica e diastólica, oximetria), avalia-se se algum valor está fora da faixa para ativar sinal de prioridade, solicita-se a seleção do tipo de sintoma e respondem-se perguntas de anamnese específicas, instancia-se um objeto Anamnese, atualiza-se o histórico do paciente (incluindo flag de prioridade, se aplicável) e armazena-se essa última anamnese em paciente.ultima_anamnese; por fim, exibe-se "Triagem concluída."

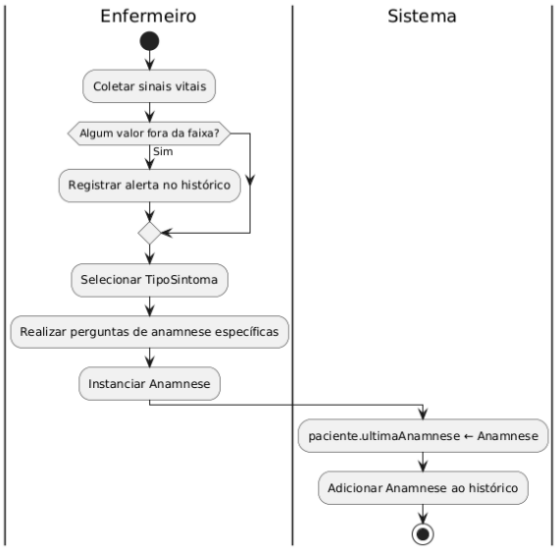


Fig. 5 – Fluxograma simplificado do processo de triagem

É importante acrescentar que as perguntas respondidas durante a anamnese não são levadas em conta para determinar o diagnóstico. São perguntas direcionadas ao tipo de sintoma, mas apenas durante a consulta será possível avaliar com precisão. Um fluxograma que detalha precisamente o processo de triagem está disponível na documentação do projeto.

5) ***Consulta***

Vamos simular uma consulta. Como vimos, é necessário que o enfermeiro faça logout pelo menu principal, e o médico faça seu login. Nosso paciente hipotético apresenta um quadro clínico de disfunção intestinal. Com o tipo de sintoma já informado durante a triagem, vamos dar a início à consulta informando o CPF do paciente. Após isso, o sistema guiará para análise de sintomas mais específicos para determinar o diagnóstico. Com base nas respostas, a árvore de decisões, que consta nos detalhes do projeto determinará o diagnóstico.

```
Mensagens do sistema                Entradas do usuário

CPF do paciente
                                              12345678900
Categoria: Gastrointestinal
1. Desconforto abdominal
2. Disfunção intestinal
3. Dor aguda
Selecione o subgrupo (1-3):
                                                        2
Febre + náuseas + diarreia? (S/N):
                                                        S
--- Diagnósticos sugeridos ---
Gastrointestinal: Possível gastroenterite
infecciosa (Exames sugeridos: Hemograma,
Coprocultura)
Deseja solicitar exame a um técnico? (S/N):
                                                        S
Solicitação enviada.

Deseja gerar receituário? (S/N):
                                                        S
...
```

```
Iniciando prescrição. Digite 0 para finalizar.
Nome da medicação (ou 0 para terminar):
                                        Ciprofloxacino
Posologia [miligramas]:
                                                   500
Intervalo das doses [horas]:
                                                    12
Período de tratamento [dias]:
                                                     5
Nome da medicação (ou 0 para terminar):
                            Lactobacillus acidophilus
Posologia [miligramas]:
                                                     2
Intervalo das doses [horas]:
                                                    24
Período de tratamento [dias]:
                                                    14
Nome da medicação (ou 0 para terminar):
                                                     0
Receituário exportado para
receituario_12345678900.txt

Deseja gerar declaração de comparecimento? (S/N):
                                                     S
Declaração exportada para
declaracao_12345678900.txt

Consulta encerrada. Retornando ao menu. Iniciando
prescrição. Digite 0 para finalizar.
Nome da medicação (ou 0 para terminar):
                                        Ciprofloxacino
Posologia [miligramas]:
                                                   500
Intervalo das doses [horas]:
                                                    12
Período de tratamento [dias]:
                                                     5
Nome da medicação (ou 0 para terminar):
                            Lactobacillus acidophilus
Posologia [miligramas]:
                                                     2
Intervalo das doses [horas]:
                                                    24
Período de tratamento [dias]:
                                                    14
Nome da medicação (ou 0 para terminar):
                                                     0
Receituário exportado para
receituario_12345678900.txt

Deseja gerar declaração de comparecimento? (S/N):
                                                     S
Declaração exportada para
declaracao_12345678900.txt

Consulta encerrada. Retornando ao menu.
```

Quando um médico autenticado seleciona “Realizar consulta”, o método op_diagnostico da classe SistemaMediclass é invocado, verificando inicialmente se o objeto usuario é instância de Medico; em seguida, solicita-se o CPF e faz-se uma busca em self.pacientes; se o paciente não existir, imprime “Paciente não encontrado.” e encerra o fluxo.

