

1.1 Problema

Como desenvolver uma aplicação computacional capaz de auxiliar o hospital no processo de triagem e dar suporte aos médicos no processo de previsão de diagnósticos usando uma base de conhecimento?

1.2 Justificativa

A proposta em questão veio pela ausência de um sistema especialista que realiza o processo de triagem em qualquer lugar e sem a presença de um especialista na área da medicina (Triagem Médica) de forma a auxiliar os profissionais da área de saúde na detenção ou previsão de paludismo e febre tifoide.

1.3 Objectivo geral

Desenvolver um protótipo de sistema especialista para a realização da triagem médica no hospital municipal da samba.

1.4 Objectivos específicos

Os objectivos específicos deste trabalho estão compreendidos em:

- Levantar uma bibliografia acerca de gestão hospitalar.
- Levantar uma bibliografia acerca dos sistemas especialistas.
- Pesquisar procedimentos tradicionais de triagem.
- Desenvolver um protótipo de sistemas especialista em triagem médica.
- Testar a viabilidade.

1.5 Objecto de estudo

Sistema especialista para área de medicina.

1.6 Campo de acção

Inteligência Artificial.

1.7 Organização do trabalho

O presente trabalho está dividido em 4 capítulos, com a seguinte distribuição:

- **Capítulo I:** onde está contido a introdução e alguns elementos pré-textuais.
- **Capítulo II:** onde está contido a fundamentação teórica na qual fez-se uma abordagem geral sobre a triagem, inteligência artificial , sistemas especialistas, focando na sua constituição, realimentação, importância, etc. Também encontra-se o estado de arte na qual é apresentado alguns sistemas relacionados com o sistema proposto.
- **Capítulo III:** onde está contido as metodologias aplicadas no projecto.
- **Capítulo IV:** onde está contido as 4 fases do desenvolvimento do sistema que são: a iniciação (modelação de processos de negócios, entrevistas, linguagens e ferramentas utilizadas), elaboração (modelagem do sistema ou seja desenho da solução), construção (procedimentos para construção do sistema ou seja codificação) e transição (testes e plano de implantação).

CAPÍTULO II: FUNDAMENTAÇÃO TEÓRICA

2.1 Atendimento Hospitalar

O processo de atendimento em hospitais é sem dúvida uma área com grande necessidade de profissionalismo, por causa do grau de seriedade da actividade, visto que por se tratar de vidas humanas torna-se num assunto muito delicado, mas infelizmente ainda sente-se a falta de uma boa gestão profissional e eficaz nessa área.

Um hospital é uma organização que presta serviços profissionais e que apresenta funções diferentes, qualificando-se, portanto, como uma organização bastante complexa. Os hospitais devem atender as necessidades do paciente oferecendo ações e serviços com a máxima qualidade. O processo de qualidade é essencial para realização de qualquer serviço de saúde, objetivando o atendimento das necessidades, desejos e expectativa dos seus pacientes.

A cada ano que passa na nossa sociedade, surge a necessidade de melhorar o processo de atendimento nos hospitais, implementando assim o atendimento hospitalar de qualidade. O atendimento de qualidade faz toda a diferença em uma instituição, pois promove a satisfação por parte de seus clientes. Existem vários factores que podem influenciar no processo de atendimento como: a estrutura física, o espaço de repouso, o uso de tecnologias para garantir maior agilidade nos processos e evitar erros.

De acordo com Castellar, Mordelet e Grabois (2003), citado por Vidal et al (2013):

As características da qualidade de um hospital baseam-se fundamentalmente em três pilares: Qualidade intrínseca, que se refere aos cuidados hospitalares; custo e atendimento. Outros aspectos devem ser considerados como a segurança, tanto para o paciente como para os funcionários durante a realização de procedimentos e serviços em geral e a qualidade moral dos funcionários, que pode ser refletida para a instituição como um todo.

O Sistema de saúde público de Portugal é uma das referência por conta de sua qualidade e preocupação com seus pacientes. Os hospitais são bem equipados, não há necessidade de espera demasiada para o atendimento e a qualidade desses serviços é evidente.

Queza (2010) afirma que :

Em angola o sector público permanece como o principal prestador dos cuidados de saúde a nível nacional verificando-se algumas dificuldades na prestação de cuidados de saúde e sem a qualidade desejada na maioria dos casos.

2.2 Procedimentos Básicos de Atendimento Hospitalar

Para que um hospital proporcione um atendimento de qualidade aos seus pacientes e tenha um desempenho com máxima eficiência e eficácia nas suas atividades devem ser aplicadas alguns procedimentos básicos que são:

1. Treinamento da equipa de trabalho;
2. Ter um espaço adequado para receber os pacientes;
3. Ter uma equipa responsável pela triagem;
4. Utilizar tecnologias para a organização dos dados;
5. Investir numa boa comunicação interna.

2.3 Triagem Inicial

Segundo Moreira (2010, p. 9) o termo “Triagem” é de origem francesa “*trier*” que significa separar ou seleccionar. Triagem é um processo sistemático para determinar quem vai ser visto e tratado primeiro. Foi introduzido nos Serviços de Emergência (SE) para tentar minimizar o problema da superlotação, permitindo cuidados imediatos para os pacientes mais urgentes.

A triagem inicial consiste no processo de gerenciamento e recolha de informações do paciente, tais como os seus dados pessoais e histórico de saúde. A triagem inicial é feita por enfermeiros especializados com objectivo de agilizar a consulta médica.

No processo de triagem o enfermeiro verifica e analiza os sintomas dos pacientes, identifica os casos mais graves, manda para as emergências e nos casos normais encaminha os pacientes para uma marcação de consulta.

2.4 Paludismo

É uma doença causada por parasitas do género plasmodium. É transmitida ao ser humano pela picada do mosquito Anopheles, existe em diversos países dos continentes: Americano, Africano e Asiático. O período de incubação entre a picada do mosquito infectado e o começo dos sintomas depende da espécie de parasita. O período mínimo de incubação corresponde a 7/8 dias, podendo ser mais longo (4 semanas). Alguns dos sintomas apresentados pelo paludismo são: febre, dor de cabeça, fraqueza e cansaço, suores e calafrios e náuseas e vômitos.

2.4.1 Métodos de obtenção do diagnóstico

Segundo Departamento de Vigilância Epidemiológica(2005, p.16) a ausência de parâmetros clínicos específicos que permitam confirmar a infecção a 100% justifica a

necessidade de métodos laboratoriais para o diagnóstico da malária. Tradicionalmente, o diagnóstico confirmatório da malária (paludismo) é feito pelo exame microscópico do sangue, necessitando de materiais e reagentes adequados, bem como de técnicos bem treinados para sua realização, objetivando a detecção e diferenciação das espécies de plasmódios. O exame microscópico do sangue pode ser feito em **esfregaço delgado (distendido) ou espesso (gota espessa)**, sendo a gota espessa a mais usual.

2.5 Febre Tifoide

A febre tifoide é uma doença bacteriana aguda, de gravidade variável. É causada pela *Salmonella Typhi*, que é uma bactéria com morfologia de bacilo Gram negativo, móvel, pertencente à família Enterobacteriaceae. Se transmite, na maioria das vezes, através de comida contaminada por portadores, durante o processo de preparação e manipulação dos alimentos. A água também pode ser um veículo de transmissão, podendo ser contaminada no próprio manancial (rio, lago ou poço) ou por ser tratada inadequadamente.

O período de incubação da doença é de 1 a 3 semanas, em média 2, podendo ser curto como três dias e longo até 56 dias em função da dose infectante e da facilidade de proliferação do agente em determinados alimentos.

2.5.1 Métodos de obtenção do diagnóstico

Segundo Ramos(2005, p.24) o diagnóstico da febre tifoide pode ser clínico ou laboratorial. O diagnóstico clínico torna-se muito difícil principalmente nos primeiros dias porque existem outras doenças que apresentam os mesmos sintomas. Os principais exames laboratoriais utilizados para a deteção da febre tifoide são: a **hemocultura**, a **coprocultura** e a **reação Widal**. A deteção da mesma também pode ser obtida pela urina e da medula óssea.

2.6 Automação de Processos

Processos correspondem a uma sequência de passos ou actividades bem estruturadas, tendo como resultado um produto específico. São utilizados para criar, inventar, projetar, transformar, produzir, controlar, manter e usar produtos ou sistemas.

Automação é um conjunto de mecanismos e funções para a realização de uma determinada tarefa sem a necessidade de intervenção humana.

Segundo Bayer e Araújo (2011, p. 11):

Automatizar e controlar um processo significa atuar sobre ele ou sobre às condições as quais o processo está sujeito, de modo a manter variáveis e quantidades estáveis com o

passar do tempo, mesmo que interferências externas tentem desviá-lo desta condição. A utilização de sistemas de controle automático se encontra difundida no dia a dia de todas as sociedades desenvolvidas. Tais sistemas agem como elementos decisivos na tentativa de se obterem progresso e desenvolvimento. Podemos dizer que o controle automático num processo produtivo tem a finalidade de otimizar os sistemas capazes de produzir bens com menor custo, com maior quantidade, em menor tempo e com maior qualidade.

A automatização de processos pode ser feita de forma manual ou automática, sendo a automática a mais utilizada. Exemplo: a utilização de softwares.

2.6.1 Vantagens da Automação de Processos

A automação de processos possui várias vantagens aplicadas na realização das actividades. Essas vantagens são:

- Reduz o tempo de execução da tarefa;
- Ajuda a optimizar a tomada de decisão;
- Diminuição de tarefas repetitivas;
- Integração das informações;
- Padronização dos serviços, etc.

2.6.2 Soluções Automatizadas para Gestão de Atendimento Hospitalar

A automação de processos na área da saúde é de extrema importância, pois agiliza o processo de atendimento dos pacientes, diminuindo assim o tempo para a execução de determinadas tarefas proporcionando uma maior satisfação por parte de todos. Existem algumas soluções automatizadas para o gerenciamento do atendimento nos hospitais a nível mundial, que são:

- Filas de espera, por ordem de chegada;
- Listas de espera, por ordem de chegada;
- Senhas numeradas, por ordem de chegada;
- Pulseiras coloridas, cada cor representa o nível de gravidade do paciente;
- Utilização de softwares para gerenciamento hospitalar.

2.6.2.1 Soluções Baseadas em software

A cada dia que passa vão surgindo novas tecnologias para auxiliar e facilitar algumas tarefas humanas. Nos dias atuais verifica-se muito a existência de softwares na área da saúde para fazer o gerenciamento de suas tarefas.

Segundo Mazzola (2010, p. 13) um software “é um conjunto de instruções que, quando executadas, produzem a função e o desempenho desejado, permitindo que as informações relativas ao problema a resolver sejam manipuladas adequadamente”. As soluções baseadas em softwares permitem exercer algumas funções básicas tanto quanto complexas como: o gerenciamento de dados, a marcação de consultas, geração de relatórios etc, aumentando a produtividade do hospital e proporcionando algumas vantagens:

- Melhoria no atendimento dos pacientes;
- Economização de tempo;
- Diminuição de custos.

Existem várias soluções baseadas em softwares que são:

- Solução Desktop: sistemas desenvolvidos para serem instalados e utilizados em computadores.
- Solução Mobile: sistemas desenvolvidos para serem instalados em um dispositivo eletrônico móvel, como tablets e smartphones.
- Solução web: sistemas desenvolvidos para serem utilizados através de um navegador por diferentes dispositivos, como computadores, tablets e smartphones.

Para o ambiente hospitalar podem ser criados vários sistemas para actuar em diferentes áreas da medicina com objetivo de solucionar diversos problemas. Estas soluções podem ser:

- **Sistema de gestão hospitalar**

Segundo Souza , Berndt ,Medeiros..(2012, p.21) a utilização de Sistemas de Informação em hospitais evoluiu significativamente, partindo de uma realidade em que os computadores eram empregados somente para operar tarefas simples e isoladas, até a integração global das informações, por intermédio de um sistema único. Um Sistema de Informação hospitalar é definido por como um sistema computacional que, instalado em um ambiente hospitalar, possue várias funções como: registrar informações sobre os pacientes de tal forma que possam ser compartilhadas por todos os setores do hospital, marcar consultas, gerar relatórios médicos, etc. Existem várias soluções informáticas de gestão hospitalar que são: Softclinic, Wingest, IELUKA, etc.

- **Sistemas de monitorização do paciente**

É definido como um sistema computacional, que serve para controlar o estado dos pacientes. Pode ser utilizadado para obtenção de alguns dados importantes (peso, temperatura, pressão,etc) que ajudam a equipe médica na detenção de problemas dos pacientes e na resolução dos mesmos.

- **Sistemas de diagnósticos de doenças por imagens**

É definido como um sistema computacional que tem o propósito de extrair informações úteis a partir de uma imagem. Na área da saúde em particular, existem vários sistemas de diagnóstico assistido por computador que procuram detectar anormalidades em exames de imagens de forma automática ou semiautomática para auxiliar o médico a tomar a decisão correta no diagnóstico de uma doença de forma certa e em tempo não excessivo. Um sistema criado para realizar o processamento de imagens digitais geralmente é composto por cinco elementos: a aquisição da imagem, o armazenamento, o processamento em si, a comunicação e a exibição do resultado final do processo.

Para o presente trabalho aplicar-se-á a solução web para a criação de um protótipo de um sistemas especialistas na área da triagem médica.

2.7 Inteligência Artificial

A muito tempo a inteligência dos seres vivos desperta curiosidade no ser humano. Um dos interesses do homem foi a criação da inteligência de forma artificial.

De acordo com STUART et al (1995), citado por Paiva (2015):

Começou-se a usar o termo Inteligência Artificial no ano de 1956, porém já existiam pesquisas voltadas para esta área. Com a criação do computador próximo ao ano de 1950, possibilitou-se um maior avanço nesta área devido a sua capacidade de processamento.

O computador forneceu um veiculo para a criação da Inteligência Artificial onde este fornecia formas de testar as teorias e ver se estas eram válidas. O estudo da inteligência não se limita a problemas relacionados com lógica, suas ideias podem ser usadas na resolução de equações matemáticas, resolução de jogos, reconhecimento de imagem e som, operações financeiras e diagnóstico de doenças.

Existem diversas formas de se definir o que é Inteligência Artificial, onde essas ideias se dividem em 4 grupos:

- Sistemas que pensam como os humanos;
- Sistemas que agem como os humanos;
- Sistemas que pensam racionalmente;
- Sistemas que agem racionalmente.

Alan Turing propôs uma forma de declarar se algo tem inteligência artificial ou não por meio de um teste denominado teste de Turing. O objectivo do teste baseia-se em colocar um ser humano (sem saber a natureza do outro participante) em conversa com a máquina a ser testada em um ambiente remoto, caso o humano não consiga distinguir se o participante na conversa é humano ou uma máquina, então considera-se que este objecto tem inteligência artificial.

A inteligência artificial é aplicada em diversas áreas que são:

- Robótica;
- Processamento de linguagem natural;
- Sistemas Especialistas etc.

Para a realização deste trabalho escolheu-se a área dos sistemas especialistas na qual será abordada com mais detalhes no próximo ponto.

