

CENTRO UNIVERSITÁRIO FEI

GUILHERME CUCIO LAGO

MARCELO NOBRE

LUCAS LOPES

DOUGLAS IINO RINALDI

SOFTWARE DE PRÉ-DIAGNÓSTICO MÉDICO PARA CLASSIFICAÇÃO DE RISCO

São Bernardo do Campo

2023

GUILHERME CUCIO LAGO

MARCELO NOBRE

LUCAS LOPES

DOUGLAS IINO RINALDI

SOFTWARE DE PRÉ-DIAGNÓSTICO MÉDICO PARA CLASSIFICAÇÃO DE RISCO

Trabalho de Conclusão de Curso, apresentado ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Bacharel em Ciência da Computação. Orientado pelo Profº. Dr. Flavio Tonidandel

São Bernardo do Campo

2023

SOFTWARE DE PRÉ-DIAGNÓSTICO MÉDICO PARA
CLASSIFICAÇÃO DE RISCO / DOUGLAS IINO RINALDI...[et al.].
SÃO BERNARDO DO CAMPO, 2023.

60 p. : il.

Trabalho de Conclusão de Curso - Centro Universitário FEI.
Orientador: Prof. Dr. FLAVIO TONIDANDEL.

1. Classificação de risco. 2. Protocolo de Manchester. 3. Random Forest. 4. Aprendizado de máquina. 5. Inteligência artificial. I. IINO RINALDI, DOUGLAS. II. CUCIO LAGO, GUILHERME. III. LOPES, LUCAS. IV. NOBRE DE MORAIS, MARCELO. V. TONIDANDEL, FLAVIO, orient. VI. Título.

Elaborada pelo sistema de geração automática de ficha catalográfica da FEI com os dados fornecidos pelo(a) autor(a).

Às nossas famílias, pelo amor incondicional e apoio constante. Ao nosso orientador, à banca avaliadora e à universidade, pelo suporte e oportunidade. A todos, nosso profundo agradecimento.

AGRADECIMENTOS

Gostaríamos de expressar nossa profunda gratidão a Deus, cuja presença e força nos guiaram em cada etapa deste percurso acadêmico. A Ele, rendemos nossos mais sinceros agradecimentos por nos capacitar e sustentar ao longo de toda a jornada.

Às nossas famílias, expressamos nossa sincera gratidão pelo amor incondicional, compreensão e apoio constante. Aos nossos colegas, amigos e companheiros de estudo, agradecemos pelas experiências enriquecedoras, troca de conhecimento e apoio mútuo.

Ao nosso orientador, Dr. Flavio Tonidandel, expressamos nossa sincera gratidão. Sua orientação competente, dedicação e disponibilidade foram fundamentais para o sucesso deste trabalho. Apreciamos imensamente suas contribuições valiosas e inspiração contínua ao longo dessa jornada.

Também gostaríamos de agradecer ao professor e amigo, Dr. Paulo Sérgio Rodrigues, pelo seu apoio e confiança em nosso potencial. Suas palavras de incentivo e orientação foram fundamentais para aprimorar nossos conhecimentos e habilidades.

Por fim, agradecemos ao Centro Universitário FEI por proporcionar uma excelente formação acadêmica, infraestrutura de qualidade e oportunidades de aprendizado enriquecedoras. Estamos gratos pela experiência proporcionada e pelo ambiente propício ao nosso crescimento intelectual.

A todos aqueles que direta ou indiretamente contribuíram para a nossa jornada acadêmica, nosso mais profundo agradecimento. Vocês foram peças-chave em nossa trajetória, e somos gratos por todo apoio e incentivo recebidos.

“A tecnologia por si só não é suficiente. É a tecnologia casada com as artes liberais, casada com as ciências humanas, que nos dá o resultado que nos torna otimistas em relação ao futuro.”

Steve Jobs

RESUMO

Atualmente a classificação de risco é um processo realizado por um profissional de saúde (na maioria dos casos por um enfermeiro) assim que o paciente chega ao hospital, a fim de determinar a gravidade dos sintomas que o mesmo apresenta, além de encaixá-lo na fila com um tempo de espera condizente a tais sintomas para que todos possam ser atendidos de forma conveniente e eficiente. A falha em classificar o paciente de forma precisa pode acarretar em consequências tanto para ele quanto para todos os outros que tiveram seu tempo de espera influenciado por sua classificação. Para realizar a classificação de risco, é utilizado o Protocolo de Manchester que estipula que os profissionais de saúde realizem uma avaliação sobre o quadro clínico do paciente e o classifiquem com uma cor, representada por uma pulseira. No Brasil, o processo é realizado sem o auxílio de qualquer software ou hardware que possa acelerar ou tornar o procedimento mais eficaz: todos os sinais vitais colhidos do paciente, como frequência cardíaca e pressão arterial, são preenchidos com papel e caneta, e qualquer consulta que o funcionário precise fazer é feita em um manual físico referente ao Protocolo. Assim, este trabalho tem como objetivo propor um software capaz de armazenar todas as informações necessárias do paciente durante o pré-diagnóstico e classificá-lo de acordo com o Protocolo de Manchester. Além disso, será implementada uma Inteligência Artificial (IA) que funcionará com base no algoritmo de Random Forest para que o software seja capaz de aumentar sua precisão de acordo com seu uso.

Palavras-chave: Classificação de risco, Protocolo de Manchester, Random Forest, Aprendizado de máquina, Inteligência artificial.

ABSTRACT

The risk rating is a process fulfilled by a healthcare professional (most of the times by a nurse) the moment the patient arrives at the hospital, in order to determine the severity of their symptoms and allocate them on queue accordingly, so everyone can be treated as efficiently and conveniently as possible. Failing to rank the patient in a precise way can possibly bring consequences for all other patients who had their wait time influenced by them. The Manchester Protocol is utilized in order to perform the risk rating, indicating that healthcare professionals must do an evaluation of the patient's clinical condition and rank them through a color, represented by a bracelet. In Brazil, the process is accomplished without the aid of any software nor hardware that is capable of making the procedure more efficient: all vital signs collected from the patient, such as heart rate and blood pressure, are filled in with pen and paper and any search made by the nurse is made through a physical manual referent to the Protocol. The objective of this paper is to propose a software capable of storing all the patient's information necessary during their pre diagnosis and rank them according to the Manchester Protocol. Furthermore, an Artificial Intelligence (AI) will be implemented, using Random Forest's algorithm to enhance the software's capabilities to be more precise throughout its use.

Keywords: Risk rating. Manchester Protocol. Random Forest. Machine learning. Artificial intelligence.

LISTA DE ILUSTRAÇÕES

Ilustração 1 – Modelo de árvore de decisão.	22
Ilustração 2 – Arquitetura de uma Random Forest.	22
Ilustração 3 – Fluxograma de alterações respiratórias do protocolo de Manchester. . .	28
Ilustração 4 – Amostra da tabela de sintomas vs. pacientes baseada no protocolo de Manchester.	29
Ilustração 5 – Swagger da api do protótipo.	30
Ilustração 6 – Amostra das entradas necessárias (query) do endpoint/pre_classificacao/read método GET.	31
Ilustração 7 – Resultado obtido após consultar o endpoint/pre_classificacao/read (método GET).	31
Ilustração 8 – Endpoint/pre_classificacao/update para treinamento do algoritmo (método PUT).	32
Ilustração 9 – Endpoint/pre_classificacao/gerar_fig para geração de árvores de decisão (método PUT).	33
Ilustração 10 – Funções utilizadas no desenvolvimento Backend do protótipo, "gerar_arvore" realiza o treinamento com base nos dados levantados e "analisar_risco" classifica um paciente de acordo com as árvores geradas. . .	34
Ilustração 11 – Amostra de uma árvore de decisão gerada pelo algoritmo Random Forest. .	35
Ilustração 12 – Modelo de interface usuário e protótipo.	36
Ilustração 13 – Casos de Uso.	37
Ilustração 14 – Diagrama de classes.	42
Ilustração 15 – Modelo de interface usuário e protótipo com botão de auxílio ao médico. .	44
Ilustração 16 – Interface de usuário e software para adição de pacientes / sintomas. . .	45
Ilustração 17 – Campos para adição de sintomas.	45
Ilustração 18 – Campos para adição de sintomas preenchido.	46
Ilustração 19 – Modelo de interface usuário com campo preenchido.	46
Ilustração 20 – Comparação entre o CSV de entrada (lado esquerdo) versus saída (lado direito) após a inserção de um novo sintoma.	47
Ilustração 21 – Interface do usuário e software com dados de um paciente preenchidos. .	48
Ilustração 22 – Campo para adicionar os dados do paciente preenchidos no exemplo anterior.	48

Ilustração 23 –Campo para atualizar o risco recomendado para o paciente em análise.	49
Ilustração 24 –Modelo de interface usuário e software já classificando o paciente com o risco atualizado.	49
Ilustração 25 –Comparação entre o CSV de entrada (lado esquerdo) versus saída (lado direito) após inserção de um novo paciente.	50
Ilustração 26 –Matriz de confusão.	51
Ilustração 27 –Gráfico: Precisão vs. Profundidade.	53
Ilustração 28 –Gráfico: Precisão vs. Quantidade de árvores.	53

LISTA DE TABELAS

Tabela 1 – Matriz de Confusão	24
Tabela 2 – Tabela de Requisitos Funcionais	41
Tabela 3 – Tabela de Requisitos Não Funcionais	42

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVO	14
1.2	JUSTIFICATIVA	14
1.3	ESTRUTURA DO TRABALHO	15
2	TRABALHOS RELACIONADOS	16
2.1	PROJETOS DE SISTEMAS COMPUTACIONAIS BASEADOS NO PROTOCOLO DE MANCHESTER	16
2.2	PROJETOS EMPREGANDO O ALGORITMO RANDOM FOREST	17
2.3	PROJETOS PARA PRÉ DIAGNÓSTICO EMPREGANDO ALGORITMOS DIVERSOS	18
2.4	SEMELHANÇAS COM TRABALHOS RELACIONADOS	19
3	CONCEITOS FUNDAMENTAIS	20
3.1	PROTOCOLO DE MANCHESTER	20
3.1.1	CLASSIFICAÇÃO DO PROTOCOLO DE MANCHESTER	20
3.2	RANDOM FOREST	21
3.3	FASTAPI	23
3.4	VUE.JS	23
3.5	MATRIZ DE CONFUSÃO	24
4	METODOLOGIA	25
4.1	FASE DE DIAGNÓSTICO	25
4.2	CONCEPÇÃO	26
4.2.1	INTEGRAÇÃO DO PROTOCOLO DE MANCHESTER AO PROTÓTIPO	27
4.2.2	TABELA DE PACIENTES	27
4.2.3	DESENVOLVIMENTO DE API'S RESTFULL	30
4.2.4	IMPLEMENTAÇÃO DO ALGORITMO RANDOM FOREST	33
4.2.5	DESENVOLVIMENTO DA INTERFACE USUÁRIO E PROTÓTIPO	35
4.3	LEVANTAMENTO E ANÁLISE DE REQUISITOS	37
4.3.1	CASOS DE USO	37
4.3.2	DIAGRAMA DE CLASSE	42
4.4	FASE DE DESENVOLVIMENTO	43
4.4.1	INTEGRAÇÃO DA FUNCIONALIDADE ADICIONAR SINTOMAS	43

4.4.2	INTEGRAÇÃO DA FUNCIONALIDADE ADICIONAR DADOS DE PACI- ENTE NO BANCO DE DADOS	47
5	RESULTADOS	51
5.0.1	MATRIZ DE CONFUSÃO	51
5.0.2	MÉTRICAS DE AVALIAÇÃO	52
6	DISCUSSÃO	54
7	CONCLUSÃO	55
8	REFERÊNCIAS	57

1 INTRODUÇÃO

O serviço de pré-diagnóstico médico se tornou uma parte crucial do departamento de emergência, como forma de distribuir melhor os recursos hospitalares (BIOMEDICAL INFORMATICS, 2007), garantindo assim que os pacientes em risco sejam atendidos de forma mais eficiente e eficaz. Por isso percebe-se a necessidade de criar um Sistema de Apoio à Decisão Clínica (SADC) para otimizar o processo e garantir maior qualidade na classificação do paciente.

O pré-diagnóstico é a fase de entrevistas pessoais com um profissional de saúde no pronto-socorro para classificar pacientes por níveis de risco e assim priorizar o atendimento daqueles que mais precisam (MARÍA M. ABAD-GRAU. et al, 2007). Alguns exemplos de protocolo de pré-diagnóstico são: Escala Nacional de Triagem Australiana (ATS) criada em 1993; o Sistema de Triagem Manchester (MTS) (desde 1997); Escala de Triagem e Acuidade do Departamento de Emergência Canadense (CTAS) influenciada pelo ATS e MTS; o Índice de Gravidade de Emergência (ESI) utilizado nos Estados Unidos, baseado no ATS, MTS e CTAS, e o Modelo Andorrano de Triagem (MAT) ou Sistema Espanhol de Triagem (SET), baseado no MAT. Todos os protocolos usam uma escala de 5 níveis para categorizar um paciente, sendo essa escala baseada geralmente nos sintomas. No Brasil o protocolo mais utilizado é o do Sistema de Triagem Manchester (MTS), que será o foco nesse projeto.

Um recurso altamente utilizado nos hospitais americanos é o Sistema de Apoio à Decisão Clínica (SADC) que por intermédio de uma inteligência artificial (IA) consegue fornecer suporte na hora de classificar um paciente de acordo com seus sintomas (MARÍA M. ABAD-GRAU. et al, 2007). Alguns requisitos precisam ser atendidos para a implementação, sendo eles: a robustez quando alguma informação estiver faltando; precisão na hora da classificação; legibilidade humana para que os especialistas médicos possam entender e até modificar o modelo usado para desenvolver o SADC; e a adaptabilidade quando novas informações são adicionadas a base de conhecimento.

Com base nas informações acima foi proposto neste projeto o desenvolvimento de um software para pré-diagnóstico de pacientes utilizando uma modelagem SADC baseada no algoritmo Random Forest (RANDOM FOREST CLASSIFIER, 2007 - 2023) seguindo o protocolo de triagem Manchester (MANUAL ACCR, 2022), que utiliza como base um conjunto de várias árvores de decisão.

1.1 OBJETIVO

Desenvolver um software capaz de auxiliar o profissional de saúde (médicos e enfermeiros) a realizar um pré-diagnóstico a fim de classificar o paciente de acordo com seu risco baseado no Protocolo de Manchester.

1.2 JUSTIFICATIVA

A área da saúde brasileira enfrenta desafios significativos nos prontos atendimentos, como reclamações frequentes relacionadas ao tempo de atendimento, precisão na classificação de pacientes e ocorrência de erros. Essas questões impactam negativamente a qualidade do serviço oferecido e a eficiência dos recursos hospitalares.

Nesse contexto, o pré-diagnóstico médico surge como uma etapa crucial para a distribuição eficiente dos recursos e a prestação de um atendimento adequado aos pacientes em risco. No entanto, a falta de suporte e ferramentas especializadas para os profissionais de saúde pode comprometer a precisão e a eficácia dessa etapa.