Caso exista, chama-se sugerir_diagnosticos(paciente), que executa perguntas com base no tipo de sintoma e determina possíveis diagnósticos com base em respostas condicionais à essas perguntas (de afirmativas/negativas); caso haja sugestões, elas são listadas, como formatado na classe Diagnostico, junto a uma lista de exames indicados para confirmar ou descartar a suspeita. Se a lista estiver vazia, exibe “Nenhum diagnóstico sugerido.”.

Depois o sistema pergunta se o médico deseja solicitar exame a um técnico (atualizando o histórico), gerar receituário e gerar declaração de comparecimento (invocando gerar_receituario e gerar_declaracao_comparecimento), finalizando com “Consulta encerrada. Retornando ao menu.” e retornando ao menu principal.

Enquanto solicitar o agendamento de exame a um técnico no momento apenas imprime que a solicitação foi realizada, as funções de exportação de documentos geram arquivos de texto. Para gerar uma prescrição médica, o sistema entra em um loop que solicita a entrada de nome da medicação, posologia, intervalo entre as doses e duração do tratamento, até que se entre com o valor “0” em um nome de medicação. Esses dados são concatenados em uma formatação específica, contendo dados do paciente, médico e timestamps, e então exportados para um arquivo TXT. A declaração de comparecimento, de forma semelhante, concatena os dados do paciente (em especial seu nome, CPF e horário de entrada) e do médico e resulta em um arquivo TXT formatado.

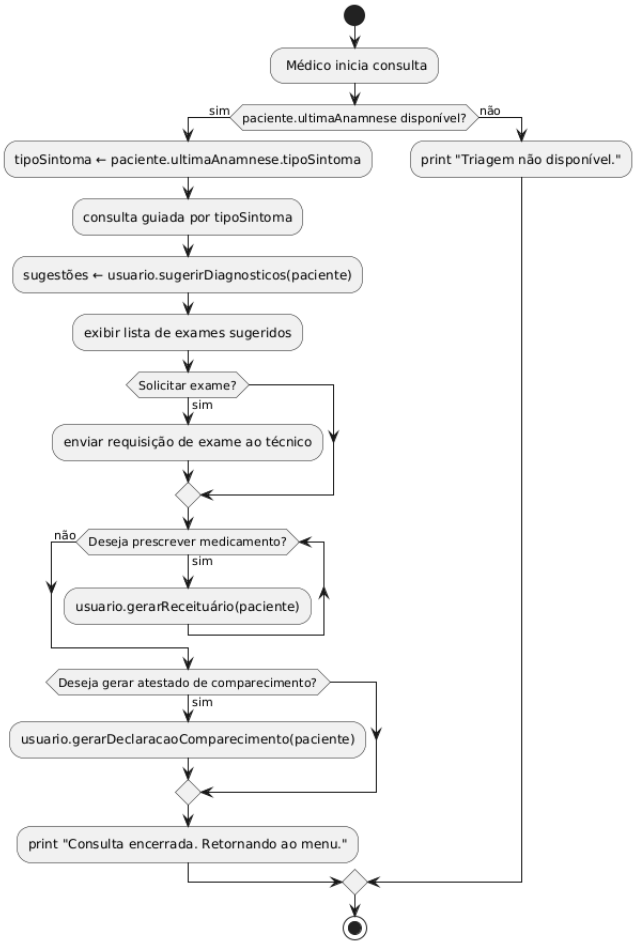


Fig. 6 – Fluxograma simplificado do processo de consulta

No fluxograma acima, podemos ver de forma simplificada o processo de uma consulta aberto por um usuário da classe Medico. A determinação dos diagnósticos pelas respostas aos questionamentos foi dispensada nesta ilustração por fins de praticidade, mas um fluxograma completo pode ser acessado pela documentação detalhada do projeto.

