



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I - CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
CURSO DE GRADUAÇÃO EM BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

ESTÊVÃO ARRUDA BORBA SANTIAGO GUIMARÃES

**O DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA O
GERENCIAMENTO HOSPITALAR**

**CAMPINA GRANDE
2024**

**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I - CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA**

ESTÊVÃO ARRUDA BORBA SANTIAGO GUIMARÃES

**O DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA O
GERENCIAMENTO HOSPITALAR**

Relatório de conclusão de curso apresentado ao Curso de Bacharelado em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de bacharel em Ciência da Computação.

Orientador: Prof. Dr. Eduardo Jorge Valadares Oliveira.

**CAMPINA GRANDE
2024**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

G963d Guimaraes, Estevao Arruda Borba Santiago.
O desenvolvimento de uma aplicação web para o gerenciamento hospitalar [manuscrito] / Estevao Arruda Borba Santiago Guimaraes. - 2024.
76 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2024.

"Orientação : Prof. Dr. Eduardo Jorge Valadares Oliveira, Coordenação do Curso de Computação - CCT. "

1. Sistema hospitalar. 2. Gestão hospitalar. 3. Aplicação web. I. Título

21. ed. CDD 005

ESTÊVÃO ARRUDA BORBA SANTIAGO GUIMARÃES

O DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA O GERENCIAMENTO
HOSPITALAR

Relatório de conclusão de curso apresentado ao Curso de Bacharelado em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de bacharel em Ciência da Computação.

Área de concentração: Desenvolvimento de software.

Aprovado em: 06/03/2024.

BANCA EXAMINADORA



Prof. Dr. Eduardo Jorge Valadares Oliveira (Orientador)
Universidade Estadual da Paraíba (UEPB)



Prof. Dr. Misael Elias de Moraes
Universidade Estadual da Paraíba (UEPB)



Prof. Dra. Kátia Elizabete Galdino
Universidade Estadual da Paraíba (UEPB)

AGRADECIMENTOS

Agradeço primeiramente a Deus, por estar aqui vivenciando essa experiência engrandecedora, por guiar os meus caminhos, me proporcionar saúde e me inspirar nos momentos sensíveis.

Agradeço aos meus pais, que são minha base, me deram educação, formaram meu caráter e são a maior prova do amor puro e incondicional, graças a eles eu sou quem sou.

A minha irmã, que sempre esteve comigo nos melhores e piores momentos.

Aos meus familiares que sempre apoiaram, em especial minha avó materna Zélia, que sempre me ensinou a importância de estudar para “ser alguém na vida”.

A minha namorada Amanda, por me tornar minha melhor versão, me ajudando quando meu instinto falou mais alto que minha razão.

A Maria de Lourdes por sempre me aguentar e fazer rir em momentos difíceis.

Aos professores do Curso de Computação da UEPB, em especial, Misael Elias de Moraes que foi minha porta de entrada para o NUTES, permitindo que esse projeto fosse realizado. Agradeço também a Eduardo Jorge Valadares Oliveira e Katia Elizabete Galdino, que contribuíram ao longo da minha jornada, me permitindo alcançar metas inimagináveis.

A todos os funcionários e colegas do NUTES, que participaram de todo o caminho ao meu lado, enfrentando os desafios com muita ternura e dedicação.

RESUMO

Este trabalho visa documentar como foi o desenvolvimento do medNUTES, uma aplicação web com o objetivo de gestão hospitalar desenvolvido no Núcleo de Tecnologias Estratégicas em Saúde na Universidade Estadual da Paraíba. O relatório aborda detalhadamente as fases necessárias para construção do software, iniciando com o levantamento e análise dos requisitos, etapa onde os envolvidos foram in loco nos hospitais para entender os fluxos e as necessidades que a aplicação deveria atender, em seguida aconteceu a fase de projeto do sistema, adotando uma arquitetura que mais se encaixasse nas exigências do problema, definindo a estrutura que os dados seriam mapeados e estabelecendo a forma como a interface iria interagir com o usuário final. Com o objetivo de aumentar a velocidade no processo de desenvolvimento, adotou-se uma metodologia ágil, permitindo que a implementação acontecesse de forma incremental, ou seja, ao final de cada ciclo uma versão executável do sistema era entregue ao usuário. Por meio do uso do software, espera-se que o mesmo afete positivamente a eficiência operacional dos fluxos hospitalares, reduzindo erros, custos, facilitando nos processos básicos e permitindo a gestão dos recursos de maneira eficaz.

Palavras-Chave: sistema hospitalar; gestão hospitalar; aplicação web.

ABSTRACT

This work aims to document how the development of medNUTES was carried out, a web application with the aim of hospital management, developed at the Núcleo de Tecnologias Estratégicas em Saúde at Universidade Estadual da Paraíba. The report addresses in detail the phases necessary to build the software, starting with the survey and analysis of requirements, a stage where those involved went on site in hospitals to understand the flows and needs that the application should meet, followed by the design phase of the system, adopting an architecture that best fits the requirements of the problem, defining the structure in which the data would be mapped and establishing the way in which the interface would interact with the end user. With the aim of increasing speed in the development process, an agile methodology was adopted, allowing implementation to happen incrementally, that is, at the end of each cycle an executable version of the system was delivered to the user. Through the use of the software, it is expected that it will positively affect the operational efficiency of hospital flows, reducing errors and costs, facilitating basic processes and allowing the management of resources effectively.

Keywords: hospital system; hospital management; web application.

LISTA DE ILUSTRAÇÕES

Figura 1 - Módulo de usuários	25
Figura 2 - Módulo de pacientes	27
Figura 3 - Módulo de agendamentos	29
Figura 4 - Módulo de almoxarifado	31
Figura 5 - Módulo de prescrições	33
Figura 6 - Módulo de relatórios	35
Figura 7 - DER referente ao módulo de usuários	42
Figura 8 - DER referente ao módulo de pacientes	43
Figura 9 - DER referente ao módulo de agendamentos	45
Figura 10 - DER referente ao módulo de almoxarifado	46
Figura 11 - Explicação da entidade de movimentação	47
Figura 12 - DER referente ao módulo de prescrições	49
Figura 13 - DER referente ao módulo de relatórios	50
Figura 14 - Definição dos layouts da interface	53
Figura 15 - Paleta de cores	54
Figura 16 - Listagem dos usuários, filtros, ordenação e botão de adicionar	55
Figura 17 - Formulário de usuário no modo de cadastro	55
Figura 18 - Parte do formulário com as permissões do usuário	56
Figura 19 - Listagem dos pacientes, filtros, ordenação e botão de adicionar	57
Figura 20 - Formulário de pacientes no modo de cadastro	57
Figura 21 - Listagem das triagens de um paciente em forma de linha do tempo	58
Figura 22 - Formulário para cadastrar uma nova triagem	59
Figura 23 - Formulário para cadastrar uma nova triagem	60
Figura 24 - Formulário para cadastrar um novo agendamento	60
Figura 25 - Ações de confirmar presença, falta e desmarcar o agendamento	61
Figura 26 - Aba de estoque contendo a listagem dos insumos	62
Figura 27 - Formulário de cadastro de insumo	63
Figura 28 - Formulário de realização de saídas de insumos	63
Figura 29 - Listagem de entradas de insumos	64
Figura 30 - Listagem de saídas de insumos	65
Figura 31 - Listagem de estornos de insumos	65
Figura 32 - Listagem de prescrições	66
Figura 33 - Cadastro de prescrição com tratamento sugerido pelo médico-infectologista	67
Figura 34 - Listagem dos protocolos associados a cada doença	68
Figura 35 - Cadastro de tratamento	68
Figura 36 - Escolha de relatórios	69
Figura 37 - Filtros do relatório de pacientes	70

LISTA DE TABELAS

Tabela 1 - Concepção de problemas relacionados aos pacientes	20
Tabela 2 - Concepção de problemas relacionados às prescrições	21
Tabela 3 - Concepção de problemas relacionados ao almoxarifado	22
Tabela 4 - Concepção de problemas relacionados à gestão	23
Tabela 5 - Requisitos do módulo de usuários	26
Tabela 6 - Requisitos do módulo de pacientes	27
Tabela 7 - Requisitos do módulo de agendamentos	29
Tabela 8 - Requisitos do módulo de almoxarifado	31
Tabela 9 - Requisitos do módulo de prescrições	33
Tabela 10 - Requisitos do módulo de relatórios	35

LISTA DE ABREVIATURAS E SIGLAS

API	Interface de Programação de Aplicação
DER	Diagrama de Entidade-Relacionamento
HTTP	Protocolo de Transferência de Hipertexto
IMC	Índice de Massa Corporal
JSON	JavaScript Object Notation
MD	Material Design
MVC	Model-view-controller
NUTES	Núcleo de Tecnologias Estratégicas em Saúde
PSQL	PostgreSQL
REST	Representational State Transfer
SPA	Single-page application
SUS	Sistema Único de Saúde
UI	Interface do Usuário
UX	Experiência do Usuário
WebApp	Aplicação Web
XML	Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	10
2	REVISÃO DE LITERATURA	12
3	METODOLOGIA	18
3.1	<i>LEVANTAMENTO E ANÁLISE DE REQUISITOS</i>	18
3.1.1	<i>Levantamento</i>	19
3.1.2	<i>Especificação</i>	23
3.2	<i>PROJETO DO SISTEMA</i>	36
3.2.1	<i>Projeto de arquitetura</i>	37
3.2.2	<i>Projeto de dados</i>	41
3.2.3	<i>Projeto de interface</i>	51
3.3	<i>IMPLEMENTAÇÃO</i>	71
4	CONSIDERAÇÕES FINAIS	73
	REFERÊNCIAS	75

1 INTRODUÇÃO

Os hospitais, por serem organizações multifacetadas que exigem práticas de gerenciamento eficientes e sofisticadas, sua gestão, na sociedade atual, torna-se um campo complexo de desafios para os que nele trabalham, sobretudo para os gestores, tendo em vista a necessidade de fornecer assistência técnica de alta qualidade para prevenção e tratamento de problemas de saúde no tocante à marcação de consultas, ao controle do estoque de utensílios hospitalares e priorização de filas de atendimento à população. Entre esses desafios estão conflitos recorrentes entre áreas técnicas e gerenciais, a ampliação da base de clientes potenciais devido ao aumento da população idosa e das doenças crônicas, escassez de recursos médicos e longas esperas por atendimento.

Diante deste cenário complexo, este trabalho visa apresentar o desenvolvimento de um sistema web dedicado ao gerenciamento hospitalar, denominado medNUTES, detalhando todos os passos necessários para o resultado final, incluindo o levantamento e análise dos requisitos, a fase do projeto de arquitetura, dados e interface do sistema, e por fim exemplificando o estágio de implementação. Todos os processos foram feitos no departamento do Núcleo de Tecnologias Estratégicas em Saúde (NUTES) na Universidade Estadual da Paraíba, campus de Campina Grande (PB).

A princípio foi necessário compreender e acompanhar como funcionava o dia-a-dia e os fluxos dentro dos hospitais, todo esse processo foi feito in loco em hospitais na Paraíba, com isso foi possível levantar e mapear todos os requisitos dos usuários para o sistema. Em seguida, foi definido toda a parte de especificação técnica do sistema, resultando na criação de *user stories*, diagramas para entendimento dos fluxos e escolha da estrutura de gerenciamento do projeto, que para esse caso foi aplicado a *framework Scrum*, uma metodologia que se baseia em gestão ágil de projetos. Ainda na especificação técnica do sistema, o próximo passo foi definir as tecnologias relacionadas a codificação da aplicação, como já citado anteriormente a aplicação foi desenvolvida para plataforma web em *Single Page Applications* (SPA), usando linguagem Javascript tanto para o back-end quanto para o front-end. Para armazenar todos os dados de forma segura e eficiente, a base de dados escolhida foi relacional, entregando a aplicação uma ótima flexibilidade, segurança e integridade dos dados.