Diante dessa necessidade, propõe-se a elaboração de um Sistema de Apoio à Decisão Clínica (SADC) como tema central deste trabalho. O objetivo é desenvolver uma solução personalizada para os prontos atendimentos brasileiros, visando otimizar o processo de pré-diagnóstico, melhorar o tempo de atendimento e aprimorar a classificação de pacientes.

Ao explorar trabalhos relacionados na área da saúde brasileira, será possível identificar as principais dificuldades enfrentadas nos prontos atendimentos, bem como as reclamações recorrentes. Dessa forma, o desenvolvimento do SADC busca superar esses desafios, fornecendo uma ferramenta eficaz e eficiente para auxiliar os profissionais de saúde na tomada de decisões clínicas.

Espera-se que este trabalho contribua para a melhoria do serviço prestado nos prontos atendimentos, proporcionando uma triagem mais precisa, reduzindo o tempo de espera dos pacientes e minimizando a ocorrência de erros. Além disso, ao fornecer apoio e orientação aos profissionais de saúde, espera-se aumentar a confiança na classificação de risco dos pacientes, resultando em um atendimento mais eficiente e uma melhor alocação de recursos hospitalares.

Com base nessas considerações, este trabalho se justifica pela relevância e necessidade de desenvolver um SADC voltado aos prontos atendimentos brasileiros, com o intuito de otimizar o processo de pré-diagnóstico, garantir um atendimento eficaz e melhorar a qualidade dos serviços prestados na área da saúde.

1.3 ESTRUTURA DO TRABALHO

O Trabalho foi dividido nos seguintes tópicos:

- No capítulo 2 foram descritos trabalhos ou estudos realizados na área, com o objetivo de apresentar as pesquisas realizadas anteriormente.
- No capítulo 3 foram mostrados os conceitos apresentados e utilizados para o tema, para entendimento mais profundo das técnicas utilizadas para construção do protótipo e do software.
- No capítulo 4 foi detalhada a metodologia que foi utilizada para o desenvolvimento do software, demonstrando as técnicas e os passos realizados para atingir o objetivo.
- No Capítulo 5 foram apresentados os resultados obtidos após a realização da metodologia.
- O Capítulo 6 detalha as discussões sobre os resultados obtidos.
- Por fim, no Capítulo 7 foi apresentada a conclusão do trabalho.

2 TRABALHOS RELACIONADOS

Este capítulo tem como premissa citar artigos relacionados ao tema deste trabalho. O mesmo foi dividido nas seguintes subseções:

- Projetos de sistemas computacionais baseados no Protocolo de Manchester - Neste tópico serão apresentados artigos que auxiliaram na determinação do sistema computacional realizado neste trabalho.
- Projetos empregando o algoritmo Random Forest - Serão apresentado trabalhos e artigos que utilizaram o algoritmo e o motivo de seu uso.
- Projetos para pré-diagnóstico empregando algoritmos diversos - Serão apresentados trabalhos e artigos de projetos para pré-diagnóstico médico, indicando a tendência de pesquisas na área.

2.1 PROJETOS DE SISTEMAS COMPUTACIONAIS BASEADOS NO PROTOCOLO DE MANCHESTER

Cicolo et al. (2018) tem como objetivo apresentar a precisão e o tempo para a realização do Sistema Manchester de Classificação de Risco (SMCR) em registro eletrônico e manual, concluindo que o uso do registro eletrônico apresenta vantagens no quesito de precisão, confiança e o tempo demandado para a realização da classificação de risco, mostrando a utilização de novas tecnologias dentro dos processos de pré-diagnóstico.

María M. Abad-Grau et al. (2007) criou um artigo citando os desafios de se desenvolver sistemas computacionais para auxílio à pré-diagnóstico baseados em diversos protocolos, entre eles, o Manchester. O estudo foi realizado devido as vantagens que esses sistemas trazem para tomada de decisão em comparação com a tomada de decisão de especialistas na área, pois os sistemas baseados em conhecimento para auxiliar a tomada de decisão no pré-diagnóstico geralmente tentam abranger um domínio muito mais amplo, garantindo resultados muitas vezes mais precisos. O artigo demonstra diversas metodologias de como implementar um sistema computacional e os critérios utilizados para selecionar qual método é o mais adequado se baseiam na porcentagem de precisão e na fácil compreensão de um especialista para avaliar o critério que o sistema utilizou para fazer o pré diagnóstico. Partindo dos critérios levantados é possível concluir que as melhores metodologias se baseiam em árvores de decisão e redes bayesianas.

Partindo dos artigos, foi decidido criar um sistema que receba entradas do usuário, garantindo um processo mais rápido e preciso do que a avaliação manual, mas mantendo a autonomia do profissional realizando o pré-diagnóstico.

2.2 PROJETOS EMPREGANDO O ALGORITMO RANDOM FOREST

As instituições Zhejiang Provincial People's Hospital e Zhejiang Chinese Medical University desenvolveram um artigo sobre análise multicêntrica e um modelo de triagem rápida para prever pneumonia por coronavírus precoce utilizando o algoritmo Random Forest (BAO, SUXIA. et al, 2021). O estudo foi realizado devido ao fato de que os sintomas iniciais de pneumonia por COVID-19 são inespecíficos e geralmente se apresentam como febre, tosse, dor de cabeça, vômito ou diarreia; além disso, alguns casos apresentam-se sem quaisquer sintomas. A partir dos pontos anteriores, o foco do estudo foi desenvolver um modelo de triagem rápido e assistido por computador com base em dados epidemiológicos, características clínicas e resultados laboratoriais e exames de imagem para detectar pneumonia precoce por COVID-19 em indivíduos suspeitos. O algoritmo Random Forest foi escolhido pois o método apresenta vantagens evidentes, incluindo uma baixa chance de overfit, redução de ruído mais robusta, velocidade de treinamento mais rápida e maior precisão de previsão. Além disso, em comparação com outros modelos de previsão, este modelo pode identificar de forma mais eficaz as interações e relações não lineares entre as variáveis.

As instituições Innovare Biomarkers Laboratory e School of Pharmaceutical Sciences (FCF) desenvolveram o artigo sobre um aplicativo de método simples de triagem para pacientes com zika vírus baseado no algoritmo Random Forest (DELAFIORI, JEANY. et al, 2018). O estudo teve como objetivo substituir o método de reação em cadeia da polimerase com transcriptase reversa (RT-PCR). Este método demanda uma quantidade considerável de tempo e recursos, como kits e reagentes que podem resultar em um ônus financeiro substancial para indivíduos afetados. O estudo apresenta uma poderosa combinação de espectrometria de massa de alta resolução e um modelo de previsão de aprendizado de máquina por meio do Random Forest para análise de dados. O algoritmo Random Forest foi selecionado para este projeto pelo alto desempenho de classificação, sem necessidade de ajustes complexos de kernel e parametrização, desempenho de execução e importância do recurso (fornece informações sobre a relevância de cada recurso nas árvores de decisão).

Os autores Emina e Abdulhamit (2016) desenvolveram um sistema de apoio à decisão médica para diagnóstico de arritmia cardíaca utilizando o algoritmo Random Forest. O

algoritmo foi selecionado pois o mesmo alcança desempenhos superiores em comparação com outros métodos de árvore de decisão usando validação cruzada de 10 vezes para os conjuntos de dados de eletrocardiograma. Os resultados obtidos sugerem melhorias significativas em termos de precisão de classificação.

Seguindo os trabalhos, foi decidido o uso do algoritmo Random Forest, pois o mesmo não exige tanto poder computacional e é mais rápido no treinamento. Além desses fatores, o Random Forest não exige grandes quantidades de dados, característica vantajosa para este trabalho.

2.3 PROJETOS PARA PRÉ DIAGNÓSTICO EMPREGANDO ALGORITMOS DIVERSOS

Gonçalves e Monticelli et al. (2022) tentam entender a aceitação e a utilização de uma inteligência artificial (IA) na tomada de decisões médicas, tendo como base a relação entre as habilidades sociais dos médicos, a utilização de IA no auxílio de tomada de decisões e intercorrência de decisões equivocadas, trazendo resultados positivos quanto a utilização de tecnologia em sistemas médicos e os benefícios de utilizar IA na área da medicina.

Karollyne, Estella e José et al. (2021) apresentam um Mapeamento Sistemático da Literatura (MSL) para identificar algumas mudanças causadas por inserir uma IA no cotidiano dos profissionais de saúde, sendo um dos resultados da pesquisa, que a utilização do processamento de texto é uma das mais significativas para buscar informações em prontuários, diagnósticos e recomendação de tratamentos.

Katia Arima (2020) explica que um dos maiores problemas durante a pandemia de COVID-19 no Brasil, era a demora e dificuldade para diagnosticar a doença, sendo a solução o emprego de IA no procedimento. Para isso, o Laboratório de Big Data e Análise Preditiva em Saúde (Labdaps) da Faculdade de Saúde Pública da Universidade de São Paulo (USP) desenvolveu um sistema de apoio a decisão médica com uma IA para analisar dados clínicos básicos, como exames de sangue, para chegar a um percentual de chance do paciente ter a doença, auxiliando a equipe médica na tomada de decisões e a compreender a severidade do caso, apresentando se o paciente tem a necessidade de utilizar ventilação externa ou até mesmo a chance do paciente ir a óbito.

Damilano, C. T. (2019) fala sobre o doutor assistente IA, criado pela empresa chinesa iFlytek, sendo o primeiro robô com aspectos humanoides que conseguiu uma licença para praticar medicina, tendo uma IA que se baseia em realizar diagnósticos e conduta dos pacientes, por meio de diagnósticos anteriores realizados por médicos humanos. Ainda estando em fase de tes-

tes, após o diagnóstico do robô, é necessário que um médico humano confirme o procedimento, para evitar qualquer intercorrência durante o processo do atendimento médico.

Os projetos citados auxiliaram na decisão da utilização de uma IA, pois mostram seu grande benefício na área de pré-diagnóstico.

2.4 SEMELHANÇAS COM TRABALHOS RELACIONADOS

Este projeto se assemelha aos trabalhos citados principalmente em relação a implementação de uma IA para auxílio em pré-diagnósticos, utilizando o algoritmo Random Forest. O objetivo geral do trabalho tem a mesma base dos demais estudos e projetos: implementar a inteligência artificial para agilizar e garantir uma maior eficácia nos diagnósticos. O diferencial ocorre em relação ao uso do Protocolo de Manchester como sua base, até então pouco utilizado anteriormente, além de sua aplicação em uma situação do cotidiano (classificação de risco) cujo demanda uma solução rápida, simples e eficaz através da implementação do software para um trabalho até então manual.

3 CONCEITOS FUNDAMENTAIS

Neste capítulo serão apresentados os conceitos, técnicas e algoritmos que serão utilizados durante o trabalho.

3.1 PROTOCOLO DE MANCHESTER

O Protocolo de Manchester é uma abordagem de pré-diagnóstico médico amplamente utilizado em serviços de emergência e prontos atendimentos ao redor do mundo. Desenvolvido por médicos e enfermeiros de nove hospitais em Manchester, Reino Unido, durante um trabalho colaborativo entre 1994 e 1995, esse protocolo tem como objetivo principal classificar os pacientes de acordo com a gravidade do seu quadro clínico.

A metodologia do Protocolo de Manchester foi introduzida no Brasil em 2008, inicialmente no estado de Minas Gerais, e desde então tem se expandido para diferentes regiões do país. O seu objetivo é otimizar o atendimento aos pacientes, garantindo que eles sejam atendidos no tempo adequado e de forma eficaz, de modo a evitar o agravamento do seu quadro clínico.

O Protocolo de Manchester utiliza um sistema de triagem baseado em cores para categorizar os pacientes de acordo com a gravidade de sua condição médica. Cada cor corresponde a um nível de prioridade, indicando a urgência do atendimento. Essa classificação auxilia os profissionais de saúde na tomada de decisões rápidas e adequadas, direcionando os recursos disponíveis para os pacientes que apresentam maior risco ou necessidade imediata.

Ao utilizar o Protocolo de Manchester, os serviços de saúde podem otimizar a utilização dos recursos disponíveis, priorizando o atendimento aos pacientes que necessitam de cuidados imediatos. Isso contribui para a redução do tempo de espera, uma melhor alocação de recursos e uma maior eficiência na prestação dos serviços de saúde.

3.1.1 CLASSIFICAÇÃO DO PROTOCOLO DE MANCHESTER

O Protocolo de Manchester classifica o paciente por cinco cores, que são elas:

- Vermelho - estado de emergência. O paciente precisa ser atendido imediatamente, pois corre risco de morte.

- Laranja - estado muito urgente. O paciente também corre risco de morte, mas sua condição é mais estável que a condição anterior. Necessitam de atendimento em até 10 minutos.
- Amarelo - estado urgente. O paciente está num estado de gravidade moderada, necessitando de atendimento em até 60 minutos. Durante a espera o enfermeiro deve reavaliar a condição do paciente em intervalos de 30 minutos.
- Verde - estado pouco urgente. Não apresentam risco potencial de agravamento. O paciente necessita de atendimento em até 120 minutos.
- Azul - estado não urgente. São casos de baixa complexidade, portanto não há necessidade de atendimento médico no mesmo dia, mas necessitam de atendimento ambulatorial.

3.2 RANDOM FOREST

Segundo a IBM Cloud Education (2020) o Random Forest é um algoritmo de machine learning baseado em árvores de decisão que combina o resultado de várias árvores para chegar a um único resultado, sendo muito utilizada para resolver problemas de classificação e regressão, sendo classificado como um algoritmo mais seguro e preciso por ter um risco de overfitting menor e apresentar resultados melhores do que uma árvore de decisão simples.

Para entender o Random Forest é necessário compreender o que é uma árvore de decisão, que nada mais é do que uma tabela de decisão em forma de árvore, começando com um único nó, onde será introduzida uma pergunta simples, gerando uma resposta, resultando em nós adicionais que ramificam em outras possibilidades. O exemplo a seguir foi retirado do IBM Cloud Learn Hub (2020), apresentando uma árvore de decisão com as condições para surfar, iniciada com uma pergunta simples: “Devo surfar?”. Após isso são realizadas uma série de perguntas para determinar um resultado e gerar uma árvore de decisão, respondendo a pergunta inicial (Figura 1).

Fonte: IBM Cloud Education.