Um exemplo de prescrição médica gerado pelo sistema:

```
receituario_12345678900.txt

Hospital da Escola de Engenharia da UFMG
Sistema MediClass
Prescrição gerada em 2025-06-23 12:26

Paciente: Matheus de Oliveira (CPF: 12345678900)
Médico: Dr. Teste (CRM: CRM123)

- Ciprofloxacino: 500mg a cada 12 horas, durante 5 dias;
- Lactobacillus acidophilus: 2mg a cada 24 horas, durante 14 dias;


Dr. Teste - CRM123
```

E um exemplo de declaração de comparecimento:

```
receituario_12345678900.txt

Hospital da Escola de Engenharia da UFMG
Sistema MediClass
Prescrição gerada em 2025-06-23 12:26

   Eu, Dr. Teste, CRM123, atesto para os devidos fins que o paciente Matheus de Oliveira, CPF 12345678900, compareceu a consulta clínica no dia 2025-06-23, dando entrada no hospital às 12:27, apresentando quadro sintomático gastrointestinal.

   Cordialmente,
Dr. Teste - CRM123
```

6) *Adicionar exame*

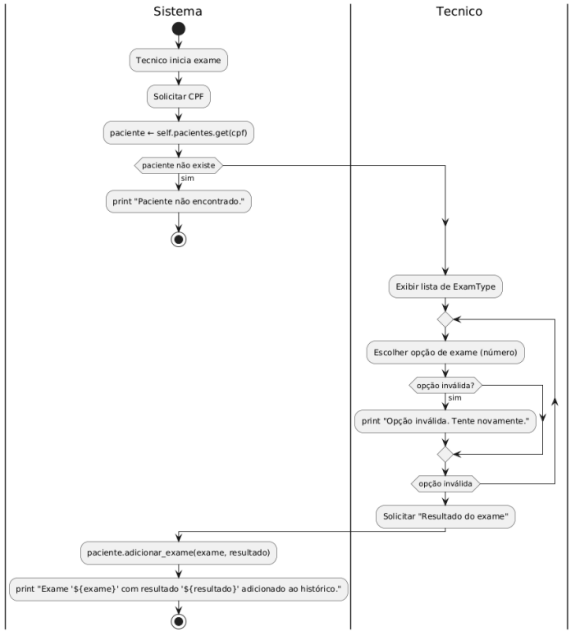


Fig. 6 – Fluxograma do processo de exame

No Sistema MediClass, quando um técnico autenticado escolhe "Adicionar exame", o método op_adicionar_exame de SistemaMediclass é invocado. Ele verifica se o usuário é instância de Tecnico, exibindo "Acesso negado. Apenas técnicos podem adicionar exames." e abortando caso contrário. Em seguida solicita o CPF e faz uma busca em self.pacientes via get(cpf); se o paciente não for encontrado, imprime "Paciente não encontrado.".

Caso exista, invoca adicionar_exame_sistema(paciente), que lista os tipos de exame definidos no enum ExamType, definido no próprio módulo, entra em um loop até receber uma escolha numérica válida, solicita o resultado do exame, que é uma string, e chama paciente.adicionar_exame(exame, resultado) para armazenar o exame e o laudo (resultado) no objeto e no histórico em arquivo, finalizando com "Exame '…' com resultado '…' adicionado ao histórico.".

7) *Exportação de prontuário*

No Sistema MediClass, quando um profissional autenticado escolhe "Exportar prontuário", o método op_exportar_prontuario é invocado; ele solicita o CPF do paciente, faz uma busca em self.pacientes com get(cpf); se não encontrar, exibe "Paciente não encontrado." e retorna; se encontrar, monta um arquivo TXT nomeado prontuario_{cpf}.txt contendo cabeçalhos (nome, CPF, contato, convênio, data de nascimento, leito, enfermeiro que realizou a triagem e flag de prioridade), adiciona todo o histórico médico via paciente.consultar_historico(), inclui os dados da última anamnese se disponível e grava o conteúdo no arquivo, finalizando com a mensagem "Prontuário exportado para {filename}".