Subsequente, têm-se a implementação e desenvolvimento do sistema, todos os requisitos e documentos levantados anteriormente foram transformados em tarefas, onde essas

tasks foram implementadas de forma incremental e sequencial, gerando pequenos entregáveis para o cliente.

O uso do medNUTES no gerenciamento hospitalar traz algumas contribuições, como por exemplo, ajudar a definir melhor os processos de trabalho, melhorando a eficiência e reduzindo os erros de prescrições médicas, erro de priorização de atendimento, erro de descontrole de estoque dos materiais existentes e de assessoramento na evolução do paciente. Isso poderia resultar em uma entrega de cuidados de saúde mais eficaz e segura para os pacientes através da digitalização e automatização de tarefas administrativas, e reduzir significativamente a sobrecarga de trabalho dos funcionários, permitindo-lhes concentrar-se mais na prestação de cuidados ao paciente.

Além disso, essa ferramenta pode fornecer uma plataforma para uma gestão mais dialógica, tanto horizontalmente (entre diferentes departamentos ou funções dentro do hospital) quanto verticalmente (entre diferentes níveis de gerência). Isso ajuda a resolver conflitos interpessoais e a melhorar a comunicação e a colaboração dentro do hospital. Um outro fator muito importante que não pode ser negligenciado em qualquer gestão, é a otimização de custos, o controle financeiro integrado ao faturamento e recepção contribui para uma maior eficiência na administração dos recursos. Isso se conecta diretamente à necessidade de criar e acompanhar métricas que funcionam como controle de qualidade do hospital.

Sendo uma aplicação web, essa ferramenta para o gerenciamento hospitalar oferece benefícios adicionais significativos. O acesso em tempo real a informações de qualquer lugar e a qualquer momento permite a mobilidade e a flexibilidade, melhorando a capacidade de resposta às situações emergentes. A integração e a interoperabilidade com outros sistemas e serviços de saúde se tornam mais viáveis, aumentando a eficiência geral e a continuidade dos cuidados. Além disso, a capacidade de realizar atualizações e melhorias regulares no sistema por meio da web pode facilitar a adoção de inovações e manter o sistema atualizado com os mais recentes desenvolvimentos tecnológicos.

Em suma, pelo o que foi exposto acima, o medNUTES traz uma série de benefícios, desde a melhoria da eficiência operacional e da comunicação até a redução de erros e custos, em conjunto com a melhor prestação de cuidados de saúde. No entanto, também é importante lembrar que a implementação bem-sucedida dessa ferramenta exigiu um compromisso significativo da liderança hospitalar, assim como treinamento adequado para o pessoal e a necessidade de lidar com questões de segurança e privacidade dos dados.

2 REVISÃO DE LITERATURA

Na computação, existem vários tipos de aplicações, as mais tradicionais são aplicações desktop, aplicações mobile e aplicações web. Uma aplicação desktop o usuário precisa instalar um software no seu computador para conseguir executá-lo, já uma aplicação mobile também necessita de instalação, porém o produto alvo é diferente, a aplicação deve ser usada em dispositivo eletrônico móvel, como smartphone, tablets, etc. Para evitar a necessidade de instalar um determinado software, têm-se a aplicação web, onde o sistema roda em um servidor remoto e os usuários conseguem acessar a aplicação por meio de um navegador web (VOLLE, 2022) sem necessidade de nenhuma instalação, podendo esse tipo de aplicação ser acessado tanto por computadores e dispositivos móveis. Em um ambiente hospitalar, os usuários precisam de uma maior mobilidade e flexibilidade, necessitando acessar o sistema de forma rápida, sem precisar instalar ou baixar a aplicação, além disso o acesso pode ser feito a partir de dispositivos diversos tais como, celulares, tablets e desktops. Como citado anteriormente, uma aplicação web cumpre todos os critérios anteriores, por esse motivo o medNUTES foi desenvolvido nesse tipo de aplicação.

No desenvolvimento de software uma parte crucial é a escolha da arquitetura, pois ela é responsável por organizar a estrutura dos componentes, mitigando erros e tornando todo o processo de desenvolvimento e manutenção mais coesos, no sistema desenvolvido foi utilizada a arquitetura MVC (*Model-View-Controller*), ela é amplamente adotada e sugerida no desenvolvimento de aplicações web (PRESSMAN et al, 2016, p.384).

Essa abordagem divide a aplicação em três componentes principais: o Modelo, responsável pela lógica de negócios e manipulação dos dados; a Visão, que apresenta a interface do usuário de forma adequada; e o Controlador, que coordena as ações do usuário, atualiza o modelo e seleciona a visão apropriada (DEACON, 1995), isso possibilita uma separação clara de responsabilidades, facilitando o desenvolvimento, a manutenção e a colaboração entre desenvolvedores. Acompanhado dessa arquitetura, também foi usado uma linguagem client-side, esse tipo de linguagem é utilizado no desenvolvimento de aplicações web para executar códigos no lado do cliente, ou seja, no navegador do usuário. Essas linguagens, como JavaScript, são responsáveis por melhorar a interatividade e a dinamicidade das páginas web, permitindo a manipulação do conteúdo da página, validação de formulários, comunicação assíncrona com o servidor e a criação de interfaces mais ricas. Ao serem executadas no navegador, as linguagens client-side reduzem a necessidade de comunicação

constante com o servidor, proporcionando uma experiência de usuário mais ágil e responsiva (ISKANDAR, 2020).

No desenvolvimento da aplicação, adotou-se a metodologia de desenvolvimento ágil onde um conjunto de práticas e princípios visam aprimorar a eficiência e a flexibilidade no processo de desenvolvimento de software que podem ser facilmente adaptados ao domínio clínico (OZCAN et al, 2013). O desenvolvimento ágil é composto por cinco etapas principais: levantamento e análise de requisitos, projeto do sistema, implementação, testes e documentação. No levantamento e análise de requisitos, os requisitos do projeto são identificados com os stakeholders e priorizados, essa é uma das partes mais importantes do processo de desenvolvimento, logo todo o conceito do sistema deve ser bem entendido para que o sistema final atenda às necessidades dos usuários. O próximo passo é projetar o sistema, definindo qual arquitetura será usada, como os dados serão modelados, prototipação de interface e entendimento de como os fluxos ocorrerão, tudo isso precisa ser bem definido para que a próxima fase aconteça, a de implementação, nessa parte as equipes trabalham em iterações curtas, entregando funcionalidades incrementais e funcionais para o cliente. O penúltimo processo é responsável por testar tudo o que foi produzido na fase anterior, logo os testes são executados em massa, garantindo que tudo que foi requisitado realmente está sendo entregue sem erros. Após os testes bem-sucedidos, é realizada a documentação do sistema, explicando seus fluxos, a forma que os dados foram projetados, documentos focados nos futuros desenvolvedores e até manuais para os usuários. Essas etapas iterativas permitem uma resposta mais ágil e com menor custo (AL-SAQQA, 2020) às mudanças, visto que toda etapa é validada com o cliente, evitando mudanças radicais e complexas, ademais esses ciclos possibilitam uma entrega mais rápida e de valor aos usuários finais.

Foi decidido também dividir as funcionalidades do sistema em módulos, onde cada módulo representa um micro serviço contendo suas responsabilidades, segundo Runhai (2021) essa divisão em partes menores gera um baixo acoplamento e independência entre módulos, sendo bastante eficaz e tornando a aplicação modular, visto que cada hospital tem suas respectivas necessidades, por exemplo, hospitais de pequeno porte não utilizarão todos os módulos oferecidos pela aplicação, logo usarão os módulos que têm interesse. Esse tipo de estrutura, segundo Demirel (2018, p. 40) reduz custos do investimento inicial da instituição, já que esses módulos podem ser liberados conforme o uso ou aquisição do hospital. A divisão dos módulos do medNUTES foi feito conforme abaixo:

- Gerenciamento de usuários;

- Gerenciamento de paciente;
- Gerenciamento de consultas;
- Gerenciamento de estoque de utilitários;
- Gerenciamento de prescrições;
- Gerenciamento de relatórios e estatísticas;

O módulo de gerenciamento de usuários trata basicamente da segurança através do controle de acesso às informações disponíveis na aplicação. Um determinado usuário só tem acesso as telas e funcionalidades que pertencem a função dele dentro do hospital, essas permissões são definidas ao cadastrar um novo usuário, “(...) *protection of a set of information to preserve the value it holds for an individual or a health organization.*” (VIDAL, 2019, p. 621). Nesse módulo também é possível definir perfis de usuários e suas permissões, facilitando a criação de novos usuários. Segundo Vidal(2019) esse módulo é de extrema importância na aplicação, visto que a segurança da informação é um dos principais fatores influenciadores a serem incluídos em um modelo abrangente para a gestão de sistemas hospitalares.

O módulo de paciente é ponto focal do sistema, mapeando vários dados acerca do mesmo, segundo Leavitt “(...) *electronic patient records are not merely automated forms of today’s paperbased medical records, but encompass the entire scope of health information in all media forms. Thus electronic patient records include medical history, current medications, laboratory test results, x-ray images and etc*” (LEAVITT, 2002, p.257). No medNUTES esse módulo ficou responsável por gerar um registro completo e preciso das informações dos pacientes, incluindo dados demográficos, histórico médico, medicamentos prescritos e triagens. Essas informações são essenciais para a tomada de decisões clínicas, proporcionando um atendimento personalizado e de qualidade. Segundo Dağlı (2011, p.3) o fato desse módulo ser feito digitalmente permite que problemas de prescrição, perda de tempo, digitação e falta de acompanhamento na evolução do paciente sejam mitigados. Ademais, os profissionais de saúde têm acesso a todas as informações de um paciente em um só lugar, recebendo atualizações instantâneas desse paciente. Por último, e não menos importante, esse módulo foi desenvolvido em um padrão de Interface de Programação de Aplicação (API), isso permite que outros sistemas hospitalares, com devida autenticação, possam usar os dados do nosso sistema, acessando por exemplo, todos os dados de um determinado paciente, isso resolve o fato de “*The decentralized nature of the healthcare industry is an obstacle for information sharing (...)*” (DAĞLI, 2011, p.4).

O módulo de gerenciamento de consultas tem o objetivo de controlar os agendamentos de cada paciente, desempenhando um papel fundamental na organização e otimização do tempo, contribuindo para a redução do tempo de espera dos pacientes, permitindo um fluxo de atendimento mais eficiente e contínuo, ajudando a antecipar os recursos necessários, como médicos, equipamentos e salas de exames, facilitando o planejamento adequado e garantindo que os recursos estejam disponíveis quando necessário, isso gera uma “(...) *benefits such as quality of care, patient safety, and member service enhancement.*” (BALARAMAN, 2013, p.55).