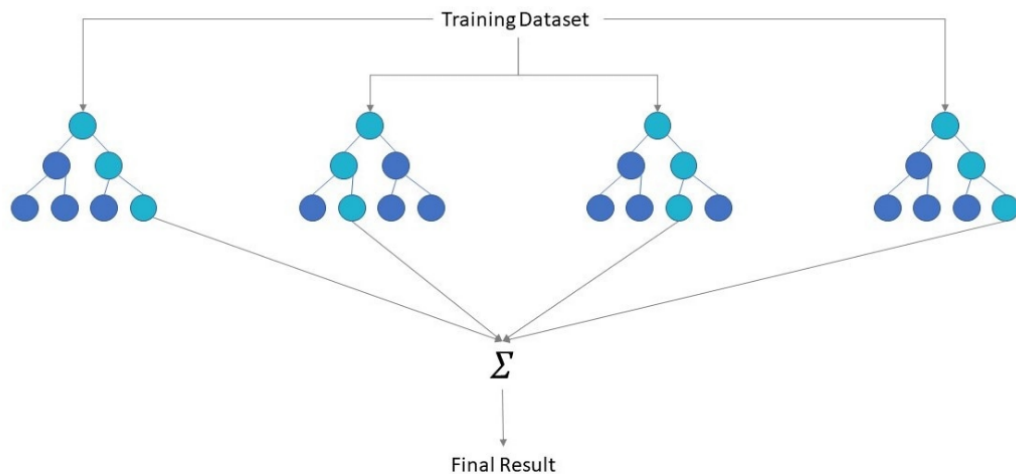
Segundo a IBM Cloud Education (2020) o Random Forest é uma extensão do método de ensacamento, também conhecido como Agregação de Bootstrap, que é usado para melhorar a precisão, reduzindo a variação e evitando o excesso de ajustes, utilizando o método de aleatoriedade de recursos para dessa forma gerar uma variedade de árvores de decisão não correlacionadas de forma aleatória. Após a geração do modelo, as previsões são feitas por meio de

Figura 1 – Modelo de árvore de decisão.



votações, sendo que cada árvore gerada toma uma decisão a partir dos dados obtidos, sendo a decisão mais votada a resposta que o algoritmo irá retornar ao final do procedimento (Figura 2).

Figura 2 – Arquitetura de uma Random Forest.



Fonte: IBM Cloud Education.

O algoritmo Random Forest é aplicado em vários meios, porém onde ele se destaca é no mundo financeiros, pois reduz o tempo gasto no gerenciamento de dados e no pré-processamento, podendo ser utilizado para avaliar possíveis clientes que tenham um alto risco de crédito. Outra área que o algoritmo é muito explorado é a área da saúde, sendo que Emina e Abdulhamit

(2016) já desenvolveram um sistema de apoio a decisão médica para diagnosticar pacientes com arritmia cardíaca, utilizando o algoritmo Random Forest.

Na prática, para aplicar o algoritmo do Random Forest, há várias bibliotecas contendo suas lógicas e funções, sendo as mais utilizadas o TensorFlow, uma biblioteca de código aberto para machine learning desenvolvida pelo Google Brains e a biblioteca Python de código aberto, sklearn, que inclui vários algoritmos de classificação, regressão e agrupamento sendo inicialmente desenvolvida para agregar e interagir com outras bibliotecas numéricas e científicas, como NumPy e SciPy.

A classe RandomForestClassifier é uma classe com o algoritmo Random Forest embutido (mais adequado para o projeto presente), descrita no apêndice A.

3.3 FASTAPI

Uma API (Application Programming Interface) é um conjunto de padrões que fazem parte de uma interface e que permite a criação de plataformas de maneira mais simples e prática para desenvolvedores.

E de acordo com a sua documentação (FastAPI, 2022), a FastAPI é um framework web moderno e rápido para construir APIs com Python 3.7+ baseado em dicas do tipo Python padrão.

Além da facilidade de criação de rotas e retornos de dado do tipo JSON (Formato padrão de mensagem enviada entre API's), o FastAPI gera toda uma documentação utilizando o OpenAPI de forma automática (Swagger).

Pelo fato do framework FastAPI gerar uma OpenAPI automática e de fácil manuseio, foi adotada a implementação do mesmo para construção de API's para o projeto em questão.

3.4 VUE.JS

Segundo a DEVMEDIA (2022) o Vue.js é um framework JavaScript de código-aberto, focado no desenvolvimento de interface de usuário.

O Vue.js permite uma vinculação fácil entre os modelos de dados e a camada de apresentação. O framework também permite reutilizar componentes em toda a aplicação, sem a necessidade da criação de modelos especiais ou coleções e registrar eventos no objeto. O usuário não precisa seguir nenhuma sintaxe especial e nem instalar nenhuma das intermináveis dependências que frameworks convencionais exigem.

Seus modelos são objetos JavaScript simples, que podem ser vinculados a qualquer componente em suas Views.

Devido a sua simplicidade de criar objetos JavaScript e pela facilidade que o framework traz ao usuário na hora de instalar dependências, foi adotada a implementação do mesmo para desenvolvimento da interface usuário e máquina (frontend) do projeto.

3.5 MATRIZ DE CONFUSÃO

Uma matriz de confusão, também conhecida como tabela de contingência, é uma ferramenta de análise utilizada em aprendizado de máquina e estatística para avaliar o desempenho de um modelo de classificação. Ela permite visualizar e analisar a relação entre as classificações feitas pelo modelo e as classes reais dos dados, essa informação foi retirado da documentação scikit-learn na seção matriz de confusão.

A matriz de confusão é geralmente organizada em formato de tabela, onde as linhas representam as classes reais e as colunas representam as classes previstas pelo modelo. Cada célula da matriz representa a contagem de instâncias que foram classificadas de acordo com determinada combinação entre a classe real e a classe prevista.

A estrutura básica de uma matriz de confusão é a seguinte:

Tabela 1 – Matriz de Confusão

		Classe Prevista	
		Positiva	Negativa
2*Classe Real	Positiva	TP	FN
	Negativa	FP	TN

Onde:

- TP (True Positive): indica a quantidade de instâncias que foram corretamente classificadas como positivas.
- FN (False Negative): representa a quantidade de instâncias que foram erroneamente classificadas como negativas, mas que são positivas na realidade.
- FP (False Positive): representa a quantidade de instâncias que foram erroneamente classificadas como positivas, mas que são negativas na realidade.
- TN (True Negative): indica a quantidade de instâncias que foram corretamente classificadas como negativas.

4 METODOLOGIA

A metodologia seguiu quatro etapas básicas de desenvolvimento de software, sendo elas:

- Fase de diagnóstico - a primeira etapa tem como objetivo conhecer e detalhar o problema: resolver incertezas que um pré-diagnóstico causa para identificar riscos nos pacientes com o máximo de eficiência e agilidade. Nesse tópico será abordado questões como origem do problema, particularidades da situação atual do problema, exigências principais geradas pelo problema e abrangência do problema e da solução.
- Concepção - esta etapa tem como objetivo criar e validar ideias para resolução do problema. A ideia proposta é desenvolver um software baseado no algoritmo Random Forest para classificar os riscos dos pacientes no pré-diagnóstico, baseando-se no Protocolo de Manchester. A validação da ideia se dará por meio de um protótipo.
- Levantamento e análise de requisitos - para terceira etapa serão levantados todos os requisitos (funcionais e não funcionais) que serão implementados no software. Para isso, serão utilizados alguns diagramas que auxiliarão na determinação de tais requisitos, entre eles o diagrama de caso de uso e diagrama de classes.
- Fase de desenvolvimento - por fim, a última etapa é o desenvolvimento, na qual haverá a implementação das primeiras linhas de código do projeto. Nesse tópico que será feito o treinamento do algoritmo com uma base de dados reais de pacientes.

4.1 FASE DE DIAGNÓSTICO

Uma série de entrevistas foram realizadas ao longo do desenvolvimento do projeto. Um dos entrevistados foi o Dr. João Victor Batista Baracho, 26 anos, formado pela Universidade Nove de Julho no ano de 2021 em Medicina. Atualmente atua como médico plantonista em diversas unidades em São Bernardo do Campo e São Paulo.

A primeira etapa consistiu na introdução do tema e do conceito do protótipo ao entrevistado para discussão de sua viabilidade e demanda em seu campo de atuação. Em seguida, o entrevistado comentou sobre sua rotina, pontos fortes e fracos no processo de atendimento aos pacientes. A partir disso, as perguntas realizadas ao Dr. João Victor tiveram como objetivo obter conhecimento sobre o sistema utilizado nas unidades de São Bernardo do Campo (Sistema

Hygia) e no resto do Brasil (DATASUS), além de obter mais informação sobre o Protocolo de Manchester e sua aplicação no pré-diagnóstico realizado para realização da classificação de risco.

Na segunda etapa, após diversas entrevistas realizadas, dados técnicos sobre o assunto foram obtidos com a adição de conhecimento sobre como o procedimento é feito nas diferentes unidades de atendimento em São Bernardo.

Com base nas entrevistas realizadas, foi concluído que o processo manual de classificação de risco, sem o auxílio de software ou tecnologia, apresentava limitações como a demora no atendimento e a imprecisão que o paciente é classificado. Diante dessa constatação, decidiu-se adotar a implementação de um software para suprir essa demanda e auxiliar os enfermeiros no pré-diagnóstico. Esse software seria capaz de oferecer resultados mais precisos ao utilizar uma abordagem baseada em inteligência artificial, especificamente o algoritmo de Random Forest. Essa solução tem o potencial de melhorar a eficiência do processo e fornecer informações mais confiáveis para o trabalho dos profissionais de saúde.

É imprescindível ressaltar o impacto que as entrevistas com o Dr. João Victor tiveram em relação a definição e desenvolvimento do tema do trabalho, que foi ajustado ao longo deste processo.

4.2 CONCEPÇÃO

A concepção do projeto será feita por intermédio de um protótipo de software para pré-diagnóstico médico utilizando como base o Protocolo de Manchester e a metodologia do algoritmo de Random Forest. Esta etapa será dividida nas seguintes subseções:

- Integração do Protocolo de Manchester ao protótipo - explicação geral da ideia que levou a construção de uma tabela fictícia para ser usado como teste para o algoritmo Random Forest.
- Tabela de pacientes - será desenvolvido uma tabela fictícia de paciente por sintomas baseado no Protocolo de Manchester. O algoritmo Random Forest realizará seu treinamento por intermédio desta tabela.
- Desenvolvimento de API's RESTfull - detalhamento das API's que serão utilizadas para integração entre frontend e backend do protótipo.

- Implementação do algoritmo Random Forest - será toda a parte backend do protótipo, como será usado o algoritmo Random Forest e a integração das API's e tabela de pacientes no mesmo.
- Desenvolvimento da interface usuário e protótipo - nesse tópico será detalhado toda a parte de frontend do protótipo, utilizando framework vue.js.

4.2.1 INTEGRAÇÃO DO PROTOCOLO DE MANCHESTER AO PROTÓTIPO

O protocolo de Manchester é caracterizada como uma árvore de decisão já definida pelos profissionais da área de saúde. Entretanto, essa árvore está no formato de texto no qual só humanos conseguem interpretar e tomar uma decisão, determinando assim qual área de risco o paciente em análise se encontra.

Como a ideia do trabalho proposto é de criar um protótipo que determina a área de risco de um paciente por intermédio do algoritmo Random Forest, algoritmo esse que gera árvores de decisão, foi tomada a escolha de criar uma tabela fictícia que fornece os dados necessários para o algoritmo gerar as árvores, porém esta tabela é criada da forma que as árvores geradas tenham o mesmo resultado que o Protocolo de Manchester, obtendo assim uma árvore de decisão que é lida pela máquina para o mesma poder realizar a classificação de riscos dos pacientes.

É importante ressaltar que a tabela fornecida para o algoritmo Random Forest pode ser atualizada com mais casos de pacientes, com isso é possível dizer que o protótipo, além de ter uma árvore de decisão inicial que nos fornece os mesmos resultados que o Protocolo de Manchester, ainda é capaz de gerar árvores de decisão muito mais precisas, garantindo assim resultados mais precisos na classificação dos pacientes.

4.2.2 TABELA DE PACIENTES

A tabela foi montada seguindo o seguinte fluxograma: (Figura 3).

Figura 3 – Fluxograma de alterações respiratórias do protocolo de Manchester.

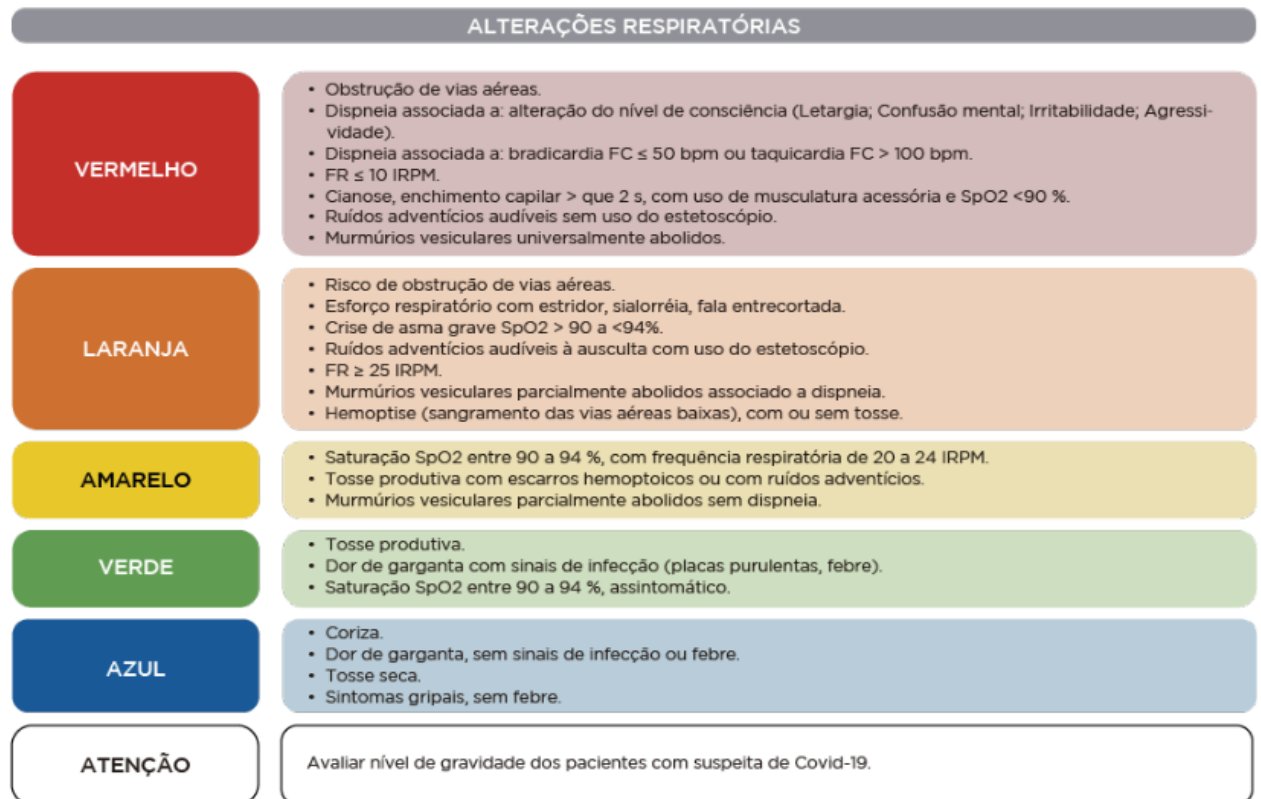


Figura retirada do MANUAL ACCR (2022).