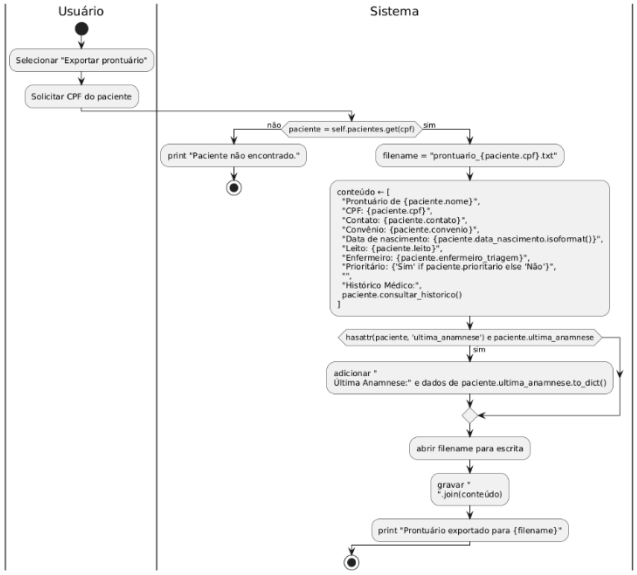


Fig. 7 – Fluxograma do processo de exportação de prontuário

O fluxograma acima ilustra o processo de exportação do documento. Uma versão de exemplo pode ser acessada na documentação detalhada do projeto.

IV. DISCLAIMER

A classificação de sintomas e a interpretação de valores de sinais vitais aqui adotadas representam aproximações baseadas em parâmetros usualmente descritos na literatura médica, ainda que tenhamos buscado a maior fidelidade possível. As sugestões de diagnóstico geradas pelo sistema têm caráter meramente demonstrativo, servindo como ilustração da lógica de um modelo orientado a objetos, e não constituem parecer clínico. Para avaliação definitiva e conduta adequada, é imprescindível a consulta de um médico habilitado.

Mais informações sobre o projeto podem ser acessadas em https://github.com/matheusmarcondes1/mediclass.

REFERÊNCIAS

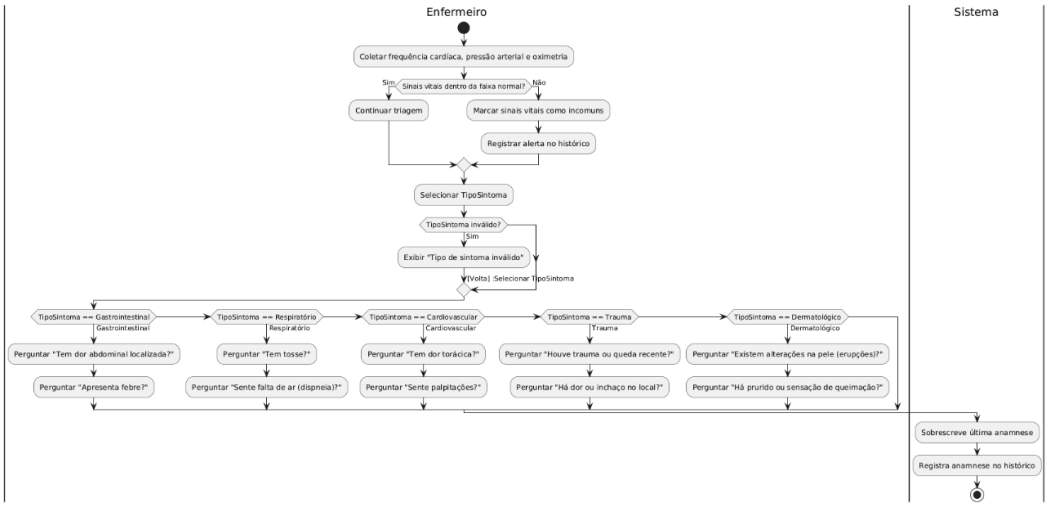
[1] MARTIN, R. C. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall, 2008.

[2] LARMAN, C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design. 2ª ed. Prentice Hall, 2007.