O módulo de gerenciamento de utilitários é um dos principais do sistema, visto que ele é responsável por gerir toda a parte de insumos de um hospital, auxiliar no controle eficaz de medicamentos, equipamentos médicos e outros materiais hospitalares (DEMIREL, 2018), isso aumenta a eficiência do hospital e reduz o tempo e o esforço necessário para manter o controle manual dos estoques, permitindo que os funcionários do hospital se concentrem em outras tarefas importantes (BALARAMAN, 2013). Além disso, este módulo é fundamental para a redução de desperdícios e custos ao proporcionar um controle preciso do estoque, ele ajuda a evitar a compra excessiva ou insuficiente de materiais, evitando o desperdício de medicamentos e outros itens com prazo de validade e impedindo a falta de itens críticos que poderiam interromper os serviços do hospital e resultar em custos adicionais. Indiretamente esse módulo também contribui para melhorar a qualidade do atendimento ao paciente assegurando que medicamentos e equipamentos necessários estejam sempre disponíveis, auxiliando em situações de emergência, quando a disponibilidade imediata de itens pode ser vital para a saúde do paciente. Ademais, essa parte também fornece um histórico de rastreabilidade para cada item do estoque, segundo Saha “(...) *any authorized person can track any drug in any time and can see how much inventory is left on that day.*” (SAHA, 2017, p. 181), todo esse controle permite aos hospitais identificar tendências e padrões de uso que podem ser úteis para prever a demanda futura e planejar as compras.

Quando o assunto é gerenciamento de prescrições, vários erros podem surgir, segundo Lesar “(...) *using the wrong drug abbreviation or dosage in paper based prescriptions, incorrect dosage calculations and prescribing a drug that previously caused an allergy to a patient are the most frequent medication errors*” (DAĞLI, 2011, p.4), dessa forma o objetivo do módulo de prescrições é contornar esses problemas, permitindo aos médicos prescrever medicamentos diretamente através do sistema, que são então registrados e monitorados. Isso ajuda a simplificar e otimizar a cadeia de processos de prescrição, desde o momento em que o médico prescreve até a administração para o paciente, reduzindo os riscos envolvendo esse

fluxo, segundo Forrester (2014), a adoção de um sistema computadorizado para realizar essa tarefa podem reduzir cerca de 30% a 84% os erros adversos relacionados a medicamentos, além do mais, no caso de prescrições com medicamentos errados ou superdoses, o médico estará ciente sobre a inconsistência do que está sendo prescrito. Esse módulo também auxilia o médico durante a prescrição, onde o mesmo visualiza uma lista de medicamentos sugeridos e suas respectivas doses, essa sugestão é elaborada por um médico-infectologista, segundo Saha ter um profissional especializado nessa área pode “(...) *they reduce the tendency of prescribing irrational and high cost. They can individualize the drugs and their dosing according to the patient needs which can reduce medication costs for treatment.*” (SAHA, 2017, p. 181). Outro fator importante é que com o monitoramento das prescrições, é possível criar relatórios abrangentes o que pode levar a melhores resultados, monitorando a eficácia dos medicamentos e identificando possíveis efeitos colaterais permitindo ajustar os regimes de tratamento conforme necessário.

Por fim, temos o módulo de relatórios e estatísticas, esta ferramenta oferece um meio de coletar, analisar e interpretar uma variedade de dados de saúde, financeiros e operacionais, permitindo o acompanhamento do desempenho do hospital. Tais indicadores fornecerão uma visão clara e abrangente de diversas métricas, como por exemplo, ele pode mostrar o número de pacientes atendidos, o tempo de espera, a utilização de medicamentos e equipamentos, entre outros dados relevantes. Esses indicadores seguem pontos corroborados por Payam Homayounfar (2012, p. 1137), segundo ele é necessário ter status sobre o desempenho geral da organização, dentre eles o monitoramento de medicamentos e controle das finanças. Balaraman (2013, p. 57) também sugere que tenha algo dedicado a sumarizar as principais informações no âmbito hospitalar, pois esses tipos de dados são frequentemente monitorados pela alta administração.

Ainda sobre esse escopo, o módulo permite a avaliação detalhada das finanças do hospital rastreando os custos, receitas, rentabilidade, eficiência de alocação de recursos e outros aspectos financeiros importantes. Com essas informações, os gestores podem tomar decisões estratégicas para melhorar a eficiência e a lucratividade, ademais o gerenciamento de relatórios e estatísticas é essencial para o cumprimento de requisitos regulatórios, facilitando a geração de relatórios precisos e atualizados que podem ser utilizados para demonstrar conformidade com várias normas e regulamentações.

Como apresentado, a aplicação conta com vários módulos abrangentes, que cobrem as principais áreas que um sistema de gestão hospitalar deve ter. Segundo Payam Homayounfar (2012, p. 1137), essas tarefas devem incluir os itens: 1) Armazenamento e monitoramento da

condição do paciente; 2) Gestão e fluxo de dados e 3) Aspectos financeiros. Na nossa proposta o item 1 equivale ao módulo de pacientes, consultas e prescrições; o item 2 equivale aos módulos de estoque de utilitários e usuários; o 3 equivale aos módulos de relatórios e estatísticas. Nessa mesma temática, Balaraman (2013, p. 56) também cita alguns módulos que são de extrema importância em um sistema de gerenciamento hospitalar, os quais serão citados a seguir: 1) módulo de registro de pacientes e agendamento de consultas; 2) módulo de gestão ambulatorial; 3) módulo de serviços; e 4) módulo gerenciador de usuários. Na nossa aplicação web, o item 1 equivale ao módulo de paciente e consultas; o ponto 2 equivale ao módulo de prescrição; o 3 corresponde ao módulo de relatório e estatísticas; e o último item corresponde ao módulo de usuários.

3 METODOLOGIA

3.1 LEVANTAMENTO E ANÁLISE DE REQUISITOS

A etapa de levantamento de requisitos, conforme a definição de Sommerville (2013), representa um período em que os engenheiros de software colaboram com clientes e usuários finais do sistema, visando obter dados sobre o domínio da aplicação, os serviços a serem providenciados pelo sistema, o desempenho esperado, as limitações de hardware, entre outros aspectos. Essa fase é desenvolvida diretamente com *stakeholder*, o qual representa alguém que exerce alguma influência, seja de forma direta ou indireta, sobre os requisitos do sistema, nesse grupo incluem-se os usuários finais que terão interações com o sistema e qualquer outro indivíduo dentro de uma organização que será impactado por ele.

Esse processo de análise e levantamento de requisitos desempenha um papel crucial no mapeamento abrangente de todas as exigências do cliente, visto que os usuários finais frequentemente não têm uma clara definição do que desejam de um sistema computacional, podendo ser desafiador para eles expressar de forma precisa o que esperam que o sistema realize, e, devido à falta de compreensão sobre o que é factível e o que não é, podem apresentar requisitos que não são realizáveis. Há uma necessidade de adaptação ao ambiente e contexto onde os *stakeholders* estão imersos, pois eles articulam requisitos em seus próprios termos, utilizando o conhecimento implícito em suas próprias atividades, com isso os engenheiros de requisitos, sem familiaridade no domínio do cliente, podem enfrentar dificuldades para compreender essas necessidades (SOMMERVILLE, 2013). Outro caso comum é que *stakeholders* distintos apresentam requisitos diversos e têm a capacidade de expressar essas variações de formas diferentes, em frente a isso o engenheiro deve identificar todas as fontes potenciais de requisitos e explorar tanto as semelhanças quanto os conflitos entre eles.

Para o desenvolvimento dessa etapa existem técnicas usadas com o objetivo de aumentar a precisão do entendimento dos requisitos e diminuir os riscos, permitindo que os engenheiros de software consigam adentrar ainda mais no contexto dos *stakeholders*, entendendo como as atividades realmente funcionam, no caso do medNUTES, usou-se em primeiro lugar a técnica de etnografia, sendo definida por Sommerville (2013, p.75):

Etnografia é uma técnica de observação que pode ser usada para compreender os processos operacionais e ajudar a extrair os requisitos de apoio para esses processos. Um analista faz uma imersão no ambiente de trabalho em que o sistema será usado.

O trabalho do dia a dia é observado e são feitas anotações sobre as tarefas reais em que os participantes estão envolvidos

Entrevistas também foram utilizadas, onde elaborou-se perguntas que foram feitas em entrevistas online e presenciais para os *stakeholders* do sistema, permitindo um maior entendimento do escopo. Por último, houve o desenvolvimento de casos de uso “*Em sua forma mais simples, um caso de uso identifica os atores envolvidos em uma interação e dá nome ao tipo de interação*” (SOMMERVILLE, 2013, p. 74), permitindo gerar documentos formais sobre os requisitos funcionais e não funcionais da aplicação, o resultado desse processo além de ser muito importante para manter documentação, serve de norte para a fase de codificação do sistema.

Dentro do processo de levantamento e análise de requisitos, existe uma complexidade envolvendo técnicas, sub processos que visam aprimorar essa fase, para seguimento do fluxo desse ciclo, dividiu-se o processo macro em processos menores, assim como definem Pressman e Maxim (2016), o resultado dessa divisão gerou três novos processos menores, o levantamento, a especificação e por fim, a validação de requisitos.

3.1.1 Levantamento

Nessa etapa ocorreu uma visita de campo no hospital Complexo de Doenças Infecto Contagiosas Clementino Fraga, localizado em João Pessoa no estado da Paraíba. O objetivo desse primeiro contato no ambiente hospitalar foi conversar com os funcionários, acompanhar seus fluxos e quais as maiores dificuldades a serem resolvidas pelo medNUTES, por fim, gerar uma documentação inicial para entendimento de futuras necessidades. Vale lembrar, que neste hospital já se utilizava um sistema simples, para tentar automatizar algumas tarefas, porém os *stakeholders* informaram vários tipos de necessidades, tanto no atual sistema, quanto em fluxos que essa aplicação não conseguia cobrir, algumas das necessidades apresentavam pontos em comuns e que potencialmente seriam requisitos para a aplicação, buscou-se mais profundamente entender os processos internos, quais processos eram dependentes e independentes uns dos outros.

O primeiro ponto muito citado foi com relação a área de pacientes, nela havia uma dificuldade muito grande no cadastro e acompanhamento deles, o problema já começava na recepção, onde para realizar a ficha com os dados da pessoa que necessitava ser atendida, recepcionistas reclamavam que o cadastro no sistema atual era bastante longo e demorado, o simples fato de buscar um paciente, imprimir sua ficha, ou acompanhar a evolução do mesmo,

se tornava algo muito custoso e trabalhoso. Com isso levantou-se alguns problemas iniciais relacionados aos pacientes, e para cada problema foi associado possíveis soluções que o sistema deveria entregar, isso no futuro permitiu mapear e entender mais precisamente os requisitos. Segue abaixo uma primeira concepção do que foi entendido:

Tabela 1 - Concepção de problemas relacionados aos pacientes

Problemas	Possível solução
Campos não necessários no cadastro, aumentando o tempo de cadastramento do paciente	Eliminar os campos não necessários
Erros de padronização de nome de cidades, doenças, sexo e outros campos	Padronização dos campos, dando preferência a entradas de seleção
Várias fichas existentes para um mesmo paciente	Identificação única para os pacientes, evitando duplicatas de fichas
Dificuldade em buscar a ficha de um determinado paciente	Possibilidade de filtrar e ordenar os pacientes
Cadastro dos dados de triagem de forma manual (usando papel)	Permitir adicionar dados de triagem relacionados a um determinado paciente
Agendamentos de pacientes de forma manual (usando papel)	Promover o agendamento dos pacientes para consultas/exames
Geração de relatórios sobre os pacientes de forma manual (montando os relatórios usando planilhas)	Viabilizar a geração de relatórios de forma prática
Geração somente de relatório simples	Oportunizar a aplicação de filtros a ferramenta de gerar os relatórios

Fonte: Elaborado pelo autor, 2024.