Após seleção do fluxograma foi iniciada a etapa de definição dos campos de sintomas para preenchimento da tabela de pacientes. Para este procedimento, foi estudado cada item do fluxograma e gerado uma sigla para cada um deles. Também foi convertida a declaração de risco do pré-diagnóstico da seguinte forma: os riscos são identificados por cores enquanto os pacientes serão identificados por números de 1 a 5 em sua tabela (facilitando a análise do algoritmo, pois o mesmo trabalha melhor com números), sendo eles, Vermelho - 1, Laranja - 2, Amarelo - 3, Verde - 4 e Azul - 5.

Em seguida, por intermédio de um planilha, foi criada uma tabela de sintomas e pacientes (Figura 4), no qual a coluna será a junção dos sintomas no formato de siglas geradas na etapa anterior com adição da coluna de riscos, que é o estado do paciente de acordo com os sintomas. As linhas serão os dados de cada paciente. Cada coluna da tabela será preenchida com informações fictícias exceto a coluna de risco que é preenchida segundo o fluxograma (Figura 3). Após finalização da tabela de sintomas, ela será convertida em um arquivo CSV.

Figura 4 – Amostra da tabela de sintomas vs. pacientes baseada no protocolo de Manchester.

P	Q	R	S	T	U	V	W
TP	DGSI1	SAT2	COR	DGSI2	TS	SG	RISCO
0	1	1	0	0	0	1	1
1	1	0	0	0	1	0	1
0	1	1	0	0	0	1	1
1	1	0	0	0	0	0	1
0	1	1	0	0	0	1	1
1	1	0	0	0	0	0	1
0	1	1	0	0	0	1	1
1	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
1	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
1	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
1	0	0	0	0	0	0	1
0	0	1	0	0	0	1	2
1	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	5
0	0	0	0	0	0	0	1

Fonte: Autores

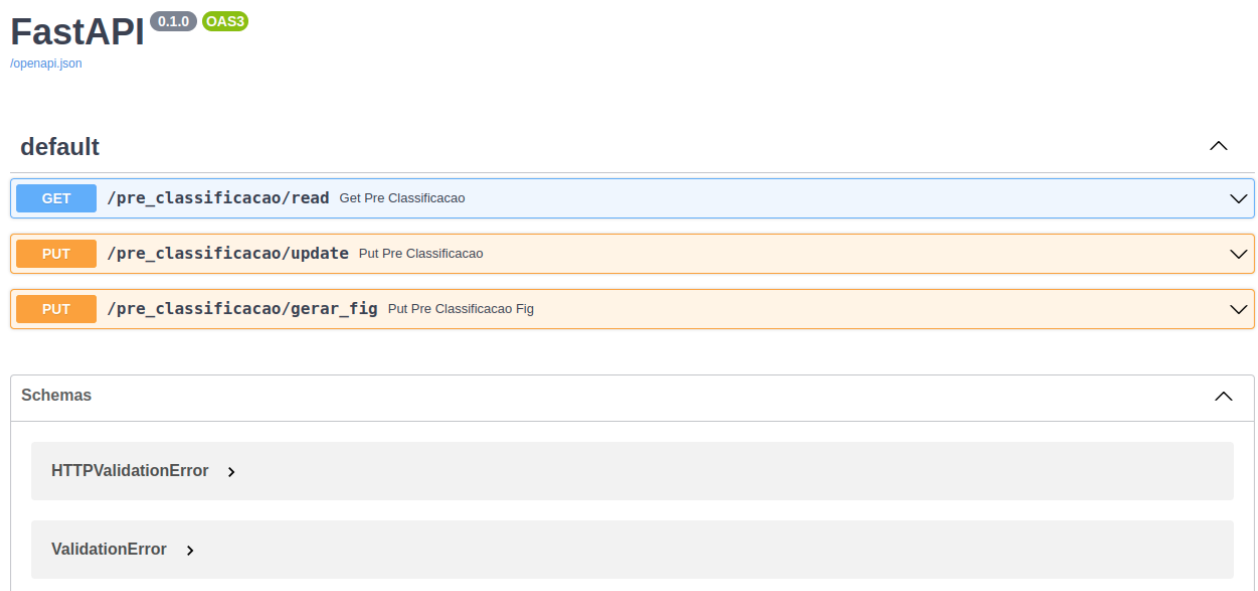
A princípio a maioria dos parâmetros descritos na tabela são do tipo booleano (Figura 4), mas ao decorrer do desenvolvimento do software os parâmetros não serão restritos a um valor binário, mas sim a um range de valores de 1 a 5, no qual o valor 1 representa que o sintoma está presente no paciente e 5 que o sintoma não está presente, os outros valores (2,3 e 4) representam possibilidades do sintoma existir sendo 4 uma possibilidade baixa e 2 uma grande possibilidade do mesmo existir. Também existe campos que registram valores numéricos baseados numa unidade de medida, como por exemplo FC (frequência cardíaca), esses campos serão mantidos para o projeto final.

4.2.3 DESENVOLVIMENTO DE API'S RESTFULL

O protótipo foi desenvolvido utilizando a linguagem de programação Python (como servidor). O software será uma aplicação web no qual será utilizado a linguagem de programação javascript (como cliente), então existe a necessidade de utilizar uma interface que faz a união entre o servidor e o cliente, que no caso serão as API's (Interface de Programação de Aplicação). Para este projeto, será utilizado o framework FastAPI por ser uma ferramenta de uso simples capaz de gerar automaticamente um swagger para análise.

Três endpoints foram criados para o desenvolvimento da API (Figura 5).

Figura 5 – Swagger da api do protótipo.



Fonte: Autores

Endpoint/pre_classificação/read - Este endpoint tem como objetivo fornecer o resultado do risco em que o paciente se encontra em um range de 1 a 5, sendo 1 - emergência, 2 - muito urgente, 3 - urgente, 4 - pouco urgente e 5 - não urgente, de acordo com o protocolo de Manchester. Para o endpoint fornecer os dados necessários, primeiro é preciso passar algumas informações para ele em formato de query. Os dados que o endpoint precisa receber são os mesmos dados utilizados para descrever as colunas na tabela de pacientes (Figura 6 e 7). Por padrão, o endpoint utiliza os dados de uma pessoa saudável para identificar o risco de um paciente, porém o mesmo pode ser editado para obtenção de um novo resultado.

Figura 6 – Amostra das entradas necessárias (query) do endpoint/pre_classificacao/read método GET.

The screenshot shows a REST client interface with a header bar indicating the method is GET and the endpoint is /pre_classificacao/read. Below the header, there is a 'Parameters' section with a 'Cancel' button. The parameters are listed in a table with columns 'Name' and 'Description'. Each parameter has a text input field with the value '0'.

Name	Description
OVA integer (query)	0
DISP integer (query)	80
FR integer (query)	0
CIA integer (query)	0
RAA1 integer (query)	0
MVUA integer (query)	0

Fonte: Autores

Figura 7 – Resultado obtido após consultar o endpoint/pre_classificacao/read (método GET).

The screenshot shows the result of a GET request to the endpoint /pre_classificacao/read. The interface includes a 'SG' parameter with a value of 0, an 'Execute' button, and a 'Clear' button. Below these, the 'Responses' section shows the request details and the server response.

SG
integer
(query)
0

Execute **Clear**

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/pre_classificacao/read?OVA=0&DISP=80&FR=0&CIA=0&RAA1=0&MVUA=0&ROVA=0&ER=0&CAG=0&RAA2=0&MVPA1=0&HEM=0&SAT1=0&TPEH=0&MVPA2=0&TP=0&DGS11=0&S' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/pre_classificacao/read?OVA=0&DISP=80&FR=0&CIA=0&RAA1=0&MVUA=0&ROVA=0&ER=0&CAG=0&RAA2=0&MVPA1=0&HEM=0&SAT1=0&TPEH=0&MVPA2=0&TP=0&DGS11=0&SAT2=0&COR=0&DGS12=0&TS=0&SG=0
```

Server response

Code	Details
200	Response body 0

Download

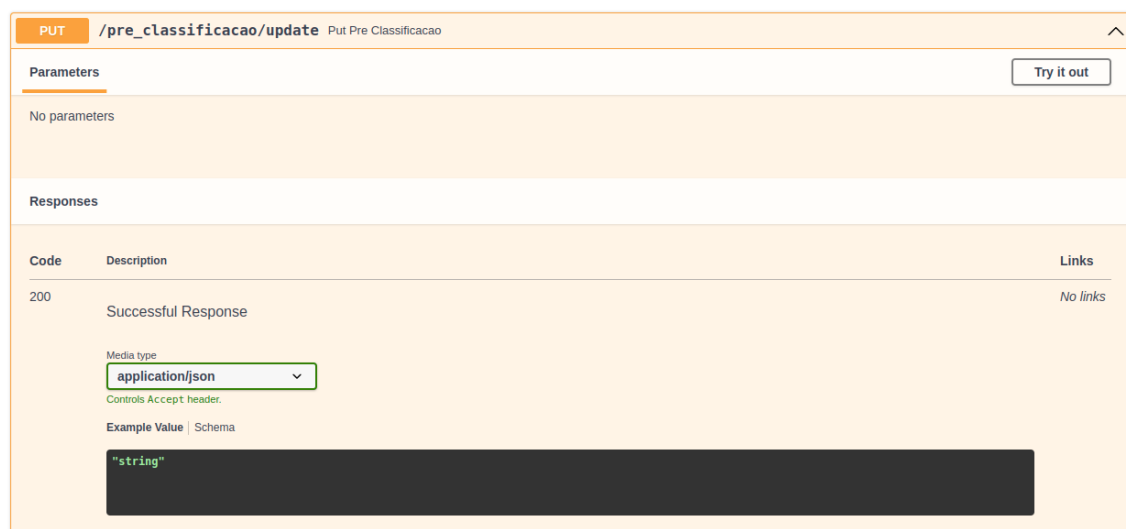
Fonte: Autores

Na figura 6 é observado os dados de entrada do endpoint, onde o usuário irá setar os dados do paciente. Na figura 7 é observado a saída que terá os valores de 0 a 5, sendo 0 falha na API e os valores de 1 a 5 a classificação do paciente.

Endpoint/pre_classificação/update - Este endpoint tem como objetivo atualizar os testes com os novos dados inseridos na tabela de pacientes (Figura 8). Para o protótipo, esta atualização é feita de forma manual, mas no projeto final haverá um mecanismo que utilizará o endpoint diariamente, sendo este o endpoint que ativa o treinamento do algoritmo Random Forest.

Sua saída é representado com um valor binário, 0 para falha na atualização e 1 para sucesso na atualização. Não há necessidade de realizar input de dados no endpoint.

Figura 8 – Endpoint/pre_classificacao/update para treinamento do algoritmo (método PUT).



Fonte: Autores

Endpoint/pre_classificação/gerar_fig - Este endpoint tem como objetivo gerar árvores de decisão utilizadas como a base da tomada de decisão do software (Figura 9). Estas serão geradas dentro do diretório "anexos do projeto". O protótipo foi setado para utilizar 10 árvores de decisão como base para determinar o risco dos pacientes. Este endpoint existe para facilitar a consulta e avaliação de um especialista.

Sua saída é representado com um valor binário, 0 para falha na geração e 1 para sucesso na geração. Não há necessidade de realizar input de dados no endpoint.

Figura 9 – Endpoint/pre_classificacao/gerar_fig para geração de árvores de decisão (método PUT).



Fonte: Autores

4.2.4 IMPLEMENTAÇÃO DO ALGORITMO RANDOM FOREST

A escolha do algoritmo Random Forest foi baseada exclusivamente no fato de que este método garante que a legibilidade humana ocorra já que o mesmo se baseia na estrutura de uma árvore de decisão que é simples para entendimento, diferente de uma Rede Neural Artificial (RNA) que tem uma lógica baseada numa caixa preta não compreensível para humanos, e o algoritmo Random Forest também é adaptável quando novas informações são adicionadas ou corrigidas na estrutura do modelo.

O algoritmo Random Forest utilizado no protótipo foi retirado a partir da documentação scikit-learn na seção RandomForestClassifier, que é um algoritmo Random Forest que utiliza árvores de decisão do tipo classificação para tomada de decisão.

Para o protótipo, foi instanciado um objeto a partir da classe "RandomForestClassifier", utilizando os atributos padrões do mesmo, com exceção dos atributos "max_depth", "random_state" e "n_estimators" que foram setados com os valores 22, 61658 e 10, respectivamente, para alcançar a acurácia desejada.

O atributo "max_depth" controla o tamanho da profundidade das árvores de decisão geradas. Foi adotada uma profundidade de vinte e dois níveis a partir do nó principal. Este foi o valor atribuído devido ao número de variáveis presentes na tabela de pacientes, que são 22. A decisão foi tomada apenas para o protótipo.

Figura 10 – Funções utilizadas no desenvolvimento Backend do protótipo,
 "gerar_arvore" realiza o treinamento com base nos dados levantados e
 "analisar_risco" classifica um paciente de acordo com as árvores geradas.

```

39 def gerar_arvore():
40     X = df.drop('RISCO',axis=1)
41     y = df['RISCO']
42     global rf
43
44     X_tr, X_ts, y_tr, y_ts = train_test_split(X,y, test_size=0.30, random_state=61658)
45
46     rf = RandomForestClassifier(max_depth=22, random_state=61658, n_estimators=10)
47     rf.fit(X_tr, y_tr)
48
49 > def gerar_img(): ...
58
59 def analisar_risco(data = dadosPaciente()):
60     amostra = [
61         data.OVA,
62         data.DISP,
63         data.FR,
64         data.CIA,
65         data.RAA1,
66         data.MVUA,
67         data.ROVA,
68         data.ER,
69         data.CAG,
70         data.RAA2,
71         data.MVPA1,
72         data.HEM,
73         data.SAT1,
74         data.TPEH,
75         data.MVPA2,
76         data.TP,
77         data.DGSI1,
78         data.SAT2,
79         data.COR,
80         data.DGSI2,
81         data.TS,
82         data.SG,
83     ]
84     global rf
85     if rf is None:
86         return 0
87     data = rf.predict(amostra)
88     return int(data[0])
89