2.7.1 Sistemas Especialistas

De acordo com MANCHINI (2003), citado por Lourenço (2013):

Sistema especialista é uma aplicação da Inteligência Artificial que é capaz de adquirir e disponibilizar o conhecimento operacional de um especialista, solucionando problemas que são resolvíveis apenas por pessoas especialistas que, durante anos, acumularam conhecimento exigido na resolução destes problemas.

Um SE é projetado e desenvolvido para atender a uma aplicação específica e limitada do conhecimento humano. Apoiado no conhecimento justificado da base de informações é capaz de emitir uma decisão. Além disso, deve ser capaz de aprender novos conhecimentos, melhorando a qualidade das decisões.

De acordo com MANCHINI (2003), citado por Lourenço (2013):

Os especialistas humanos têm altos salários. Eles podem reorganizar as informações de que dispõem e usá-las para sintetizar novos conhecimentos,

manusear eventos inesperados usando imaginação ou novas abordagens, mas são imprevisíveis, podem sofrer alterações no humor, e isto pode afetar no diagnóstico.

O especialista humano possui o conhecimento advindo do senso comum, ou seja, se ele for questionado sobre informações que não disponha ou não existam, na mesma hora ele responde que não sabe ou que tal coisa não existe.

O SE têm o custo nominal de o computador rodar o programa, seu trabalho é uma rotina, sem criatividade nem inspiração, porém, são consistentes e inflexíveis em suas decisões; possuem um conhecimento técnico, ou seja, se perguntar a um sistema especialista sobre alguma coisa que ele não tem conhecimento ou que não existe ele não detecta esta situação e inicia uma pesquisa nos seus fatos e regras para encontrar a solução e, quando esta não for encontrada, pode julgar que seu conhecimento está incompleto e solicita informação adicional para completar sua base de conhecimento.

É por isso que os Sistemas Especialistas são mais utilizados como consultores pelo especialista humano ou iniciante na solução de algum problema. Um Sistema Especialista é capaz de:

- 1- Resolver problemas complexos tão bem quanto especialistas humanos;
- 2- Raciocinar heuristicamente utilizando, para isto, as regras práticas contidas na base de conhecimento;
- 3- Interagir com usuários humanos utilizando até linguagem natural;
- 4- Funcionar com dados errados e regras incertas de julgamento;
- 5- Escolher hipóteses múltiplas ao mesmo tempo;
- 6- Explicar porque estão fazendo determinada pergunta;
- 7- Justificar suas conclusões.

Neste trabalho será implementado um protótipo de sistemas especialistas na área da medicina (Triagem) para diagnóstico de possíveis ocorrências de paludismo ou febre tifoide.

2.7.1.1 Arquitetura de um Sistema Especialista

Como um Sistema Especialista toma suas decisões baseado nas informações existentes na base de conhecimento, o sucesso dele depende da forma como o conhecimento é representado e dos mecanismos usados para a exploração deste conhecimento.

A estrutura de um Sistema Especialista é composta por 3 elementos principais que são:

- **A Base de Conhecimento**

De acordo com KELLER (1991), citado por Lourenço (2013):

É um depósito de informações que mudará toda vez que houver mudanças nas regras do negócio, e este processo de mudança precisa ser administrado. É na base de conhecimento que ficam armazenadas as informações especialistas necessárias para resolver problemas de um domínio específico.

De acordo com CHAIBEN (2003), citado por Lourenço (2013):

Essas informações consistem de fatos e heurísticas , onde os fatos são as informações que estarão sempre disponíveis para o sistema e deverão ser atualizadas pelo especialista humano de acordo com a necessidade e as heurísticas são as regras práticas que caracterizam o nível de tomada de decisão do especialista em um domínio.

- **Motor de Inferência**

É considerado o núcleo de um Sistema Especialista, pois é através dele que os fatos e heurísticas contidos na base de conhecimento são aplicados no processo de solução do problema.

O motor de inferência, de certo modo, tenta imitar o pensamento que os especialistas humanos empregam quando resolvem um problema, ou seja, ele pode começar com uma conclusão e procurar uma evidência que a comprove, ou pode iniciar com uma evidência para chegar a uma conclusão.

Esse tipo de pensamento, nos Sistemas Especialistas, está dentro da classe de raciocínio via Regras de Encadeamento e são chamados de **backward chaining** e **forward chaining** respectivamente.

Forward chaining: também chamado de encadeamento para frente, ou também raciocínio progressivo, é utilizado para se chegar a uma conclusão, as informações são fornecidas ao sistema pelo usuário, e o motor de inferência busca na base de conhecimentos fatos e heurísticas

que melhor se aplicam a cada situação. Essa interação com o usuário continua até que a solução do problema seja encontrada.

Backward chaining, também chamado de encadeamento para trás ou raciocínio regressivo, é utilizado para encontrar evidências, o trabalho do motor de inferência ocorre de forma inversa, ou seja, o sistema já parte de uma conclusão, podendo esta ser sugerida pelo próprio usuário, e inicia uma pesquisa nas informações da base de conhecimentos com o objetivo de saber se a conclusão é a mais adequada solução do problema proposto ao sistema.

- **Interface com o Usuário**

É a parte do sistema especialista que interage directamente com o usuário, onde são feitas as entradas e saídas de qualquer problema ou resolução.

De acordo com MENDES (2003), citado por Lourenço (2013):

Dos três elementos fundamentais na estrutura de um Sistema Especialista este, geralmente, é o que necessita de mais tempo para projeção e implementação, para que a comunicação entre o Sistema Especialista e o usuário seja fácil.

A Figura 1 ilustra como está estruturada um sistema especialista.

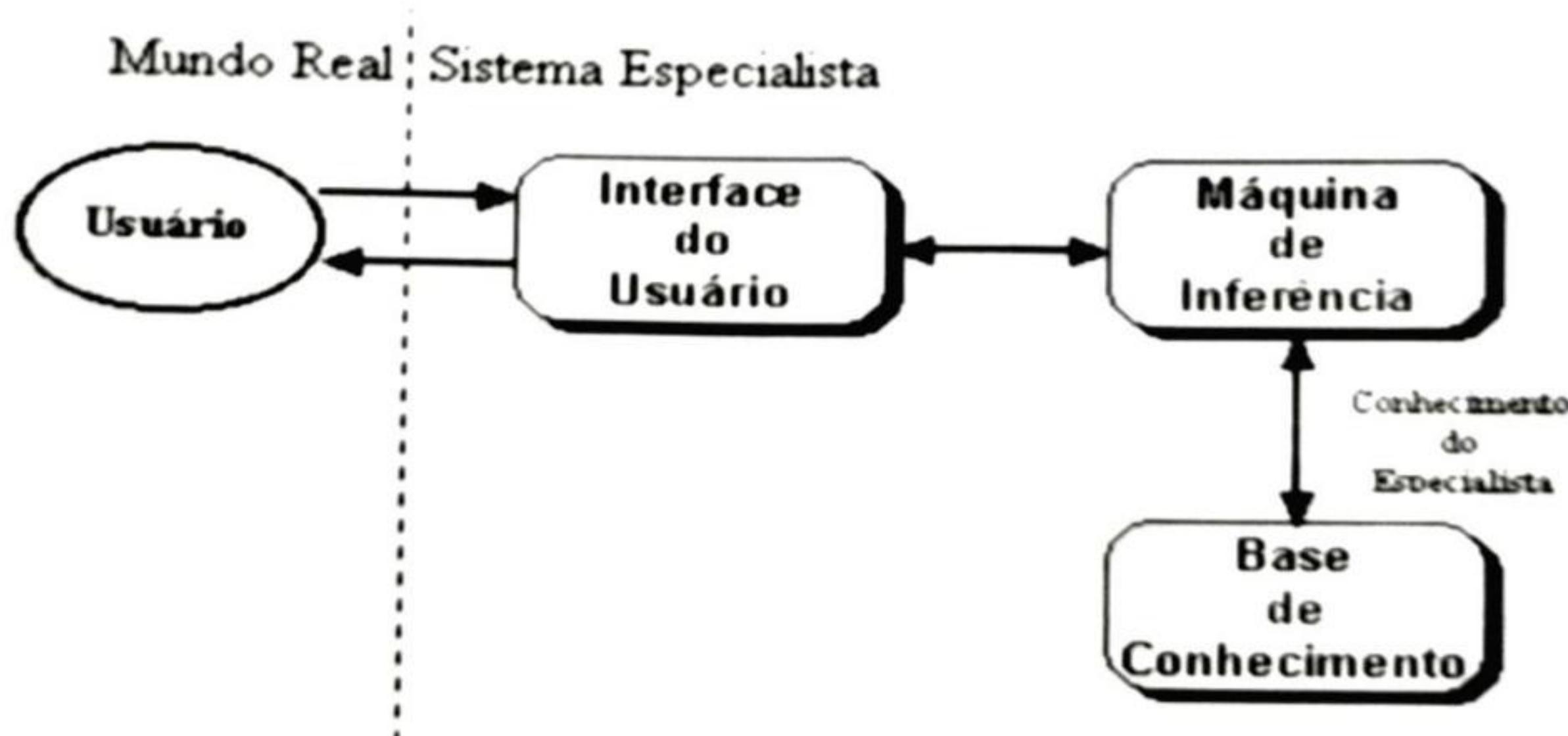


Figura 1: Estrutura de um Sistema Especialista.

Fonte:(Janddy Oliveira, 2014).

Através da interação da base de conhecimento com o motor de inferência e com o usuário, o sistema especialista identifica o problema a ser resolvido e as possíveis soluções.

Na interação da base de conhecimento com o usuário, o Sistema Especialista ainda obtém as informações necessárias para a solução do problema, sendo que a cada nova informação que o usuário fornece ao sistema o espaço entre o problema e a solução diminui. É possível também identificar o raciocínio que levou à conclusão do problema submetido ao sistema.

Para o presente trabalho será construída uma base de conhecimento aplicando todos os conhecimentos do especialista em triagem (Enfermeiro) relacionados as doenças paludismo e febre tifoide. Será construído um motor de inferência com base ao raciocínio **forward chaining**.

2.7.1.2 Métodos de alimentação da base de conhecimento

O aprendizado de um especialista vem do processo de experiência, e de seus resultados experimentais. É portanto de fundamental importância que os sistemas especialistas sejam capazes de aprender e fazer crescer o seu conhecimento a partir de um conhecimento básico sobre o assunto.

O sistema especialista provido de mecanismos de aprendizagem é capaz de analisar e gerar novas regras na base conhecimento ou armazenar informações sobre novos fatos, ampliando a capacidade do sistema em resolver problemas, cada vez em que este for utilizado. Isto é transparente para o usuário, ou seja, o usuário não percebe que todo este processo acontece durante uma sessão de utilização do sistema especialista.

A etapa de alimentação da base de conhecimento pode ser considerada um dos processos de desenvolvimento de um SE mais difícil, pois pode-se encontrar algumas dificuldades durante essa etapa como:

- A dificuldade de transmissão do conhecimento por parte do especialista, quando o conhecimento não é bem definido;
- A dificuldade em expressar o conhecimento em palavras.

O sistema especialista pode ser alimentado por vários métodos que são:

2.7.1.2.1 Método Intermediário

Engenheiro do conhecimento usa avaliação do especialista para alimentar a base como:

- Observação
- Entrevista

2.7.1.2.2 Método Semi-Intermediário

Engenheiro do conhecimento age sem auxílio do especialista utilizando ferramentas que o auxiliam no diagnóstico da nova informação mas correndo alguns riscos por conta da falta de senso humano especialista.

2.7.1.2.3 Método Semidireito

Sem interferência do engenheiro do conhecimento, o especialista interage direto com a ferramenta, necessitando de treinamento e a interface deve ser mais próxima ao entendimento do especialista.

2.7.1.2.4 Método Directo Supervisionado

Sem engenheiro do conhecimento e sem especialista no assunto, usando ferramentas heurísticas tais como:

- Lógica fuzzy
- Redes Neurais

A figura 2 demonstra o processo de aquisição de conhecimento de um sistema especialista.

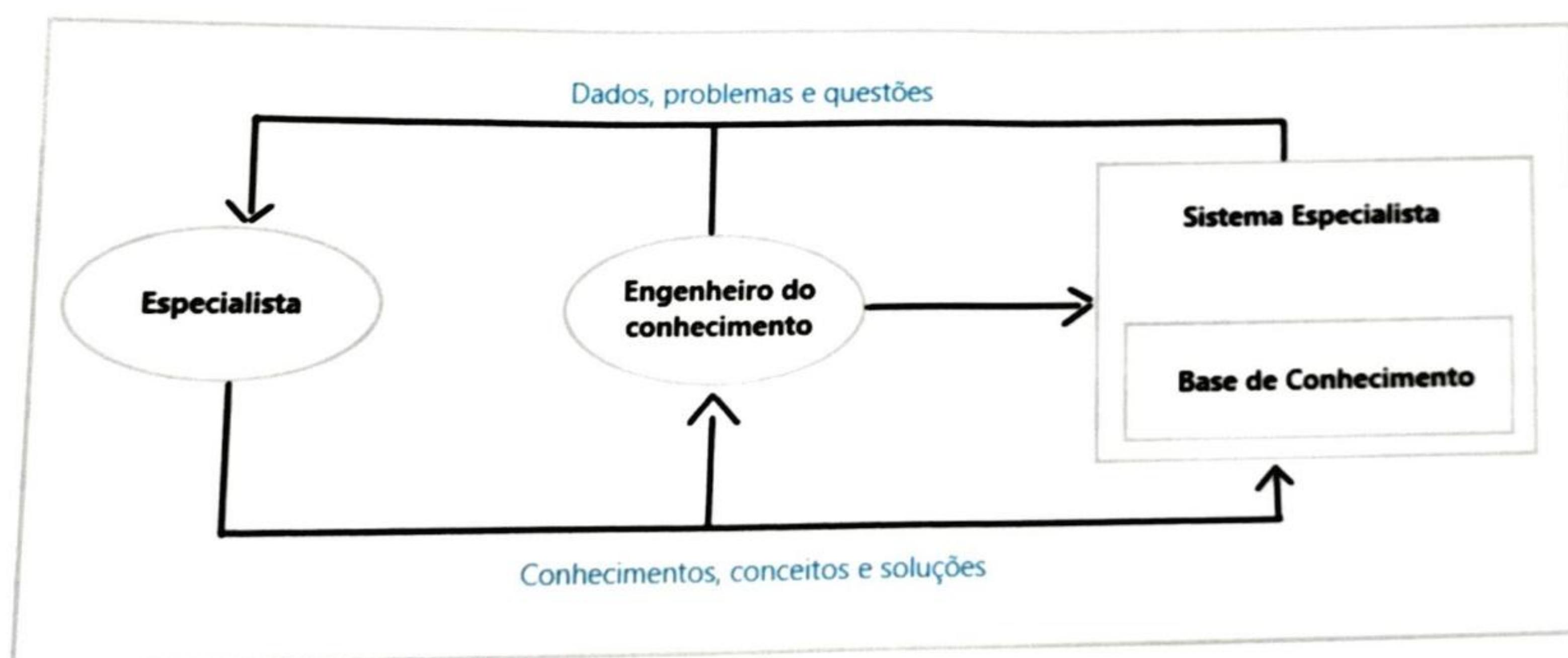


Figura 2: Processo de Aquisição do Conhecimento

Fonte: Autora.