Ao aprofundar mais sobre os problemas relacionados aos pacientes, surgiu um novo ponto relacionado a isso, o gerenciamento das prescrições, visto que muitos funcionários reclamaram sobre ocorrências de prescrições erradas, com doses de medicamentos exageradas e não seguimento de protocolos para certas doenças, isso além de aumentar os custos do hospital devido há um excesso de doses desnecessárias, também aumentava o potencial de surgimentos de bactérias multirresistentes dentro do âmbito hospitalar. Além do mais foram listados casos de prescrições duplicadas, onde um médico realizava uma prescrição para um

paciente e após um tempo, outro médico por falta de comunicação cadastrava uma nova prescrição para o mesmo paciente, isso gerava confusão sobre qual prescrição deveria ser seguida, se a nova prescrição deveria substituir a anterior, entre outros problemas. Com as informações coletadas, foi levantado concepções sobre as dores do usuário final, agora relacionados a parte de gerenciamento de prescrições, como pode-se observar abaixo:

Tabela 2 - Concepção de problemas relacionados às prescrições

Problemas	Possível solução
Dificuldade em identificar se um paciente já teve sua prescrição realizada ou não	Oportunizar o controle das prescrições, onde cada prescrição é atrelada a um paciente
Falta de um protocolo antimicrobiano digital	Possibilidade de acomodar os protocolos antimicrobianos de forma digital
Dificuldade dos médicos em associar um melhor protocolo de antimicrobianos a uma determinada bactéria	Permitir sugestão de protocolos de tratamento para o médico prescritor
Dificuldade em acompanhar o uso de medicamentos e insumos por um determinado paciente	Viabilizar geração de relatório que rastreia quais insumos que foi consumido por um determinado paciente
Excesso de doses de um medicamento ou descumprimento de protocolos	Possibilidade da médica infectologista em conjunto com o médico, alterar essa decisão
Geração de relatórios relacionados a insumos, medicamentos, etc	Permitir a geração de relatórios com movimentação de insumos, medicamentos, etc, entre setores e pacientes

Fonte: Elaborado pelo autor, 2024.

A maior parte das necessidades envolvendo pacientes, agendamentos e prescrições foram mapeadas, após a visita das recepções e salas de atendimentos, prosseguiu-se para setores relacionados ao almoxarifado, uma das áreas mais importantes visto que é um plano de fundo que mantém todo o hospital funcionando. Nesse setor, há uma movimentação muito grande de saída e entrada de insumos, envolvendo notas fiscais e incremento e decréscimo de quantidades, os *stakeholders* relataram várias dificuldade em manter o gerenciamento desses insumos, começando com a necessidade de saber qual o estoque real dos suprimentos, a obtenção dessa informação era muito custosa, dado que todo esse controle de quantidades estava sendo feito em planilhas, muitas das vezes um insumo era usado porém não acontecia a

baixa desse insumo na planilha, ou o inverso, quando novos insumos chegavam esse incremento não acontecia, além do mais eles não conseguiam padronizar a nomenclatura dos insumos, ocorrendo de existir um único insumo mapeado várias vezes na planilha, somente com nomes distintos, isso gerava muita discrepância entre o estoque real e o estoque digital.

Outro obstáculo, não se limita só a parte de estoque, mas sim o rastreamento dos insumos, uma vez que o trânsito de suprimentos entre setores e pacientes dentro do hospital, acontece com uma alta frequência, gerando assim a necessidade de informações do tipo: “Para onde foi o insumo X?”, “De onde chegou o insumo Y”, ou “Quantos insumos do tipo Z ainda tem no estoque?”, os usuários finais relataram que muitas da vezes não conseguiam obter esses dados, e quando conseguiam se tratava de algo limitado, cabe destacar que os *stakeholders* também precisavam do rastreamento de notas fiscais para questões de auditoria, essas informações demoravam demais a serem obtidas, tanto que uma das necessidades foi a maior agilidade no levantamento de dados com notas fiscais para serem auditáveis. Sendo assim, o mapeamento das necessidades iniciais relacionadas a gestão de insumos ficou da seguinte forma:

Tabela 3 - Concepção de problemas relacionados ao almoxarifado

Problemas	Possível solução
Dificuldade em padronizar a nomenclatura dos insumos	Padronização da nomenclatura usando os padrões do SUS
Dificuldade de manter o estoque real dos insumos	Promover realização de saídas e entradas de insumos
	Permitir relacionar as saídas e entradas e automaticamente atualizar os estoques
Dificuldade de mapear saídas e entradas de insumos	Possibilidade de visualizar as saídas e entradas entre setores e pacientes
Dificuldade em rastrear as notas fiscais dos insumos	Viabilizar o rastreamento de notas fiscais atrelada aos insumos

Fonte: Elaborado pelo autor, 2024.

Após a visita dos setores relacionados a estoques e insumos, visitou-se a parte mais gerencial do hospital, os usuários finais relataram problemas em levantar dados com uma visão mais administrativa, como saber o fluxo de entrada de pacientes em um determinado

tempo, quais insumos que um específico paciente consumiu, ou quais insumos devem ser comprados pois o estoque está crítico. Em frente ao que foi requisitado, percebeu-se que em todos os problemas citados, há um ponto em comum, que é a geração de relatórios com a possibilidade de unir dados de várias áreas do hospital, como relacionar dados de pacientes a insumos, ou dados de agendamento a parte de triagem, ou quantidade de pacientes em um determinado período do tempo, dito isso desenvolveu-se a primeira concepção de problemas e soluções para esse caso:

Tabela 4 - Concepção de problemas relacionados à gestão

Problemas	Possível solução
Dificuldade em gerar relatórios de forma simples e rápida	Possibilidade de gerar relatórios
	Possibilidade de realizar filtros na geração de relatórios
	Possibilidade de imprimir os relatórios gerados

Fonte: Elaborado pelo autor, 2024.

3.1.2 Especificação

Nesse estágio, o objetivo é unir todas as informações que foram extraídas dos *stakeholders* e fazer uma especificação mais consistente e, portanto, mais compreensível, gerando artefatos como documentos escritos em linguagem natural, ou um conjunto de gráficos e diagramas, é nesse momento que as necessidades do cliente se transformam em funções que o sistema deverá executar.

No medNUTES, escolhemos especificar usando casos de usos, uma técnica na engenharia de software usada para capturar e descrever as interações entre os usuários (atores), e um sistema, fornecendo assim uma representação visual e narrativa das funcionalidades do sistema do ponto de vista do usuário, ajudando a entender como o sistema será utilizado na prática. O primeiro passo feito no desenvolvimento dos casos de uso, foi identificar quem são os atores “(...) *Atores são as diferentes pessoas (ou dispositivos) que usam o sistema ou produto no contexto da função e do comportamento a ser descritos.*” (PRESSMAN et al, 2016, p. 149), no caso do medNUTES identificou-se 8 atores diferentes:

- Recepção: Esse ator é responsável por realizar o primeiro contato com o paciente, realizando seu cadastro de forma correta no sistema, caso o mesmo já tenha registro no hospital, a recepção pode atualizar algum dado inconsistente. Em alguns casos, o paciente necessita de agendamento para algum serviço de consulta ou exame, isso também é executado por esse ator, que vai ficar responsável por fazer essa marcação.
- Triage: O pessoal da triagem é encarregado de avaliar a condição do paciente, colhendo alguns dados como pressão arterial, oximetria, febre e sintomas, após isso, o paciente terá o grau de gravidade atribuído.
- Médico-prescritor: Responsável por cadastrar as prescrições no sistema e acompanhar a evolução do paciente.
- Médico-infecologista: Responsável por cadastrar protocolos de tratamento para doenças específicas, ajudando o médico na hora da realização de uma prescrição.
- Almoarifado: Funcionários de vários setores dentro do hospital, responsáveis por manter toda a parte de estoque em dia, dando baixa nos insumos que saíram e incrementando os insumos que chegaram, eles também acessam quais os insumos estão com estoque críticos, dessa forma podem pedir novas remessas.
- Gestor: Esse ator tem uma visão mais gerencial de tudo o que acontece no hospital, ele é responsável por gerar gráficos e analisá-los, tomando medidas importantes com base nos dados e realizar auditorias.
- Administrador: Funcionário com o objetivo mais de gestão dos usuários que podem acessar a plataforma, fornecendo acesso para novos funcionários, e em casos específicos, verificando logs de auditoria do fluxo no sistema.

Posteriormente a identificação dos atores no sistema, partiu-se para o desenvolvimento dos casos de uso, um detalhe importante é que com base nas necessidades coletadas na fase de levantamento, optou-se por dividir o sistema em módulos, onde esses módulos tornaram a aplicação mais flexível e menos acoplada, essa decisão permitiu que o medNUTES fosse desenvolvido utilizando desenvolvimento ágil, uma metodologia que será abordada mais à frente, por essa razão cada módulo foi lançado sem dependência de outros, permitindo *releases* de incrementais. Dito isso, o desenvolvimento dos casos de uso foi feito respeitando

essa divisão, segue abaixo as especificações desenvolvidas sobre cada módulo e uma descrição sobre eles.

O primeiro módulo definido no medNUTES foi o de usuários, essa parte não foi uma necessidade requisitada diretamente pelos usuários finais, tendo em vista que os problemas giram em torno do fluxo de trabalho dos *stakeholders*, a recepção forneceu os problemas relacionados a seu fluxo, como dificuldade em gerenciar os pacientes e agendamentos, já o almoxarifado, necessitava de um maior controle dos estoques, saídas e entradas de insumos, os médicos tinham dificuldade no cadastramento de prescrições, e assim por diante, cada usuário citou suas necessidades na sua respectiva área, notou-se que não seria interessante os funcionários do almoxarifado acessarem a área de pacientes, como cadastro ou agendamentos de consultas, em função de que isso não pertencia a seu papel dentro da organização, da mesma forma a recepção não deveria ter permissão para cadastrar prescrições para o paciente, logo, definiu-se o módulo de gestão de usuário, onde cada para cada perfil de usuário, existem permissões associadas a ele, impedindo que um usuário possa acessar e executar ações que não estão no seu escopo.

