



INSTITUTO FEDERAL DE SERGIPE
COORDENADORIA DO CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO
ALEX DE OLIVEIRA SOUZA

SITPA - SISTEMA INTELIGENTE BASEADO EM REGRAS ADAPTÁVEIS PARA TRIAGEM
DE PACIENTES EM UNIDADES DE SAÚDE

TRABALHO DE CONCLUSÃO DE CURSO

LAGARTO
2021

ALEX DE OLIVEIRA SOUZA

SITPA - SISTEMA INTELIGENTE BASEADO EM REGRAS ADAPTÁVEIS PARA TRIAGEM
DE PACIENTES EM UNIDADES DE SAÚDE

Trabalho de conclusão de curso apresentado ao Curso de Bacharelado em Sistemas de Informação, como parte dos requisitos necessários à obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Gilson Pereira dos Santos Júnior
Coorientador: Thiers Garreti Ramos Sousa

Souza, Alex de Oliveira.

S713s SITPA – Sistema inteligente baseado em regras adaptáveis para triagem de pacientes em unidades de saúde / Alex de Oliveira Souza. – Lagarto, 2022. 65 f. ; il.

Monografia (Graduação) – Bacharelado em Sistemas de Informação. Instituto Federal de Educação Ciência e Tecnologia de Sergipe – IFS, 2022.

Orientador: Prof. MSc. Gilson Pereira dos Santos Júnior.

Co-Orientador: Prof. MSc. Thiers Garreti Ramos Souza.

1. Tecnologia da informação. 2. Desenvolvimento de sistemas. 3. Rede de computador. 4. Sistema inteligente. I. Instituto Federal de Educação Ciência e Tecnologia de Sergipe. II. Título.

CDU: 005.94



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SERGIPE
Av. Jorge Amado, 1551 – Loteamento Garcia, Bairro Jardins – CEP 49025-330 – Aracaju/SE Fone: (79) 3711 1400 – E-mail: reitoria@ifs.edu.br

APÊNDICE 10 – TERMO DE APROVAÇÃO
(com coordenador)

CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

TERMO DE APROVAÇÃO

SITPA – SISTEMA INTELIGENTE BASEADO EM REGRAS ADAPTÁVEIS PARA
TRIAGEM DE PACIENTES EM UNIDADES DE SAÚDE

ALEX DE OLIVEIRA SOUZA

Esta monografia foi apresentada às 20:00 horas do dia 13 de setembro de 2021 como requisito parcial para a obtenção do título de **Bacharel em Sistemas de Informação**.

O candidato foi arguido pela banca examinadora composta pelos examinadores abaixo assinados.

Após deliberação, a Banca Examinadora considerou o trabalho APROVADO.

Documento assinado digitalmente
 CATUXE VARJAO DE SANTANA OLIVEIRA
Data: 25/07/2022 12:42:34-0300
Verifique em <https://verificador.iti.br>

MSc. Catuxe Varjão de Santana Oliveira
Instituto Federal de Sergipe



MSc. Gilson Pereira dos Santos Júnior
(Orientador)
Instituto Federal de Sergipe

Documento assinado digitalmente
 LAURO BARRETO FONTES
Data: 25/07/2022 11:05:24-0300
Verifique em <https://verificador.iti.br>

MSc. Lauro Barreto Fontes
Instituto Federal de Sergipe

ASSINADO DIGITALMENTE
THIERS GARRETTI RAMOS SOUSA
CPF
00683958500
DATA
25/07/2022
A conformidade com a assinatura pode ser verificada em:
<http://serpro.gov.br/assinador-digital> 

MSc. Thiers Garretti Ramos Sousa
(Coordenador)
Instituto Federal de Sergipe

Documento assinado digitalmente
 CATUXE VARJAO DE SANTANA OLIVEIRA
Data: 25/07/2022 12:39:32-0300
Verifique em <https://verificador.iti.br>

Assinatura do (a) Coordenador (a) do Curso
Instituto Federal de Sergipe

Lagarto/SE, 13 de setembro de 2021

Dedicatória. . .

Aos meus pais, à minha companheira Grace, a todos os meus professores, desde o ensino primário até a graduação, e a todos os meus amigos que sempre me estimularam a estudar e a seguir em frente.

AGRADECIMENTOS

Não foi fácil chegar até aqui, mas em meio a muitas adversidades impostas pela vida, enfim chegou esse dia! Em todo esse tempo de graduação, aconteceram várias coisas em minha vida, coisas que algumas vezes me desmotivaram a continuar estudando, mas felizmente tenho pessoas em minha vida que me impulsionam a seguir em frente, independente do que aconteça.

Agradeço em especial a minha mãe e ao meu pai, pois sempre estiveram, estão e estarão ao meu lado. Sempre me deram incentivo para estudar e nunca mediram esforços para tal, mesmo sendo de uma simples família do interior, me deram todo o suporte necessário para que eu continuasse estudando, tenho certeza que é um enorme orgulho para eles, presenciar esse momento tão importante na minha vida.

Agradeço à minha companheira, Grace, pois, como meus pais, sempre me incentivou a estudar e há 17 anos ao meu lado, me apoiando e me aconselhando nas decisões mais importantes da minha vida. Ocorreram momentos em que as coisas ficaram difíceis e ela sempre lá, sempre aqui, me motivando a seguir em frente, sem essa motivação talvez eu não tivesse chegado até aqui. Obrigado Grace, te amo!

A todos os meus professores, sem exceção. Desde a minha primeira professora, que me alfabetizou: minha mãe, até os meus professores da graduação, dentre os quais levarei pra vida toda como exemplos pessoais e profissionais a serem seguidos. Gostaria de agradecer ao professor Wilhelm, que pra mim é um exemplo de pessoa, de professor, de profissional, enfim, um exemplo de ser humano. Lembro-me que em momentos difíceis no IFS, Will me ajudou bastante, sempre muito atencioso e disposto a ajudar no que fosse necessário, seja na minha vida pessoal, profissional ou acadêmica, muito obrigado Willy! Tenha certeza que você é um exemplo pra mim e para todos que o rodeiam! Agradeço também às professoras Jislane, que me atura em um dos seus projetos, a professora Cristiane que conheço lá da época que eu estudei Automação Industrial, e sempre foi essa simpatia em forma de pessoa, a Ana Júlia, que mesmo diante das dificuldades impostas por uma pandemia, me ensinou um idioma novo, a LIBRAS. Agradeço também ao professor Jonatas, pois assim como eu, é um amante da aviação, lembro que às vezes conversávamos sobre o tema nos momentos de intervalo e muito do que sei de Administração devo a esse cara! Muito obrigado Glauco, Jean, Jonas, Lauro, Almerindo e Rubens, que me mostraram o mundo maravilhoso das redes de computadores, à professora Catuxe, por toda atenção e carinho, enfim, a todos que contribuíram para que eu me tornasse uma pessoa melhor, muito obrigado!

Agradeço ao professor Gilson, meu orientador, por ter aceitado essa empreitada e por todos os ensinamentos passados nas disciplinas, sempre atencioso e preocupado. Agradeço pela atenção que me foi dada em todo esse período que compartilhamos no IFS e fora dele. Muito obrigado por me apresentar ao mundo da Inteligência Artificial,

esse mundo fascinante que me apaixono mais a cada dia. Ao professor Thiers, que foi de suma importância para este trabalho, sem ele, eu não teria a menor noção de como funciona um hospital. Obrigado Thiers! Agradeço também ao professor Francisco, sem seus ensinamentos, eu não saberia programar utilizando os conceitos de Orientação a Objeto. Um agradecimento especial ao professor George, que me apresentou o mundo Java WEB, utilizado no desenvolvimento desse trabalho. Sempre disposto a responder minhas perguntas da forma mais clara possível.

Agradeço a todos os meus amigos e familiares que sempre estiveram comigo, independente de qualquer coisa, me estimulando e me dando suporte quando necessário. Em especial ao meu amigo Uanderson, um cara que tenho uma admiração muito grande e que me ensinou muita coisa. Meu amigo Plínio Elkson, que está presente desde a infância. Sempre incentivando os amigos a estudar e a alcançar melhores resultados na vida! Muito obrigado irmão! Aos meus amigos Laécio e Isaac, pelos conhecimentos compartilhados e conselhos dados. E um agradecimento muito especial a uma das melhores pessoas que já conheci nessa vida, minha amiga Isabela Marília, que suporta minhas reclamações de trabalho quase que diariamente. Sempre atenciosa e amável, um exemplo de pessoa, de profissional, de ser humano. O tipo de pessoa que a gente se sente bem só de estar perto sabe? Muito obrigado Belinha! Agradeço aos meus companheiros das bandas Capão Black e Flaming Blood, os momentos de estresse sempre descarregamos nas músicas que tocamos, e por falar em música, agradeço também ao professor Alberto Silveira, meu professor de música do conservatório,

Enfim, agradeço a todos que direta ou indiretamente contribuíram para me conduzir até este momento peço desculpas às pessoas que não mencionei diretamente, pois não caberiam todos neste pequeno espaço reservado aos agradecimentos.

“Se continuarmos desenvolvendo nossa tecnologia sem sabedoria ou prudência, nosso servo pode acabar se tornando nosso carrasco”.

Osmar Bradley

RESUMO

Com o aumento da superlotação nas urgências e emergências dos hospitais, seja por motivos gerenciais, ou problemas de infraestrutura, faz-se necessário o uso de ferramentas para tentar reduzir esse fluxo, entre essas ferramentas estão os protocolos de classificação de risco. No entanto, nem sempre esses protocolos são aplicados de forma correta. Uma vez que o fluxo nas urgências e emergências é muito intenso, às vezes não sobra tempo para realizar uma triagem adequada dos pacientes e em algumas ocasiões a triagem deixa de ser feita, o que implica em uma diminuição na qualidade do atendimento. Surge então a necessidade de uma ferramenta capaz de classificar os pacientes de forma automatizada, por meio de sessões onde os dados informados pelo usuário ativam regras pré definidas capazes de inferir as classificações de risco de um paciente por conseguinte uma possível melhoria no fluxo de pessoas nas urgências e emergências dos hospitais. Este trabalho tem como tema o desenvolvimento de um sistema inteligente, baseado em regras adaptáveis, para a triagem de pacientes nos serviços de urgência e emergência dos hospitais, que atue especificamente nesses ambientes. O sistema desenvolvido pretende, com a sua implantação, possibilitar a diminuição do tempo de espera nas urgências e emergências dos hospitais, consequentemente diminuindo as longas filas e por conseguinte uma possível melhora na qualidade do atendimento.

Palavras-chave: superlotação, protocolos de triagem, sistema especialista, raciocínio baseado em regras.

ABSTRACT

Due to the increase in overcrowding in the emergency hospitals, whether for management reasons or infrastructure problems, it is necessary to use tools to try to reduce this flow, among these tools we can mention the risk classification protocols. However, these protocols are not always applied correctly. Since the flow in the emergencies is very high, sometimes there is no time to carry out an adequate triage and, in some occasions, the triage is not performed, which implies in a decrease in the quality of the health care. Due to this fact, there is a need for a tool capable of classifying patients in an automated way, through sessions where the data informed by the user activates pre-defined rules capable of inferring a patient's risk classification, therefore a possible improvement in the flow of people in the emergencies in the hospitals. This work has as main objective the development of an intelligent system, based on adaptable rules, for the triage of patients in the emergency services of hospitals. The system developed intends, with its implementation, to make it possible to reduce the waiting time in hospital emergencies and consequently reducing long queues, therefore a possible improvement in the quality of health care.

Keywords: overcrowding, triage protocols, intelligent systems, rule-based reasoning.

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxograma “Agressão” presente no Protocolo de Triage <i>Manchester</i> .	20
Figura 2 – Usuários de Internet, por dispositivos utilizados	22
Figura 3 – Exemplo de uma regra de produção no framework Jboss Drools	26
Figura 4 – Arquitetura de um sistema especialista baseado em regras	28
Figura 5 – Diagrama de classes simplificado do SITPa.	31
Figura 6 – Diagrama de casos de usos do SITPa.	32
Figura 7 – Drools Engine	34
Figura 8 – Regra de negócio em linguagem .drl	36
Figura 9 – Arquitetura simplificada do SITPa.	37
Figura 10 – Diagrama Entidade-Relacionamento da aplicação	39
Figura 11 – Classe de configuração do <i>Drools</i>	40
Figura 12 – Tela inicial do SITPa.	44
Figura 13 – Tela para criação de um novo grupo	44
Figura 14 – Tela para criação de novo fluxogramas	45
Figura 15 – Tela para criar uma nova classificação	46
Figura 16 – Tela de cadastro de sintomas	47
Figura 17 – Tela para associar sintomas a um fluxograma	48
Figura 18 – Tela para classificar um paciente	49
Figura 19 – Tela para escolha de sintomas	50
Figura 20 – Tela de resumo dos dados informados na figura 19	50
Figura 21 – Tela de classificação final	51
Figura 22 – Classificações no banco de dados do SITPa	51
Figura 23 – Método que verifica a prioridade dos sintomas informados	53
Figura 24 – Tela teste - Diabetes	53
Figura 25 – Resultado da classificação do paciente da Figura 24	54
Figura 26 – Regra utilizada para inferir a classificação da Figura 25	54
Figura 27 – Tela teste - Gravidez	55
Figura 28 – Resultado da classificação da paciente da Figura 27	56
Figura 29 – Regra utilizada para inferir a classificação da Figura 28	56
Figura 30 – Tela teste - Alergia	56
Figura 31 – Resultado da classificação do paciente da Figura 30	57
Figura 32 – Regra utilizada para inferir a classificação da Figura 30	58
Figura 33 – Tela teste - Convulsões	58
Figura 34 – Resultado da classificação do paciente da Figura 33	59
Figura 35 – Regra utilizada para inferir a classificação da Figura 34	59
Figura 36 – Classificações de teste salvas no banco de dados	60

LISTA DE TABELAS

Tabela 1 – Classificação dos sintomas escolhidos na Figura 24	54
Tabela 2 – Classificação dos sintomas escolhidos na Figura 27	55
Tabela 3 – Classificação dos sintomas escolhidos na Figura 30	57

LISTA DE ABREVIATURAS E SIGLAS

ATS	Australasian Triage Scale
BC	Base de Conhecimento
CTAS	Canadian Triage and Acuity Scale
DER	Diagrama Entidade-Relacionamento
DMN	Decision Model and Notation
DRL	Drools Rule Language
HTML	Hyper Text Markup Language
IA	Inteligência Artificial
IDE	Integrated Development Enviroment
IHC	Interface Homem-Computador
JDK	Java Development Kit
JVM	Java Virtual Machine
MI	Mecanismo de Inferências
MT	Memória de Trabalho
MTS	Manchester Triage System
MVC	Model, View, Controller
S.I.T.P.	Sistema Inteligente de Triagem de Pacientes
SGBD	Sistema Gerenciador de Banco de Dados
TCU	Tribunal de Contas da União
TIC	Tecnologia da Informação e Comunicação
UML	Unified Modeling Language

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.1.1	Objetivo geral	14
1.1.2	Objetivos específicos	14
1.2	Justificativa	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Classificação de Risco	16
2.2	Sistemas de Triagem	17
2.2.1	<i>Emergency Severity Index (ESI)</i>	17
2.2.2	<i>Australasian Triage Scale (ATS)</i>	18
2.2.3	<i>Canadian Triage and Acuity Scale (CTAS)</i>	18
2.2.4	<i>Manchester Triage System (MTS)</i>	18
2.3	Informática na área da saúde	20
2.4	Sistemas informatizados na área da saúde	22
2.5	Inteligência Artificial e os Sistemas Inteligentes	23
2.5.1	Sistemas Especialistas ou Sistemas de Dedução Baseados em Regras	24
2.5.1.1	Componentes de um sistema especialista	25
2.5.1.1.1	<i>Base de Conhecimento (BC)</i>	26
2.5.1.1.2	<i>Mecanismo de Inferências (MI)</i>	26
2.5.1.1.3	<i>Memória de Trabalho (MT)</i>	27
2.5.1.1.4	<i>Interface Homem-Computador (IHC)</i>	27
2.5.1.1.5	<i>Arquitetura de um Sistema Especialista Baseado em Regras</i>	27
3	SITPA	29
3.1	Especificação de Requisitos	29
3.1.1	Requisitos Funcionais	29
3.1.2	Requisitos não -funcionais	30
3.1.3	Modelagem do sistema	31
3.1.3.1	Diagrama de Classes	31
3.1.3.2	Diagrama de casos de uso	31
3.2	Construção da Base de Conhecimento	32
3.2.1	Linguagem Java	33
3.2.2	JBoss Drools	33
3.2.2.1	<i>Drools Engine</i>	34
3.2.2.2	Métodos do JBoss <i>Drools</i> na construção da base de conhecimento	35
3.3	Implementação do Sistema	37

3.3.1	Back End	40
3.3.2	Front End	41
4	RESULTADOS E DISCUSSÃO	43
4.1	Resultados	43
4.2	Discussão	52
5	CONCLUSÕES E TRABALHOS FUTUROS	61
5.1	Conclusões	61
5.2	Principais contribuições	61
5.3	Trabalhos futuros	62
	REFERÊNCIAS	63

1 INTRODUÇÃO

A triagem é um processo que consiste na separação de acordo com características comuns com o intuito de se identificar grupos de acordo com o padrão ou a importância. A separação de objetos ou pessoas por características é um exemplo de triagem. Este procedimento é muito utilizado na área da saúde, em hospitais, postos de saúde e unidades de pronto atendimento. A utilização da triagem nestes tipos de organizações é fundamental para determinar a prioridade na hora de tratar os pacientes, separando-os em grupos, baseando-se na gravidade dos casos destas pessoas e o grau de urgência no seu atendimento. Assim, o uso da triagem de paciente em uma unidade de saúde otimiza o atendimento dos pacientes, priorizando os casos mais urgentes e utilizando de maneira eficiente os recursos disponíveis na instituição que, geralmente, são limitados.