A base de conhecimento do protótipo de sistemas especialista do presente trabalho será criada por um engenheiro do conhecimento que recolherá informações de um enfermeiro especialista em triagem. Para a realimentação da base de conhecimento será utilizado o método semidireto.

2.7.1.3 Sistemas Convencionais X Sistemas Especialistas

De acordo com MANCHINI & PAPPA (2003), citado por Lourenço (2013):

Enquanto um sistema convencional é baseado em um algoritmo que, passo a passo, conduz a uma resposta, os sistemas especialistas são baseados na busca heurística e lidam com problemas que não têm uma solução convencional algoritmizada, ou esta possui complexidade elevada para ser executada. Um processo heurístico conduz a soluções de maneira rápida, mas pode não chegar a solução nenhuma, ou também pode errar, como os especialistas humanos. Um Sistema Especialista processa conhecimentos e não dados. Esses conhecimentos ficam armazenados na base de conhecimentos e algumas vezes é incerto ou incompleto.

2.7.1.4 Vantagens dos Sistemas Especialistas

Os Sistemas Especialistas, geralmente, são mais rápidos na determinação dos problemas, porque sua decisão está fundamentada em uma base de conhecimentos, portanto, se os fatos e heurísticas contidas na base de conhecimentos estiverem corretos a segurança do sistema especialista é total, considerando que as informações passadas ao programa pelo usuário também estarão totalmente corretas. É necessário um número pequeno de pessoas para interagir com um Sistema Especialista.

Como já foi citado, um Sistema Especialista é estável e flexível e suas decisões sempre estão de acordo com as informações passadas pelo usuário.

A dependência de pessoas específicas é decrescente, ou seja, à medida que seus conhecimentos aumentam diminui a dependência do sistema especialista em relação ao especialista humano.

2.6.1.4 Desvantagens dos Sistemas Especialistas

- **Fragilidade:** Não são capazes de resolver problemas mais genéricos porque o conhecimento de que dispõe é bem específico do seu domínio.
- **Falta de Metaconhecimento:** A aquisição do conhecimento ainda é um dos maiores obstáculos a aplicação de tecnologia dos sistemas especialistas a novos domínios, porque eles não possuem conhecimentos sofisticados sobre sua própria operação, portanto não conseguem raciocinar sobre seu próprio escopo e restrições .
- **Validação:** Por não saber quantificar o uso de conhecimento de um sistema especialista é que se torna difícil medir o desempenho do mesmo.

2.8 Sistemas Relacionados ao Objecto de Estudo

- **MYCIN**

Foi um dos primeiros sistema especialistas, desenvolvido no início do ano de 1970 na Universidade de Stanford. Este sistema foi desenvolvido para identificar as bactérias causadoras de infecções graves, tais como bacteremia e meningite, e para recomendar antibióticos, com a dose ajustada para o peso corporal do paciente. O sistema Mycin também foi utilizado para o diagnóstico de doenças da coagulação sanguínea.

Operava usando um motor de inferência bastante simples, e uma base de conhecimento de aproximadamente 600 regras. Seria uma consulta ao médico, executando o programa através de uma longa série de simples questões do tipo sim/não ou questões textuais. No final, era fornecida uma lista de possíveis bactérias culpadas classificadas em ordem de alta a baixa probabilidade de cada diagnóstico e o grau de confiança em cada diagnóstico.

O **MYCIN** nunca foi realmente usado na prática porque exigia que o usuário digitasse todas as informações relevantes sobre o paciente sempre que era feito uma consulta consequentemente uma sessão com o sistema poderia consumir 30 minutos ou mais, um compromisso de tempo irreal para um médico ocupado.

- **INTERNIST-I**

Foi uma ferramenta de diagnóstico assistida por computador de base ampla desenvolvida no início dos anos 1970 na Universidade de Pittsburgh como um experimento educacional. Foi utilizada para diagnóstico de possíveis doenças relacionadas a medicina interna. Médicos e paramédicos querendo usar o INTERNIST-I acharam o período de treinamento longo e a interface pesada. Uma consulta média com o INTERNIST-I exigiu cerca de noventa minutos, tempo demais para a maioria das clínicas.

- **CADUCEUS**

Foi um sistema especialista médico para diagnóstico de possíveis doenças relacionadas a medicina interna. Foi terminado em meados dos anos 1980 por Harry Pople (da Universidade de Pittsburgh). Sua motivação foi a intenção de melhorar o **MYCIN** “que focava sobre bactérias infecciosas do sangue” concentrando-se em questões mais abrangentes do que um campo estreito como o do envenenamento do sangue (embora fosse fazê-lo de forma similar). **CADUCEUS**, eventualmente, poderia diagnosticar até 1000 doenças diferentes.

2.9 Paradigma de Desenvolvimento de Software

O paradigma de uma linguagem de programação é a sua identidade. Corresponde a um conjunto de características que, juntas, definem como ela opera e resolve os problemas. Algumas linguagens, inclusive, possuem mais de um paradigma e são chamadas multi paradigmas. Os principais paradigmas são: Imperativo, Funcional, Lógico, Orientado a objectos e estruturado.

2.9.1 Paradigma Orientado a Objectos

Este paradigma é o que mais se reflete os problemas atuais. Linguagens orientada a objectos (OO) são projetadas para implementar diretamente a abordagem OO na solução dos problemas.

Tal abordagem tornou-se uma ferramenta importante para solucionar muitos tipos de problemas através da simulação. Um programa orientado a objectos consiste em objectos que enviam mensagens uns para os outros. Estes objetos correspondem diretamente aos objetos atuais, tais como pessoas, máquinas, departamentos etc.

O paradigma orientado a objectos é formado por 5 elementos principais que são:

- **Objeto:** consiste em um conjunto de operações encapsuladas (métodos) e um estado (determinado pelo valor dos atributos) ou seja um objecto possui todas as características necessárias relacionadas a ele. Os objectos representam uma coleção de dados relacionados a um tema específico. Ex: pessoa, veículo, um documento etc.
- **Mensagem:** são as requisições enviadas de um objeto para outro, para que o objeto receptor forneça um resultado de acordo com a operação executada.
- **Método:** consiste nas descrições das operações que um objeto executa quando recebe uma mensagem. Há uma correspondência um para um entre mensagens e métodos que são executadas quando uma mensagem é recebida através de um determinado objeto.
- **Atributo:** informação de estado na qual cada um dos objectos de um classe tem o seu próprio valor. Existem 2 tipos de atributos que são: atributos de classes e atributos de objectos.
- **Classe:** define as características de uma coleção de objetos. Consiste em descrições de métodos e atributos que os objetos que pertencem a classe irão possuir.

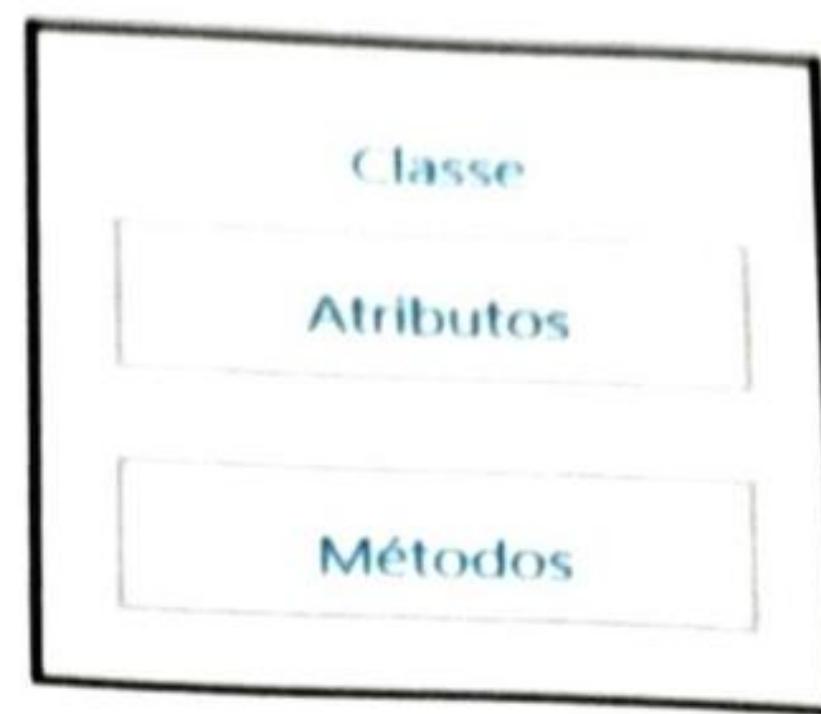


Figura 3: Estrutura da programação baseada a objetos

Fonte: Autora.

2.9.2 Paradigma Funcional

Este paradigma trata a computação como uma avaliação de funções matemáticas. Este método enfatiza a aplicação de funções, as quais são tratadas como valores de primeira importância, ou seja, funções podem ser parâmetros ou valores de entrada para outras funções e podem ser os valores de retorno ou saída de uma função como por ex: LISP, etc.

2.9.3 Paradigma Lógico

O Paradigma Lógico enfatiza a descrição declarativa de um problema, ao invés da decomposição do problema em uma implementação algorítmica. Tais programas são mais próximos de uma especificação do que a tradicional forma de programar. Uma das principais ideias da programação em lógica é de que um algoritmo é constituído por dois elementos disjuntos: a lógica e o controle.

O componente lógico corresponde à definição do que deve ser solucionado, enquanto que o componente de controle estabelece como a solução pode ser obtida. O programador precisa somente descrever o componente lógico de um algoritmo, deixando o controle da execução para ser exercido pelo sistema de programação em lógica utilizado. Em outras palavras, a tarefa do programador passa a ser simplesmente a especificação do problema que deve ser solucionado, razão pela qual as linguagens lógicas podem ser vistas simultaneamente como linguagens para especificação formal e linguagens para a programação de computadores. Ex: QLISP, PROLOG, etc.

2.9.3.1 Prolog

A principal utilização da linguagem Prolog reside no domínio da programação simbólica, não numérica, sendo especialmente adequada à solução de problemas, envolvendo objetos e relações entre objetos.

O advento da linguagem Prolog reforçou a tese de que a lógica é um formalismo conveniente para representar e processar conhecimento. Seu uso evita que o programador descreva os procedimentos necessários para a solução de um problema, permitindo que ele expresse declarativamente apenas a sua estrutura lógica, através de fatos, regras e consultas.

Algumas das principais características da linguagem Prolog são:

- É uma linguagem orientada ao processamento simbólico;
- Representa uma implementação da lógica como linguagem de programação;
- Apresenta uma semântica declarativa inerente à lógica;
- Suporta código recursivo e iterativo para a descrição de processos e problemas, dispensando os mecanismos tradicionais de controle, tais como while, repeat, etc;

Os elementos básicos da linguagem Prolog são fatos, regras e consultas.

2.9.3.1 Fatos

Os fatos servem para estabelecer um relacionamento existente entre objetos de um determinado contexto de discurso. Por exemplo, progenitor(maria, josé) é um fato que estabelece que Maria é mãe de José, ou seja, que a relação progenitor existe entre os objetos denominados maria e josé.

Em Prolog, identificadores de relacionamentos são denominados predicados e identificadores de objetos são denominados átomos. Tanto predicados quanto átomos devem iniciar com letra minúscula. Ex: Uma árvore genealógica:

progenitor(maria, josé).

progenitor(joão, josé).

progenitor(joão, ana).

progenitor(josé, júlia).

progenitor(josé, íris).

progenitor(íris, jorge).

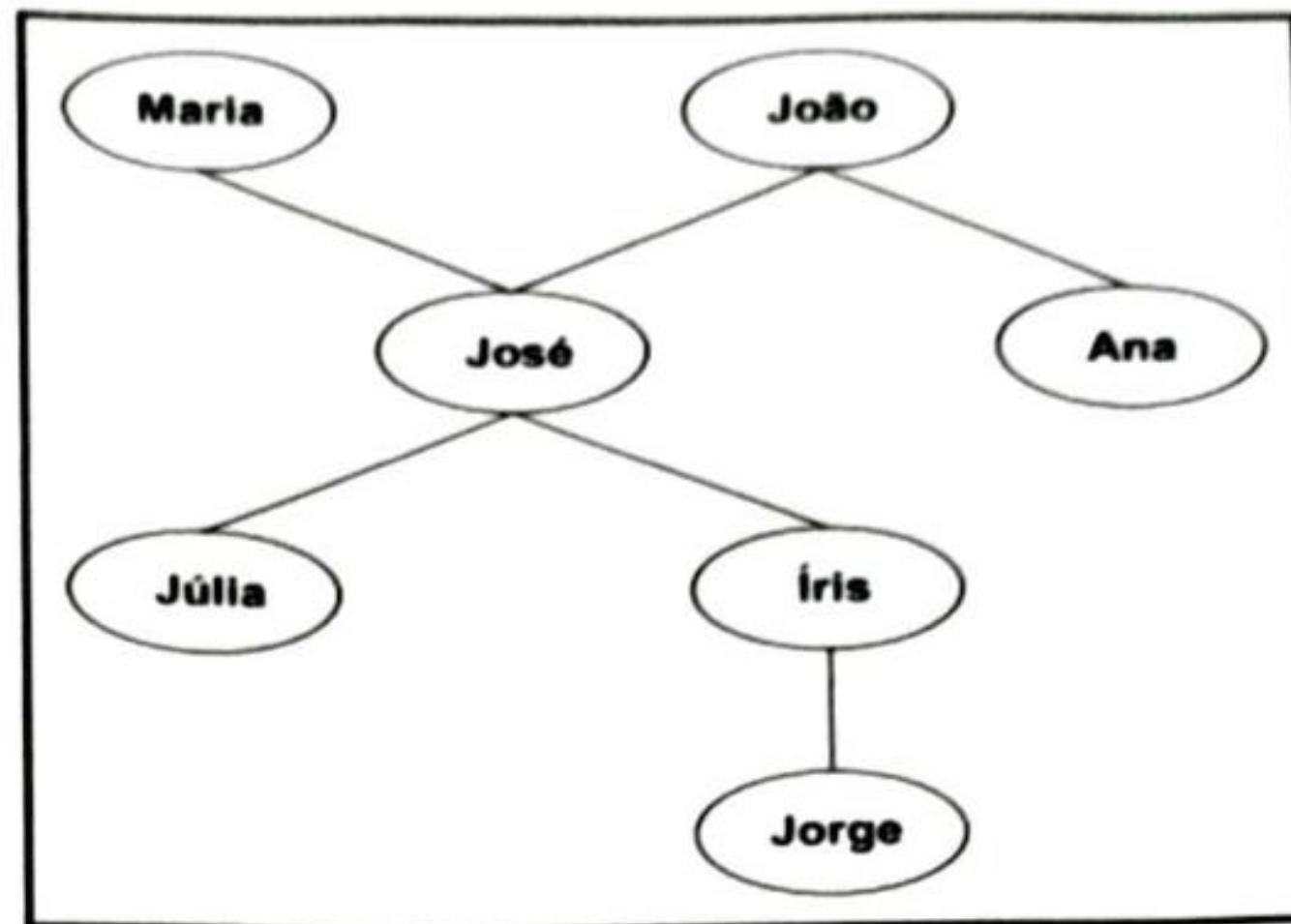


Figura 4: Árvore Genealógica

Fonte:(Luiz A. M. Palazzo, 1997).