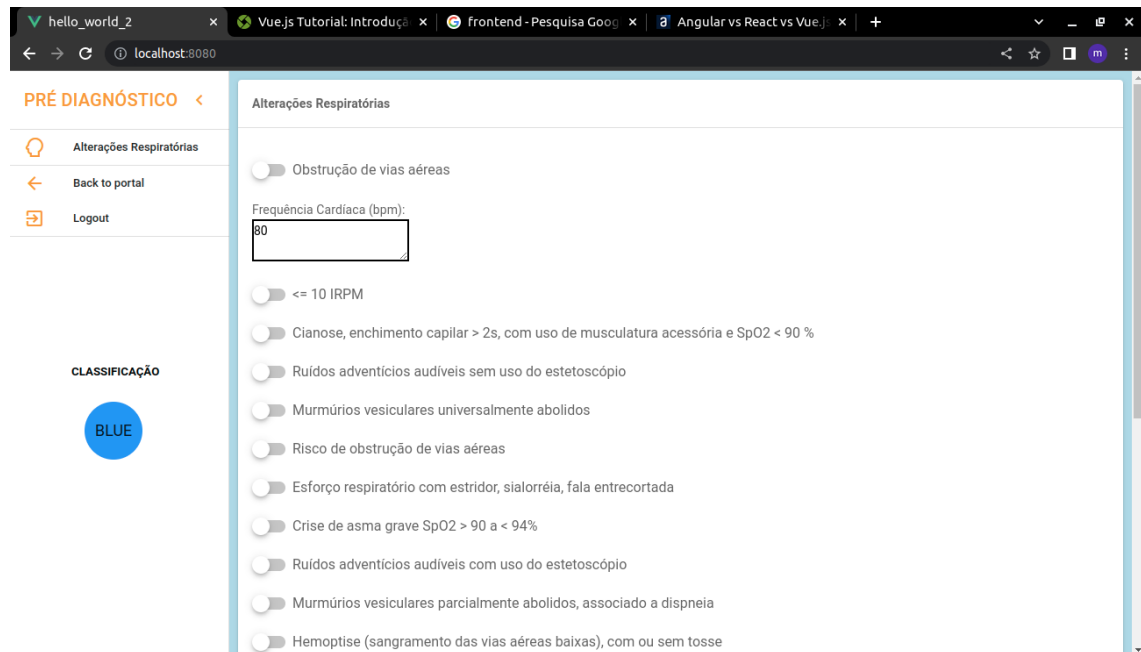
```

Fonte: Autores

O atributo "random_state" controla tanto a aleatoriedade do bootstrap das amostras usadas ao construir árvores quanto a amostragem dos recursos a serem considerados ao procurar a melhor divisão em cada nó. Para esse atributo foi decidido utilizar o valor 61658 para garantir repetibilidade dos testes, no qual outras pessoas podem validar o modelo de dados utilizados para esse projeto. Caso esse valor não seja setado, a classe RandomForestClassifier irá setar um valor aleatório, resultando em um sacrifício na precisão do resultado obtido.

O atributo "n_estimators" indica quantas árvores de decisão serão formadas a partir dos dados de testes fornecidos. Para o protótipo em questão foi estabelecido a geração de dez árvores.

Figura 12 – Modelo de interface usuário e protótipo.



Fonte: Autores

Na interface do software, o menu de navegação está localizado no canto esquerdo, onde é possível encontrar todas as classes disponíveis de sintomas. No momento, apenas a classe 'Alterações Respiratórias' está ativa. Nesse mesmo menu, os usuários têm a opção de fazer logout do software clicando em 'Logout' ou retornar ao portal do hospital com o botão 'Back to portal'.

A classificação de risco, que é o foco principal do projeto, está localizada abaixo do menu de navegação. Essa classificação é responsável por informar aos usuários se o paciente em análise está em risco de vida ou não, oferecendo uma avaliação crucial.

Na parte principal da tela, no canto direito, estão listados todos os sintomas possíveis para a classe de sintomas selecionada. É nessa área que o software espera receber os inputs dos usuários sobre o estado atual do paciente. Sempre que o usuário editar qualquer campo nessa tela, o software realizará automaticamente o processamento dos dados e calculará o risco em que o paciente se encontra, conforme indicado no menu de navegação do lado esquerdo, conforme explicado anteriormente.

Durante a fase de desenvolvimento, a interface do usuário será detalhada com maior precisão, incluindo mais informações e elementos específicos.

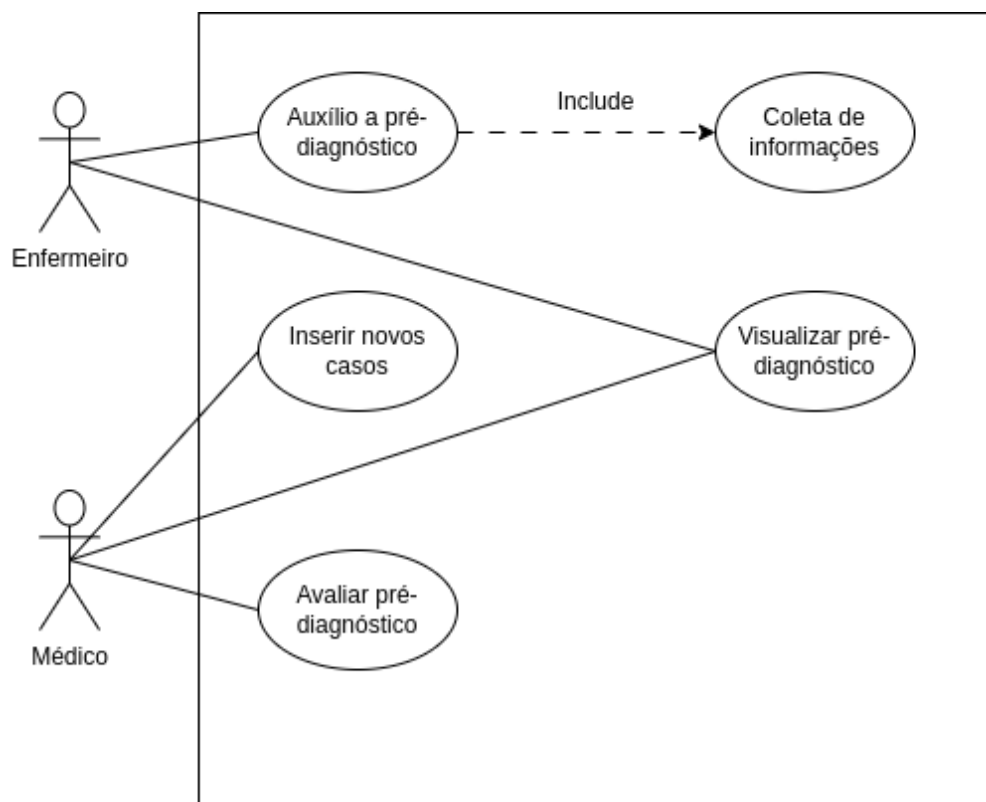
4.3 LEVANTAMENTO E ANÁLISE DE REQUISITOS

Neste tópico, serão apresentados os requisitos funcionais e não funcionais de um software. Para capturar esses requisitos de forma clara e precisa, serão utilizados diagramas UML (Linguagem de Modelagem Unificada). Os diagramas escolhidos para essa tarefa são o Diagrama de Casos de Uso e o Diagrama de Classes, com o objetivo de criar um modelo de sistema orientado a objetos eficiente e abrangente. Essa abordagem ajudará a compreender e comunicar os requisitos do software de maneira visual e estruturada.

4.3.1 CASOS DE USO

A descrição de um modelo de caso de uso envolve a interação entre diferentes usuários e o sistema para resolver um problema específico. Por meio dele se obtém os requisitos de um sistema, a seguir segue o casos de uso (Figura 13) desse projeto mais seu descritivo.

Figura 13 – Casos de Uso.



Fonte: Autores

UC001 Auxílio ao Pré-Diagnóstico Identificação

UC001 - Auxílio ao Pré-Diagnóstico

Função

O Enfermeiro utiliza o software para auxiliar no processo de diagnóstico do paciente.

Ator

Enfermeiro

Pré-condição

Paciente disponível para a avaliação.

Pós-condição

Diagnóstico do paciente auxiliado pelo software.

Fluxo Principal

- a) O Enfermeiro inicia o processo de diagnóstico com o paciente coletando as informações.
- b) O Enfermeiro analisa as informações coletadas e utiliza as ferramentas do software para auxiliar no diagnóstico do paciente.
- c) O Software gera recomendações e sugestões de diagnóstico com base nas informações fornecidas.
- d) O Enfermeiro avalia as recomendações geradas pelo software e utiliza seu conhecimento profissional para realizar o diagnóstico final.
- e) O Enfermeiro registra o diagnóstico final no sistema.

Extensões

Nenhuma.

Inclusões

UC002 - Coleta de informações.

UC002 Coleta de Informações**Identificação**

UC002 - Coleta de Informações

Função

O Enfermeiro coleta informações do paciente durante a consulta.

Ator

Enfermeiro

Pré-condição

Consulta em andamento.

Pós-condição

Informações do paciente registradas.

Fluxo Principal

- a) O Enfermeiro faz perguntas ao paciente sobre sintomas, histórico médico, estilo de vida e outras informações relevantes.
- b) O Enfermeiro registra as respostas do paciente no sistema.

Extensões

Nenhuma.

Inclusões

Nenhuma.

UC003 Inserir Novos Casos**Identificação**

UC003 - Inserir Novos Casos

Função

O Enfermeiro ou Médico insere novos sintomas no sistema para auxílio ao pré-diagnóstico.

Ator

Médico

Pré-condição

Nenhum.

Pós-condição

Novo sintoma registrado no sistema.

Fluxo Principal

- a) O Enfermeiro ou Médico seleciona a opção de inserir novo sintoma no sistema.
- b) O Enfermeiro ou Médico fornece as informações do sintoma (sintoma, descrição, condição).
- c) O Enfermeiro ou Médico salva as informações do novo sintoma no sistema.

Extensões

Nenhuma.

Inclusões

Nenhuma.

UC004 Visualizar Pré-Diagnóstico Identificação

UC004 - Visualizar Pré-Diagnóstico

Função

O Enfermeiro ou Médico visualiza o pré-diagnóstico de um paciente específico.

Ator

Enfermeiro, Médico

Pré-condição

Pré-diagnóstico do paciente disponível no sistema.

Pós-condição

Pré-diagnóstico do paciente visualizado pelo Enfermeiro ou Médico.

Fluxo Principal

- a) O Enfermeiro ou Médico seleciona a opção de visualizar pré-diagnóstico.
- b) O Enfermeiro ou Médico busca e seleciona o paciente desejado.
- c) O sistema exibe o pré-diagnóstico do paciente.

Extensões

Nenhuma.

Inclusões

Nenhuma.

UC005 Avaliar Pré-Diagnóstico Identificação

UC005 - Avaliar Pré-Diagnóstico

Função

O Médico avalia o pré-diagnóstico de um paciente e realiza ajustes ou confirma o diagnóstico final.

Ator

Médico

Pré-condição

Pré-diagnóstico do paciente disponível no sistema.

Pós-condição

Avaliação do pré-diagnóstico realizada pelo Médico.

Fluxo Principal

- a) O enfermeiro identifica um erro na avaliação de risco informado pelo software.
- b) O Médico busca e seleciona o paciente desejado.
- c) O sistema exibe a classificação de risco do paciente.
- d) O Médico avalia as informações apresentadas, considerando seu conhecimento profissional.
- e) O Médico pode realizar ajustes ou confirmar o diagnóstico final com base no pré-diagnóstico apresentado.
- f) O Médico registra o diagnóstico final no sistema.

Extensões

Nenhuma.

Inclusões

Nenhuma.

Com o caso de uso detalhado foi obtido os requisitos funcionais e não funcionais do software apresentado nas tabelas 2 e 3 respectivamente.

Tabela 2 – Tabela de Requisitos Funcionais

Número	Requisito Funcional
RF-01	Classificação de pacientes com base em critérios pré-definidos para priorizar a triagem (Obtido dos casos de uso UC-001, UC-002 e UC-004).
RF-02	Adição de novos sintomas (UC-003).
RF-03	Atualizar classificação de risco dos pacientes (UC-005).

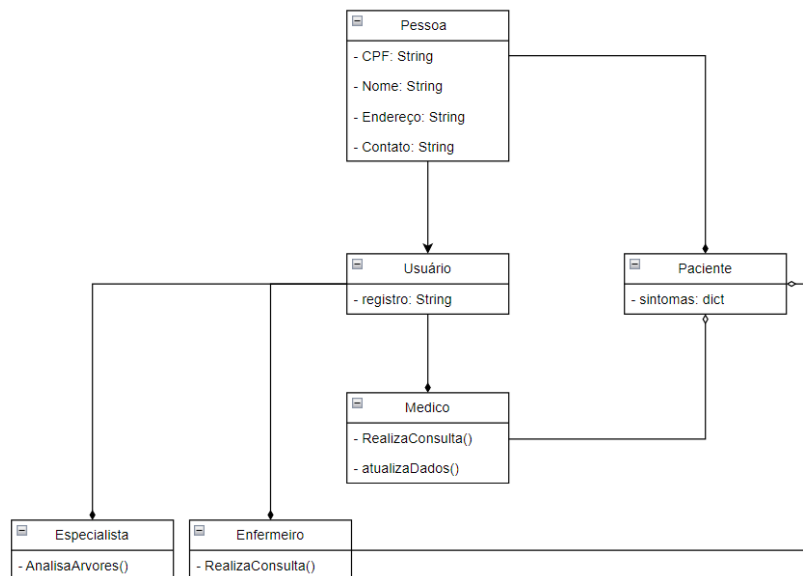
Tabela 3 – Tabela de Requisitos Não Funcionais