Na grande maioria dos hospitais brasileiros existe um grande problema, a superlotação nas urgências e emergências. De acordo com um levantamento feito pelo Tribunal de Contas da União em 2014, onde 116 hospitais, de todas as esferas, federais, estaduais e municipais, foram visitados, foi constatado que 83% destes hospitais, sempre, ou na maioria das vezes, há na emergência, pacientes em atendimento em quantidade superior à quantidade de leitos oficiais, e em 64% desses hospitais, a taxa de ocupação das emergências ultrapassa 100%, (TCU, 2014).

Esse excesso de pacientes é causado, geralmente, por falhas na gestão, precariedade na infraestrutura desses ambientes, aumento da procura por hospitais em desvantagem dos postos de saúde, e sem deixar de mencionar a deficiência ou falta da triagem adequada dos pacientes. Outro fator que tem um papel significativo, neste caso, é a falta de pessoal qualificado. Ainda de acordo com o levantamento realizado pelo TCU, constatou-se que em 80% dos hospitais faltam médicos e enfermeiros, isso é mais um fator que contribui para a superlotação. Devido a falta destes profissionais, leitos tendem a ser fechados, e a triagem deixa de ser feita, o que implica no acúmulo de pacientes nas instituições de saúde no Brasil.

Com essa carência de mão de obra especializada e poucos recursos, muitas unidades de saúde não conseguem realizar a triagem de pacientes, desta forma gera-se um problema muito grave que é a definição de prioridades na hora do atendimento, isso acarreta em aumento no tempo de espera por atendimento médico, além de pôr em risco a saúde dos pacientes.

Além das causas já mencionadas, pode-se destacar a deficiência na assistência primária à saúde, que segundo Santos et. al. (2003), este é o primeiro contato do usuário com o atendimento médico e onde estão contidos os “serviços ambulatoriais providos de profissionais de formação geral que devem oferecer os cuidados básicos de promoção, manutenção e recuperação da saúde”, o que resolveria a maior parte dos problemas que aparecem. Seria de suma importância que este tipo de atendimento fosse mais abrangente,

acessível e refletisse a realidade da comunidade. Muitas vezes, as pessoas encontram dificuldades na hora de marcar um atendimento, pois a demanda por atendimento em muitos casos é maior do que a oferta, de modo que, os pacientes se deslocam para as urgências e emergências dos hospitais, onde, segundo Rocha (2005), são neles que a população tem garantia de atendimento médico.

Dessa forma, a superlotação é um problema crescente, uma vez que pacientes que, teoricamente, poderiam ser atendidos pela rede primária, se misturam aos que necessitam de atendimento nas redes secundárias ou terciárias. E por consequência, os recursos ficam cada vez mais escassos, além de que, pacientes com doenças de menor gravidade podem ser expostos a riscos de infecção e contaminação, pondo em risco suas vidas.

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo geral do presente trabalho é desenvolver um sistema inteligente baseado em regras adaptáveis para a triagem de pacientes em unidades de saúde.

1.1.2 Objetivos específicos

- Desenvolver um sistema inteligente, tendo como referência protocolos de classificação de risco já consolidados.
- Investigar tecnologias para o desenvolvimento de sistemas *web* baseados em regras.
- Construir uma arquitetura adaptável a diferentes tipos de processos de triagem por meio de regras.
- Desenvolver uma estrutura para implementação de um sistema baseado em regras para o processo de triagem de pacientes.
- Desenvolver uma interface *web* de fácil navegação para consulta no processo de triagem de pacientes.
- Validar a aplicação usando regras criadas a partir de protocolos de classificação de risco validados.

1.2 Justificativa

Os hospitais brasileiros vêm sofrendo com o problema da superlotação nas urgências e emergências há vários anos e com o intuito de melhorar o atendimento nesses estabelecimentos, visando a diminuição dessa superlotação, pensou-se na criação de um

sistema inteligente, baseado em regras adaptáveis, para triagem de pacientes em unidades de saúde, tendo como referência protocolos de classificação de risco já consolidados.

O atual sistema de classificação utilizado nos hospitais, requer um tempo considerável para classificar um paciente, uma vez que, ao acolhê-lo no hospital, o profissional responsável deve verificar o principal sintoma apresentado e após isso, normalmente, consultar o Protocolo de classificação de risco utilizado na organização, em busca do sintoma apresentado. Após encontrá-lo, será necessário observar os sintomas específicos relacionados ao sintoma principal para, só então, classificar o paciente. Esse processo atrasa muito a triagem, o que acarreta no acúmulo de pessoas nas urgências e emergências dos hospitais.

A escassez de trabalhos publicados na literatura que tratem de sistemas computacionais para facilitar o processo de triagem nas emergências e urgências motivou o desenvolvimento deste trabalho, pois, a saúde necessita de uma solução rápida na hora da realização da triagem médica, visto que uma triagem eficiente pode salvar vidas.

Foi encontrado, no Brasil, um sistema desenvolvido para melhoria no sistema de triagem intitulado *Smart Sorting*. A solução denominada *Smart Sorting*, segundo Júnior et. al. (2019), consiste no uso de um conjunto de sensores conectados à uma placa microcontroladora que coleta informações de alguns sinais vitais do paciente e envia os dados para um sistema, onde, essas informações serão filtradas e o paciente será classificado de acordo com a gravidade dos sintomas apresentados, e esse nível de gravidade será baseado no padrão de classificação de risco utilizado pela instituição de saúde. Esta solução foi apresentada no Seminário de Tecnologias Aplicadas em Educação e Saúde em 2019.

Diferentemente do *Smart Sorting*, a solução proposta neste trabalho não utiliza sensores para aquisição dos sinais vitais do paciente durante o processo de triagem. Esses dados são informados pelo usuário do sistema ao consultá-lo. As regras de classificação de risco utilizadas serão cadastradas ou adaptadas pela unidade de saúde, de modo que um banco de dados armazenará os fluxogramas e os possíveis sintomas para instânciação do sistema de classificação de risco.

Neste sistema, a classificação de risco será feita por meio de uma interface *web* com telas de fácil navegação, consultadas com apenas alguns cliques, para otimizar o processo de triagem e automatizá-lo, com respostas reproduzíveis, cabendo ao profissional, apenas, informar os sintomas apresentados pelo paciente durante a consulta do sistema. Ressalta-se que é responsabilidade da unidade de atendimento cadastrar e adaptar, previamente, as regras para classificação de risco dos pacientes no sistema de acordo com o protocolo estabelecido na unidade.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Classificação de Risco

A classificação de risco foi amplamente utilizada por militares ao longo da história. Esse tipo de classificação foi utilizado com o objetivo de identificar as pessoas, no campo de batalha, que necessitavam de atenção médica mais urgente e priorizar estas pessoas de tal maneira a receber o atendimento médico mais rapidamente. (COUTINHO 2012).

De acordo com Souza et. al. (2013), a classificação de risco pode ser considerada como um mecanismo onde os pacientes que se encontram em situação de risco tenham seu atendimento priorizado e, dessa forma, sejam separados de acordo com a gravidade dos casos, eliminando assim as filas e o atendimento por ordem de chegada, que é comumente utilizado nos hospitais públicos do Brasil.

Ainda de acordo com Souza et. al. (2013), devido à alta demanda, os serviços de urgência e emergência no Brasil ainda permanecem lutando para organizar o fluxo de pacientes, bem como administrar todo esse fluxo, consequência, normalmente, de um atendimento por ordem de chegada, e isso pode acarretar em sérios danos à saúde de pacientes que necessitam de atendimento prioritário.

Nas palavras de Pires (2003), a triagem significa uma escolha sobre quais pacientes serão priorizados na hora do atendimento, bem como a organização do fluxo dos pacientes no sistema e selecionado os meios adequados para um diagnóstico e posterior tratamento.

De forma semelhante, Gravel et. al.(2009),fala, que uma triagem válida, deve atribuir prioridade aos pacientes que necessitam de cuidados urgentes e identificar os pacientes que podem esperar com segurança pelo atendimento.

Nas palavras de Moll (2010), os sistemas de triagem são fundamentados no consenso entre especialistas que definem regras para a avaliação do risco clínico, bem como dos recursos necessários para definir os níveis de urgência. Estes sistemas devem ser simples para serem efetivos, no entanto, devem possuir uma gama de sintomas que fará com que o sistema possa ser aplicado à população em geral.

O processo de classificação de risco ou triagem de pacientes, foi evoluindo ao longo da história, e a partir da década de 60 começou-se a aplicá-lo à população civil.

De acordo com Jiménez (2013), os objetivos da classificação de risco são:

- Identificar rapidamente os pacientes em situação de risco de morte.
- Determinar a área mais adequada para tratar o doente que se apresenta ao serviço de emergência.
- Reduzir o congestionamento nas áreas de tratamento do serviço de emergência, para melhoria do fluxo de pacientes.

- Garantir a reavaliação periódica dos pacientes.
- Informar aos pacientes e famílias o tipo de serviço de que necessita e o tempo estimado de espera.
- Assegurar as prioridades em função do nível de classificação.
- Contribuir com informações que ajudem a definir a complexidade do serviço, casuística, eficiência, carga de trabalho, consumo de recursos e satisfação do usuário.
- Priorizar apenas o acesso ao atendimento e não fazer diagnósticos.

Na visão de Cordeiro et. al. (2002), um sistema de triagem tem como objetivos:

- Identificar de forma eficiente, os pacientes que apresentam doença ou lesão emergente ou urgente.
- Avaliar precocemente todos os pacientes, a fim de assegurar-lhes a prioridade de acesso aos cuidados, de acordo com a gravidade do caso.
- Antecipar exames de apoio ao diagnóstico ao atendimento médico, seguindo os protocolos já definidos.
- Determinar a área mais adequada para atendimento do paciente.
- Regular o fluxo de usuários e descongestionar os serviços de emergências.

2.2 Sistemas de Triagem

Segundo Moll (2010) os sistemas que são mais utilizados nos serviços de emergência são: o *Emergency Severity Index* (ESI), o *Australasian Triage Scale* (ATS), a *Canadian Triage and Acuity Scale* (CTAS), e o *Manchester Triage System* (MTS).

2.2.1 *Emergency Severity Index (ESI)*

Desde 1998, o ESI tem sido o principal recurso de triagem utilizado por quase todas as enfermeiras de emergência nos Estados e atualmente é usado por aproximadamente 80% dos hospitais dos EUA. (ENA, 2021).

Este sistema de triagem identifica a prioridade de um paciente através de uma escala composta por cinco níveis de gravidade, organizados em ordem crescente onde, através de um algoritmo, o enfermeiro identifica a prioridade do atendimento.

2.2.2 *Australasian Triage Scale (ATS)*

Escala que surgiu em Melbourne na década de 70, inicialmente chamada *Ipswich Triage Scale*, e com o passar do tempo foi informatizada e implantada em vários hospitais australianos. No ano 2000 foi padronizada pelo sistema de saúde australiano e passou a denominar-se *Australasian Triage Scale*. (ACEM, 2000; DIOGO, 2007; PARDEY, 2006).

Além de estabelecer um registro de data e horário da avaliação, responsável pela triagem, queixa principal, entre outros, como dito por Pires (2003), esta escala ainda estabelece que o paciente, enquanto espera, ou caso surjam achados clínicos relevantes, deve ser constantemente reavaliado.

2.2.3 *Canadian Triage and Acuity Scale (CTAS)*

A CTAS, é uma escala para triagem de pacientes, composta por cinco níveis de prioridades associados a cores e possui um manual com a descrição e detalhamento das características em cada nível de prioridade.

A Canadian Triage and Acuity Scale, disponibiliza um arquivo eletrônico contendo 406 dispositivos que permitem que o profissional de saúde acesse rapidamente os protocolos disponíveis para auxiliá-lo no momento da triagem.

2.2.4 *Manchester Triage System (MTS)*

O Sistema de Triagem *Manchester*, desenvolvido pelo *Manchester Triage Group*, segundo Lahdet et. al. (2009), vem sendo adotado em vários países ao redor do mundo, como Reino Unido e Irlanda, onde é o método de triagem nacional destes países e está presente em países como Japão, Portugal entre outros.

Este sistema de triagem foi muito estudado, o que gerou evidências científicas quanto à validação de constructo, sendo muito sensível na identificação de pacientes críticos nos serviços de emergência (MACKWAY-JONES, 2010).

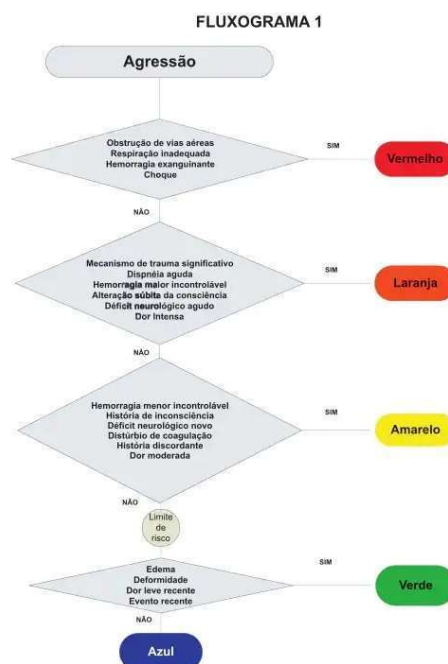
O *MTS* é uma metodologia científica que classifica o risco para pacientes que vão em busca de atendimento médico nos hospitais, postos de saúde, unidades de pronto atendimento etc. Este sistema de triagem, de acordo com (MACKWAY-JONES, 2010), referencial utilizado neste trabalho, é composto por 52 fluxogramas que são utilizados para classificar a gravidade, estes fluxogramas são divididos de tal forma que se possa identificar a queixa principal apresentada pelo paciente, e esta, será utilizada para escolher o fluxograma adequado para a classificação.

O Sistema de Triagem *Manchester* é dividido em cinco categorias. Estas, possuem um número, um nome, uma cor e um tempo máximo aceitável até o atendimento médico, vejamos abaixo estas categorias:

- Nível 1: emergente, vermelho: imediato;
- Nível 2: muito urgente, laranja: 10 minutos;
- Nível 3: urgente, amarelo: 60 minutos;
- Nível 4: pouco urgente, verde: 120 minutos;
- Nível 5: não urgente, azul: 240 minutos

Uma das grandes vantagens da utilização deste sistema é que os pacientes são priorizados para serem atendidos diante da gravidade da situação e não por ordem de chegada ao local de atendimento, isso é algo muito positivo para os profissionais da saúde, pois, desta forma eles conseguem ter uma visão geral da quantidade de pacientes presentes no ambiente, bem como a gravidade das situações de cada um, o que possibilita que esses profissionais, ajam de maneira mais eficaz.

O processo de triagem utilizando este sistema, tem início com o preenchimento de uma ficha no local de atendimento. Após isso, o paciente é encaminhado para uma sala de triagem. Este serviço é feito comumente por um enfermeiro, o qual identifica a principal queixa do paciente e associa o sintoma informado a um dos fluxogramas que compõem o protocolo. Mediante as respostas informadas pelo paciente, o profissional responsável irá atribuir uma cor para o risco do paciente em questão, a triagem então termina com o registro de todos os dados colhidos pelo profissional e logo em seguida o encaminhamento do paciente para outros serviços de saúde ou, em muitas vezes, para o local de espera para o atendimento.

Figura 1 – Fluxograma “Agressão” presente no Protocolo de Triage *Manchester*

Fonte: MACKWAY-JONES (2010)

Para o desenvolvimento deste trabalho, não será adotado nenhum dos protocolos de classificação de risco mencionados anteriormente, visto que, a proposta consiste na implementação de um sistema inteligente baseado em regras adaptáveis, sendo assim o protocolo a ser utilizado será proveniente do hospital que fizer uso desta solução.

No entanto, diante da dificuldade em realizar um estudo de caso, em virtude do isolamento social advindo da pandemia de COVID-19, e mediante a necessidade de validar o sistema através de regras já consolidadas por protocolos validados, far-se-á uso do protocolo de triagem *Manchester*, única e exclusivamente para fins de validação da solução proposta neste trabalho.

Como principal referência relacionada ao sistema de triagem *Manchester*, foi utilizado o livro “Sistema Manchester de classificação de risco”, versão brasileira do livro escrito por Mackway-Jones, Mardsen e Windle, em 2010.

2.3 Informática na área da saúde

O uso da informática na área da saúde é uma das formas de aplicação das Tecnologias da Informação e Comunicação que mais vem crescendo nos últimos anos. De acordo com uma pesquisa realizada pelo cetec.br no ano de 2017, 85% dos estabelecimentos públicos de saúde fazem uso de computadores, enquanto que nos estabelecimentos de saúde privados esse percentual aumenta para 100% (CETIC BR, 2018).

Segundo Camargo (2016), o uso das TIC's permite uma grande melhoria nas condi-

ções de saúde da população, visto que, elas têm o papel de melhorar processos, além de permitir que profissionais de várias áreas da medicina troquem conhecimento entre eles, o que acarreta em uma melhoria na qualidade do atendimento.

De acordo Shortliffe e Blois (2006), a informática na área da saúde é conceituada como:

“um campo científico que trata do armazenamento, recuperação, e do uso da informação biomédica, dados e conhecimentos para a solução rápida de problemas e tomadas de decisão.” (SHORLIFFE e BLOIS, 2006).

Já para Evora (2007), a informática pode ser compreendida atualmente, como o ramo tecnológico que trata de processamento de informações, onde, “processamento de informação” pode ser substituído, em qualquer contexto, por Tecnologia de Informação ou Sistemas de Informação.

Segundo Baggio et. al.(2010), a tecnologia da informação vem tendo um avanço constante na área da saúde e devido a esse grande avanço os recursos de TI estão sendo cada vez mais utilizados por pessoas em todo o mundo.

