**CENTRO REGIONAL UNIVERSITÁRIO DE ESP. SANTO DO PINHAL**

**CURSO DE ENGENHARIA DA COMPUTAÇÃO**

**Arthur Elidio da Silva, 150271**

**Leonardo Oliveira Freitas, 150332**

**PROJETO E IMPLEMENTAÇÃO DE UM SISTEMA INTELIGENTE PARA COLETA E ANÁLISE DE SISTOMAS DE PACIENTES EM UNIDADES DE PRONTO-ATENDIMENTO**

**Espírito Santo do Pinhal**

**2019**

**CENTRO REGIONAL UNIVERSITÁRIO DE ESP. SANTO DO PINHAL**

**CURSO DE ENGENHARIA DA COMPUTAÇÃO**

**Arthur Elidio da Silva, 150271**

**Leonardo Oliveira Freitas, 150332**

**PROJETO E IMPLEMENTAÇÃO DE UM SISTEMA INTELIGENTE PARA COLETA E ANÁLISE DE SISTOMAS DE PACIENTES EM UNIDADES DE PRONTO-ATENDIMENTO**

Trabalho de Conclusão de Curso apresentado como exigência parcial para obtenção do título de Bacharel em Engenharia da Computação do Centro Regional Universitário de Espírito Santo do Pinhal à Banca Examinadora sob orientação do(a) Prof.(a.) Dr. José Tarcísio Franco de Camargo.

**Espírito Santo do Pinhal**

**2019**

**FOLHA DE APROVAÇÃO**

**Autor:** Arthur Elidio da Silva

Leonardo Oliveira Freitas

**Título:** Projeto e implementação de um sistema inteligente para coleta e análise de sintomas de pacientes em unidades de pronto-atendimento

**Avaliação:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Banca Examinadora**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Prof. (a.) Dr. José Tarcísio Franco de Camargo**

**Orientador**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Prof. (a.) MsC Patrícia Aparecida Zibordi Aceti**

**Membro**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Prof. (a.) Dr. Mônica Luri Giboshi**

**Membro**

Espírito Santo do Pinhal, 02 de Dezembro de 2019

**Dedicatória**

Dedico este trabalho a todos que tentaram usar de seu conhecimento para transformar, da melhor maneira possível, a sociedade ao seu redor.

**Agradecimentos**

Agradecemos à todos os professores que passaram os conhecimentos necessários para que esse trabalho fosse feito, em especial ao meu professor orientador Dr. José Tarcísio Franco de Camargo e nossa coordenadora Msc. Patrícia Aparecida Zibordi Aceti.

Agradecemos à minha família por sempre estar me apoiando e auxiliando em toda minha trajetória.

Agradecemos aos nossos amigos por nos ajudarem, em especial Bryan Alexander Rodrigues Rossi por me auxiliar com sua criatividade, Ana Flávia Pirola por cuidar de nossa saúde, tanto física quanto psicológica e a seus conhecimentos especializados que ajudaram no desenvolvimento do projeto.

**Epígrafe**

“Não há fatos eternos, como não há verdades absolutas.”

- Friedrich Nietzsch

Projeto e implementação de um sistema inteligente para coleta e análise de sintomas de pacientes em unidades de pronto-atendimento

Elidio da Silva, Arthur (Unipinhal) [arthur.elidio.ae@gmail.com](mailto:arthur.elidio.ae@gmail.com)

Oliveira de Freitas, Leonardo (Unipinhal) [leoff@live.com.br](mailto:leoff@live.com.br)

Franco de Camargo, José Tarcísio (Unipinhal) [eng\_unipinhal@hotmail.com](mailto:orientador@instituicao.com.br)

Resumo

Atualmente, ao ir a algum posto de pronto atendimento público, pode-se notar vários casos de demora para diagnóstico, sendo a maioria equivocados, resultando em esperas intermináveis para tratamento. Com isso em mente, o objetivo deste trabalho foi criar um sistema capaz de coletar informações de pacientes, entender o contexto dos sintomas do mesmo e, com base nessas informações, ser capaz de classificar o risco vital do paciente. Isso será realizado a partir dos conhecimentos obtidos em Lógica Nebulosa, com o intuito de solucionar as não conformidades apresentadas anteriormente nos locais de atendimentos de saúde, auxiliar profissionais no momento de leitura dos sintomas para o diagnóstico final e auxilar estudantes na formação acadêmica e entusiastas da área da saúde na procura de conhecimento.