O responsável por manter o controle dessa área do sistema é o Administrador, segue abaixo o caso de uso desenvolvido representando os requisitos deste usuário:

Figura 1 - Módulo de usuários

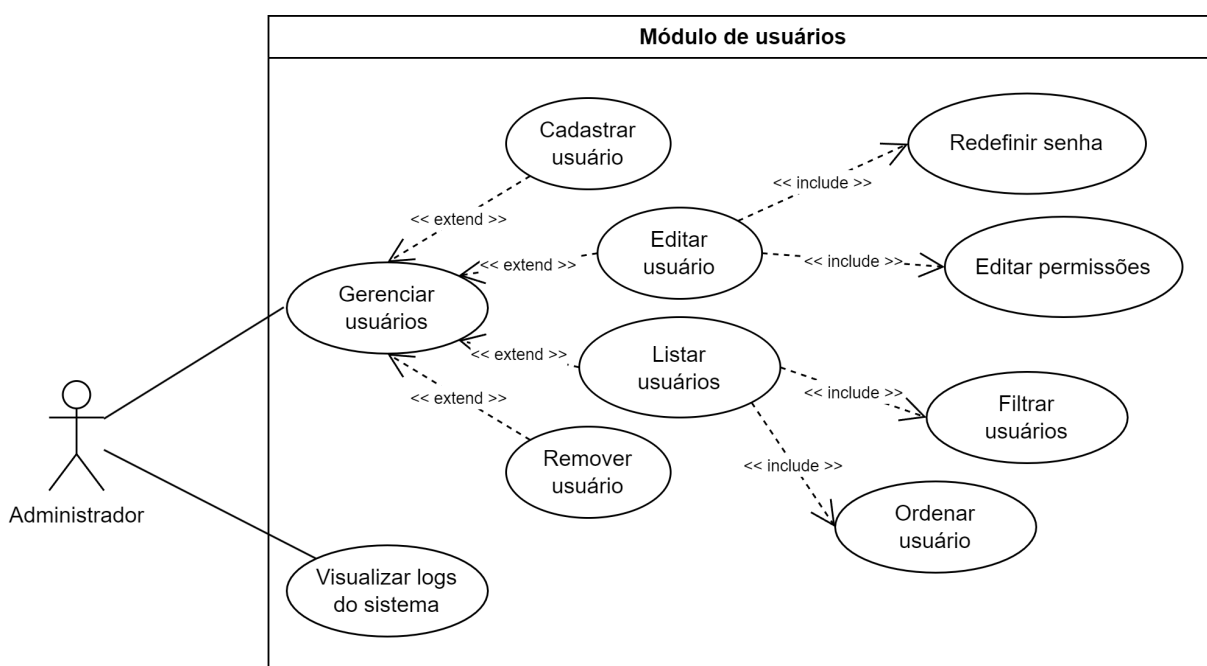


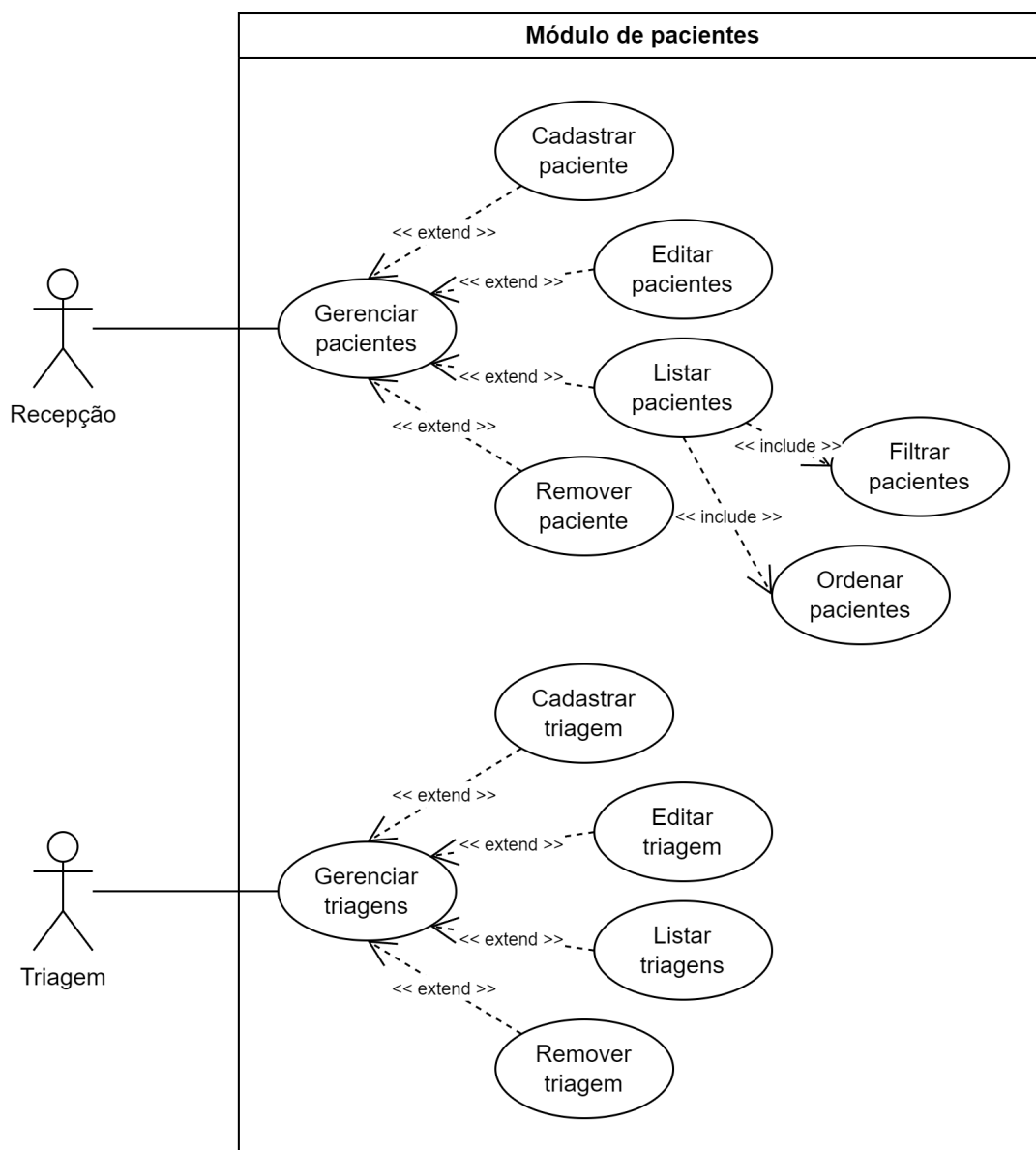
Tabela 5 - Requisitos do módulo de usuários

UC	Nome UC	Ator UC	Descrição UC
UC-01	Cadastrar usuário	Administrador	Cadastrar novo usuário no sistema, definindo o perfil de permissões que esse usuário se encaixa
UC-02	Editar usuário	Administrador	Editar usuário já existente no sistema, atualizando as informações fornecidas no cadastro
UC-03	Listar usuário	Administrador	Visualizar todos os usuários do sistema em uma listagem
UC-04	Filtrar usuários	Administrador	Filtrar usuários por nome, cpf, e email
UC-05	Ordenar usuários	Administrador	Ordenar usuários por nome e email
UC-06	Remover usuário	Administrador	Remover usuário já existente no sistema, removendo assim seu acesso ao medNUTES.
UC-07	Redefinir senha	Administrador	Redefinir senhas de usuários já existentes na plataforma, em caso de esquecimento de senha por parte dos usuários
UC-08	Editar permissões	Administrador	Editar as permissões de um usuário específico, permitindo ou bloqueando o acesso de menus específicos no medNUTES
UC-09	Visualizar logs do sistema	Administrador	Visualizar atividades dos usuários no sistema em uma listagem

Fonte: Elaborado pelo autor, 2024.

O próximo módulo da aplicação é o de pacientes, esse módulo reúne as soluções de problemas como o cadastro do paciente, como remoção de campos desnecessários, centralização da ficha de um paciente evitando duplicatas, possibilidade de buscar e ordenar os pacientes facilmente, entre outros. Além disso, a parte de triagem também é atendida, segue abaixo o caso de uso desse módulo:

Figura 2 - Módulo de pacientes



Fonte: Elaborado pelo autor, 2024.

Tabela 6 - Requisitos do módulo de pacientes

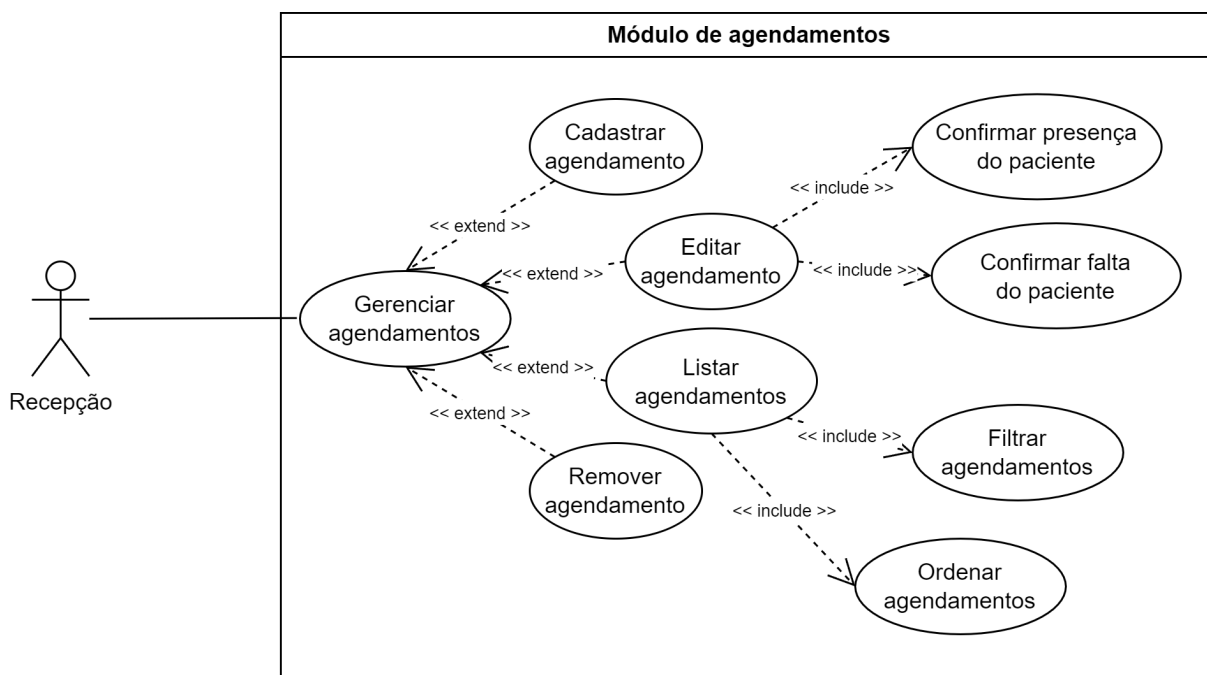
UC	Nome UC	Ator UC	Descrição UC
UC-10	Cadastrar paciente	Recepção	Cadastrar novo paciente no sistema de forma prática, com a remoção de campos não necessários
UC-11	Editar paciente	Recepção	Editar usuário já existente no sistema, permitindo atualizar dados que foram cadastrados de maneira incorreta

UC-12	Listar pacientes	Recepção	Visualizar todos os pacientes do sistema em uma listagem
UC-13	Remover paciente	Recepção	Remover um paciente já cadastrado
UC-14	Filtrar pacientes	Recepção	Filtrar pacientes por nome, cpf, cartão sus e nome da mãe
UC-15	Ordenar pacientes	Recepção	Ordenar pacientes por nome, cpf e cartão sus e data de cadastro
UC-16	Cadastrar triagem	Triagem	Cadastrar triagem relacionada a um paciente, informando dados como sinais vitais e grau do paciente
UC-17	Editar triagem	Triagem	Editar triagem já existe
UC-18	Listar triagens	Triagem	Listar todas as triagens de um paciente
UC-19	Remover triagem	Triagem	Remover triagem de um determinado paciente

Fonte: Elaborado pelo autor, 2024.

Como notou-se anteriormente no processo de levantamento, a área referente ao gerenciamento de consultas e exames também era um problema recorrente, a dificuldade em controlar os pacientes que seriam atendidos em um determinado dia, quais paciente iam realizar consulta ou exames, e quem seria o profissional responsável por atender esses paciente se tornavam informações confusas, e que no fim do dia acabavam gerando confusão nas salas de espera. Em frente a isso, desenvolveu-se o módulo de agendamento, que atende as necessidades especificadas pelos usuários, os casos de uso referente a esse escopo do sistema ficaram da seguinte forma:

Figura 3 - Módulo de agendamentos



Fonte: Elaborado pelo autor, 2024.