Como sabemos, no setor da saúde tudo é baseado em informação e essas informações precisam estar dispostas de forma organizada, para que o profissional da saúde tenha uma maior acurácia na hora de diagnosticar um paciente. Estes diagnósticos são feitos, através da análise das informações coletadas previamente através do contato médico/paciente, juntamente com o conhecimento técnico que estes profissionais possuem, o que as torna imprescindível na tomada de decisão.

De acordo com Paes (2003) a disseminação da tecnologia ligada às ciências da informação, como a informática, permite o avanço e o desenvolvimento de ferramentas de organização das informações obtidas pelo profissional da saúde.

Pelo pensamento de Shortliffe et. al. (2006), o grande desafio da informática médica é compor instrumentos que auxiliem os profissionais da saúde, e partindo deste pressuposto, Blum (1986), afirmou que a tecnologia não visa a substituição do médico, mas sim auxiliá-lo, montando uma base organizada de conhecimento técnico de informações sobre os pacientes, permitindo, desta forma, o fácil acesso a elas.

Sendo a medicina, juntamente com a informática, disciplinas com altos níveis técnicos, é de fundamental importância que haja um diálogo entre os profissionais destas áreas para que as necessidades de todos sejam satisfeitas, possibilitando assim o sucesso na implementação de sistemas que atendam às necessidades destes profissionais, pois como dito por Paes:

“A compreensão do funcionamento dos processos médicos e das expectativas dos profissionais permitirá desenvolver sistemas mais adequados ao uso para o cuidado do paciente e aumentar sua utilização”. (PAES, 2003).

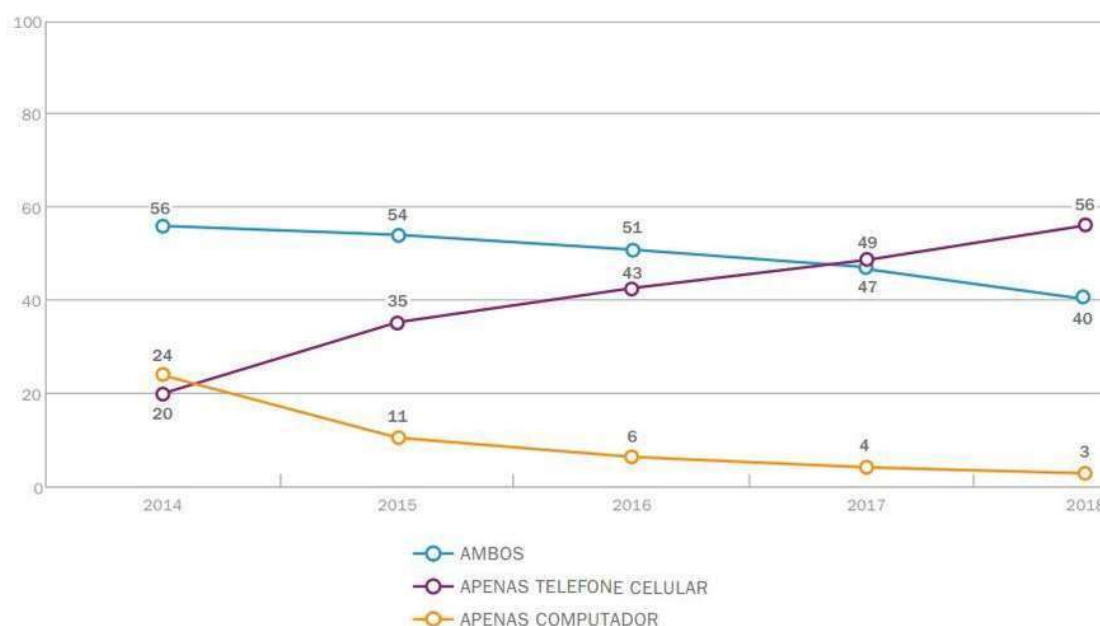
2.4 Sistemas informatizados na área da saúde

Mediante o avanço da tecnologia nos últimos tempos, busca-se constantemente a procura e o desenvolvimento por ferramentas que facilitem as atividades das várias áreas no mercado, visando uma maior eficiência no atendimento e também a modernização do modo como as atividades são realizadas.

De acordo com pesquisa intitulada “Pesquisa sobre o uso das tecnologias de informação e da comunicação no Brasil”, realizada pelo Cetic, em 2018, 42,1 milhões de domicílios brasileiros tinham acesso à internet, sendo 96% apenas por meio de dispositivos móveis. (CETIC, 2018).

Segundo essa pesquisa, existem cerca de 127 milhões de internautas no Brasil, o que representa um crescimento de 37% nos últimos cinco anos, pois no ano de 2013 apenas metade da população estava conectada, agora esse percentual chega a 70%.

Figura 2 – Usuários de Internet, por dispositivos utilizados



Fonte: Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (Cetic.br), Pesquisa sobre o Uso das Tecnologias de Informação e Comunicação nos domicílios brasileiros - TIC Domicílios 2018.

Com o crescimento constante no uso da tecnologia, a área da saúde buscou também adentrar neste meio digital. Porém, a maior parte dos sistemas desenvolvidos para essa área são considerados multiprofissionais, ou seja, servem como uma ferramenta e como auxílio para diversos profissionais da saúde (TIBES et. al. 2014).

O uso de ferramentas digitais pelos profissionais da saúde permite que a tomada de decisão se dê de forma mais rápida, porém ainda segundo Tibes (2014), uma lacuna a ser explorada ainda é a dos softwares voltados para o paciente, pois quando em comparação à

quantidade voltada para os profissionais são poucos os que auxiliam os pacientes, sejam para acompanhamento de tratamentos, farmacológicos ou simplesmente informativos.

2.5 Inteligência Artificial e os Sistemas Inteligentes

Definir “artificial” é relativamente simples, pois, já que tudo que não ocorre de forma natural, pode ser considerado artificial. Já a definição de “inteligência” é algo mais complexo. E “Inteligência Artificial?”. Bem, esta, segundo Rich (1994), IA é o estudo de como fazer com que os computadores possam executar tarefas que, até então, os homens conseguem fazer melhor do que as máquinas.

De acordo com Russel e Norvig (1995), a inteligência artificial herdou muitas ideias e técnicas de outras disciplinas, dentre elas, a ciência cognitiva. A hipótese de que as pessoas compreendem o mundo através da construção de modelos mentais permite enumerar os itens fundamentais para todos os campos da ciência cognitiva: Psicologia, Linguística, Filosofia e Ciência da Computação.

A Inteligência artificial pode ser aplicada em vários campos de atuação. Muitas das tarefas de que a IA trata podem ser divididas em tarefas “corriqueiras”, como aquelas realizadas através da percepção, ou ainda aquelas de raciocínio de senso comum, as tarefas formais, que englobam a geometria, lógica, jogos, cálculo integral, entre outros e as tarefas especialistas, como análise científica ou diagnóstico médico, por exemplo. (ROSA, 2011).

Para Rowe (1998), a IA inclui:

- Fazer o computador se comunicar com o ser humano em línguas naturais, através do processamento de linguagens naturais.
- Fazer com que o computador se lembre de fatos inter-relacionados e fazer uma inferência lógica sobre esses fatos.
- Fazer o computador planejar sequencias de ações para o alcance de metas.
- Fazer o computador oferecer ajuda baseada em regras para várias situações (sistemas especialistas ou sistemas de dedução baseados em regras).
- Fazer o computador olhar através de câmeras e enxergar o mundo à sua volta.
- Fazer com que o computador mova objetos do mundo real (robótica).

Para ser capaz de realizar as tarefas descritas por Rowe (1998), a IA utiliza várias ferramentas, desde a lógica clássica até as simulações de redes neurais.

Sistemas inteligentes é um termo que abrange as descrições de diversas aplicações que fazem uso de IA na sua construção. A inteligência artificial gira em torno de duas linhas

de raciocínio: estudar os processos do pensamento humano e recriar esses processos por meio de máquinas, tais como computadores e robôs. Logo, seu principal objetivo é projetar e construir máquinas que reproduzam a inteligência humana.

Os principais sistemas inteligentes são:

- Sistemas especialistas ou Sistemas de Dedução Baseados em Regras;
- Processamento de linguagem natural;
- Reconhecimento por voz;
- Redes neurais;
- Lógica difusa;

Para desenvolver a solução proposta nessa pesquisa, serão utilizados os conceitos de sistemas especialistas, ou sistemas de dedução baseados em regras, pois, esse tipo de sistema é o mais indicado, devido à necessidade que o sistema possui em armazenar conhecimento, uma vez que, por se tratar de um tipo de sistema flexível, permite que as regras que irão compor a base de conhecimentos, sejam adaptáveis a qualquer unidade de saúde.

2.5.1 Sistemas Especialistas ou Sistemas de Dedução Baseados em Regras

Os sistemas especialistas são sistemas inteligentes que pretendem “imitar” o comportamento dos especialistas humanos dentro de uma determinada área, baseando-se em processos heurísticos (método de investigação baseado na aproximação progressiva de um dado problema).

Segundo Munárriz (1994), o comportamento de um sistema especialista diante de um problema em um domínio específico, em nada deve diferir do comportamento de uma pessoa especialista neste domínio, diante de um mesmo problema.

Já nas palavras de Heinzle (1995), os sistemas especialistas são aqueles que foram projetados para encontrar a solução de problemas que normalmente exigem especialistas humanos com o conhecimento na área de domínio da solução.

Ainda de acordo com Heinzle, os sistemas especialistas caracterizam-se por armazenar um vasto e profundo conhecimento acerca de uma área específica devendo armazenar estas informações de forma organizada permitindo simplificar a busca as respostas requeridas. HEINZLE (1995).

Na literatura, alguns autores apresentam conceitos formais para os sistemas especialistas:

Feigenbaum escreve:

"... um programa inteligente de computador que usa conhecimento e procedimentos inferenciais para resolver problemas que são bastante difíceis, de forma a requererem para a sua solução muita perícia humana." (FEIGENBAUM, 1983).

Na visão de Ribeiro:

"Um sistema especialista deve, além de inferir conclusões, ter capacidade de aprender novos conhecimentos e, desse modo, melhorar o seu desempenho de raciocínio e a qualidade de suas decisões." (RIBEIRO, 1987).

Já para Rabuske:

Sistemas especialistas devem, também, ter habilidade para aprender com a experiência e explicar o que estão fazendo e porque o fazem. Esta última é uma das principais características que distinguem estes sistemas dos tradicionais sistemas de informação". (RABUSKE, 1995).

De acordo com Heinze (1995), os sistemas especialistas possuem duas características principais que os distinguem dos sistemas tradicionais. A primeira é a manipulação de informações compostas por fatos a respeito do domínio e regras formais que descrevem essas relações, enquanto que a segunda é a utilização de técnicas de inferência para manipular informações visando uma solução. (HEINZE, 1995).

O objetivo destes tipos de sistemas é captar o conhecimento do especialista no domínio escolhido e representar este conhecimento em uma base, chamada de base de conhecimento, para então transmiti-lo ao usuário, permitindo que este, obtenha respostas às perguntas relacionadas ao domínio de conhecimento do sistema.

Portanto, podemos afirmar que os Sistemas Especialistas podem ser entendidos como uma das técnicas de IA que foi desenvolvida para resolver problemas em um determinado contexto. Dentro deste mesmo contexto, o conhecimento necessário é extraído das pessoas que são especialistas em determinado domínio, através das ferramentas de aquisição utilizadas por um engenheiro de conhecimento, e então todo o conhecimento é posto na chamada Base de Conhecimento do sistema.

2.5.1.1 Componentes de um sistema especialista

De forma geral, um sistema especialista é composto por quatro componentes básicos:

- Base de conhecimento;
- Mecanismo de inferências;
- Memória de trabalho;
- Interface Homem-Computador

2.5.1.1.1 Base de Conhecimento (BC)

A base de conhecimento (BC) é o componente que armazena todo o conhecimento. Normalmente, a base é conhecida como Base de Regras de Produção (BRP) do tipo: Se (situação) Então (ação). Ver Figura 3.

A quantidade e a qualidade de conhecimento armazenado neste componente é crucial para determinar a qualidade do sistema que está sendo implementado, e para Heinzle (1995) esta base de conhecimento deve ser o mais flexível possível às atualizações.

Por ser um componente indispensável nestes tipos de sistema, a fase de construção desta base é muito complexa, uma vez que o conhecimento do especialista, na maioria das vezes, está dentro da própria cabeça, e para isso faz-se necessário toda uma preparação para a extração desse conhecimento.

Figura 3 – Exemplo de uma regra de produção no framework Jboss Drools

```
rule "Hello World"
  when
    m: Message( status == Message.HELLO, myMessage: message)
  then
    System.out.println( myMessage );
    m.setMessage( "Goodbye cruel word1" );
    m.setStatus( Message.GOODBYE );
    update( m );
  end

rule "GoodBye"
  when
    Message( status == Message.GOOGBYE, myMessage : message )
  then
    System.out.println( myMessage);
  end
```

Fonte: https://www.tutorialspoint.com/drools/images/default_rule.jpg

2.5.1.1.2 Mecanismo de Inferências (MI)

O mecanismo de inferências (MI) é o processador ou interpretador do conhecimento, e pode ser considerado como o coração de um Sistema Especialista, pois, ele é responsável por buscar, selecionar e avaliar todas as regras oriundas da base de conhecimento.

Nas palavras de Heinzle (1995):

“O motor de inferência é o elemento do sistema especialista que é capaz de buscar na base o conhecimento necessário a ser avaliado em cada situação, direcionar o processo de raciocínio gerenciar situações de incerteza e levar ao resultado final.”(HEINZLE, 1995).

A principal função do MI é combinar o conhecimento abstrato contido na base de de conhecimento com o conhecimento concreto armazenado na base de fatos, ou seja

cruzar as informações que foram passadas pelo usuário com as já contidas na base de conhecimento do sistema, inferindo assim conclusões para as informações que foram utilizadas no processo.

2.5.1.1.3 Memória de Trabalho (MT)

A memória de trabalho (MT) é o elemento que armazena o conhecimento concreto, ou seja, o conhecimento que pode ser considerado como fato antes de passar pelo mecanismo de inferência. Esta memória é de caráter transitório, pois continuamente fatos estão sendo acrescentados e apagados, uma vez que o mecanismo de inferências faz constantes consultas em busca de fatos novos.

2.5.1.1.4 Interface Homem-Computador (IHC)

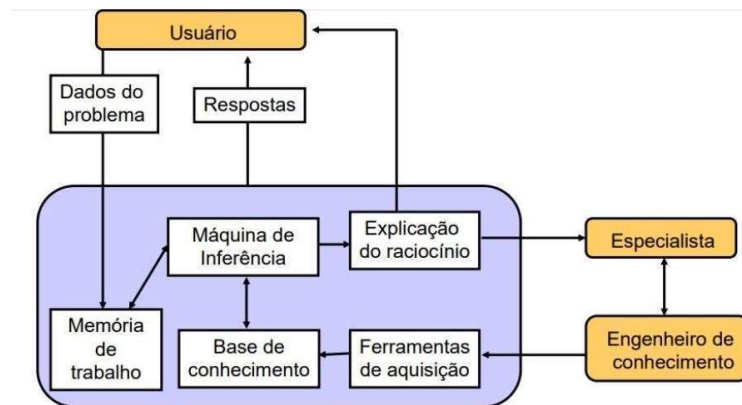
É uma parte muito importante na hora da construção do sistema, pois é ela que irá fornecer o meio de comunicação entre o usuário e o sistema, embora esta seja uma área muito negligenciada no desenvolvimento dos sistemas especialistas.

Um sistema especialista deve fornecer ao usuário uma interface de simples operação, onde seu uso seja compreendido por todos, inclusive as pessoas que não possuem conhecimentos computacionais, e como foi dito por Heinzle:

“Aspectos internos dos sistemas, terminologia computacional etc. devem ser evitados e detalhes técnicos relativos a implementação devem ser transparentes ao usuário. A linguagem a ser utilizada deve ser orientada para o problema ou para a área do especialista e o mais perto possível da linguagem natural.” (HEINZLE, 1995).

2.5.1.1.5 Arquitetura de um Sistema Especialista Baseado em Regras

A Figura 4 detalha a arquitetura de um sistema especialista baseado em regras, mostrando a interação entre vários dos seus componentes, inclusive os já mencionados neste trabalho.

Figura 4 – Arquitetura de um sistema especialista baseado em regras

Fonte: <https://image.slidesharecdn.com/se-1227968774480962-9/95/sistemas-especialistas-26-728.jpg?cb=1227940623>

O engenheiro de conhecimento, através de técnicas de extração de conhecimento, faz uso de ferramentas de aquisição, como entrevistas, a fim de obter o conhecimento necessário para a construção da base de conhecimento. Com este conhecimento adquirido e representado na base, o sistema é capaz de inferir sobre o domínio para o qual ele foi projetado.

O usuário insere as informações (fatos) e esses são carregados para a memória de trabalho, então o mecanismo de inferências busca esses dados juntamente com os dados armazenados na base de conhecimento. De posse dessas informações a máquina de inferências cruza os dados em busca de resultados.

Uma vez que os resultados são obtidos, o usuário recebe uma resposta do sistema, bem como a explicação do raciocínio, que por sua vez também é apresentada ao especialista com fins de validação.

Assim que o especialista valida o raciocínio feito pelo sistema, pode-se dizer que este, está pronto para uso. Caso os resultados não sejam satisfatórios por algum motivo, o engenheiro de conhecimento utiliza novamente as ferramentas de aquisição de conhecimento com a finalidade de obter este conhecimento de uma forma mais precisa, e então corrige os dados da base de conhecimento e todo o ciclo se inicia novamente.