2.9.3.1.2 Regras

Regras nos permitem definir novas relações em termos de outras relações já existentes. Por exemplo, a regra $\text{avô}(X, Y) :- \text{progenitor}(X, Z), \text{progenitor}(Z, Y)$. define a relação avô em termos da relação progenitor, ou seja, estabelece que X é avô de Y se X gerou Z e Z gerou Y.

Há uma diferença importante entre regras e fatos. Um fato é sempre verdadeiro, enquanto regras especificam algo que "pode ser verdadeiro se algumas condições forem satisfeitas". As regras têm:

- Uma parte de conclusão (o lado esquerdo da cláusula).
- Uma parte de condição (o lado direito da cláusula).

Exemplo: $\text{filho}(Y, X) :- \text{progenitor}(X, Y)$.

O símbolo ":" significa "se" e separa a cláusula de conclusão, ou cabeça da cláusula, da cláusula de condição ou corpo da cláusula. Se a condição expressa pelo corpo da cláusula " $\text{progenitor}(X, Y)$ " é verdadeira então, segue como consequência lógica que a cabeça " $\text{filho}(Y, X)$ ", também é. Por outro lado, se não for possível demonstrar que o corpo da cláusula é verdadeiro, o mesmo irá se aplicar à cabeça.

2.9.3.1.3 Consultas

Para recuperar informações de um programa lógico, usamos consultas. Uma consulta pergunta se uma determinado relacionamento existe entre objetos.

Exemplo:

?- filho(josé,maria).

yes

A consulta acima pergunta se a relação filho vale para os objetos José e Maria ou, em outras palavras, pergunta se José é filho de Maria. Então de acordo com a árvore genealógica da **figura 4**, a resposta a essa consulta será yes(sim). Sintaticamente, fatos e consultas são muito similares. A diferença é que fatos são agrupados no arquivo que constitui o programa, enquanto consultas são sentenças digitadas no prompt do interpretador Prolog. Responder uma consulta com relação a um determinado programa corresponde a determinar se a consulta é consequência lógica desse programa, ou seja, se a consulta pode ser deduzida dos fatos expressos no programa.

Para o desenvolvimento do sistema proposto no presente trabalho utilizar-se-á o paradigma orientado a objecto na construção e manipulação do backend da aplicação e o paradigma lógico para a construção da base de conhecimento.

2.10 Quantificação, Caracterização e Produção de Certezas

A incerteza num domínio de aplicação pode estar presente nos dados de entrada, na solução do problema ou em ambos. Por exemplo, no caso de diagnóstico médico os sinais e sintomas que o médico coleta e trabalha apresentam vários problemas de incerteza, tais como: a inexatidão dos relatos do paciente e a percepção da intensidade de cada sintoma, entre outros. Além disso, o próprio raciocínio clínico não pode ser realizado com certeza, isto é, podem existir dois pacientes com dois conjuntos similares de sinais, sintomas e seus diagnósticos serem diferentes. Torna-se então necessário pesquisar formas de representação de incerteza nos inputs e no raciocínio de sistemas especialistas aplicados a domínios com a presença de incerteza.

As principais formas de tratamento de incerteza são:

- Redes Bayesianas
- Lógica Fuzzy

2.10.1 Redes Bayesianas

As Redes Bayesianas são ferramentas que utilizam o raciocínio probabilístico, ou seja, toda sua metodologia é baseada em probabilidades, especialmente a probabilidade condicional.

Probabilidade Condicional

Trata-se da probabilidade de um evento A acontecer sabendo que um evento F aconteceu.

$$P(A|F) = \frac{P(A \cap F)}{P(F)}$$

Teorema de Bayes

O teorema de Bayes foi desenvolvido pelo Matemático e teólogo Inglês Thomas Bayes, na qual é amplamente usado para calcular a probabilidade de ocorrência que um evento tem, conhecendo um fragmento relacionado de informação.

$$P(B|A) = \frac{P(A|B) * P(B)}{P(A)}$$

P(B): é chamada de probabilidade a priori de B.

P(B|A): é chamada de probabilidade condicionada ou a posteriori de B.

Rede bayesiana é uma ferramenta gráfica para raciocínio e representação de conhecimento frente a incertezas. Ela é uma representação compacta da distribuição de probabilidades conjunta do universo do problema. Têm a finalidade de representar o conhecimento em sistemas inteligentes que utilizam o raciocínio probabilístico e permite calcular eficientemente, probabilidade condicional de qualquer variável. São aplicadas em diversas áreas como: Medicina, Economia, Bioinformática, etc.

Uma rede bayesiana é um grafo orientado acíclico, onde os nós nos grafos representam evidências ou hipóteses e onde um arco que conecte dois nós representa a dependência entre estes nós.

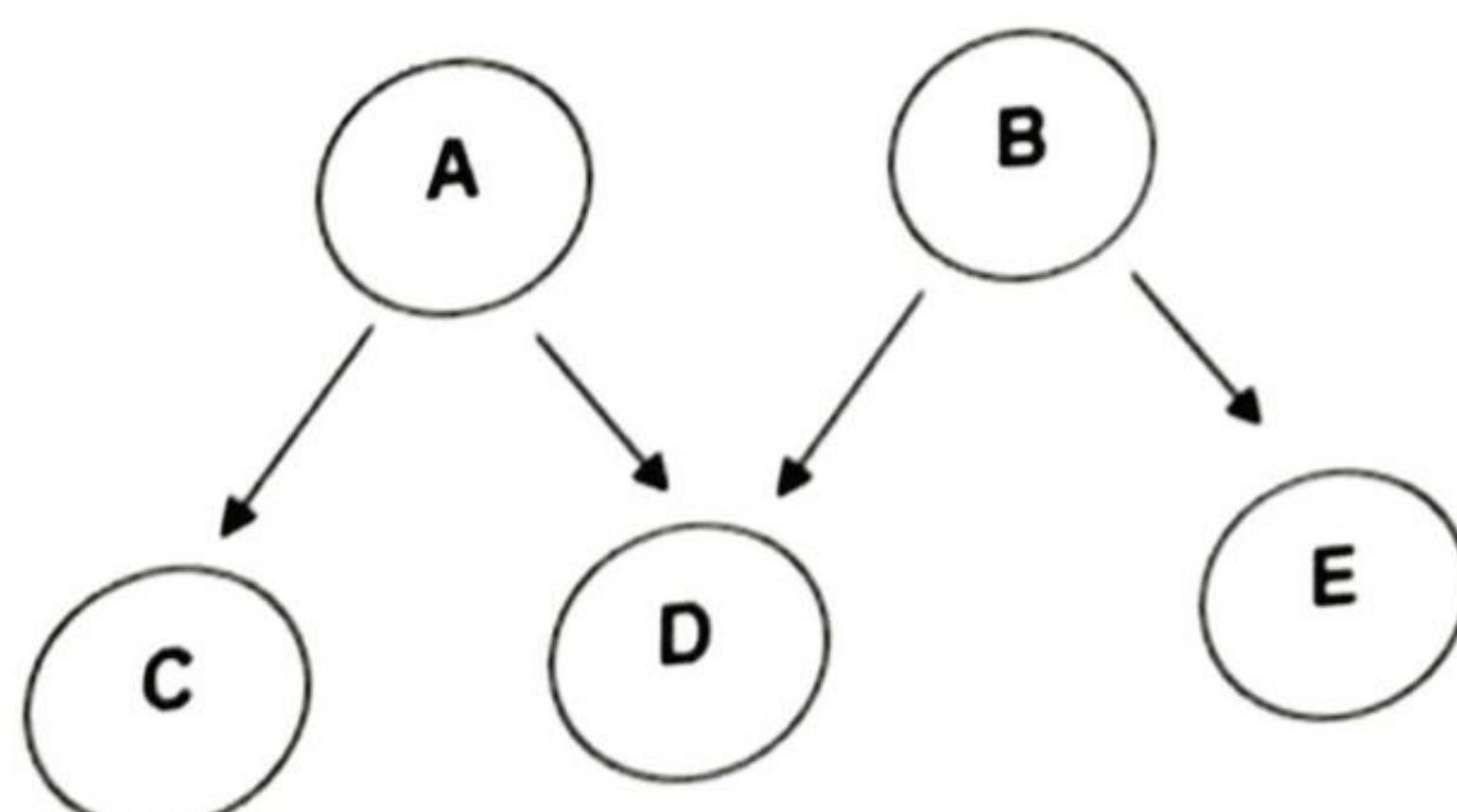


Figura 5: Grafo Orientado Acíclico.

Fonte: Autora.

No grafo da figura 4, as variáveis A e B representam evidências e as variáveis C, D e E representam as hipóteses. C e D são ambos dependentes de A, e D e E são ambos dependentes de B. Dois nós que não tenham um arco entre elas são independentes um do outro, assim sendo A e B são independentes.

2.10.1.1 Vantagens e Desvantagens das Redes Bayesianas

Vantagens

- Redes bayesianas podem ser diretamente geradas a partir de dados coletados de processos do mundo real;
- Raciocínio probabilístico é capaz de tratar o grau de incerteza de domínios bem variados, além de propor representações parcimoniosa, tornando o tratamento das incertezas mais simples por parte das redes bayesianas;

Desvantagens

- Requer conhecimento a priori;
- Número de variáveis do problema (nós) é alto;
- A relação entre as variáveis pode ser muito complexa;

2.10.2 Lógica Fuzzy (Lógica Nebulosa ou Difusa)

A lógica fuzzy tem como objectivo modelar modos de raciocínio aproximados ao invés de precisos. Na lógica binária (clássica) as preposições são unicamente “Verdadeiras” ou “Falsas”. Na lógica difusa as preposições podem ter valores intermediários entre “Verdadeiro” e “Falso”.

Muitas vezes utiliza-se uma discretização dos valores para um domínio (lógica de múltiplos valores). Exemplo: {0, 0.5, 1} para valores que indiquem falso, talvez verdadeiro e verdadeiro. A lógica difusa então visa modelar modos de raciocínio imprecisos, tendo os casos precisos como situações limite.

2.10.2.1 Lógica Clássica Vs Fuzzy

Existem algumas características que representam cada uma das delas:

Clássica

- Valores falsos ou verdadeiros(0 ou 1);
- Exige definição exata;

Fuzzy

- Valores no intervalo de [0..1];
- Não exige definição exata;
- As respostas são relativas, possuem um grau de veracidade que variam entre totalmente falso e totalmente verdadeiro.

No mundo real os problemas muitas vezes não conseguem ser representados pela lógica clássica. Conjuntos convencionais têm apenas os critérios de pertinência “pertence” ou “não pertence”, ou seja, um elemento não pode pertencer parcialmente a um conjunto, da mesma forma que um conjunto não pode estar parcialmente contido em outro.

2.10.2.2 Grau de pertinência

Cada elemento do conjunto difuso tem um grau de pertinência no intervalo [0..1], dessa forma permite uma transição gradual da falsidade para a verdade. O grau de pertinência permite representar valores imprecisos como quente e frio.

2.10.2.1 O universo

O universo contém os elementos que podem ser considerados no conjunto. Seu objectivo é não permitir o uso de dados incorrectos ou incoerentes. Exemplo: o universo de um conjunto que mede o sabor poderia ser {doce, amargo, etc}.

Representação

Um conjunto fuzzy A é uma coleção de pares: $A = \{(x, \mu(x))\}$, onde :

- $\mu(x)$ é o grau de pertinência do elemento x.

2.10.2.2 Operações da lógica Fuzzy

A lógica fuzzy possui várias operações, que são utilizadas no processo de inferência, nas quais são:

- Intersecção: $u(A \cap B) = \min(u(A), u(B))$;
- União: $u(A \cup B) = \max(u(A), u(B))$;
- Complemento: $u(A') = 1 - u(A)$.

Exemplo Problema: Uma família com quatro integrantes deseja comprar uma casa.

Critérios:

- A casa tem que ser confortável (conforto refere-se ao número de quartos).

- A casa tem que ser grande.

Resolução "Inferência"

O conjunto de casas descritas pelo número de quartos pode ser descrito como:

- $u = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$.

O conjunto fuzzy e que caracteriza o conforto da casa pode ser descrito como:

- $c = [0.2 \ 0.5 \ 0.8 \ 1 \ 0.7 \ 0.3 \ 0 \ 0 \ 0 \ 0]$.

O conjunto fuzzy que caracteriza a noção de grande pode ser descrito como.

- $i = [0 \ 0 \ 0.2 \ 0.4 \ 0.6 \ 0.8 \ 1 \ 1 \ 1 \ 1]$.

Aplicando a intersecção fica:

$c \cap i = [0 \ 0 \ 0.2 \ 0.4 \ 0.6 \ 0.3 \ 0 \ 0 \ 0 \ 0]$. Conclui-se que uma casa com 5 quartos é mais satisfatória tendo o grau de pertinência de 0.6, a segunda melhor opção seria uma casa de 4 quartos com um grau de 0.4.

Para a construção do protótipo de sistema especialista proposto no presente trabalho, aplicar-se-á a lógica fase no processo de inferência do sistema, nos casos em que terá duas saídas para uma única entrada, ou seja, quando o sistema associar um conjunto de sintomas a paludismo e ao mesmo tempo a febre tifoide. Será aplicada as operações adequadas da lógica fuzzy citadas acima para auxiliar o sistema a tomar uma decisão.

2.10.2.3 Vantagens da Lógica Fuzzy

A lógica fuzzy possui várias vantagens como:

- Melhor tratamento das imprecisões;
- Simplifica a solução de problemas;
- Proporciona um rápido protótipo dos sistemas;
- Simplifica a aquisição da base do conhecimento.

2.11 Hospital Municipal da Samba

O hospital municipal da samba está localizado na Av. Pedro de Castro Van-Dúnem Loy em Luanda, foi inaugurado no dia 24 de maio de 2011 com objectivo de proporcionar serviços sanitários a população. O seu horário de atendimento é de segunda-feira à segunda-feira, aberto 24 horas ao dia.

O processo de atendimento do paciente é feito da seguinte forma:

- O paciente preenche uma lista por ordem de chegada, para ser analizado pelos enfermeiros(Triagem Inicial);
- Obedecendo a lista de espera, o paciente é chamado para a sala de triagem onde será recolhido algumas informações sobre os seus dados pessoais e sintomas, analizando assim o seu estado de saúde;
- De acordo com a analise feita, o paciente estará sujeito a : receber uma receita caso o seu estado não seja grave, marcar uma consulta caso tenha que ser acompanhado por um médico o quanto antes e ser transferido no banco de urgência caso o estado do paciente seja muito grave.

A marcação de consultas é feita tendo acesso a uma agenda, onde se verifica a disponibilidade de cada um dos médicos. É preenchido um formulário com as seguintes informações: nome do paciente, data e hora da consulta, o nome do médico e o tipo de consulta.

Verifica-se uma certa deficiência no atendimento dos seus pacientes, pois o processo não está automatizado com soluções eficazes(todo o processo de recolha de dados é feito com base a papeladas) provocando a insatisfação por parte dos seus pacientes.