Número	Requisito Não Funcional
RNF-01	Segurança: garantir a proteção dos dados pessoais e médicos dos pacientes, seguindo as regulamentações e padrões de segurança.
RNF-02	Desempenho: tempo de resposta rápido para garantir uma triagem eficiente.
RNF-03	Usabilidade: interface do usuário intuitiva e de fácil uso, com recursos de busca e filtros para acesso rápido às informações.
RNF-04	Escalabilidade: dimensionar o sistema para lidar com um grande volume de pacientes e usuários simultâneos.

4.3.2 DIAGRAMA DE CLASSE

O diagrama de classe é uma representação visual da estrutura estática de um sistema orientado a objetos. Ele mostra as classes do sistema, seus atributos, métodos e relacionamentos, sendo amplamente utilizado na engenharia de software para modelar a arquitetura e o design do sistema. A seguir segue o diagrama de classe (Figura 14) do projeto mais seu descritivo.

Figura 14 – Diagrama de classes.



Fonte: Autores

- A classe Pessoa representa informações básicas compartilhadas por todas as pessoas do sistema. É uma classe abstrata, pois não faz sentido ter uma pessoa sem ser um tipo específico.
- A classe Usuario representa o registro de um usuário no sistema, associado ao seu registro.
- As classes Enfermeiro, Médico e Especialista herdam de Usuario e possuem métodos específicos para suas atividades no sistema.
- Os métodos realizaConsulta, atualizaDados e analisaArvores são os principais processos dos atores, conforme especificado nas informações fornecidas.
- As variáveis de instância CPF, nome, endereço e contato em Pessoa são apenas um exemplo de informações que poderiam ser armazenadas. Outras variáveis e métodos podem ser adicionados conforme necessário.

4.4 FASE DE DESENVOLVIMENTO

A fase de desenvolvimento é uma réplica da fase de concepção com adição das seguintes subseções:

- Integração da funcionalidade adicionar sintomas - Funcionalidade que permiti a inserção de novos sintomas no banco de dados.
- Integração da funcionalidade adicionar dados de paciente no banco de dados - Funcionalidade que permiti adicionar dados de novos pacientes, aumentando a precisão das árvores de decisão.

4.4.1 INTEGRAÇÃO DA FUNCIONALIDADE ADICIONAR SINTOMAS

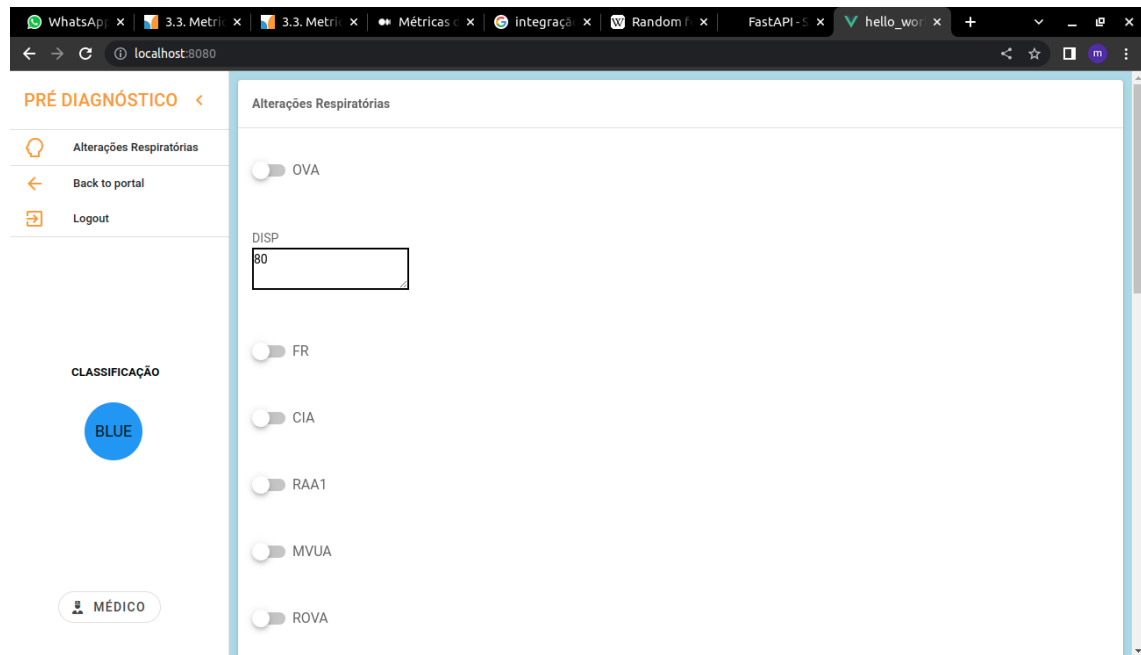
A funcionalidade de adicionar sintomas apresenta outra janela de interface para o usuário. Seu acesso é realizado por intermédio de um botão denominado 'MÉDICO' (Figura 15). Este botão se encontrará desabilitado caso o usuário seja um enfermeiro, com o intuito de evitar que um usuário sem o devido acesso utilize a funcionalidade em questão.

A interface será exibida ao pressionar o botão, possibilitando a adição de um novo paciente ou sintoma (Figura 16). Ao clicar no botão 'SIM' para adicionar um novo sintoma, novos campos são revelados para a inserção de seu nome, descrição e valor default para um paciente

normal. Estes dados serão inseridos no banco de dados ao clicar no botão 'SAVE' (Figuras 17 e 18).

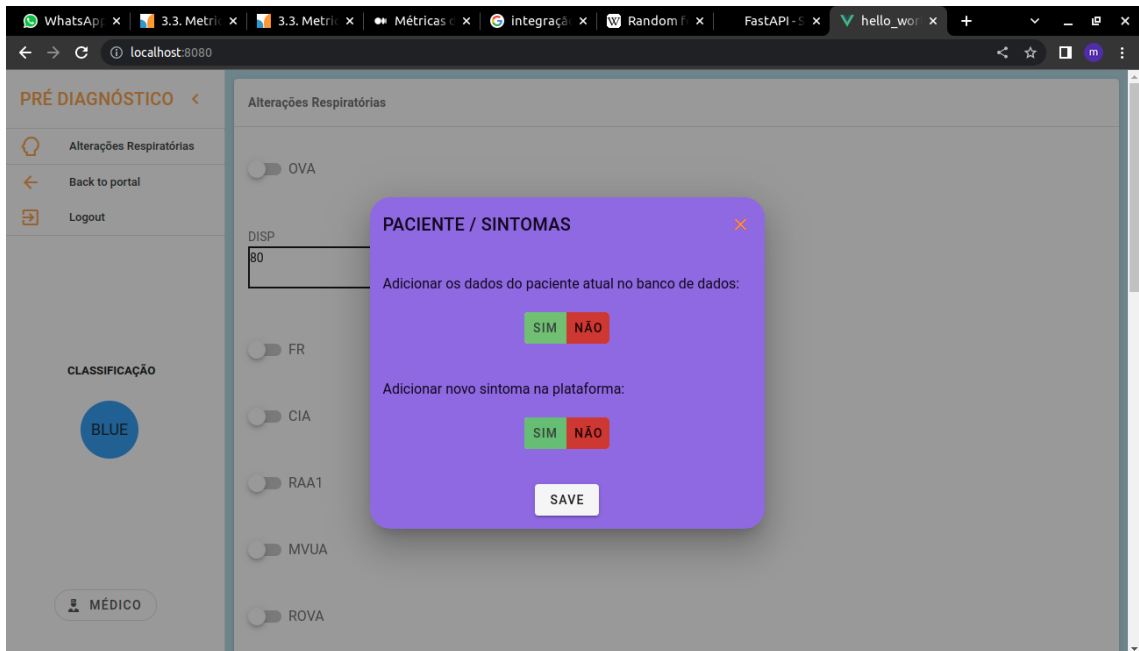
Ao inserir um sintoma, o mesmo será exibido na tela inicial (Figura 19). Como exemplo, é possível observar no CSV que uma nova coluna com a descrição 'CORONA' foi adicionada e cada linha contém o valor default inserido previamente (Figura 20).

Figura 15 – Modelo de interface usuário e protótipo com botão de auxílio ao médico.



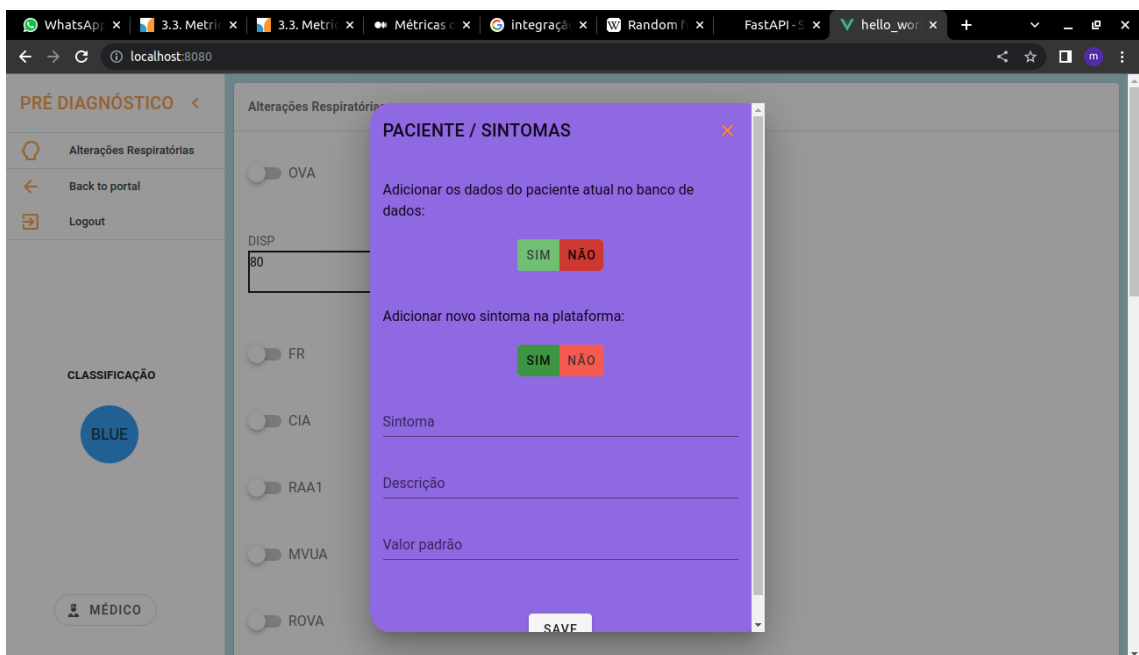
Fonte: Autores

Figura 16 – Interface de usuário e software para adição de pacientes / sintomas.



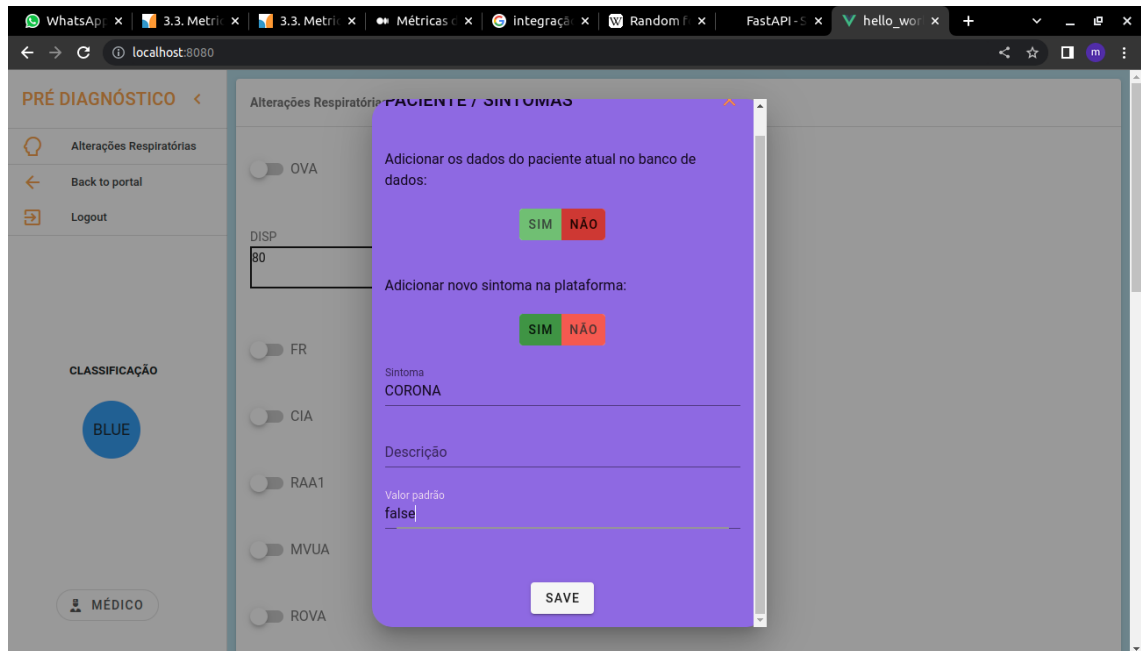
Fonte: Autores

Figura 17 – Campos para adição de sintomas.



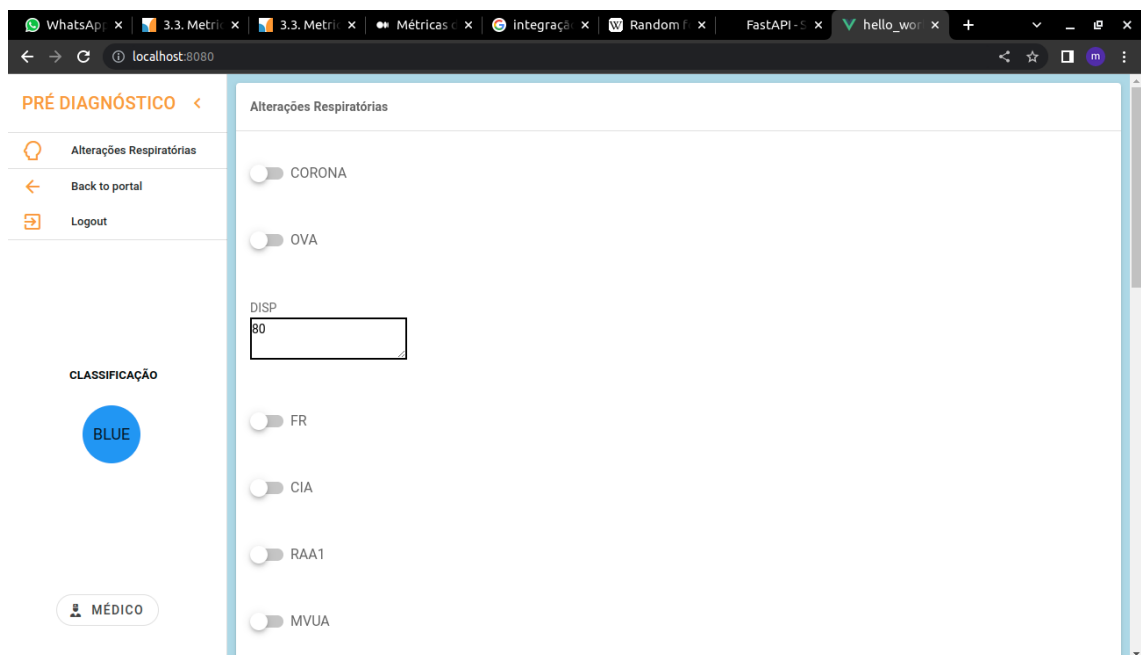
Fonte: Autores

Figura 18 – Campos para adição de sintomas preenchido.



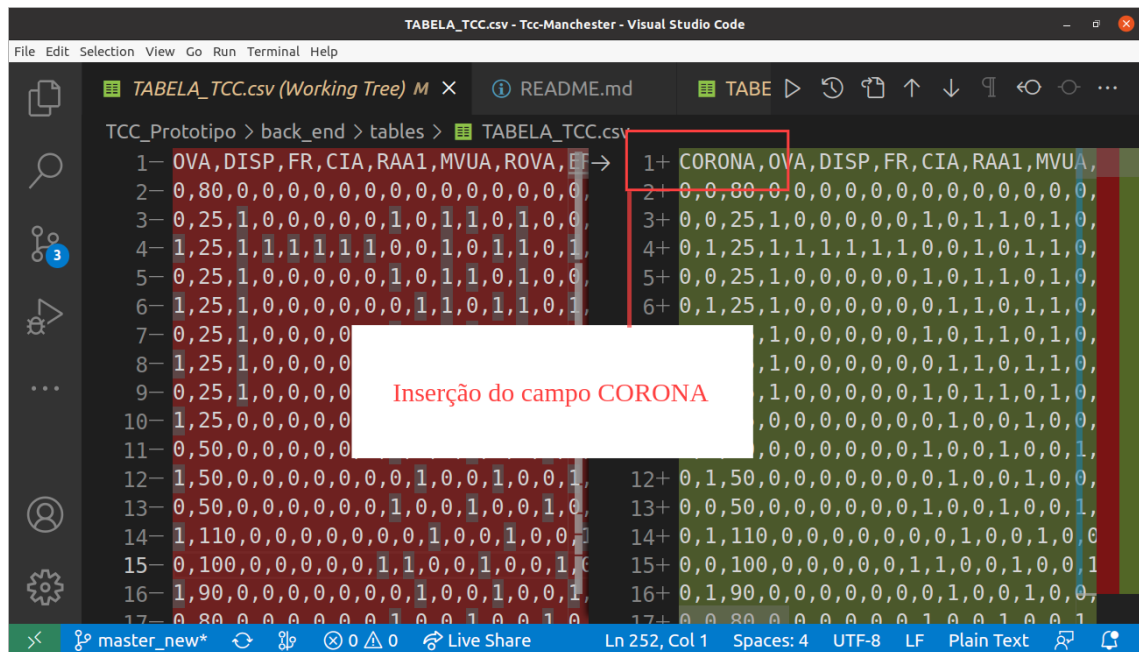
Fonte: Autores

Figura 19 – Modelo de interface usuário com campo preenchido.



Fonte: Autores

Figura 20 – Comparação entre o CSV de entrada (lado esquerdo) versus saída (lado direito) após a inserção de um novo sintoma.



Fonte: Autores

4.4.2 INTEGRAÇÃO DA FUNCIONALIDADE ADICIONAR DADOS DE PACIENTE NO BANCO DE DADOS

A funcionalidade adicionar dados de paciente possui a mesma dinâmica da funcionalidade adicionar sintomas em relação a abrir uma janela a parte para utilizar seus recursos.

O procedimento para utilizar esta função também é feito através do botão 'MÉDICO' e posteriormente no botão 'SIM' no campo responsável por adicionar dados de paciente. Para inserir os dados preenchidos no banco de dados, basta pressionar o botão 'SAVE'.

Após o cumprimento dos passos mencionados, a plataforma irá fazer um novo ensaio com os valores inseridos (Figuras 21 a 23) e consequentemente o risco será alterado de acordo (Figura 24). É possível notar que na comparação entre os dois CSVs (Banco de dados, Figura 25), haverá uma nova linha de dados (lado direito, CSV modificado) que representa um novo paciente classificado com o risco sugerido.

Figura 21 – Interface do usuário e software com dados de um paciente preenchidos.

The screenshot displays a web application interface for patient management. The browser's address bar shows 'localhost:8080'. The left sidebar contains a 'PRÉ DIAGNÓSTICO' section with a 'Back to portal' link and a 'Logout' button. Below this, a 'CLASSIFICAÇÃO' section features a blue circular button labeled 'BLUE' and a 'MÉDICO' button. The main content area, titled 'Alterações Respiratórias', includes a list of respiratory changes with toggle switches: CORONA (checked), OVA, FR, CIA, RAA1, and MVUA. A text input field labeled 'DISP' contains the value '80'.

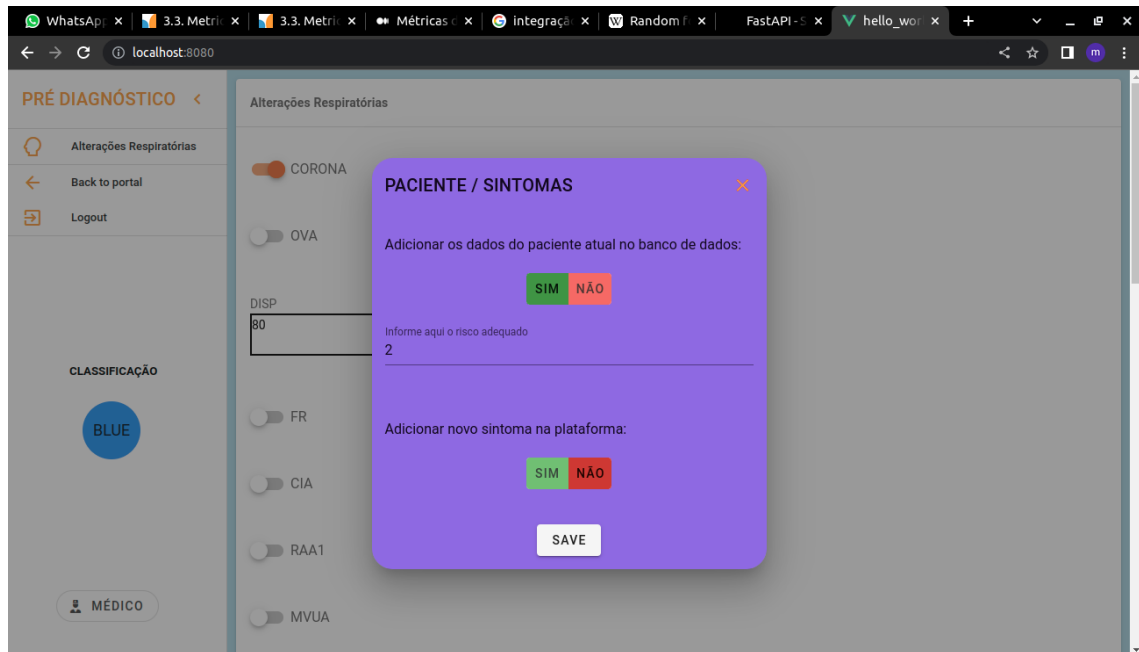
Fonte: Autores.

Figura 22 – Campo para adicionar os dados do paciente preenchidos no exemplo anterior.

This screenshot shows the same web application interface as Figure 21, but with a modal dialog box open in the center. The dialog is titled 'PACIENTE / SINTOMAS' and contains two sections. The first section, 'Adicionar os dados do paciente atual no banco de dados:', has 'SIM' and 'NÃO' buttons. The second section, 'Informe aqui o risco adequado:', has a text input field. Below these sections, there is another 'Adicionar novo sintoma na plataforma:' section with 'SIM' and 'NÃO' buttons, and a 'SAVE' button at the bottom. The background interface is dimmed.

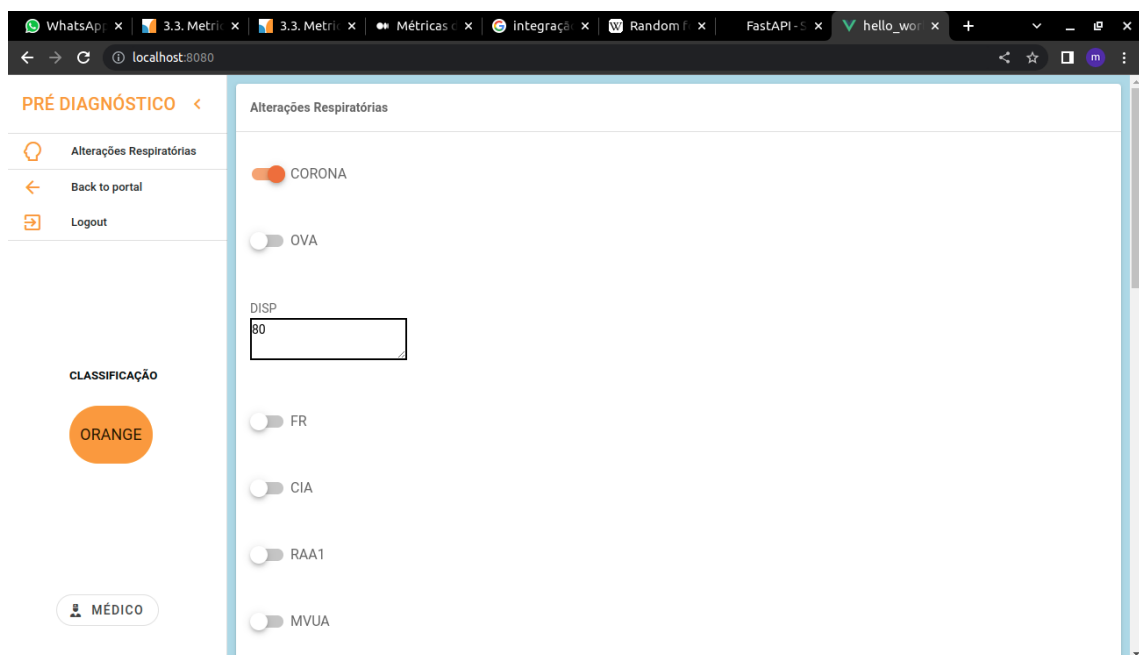
Fonte: Autores

Figura 23 – Campo para atualizar o risco recomendado para o paciente em análise.



Fonte: Autores

Figura 24 – Modelo de interface usuário e software já classificando o paciente com o risco atualizado.



Fonte: Autores

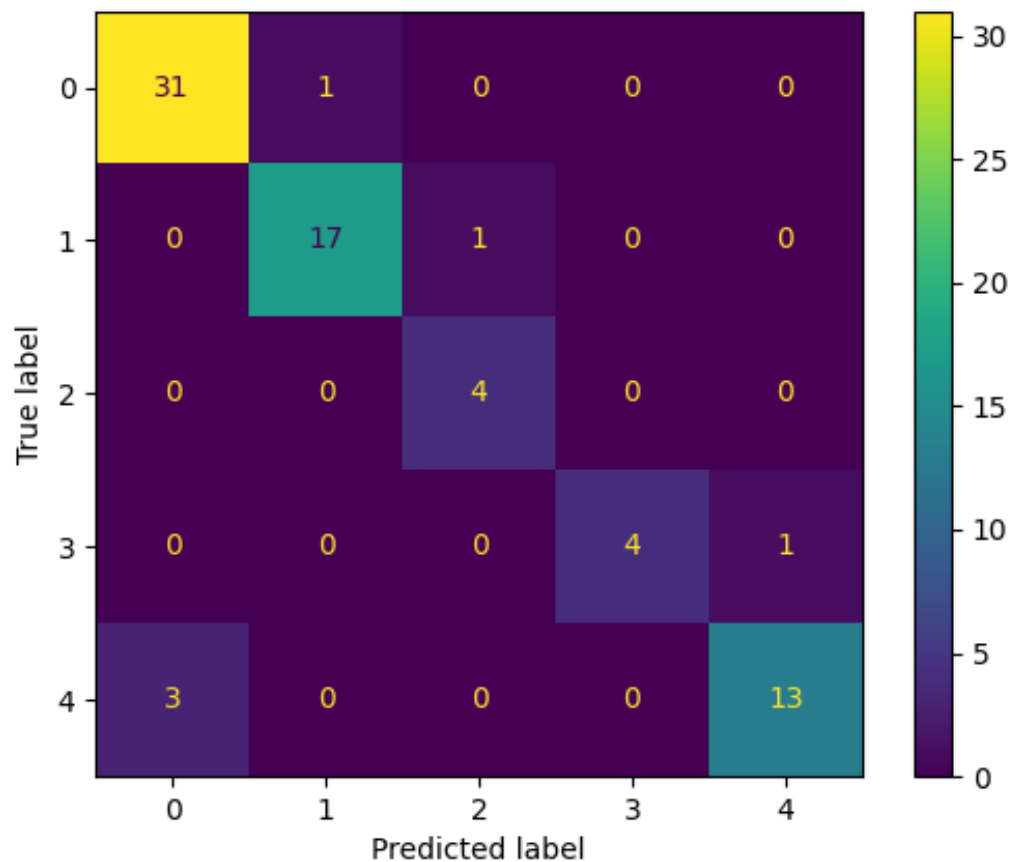
5 RESULTADOS

Este capítulo apresentará os resultados obtidos pela matriz de confusão e das métricas de avaliação de um algoritmo de classificação, divididos em suas seções correspondentes.

5.0.1 MATRIZ DE CONFUSÃO

O software foi alimentado com um modelo de duzentos e cinquenta casos de pacientes gerados de acordo com o Protocolo de Manchester. A partir disso, foi obtida a matriz de confusão representada pela imagem abaixo (Figura 25).