3 SITPA

Este trabalho destinou-se a desenvolver o SITPa, um sistema Inteligente Baseado em regras Adaptáveis para Triagem de Pacientes em Unidades de Saúde. Nele, as regras serão cadastradas e adaptáveis para qualquer protocolo que seja utilizado nos locais onde a solução for implantada. Conforme mencionado, neste trabalho, o SITPa busca auxiliar os profissionais de saúde na hora de realizar a triagem médica, vale salientar que a solução é totalmente adaptável a qualquer protocolo e não faz uso de nenhum dos sistemas de triagem citados. No entanto, foi necessário conhecer a estrutura destes protocolos para modelagem, desenvolvimento, teste e validação do sistema proposto.

Optou-se pelo desenvolvimento de um sistema baseado em regras utilizando o *JBoss Drools*, devido ao fato de que, normalmente, protocolos possuem regras muito bem definidas, não deixando margem para erros, além de que, por usar o Drools, as regras ficam independentes da aplicação, e isso implica em uma maior facilidade nas manutenções, adições ou exclusões de regras na base de conhecimento do sistema.

3.1 Especificação de Requisitos

De acordo com Pressman (2011), a especificação de requisitos consiste em um documento, elaborado antes do início do desenvolvimento, que contém uma descrição detalhada das características de um projeto. Normalmente essa especificação de requisitos é dividida em dois tipos: requisitos funcionais e não funcionais.

3.1.1 Requisitos Funcionais

De acordo com Ramos (2006), os requisitos funcionais são aqueles que especificam as atividades ou funções do sistema a ser desenvolvido. Inicialmente, os requisitos listados abaixo foram estabelecidos para o SITPa:

- **Criar um grupo :** O usuário será capaz de criar um grupo para ser associado a fluxogramas.
- **Criar uma classificação:** O usuário deverá criar uma classificação, contendo uma cor e um nível de prioridade que varia de 1 a 5, sendo 1 o mais grave e 5 o menos grave.
- **Criar fluxograma:** O usuário será capaz de criar um fluxograma onde, neste, serão adicionados sintomas relacionados a ele. Para a criação de um fluxograma é obrigatório ter pelo menos um grupo cadastrado.
- **Criar sintoma:** Será possível a criação de sintomas que serão utilizados para a construção dos fluxogramas.

- **Associar sintomas a um fluxograma:** O usuário será capaz de associar sintomas já cadastrados aos fluxogramas cadastrados previamente.
- **Selecionar um grupo:** Dentre os grupos criados, o usuário será capaz de selecionar o grupo desejado e dessa forma todos os fluxogramas associados àquele grupo ficarão disponíveis para uso.
- **Selecionar um fluxograma:** O usuário será capaz de selecionar um fluxograma dentre os cadastrados, e então terá acesso aos sintomas associados ao fluxograma escolhido.
- **Selecionar sintomas de um fluxograma:** O usuário será capaz de escolher múltiplos sintomas na tela de sintomas relacionados a um fluxograma escolhido.
- **Verificar informações inseridas no sistema:** O sistema deverá mostrar antes de cada classificação, as informações de: grupo, fluxograma e sintomas informados pelo usuário.
- **Classificar um paciente:** O usuário será capaz de classificar um paciente informando apenas o grupo, o fluxograma e os sintomas apresentados.
- **Mostrar o resumo da classificação feita pelo sistema:** O sistema será capaz de mostrar um resumo da classificação feita pelo sistema contendo: Classificação, grupo escolhido, local mais adequado para atendimento, sintoma mais grave apresentado pelo paciente, o tipo de atendimento e o tempo para atendimento.
- **Reiniciar classificação:** A qualquer momento deverá ser possível reiniciar uma classificação.
- **Alterar Informações:** O sistema deverá permitir operações de criação, edição e deleção de grupos, fluxogramas, classificação e sintomas.

3.1.2 Requisitos não -funcionais

Os requisitos não funcionais estabelecem condições que o sistema deve satisfazer (RAMOS, 2006). Abaixo, são listados os requisitos não funcionais para o SITPa:

Plataforma de desenvolvimento: A interface *web* deve ser desenvolvida na linguagem java 8 e *spring boot*, utilizando a IDE Eclipse.

Interface com os usuários: deve ser desenvolvida uma interface *web* para que os usuários possam interagir com o sistema.

Persistência de dados: O SITPa deve utilizar como banco de dados relacional, o Postgresql em sua versão 10 ou superior.

3.1.3 Modelagem do sistema

Antes de iniciar o desenvolvimento do sistema, foi elaborada uma modelagem simplificada, usando os diagramas da UML (*Unified Modeling Language*). Onde, nas palavras de Folerca, a UML é:

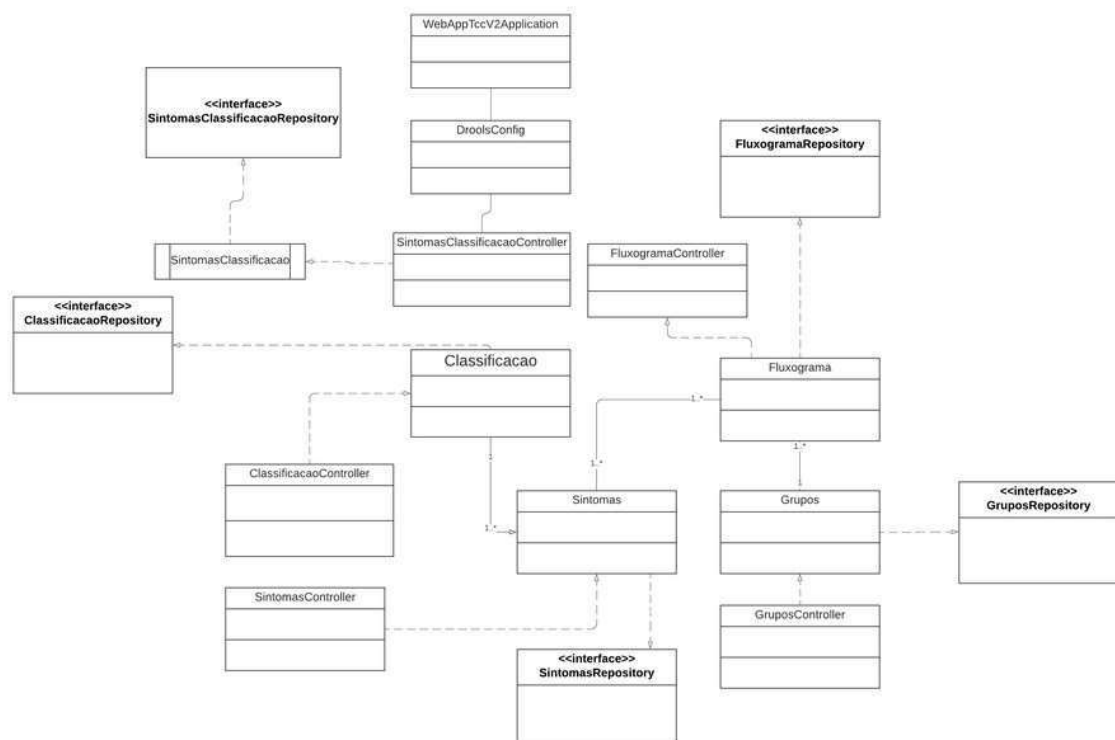
“uma família de notações gráficas, apoiada por um metamodelo único, que ajuda na descrição e no projeto de sistemas de software, particularmente daqueles construídos utilizando o estilo orientado a objetos.(FOLERCA, 2005)

3.1.3.1 Diagrama de Classes

De acordo com Fowler (2005), os diagramas de classe representam os tipos de objetos (classes) existentes no sistema, incluindo os relacionamentos entre estes objetos.

A Figura 5 mostra o diagrama de classes, simplificado, do SITPa. Ressalta-se que no diagrama estão apresentadas as classes básicas para o funcionamento do sistema.

Figura 5 – Diagrama de classes simplificado do SITPa.



Fonte: Elaborada pelo autor

3.1.3.2 Diagrama de casos de uso

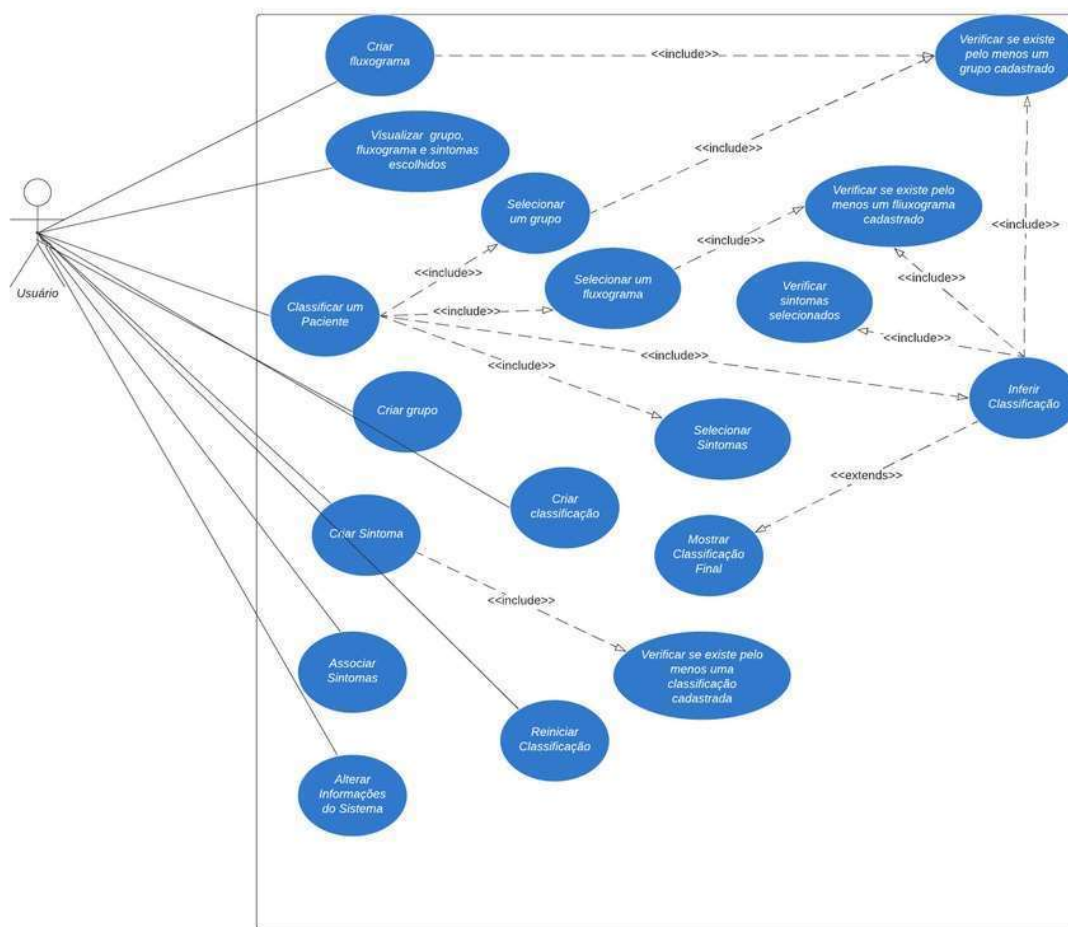
Segundo Larman:

“Casos de uso são narrativas textuais que descrevem as interações dos usuários com um sistema na realização de uma tarefa, a fim de alcançar um determinado objetivo (LARMAN, 2007).

O diagrama de casos de uso fornece uma narrativa sobre como o sistema será utilizado, e é fundamental para a detecção dos requisitos funcionais do sistema (FOWLER, 2005).

Na Figura 6 são mostrados os casos de uso básicos do SITPa.

Figura 6 – Diagrama de casos de usos do SITPa.



Fonte: Elaborado pelo autor

3.2 Construção da Base de Conhecimento

A base de conhecimento foi desenvolvida usando a linguagem JAVA juntamente com a linguagem DRL (*Drools Rule Language*) utilizada pelo Sistema de gerenciamento de regras de negócio, *JBoss Drools*. A seguir serão dados mais detalhes acerca das ferramentas utilizadas.

3.2.1 Linguagem Java

Java é uma linguagem de programação adequada para aplicações baseadas na Internet, Intranets, bem como aplicações Stand-alone (CAMPIONE et. al., 1998). A linguagem também é muito poderosa e possui recursos suficientes para a construção de uma variedade de aplicativos.

Para a utilização da linguagem *JAVA* é necessário possuir um pacote chamado Java Development Kit (JDK). Este kit permite que o desenvolvedor possa utilizar diversas ferramentas, entre elas, duas merecem destaque: O compilador e o interpretador.

O compilador chamado *javac* lê os arquivos fonte *.java* e gera os arquivos de classe *.class* em formato de *bytecodes*. O interpretador *java* então é utilizado para executar as aplicações em *Java*, ele é o responsável por interpretar todos os *bytecodes* gerados pelo *javac*.

3.2.2 JBoss Drools

Programadores estão acostumados a usarem uma abordagem processual quando se trata da implementação de requisitos de negócios. A consequência é que a implementação destas regras depende apenas da competência e do entendimento que o desenvolvedor de software possui das regras de negócios que ele deve implementar.

A programação orientada a objetos pode ser vista como “a solução” para a implementação e a agilidade dos negócios, e muitos desenvolvedores utilizam esse tipo de desenvolvimento. No final, a abordagem orientada a objetos termina com o “código Spaghetti” clássico, onde o autor escreve da forma desejada, sem seguir nenhuma regra, o que deixa o código bagunçado e de difícil leitura, além de confusos e mais propensos a erros.

Para evitar essa prática, foi adotada uma nova abordagem chamada “programação declarativa”. Este tipo de programação permite que o programador modele um problema através de assertivas sobre o domínio do problema, definindo o relacionamento lógico existente entre todas as partes, e não descrevendo como seus procedimentos funcionam.

O *Drools* é um sistema de gerenciamento de regras de negócios que possui um motor de regras de negócio principal, um aplicativo web para o gerenciamento das regras, suporte completo em tempo de execução para Modelos de Notação e Decisão (DMN) e um plug-in IDE do Eclipse para o desenvolvimento principal. (Drools Manual Reference, 2021).

Drools é um software de código aberto, lançado sob a Licença de Software Apache, é escrito 100% em java puro, pode ser executado em qualquer *JVM* e está disponível no repositório *Maven Central*. (DROOLS DOCUMENTATION, 2021).

3.2.2.1 Drools Engine

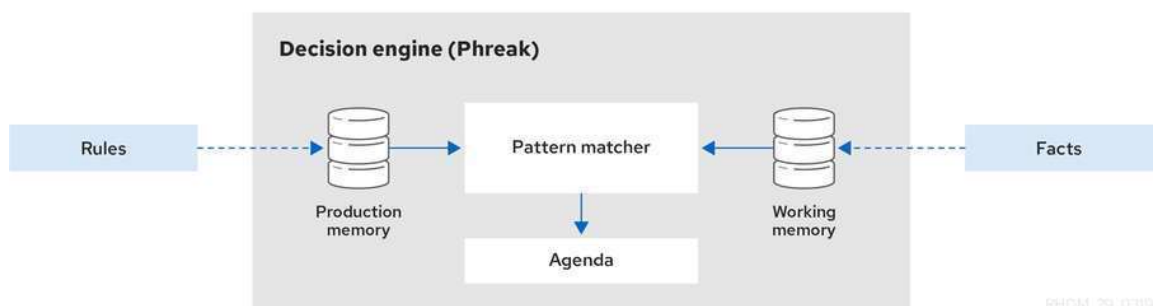
O *Drools Engine* é o motor de regras principal do *Drools*. Ele armazena, processa e avalia os dados para só então executar as regras de negócio que foram definidas. A função básica desse motor é verificar os dados ou fatos recebidos e mediante as condições das regras, determinar como estas serão executadas.

O *Drools* opera basicamente com cinco componentes principais:

- Regras - São as regras de negócios definidas pelo programador e estas, devem conter as condições as quais irão acioná-las, bem como a ação a ser realizada quando uma regra for ativada.
- Fatos - São os dados de entrada para o *Drools*, são eles que acionarão as regras correspondentes aos dados de entrada.
- Memória de produção - Repositório onde as regras estão armazenadas.
- Memória de trabalho - Repositório onde os fatos são armazenados, testados e manipulados.
- Agenda - Local onde as regras ativadas são colocadas e classificadas para serem executadas.

O funcionamento dá-se da seguinte maneira: Quando o usuário passa a informação relacionada a uma regra, essa informação é colocada na memória de trabalho do *Drools* na forma de fatos, o mecanismo então busca nas regras que estão armazenadas na memória de produção se alguma regra corresponde ao fato informado, em caso positivo o *Drools* ativa e registra a regra na agenda para então classificar todas as regras e prepará-las para execução.

Figura 7 – Drools Engine



RHDM_29_0319

Quando novos dados de regras entram na memória de trabalho da *engine* do Drools, essas regras podem se tornar coincidentes ficando assim elegíveis para a sua execução, dessa forma, uma ação na memória de trabalho pode resultar em múltiplas execuções de regras que estão elegíveis para a execução.

De acordo com a *Drools Documentation* (2021), quando uma regra é coincidente, a própria *engine* cria uma instância de ativação fazendo referência às regras e fatos coincidentes, e então adiciona sua ativação na agenda da *engine*, que controla a ordem de execução das ativações a fim de evitar possíveis conflitos.

Após o primeiro disparo de regras na aplicação, esses ciclos se repetem em duas fases: validação da agenda e ações da memória de trabalho. Para a primeira, a *engine* do Drools seleciona todas as regras que podem ser executadas e registra na agenda, depois move para a memória de trabalho e então as ações consequentes das regras serão realizadas. No entanto, se não houver regras executáveis, o ciclo de execução é encerrado.