CAPÍTULO III: METODOLOGIAS

3.1 Metodologia Científica

Para a recolha de dados do presente trabalho utilizou-se alguns métodos e técnicas que são:

➤ Método de abordagem

- **Método Indutivo:** por ser um método que parte de algo específico para o geral. De acordo com a natureza do projecto aplicou-se esse método tendo como o particular o levantamento de dados no hospital municipal da samba para detenção de paludismo e febre tifoide.

➤ Método de procedimento

- **Modelagem:** permitiu a criação de modelos para a representação do sistema bem como para proporcionar o fácil entendimento do funcionamento e interação entre os processos do sistema.
- **Método Comparativo:** permitiu confrontar elementos apartir dos seus atributos de modo a verificar diferenças ou semelhanças. Aplicou-se este método para comparar o sistema proposto com outras soluções existentes.

➤ Técnicas de colecta de dados

- **Pesquisa Documental:** permitiu uma pesquisa na qual a fonte de colecta de dados está relacionada a documentos escritos ou não, ou seja fontes primárias. Ex: Arquivos públicos, particulares, fontes estatísticas.
- **A pesquisa bibliográfica:** permitiu uma pesquisa que abrange toda bibliografia publicada em relação ao tema de estudo, desde publicações, jornais, revistas, livros, pesquisas, monografias, teses, material cartográfico.
- **Entrevista:** permitiu o encontro com o especialista em triagem (enfermeiro), a fim de obter informações a respeito dos procedimentos feitos na realização da triagem e dos sintomas relacionados a paludismo e febre tifoide.

➤ Tipo de pesquisa

- **Qualitativa:** permitiu a realização de uma entrevista com o cliente para avaliação da viabilidade do protótipo do sistema proposto.

3.2 Metodologia de Desenvolvimento de Software

Qualidade hoje em dia, não é apenas um diferencial de mercado para a empresa conseguir vender e lucrar mais, é um pré-requisito que a empresa deve conquistar para conseguir colocar o produto no mercado global.

A qualidade de software não pode ser avaliada isoladamente. No desenvolvimento de software, um método pobre ou a ausência de uma metodologia pode ser a causa da baixa qualidade. A avaliação da qualidade está diretamente relacionada com a qualidade de processos e metodologias utilizadas no desenvolvimento do software.

3.2.1 Metodologia Aplicada no Projeto

Para o desenvolvimento do presente projecto aplicou-se alguns conceitos da metodologia RUP para a obtenção dos artefatos em todas as fases de desenvolvimento. O ciclo de desenvolvimento no RUP possui quatro fases: iniciação, elaboração, construção e transição.

Na fase de **iniciação** fez-se um planeamento de todo o projecto, avaliou-se os possíveis riscos, estabeleceu-se as prioridades e levantou-se requisitos para a realização do mesmo (entrevistas com o cliente).

Na fase de **elaboração** fez-se uma modelagem para analizar de forma mais detalhada a solução proposta.

Na fase de **construção** construiu-se o software, com foco no desenvolvimento de componentes e outros recursos do sistema de acordo as objectivos estabelecidos na fase inicial.

Na fase de **transição** realizou-se muitos testes para medir a funcionalidade do sistema e elaborou-se o plano de implantação.

CAPÍTULO IV: APRESENTAÇÃO DA PROPOSTA

4.1 Iniciação

4.1.1 Modelação de Processos de Negócio

Todo paciente que vai para ser tratado no hospital municipal da samba tem que passar por um processo de triagem inicial. Ao chegar no hospital o paciente terá que assinar uma lista de presença e aguardar pela chamada no local de espera, onde faz-se a separação por categorias (pediatria, gestantes etc..).

O enfermeiro faz a chamada por ordem de chegada (os casos graves serão priorizados) e indicará o paciente para a sala de triagem, onde o enfermeiro verificará o seu estado de saúde e o encaminhará para uma marcação de consulta (para os casos sem emergência).

A marcação é feita manualmente e o processo é bastante demorado gerando vários problemas como: a desorganização das informações, a cunha para a marcação de consultas priorizando os pedidos de outros e deixando de lado a população e como resultado alguns dos pacientes voltam para as suas casas sem marcar a consulta.

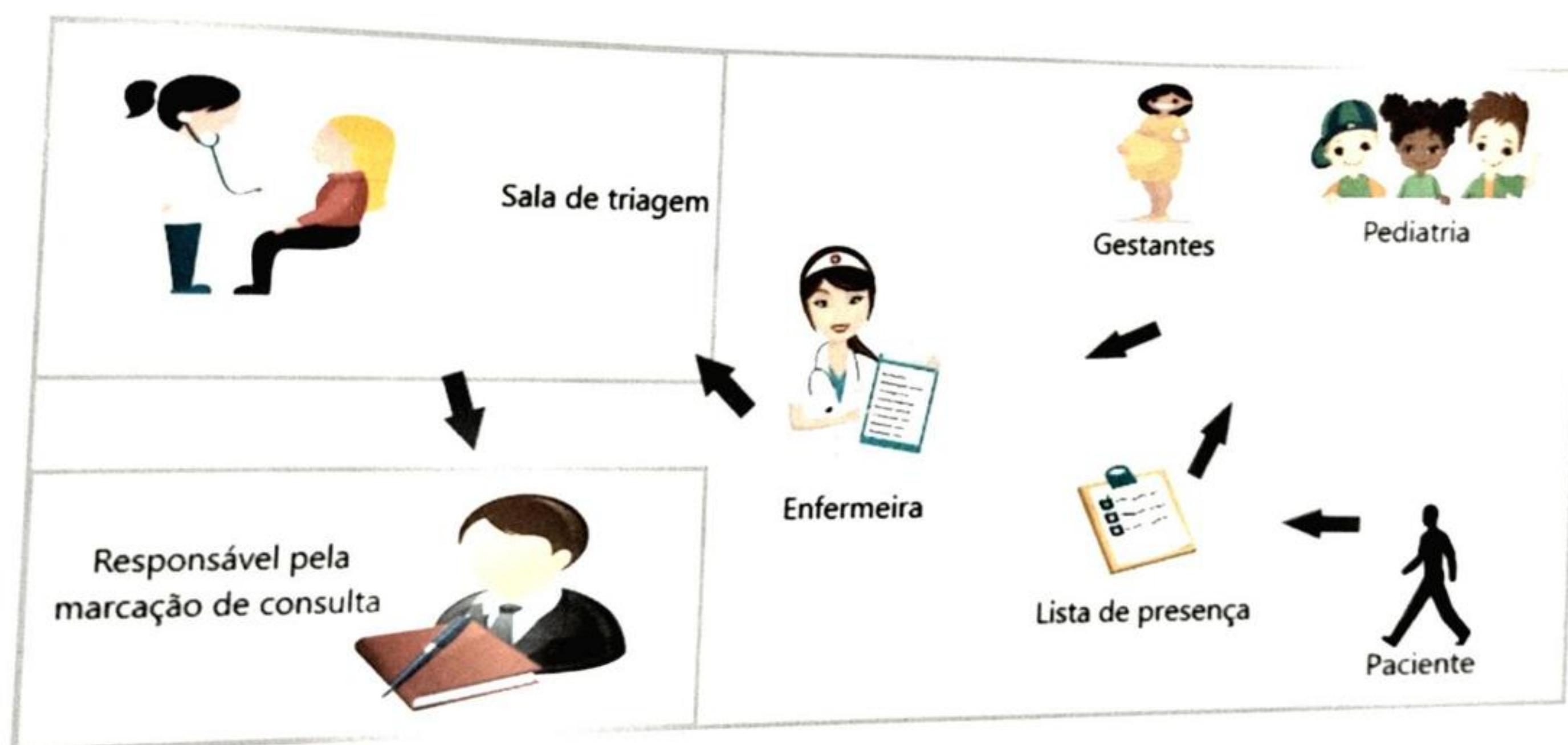


Figura 6: Processo de atendimento do hospital

Fonte: Autora.

4.1.2 Entrevista

As entrevistas foram realizadas de forma informal, tendo como foco principal a obtenção de informações relevantes sobre o processo de triagem e marcação de consultas no hospital, tendo como entrevistados: uma enfermeira e o responsável pela marcação de consultas (ver apêndice 20).

4.1.3 Casos de Uso

Os casos de usos são utilizados para especificação de requisitos funcionais do sistema através de desenhos em UML, ou seja mostram a funcionalidade do sistema com base na visão de seus usuários.

Um diagrama de caso de uso apresenta um conjunto de casos de usos, actores e suas relações. O actor representa uma entidade (ser humano, dispositivo de hardware ou mesmo um outro sistema) que interage com sistema a ser modelado.

4.1.3.1 Casos de Uso de Negócio

Os casos de uso de negócio descrevem os processos de um negócio e suas interações com as partes externas, como clientes e parceiros. Para a ilustração e melhor compreensão do funcionamento dos processos de negócio do hospital, criou-se: um diagrama de caso de uso de negócio especificando assim os actores e suas actividades e uma tabela de descrição dos actores do negócio (ver apêndices 1 e 2).

4.1.4 Especificação de Requisitos

A especificação de requisitos é um processo de estabelecimento dos serviços que o cliente deseja para o software e as restrições sob as quais deve ser desenvolvido, ou seja é um processo de aquisição, refinamento e verificação das necessidades do sistema. A fase de especificação de requisitos é muito importante, pois, pode levar o software ao sucesso ou ao fracasso. Alguns factores que contribuem para o fracasso de um software são:

- Requisitos incompletos;
- Falta de envolvimento por parte do cliente;
- Falta de planeamento;
- Modificações bruscas nos requisitos e nas especificações, etc.

4.1.4.1 Requisitos Funcionais do Sistema

Os requisitos funcionais focam-se na funcionalidade e serviços do sistema ou seja as funções que o sistema deve fornecer ao cliente (ver apêndice 3).

4.1.4.2 Requisitos Não Funcionais do Sistema

Os requisitos não funcionais estão relacionados ao uso da aplicação em termo de desempenho, usabilidade, confiabilidade, segurança, disponibilidade e manutenção (ver apêndice 4).

4.1.5 Arquitetura do Sistema

Para a construção do sistema aplicou-se a arquitetura MVC (Modelo, Visão e Controlador). Na qual a Visão dispara eventos (baseados entrada na do usuário) ao Controlador que modifica o estado do Modelo.

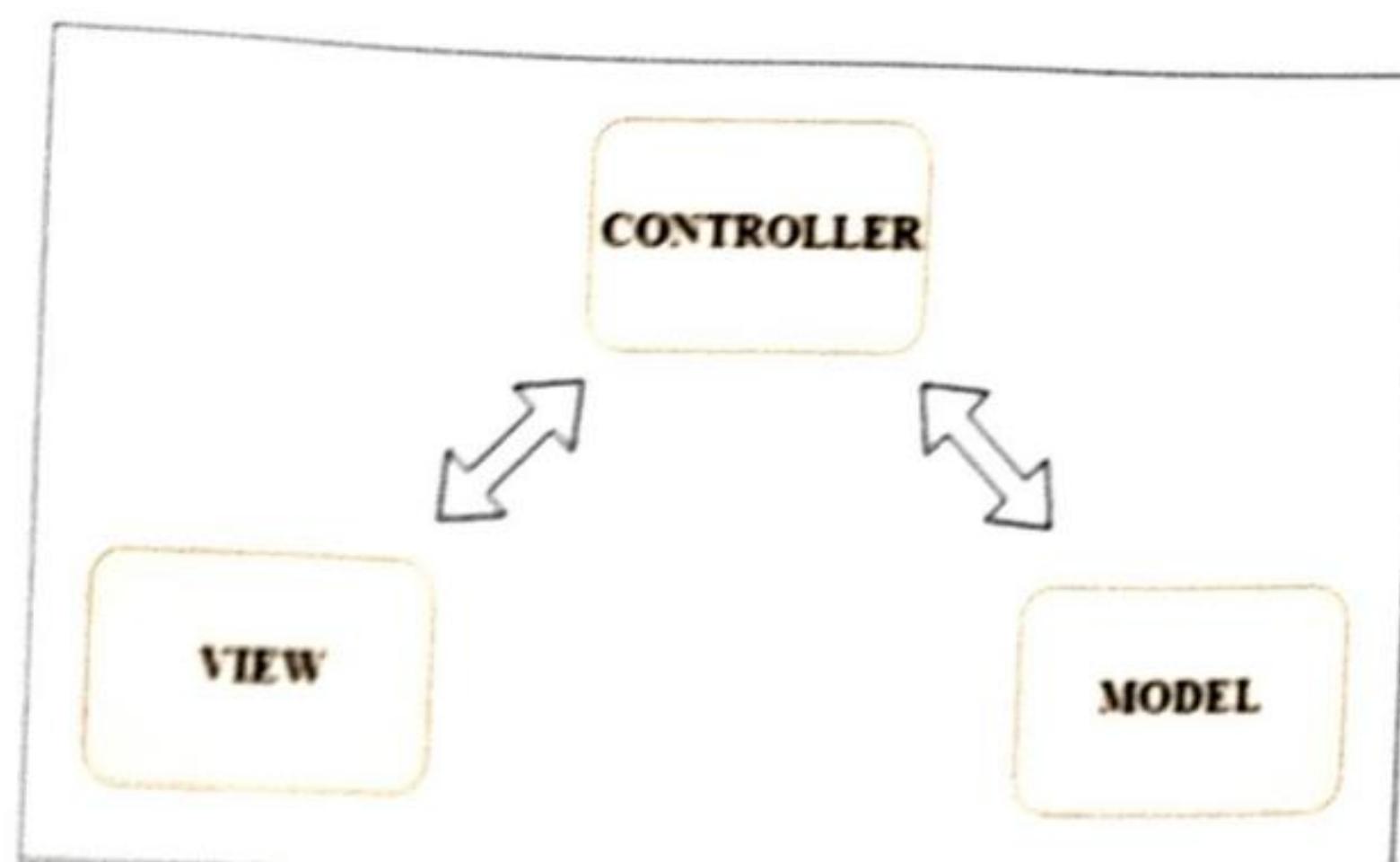


Figura 7: Arquitetura do Sistema

Fonte: Autora.

4.1.6 Ferramentas e Linguagens Utilizadas

Para o desenvolvimento do sistema proposto utilizou-se as seguintes linguagens e ferramentas:

1. Linguagens

- Front-end: HTML5, CSS3, Javascript, Jquery e Ajax.
- Back-end: PHP, seguindo o padrão Model View Controller (MVC).
- BD: SQL.

2. Ferramentas

- MySQL Workbench 6.3
- IntelliJ IDEA 2018 1.4
- Paint 3D
- Visual Paradigm 13.0
- Draw.io

4.2 Elaboração

Depois da fase de iniciação passou-se para a fase de elaboração na qual foi feito a modelagem do sistema proposto para melhor percepção e refinamento dos requisitos.