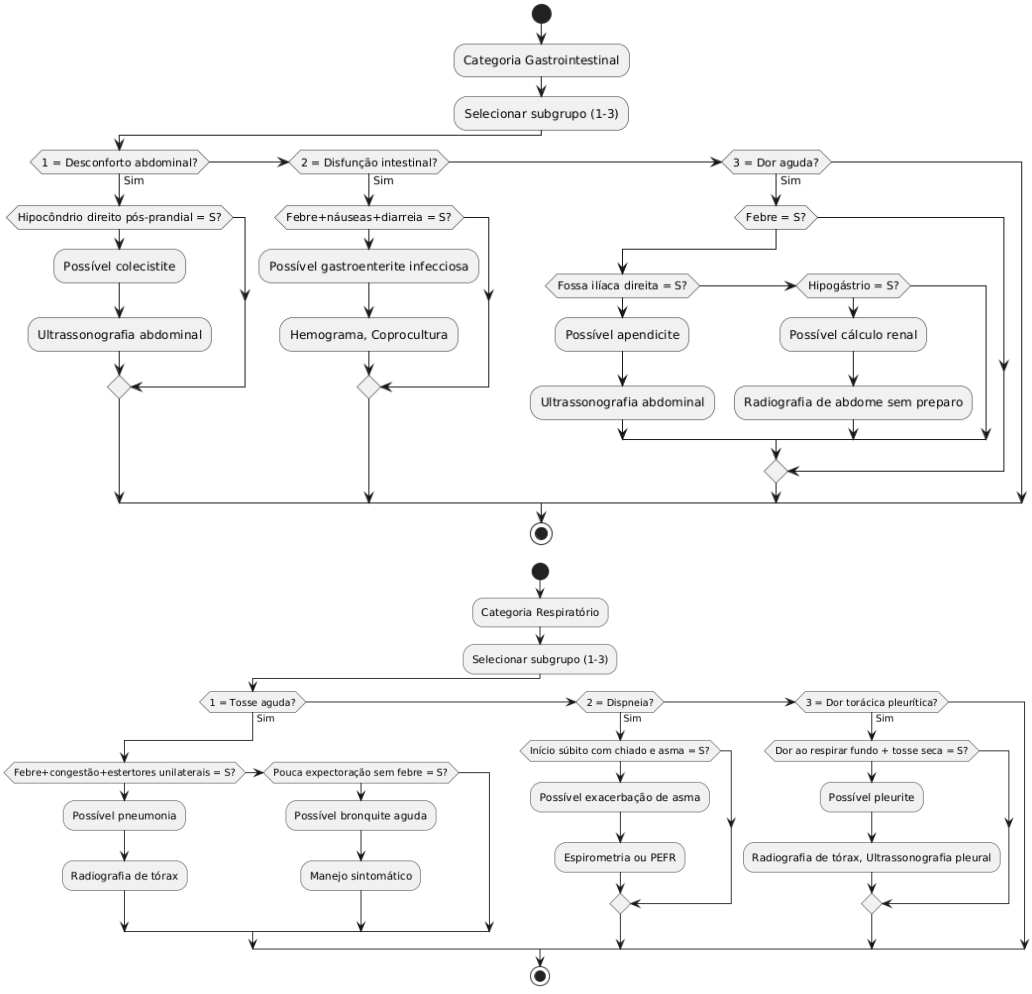
[3] GUYTON, A. C.; HALL, J. E. Tratado de Fisiologia Médica. 12ª ed. Elsevier, 2011.

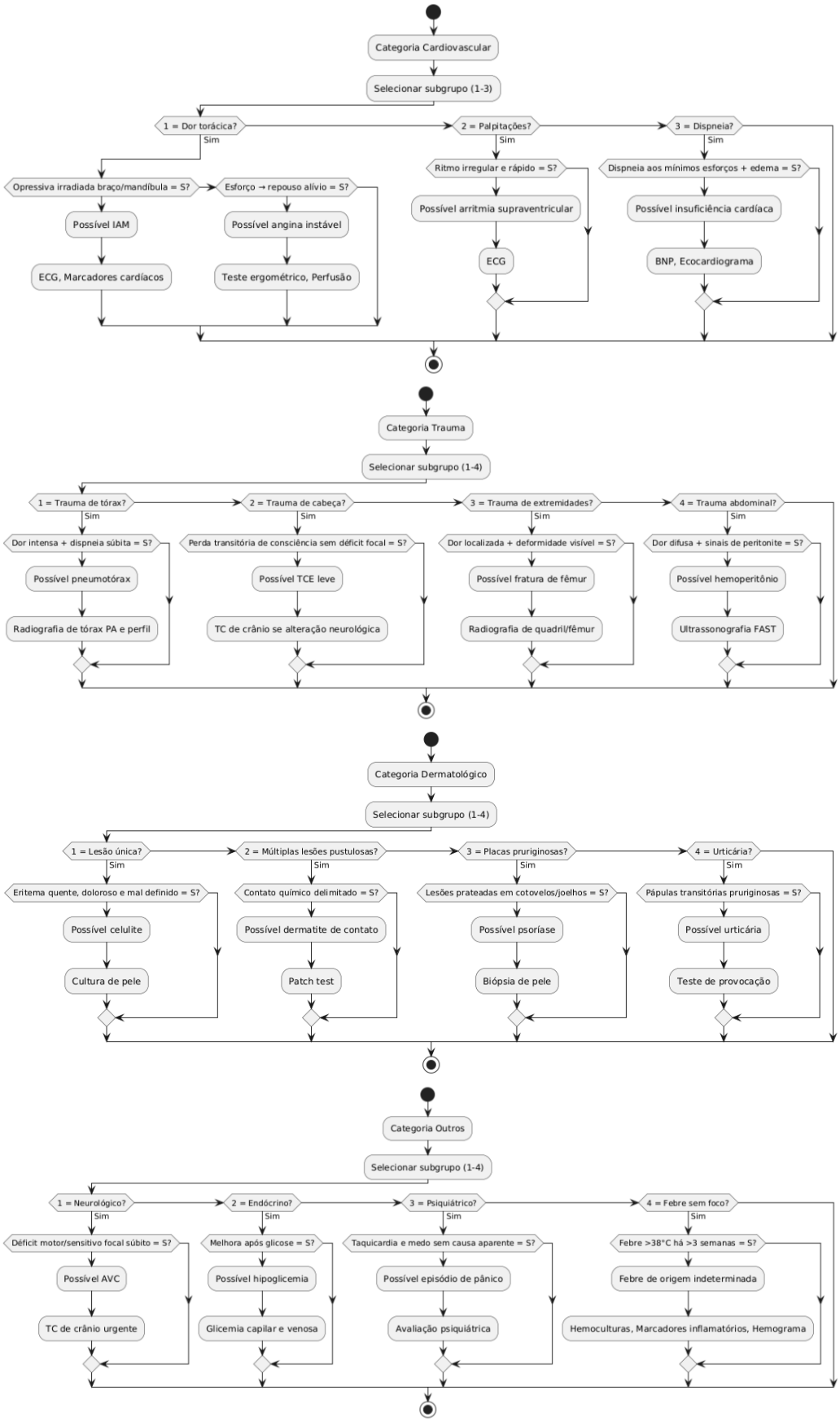
[4] NUNES LOPES, G. *POO – Objeto e Classe (Notas de aula), UFMG, 2025*.