Para ações da memória de trabalho, a *engine* do Drools realiza as ações consequentes das regras para todas as regras que já foram ativadas previamente na agenda. Após todas as ações serem completadas, ou após um novo disparo de regras, do processo principal da aplicação Java, a *engine* do Drools retorna para a fase de avaliação da agenda em procura de reavaliar as regras registradas.

3.2.2.2 Métodos do JBoss Drools na construção da base de conhecimento

Além da linguagem nativa do Drools, *.drl*, para a construção da base de conhecimento foram utilizadas algumas funcionalidades (métodos) do JBoss Drools tais como:

- *KieSession* - A *KieSession* armazena e executa os dados em tempo de execução. Ela é criada a partir de um *KieBase* ou diretamente de um *KieContainer*, se esse, foi definido no arquivo *kmodule.xml*.
- *KieServices* - O *KieServices* é um mecanismo que utiliza o padrão de projetos *singleton*, este padrão garante a existência de apenas uma instância de uma classe, mantendo um ponto global de acesso ao seu objeto, ele também age como um *hub*, dando acesso para serviços que são fornecidos pelo *Kie*.
- *KieContainer* - Para compreender o que é o *KieContainer*, precisamos entender o que são o *KieBases* e o *KieModules*.

O *KieBase* é uma *interface* que funciona como um repositório de todas as definições de conhecimento do sistema. Nele estão contidas as regras, processos, e funções. O *KieBase* em si não contém dados de tempo de execução; em vez disso, são criadas sessões a partir do *KieBase*, nas quais os dados podem ser inseridos e as instâncias dos processos iniciados.

O *KieModules* é uma *interface* que funciona como um contêiner com todos os recursos necessários para definir um conjunto de *KieBases*, como por exemplo um *pom.xml*, que por sua vez define seu *ReleaseId*: sendo este um arquivo *kmodule.xml*, que declara os nomes e configurações do *KieBases*, juntamente com todo o *KieSession*. O *KieSession* pode ser criado a partir dos *KieBases* e todos os outros arquivos necessários para construir os próprios *KieBases*.

Sabendo os conceitos de *KieBases* e *KieModules* podemos dizer que o *KieContainer* é uma interface que funciona como um contêiner para todas as *KieBases* de um determinado *KieModule*.

A função básica do *Drools* é relacionar os dados às regras de negócios e determinar se e como essas regras serão executadas. Para garantir que os dados sejam aplicados às regras apropriadas, o *Drools* faz inferências com base no conhecimento existente e então executa as ações com base nas informações inseridas. As regras escritas na linguagem *DRL* (*Drools Rule Language*), estão em um arquivo com extensão *.drl*.

Em um arquivo deste tipo podem conter uma ou mais regras que definem as *rules conditions* (*when*) e as *actions* (*then*).

Figura 8 – Regra de negócio em linguagem *.drl*

```
rule "Verificar Idade Adulta"
when
    $p : Pessoa (idade >= 18)
then
    insert(new IsAdult($p))
end
```

Fonte: Elaborada pelo autor

Analisando a regra da Figura 8, o *Drools* irá verificar se uma pessoa é adulta e então executa a ação especificada na regra (*then* é a consequência). Toda pessoa com 18 anos ou mais tem uma instância de *IsAdult* inserida na memória de trabalho.

Após o estudo do *Drools*, algumas regras foram modeladas e implementadas baseadas no protocolo de triagem *Manchester*, consultado em (MACKWAY-JONES, 2010). Vale salientar que essas regras serão utilizadas exclusivamente para fins de teste e validação do mecanismo de inferências do sistema, pois como dito anteriormente neste trabalho, as regras existentes no protocolo supracitado já são consolidadas e foram validadas há muitos anos.

Com a base de conhecimento montada, o passo seguinte foi construir a aplicação para classificar os pacientes a fim de tornar mais eficiente o processo de triagem médica.

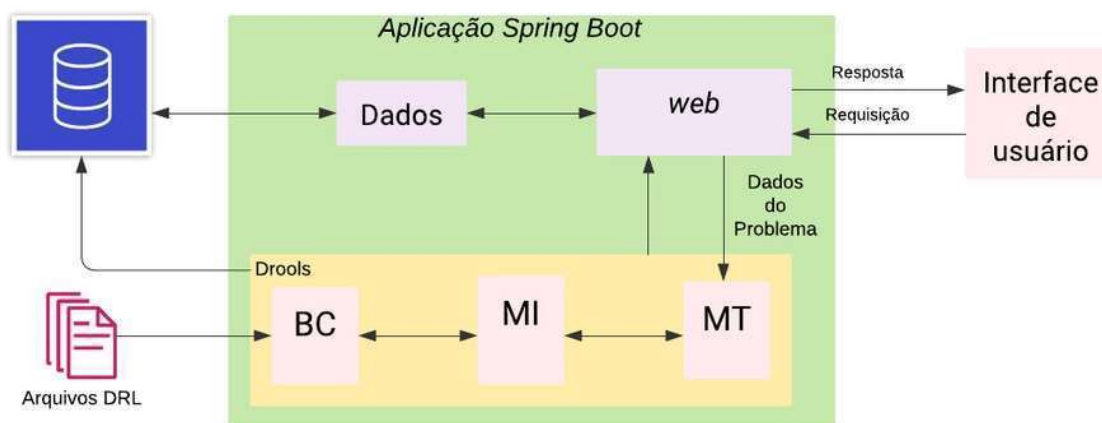
3.3 Implementação do Sistema

Para a implementação do sistema, foi utilizada a linguagem *Java* para a escrever o *back end* em um projeto *Spring Boot* criado no *Eclipse*, onde foi necessário o uso do *JBoss Drools* para a construção da base de conhecimento do sistema, bem como o mecanismo de inferências.

O usuário interagirá com o sistema através de uma interface *web*. Esta, será responsável por captar os dados informados, colocar na memória de trabalho para que o mecanismo de inferências busque na base de conhecimento do sistema, que já contém as regras definidas, quais regras serão disparadas. Após isso, o *drools* envia o resultado do processamento dos dados para o usuário por meio da interface *web*.

Segue abaixo um diagrama que representa uma arquitetura simplificada do SITPa.

Figura 9 – Arquitetura simplificada do SITPa.



Fonte: Elaborado pelo autor

Na camada de visualização foi utilizado o *thymeleaf* e no *front end*, o *Materialize* foi o escolhido. Como banco de dados, foi utilizado o *Postgresql* e para controle de versão foi utilizado o *github*.

Para a utilização do SITPa na triagem de pacientes nas unidades de saúde, é necessário modelar as regras da unidade de saúde em questão, com o propósito de construir a base de conhecimentos que o sistema irá utilizar para realizar as classificações de risco de forma automatizada.

Uma vez identificado o profissional que possui o conhecimento necessário para o funcionamento do SITPa, um engenheiro de conhecimento através de técnicas de aquisição de conhecimento, obterá as informações necessárias para modelar todas as regras que serão utilizadas na base de conhecimento do sistema.

Com o conhecimento necessário já colocado na BC do sistema, um usuário designado pela instituição que esta usando o sistema, deverá fazer o cadastro das informações necessárias para a implementação do protocolo de risco a ser utilizado pelo sistema. Vale salientar que os passos listados abaixo devem ser seguidos sequencialmente.

O primeiro passo é o cadastro de todos os grupos onde serão agrupados os fluxogramas contidos no protocolo utilizado pela unidade de saúde.

Após a criação dos grupos, será necessário cadastrar os fluxogramas que conterão os sintomas, que serão utilizados posteriormente como fatos na hora de fazer a classificação de um paciente. É importante salientar que não é possível a criação de um fluxograma sem que pelo menos um grupo tenha sido criado, uma vez que um fluxograma deve fazer parte de um grupo.

Em seguida, o profissional deverá cadastrar as classificações contidas no protocolo a ser utilizado pelo SITPa. A classificação deverá conter um nome, um grau de prioridade, representado por um número inteiro e uma cor associada a ela.

O próximo passo é criar os sintomas que serão utilizados para compor os fluxogramas. Não é possível criar um sintoma sem que tenha pelo menos uma classificação criada, uma vez que, um sintoma deve possuir uma classificação que indique o seu grau de prioridade dentre os outros vários sintomas existentes.

Com todos os grupos, fluxogramas, classificações e sintomas cadastrados, é necessário associar quais sintomas compõem cada fluxograma presente no protocolo utilizado, e feito isso, o sistema estará pronto para uso.

Neste sistema o profissional da saúde, através das interfaces de comunicação com o sistema, poderá escolher dentre os vários fluxogramas, que se encontram separados por grupos, aquele que é mais indicado em relação aos sintomas os quais estão sendo apresentados pelo paciente.

Partindo destes sintomas o sistema fará a classificação de risco de acordo com o que foi informado pelo usuário, baseado no protocolo utilizado no hospital onde a solução está sendo usada. Para isso, fará o cruzamento dos dados informados pelo usuário, com os já existentes na sua base de conhecimento através do *JBoss Drools*, e a partir desta classificação, será informado o local mais adequado para o atendimento. Este local pode ser o próprio hospital ou alguma unidade de saúde primária, dependendo da classificação de risco inferida pelo sistema.

Ao final da classificação, o sistema mostrará ao usuário um resumo contendo os dados informados por ele, assim como as informações inferidas pelo sistema. Este resumo será composto por: Classificação de risco, o grupo ao qual este fluxograma pertence, o sintoma mais grave apresentado, o local, o tipo e o tempo de atendimento.

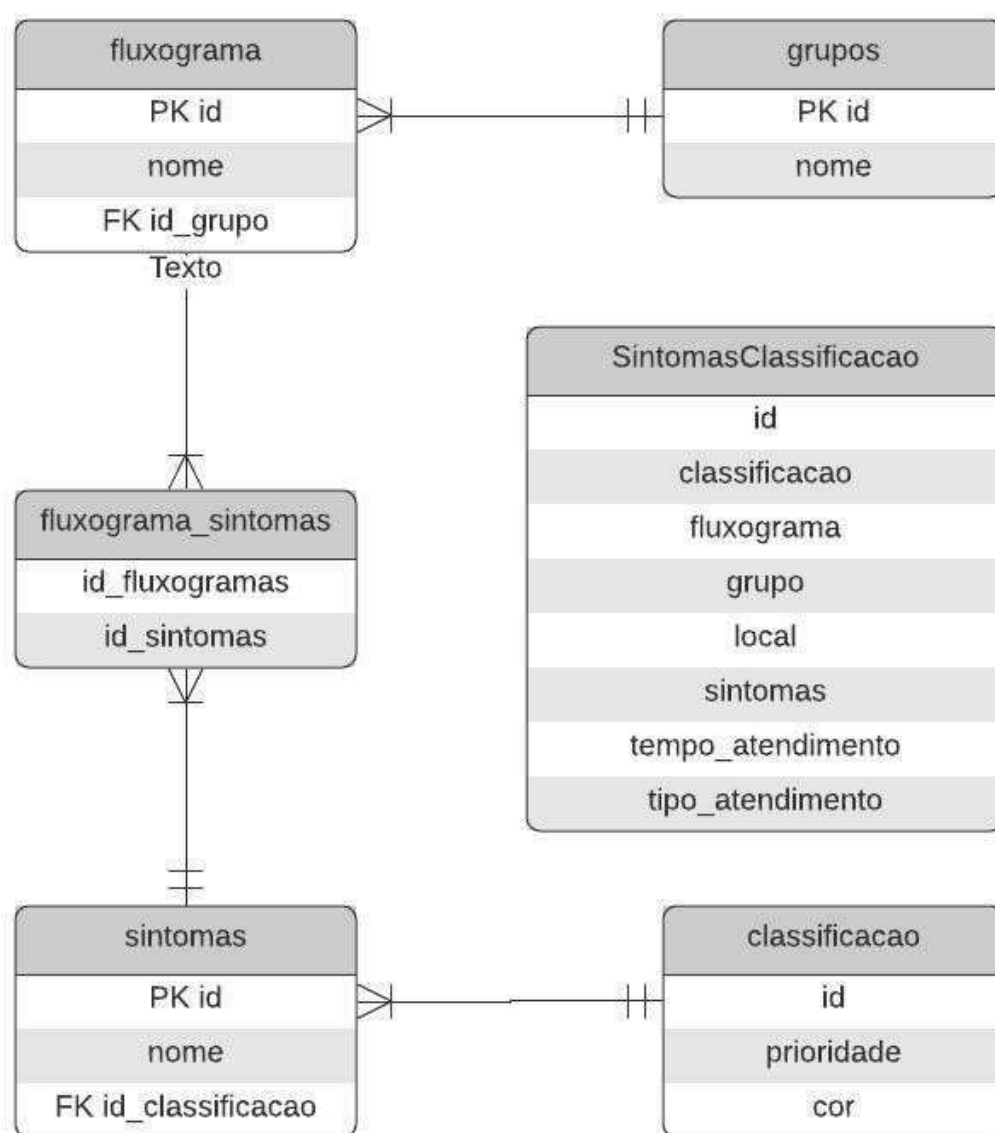
Única e exclusivamente para fins de teste, serão utilizados alguns fluxogramas do Sistema de Triagem *Manchester*, diante do fato que este, como já mencionado, é um sistema já validado e com regras consolidadas há muitos anos.

Uma vez que, o hospital que fará uso do sistema disponibilize os fluxogramas do protocolo utilizado para triagem dos pacientes, com já dito anteriormente, estes serão separados em grupos. Após isso, sintomas de classificações iguais também serão separados.

Este sistema conta com um banco de dados implementado no *SGBD Postgresql*, onde serão armazenados todos os fluxogramas, os sintomas relacionados, os grupos de fluxogramas, e suas classificações.

Os resultados das classificações feitas pelo sistema, também serão armazenados para fins estatísticos e trabalhos futuros que necessitem de dados históricos.

Figura 10 – Diagrama Entidade-Relacionamento da aplicação



Fonte: Elaborada pelo autor

Inicialmente o banco de dados é composto por seis tabelas para inserção de flu-

xogramas, grupos, sintomas, etc. no sistema. Ressalta-se que, conforme o surgimento de novas necessidades, mais entidades poderão ser adicionadas ao banco de dados da aplicação.

3.3.1 Back End

O *back end* do SITPa foi desenvolvido na linguagem Java e o *framework Spring Boot* com *Spring MVC*.

O primeiro passo para a construção, foi a criação das classes que vão servir como modelos dos objetos no sistema, na arquitetura MVC: os *Models*.

Em seguida, foram criados os *Controllers*, sendo estes as classes que irão manipular os dados passados pelo usuário através da interface do sistema. Com todos os *models* e *controllers* criados, foram implementadas todas as telas do sistema, as *Views*, as quais o usuário irá interagir diretamente.

Posteriormente, foi necessária a criação da classe de configuração do *Drools*, contendo todos os métodos para a criação dos *containers* e repositórios necessários para o correto funcionamento do mecanismo de inferências do sistema. Essa classe contém os seguintes métodos: *getKieFileSystem*, *getKieContainer*, *getKieRepository* e o *getKieSession*.

Para entender como esses métodos funcionam é necessário que se entenda os conceitos de *KieModule* e *KieBase*.

KieModule - É um *container* com todos os recursos necessários para definir um conjunto de *KieBases*.

KieBase - É um repositório de todas as definições do conhecimento do sistema.

Figura 11 – Classe de configuração do *Drools*

```
@Configuration
public class DroolsConfig {

    private KieServices kieServices = KieServices.Factory.get();

    private KieFileSystem getKieFileSystem() throws IOException {

        KieFileSystem kieFileSystem = kieServices.newKieFileSystem();
        kieFileSystem.write(ResourceFactory.newClassPathResource("regras.drl"));
        return kieFileSystem;
    }

    @Bean
    public KieContainer getKieContainer() throws IOException {
        System.out.println("Container criado...");
        getKieRepository();
        KieBuilder kb = kieServices.newKieBuilder(getKieFileSystem());
        kb.buildAll();
        KieModule kieModule = kb.getKieModule();
        KieContainer kContainer = kieServices.newKieContainer(kieModule.getReleaseId());
        return kContainer;
    }

    private void getKieRepository() {
        final KieRepository kieRepository = kieServices.getRepository();
        kieRepository.addKieModule(new KieModule() {
            public ReleaseId getReleaseId() {
                return kieRepository.getDefaultReleaseId();
            }
        });
    }

    @Bean
    public KieSession getKieSession() throws IOException {
        System.out.println("Sessão criada...");
        return getKieContainer().newKieSession();
    }
}
```

Fonte: Elaborado pelo autor

O primeiro método, *getKieFileSystem()*, tem a função de escrever no *FileSystem* todas as regras que foram implementadas no arquivo *.drl*. Essas informações são necessárias para que sejam definidos os recursos que irão compor o *KieModule* necessário para a criação do *KieContainer*, que é o local onde irá ficar armazenado todos os *KieBases* de um determinado *KieModule*.

O método *getKieRepository()* é onde estão todos os *KieModules* disponíveis, e retorna o *releaseId* que irá identificar o *KieModule* que será usado para a criação do *KieContainer*.

Com todos esses métodos implementados, o passo seguinte é criar uma nova sessão e passar o *container* criado anteriormente, dessa forma, o sistema estará pronto para inferir as classificações de risco para o qual ele foi projetado.

A ação do *KieSession* é injetar os dados nos *controllers* da aplicação e consequentemente disparar a execução das regras contidas na base de conhecimento.

As anotações *@Bean* nos métodos *getKieContainer()* e *getKieSession()*, sinalizam para o *spring framework*, que esses objetos devem ser criados e deixados disponíveis para as outras classes usarem ele como dependência.

Com todos os *models*, *controllers* e a classe de configuração do *Drools* criadas, foi iniciada a implementação das telas que servirão de interface de comunicação entre o usuário e o sistema.

3.3.2 Front End

Para a interface do sistema, foram desenvolvidas algumas telas que auxiliam o usuário na hora de classificar um paciente, são telas simples, objetivas, e de fácil navegação, que não deixam margem de erro na hora do uso do sistema.