**Palavras-chave**: Inteligência artificial; Lógica Nebulosa; Aplicação de informática médica.

Abstract

Currently in any public health care station, anyone can notice several cases of diagnostic delay, which are mostly misleading, resulting in endless waiting for treatment. Knowing this, the goal of this work is to create a system that is able to collect information from patients, understand the context of the patient's symptoms and based on this information be able to develop a previous diagnosis and vital classification. This will be done from the knowledge obtained in fuzzy logic with the purpose of solving the nonconformities presented previously in health care stations, assisting professionals when reading the symptoms for the final diagnosis, assisting students in academic training and health enthusiasts in search of knowledge.

**Key words**: [key1; key2].

1. Introdução

Atualmente, em qualquer unidade de saúde, existe um processo para o atendimento do paciente a ser respeitado. Inicialmente o mesmo deve passar pela etapa de triagem, que consiste em um sistema de seleção, coleta de dados e sintomas, sendo a seguir classificado de acordo com seu risco vital, a partir da análisedos sintomas apresentados. Uma vez feita a análise, o paciente é classificado de acordo com a sua respectiva urgência vital e, a partir desse momento, o paciente é encaminhado à espera para o tratamento do médico responsável. Essa espera é relacionada com a classificação atribuída, podendo chegar a até mais de 4 (quatro) horas (HEELJ, 2018).

O objetivo proposto pelo sistema em desenvolvimento é construir um software utilizando conhecimentos em inteligência artificial e lógica nebulosa que seja capaz de contribuir para a solução da latência do processo de coleta de sintomas e classificação de pacientes, tornando assim tanto a triagem como a pós-triagem mais ágeis.

1. Fundamentos Teóricos

Para o desenvolvimento do projeto foi necessário um estudo sobre a área da saúde, especificamente a parte de triagem, coleta de sintomas e tratamento de pacientes, para que a partir do conhecimento obtido fossemos capazes de desenvolver um sistema fuzzy apto a replicar o comportamento humano, sendo assim, capaz de escolher uma classificação acurada do paciente a partir de seus sintomas apresentados.

* 1. Conhecimentos na área da saúde

Após estudos em campo e conteúdos já publicados, foram obtidas as seguintes informações a respeito do tratamento de pacientes em postos de pronto atendimento. A triagem tem como objetivo classificar o risco vital do paciente a partir de sintomas coletados, tais como, pressão, pulso, respiração, temperatura, glicemia capilar, peso, saturação e outros. Uma vez feita a análise de sintomas, o paciente é classificado de acordo com o seu grau de urgência vital seguindo o padrão de classificação de manchester (BERNARDO, Elisângela Maria de Souza et al., 2017-2018).

De acordo com a classificação de manchester, o paciente pode ser classificado em 5 graus diferentes, são eles: Emergência, Muita Urgência, Urgente, Pouco Urgente e Não Urgente, os quais são distinguidos a partir das cores vermelho, laranja, amarelo, verde e azul, respectivamente. Cada tipo de classificação contém uma previsão de atendimento, seguindo a classificação de coloração vermelha até azul. São estas as previsões de atendimento: imediato, em até 20 minutos, em até 60 minutos, em até 120 minutos e em até 240 minutos. A figura 1 ilustra esta situação (HEELJ, 2018).

É importante ressaltar que a triagem hospitalar tem como objetivo obter informações do paciente, tais como a dados pessoais, coleta de sintomas e classificação do paciente e não realizar diagnóstico final do paciente.

Figura 1: Classificação de Manchester. Fonte: Elaborada pelos autores



* 1. Inteligência Artificial

O estudo em inteligência artificial tem como objetivo replicar o comportamento humano contemplando a capacidade cognitiva, reconhecimento de contexto e tomada de decisão. Ou seja, um “agente”, em uma determinada situação, para ser considerado inteligente, deve ser capaz de analisar, compreender e realizar uma tomada de decisão (NORVIC; RUSSELL, 2010).

Para o desenvolvimento da Inteligência Artificial em questão foi utilizado o referencial teórico apresentado a seguir.

* + 1. Lógica Fuzzy

O conhecimento em lógica fuzzy consiste em uma lógica multi-valorada, capaz de capturar informações vagas, em geral descritas em uma linguagem natural, e convertê-las para um formato numérico, de fácil manipulação pelos computadores atuais. Ou seja, ela permite modos de raciocínio aproximados e não exatos, podendo assim criar um conjunto de regras, onde, a partir de uma função de pertinência é possível criar uma transição gradual da “não-verdade” para a “verdade”. É apresentada na figura 2 a diferença da lógica booleana aristotélica, baseada em “verdadeiro” ou “falso”, e a lógica fuzzy, capaz de admitir vários valores (CAVALCANTI et al., 2012).