4.2.1 Modelo Aplicacional

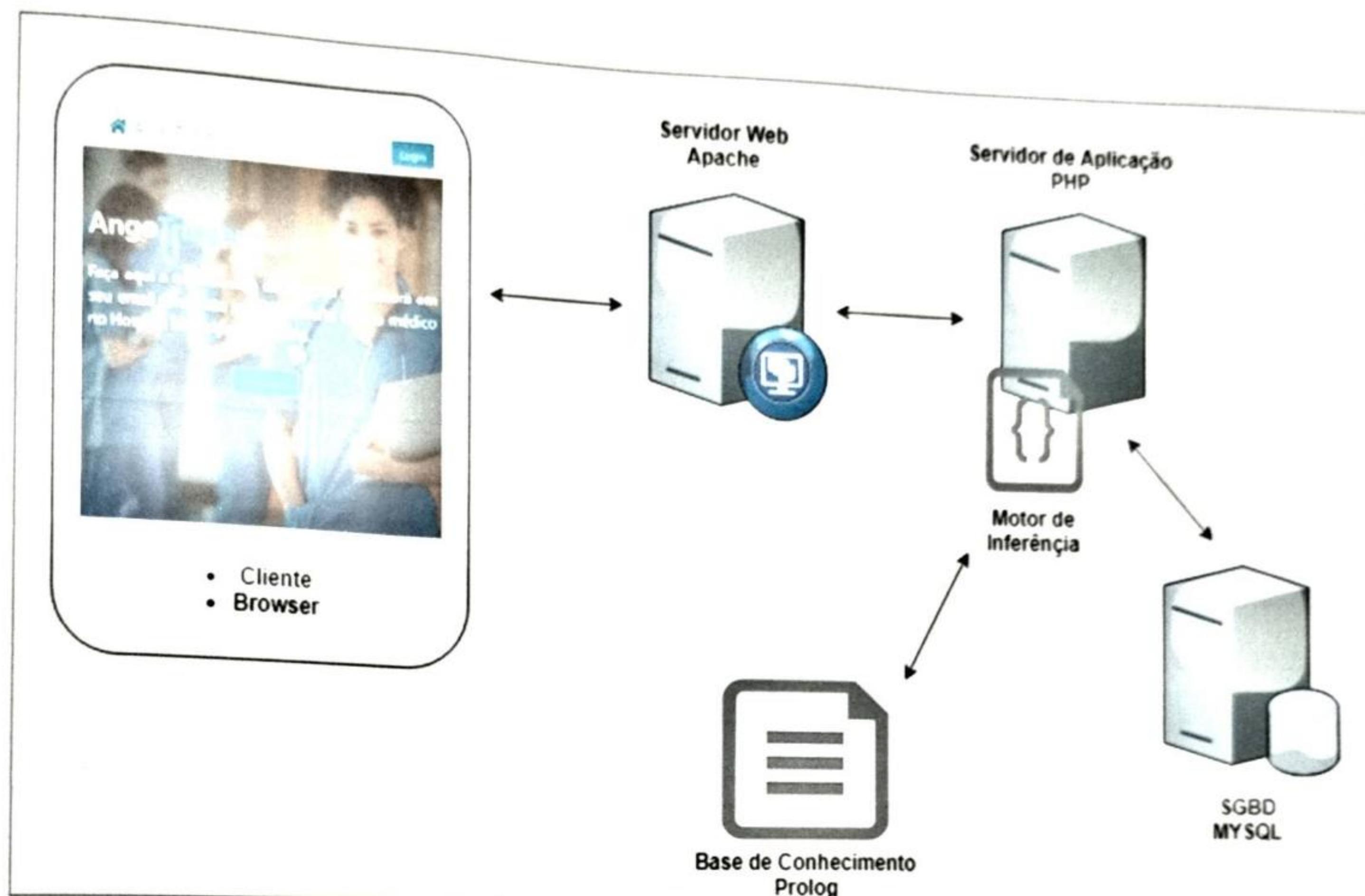


Figura 8: Modelo Aplicacional

Fonte: Autora.

O usuário acessará o sistema (**Interface do Usuário**) apartir de um browser para efectuar as suas ações, em seguida será enviada uma requisição para o **Servidor Web** que é o responsável no processamento das requisições (HTML) feitas pelo usuário e na devolução das mesmas. Neste caso o servidor web encaminhará a requisição para o servidor de aplicação.

O **Servidor de Aplicação**, processará os códigos em linguagem de programação devolvendo a resposta (HTML) para o servidor web. O servidor de aplicação poderá fazer uma consulta a **Base de Dados**, aonde estão armazenadas todas os dados da aplicação. O **Motor de Inferência (contido no servidor de aplicação)**, fará uma consulta a **Base de Conhecimento** na qual está contido todo o conhecimento do especialista, sobre sintomas e possíveis patologias. A máquina de inferência pegará nos dados resultantes da consulta e fará análises e cálculos para chegar a uma conclusão ou seja a uma possível patologia com um certo grau de certeza (percentagem%).

4.2.2 Caso de Uso de Sistema

Os casos de uso de sistema descrevem o funcionamento do sistema a ser modelado. Para a ilustração e melhor compreensão do funcionamento do protótipo do sistema, criou-se: um diagrama de caso de uso de sistema especificando assim os actores e suas actividades, tabelas de descrição para cada caso de uso e um diagrama de actividade (ver apêndices 5 até 18).

4.3 Construção

Para a construção do protótipo do sistema proposto foram feitas várias implementações como: a interface do usuário, o motor de inferência, a base de conhecimento e a base de dados.

4.3.1 Interface do Usuário

A interface é o mecanismo que faz a comunicação entre o sistema especialista e o usuário. Por ser a parte mais sensível, foi desenvolvida de uma maneira simples e organizada para que o usuário não tenha dificuldades de utilização.

A tela apresenta alguns campos de entrada de dados e uma serie de perguntas que o usuário terá que preencher e responder. As perguntas são feitas apresentando uma animação simples de uma enfermeira com posições diferentes para tentar aproximar o sistema a realidade, como mostra as figuras abaixo.

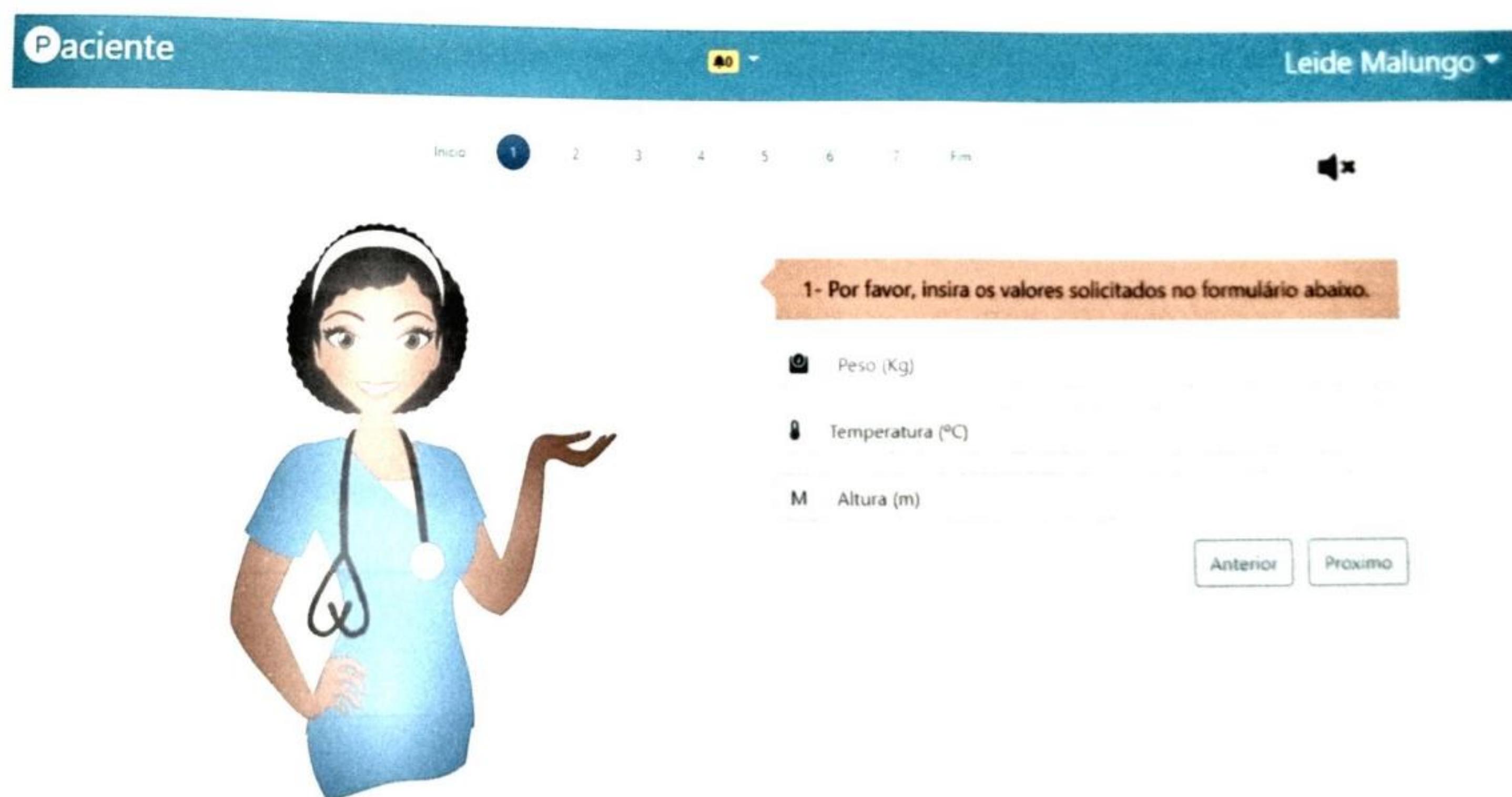


Figura 9: Tela do paciente 1

Fonte: Autora.

A figura acima mostra a tela de inserção de alguns dados do paciente que são obrigatórios como: peso, temperatura, altura.



2- Por favor, responda as perguntas abaixo.

Estás com Febre?

- Provavelmente Não
- Não
- Não Sei
- Provavelmente Sim
- Sim

Estás com Dor de Cabeça?

- Provavelmente Não
- Não
- Não Sei
- Provavelmente Sim
- Sim

Anterior

Próximo

Figura 10: Tela do Paciente 2

Fonte: Autora.

A figura acima mostra a tela de perguntas, na qual o usuário terá que responder para ajudar o sistema na detenção da possível patologia do mesmo.

4.3.2 Base de Conhecimento

A representação do conhecimento foi elaborada utilizando factos e regras de produção na linguagem prolog. O protótipo desenvolvido possui duas bases de conhecimento que são:

1ª Base de conhecimento para armazenamento dos sintomas

Armazena o conhecimento relacionado com os sintomas das patologias em estudo, na qual as suas regras foram manualmente inseridas pelo engenheiro do conhecimento, baseando-se nas entrevistas e pesquisas feitas durante a fase de elaboração.

Possui regras consistentes e perfeitamente definidas na qual, são importantes para o tipo de patologia considerada (Paludismo e Febre Tifoide) e servem de base para uma boa tomada de decisão. Essa representação do conhecimento leva a resultados satisfatórios ou seja leva a implementação de um bom protótipo para o sistema.

Abaixo estão representados em pseudocódigo os fatos e regras utilizadas para a representação do conhecimento do sistema proposto:

- **Fatos**

Paludismo { Febre, Dor de Cabeça, Nauseas ou Vomitos, Suores ou Calafrios, Fraqueza ou Cansaço }

Febre Tifoide { Febre, Dor de Cabeça, Dor Muscular , Tosse, Dor de Barriga, Inchaço na Barriga, Perca de Peso e de Apetite }

Paludismo e Febre Tifoide{ Febre, Dor de Cabeça, Nauseas ou Vomitos, Suores ou Calafrios, Fraqueza ou Cansaço, Dor Muscular , Tosse, Dor de Barriga, Inchaço na Barriga, Perca de Peso e de Apetite }

- **Regras**

Se sintoma == "Febre" **e** sintoma == "Dor de Cabeça" **e** sintoma == "Nauseas ou Vomito" **e** sintoma == "Suores ou Calafrios" **e** sintoma == "Fraqueza ou Cansaço"
Entao
Patologia == Paludismo.

Se sintoma == "Febre" **e** sintoma == "Dor de Cabeça" **e** sintoma == "Dor Muscular"
e sintoma == "Tosse" **e** sintoma == "Dor de Barriga" **e** sintoma == "Inchaço na Barriga" **e** sintoma == "Perca de Peso e de Apetite"
Entao

Patologia == Febre Tifoide.

Se sintoma == "Febre" **e** sintoma == "Dor de Cabeça" **e** sintoma == "Nauseas ou Vomito" **e** sintoma == "Suores ou Calafrios" **e** sintoma == "Fraqueza ou Cansaço" **e** sintoma == "Dor Muscular" **e** sintoma == "Tosse" **e** sintoma == "Dor de Barriga" **e** sintoma == "Inchaço na Barriga" **e** sintoma == "Perca de Peso e de Apetite"
Entao

Patologia == Paludismo.

2ª Base de conhecimento para sinônimos dos sintomas

Armazena os sinônimos dos sintomas existentes no sistema, as suas regras foram inseridas manualmente pelo engenheiro do conhecimento, mas os seus fatos são inseridos toda vez que o especialista associa uma patologia nova a um sintoma antigo. Abaixo está representado em pseudocódigo o fato e a regra utilizada para a representação do conhecimento.

- **Fatos**

Febre{ hipertermia, pirexia}

- **Regras**

Se sintoma == "hipertermia" **ou** sintoma == "pirexia" **Entao**

Se sintoma == "hipertermia" **ou** sintoma == "pirexia" **Entao**

Sinonimo == Febre.

4.3.2.1 Realimentação da Base de conhecimento

A realimentação da base de conhecimento é feita pelo especialista na qual, informa ao sistema se determinados sintomas pertencem ou não a uma determinada patologia e se os novos sintomas inseridos pelo paciente fazem parte ou não de uma das patologias consideradas.

Se um novo sintoma for sinónimo de um sintoma existente, o especialista poderá associa-los e assim o sistema fará um append na base de conhecimento dos sinónimos com um novo fato. Em seguida se o novo sintoma for aceite pelo especialista o sistema fará um append na base de conhecimento com um novo fato e se o especialista informar ao sistema uma nova patologia associada a alguns sintomas é feito um append na base de conhecimento com uma nova regra. Utilizou-se o código abaixo:

File_put_contents("Nome do Ficheiro.pl", "\n \$NovaRegra", FILE_APPEND).

A figura abaixo mostra um fluxograma do momento da ocorrência do processo de realimentação do sistema nas duas bases de conhecimento.