Partindo da ideia de tornar a triagem médica mais eficiente, as telas foram desenvolvidas através de princípios que garantissem rapidez e objetividade na hora do uso do sistema.

Todas as telas foram desenvolvidas na linguagem *HTML* e também foi usado o *Thymeleaf*, uma *template engine* que pode ser usada tanto em aplicações *web*, quanto nas *stand alone*.

Inicialmente foi criada a tela principal do SITPa, que contém atalhos para as funcionalidades disponíveis no sistema, são elas: criar grupo, criar fluxograma, criar classificação, associar sintomas, cadastrar sintoma, classificar paciente e verificar classificações salvas no banco de dados.

Após a criação da tela inicial, foram criadas as telas que serão utilizadas para cadastrar os fluxogramas contidos no protocolo utilizado pela unidade de saúde, as telas implementadas foram as seguintes:

- Criar um novo grupo

- Criar um novo fluxograma
- Criar uma nova classificação
- Cadastrar sintoma
- Associar sintomas a um fluxograma

Feito isso, foram criadas as telas para classificação de risco. No total são quatro telas:

- Classificar paciente.
- Escolha dos sintomas.
- Revisão dos dados informados.
- Classificação Final.

4 RESULTADOS E DISCUSSÃO

4.1 Resultados

No decorrer desta pesquisa, foram investigadas várias tecnologias para o desenvolvimento de uma aplicação que fosse capaz de raciocinar diante de um conjunto de dados armazenados em uma base de conhecimento.

Após uma revisão da literatura e a aquisição das ferramentas necessárias para o desenvolvimento da aplicação, foi iniciado o processo de desenvolvimento da solução, onde, inicialmente foi construída a base de conhecimento que o sistema utilizará para atingir os objetivos para os quais foi projetado.

Essa base de conhecimento foi composta por algumas regras, modeladas com base no protocolo *Manchester*, tendo em vista que não foi possível realizar um estudo de caso, devido ao distanciamento social como reflexo da pandemia de COVID-19, o que impossibilitou a implantação do sistema nos serviços hospitalares, uma vez que faz-se necessário visitas à esses locais, necessidade de entrevistar profissionais, aplicação de questionários, etc.

Vale ressaltar, que a base de conhecimento do SITPa, quando estiver em produção, será construída com base no protocolo utilizado pela unidade de saúde que estiver fazendo uso do sistema, pois, como dito anteriormente, as regras são totalmente adaptáveis e o SITPa não faz uso permanente de nenhum dos protocolos citados neste trabalho. Foram utilizadas algumas regras do protocolo *Manchester* apenas para fins de validação do sistema.

Após a construção da base de conhecimento, foi desenvolvido o *back end* da aplicação, composto pelas classes já mostradas no diagrama de classes da Figura 5, na seção 3.1.3.1. O *back end* desenvolvido neste trabalho possibilita a manipulação dos dados informados pelo usuário e com base nestes dados o sistema é capaz de inferir, com precisão, uma classificação de risco a um paciente.

Além da classificação de risco, o SITPa é capaz de informar ao usuário, qual o local correto para atendimento do paciente classificado, bem como o tempo de espera para o atendimento, o tipo de atendimento, entre outros dados de menor relevância.

Os dados necessários para o sistema realizar as tarefas para o qual foi projetado, são obtidos através de uma interface *web* desenvolvida de maneira a facilitar a navegação, não só na hora de classificar um paciente, mas em qualquer operação disponibilizada pelo SITPa.

Ao entrar no sistema, é apresentada a tela inicial do SITPa. Esta tela possui uma *navbar* onde estão dispostas as funcionalidades oferecidas pelo sistema.

Figura 12 – Tela inicial do SITPa.



Fonte: Elaborada pelo autor

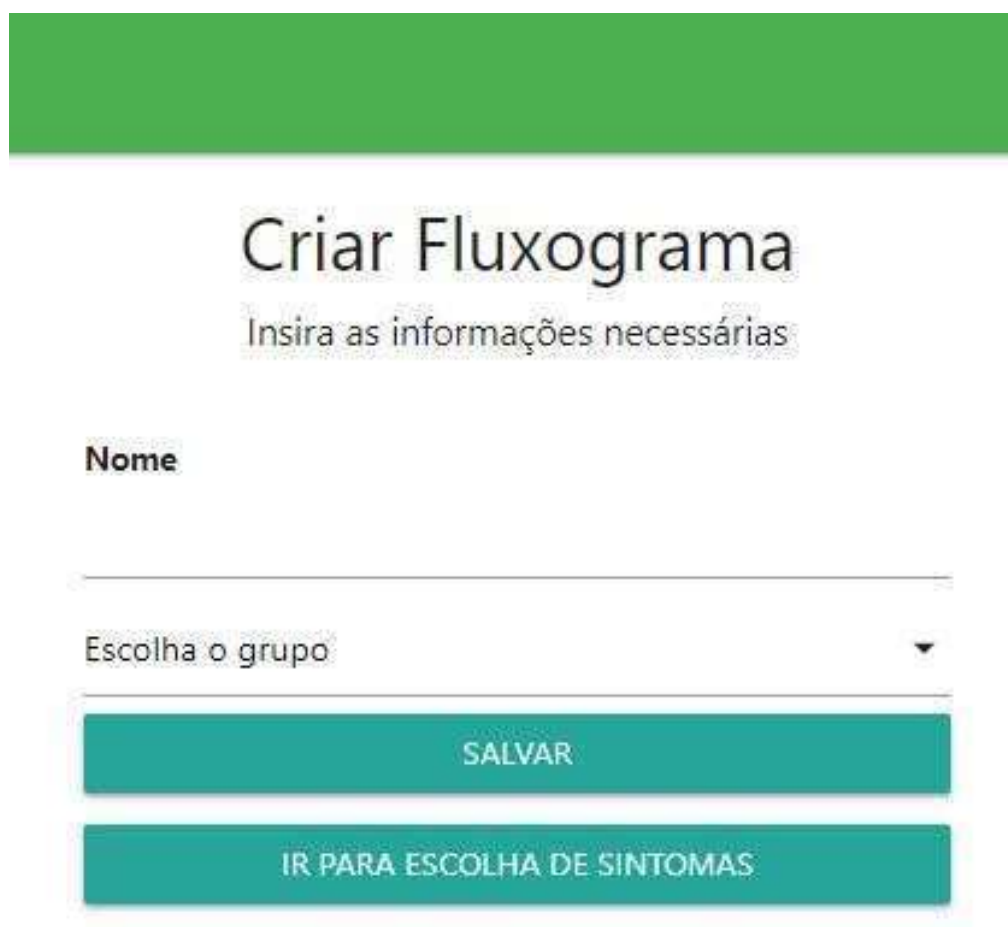
No item “Criar grupo”, é possível a criação dos grupos que serão relacionados com os fluxogramas do protocolo utilizado pela instituição que estiver usando o sistema. Esta tela possui apenas um *input* do tipo texto, para ser colocado o nome do grupo e um botão para salvá-lo no banco de dados da aplicação.

Figura 13 – Tela para criação de um novo grupo

Fonte: Elaborada pelo autor

Voltando a tela inicial, ao clicar no item “Criar fluxograma”, o usuário será redirecionado para a página que possibilitará a inclusão de um fluxograma no sistema. Esta tela é composta por um input do tipo texto, para colocar o nome do fluxograma que será salvo na base de dados do sistema, e o grupo ao qual este fluxograma será associado. Vale ressaltar que a criação de um fluxograma só é possível se houver pelo menos um grupo cadastrado no sistema, pois, um fluxograma, de acordo com a modelagem do sistema, deve obrigatoriamente ser associado a um grupo.

Figura 14 – Tela para criação de novo fluxogramas

A imagem mostra a interface de usuário para a criação de um novo fluxograma. No topo, há uma barra verde sólida. Abaixo, o título "Criar Fluxograma" é exibido em uma fonte grande e escura, seguido pelo subtítulo "Insira as informações necessárias" em uma fonte menor e cinza. O formulário contém um campo de texto rotulado "Nome" e um menu suspenso rotulado "Escolha o grupo". Abaixo dos campos, há dois botões de ação: "SALVAR" e "IR PARA ESCOLHA DE SINTOMAS", ambos em uma cor verde-água.

Fonte: Elaborada pelo autor

Semelhante às telas anteriores, a interface para o cadastro de uma nova classificação no sistema, pode ser acessada clicando-se no item “criar classificação” na tela inicial do sistema. Esta tela, contém uma estrutura do tipo *select* proveniente do *Thymeleaf* onde é possível escolher a prioridade da classificação que está sendo criada, e uma cor.

Para facilitar o uso da aplicação, os dois campos desta tela serão preenchidos por dados inseridos anteriormente no banco de dados da aplicação, cabendo ao usuário apenas selecioná-los.

Figura 15 – Tela para criar uma nova classificação

Criar Classificação
Insira as informações necessárias

Nível de prioridade
escolha o nível de prioridade ▼

Cor
escolha a cor ▼

SALVAR

Os níveis de prioridade são:

- 1: Emergente
- 2: Muito Urgente
- 3: Urgente
- 4: Pouco urgente
- 5: Não urgente

Fonte: Elaborada pelo autor

O item “Cadastrar sintoma” presente na tela inicial do sistema, leva o usuário à interface onde é possível criar um novo sintoma, para posteriormente ser associado aos fluxogramas que precisam dele. Esta tela possui um campo para informar o nome do sintoma a ser cadastrado e outro campo onde o usuário irá escolher a classificação a qual o sintoma pertence.

Figura 16 – Tela de cadastro de sintomas



Cadastrar sintoma

Insira as informações necessárias

Nome

Escolha a classificação

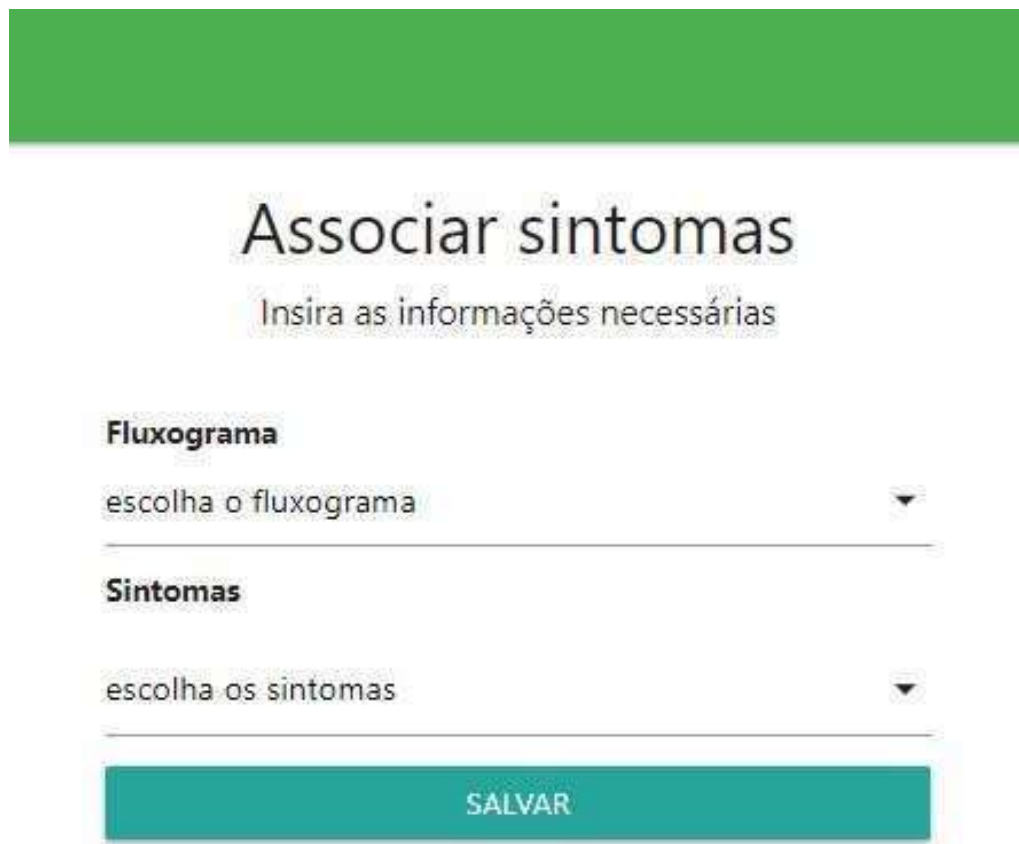
SALVAR

Fonte: Elaborada pelo autor

A tela para associar sintomas aos fluxogramas existentes no sistema, pode ser acessada através do item “Associar sintomas” na *navbar* da página inicial do SITPa.

Esta tela contém dois campos: Fluxograma e Sintomas. Estes, devem ter sido cadastrados anteriormente, visto que só é possível associar sintomas a um fluxograma se cadastrados previamente no sistema.

Figura 17 – Tela para associar sintomas a um fluxograma



Associar sintomas

Insira as informações necessárias

Fluxograma

escolha o fluxograma

Sintomas

escolha os sintomas

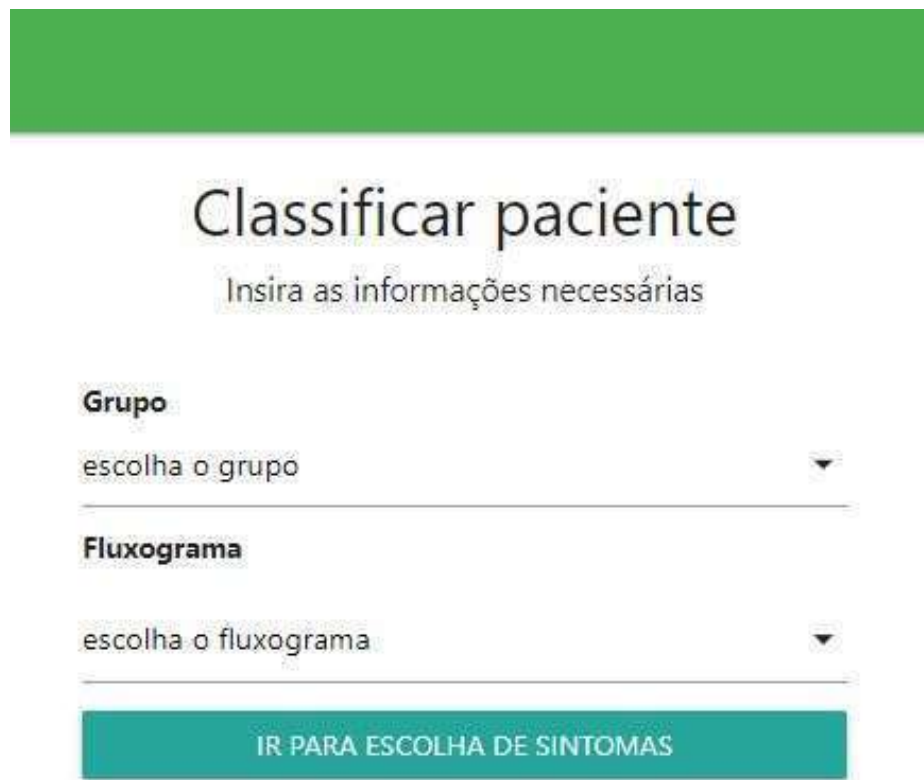
SALVAR

Fonte: Elaborada pelo autor

Para classificar um paciente, o sistema deve possuir todos os dados exigidos para esta finalidade. Que são: o grupo, o fluxograma e os sintomas.

A classificação de um paciente pode ser feita clicando no item “Classificar paciente” na tela inicial do sistema. Ao clicar neste item, o usuário será redirecionado para a tela da figura abaixo:

Figura 18 – Tela para classificar um paciente



Classificar paciente

Insira as informações necessárias

Grupo

escolha o grupo

Fluxograma

escolha o fluxograma

IR PARA ESCOLHA DE SINTOMAS

Fonte: Elaborada pelo autor

O primeiro passo para a classificação é feito através da tela da Figura 16.

Primeiramente o usuário deve escolher um grupo, feito isso, todos os fluxogramas relacionados ao grupo escolhido, serão apresentados no campo “Fluxograma”, cabendo ao usuário apenas escolher o mais adequado à situação.

Ao clicar no botão “Ir para escolha de sintomas”, o usuário será redirecionado para a interface de escolha dos sintomas relacionados ao fluxograma selecionado. A tela é a que segue logo abaixo:

Figura 19 – Tela para escolha de sintomas

Agressão

Selecione os sintomas apresentados pelo paciente

☐ Obstrução das vias aéreas	☐ Respiração inadequada	☐ Choque
☐ Hemorragia Exaguinante	☐ Mecanismo de trauma significativo	☐ Dispnéia aguda
☐ Hemorragia maior incontrolável	☐ Alteração súbita da consciência	☐ Déficit neurológico agudo
☐ Dor intensa	☐ Hemorragia menor incontrolável	☐ História de inconsciência
☐ Déficit neurológico novo	☐ Distúrbio de coagulação	☐ História discordante
☐ Dor moderada	☐ Edema	☐ Deformidade
☐ Dor leve recente	☐ Evento recente	

Fonte: Elaborada pelo autor

A tela da Figura 19 mostra os sintomas do fluxograma “Agressão”, presente no protocolo *Manchester*.

Se o usuário desejar, é possível voltar para as telas anteriores para uma possível correção na escolha. Caso todas as escolhas feitas estejam corretas, com os sintomas selecionados é possível ir para a finalização da classificação clicando no botão “Ir para classificação”, porém, antes de finalizar a classificação, o sistema mostra um resumo de tudo que o usuário escolheu até este ponto, caso tudo esteja correto, segue-se para a classificação final do paciente.