Tabela 7 - Requisitos do módulo de agendamentos

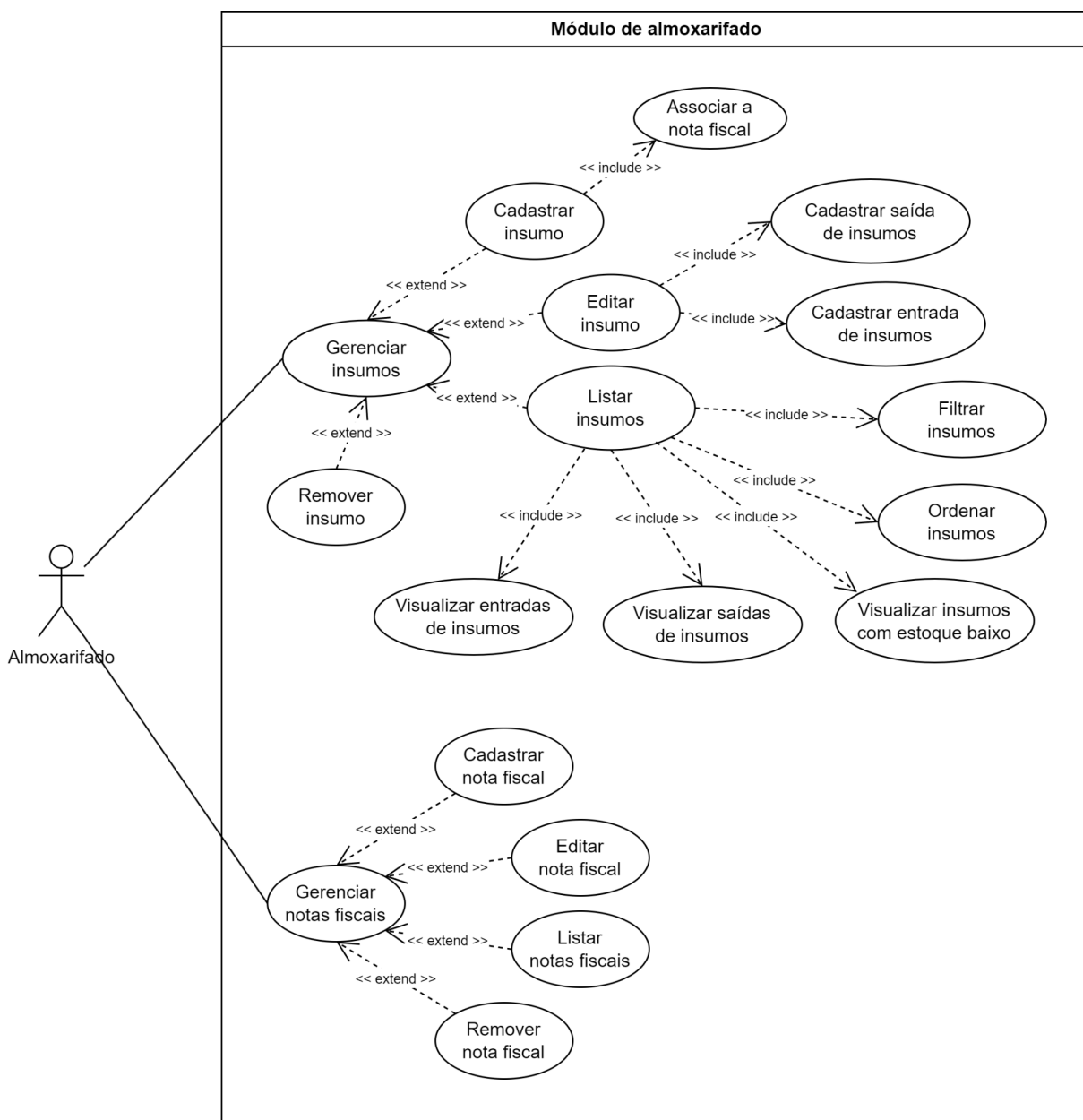
UC	Nome UC	Ator UC	Descrição UC
UC-20	Cadastrar agendamento	Recepção	Cadastrar agendamento para um determinado paciente, associado a um determinado médico
UC-21	Editar agendamento	Recepção	Editar agendamento já existente
UC-22	Listar agendamentos	Recepção	Visualizar todos os agendamentos com base em um calendário
UC-23	Remover agendamento	Recepção	Remover um agendamento já cadastrado
UC-24	Filtrar agendamentos	Recepção	Filtrar agendamentos por dia de atendimento, por médico, por paciente, e por recurso solicitado
UC-25	Ordenar agendamentos	Recepção	Ordenar agendamentos por nome do paciente, nome do médico e por dia do agendamento
UC-26	Confirmar presença do paciente	Recepção	Confirmar a presença do paciente caso o mesmo compareça para realização do recurso requerido

UC-27	Confirmar falta do paciente	Recepção	Confirmar a falta do paciente caso o mesmo não compareça para realização do recurso requerido
--------------	------------------------------------	-----------------	---

Fonte: Elaborado pelo autor, 2024.

Partimos agora para um dos módulos extremamente importantes para a manutenção dos hospitais, o setor de gerenciamento de insumos, nas etapas anteriores levantou-se várias reclamações dos funcionários acerca do gerenciamento de produtos internos como, medicamentos, materiais hospitalares, suprimentos, etc. Gerenciar esses insumos é uma tarefa muito delicada, visto que há muitos detalhes que precisam estar em sintonia, para que o controle seja bem feito, começando com o estoque real dos produtos, isso pode ocasionar diversos obstáculos caso esse dado seja falho, junto a isso é necessário rastrear de onde e para onde vão todos esses insumos, o tráfego interno entre setores e pacientes do próprio hospital e o tráfego externo, que é quando os produtos são adquiridos e devem ser incrementados no sistema mediante a notas fiscais. Unindo todas as informações levantadas com os *stakeholders* desse setor, concluiu-se que:

Figura 4 - Módulo de almoxarifado



Fonte: Elaborado pelo autor, 2024.

Tabela 8 - Requisitos do módulo de almoxarifado

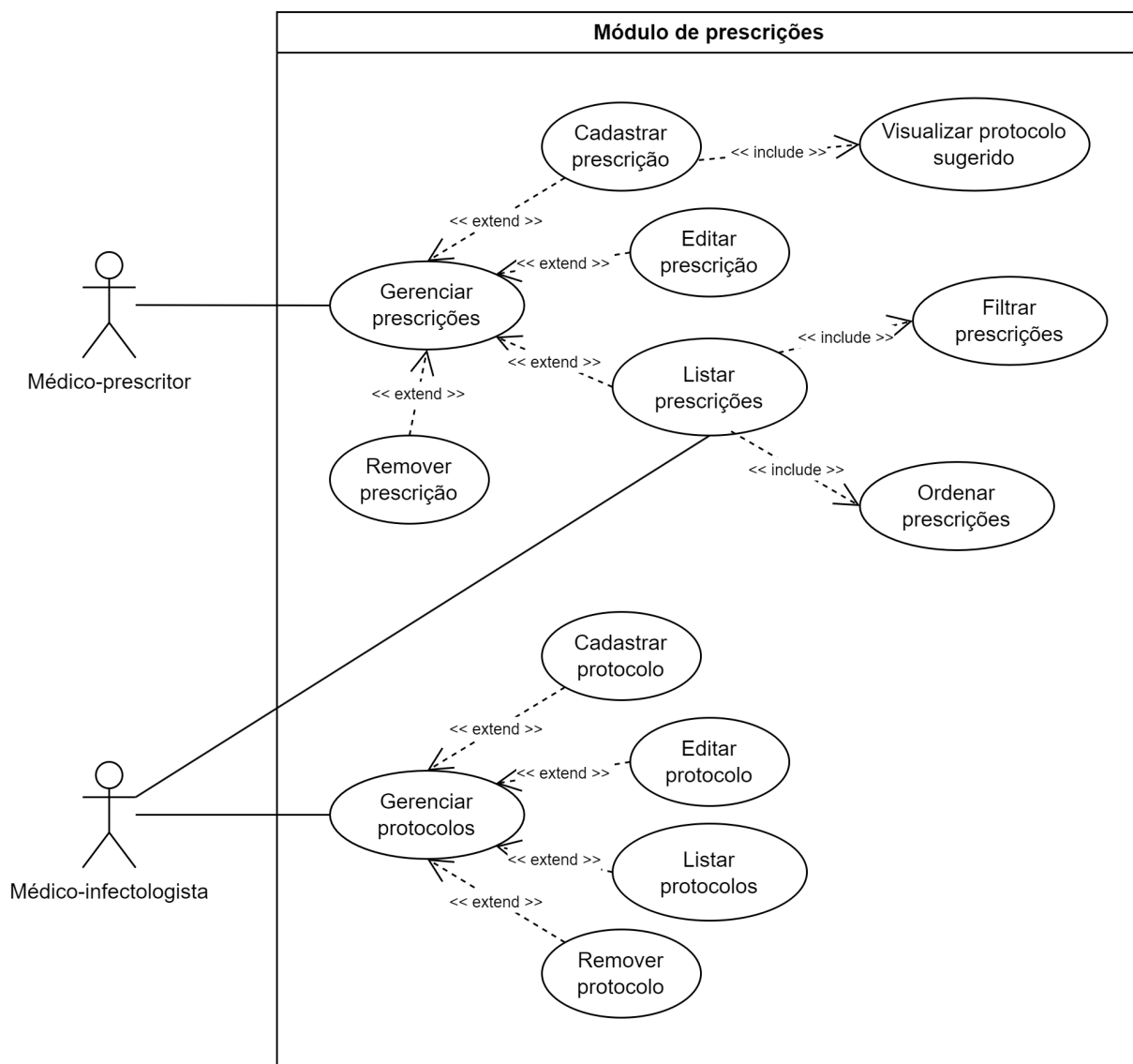
UC	Nome UC	Ator UC	Descrição UC
UC-28	Cadastrar insumo	Almoxarifado	Cadastrar insumo no sistema, o nome do insumo deve seguir padronização de nomenclatura via SUS
UC-29	Associar a nota fiscal	Almoxarifado	Associar uma nota fiscal a um insumo

UC-30	Editar insumo	Almoxarifado	Editar insumo já existente, exceto o dado de quantidade
UC-31	Cadastrar entrada de insumo	Almoxarifado	Cadastrar a quantidade de entrada de novo insumo
UC-32	Cadastrar saída de insumo	Almoxarifado	Cadastrar a quantidade de saída de um insumo já existente
UC-33	Listar insumos	Almoxarifado	Visualizar todos os insumos cadastrados em uma listagem
UC-34	Filtrar insumos	Almoxarifado	Filtrar insumos por nome e quantidade
UC-35	Ordenar insumos	Almoxarifado	Ordenar insumos por nome, quantidade e última vez atualizado
UC-36	Visualizar insumos com estoque baixo	Almoxarifado	Visualizar de forma clara e chamativa insumos com baixo estoque na listagem de insumos
UC-37	Visualizar saídas de insumos	Almoxarifado	Visualizar todas as saídas de insumos, exibindo origem e destino, juntamente com a quantidade
UC-38	Visualizar entradas de insumos	Almoxarifado	Visualizar todas as entradas de insumos, exibindo origem e destino, juntamente com a quantidade
UC-39	Remover insumo	Almoxarifado	Remover um insumo já existente

Fonte: Elaborado pelo autor, 2024.