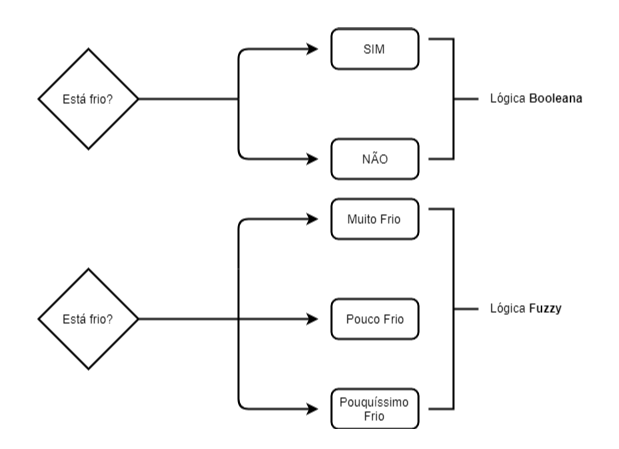


Figura 2: Diferença da Lógica Booleana para Lógica Fuzzy. Fonte: Elaborada pelos autores

* + 1. Serviços Web

De acordo com Pereira (2016), a evolução tecnológica da sociedade humana tem proporcionado a criação de sistemas apoiados em “serviços web”. Um serviço web consiste na manipulação de informações de forma centralizada, permitindo que estas sejam tratadas separadamente da interface, a qual será utilizada apenas para a visualização pelo usuário final.

Portanto, para trabalhar com a disposição dos dados aplicaremos o modelo Rest, que é uma forma de arquitetura para organização de serviços web.

* + 1. Modelo REST

O modelo REST (Representational State Transfer) representa a arquitetura atual para criação de serviços web. Nesse modelo é utilizada a semântica dos métodos HTTP (GET, POST, PUT e DELETE), o que torna esse padrão de envio de dados mais simples, leve e dinâmico (GRINBERG, 2014).

1. Materiais e Métodos.
   1. Implementação camada serviços web

Para o desenvolvimento da camada de serviços estamos utilizando a IDE Visual Code, com a linguagem de programação JavaScript e Framework Express, que é o core para criação dos serviços.

* 1. Execução dos Serviços Web

Para execução dos serviços web será utilizado o servidor Node, que se encarregará do processo de execução da aplicação, para que fiquem visíveis os serviços para consumo da camada de interface.

* 1. Interface para demonstração do processo de triagem

Será utilizada o framework Flutter da Google para desenvolvimento da interface de visualização do processo de triagem. Essa biblioteca facilita a criação de aplicações mobiles híbridas que contenham interação com usuário entre outros benefícios utilizando a linguagem Dart.

* 1. Armazenamento da triagem com banco de dados MongoDB

MongoDB é um banco de dados não relacional, o qual será usado para guardar o processo de triagem e informações do paciente, além de conter um histórico para futuras consultas.

* 1. Implementação Lógica Fuzzy

Para o desenvolvimento do sistema em questão será utilizada a linguagem Python na IDE Spyder, disponível através do pacote de programas Anaconda, que contém várias outras IDEs para desenvolvimento em Python. O Spyder foi escolhido pois disponibiliza de forma simplificada o uso das bibliotecas Scikit-Fuzzy e Numpy.

* 1. NumPy

NumPy é um pacote para a linguagem Python que suporta vetores e matrizes multidimensionais, possuindo uma larga coleção de funções matemáticas para trabalhar com estas estruturas.

* 1. Scikit-Fuzzy

Scikit-Fuzzy é uma biblioteca que contém uma coleção de algoritmos lógicos escrito em Python para o uso e implementação da lógica fuzzy. Algumas das principais funções utilizadas.

* + 1. TrimMF e TrapMF

Utilizadas para definir as funções de ativação **(Membership Function)**, que tem como propósito calcular o grau de pertinência das entradas no sistema fuzzy criado.

A biblioteca Scikit-Fuzzy disponibiliza um total de onze funções de ativações. Dessas onze funcionalidades, para a construção de um sistema fuzzy que atendesse aos requisitos levantados, foram utilizadas apenas duas, sendo elas TrimMF e TrapMF. Através delas são geradas uma função triangular e uma função trapezoidal decrescente ou crescente, respectivamente. Na figura 3 é demonstrado como as funções ficariam em forma de gráfico.