Figura 20 – Tela de resumo dos dados informados na figura 19

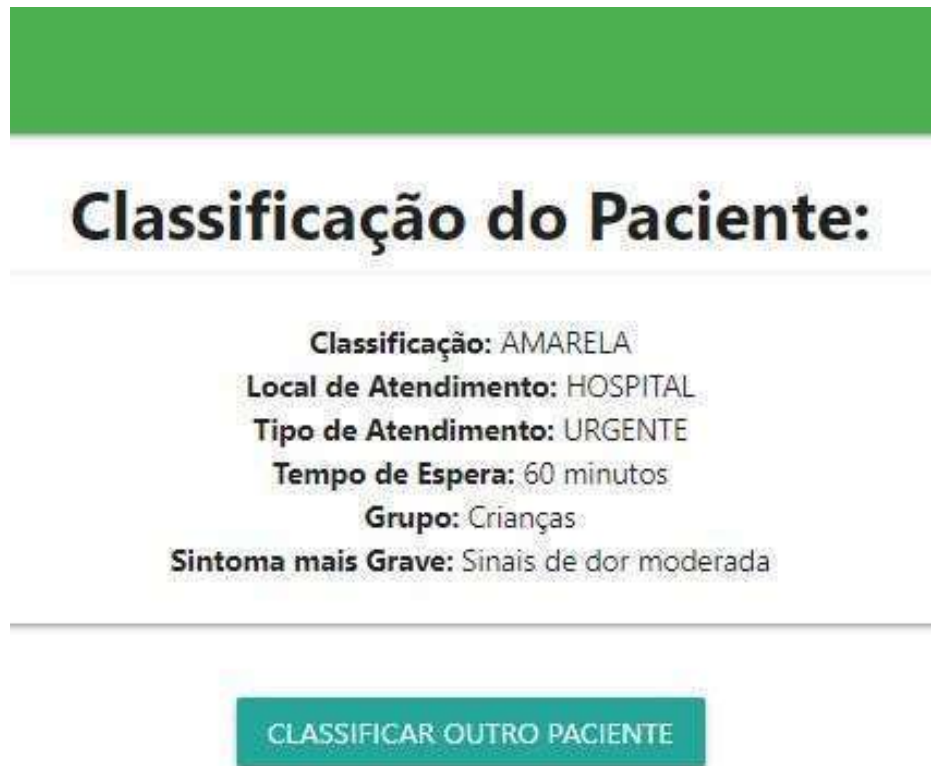
Revisão dos dados informados

Grupo: Crianças
Sintoma mais Grave: Obstrução das vias aéreas
Fluxograma escolhido: Mal estar em crianças

Fonte: Elaborada pelo autor

Se tudo estiver correto o usuário deve clicar no botão “Confirmar dados informados” e então a classificação do paciente é exibida na próxima tela.

Figura 21 – Tela de classificação final



A tela de classificação final do paciente apresenta um cabeçalho verde sólido. Abaixo dele, o título "Classificação do Paciente:" é exibido em uma fonte grande e preta. Seguem-se as informações de classificação em um formato de lista: "Classificação: AMARELA", "Local de Atendimento: HOSPITAL", "Tipo de Atendimento: URGENTE", "Tempo de Espera: 60 minutos", "Grupo: Crianças" e "Sintoma mais Grave: Sinais de dor moderada". No rodapé da tela, há um botão verde com o texto "CLASSIFICAR OUTRO PACIENTE".

Fonte: Elaborada pelo autor

Na tela da Figura 21 é possível visualizar o resultado da classificação final do paciente.

O último item da *navbar* da página inicial, redireciona o usuário para uma página que contém todas as classificações salvas no banco de dados.

Figura 22 – Classificações no banco de dados do SITPa



A tela de "Classificações salvas no Banco" possui um cabeçalho verde com o título e um botão "NOVA CLASSIFICAÇÃO" com um ícone de mais. Abaixo, há uma tabela com as seguintes colunas: Classificação, Local de Atendimento, Tipo de Atendimento, Espera (minutos), Grupo, Sintoma mais grave e Fluxograma. A tabela contém uma única entrada com os dados: VERMELHA, HOSPITAL, ATENDIMENTO IMEDIATO, 0, outros, Choque e convulsões. No rodapé da tabela, há uma barra de paginação que indica "Showing 1 to 1 of 1 entries" e botões "Previous" e "Next".

Classificação	Local de Atendimento	Tipo de Atendimento	Espera (minutos)	Grupo	Sintoma mais grave	Fluxograma
VERMELHA	HOSPITAL	ATENDIMENTO IMEDIATO	0	outros	Choque	convulsões

Fonte: Elaborada pelo autor

4.2 Discussão

Ao final do desenvolvimento do SITPa esperava-se que fosse possível a realização de um estudo de caso, objetivando a implantação do sistema em algum hospital da região para ser avaliado e validado pelos profissionais de saúde que o utilizariam. No entanto, devido à pandemia de COVID-19 e as restrições de distanciamento impostas por ela, não foi possível realizar o estudo.

O sistema desenvolvido tem por objetivo classificar um paciente de forma inteligente, no processo de triagem médica. Para isso, é necessário que a unidade de saúde possua um protocolo de classificação de risco, que será transformado em regras escritas em um arquivo .drl para a construção da base de conhecimento usada pela aplicação para alcançar os objetivos propostos.

Para a validação do sistema, foram modeladas algumas regras, utilizando o protocolo *Manchester*, e essas, foram inseridas na memória de produção do sistema.

Foram escolhidos alguns fluxogramas do protocolo mencionado, única e exclusivamente para validar as regras do sistema. Dentre eles estavam:

- Diabetes
- Gravidez
- Alergia
- Convulsões

Escolhidos o grupo e o fluxograma, campos obrigatórios para classificar um paciente, chegou a hora de escolher os sintomas para o fluxograma escolhido.

Ao escolher os sintomas, é feita uma verificação de prioridade dos mesmos. Esta verificação é obtida da seguinte maneira:

Ao selecionar os sintomas, estes são colocados em uma lista e então uma instância é criada no *controller* responsável pela manipulação desses dados. Cada sintoma é comparado individualmente com uma posição de um vetor constante. Esse vetor contém todos os sintomas presentes no protocolo de classificação utilizado.

Uma vez encontrado um sintoma, uma varredura em um *HashMap* é iniciada em busca do sintoma informado, ao ser encontrado, verifica-se o valor do *value* do sintoma, sendo esse o nível de prioridade.

Encontrado o sintoma o *value* é comparado com uma variável que armazena a maior prioridade (ao iniciar o sistema essa variável está com o valor 10), se esse valor for menor do que o conteúdo da variável, esse sintoma passa a ser o que tem maior prioridade, então esse processo é feito até que reste apenas o sintoma mais grave dentre os que foram informados.

Figura 23 – Método que verifica a prioridade dos sintomas informados

```

public void definirPrioridade() {
    String sintomaMaisGrave;
    int maiorPrioridade = 10;

    for (int i = 0; i <= 181; i++) {
        if (sintomasSelecionados.contains(sintomasVetor[i])) {
            if (sintomas_e_prioridades.get(sintomasVetor[i]).intValue() < maiorPrioridade) {
                maiorPrioridade = sintomas_e_prioridades.get(sintomasVetor[i]).intValue();
                sintomaMaisGrave = sintomasVetor[i];
                scm.setSintomas(sintomaMaisGrave);
                System.out.println("Sintoma String: " + scm.getSintomas());
            }
        }
    }
}

```

Fonte: Elaborada pelo autor

Foram implementadas algumas telas para testes, contendo os sintomas dos fluxogramas que foram escolhidos para validar este sistema. Estas telas não possuem um estilo e nem formatação definidas, visto que são telas temporárias que serão usadas exclusivamente para testes de validação do sistema.

A primeira tela foi a do fluxograma para diabetes.

Figura 24 – Tela teste - Diabetes

Fluxograma 13 - Diabetes

☐ Obstrução das vias aéreas	☐ Respiracao_inadequada	☒ Choque
☐ Crianca_nao_reativa	☐ Hipoglicemia	☐ Alteracao_subita_da_consciencia
☐ Hiperglicemia_com_cetose	☐ Crianca_quente	☐ Adulto_muito_quente
☒ Hipotermia	☐ Dor_intensa	☐ Hiperglicemia
☐ Adulto_quente	☒ Vomitos_persistentes	☐ Dor_moderada
☒ Febril	☐ Vomitos	☐ Dor_leve_recente
	☐ Evento_recente	

[Voltar para a tela de fluxograma para Sintomas Comuns](#)

Fonte: Elaborada pelo autor

A tela da Figura 24 simula a classificação de um paciente diabético que chegou apresentando os sintomas assinalados.

De acordo com Mackway-Jones (2010), as classificações para os sintomas escolhidos na tela supracitada são:

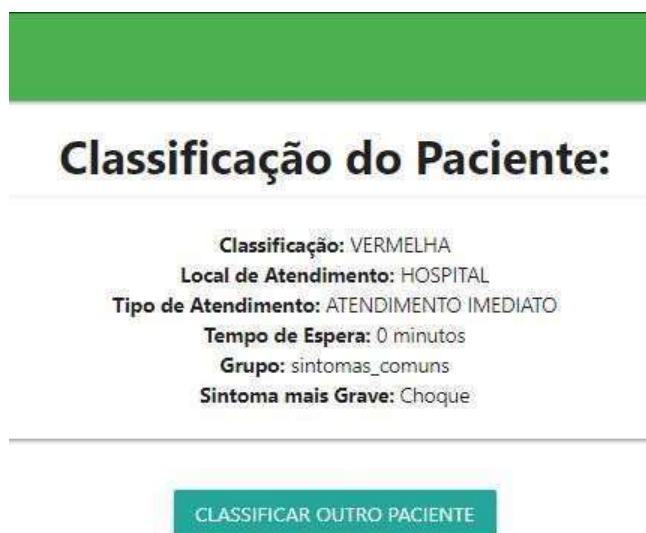
Mediante os sintomas informados na Figura 24, o suposto paciente seria classificado como “Emergente”, pois, o sintoma mais grave apresentado, foi “Choque”, que de acordo com o protocolo *Manchester*, possui a cor “vermelha”, ou seja, o paciente deveria ter atendimento imediato.

O resultado da classificação do suposto paciente é mostrado na Figura 25.

Tabela 1 – Classificação dos sintomas escolhidos na Figura 24

Sintoma	Classificação
Choque	Vermelha
Hipotermia	Laranja
Vômitos persistentes	Amarela
Febril	Verde

Fonte: Elaborada pelo autor

Figura 25 – Resultado da classificação do paciente da Figura 24

Fonte: Elaborada pelo autor

A regra que foi disparada para que essa classificação fosse possível, é mostrada na imagem da figura abaixo:

Figura 26 – Regra utilizada para inferir a classificação da Figura 25

```
rule "sintomas_comuns_Choque"
  when
    offerObject: SintomasClassificacaoModel(Grupo=="sintomas_comuns" && Sintomas=="Choque")
  then
    offerObject.setClassificacao("VERMELHA");
  end
```

Fonte: Elaborada pelo autor

Para os outros fluxogramas escolhidos para teste: gravidez, alergia e convulsões, o mesmo processo de simulação foi realizado e os resultados gerados são mostrados respectivamente nas figuras seguintes.

Figura 27 – Tela teste - Gravidez

Fluxograma 30 - Gravidez

☐ Obstrução das vias aéreas ☐ Choque ☐ Convulsionando ☐ História de convulsão ☐ Dor intensa ☐ História de trauma ☐ Sangramento vaginal	☐ Respiração inadequada ☐ Apresentação de partes fetais ☐ Trabalho de parto ☒ Sangramento vaginal intenso ☒ Dor abdominal ☐ História discordante ☐ Dor moderada ☒ Dor leve recente	☐ Hemorragia hexanguinante ☐ Prolapso de cordão umbilical ☐ Alteração súbita da consciência ☐ Sangramento vaginal mais de 20 semanas de gravidez ☐ Pressão arterial elevada ☐ Dor irradiada para o ombro ☐ Febre
---	--	---

[Ir para classificação](#)
[Voltar para a tela de fluxograma para Sintomas Comuns](#)

Fonte: Elaborada pelo autor

A tela da Figura 27 simula a classificação de uma paciente grávida que chegou apresentando os sintomas assinalados.

Ainda de acordo com Mackway-Jones (2010), as classificações para os sintomas escolhidos na figura 27 são:

Tabela 2 – Classificação dos sintomas escolhidos na Figura 27

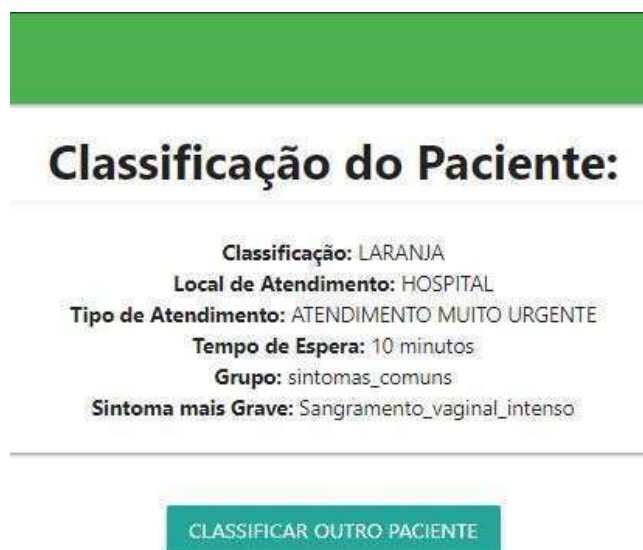
Sintoma	Classificação
Sangramento vaginal intenso	Laranja
Dor abdominal	Amarela
Dor leve recente	Verde

Fonte: Elaborada pelo autor

De acordo com os sintomas informados na Figura 27, a suposta paciente seria classificada como “Muito Urgente”, pois, o sintoma mais grave apresentado, foi “Sangramento vaginal intenso”, que de acordo com o protocolo *Manchester*, possui a cor “laranja” e isso significa que a paciente teria que ser atendida em no máximo, 10 minutos.

O resultado da classificação da suposta paciente é mostrado na Figura 28.

Figura 28 – Resultado da classificação da paciente da Figura 27



Classificação do Paciente:

Classificação: LARANJA
 Local de Atendimento: HOSPITAL
 Tipo de Atendimento: ATENDIMENTO MUITO URGENTE
 Tempo de Espera: 10 minutos
 Grupo: sintomas_comuns
 Sintoma mais Grave: Sangramento_vaginal_intenso

CLASSIFICAR OUTRO PACIENTE

Fonte: Elaborada pelo autor

A regra disparada para que essa classificação fosse possível, é mostrada na imagem abaixo:

Figura 29 – Regra utilizada para inferir a classificação da Figura 28

```
rule "sintomas_comuns Sangramento_vaginal_intenso"
  when
    offerObject: SintomasClassificacaoModel(Grupo=="sintomas_comuns" && Sintomas=="Sangramento_vaginal_intenso")
  then
    offerObject.setClassificacao("LARANJA");
  end
```

Fonte: Elaborada pelo autor

O próximo teste foi feito com o fluxograma “Alergia”.

Figura 30 – Tela teste - Alergia

Fluxograma 02 - Alergia

☐ Obstrução das vias aéreas	☐ Respiracao_inadequada	☐ Choque
☐ Crianca_nao_reativa	☐ Estridor	☐ Edema_facial
☐ Edema_de_lingua	☐ Historia_alergica_importante	☐ Frases_entrecortadas
☐ Pulso_anormal	☐ Alteracao_subita_da_consciencia	☐ Sat_O2_muito_baixa
☐ Dor_ou_coceira_intensas	☒ Sat_O2_baixa	☐ Erupcao_ou_vesiculas_disseminadas
☐ Dor_ou_coceira_moderadas	☒ Inflamacao_local	☐ Dor_ou_coceira_leve_recente
	☐ Eventos_recents	

[Ir para classificação](#)

[Voltar para a tela de outros fluxogramas](#)

Fonte: Elaborada pelo autor

A figura acima simula a classificação de um paciente que apresenta um quadro de alergia e saturação de oxigênio baixa além de uma inflamação local.

A classificação dos sintomas é mostrada na Tabela 3.

Tabela 3 – Classificação dos sintomas escolhidos na Figura 30

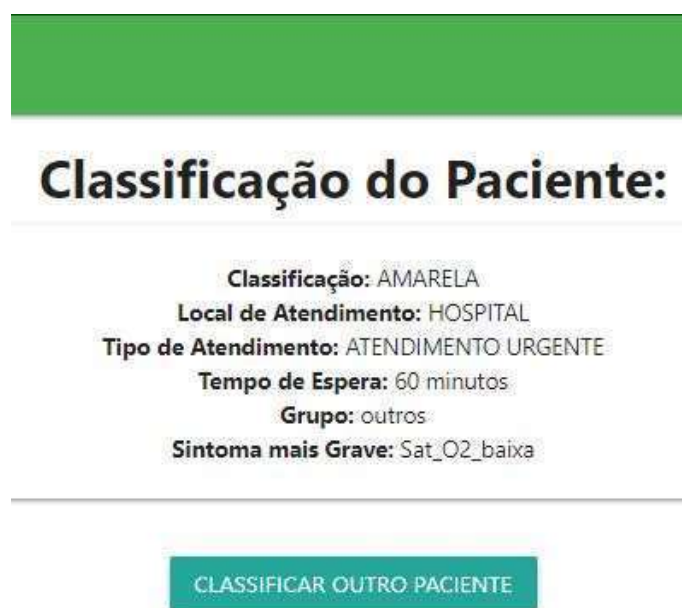
Sintomas	Classificação
Sat. O2 baixa	Amarela
Inflamação local	Verde

Fonte: Elaborada pelo autor

De acordo com os sintomas informados na Figura 30, o suposto paciente seria classificado como “Urgente”, pois, o sintoma mais grave apresentado, foi “Sat. O2 baixa”, que possui a cor “amarela” e isso significa que, de acordo com o protocolo, o paciente poderia esperar até 60 minutos para ser atendido.