Figura 26 – Matriz de confusão.



Fonte: Autores

Esta matriz indica os erros e acertos do modelo, comparando com o resultado esperado fornecido pelo Protocolo de Manchester. Os índices 0, 1, 2, 3 e 4 exibidos nela simbolizam

as classificações de risco disponíveis para o paciente, sendo elas: vermelho, laranja, amarelo, verde e azul respectivamente.

Vale ressaltar que o Random Forest gera várias árvores de decisão a partir de alguns elementos do modelo, conseqüentemente, serão formadas várias matrizes de confusão, sendo que cada uma representa uma árvore de decisão gerada. A matriz de confusão fornecida no exemplo representado pela figura 25 é uma matriz formada a partir de uma árvore de decisão que utiliza como base um conjunto de 75 elementos.

5.0.2 MÉTRICAS DE AVALIAÇÃO

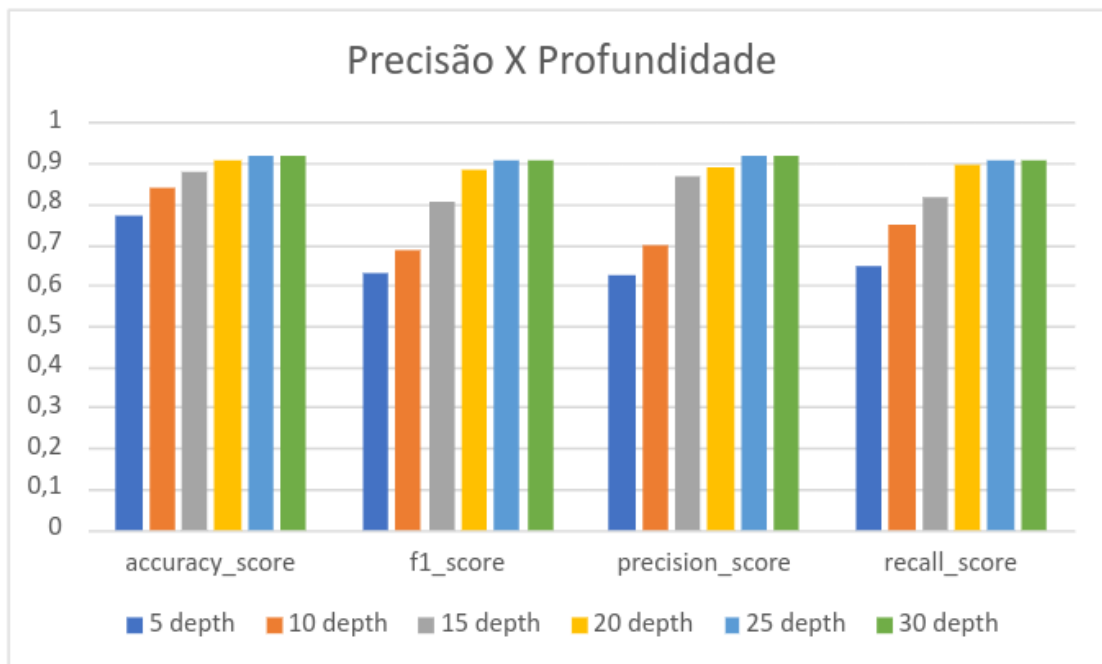
Existem ao todo quatro métricas para problemas de classificação de acordo com a documentação scikit-learn:

- Acurácia: indica uma performance geral do modelo. Dentre todos os casos, quantas o modelo classificou corretamente;
- Precisão: Ao realizar medições de qualquer natureza, ocorrem variações nos processos. Portanto, entende-se como precisão a proximidade entre os valores obtidos pela repetição do processo de medição;
- Recall/Revocação/Sensibilidade: dentre todas as situações de caso positivo como valor esperado, quantas estão corretas;
- F1-Score: média harmônica entre precisão e recall.

Para determinar qual a melhor profundidade de uma árvore de decisão, foi construído um gráfico de porcentagem de 'precisão vs profundidade' da árvore utilizando as métricas estabelecidas (Figura 26). Para construção do gráfico, o número de árvores de decisão foi fixado para 10 no Random Forest.

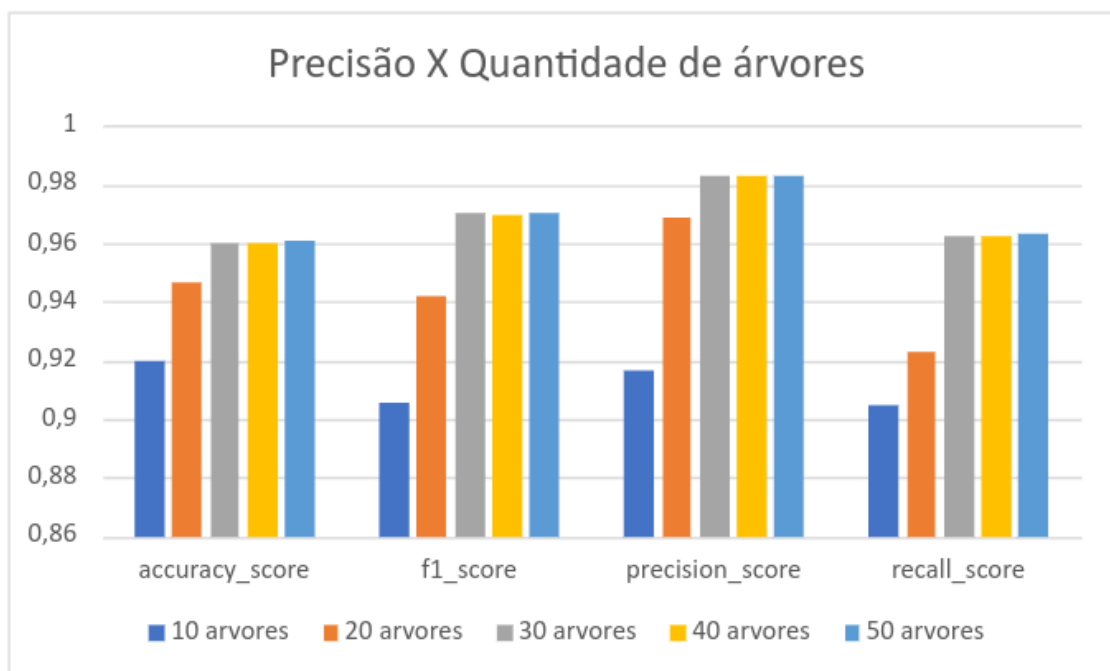
Para determinar quantas árvores serão utilizadas para melhor precisão no Random Forest, foi construído um gráfico de porcentagem de precisão vs quantidade de árvores (Figura 27), para a construção do gráfico foi fixado uma profundidade de 22 níveis.

Figura 27 – Gráfico: Precisão vs. Profundidade.



Fonte: Autores

Figura 28 – Gráfico: Precisão vs. Quantidade de árvores.



Fonte: Autores

6 DISCUSSÃO

O trabalho aqui proposto apresenta um novo método de classificação de risco de pacientes utilizando o algoritmo Random Forest. Uma vez que se trata de um problema não usual na comunidade científica, não será considerada a comparação de trabalhos similares, apenas a comparação de uma classificação realizada pelo homem, já que esta, em território brasileiro, é feita manualmente.

Trata-se, portanto, de uma contribuição relevante, pois a partir dos resultados obtidos é possível fomentar uma discussão sobre os resultados obtidos com outras áreas que utilizam o Random Forest como método de classificação.

Sobre a profundidade da árvore de decisão, foram feitos vários testes entre 5 a 30 níveis de profundidade (Figura 26) e foi determinado que um nível de profundidade de 22 com uma acurácia de 91% (estabelecendo uma quantidade fixa de 10 árvores para o Random Forest) é o valor mais adequado ao projeto proposto. Uma profundidade maior apresentará um rendimento decrescente em sua acurácia.

Já em relação a quantidade de árvores de decisão no Random Forest, foram feitos vários testes de 10 a 50 árvores (Figura 27) e foi determinado que uma quantidade de 30 árvores com uma acurácia de 95,5% (estabelecendo uma quantidade fixa de 22 níveis de profundidade na árvore de decisão, valor obtido na análise anterior) é o valor mais adequado ao projeto proposto. Uma quantidade maior de árvores apresentará um rendimento decrescente em sua acurácia.

7 CONCLUSÃO

O objetivo central deste trabalho foi desenvolver um software com o propósito de auxiliar no pré-diagnóstico de pacientes baseado no Protocolo de Manchester. Para isso, foi criado um protótipo que utiliza o Random Forest e dados pré estabelecidos deste protocolo para classificar o estado do paciente em diferentes categorias.

Com relação à implementação do protótipo, foi possível desenvolver e testar o mesmo, obtendo as respostas esperadas tanto nas funcionalidades do sistema assim como na implementação do algoritmo. Foi possível também observar que o sistema ameniza algumas dificuldades enfrentadas pela avaliação feita com base no Protocolo de Manchester, sendo a principal delas as limitações no que se refere às escalas de avaliação das queixas de baixa complexidade, tornando-as mais precisas.

Os resultados mostram que utilizar o algoritmo de Random Forest para desenvolver um software de pré classificação de risco é mais adequado com valores de 95,5% de acurácia, 97,1% de F1 score, 98% precisão e 96,2% de recall. Vale ressaltar que estes dados foram obtidos através de dados gerados a partir de um Protocolo de Manchester, e não de dados reais. A relação estabelecida é que, com o nível de precisão apresentado extremamente alto mesmo com dados gerados, o resultado do mesmo procedimento realizado com dados de pacientes existentes seria também, extremamente preciso, podendo atingir uma acurácia de até 90%. Apesar da pequena redução na precisão (95,5% para 90%), o software consegue ser superior a uma classificação feita de forma manual pelo homem, que pode alcançar uma precisão de até 68,8% (PEREIRA DA COSTA, JAQUELINE et. al, 2020).

A presente pesquisa apresenta uma análise abrangente das áreas de trabalho futuro relacionadas à classificação de risco no contexto hospitalar. Além do desenvolvimento e implementação do software proposto, foram identificadas diversas direções promissoras para aprimorar a eficácia e a precisão desse processo crucial. Dentre as possibilidades destacam-se o refinamento do algoritmo de classificação de risco com base no Protocolo de Manchester, a exploração de diferentes técnicas de inteligência artificial para aprimorar os resultados obtidos, a integração com dispositivos de monitoramento de pacientes, a condução de estudos clínicos para validar a eficácia do software e a personalização do modelo de classificação de acordo com as particularidades de cada instituição de saúde. Essas áreas de trabalho futuro oferecem um grande potencial para aprimorar o atendimento e a tomada de decisões clínicas, contribuindo para a melhoria contínua dos serviços de saúde e a maximização dos resultados para os pacientes.

A Ciência da Computação desempenha um papel fundamental como um suporte indispensável para a área médica, em especial nos pré-diagnósticos, que ainda carecem de um aproveitamento adequado. A aplicação de tecnologias de informação e inteligência artificial nesse contexto apresenta um potencial considerável. Ao integrar os avanços computacionais com a medicina, surgem inúmeras oportunidades para aprimorar os processos de pré-diagnóstico, otimizando a análise de dados clínicos, possibilitando uma detecção precoce de doenças e promovendo uma tomada de decisão médica mais precisa. Nesse sentido, reconhece-se a relevância da Ciência da Computação como uma aliada estratégica no campo da saúde, impulsionando a pesquisa, desenvolvendo soluções inovadoras e contribuindo significativamente para a melhoria dos resultados em saúde.