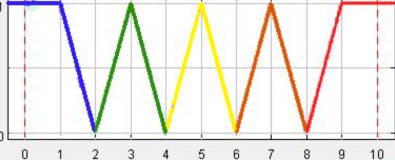


Figura 3: Sistema Fuzzy. Fonte: Elaborada pelos autores

A função triangular foi selecionada pois a mesma pode definir as transições intermediárias entre as regras de avaliação do paciente. Já a função trapezoidal decrescente (em azul) define o início das transições entre as regras e a crescente (em vermelho) define ao final destas transições. Por exemplo, caso o paciente não apresente nenhum sintoma, o sistema automaticamente deverá retornar a classificação vital mais leve e, caso tenha alguma definição crítica, deve ser indicada a classificação mais urgente do diagnostico em questão.

1. Resultados obtidos

Com base no referencial teórico adotado para construção deste sistema de inteligência artificial, quando confrontados os resultados apresentados pela IA (figuras 4 e 5) e os equivalentes de origem humana, observa-se que o sistema de lógica nebulosa atendeu às expectativas de forma plenamente satisfatória.

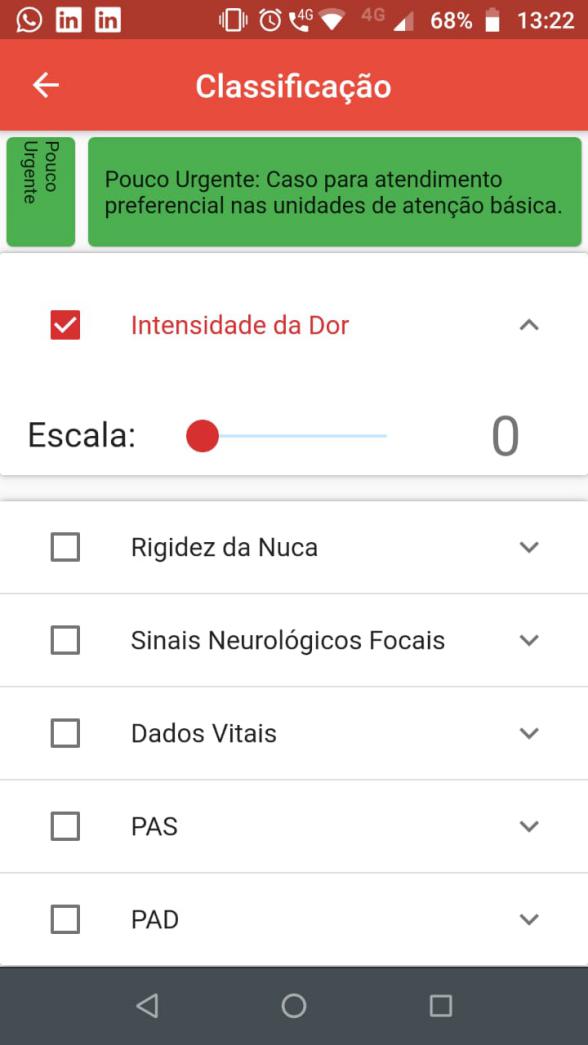
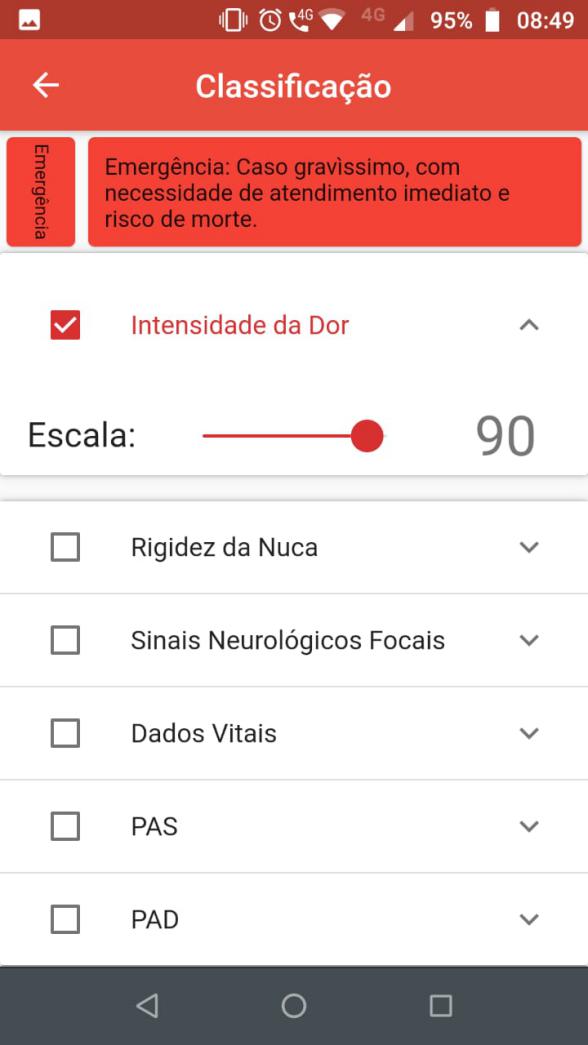
 