Prosseguindo, foi especificado o módulo de prescrições, segundo os relatos dos *stakeholders* as maiores necessidades dessa etapa foi a falta de controle do insumos prescritos para o paciente, a ocorrência de múltiplas prescrições para uma mesma pessoa, gerando confusão sobre o que deveria ser seguido, outro contratempo é que, dependendo do caso do paciente, alguns médicos não aplicavam o protocolo de tratamento correto, passando assim doses exageradas, ou usando um medicamento que não seria tão eficaz para a situação, isso pode acarretar mais custos para o hospital, e no pior dos casos, surgimento de bactérias multirresistentes. Mediante a isso foi desenvolvido os casos de usos referentes a esse módulo da aplicação, conforme observa-se abaixo:

Figura 5 - Módulo de prescrições



Fonte: Elaborado pelo autor, 2024.

Tabela 9 - Requisitos do módulo de prescrições

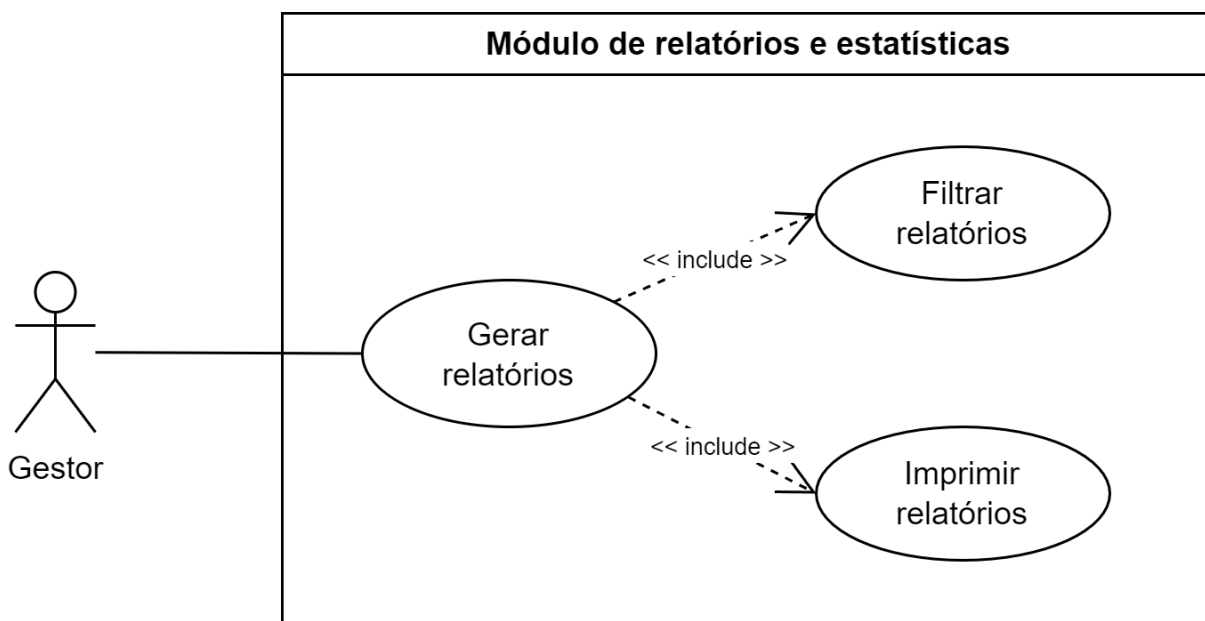
UC	Nome UC	Ator UC	Descrição UC
UC-40	Cadastrar prescrição	Médico-prescritor	Cadastrar prescrição no sistema, associando a ela um paciente e seu tratamento
UC-41	Visualizar protocolo sugerido	Médico-prescritor	Visualizar protocolos sugeridos pelo médico-infecologista associados ao conjunto de sintomas fornecidos na prescrição

UC-42	Editar prescrição	Médico-prescritor	Editar uma prescrição já existente
UC-43	Listar prescrições	Médico-prescritor	Listar as prescrições cadastradas
UC-44	Filtrar prescrições	Médico-prescritor	Filtrar prescrições por nome do paciente, data de cadastro e status
UC-45	Ordenar prescrições	Médico-prescritor	Ordenar prescrições por nome do paciente e data de cadastro
UC-46	Remover prescrição	Médico-prescritor	Remover mediante a um motivo uma prescrição cadastrada
UC-47	Cadastrar protocolo	Médico-infectologista	Cadastrar protocolo de tratamento para específicas doenças, bactérias ou conjunto de sintomas
UC-48	Editar protocolo	Médico-infectologista	Editar protocolo de tratamento já cadastrado
UC-49	Listar protocolos	Médico-infectologista	Listar protocolos cadastrados em forma de árvore
UC-50	Remover protocolo	Médico-infectologista	Remover protocolo de tratamento

Fonte: Elaborado pelo autor, 2024.

Por último e não menos importante, especificou-se o módulo de relatório e estatísticas, o seu uso foi mapeado visando usuários de cunho administrativo, haja visto que os *stakeholders* mostraram a necessidade de gerar relatórios sobre diversas áreas do hospital, como dados relacionados a tempo médio de atendimento dos pacientes, ou um relatório com todos os insumos que foram consumidos no hospital em um mês específico, dentre outros tipos de relatórios. Constatou-se que o dado final dos relatórios era algo bastante genérico, e conversando com os usuários finais não existe um modelo de relatório a seguir, então ficou claro que essa funcionalidade deveria apresentar uma alta flexibilidade, permitindo ao usuário montar o resultado final do seu relatório, com isso os casos de uso desse módulo seguiram o modelo abaixo:

Figura 6 - Módulo de relatórios



Fonte: Elaborado pelo autor, 2024.

Tabela 10 - Requisitos do módulo de relatórios

UC	Nome UC	Ator UC	Descrição UC
UC-50	Gerar relatórios	Gestor	Gerar relatórios pré-definidos porém com opções de filtragem
UC-51	Filtrar relatórios	Gestor	Filtrar relatórios pré-definidos por paciente, data, setores, insumos, médicos
UC-52	Imprimir relatórios	Gestor	Imprimir relatórios gerados em formato .pdf e .xlsx

Fonte: Elaborado pelo autor, 2024.

Ao fim do processo de levantamento e análise de requisitos, o mapeamento de todas as necessidades do usuário final foi realizada, transformando essas necessidades em funcionalidades (requisitos) que o sistema deveria seguir e executar, além disso gerou-se documentos especificando toda essa etapa, esses documentos foram de extrema importância para os próximos passos, uma vez que entendendo claramente o que os *stakeholders* necessitavam, decisões de estrutura de projeto, arquitetura, tecnologias e divisão de atividades, seriam feitas com muito mais assertividade.

Nesse momento, a fase envolvendo requisitos seria encerrada, dando seguimento para o processo de projeto do sistema, mas antes, visto que, a partir desse ponto em diante as fases iriam gerar artefatos entregáveis, como documentos, protótipos, funcionalidades do sistema, etc, isso implicaria em uma comunicação frequente com o cliente, ademais o processo de desenvolvimento das atividades por parte da equipe do NUTES iriam começar em massa, logo, visando a organização de tarefas, estruturação da colaboração em equipe, teve-se a necessidade de adotar uma metodologia de desenvolvimento focado no desenvolvimento ágil, como o próprio nome já sinaliza, metodologias de desenvolvimento ágil focam em diminuir significativamente o tempo que o *software* necessita para passar por todos os processos de desenvolvimento, para que essa meta seja atingida é necessário um engajamento constante dos *stakeholders*, fornecendo e priorizando novos requisitos e avaliando o que está sendo produzido, outro ponto que atrai o usuário final são as entregas incrementais, onde o sistema é feito em pedaços, facilitando a validação durante as etapas e como bonês, gerando entregáveis ao fim de cada incremento. Esse tipo de desenvolvimento também acomoda bem as mudanças, visto que o processo de produção é baseado em incrementos, ao surgimentos de novas mudanças, isso pode ser facilmente alocado no próximo incremento, vale ressaltar que a simplicidade é um lema dessa metodologia, evitando um desenvolvimento muito pesado e burocrático.

Para o medNUTES, foi escolhido o scrum, uma metodologia de desenvolvimento ágil muito usada nos processos de *softwares*, esse conceito se divide basicamente em 3 etapas, a primeira fase é planejamento geral de todas as atividades a serem executadas, estabelecendo objetivos gerais do projeto, na etapa dois, temos os processos mais importantes, a geração de ciclos que são chamados de *sprints*, contendo subatividades que serão realizadas em um determinado período de tempo, para o exemplo do sistema desenvolvido neste relatório, definiu-se *sprints* com cerca de 21 dias, onde as subatividades em cada ciclo deveriam somar um total de 120 horas. Ao fim de cada ciclo, era realizada uma reunião com os usuários finais com o objetivo de mostrar o que foi executado e validar se tudo estava de acordo com o que foi requisitado. Na fase final do scrum, são realizadas atividades relacionadas a documentação exigida, como manuais de usuário, diagramas e documentos sobre o sistema, por fim avalia-se as lições aprendidas com o projeto.

3.2 PROJETO DO SISTEMA

A fase de projetar o sistema é um passo inevitável no processo de desenvolvimento de *software*, segundo Pressman e Maxim (2016) essa etapa se aplica a qualquer tipo de sistema que esteja sendo construído, e se inicia quando os requisitos são mapeados e especificados.

Durante essa etapa, serão tomadas decisões essenciais para dar forma à implementação prática do sistema, visando transformar os requisitos levantados durante as fases anteriores em uma estrutura funcional coesa. Esta etapa compreende a definição da arquitetura do sistema, a tomada de decisões de design relacionadas à interface e à lógica de negócios, a modelagem de dados para armazenamento eficiente, e a escolha de tecnologias apropriadas, incluindo linguagens de programação, *frameworks* e bibliotecas. A elaboração de protótipos para visualização e validação da aparência e funcionalidades do sistema, também são executadas, isso permite mapear os fluxos de informação, destacando a trajetória dos dados desde a entrada até a saída.

Para o caso do medNUTES, essa etapa foi realizada baseando-se em três modelos de projeto mais importantes dentro da engenharia de software (PRESSMAN et al, 2016), o projeto de arquitetura, nele acontece a elaboração da arquitetura geral, identificando componentes-chave, seus relacionamentos e padrões de comunicação, em seguida, será visto o projeto de dados do sistema, esse processo tem o alvo de desempenhar um papel crucial na definição da estrutura e organização dos dados que serão manipulados e armazenados pelo sistema. Por fim, será descrito o projeto de interface, que descreve a forma em que o software se comunica com as pessoas que o utilizam.

3.2.1 Projeto de arquitetura

Para entender um pouco como o processo de projeto de arquitetura funciona, deve-se entender primeiro o que é uma arquitetura de software, segundo Pressman e Maxim (2016, p.232):

Arquitetura de software refere-se “à organização geral do software e aos modos pelos quais ela disponibiliza integridade conceitual para um sistema”. Em sua forma mais simples, arquitetura é a estrutura ou a organização de componentes de programa (módulos), a maneira como esses componentes interagem e a estrutura de dados que são usados pelos componentes.

Tentando fazer uma associação com uma casa, a arquitetura seria a planta de uma casa, fornecendo a estrutura e o desenho fundamentais para o desenvolvimento de sistemas de

software, da mesma forma que uma planta arquitetônica define a disposição dos quartos, conexões e fluxos dentro de uma casa, a arquitetura de software estabelece a estrutura, os componentes e as interações essenciais de um sistema. Dito isso, a etapa de projeto de arquitetura tem o objetivo de definir uma estrutura global do sistema, delineando como seus diversos componentes interagem, com isso há a necessidade de elaborar uma arquitetura geral, identificando componentes-chave, seus relacionamentos e padrões de comunicação.