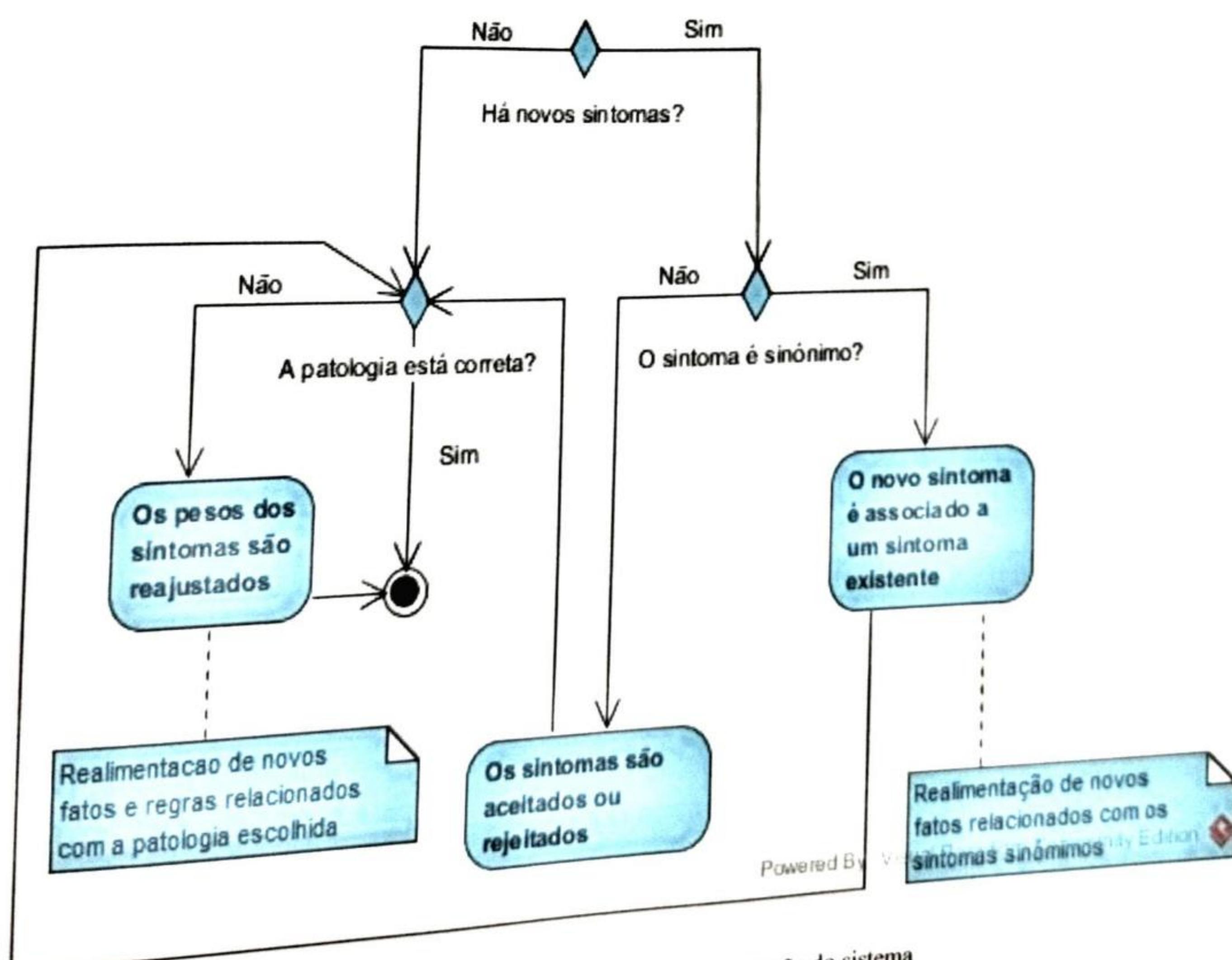


Figura 11: Fluxograma de realimentação do sistema
Fonte: Autora.

4.3.3 Motor de Inferência

Para a construção do motor de inferência utilizou-se o encadramento progressivo e para a quantificação de certezas utilizou-se a lógica fuzzy.

Para cada pergunta feita pelo sistema, existem 5 possíveis respostas que correspondem a uma percentagem de acordo aos conceitos de lógica fuzzy. A figura abaixo apresenta as possibilidades de respostas com as suas respectivas percentagens.

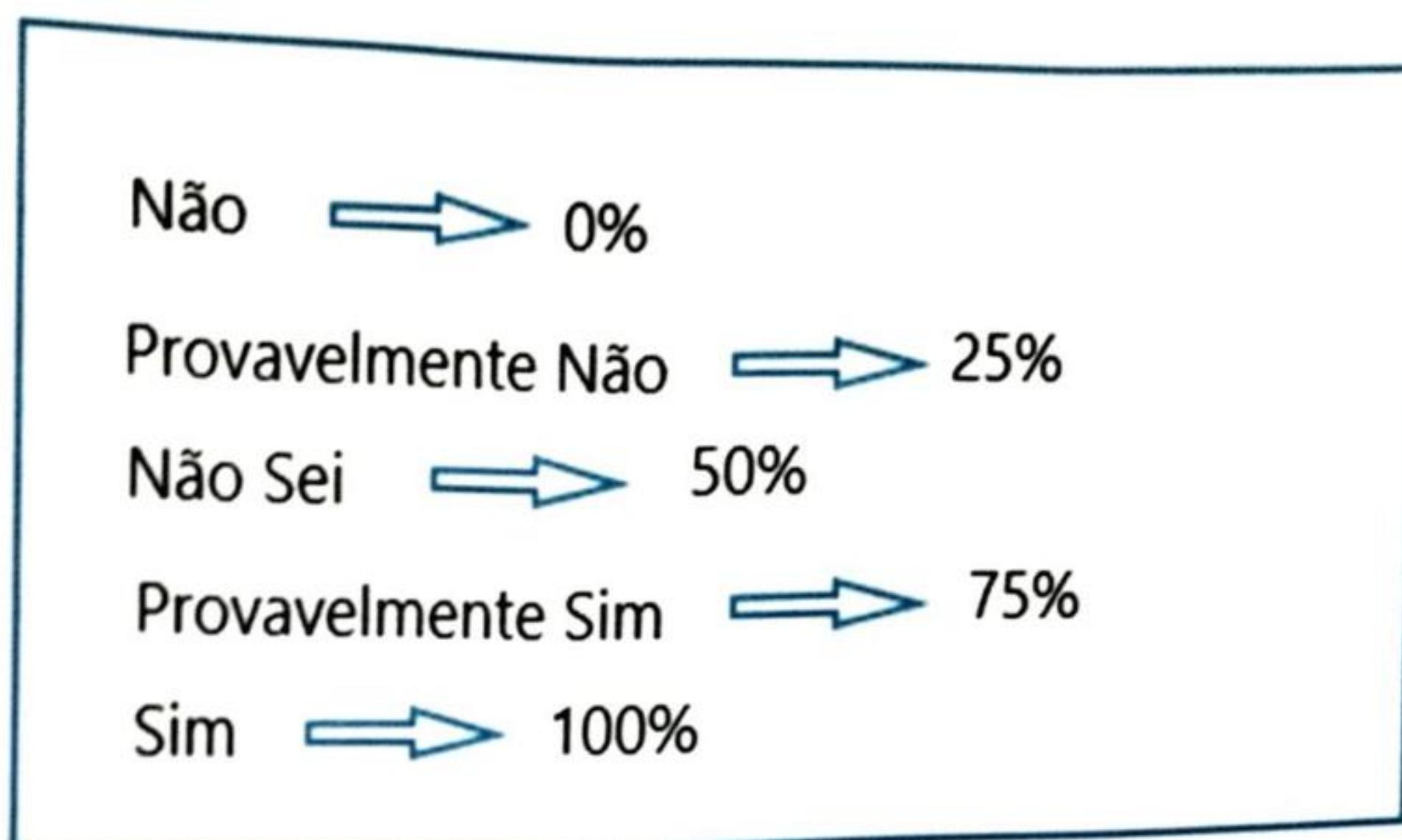


Figura 12: Valores em percentagem para cada Resposta

Fonte: Autora.

Obs: O sintoma só é considerado se o usuário não selecionar a opção **não**.

Depois de identificar os sintomas, a máquina de inferência usa argumentos de linha de comando para fazer o SWI-Prolog carregar e executar consultas na base de conhecimento.

Saida = ‘swipl -s nomeBC.pl -g “paludismo(Sintomas_Estrada)” -t half.’;

O código acima informa ao SWI-Prolog para carregar o ficheiro nomeBC.pl usando o **-s**, passa os argumentos de entrada utilizando o **-g** e encerra o ficheiro com o **-t half**. A variável **Saida** armazenará o retorno (patologia) da consulta feita na base de conhecimento.

Cada sintoma apresentado no sistema tem um valor definido pelo especialista que representa um peso na patologia associada a este sintoma.

Obs: O somatório dos pesos de todos os sintomas associados a uma patologia tem que ser igual a 100%.

O cálculo da percentagem total de certeza de uma patologia é feito utilizando a fórmula:

$$PF = \sum_{i=1}^n \frac{PR * PS}{100}$$

Onde:

PF é a percentagem final;

PR é a percentagem das respostas do usuário;

PS é o peso de cada sintoma considerado pelo usuário;

n é o numero de sintomas considerados pelo usuário.

A baixo encontra-se a representação do cálculo da percentagem final feita pelo motor de inferência do protótipo proposto supondo que o retorno da base de conhecimento seja paludismo.

- **Sintomas do Paciente**

Febre(75%), Dor de Cabeça (75%), Nauseas e Vomitos(100%), Suores ou Calafrios(100%) e Fraqueza e Cansaço(25%).

- **Peso de cada Sintoma**

Febre(25%), Dor de Cabeça(25%), Nauseas e Vomitos(15%), Suores ou Calafrios(15%) e Febre(25%), Dor de Cabeça(25%), Nauseas e Vomitos(15%), Suores ou Calafrios(15%) e Fraqueza e Cansaço(20%).

- **Cálculo da Percentagem final**

$$PF = \frac{(75 * 25) + (75 * 25) + (100 * 15) * (100 * 15) + (25 * 20)}{100}$$

$$PF = 72,5$$

Logo o motor de inferência conclui que o paciente tem 72,5% de Paludismo como mostra a figura abaixo.

■ Diagnóstico

Sintomas

Os Sintomas apresentados pelo paciente são:

Febre, Dor de Cabeça, Nauseas ou Vomitos, Suores ou Calafrios, Fraqueza ou Cansaço.

Novo(s) Sintoma(s): Nenhum

Peso Ideal: 62,5 kg

Possível Patologia: Paludismo 72,5%

Voltar Consulta Realizada

Figura 13: Tela do Diagnóstico

Fonte: Autora.

4.3.4 Estatística

Para o cálculo da probabilidade de ocorrência de cada uma das patologias, utilizou-se a o conceito da probabilidade condicional das redes Bayesianas.

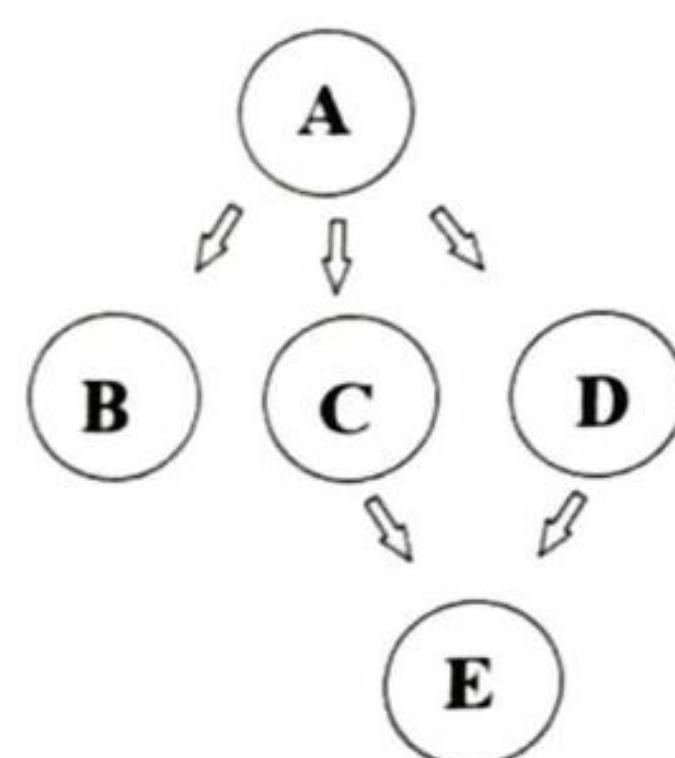


Figura 14: Grafo Orientado das Patologias

Onde:

A: Estar Doente

B: Outra Doença

C: Ter Paludismo

D: Ter Febre Tifoide

E: Ter Paludismo e Febre Tifoide

$$\mathbf{A} = \mathbf{B} + \mathbf{C} + \mathbf{D} + \mathbf{E}$$

Todas as possíveis patologias **B,C e D** para acontecerem dependem de um caso **A** ou seja, o paciente tem que estar doente para que uma das patologias seja detetada. Já a patologia **E** depende das patologias **C e D** ou seja, para que **E** aconteça **C e D** têm que acontecer.

Abaixo, encontra-se a representação do cálculo da probabilidade de ocorrência de paludismo em percentagem supondo que:

$$\mathbf{A} = 25$$

$$\mathbf{C} = 10$$

$$P(C|A) = \frac{P(C \cap A)}{P(A)} * 100$$

$$P(C|A) = \frac{10/25}{25/25} * 100$$

$$P(C|A) = \frac{10}{25} * 100$$

$$P(C|A) = 40\%$$

Então chega-se a conclusão que o paludismo tem 40% de probabilidade em acontecer na próxima triagem que será realizada pelo sistema.

4.4 Transição

4.4.1 Testes

A última etapa do desenvolvimento de software é a fase de testes. É fundamental, pois ajuda a detectar e solucionar erros no software. Para o sistema proposto foram aplicados vários testes que são:

4.4.1.1 Teste de Caixa Preta

É um dos testes mais aplicados em softwares, tem como foco a análise das entradas e saídas de dados dentro do sistema, ou seja ele foca-se na parte externa do software. Aplicou-se o teste de caixa preta cumprindo 3 passos:

1- Análise de requisitos estabelecidos na fase de levantamento de requisitos.

Ex: O paciente não registrado, para ter acesso as funcionalidades do sistema terá que criar uma conta.

• 1998年卷

1. <i>Alkaline</i>	2. <i>Acidic</i>
3. <i>Neutral</i>	4. <i>Very Acidic</i>
5. <i>Extremely Acidic</i>	6. <i>Extremely Alkaline</i>
7. <i>Very Alkaline</i>	8. <i>Extremely Alkaline</i>
9. <i>Extremely Alkaline</i>	10. <i>Extremely Acidic</i>



Figura 16: Teste de caixa preta
Fonte: Autora.

4.4.1.2 Teste de Funcionalidade do Sistema

Este tipo de teste foi aplicado para verificar se todas as funcionalidades do sistema estão de acordo aos requisitos especificados na fase de levantamento. As imagens que se seguem mostram alguns dos testes de funcionalidades feito no sistema.

A figura abaixo mostra a tela do login na qual o usuário insere as suas credenciais.

- **Manipulação de senha**

Para a manipulação de senha utilizou-se o algoritmo **hash** do php para encriptação das senhas dos utilizadores do sistema. O utilizador pode alterar a sua senha desde que a mesma tenha no minimo 4 caracteres.

- **Utilização do PDO (PHP Data Objects)**

Utilizou-se o PDO para fornecer uma camada de abstração em relação a conexão com a base de dados, é bastante útil e facil de implementar. O PDO proteje o sistema de alguns ataques SQL Injection, utilizando o “prepared statements” na formação das queries.

A figura abaixo mostra a inserção de SQL injection no sistema para ter acesso as funcionalidades sem estar registado.