Figura 5: Classificação de risco na aplicação com sintomas graves. Fonte: print screen do aplicativo no sistema android

Figura 4: Classificação de risco na aplicação sem sintomas graves. Fonte: print screen do aplicativo no sistema android

**(Descrever com mais detalhes os resultados obtidos. Descrever melhor com o sistema – ou sistemas – funcionam. Vocês fizeram um único aplicativo ou são dois aplicativos? Se fizeram dois, por que?)**

1. Conclusões

Em uma análise crítica deste sistema, devem ser levados em conta dois pontos importantes: a análise apurada de diagnósticos e a granulação de sintomas para Inteligência Artificial.

Em relação à análise apurada de diagnósticos, no momento em que se assume a responsabilidade de transportar um conhecimento/comportamento humano para a área da tecnologia, por consequência se assume que a automação realizada deverá ser capaz de reproduzir com fidelidade este conhecimento/comportamento. Com esse propósito em mente, a abordagem escolhida foi a inteligência artificial a partir de uma lógica nebulosa, como indicado previamente, suportada por um modelo real utilizado em um ambiente hospitalar. Para tanto, em nosso escopo, o modelo de análise dos diagnósticos e sintomas deve ser feito por um profissional da saúde e a implementação digital por um profissional da computação, de forma a aumentar o índice de acertos e não permitir equívocos na interpretação dos dados. É de extrema importância que as duas áreas trabalhem em conjunto para que ruídos não sejam gerados, assim construindo uma aplicação com elevada eficiência.

Com relação à granulação de sintomas em IA, conforme apresentação prévia a respeito de arquiteturas de sistemas e lógica fuzzy, cada sistema fuzzy foi montado para ser um sistema a parte, ou seja, cada diagnóstico tem seu próprio sistema fuzzy capaz de gerar classificações. Contudo, os sintomas de alguns diagnósticos também podem ser utilizados em outros diagnósticos, sendo assim necessário que um sistema consulte outros para realizar uma classificação mais apurada.

1. Referências Bibliográficas

BERNARDO, Elisângela Maria de Souza et al. **Procedimento Operacional Padrão Classificação de Risco**. Espírito Santo do Pinhal: [s. n.], 2017-2018.

CAVALCANTI, José Homero Feitosa et al. **Lógica Fuzzy Aplicada Às Engenharias**. João Pessoa PB: [s. n.], 2012. Disponível em: http://www.logicafuzzy.com.br/wp-content/uploads/2013/04/logica\_fuzzy\_aplicada\_as\_engenharias.pdf. Acesso em: 22 maio 2019.

GRINBERG, Miguel. **Flask Web Development**: Developing Web Applications with Python. 1. ed. [*S. l.*]: O'Reilly Media, 2014.

HEELJ (Pirenópolis). Entenda a Classificação de Risco e o tempo de espera por atendimento. *In*: HEELJ (Pirenópolis). **Entenda a Classificação de Risco e o tempo de espera por atendimento**. Pirenópolis, 6 ago. 2018. Disponível em: http://heelj.org.br/noticias/entenda-a-classificacao-de-risco-e-o-tempo-de-espera-por-atendimento/#faleconosco. Acesso em: 15 nov. 2019.

NORVIC, Peter; RUSSUL, Stuart. **Inteligência Artificial**. [S. l.: s. n.], 2010.

SKFUZZY. **Api Reference**. Disponível em: [https://pythonhosted.org/scikit-fuzzy/api/api.html](https://pythonhosted.org/scikit-fuzzy/api/api.html.) Acesso em: 26 de maio de 2019.