O resultado da classificação do suposto paciente é mostrado na Figura 31.

Figura 31 – Resultado da classificação do paciente da Figura 30



Fonte: Elaborada pelo autor

A regra que foi disparada para que essa classificação fosse possível, é mostrada na imagem abaixo:

Figura 32 – Regra utilizada para inferir a classificação da Figura 30

```
rule "outros Sat_02_baixa"
  when
    offerObject: SintomasClassificacaoModel(Grupo=="outros" && Sintomas=="Sat_02_baixa")
  then
    offerObject.setClassificacao("AMARELA");
  end
```

Fonte: Elaborada pelo autor

O último teste foi feito com o fluxograma “Convulsões”, em que foi simulada a classificação de um paciente que apresentava apenas o sintoma “Cefaléia”.

Figura 33 – Tela teste - Convulsões

Fluxograma 08 - Convulsões

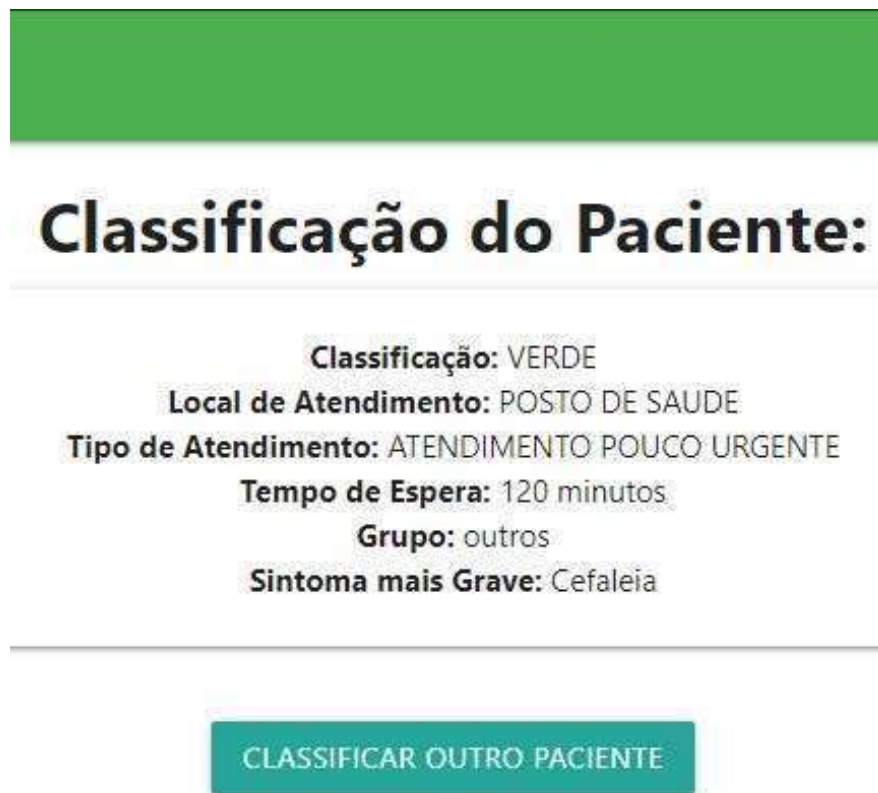
☐ Obstrução das vias aéreas	☐ Respiracao_inadequada	☐ Crianca_nao_reativa
☐ Choque	☐ Convulsionando	☐ Hipoglicemia
☐ Alteracao_subita_da_consciencia	☐ Deficit_neurologico_agudo	☐ Sinais_de_meningismo
☐ Historia_de_overdose_ou_envenenamento	☐ Erupcao_cultanea_fixa	☐ Purpura
☐ Crianca_quente	☐ Adulto_muito_quente	☐ Historia_de_trauma_cranioencefalico
☐ Historia_discordante	☐ Deficit_neurologico_novo	☐ Adulto_quente
☐ Historia_de_convulsao	☐ Febril	☒ Cefaleia
☐ Dor_leve_recente	☐ Evento_recente	

[Voltar para a tela de outros fluxogramas](#)

Fonte: Elaborada pelo autor

De acordo com o sintoma informado, o paciente seria classificado como “Pouco urgente”, podendo esperar até 120 minutos para ser atendido, ou então os profissionais encaminhariam este paciente para um posto de saúde, por exemplo, pois, dada a pouca gravidade do caso, não haveria problema algum em encaminhá-lo a alguma unidade primária de saúde. O resultado da classificação do suposto paciente é mostrado na Figura 34.

Figura 34 – Resultado da classificação do paciente da Figura 33



Fonte: Elaborada pelo autor

A regra utilizada para a classificação da Figura 33 pode ser vista na figura abaixo:

Figura 35 – Regra utilizada para inferir a classificação da Figura 34

```
rule "outros Febril"
  when
    offerObject: SintomasClassificacaoModel(Grupo=="outros" && Sintomas=="Febril")
  then
    offerObject.setClassificacao("VERDE");
  end
```

Fonte: Elaborada pelo autor

Após todos estes testes, foi verificado que o sistema está se comportando da forma esperada, pois ao relacionar os resultados obtidos, com os resultados dos fluxogramas presentes no protocolo usado para testes, o sistema utiliza seu conhecimento para inferir o resultado, que é exatamente o mesmo que seria obtido através da forma tradicional de classificação, porém, de forma muito mais rápida.

A seguir é mostrada a tela das classificações salvas no banco de dados da aplicação.

Figura 36 – Classificações de teste salvas no banco de dados

Classificação	Local de Atendimento	Tipo de Atendimento	Espera (minutos)	Grupo	Sintoma mais grave	Fluxograma
VERMELHA	HOSPITAL	ATENDIMENTO IMEDIATO	0	outros	Choque	asma
AMARELA	HOSPITAL	ATENDIMENTO URGENTE	50	sintomas_comuns	Hiperglicemia	diabetes
VERDE	POSTO DE SAUDE	ATENDIMENTO POUCO URGENTE	120	sintomas_comuns	Dor leve	queimaduras
VERMELHA	HOSPITAL	ATENDIMENTO IMEDIATO	0	sintomas_comuns	Choque	diabetes
LARANJA	HOSPITAL	ATENDIMENTO MUITO URGENTE	10	sintomas_comuns	Sangramento_vaginal_intenso	gravidez
AMARELA	HOSPITAL	ATENDIMENTO URGENTE	60	outros	Sat_O2_baixa	alergia
VERDE	POSTO DE SAUDE	ATENDIMENTO POUCO URGENTE	120	outros	Febril	convulsões
VERMELHA	HOSPITAL	ATENDIMENTO IMEDIATO	0	traumas	Hipoglicemia	traumaCranioencefalico

Showing 651 to 658 of 658 entries

Previous 1 ... 62 63 64 65 66 Next

Fonte: Elaborada pelo autor

5 CONCLUSÕES E TRABALHOS FUTUROS

5.1 Conclusões

A implementação de um sistema inteligente baseado em regras para triagem de pacientes desenvolvido neste trabalho, necessita de um estudo de caso futuro para verificar a usabilidade e a aceitação do sistema.

Tendo em vista que ao final do desenvolvimento da aplicação, não foi possível realizar um estudo de caso para verificar se o sistema se comportará adequadamente quando submetido a um uso constante. No entanto, foram feitos testes que comprovaram a confiabilidade do sistema, pois ao comparar resultados deduzidos pela aplicação com os resultados obtidos da maneira tradicional de classificação de risco, percebeu-se que os resultados foram exatamente os mesmos, o que significa que o mecanismo de inferências está processando os dados informados pelos usuários de forma correta e assertiva.

O fato de ter sido escolhido o *framework Spring Boot* para implementar o sistema, contribuiu de forma bastante positiva, uma vez que a integração com o *JBoss Drools*, utilizado para a implementação da parte “inteligente” do sistema, facilitou a codificação. A escolha da plataforma do desenvolvimento de uma aplicação *web*, também mostrou-se assertiva, uma vez que com o sistema pronto, é possível acessá-lo de qualquer lugar do mundo, de qualquer dispositivo em qualquer sistema operacional.

Por fim, os testes demonstraram que o sistema está se comportando como o planejado na fase inicial deste projeto. Embora não tenha sido feito um estudo de caso, testes com um protocolo validado e regras consolidadas, foram feitos e os resultados foram extremamente satisfatórios, uma vez que o sistema, através do seu conhecimento, foi capaz de inferir resultados exatamente como previsto pelo protocolo usado para validação.

5.2 Principais contribuições

Neste trabalho foi desenvolvido um sistema especialista baseado em regras para triagem de pacientes nas urgências e emergências dos hospitais.

Demonstrou-se ao longo da pesquisa que é possível a modelagem de regras que se adaptem à realidade da unidade de saúde que fará uso do sistema, no caso de protocolos já estabelecidos, o especialista é o próprio protocolo, pois como um protocolo possui toda uma documentação formal, é possível utilizar essas informações para modelar as regras que constituirão a base de conhecimento do sistema especialista.

Como base desse sistema foi construído um mecanismo de raciocínio utilizando o *Drools*, para a inferência de novos fatos tendo como base fatos já conhecidos. Acima desse mecanismo, foi desenvolvida uma interface *web* que pode ser acessada de qualquer dispositivo.

5.3 Trabalhos futuros

Propõe-se como possível continuação do presente trabalho, a possibilidade da implementação de uma interface que possibilite a inclusão, edição e deleção das regras que compõem a base de conhecimento do sistema, pois, dessa forma qualquer usuário que entenda do domínio do problema, será plenamente capaz de manipular as regras, mesmo sem ter conhecimento prévio em programação de computadores.

No sistema atual, as regras necessitam ser escritas diretamente no arquivo .drl, o que dificulta um pouco para aqueles que não possuem um conhecimento em programação de computadores, sendo assim, o desenvolvimento de uma interface que facilitasse essa manipulação das regras, seria de extrema relevância.

Com o uso deste sistema, dados de classificações serão armazenados em um banco de dados, o que possibilita que trabalhos futuros na área de *Business Intelligence* sejam desenvolvidos, pois como foi visto no início deste trabalho, um dos grandes fatores que contribuem para a superlotação das urgências é uma má gestão dos recursos. Dessa forma, um trabalho na área supracitada é de extrema relevância, uma vez que informações mais precisas podem ser extraídas a partir de dados históricos, e essas informações serão capazes de orientar os gestores quanto à um melhor administração dos recursos disponíveis na unidade de saúde.

REFERÊNCIAS

- AL. JUNIOR, A. C. dos S. et. Triagem Inteligente: O Uso da Internet das Coisas na Classificação dos Riscos nas Emergências. In: **Seminário Tecnologias Aplicadas a Educação e Saúde**. [S.l.: s.n.], 2019.
- BAELDUNG. **Introduction to Drools**. 2018. Disponível em: <https://www.baeldung.com/drools>. Acesso em: 20/04/2021.
- BAGGIO, MARIA APARECIDA; ERDMANN, ALACOQUE LORENZINI; DAL SASSO, GRACE TERESINHA MARCON. Cuidado Humano e Tecnologia na Enfermagem Contemporânea e Complexa. **Texto & Contexto Enfermagem**, v. 19, n. 2, 2010.
- BLUM, BRUCE I. Clinical Information Systems. **Springer-Verlag**, 1986.
- BULLARD, M. J. et al. Revisions to the Canadian emergency department triage and acuity scale (CTAS) adult guidelines. **Canadian Journal of Emergency Medicine**, v. 10, n. 2, p. 136 – 142, 2008.
- CAMPIONE, M.; WALRATH, K. **The Java Tutorial**: Object-Oriented Programming for the Internet. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1998.
- CENTRO REGIONAL DE ESTUDOS PARA O DESENVOLVIMENTO DA SOCIEDADE DA INFORMAÇÃO. **Pesquisa sobre o uso das tecnologias da informação e da comunicação no Brasil**. 2018. Disponível em: <https://www.cetic.br/tics/domicilios/2018/individuos/C16/>. Acesso em: 2020.
- COUTINHO, A. A. P.; CECÍLIO, L. C. de O.; MOTA, J. A. C. Classificação de risco em serviços de emergência: uma discussão da literatura sobre o Sistema de Triagem de Manchester. **Rev Med Minas Gerais**, v. 2, n. 22, p. 188 – 198, 2012.
- COUTINHO, ANA AUGUSTA PIRES; DE OLIVEIRA CECÍLIO, LUIZ CARLOS; MOTA, JOAQUIM ANTÔNIO CÉSAR. Classificação de risco em serviços de emergência: uma discussão da literatura sobre o Sistema de Triagem de Manchester. **Revista Médica de Minas Gerais-Rmmg**, v. 22, n. 2, 2012.
- DIOGO, C. S. **Impacto da relação cidadão-sistema de triagem de Manchester na requalificação das urgências do SMS**. 2008. Tese (Doutorado).
- ÉVORA, YOLANDA DORA MARTINEZ. A enfermagem na era da informática. **Revista Eletrônica de Enfermagem**, v. 9, n. 1, 2007.
- FEIGENBAUM, E. A.; MCCORDUCK, P. The fifth generation. Addison-Wesley Pub, 1983.
- FOWLER, M. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. Bookman, Porto Alegre, 2005.
- GRAVEL, J.; MANZANO, S.; ARSENAULT, M. Validity of the Canadian Paediatric Triage and Acuity Scale in a tertiary care hospital. **Canadian Journal of Emergency Medicine**, v. 11, n. 1, p. 23 – 28, 2009.
- HEINZLE, R. et al. Protótipo de uma ferramenta para criação de sistemas especialistas baseados em regras de produção. 1995.

- JIMÉNEZ, J. GÓMEZ. Clasificación de pacientes en los servicios de urgencias y emergencias: Hacia un modelo de triaje estructurado de urgencias y emergencias. **emergencias**, v. 15, n. 1, p. 165 – 174, 2003.
- LÄHDET, E. F. et al. Analysis of triage worldwide. **Emergency Nurse**, v. 17, n. 4, 2009.
- MACKWAY-JONES, K.; MARSDEN, J.; WINDLE, J. **Sistema Manchester de classificação de risco, Versão brasileira de Cordeiro Jr Welfane**. 2^a. ed. [S.l.: s.n.], 2010. Acesso em: 2020.
- MACKWAY-JONES, K. et al. Das Manchester Triage System. **Ersteinschätzung in der Notaufnahme-Das Manchester-Triage-System**, v. 3, p. 176 – 177, 2006.
- MOLL, H. A. Challenges in the validation of triage systems at emergency departments. **Journal of clinical epidemiology**, v. 63, n. 4, p. 384 – 388, 2010.
- MUNÁRRIZ, L. A. Fundamentos de inteligencia artificial. Editum,, 1994.
- PAES, L. R. de A. **O uso da informática no processo de tomada de decisão médica em cardiologia**: um estudo de casos múltiplos em hospitais de São Paulo. 2003. 157 p. Dissertação (Mestrado) — FGV - Fundação Getúlio Vargas. Acesso em: 2020.
- PARDEY, T. G. M. The clinical practice of Emergency Department Triage: Application of the Australasian Triage Scale—An extended literature review: Part I: Evolution of the ATS. **Australasian Emergency Nursing Journal**, v. 9, n. 4, p. 155 – 162, 2006.
- PRESSMAN, R. S. Engenharia de Software. AMGH, Porto Alegre, 2011.
- RAMOS, R. A. Treinamento prático em UML. Digerati Books, São Paulo, 2006.
- REFERENCE manual Drools 7.29.0.Final. Disponível em: https://docs.jboss.org/drools/release/7.29.0.Final/drools-docs/html_single/index.html. Acesso em: 2020.
- RIBEIRO, H. da C. Introdução aos sistemas especialistas. **Livros Téc. e Científicos**, São Paulo, 1987.
- ROCHA, A. F. S. Determinantes da procura de atendimento de urgência pelos usuários nas unidades de pronto atendimento da secretaria municipal de saúde de Belo Horizonte. 2005.
- SANTOS, J. S. et al. Avaliação do modelo de organização da unidade de emergência do HCFMRP-USP, adotando, como referência, as políticas nacionais de atenção. **Medicina**, Ribeirão Preto, v. 36, n. 2/4, p. 498 – 515, 2003.
- SHORTLIFFE, EDWARD H; BLOIS, MARSDEN S. The computer meets medicine and biology: emergence of a discipline. **Journal of Biomedical Informatics**, Springer, New York, p. 3 – 45, 2006. Acesso em: 2020.
- SINGH, A. Medicina moderna: rumo à prevenção, à cura, ao bem-estar e à longevidade. **Revista Latinoamericana de Psicopatologia Fundamental**, v. 13, p. 265 – 282, 2010.
- SOUZA, DALVA INÊS DE ET AL. Manual de orientações para projetos de pesquisa. In: **FESLSVC**. [S.l.: s.n.], 2013.

TIBES, C. M. dos S.; DIAS, J. D.; ZEM-MASCARENHAS, S. H. Aplicativos móveis desenvolvidos para a área da saúde no Brasil: revisão integrativa da literatura. **Revista Mineira de Enfermagem**, Minas Gerais, v. 18, n. 2, p. 471 – 486, 2014.

UNIÃO, T. de Contas da. **ACÓRDÃO Nº 693/2014 – TCU – Plenário, Processo nº TC 032.624/2013-1**. 2014. Disponível em: <https://www.camara.leg.br/internet/comissao/index/mista/orca/tcu/PDFs/2014/Acordao6932014-TCU-Plen%C3%A1rio.pdf>. Acesso em: 2021.

WARREN, D. W. et al. Revisions to the Canadian triage and acuity scale paediatric guidelines (PaedCTAS). Canadian Journal of Emergency Medicine. **Canadian Journal of Emergency Medicine**, v. 10, n. 3, p. 224 – 232, 2008.