The screenshot shows a login interface with the following fields:

- Username:** Contains the value '' or 50 = 50;#
- Password:** Contains the value
- Tipo de Utilizador:** Set to Paciente
- Entrar (Enter) button:** A large blue button with a circular arrow icon.
- Cancel button:** A red button with a white cancel symbol.
- Error message:** A red banner at the top says "Dados Inválidos".
- Registration link:** "Nao tem conta? Crie uma!" (Don't have an account? Create one!)

Figura 32: Teste de Segurança 1

Fonte: Autora.

• Recaptcha

Recaptcha consiste em um algoritmo de análise de risco que ajuda na proteção dos sites contra a invasão dos robôs. Fornece uma caixa, na qual o utilizador precisa clicar para informar que não é um robô.

Para a sua análise utiliza: movimento do mouse, local de clique, velocidade, rolagem na página etc. Caso o comportamento for normal o sistema valida diretamente o usuário. Caso o comportamento for suspeito o sistema apresenta um desafio de imagem para ter a certeza que o usuário é um humano.

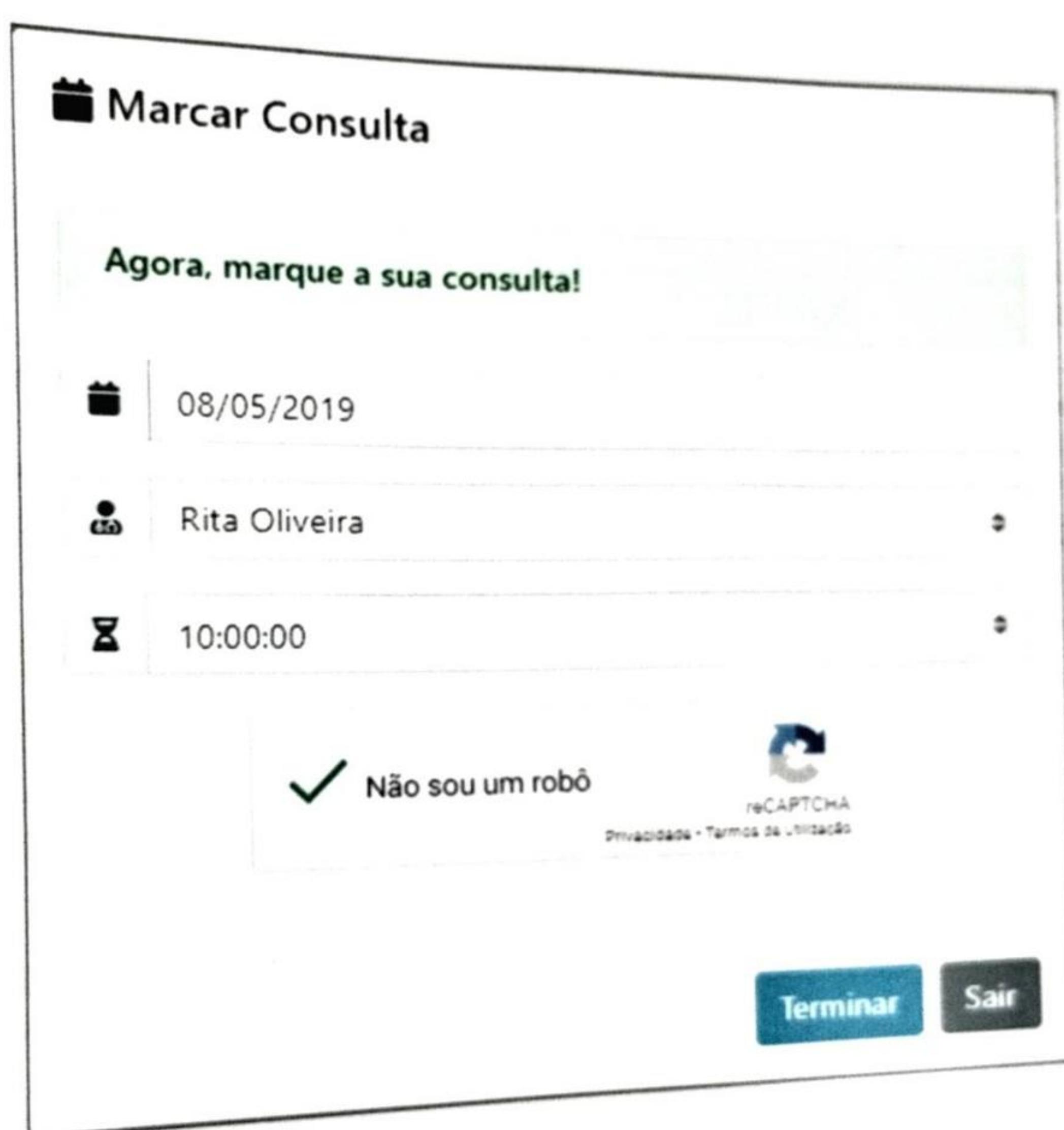


Figura 33: Recaptcha.

Fonte: Autora.

4.4.3 Implantação

Para a estrutura de implantação, o sistema deverá ser implementado num servidor não dedicado que é uma máquina qualquer que compartilha vários serviços para os seus clientes em uma rede como: arquivos, impressoras, etc.

A tabela abaixo mostra os requisitos de hardware e software para a devida implementação do sistema.

Servidor Não Dedicado (Base de dados e Aplicação)		
Processador	Intel(R) Core(TM) i7-8500Y CPU @ 4,20 GHz	
Memória	8Gb	16Gb
HD	2Tb	
Placa de Rede	10/100/100 Mbps	
Base de Dados	MYSQL	
Sistema Operacional	Windows Server 2012	
Web Server	Apache	

Tabela 3: Requisitos de Hardware e Software do Servidor

Fonte: Autora.

Requisitos da Rede

Protocolo	TCP/IP
-----------	--------

Tabela 4: Requisitos da Rede

Fonte: Autora.

PC- CLIENTE

Memória	4Gb
Navegador	Google Crome 72.0.3626.119/Internet Explorer 42.17134.1.0

Tabela 5: Requisitos de Hardware e Software dos Clientes

Fonte: Autora.

4.4.3.1 Diagrama de Implantação

A figura abaixo representa o diagrama de implantação do sistema com os respetivos dispositivos que serão utilizados para o funcionamento da aplicação.

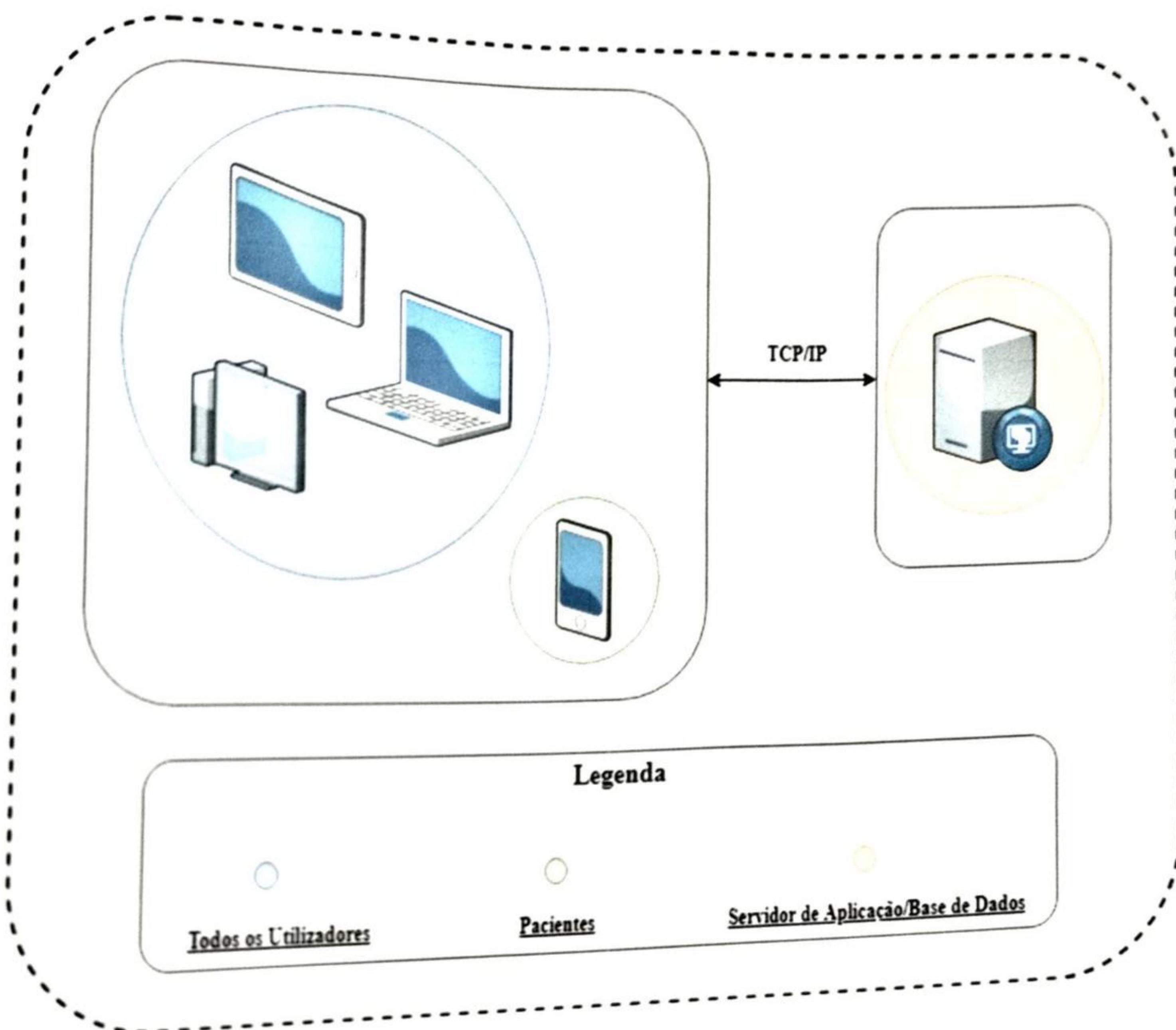


Figura 34: Diagrama de Implementação

Fonte: Autora.

DISCUSSÃO DOS RESULTADOS

Fazendo uma análise comparativa entre o protótipo de sistema especialista desenvolvido (Angotriagem) e os outros existentes (Mycin, Caduceus e Internist-I), o protótipo desenvolvido possui inúmeras vantagens em termos de construção e funcionamento, como é mostrado na tabela abaixo:

Características	MYCIN	INTERNIST-I	CADUCEUS	Sistema Proposto (AngoTriagem)
Área de Aplicação	Medicina	Medicina	Medicina	Medicina
Linguagem de programação para IA	Lisp	-----	Lisp	Prolog
Foco do Sistema	Identificar as bactérias causadoras de infecções graves.	Diagnosticar possíveis doenças relacionadas a medicina interna.	Diagnosticar possíveis doenças relacionadas a medicina interna.	Diagnosticar patologias como paludismo e febre tifoide
Aquisição de Conhecimento	-----	Nenhuma	Raciocínio abdutivo	Método semidireto
Tipo de Aplicação	Desktop	Desktop	Desktop	Web

Tabela 6: Sistema proposto vs outros sistemas relacionados

Fonte: Autora.

Angotriagem é um sistema web que permite o acesso dos pacientes em qualquer lugar desde que esteja vinculado a internet, diferente de todos os outros citado acima.

Possui um método de aprendizado baseado no método semi-direto, na qual depende das informações passadas pelo o especialista. O caduceus possui um método de aprendizado baseado no raciocínio abdutivo que não é muito confiável porque a área da saúde é muito delicada.

Angotriagem tem um módulo de gestão na qual utiliza uma base de dados para armazenamento dos dados da triagem, que é utilizado para históricos e etc, diferente dos outros sistemas especialistas que não trabalham com uma base de dados e tratam todos os casos com uma primeira vez.

Angotriagem engloba dois módulos totalmente diferentes: o módulo de gestão e o módulo especialista. No módulo de gestão é permitido fazer todo gerenciamento dos utilizadores (registar, atualizar, eliminar e pesquisar). Possui 3 utilizadores que são: o paciente, o médico e o administrador, na qual o paciente beneficia-se desse módulo para fazer as suas marcações de consultas e o médico beneficia-se para ter listado todas as informações de consultas marcadas de forma organizada.

O módulo de gestão do protótipo Angotriagem não é o foco principal do sistema mas sim uma consequência de implementação, porque durante a fase de elaboração verificou-se a necessidade da utilização de uma base de dados para os benefícios citados acima e também pensando em implementações futuras.

No módulo especialista que realmente é o foco do sistema efetua uma triagem no paciente, aplicando algumas questões e no final fornece ao médico uma patologia analisada pela sua base de conhecimento, acompanhada com um grau de certeza (calculado pelo seu motor de inferência) para posteriormente ser analizada pelo médico.

Também permite verificar a probabilidade de ocorrência de cada uma das patologias e com isso permitirá fazer uma análise dos dados para saber se há a necessidade de estudar as outras patologias fora do escopo do trabalho ou seja, quanto mais a probabilidade das patologias estudadas se aproximar a 0%, significa que essas deixaram de ser as mais frequentes.

Angotriagem está preparado para implementações futuras, atualmente analisa os dados apartir de sintomas mas futuramente também poderá analisar apartir de um histórico de cada paciente, ajudando assim na detenção da possível patologia.

De acordo com resultados da aplicação de um inquérito com base a um questionário (ver apêndice 21 e 22), feito para o médico e para o paciente, conclui-se que Angotriagem é um sistema simples de usar mas de acordo a algumas limitações apresentadas, não substitui um enfermeiro mas auxilia o médico na detenção da possível patologia e na previsão da mesma (consultas futuras).

CONCLUSÕES

Atravez de pesquisas feitas foi possível fazer o levantamento bibliográfico acerca da gestão hospitalar, saber mais sobre o funcionamento dos hospitais, quais são as falhas e soluções a aplicar.

Aplicou-se uma pesquisa bibliográfica das patologias paludismo e febre tifoide, sabendo quais são as causas, sintomas e métodos de prevenção.

Fez-se o levantamento bibliográfico dos sistemas especialistas, sabendo a sua constituição, maneira de construção, vantagens e desvantagens para aplicar no desenvolvimento do protótipo do sistema que foi desenvolvido.

Atravez das pesquisas feitas no hospital municipal da samba, aplicando as técnicas de observação e entrevistas, foi possível saber quais são as técnicas ou procedimentos tradicionais da triagem médica, como são detetadas as patologias etc, a obtenção dessas informações auxiliaram na modelação de um sistema especialista com foco na deteção de patologias.

O prototipo desenvolvido “Angotriagem” auxilia o médico na deteção de possíveis patologias como paludismo e febre tifoide calculando um nível de certeza em percentagem. Ajuda o médico a saber qual patologia tem mais probabilidade de acontecer no próximo paciente a ser atendido. Possue uma base de conhecimento com todas as regras possíveis mais está preparado para o aprendizado supervisionado pelo médico.

Angotriagem não sofrerá com algumas desvantagens dos sistemas especialista porque:

- Atua numa área específica (triagem médica) e deteta patologias específicas (paludismo e febre tifoide).
- Tem um procedimento de aprendizado supervisionado, baseado no método semi-direto.
- Tem uma técnica de medição do seu desempenho baseada na quantificação de acertos e erros das patologias inferidas pelo sistema.