8 REFERÊNCIAS

ARIMA, KATIA. **Sensores e inteligência artificial auxiliam equipes médicas na pandemia**, 2020. Disponível em:
<<https://febrabantech.febraban.org.br/temas/inteligencia-artificial/sensores-e-inteligencia-artificial-auxiliam-equipes-medicas-na-pandemia>>

BAO, SUXIA. et al. **Multicenter analysis and a rapid screening model to predict early novel coronavirus pneumonia using a random forest algorithm**, 2021.

CICOLO, EMILIA APARECIDA. **AVALIAÇÃO DO SISTEMA MANCHESTER DE CLASSIFICAÇÃO DE RISCO COM O USO DE REGISTRO ELETRÔNICO E MANUAL**, 2018.

COMO funciona e quais são as classificações do protocolo de Manchester? **feegow**, 2022. Disponível em: <<https://feegowclinic.com.br/protocolo-de-manchester-blog/>>

DAMILANO, CLÁUDIO TEIXEIRA. **Inteligência artificial e inovação tecnológica: as necessárias distinções e seus impactos nas relações de trabalho**, 2019.

DELAFIORI, JEANY. et al. **A Machine Learning Application Based in Random Forest for Integrating Mass Spectrometry-Based Metabolomic Data: A Simple Screening Method for Patients With Zika Virus**, 2018.

EMINA ALICKOVIC; ABDULHAMIT SUBASI. **Medical Decision Support System for Diagnosis of Heart Arrhythmia using DWT and Random Forests Classifier**, 2016.

E. F. MAIA, MARCIO. **Diagramas UML**, 2008. Disponível em:
<<http://disciplinas.lia.ufc.br/engsof081/arquivos/ApresentacaoDiagramas.pdf>>

FastAPI **FastAPI**, 2022. Disponível em: <<https://fastapi.tiangolo.com/>>

GONÇALVES, JÉFERSON DOS SANTOS; MONTICELLI, JEFFERSON MARLON. **Utilização e Aceitabilidade da Inteligência Artificial na Tomada de Decisão Médica**, 2022.

IBM Cloud Education. **Random Forest**, 2020. Disponível em:
<<https://www.ibm.com/cloud/learn/random-forest>>

MANUAL ACCR. **Acolhimento com classificação de risco**, 2022.

MARÍA M. ABAD-GRAU. et al. **Evolution and challenges in the design of computational systems for triage assistance**, 2007.

METRICS AND SCORING: QUANTIFYING THE QUALITY OF PREDICTIONS. **Scikit learn**, 2007 - 2023. Disponível em: <https://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-f-measure-metrics>

NEGRÃO, LEONARDO. **FRAMEWORKS JAVASCRIPT**, 2022. Disponível em:
<<https://www.alura.com.br/artigos/angular-vs-react-vs-vue-js>>

PEREIRA DA COSTA, JAQUELINE. et al. **Acurácia do Sistema de Triagem de Manchester em um serviço de emergência**, 2020.

RANDOM FOREST CLASSIFIER. **Scikit learn**, 2007 - 2023. Disponível em: <<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>>

MATRIZ DE CONFUSÃO. **Scikit learn**, 2007 - 2023. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html>

RESENDE, FABRÍCIA KAROLLYNE SANTOS; INVENCÃO, MARIA ESTELLA SANTOS DA; SILVA, GILTON JOSÉ FERREIRA DA. **Impactos da Inteligência Artificial na Tomada de Decisão Médica: Um Mapeamento Sistemático**, 2021.

VUE.JS Tutorial **DEVMEDIA**, 2022. Disponível em: <<https://www.devmedia.com.br/vue-js-tutorial/38042>>

APÊNDICE A - PARÂMETROS E ATRIBUTOS DA CLASSE RANDOM FOREST CLASSIFIER

Parâmetros:

n_estimators: int, default = 100 - Número de árvores a serem geradas;
criterion{“gini”, “entropy”, “log_loss”}, default = “gini” - Seleciona a função para medir a qualidade de uma divisão;

max_depth: int, default = None - Profundidade máxima da árvore, os nós iram expandir até que todas as folhas sejam puras ou contenham menos que min_samples_split amostras;

min_samples_split : int or float, default = 2 - Número mínimo para dividir um nó interno;

min_samples_leaf : int or float, default = 1 - Número mínimo de amostras para estar em um nó folha;

min_weight_fraction_leaf: float, default = 0.0 - Fração ponderada mínima da soma de todos os pesos, de todas as amostras de entrada, necessárias para estar em um nó folha;

max_features{“sqrt”, “log2”, None, int or float, default = “sqrt” - Número de recursos a serem considerados quando procurar a melhor divisão;

max_leaf_nodes : int, default = None - Cultiva a árvore com max_leaf_nodes da melhor maneira. Os melhores nós são definidos como redução relativa na impureza;

min_impurity_decrease : float, default = 0.0 - Um nó será dividido se esta divisão induzir uma diminuição da impureza maior ou igual a este valor;

Bootstrap : bool, default = True - Se as amostras bootstrap são usadas ao construir árvores. Se False, todo o conjunto de dados é usado para construir cada árvore;

oob_score : bool, default = False - Deve usar amostras prontas para estimar uma pontuação de generalização, estando disponível apenas se bootstrap = True;

n_jobs : int, default = None - Número de trabalhos a serem executados em paralelo;

random_state : int, RandomState instance or None, default = None - Controla a aleatoriedade do bootstrap das amostras usadas ao construir árvores e a amostragem dos recursos que são considerados quando procura a melhor divisão de cada nó;

verbose : int, default = 0 - Controla a verbosidade ao ajustar e prever;

warm_start : bool, default = False - Se definido como True, reutiliza a solução da chamada anterior para ajustar e adicionar mais estimadores ao conjunto, caso contrário, somente ajusta uma nova;

class_weight : {“balanced”, “balanced_subsample”}, dict or list of dicts, default = None - Pesos associados a classes no formato {class.label: weight}. Caso não for fornecido, todas as classes devem ter peso um. Quanto tiver múltiplas saídas, uma lista de dicts pode ser fornecida na mesma ordem das colunas de y;

ccp_alpha : non-negative float, default = 0.0 - Parâmetro de complexidade usado para a redução de complexidade de custo mínimo;

max_samples : int or float, default = None - Caso o bootstrap for True, o número de amostras a serem extraídas de X para treinar cada estimador de base;

base_estimator_: **DecisionTreeClassifier** - O template estimador filho usado para criar a coleção de sub estimadores utilizados;

estimators_: **lista de DecisionTreeClassifier** - A coleção dos sub estimadores utilizados;

classes_: **ndarray de shape (n_classes,)** ou **a lista destes arrays** - O label da classe (problema de output único), ou uma lista de arrays de label de classes (problema de output múltiplo);

n_classes_: **int ou lista** - O número de classes (problema de output único) ou a lista contendo o número de classes para cada output (problema de output múltiplo);

n_features_in_: **int** - Número de features durante “fit”.

Métodos:

apply(X) - Aplica a árvore dentro da variedade de árvores para X, retornando o índice de folhas;

decision_path(X) - Retorna a decisão da variedade de árvores geradas;

Fit - Gera uma variedade de árvores a partir do conjunto de treinamento;

get_params - Obtém o parâmetro para o estimador;

predict(X) - Faz uma previsão para classe;

predict_log_proba(X) - Prevê a probabilidade de log para classe;

predict_proba(X) - Prevê a probabilidade de classe;

score - Retorna a precisão média dos dados e rótulos de teste;

set_params - Define os parâmetros do estimador.