APÊNDICE A: FLUXOGRAMA COMPLETO DO MÉTODO DE TRIAGEM



APÊNDICE B: ÁRVORES DE DECISÃO PARA ELABORAÇÃO DO DIAGNÓSTICO CLÍNICO




Categoria Cardiovascular
Selecionar subgrupo (1-3)
1 = Dor torácica?
2 = Palpitações?
3 = Dispneia?
Sim
Opressiva irradiada braço/mandíbula = S?
Esforço → repouso alívio = S?
Possível IAM
Possível angina instável
ECG, Marcadores cardíacos
Teste ergométrico, Perfusão
Ritmo irregular e rápido = S?
Possível arritmia supraventricular
ECG
Dispneia aos mínimos esforços + edema = S?
Possível insuficiência cardíaca
BNP, Ecocardiograma
Categoria Trauma
Selecionar subgrupo (1-4)
1 = Trauma de tórax?
2 = Trauma de cabeça?
3 = Trauma de extremidades?
4 = Trauma abdominal?
Dor intensa + dispneia súbita = S?
Possível pneumotórax
Radiografia de tórax PA e perfil
Perda transitória de consciência sem déficit focal = S?
Possível TCE leve
TC de crânio se alteração neurológica
Dor localizada + deformidade visível = S?
Possível fratura de fêmur
Radiografia de quadril/fêmur
Dor difusa + sinais de peritonite = S?
Possível hemoperitônio
Ultrassonografia FAST
Categoria Dermatológico
Selecionar subgrupo (1-4)
1 = Lesão única?
2 = Múltiplas lesões pustulosas?
3 = Placas pruriginosas?
4 = Urticária?
Eritema quente, doloroso e mal definido = S?
Possível celulite
Cultura de pele
Contato químico delimitado = S?
Possível dermatite de contato
Patch test
Lesões prateadas em cotovelos/joelhos = S?
Possível psoríase
Biópsia de pele
Pápulas transitórias pruriginosas = S?
Possível urticária
Teste de provocação
Categoria Outros
Selecionar subgrupo (1-4)
1 = Neurológico?
2 = Endócrino?
3 = Psiquiátrico?
4 = Febre sem foco?
Déficit motor/sensitivo focal súbito = S?
Possível AVC
TC de crânio urgente
Melhora após glicose = S?
Possível hipoglicemia
Glicemia capilar e venosa
Taquicardia e medo sem causa aparente = S?
Possível episódio de pânico
Avaliação psiquiátrica
Febre >38°C há >3 semanas = S?
Febre de origem indeterminada
Hemoculturas, Marcadores inflamatórios, Hemograma