Essa etapa considera decisões estruturais cruciais, como escolhas tecnológicas, *frameworks* e padrões arquiteturais, que impactam diretamente no desempenho e na escalabilidade do sistema, com isso a integração eficiente de módulos e a consideração de aspectos como segurança, manutenibilidade e futuras expansões são pontos centrais, fornecendo assim uma visão estratégica que orienta a implementação, garantindo uma base sólida para a construção do sistema de forma coesa, adaptável e alinhada aos requisitos identificados.

No processo de levantamento e análise de requisitos, a especificação das necessidades dos usuários foi levantada em módulos, isso permitiu o entendimento melhor do escopo da aplicação, na fase de projeto do medNUTES optou-se por continuar com essa divisão, dado que dentro dos hospitais há diferentes tipos de serviços e muitas das vezes eles são independentes entre si, em frente a isso, usar módulos permite a reusabilidade dos próprios módulos ou de componentes já existentes para a criação de novos módulos, ilustrando, o módulo de paciente está intrinsecamente ligado ao módulo de prescrições, já que mediante as necessidades levantadas pelos *stakeholders* um médico ao cadastrar uma nova prescrição, necessita associar um determinado paciente a ela, logo o custo necessário para desenvolver a parte relacionada aos pacientes dentro do módulo de prescrições, seria zero, pois haveria o reuso da parte já feita, isso não apenas acelera o processo, todavia promove a consistência e confiabilidade da parte que foi reusada, sendo necessário testá-la em um só ponto focal.

A modularidade também afeta diretamente na manutenção do sistema, em virtude do ambiente que o sistema seria executado, a adição de novos requisitos é algo recorrente, devido às novas necessidades ou necessidades já existentes, mas que ainda não atendente completamente o usuário final, com isso alterações em uma área específica poderiam ser realizadas sem afetar o sistema como um todo, não comprometendo a confiabilidade da aplicação.

Por se tratar de um sistema de gerenciamento hospitalar, isso implica de forma óbvia que ele iria funcionar dentro de um hospital, resultando em alguns pontos críticos que a aplicação deveria seguir. Ao falar de hospitais, deve-se entregar um sistema com alta

confiabilidade e principalmente com um alto índice de segurança, como a segurança deveria ser um dos pontos fortes do medNUTES, havia uma necessidade de aplicação de controle de acessos, algoritmos de criptografias e várias técnicas que permitissem garantir a segurança de fluxos e informação, isso deveria ser feito em várias camadas do sistema, desde de uma criptografia da base de dados, até a criptografia dos fluxos na interface.

Ainda sobre o público e local alvo do sistema, notou-se que hospitais tendem a aumentar seus fluxos, expandindo capacidades, funcionários e recursos ao longo do tempo, com isso a aplicação deveria apresentar um grau de escalabilidade, permitindo dimensionar recursos conforme a demanda em áreas específicas do sistema hospitalar, observou-se que para cumprir tal necessidade, desenvolver isso em camadas traria uma facilidade e manutenibilidade muito maior para a aplicação, como efeito colateral positivo, isso geraria uma possibilidade de integração com sistemas externos, onde cada camada poderia ser configurada para interagir de maneira eficiente com diferentes interfaces e protocolos.

Pelo o que foi levantado na fase de requisitos, notou-se que hospitais na maior parte das vezes, não apresentam um padrão financeiro semelhante, em frente a variedade de tipos de hospitais existentes como, públicos, privados, regionais, universitários, etc. Isso significa que nem todos conseguem disponibilizar para os funcionários bons dispositivos, ou uma frota de dispositivos padronizados, com isso, observou-se que o sistema deveria atender a vários dispositivos como, desktops, notebooks, tablets, celulares, dispositivos com diferentes capacidades de processamento e tamanhos de telas, e assim por diante, isso se sucede devido a uma falta padronização nos dispositivos que serão usados para acessar o sistema, ao conseguir cumprir essa necessidade, seria gerado um ponto extra positivo, já que dando suporte a dispositivos móveis, exemplificando, celulares, tablets e notebooks, o sistema passaria a oferecer um acesso remoto da aplicação.

Mediante ao que foi mensurado, partiu-se para a escolha de uma arquitetura que mais se encaixasse nas condições do sistema, foi percebido que vários dos pontos levantados batiam com uma arquitetura baseado em camadas, esse tipo de arquitetura pode ser usado para *softwares* com diversos propósitos, sendo notável por sua capacidade de estruturar as responsabilidades de diferentes partes (camadas), buscando estabelecer um isolamento claro e atribuir propósitos bem definidos a cada parte, proporcionando a possibilidade de reutilização por níveis superiores ou até mesmo a substituição eficiente dessas camadas.

Observou-se vários pontos em comuns com relação às necessidades discutidas anteriormente e a proposta desse tipo de arquitetura, um exemplo disso é a questão da divisão da aplicação em módulos, onde cada um funcionava de forma independente, essa

característica está presente nos fundamentos desse tipo de arquitetura, além do mais ela corrobora com a metodologia escolhida para gerir o desenvolvimento do sistema, que foi o scrum “(...) *abordagem em camadas apoia o desenvolvimento incremental de sistemas.*” (SOMMERVILLE, 2013, p. 110). No medNUTES a segurança era algo a ser priorizado, como dito anteriormente, logo a escolha dessa arquitetura ia facilitar essas atividades, dado que cada camada poderia receber mecanismos de proteção, funcionando semelhante a várias barreiras sequenciais, onde a medida que se avançava nessas barreiras, as técnicas de proteção seriam aplicadas, trazendo bastante segurança e confiabilidade aos dados armazenados. O fato de apresentar camadas, também influencia na manutenção do sistema, visto que uma camada pode ser alterada, ou até mesmo substituída sem que outras partes do sistema sofram com essas mudanças, pois cada camada apresenta suas respectivas responsabilidades, permitindo que outras partes continuem funcionando perfeitamente.

Ao mencionar a arquitetura em camadas, um dos padrões que seguem essa filosofia, principalmente quando tratamos de aplicações web, é o MVC (Modelo-Visão-Controlador), essa abordagem separa claramente a apresentação e a manipulação de dados dentro do sistema, que é organizado em três componentes lógicos que colaboram entre si de maneira interativa. O componente modelo supervisiona o sistema de dados, cuidando das operações relacionadas a esses dados, por sua vez, o componente visão determina e supervisiona como os dados são exibidos ao usuário, e por fim, o controlador administra a interação do usuário, transferindo essas interações de maneira eficiente entre a visão e o modelo.

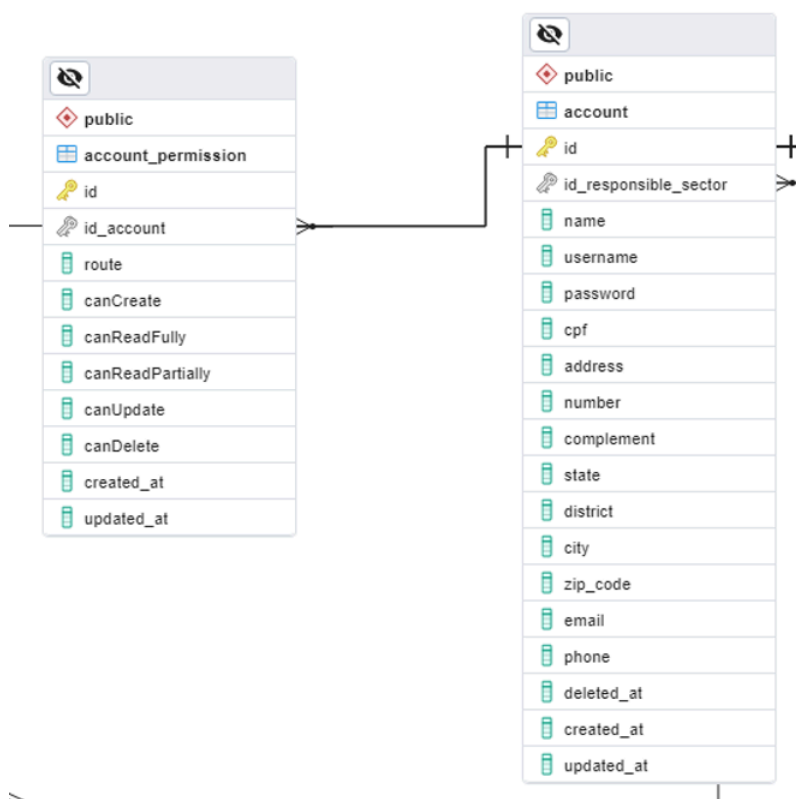
Diante do conceito do MVC (*Model-View-Controller*), identificou-se que usar esse padrão focado em uma aplicação web, seria a melhor escolha para o medNUTES, atendendo as necessidades citadas anteriormente, como por exemplo a falta de padronização dos dispositivos, uma aplicação web funciona em qualquer aparelho que tenha acesso a internet e um browser, claro, esse sistema deve ser otimizado para tal, mas isso resolveria a não certeza dos dispositivos que iriam executar o sistema. As manutenções em *WebApp* também são centralizadas, dado que ao surgimento de uma nova atualização não é necessário baixar ou executar nada, ao abrir novamente o domínio da aplicação o usuário final já contará com a última versão do sistema. Ao fim desse passo, decidiu-se que a arquitetura a ser usado no sistema seria uma arquitetura em camadas, juntamente com o padrão de projeto MVC (*Model-View-Controller*), que atende vários dos pontos necessários para o medNUTES, e finalmente, a aplicação se tratará de um sistema web, que pode ser acessado remotamente e em qualquer dispositivo.

3.2.2 Projeto de dados

A etapa de projeto de dados vai analisar tudo que foi obtido em etapas anteriores, levando em consideração um ponto de vista de dados, para mapear essas informações em classes de projeto e nas estruturas de dados necessárias para compor o sistema e seus fluxos. Os engenheiros de software se dedicam a criar uma representação estruturada e coerente do modelo de dados que o sistema utilizará, isso envolve a identificação de todas as entidades relevantes para o domínio do problema, bem como a definição das relações entre essas entidades, além do mais são estabelecidos os atributos associados a cada entidade, especificando suas características e propriedades. No medNUTES, além de buscar entender o fluxo de dados, como dito anteriormente, o objetivo dessa fase foi desenvolver o Diagrama de Entidade-Relacionamento (DER), ele tem o objetivo de abstrair e mapear em entidades os requisitos levantados nas fases anteriores, ilustrando seus atributos e as relações entre elas, essa representação visual auxilia na compreensão do funcionamento do sistema em termos de manipulação e armazenamento de dados.

Semelhante a etapa anterior, onde os casos de usos foram mapeados respeitando os módulos do sistema, o mesmo foi feito nesta etapa, mas agora utilizando os DERs, ao final, uniu-se todos, dessa forma foi garantido que os módulos apresentassem o mínimo de dependência de dados de outros módulos, mantendo o aspecto de modularidade e reuso requeridos para esse tipo de aplicação. O primeiro diagrama desenvolvido foi envolvendo o módulo de gerência de usuários, nesse módulo o usuário administrador pode gerenciar todos os usuários que tem acesso ao sistema, conforme abaixo:

Figura 7 - DER referente ao módulo de usuários

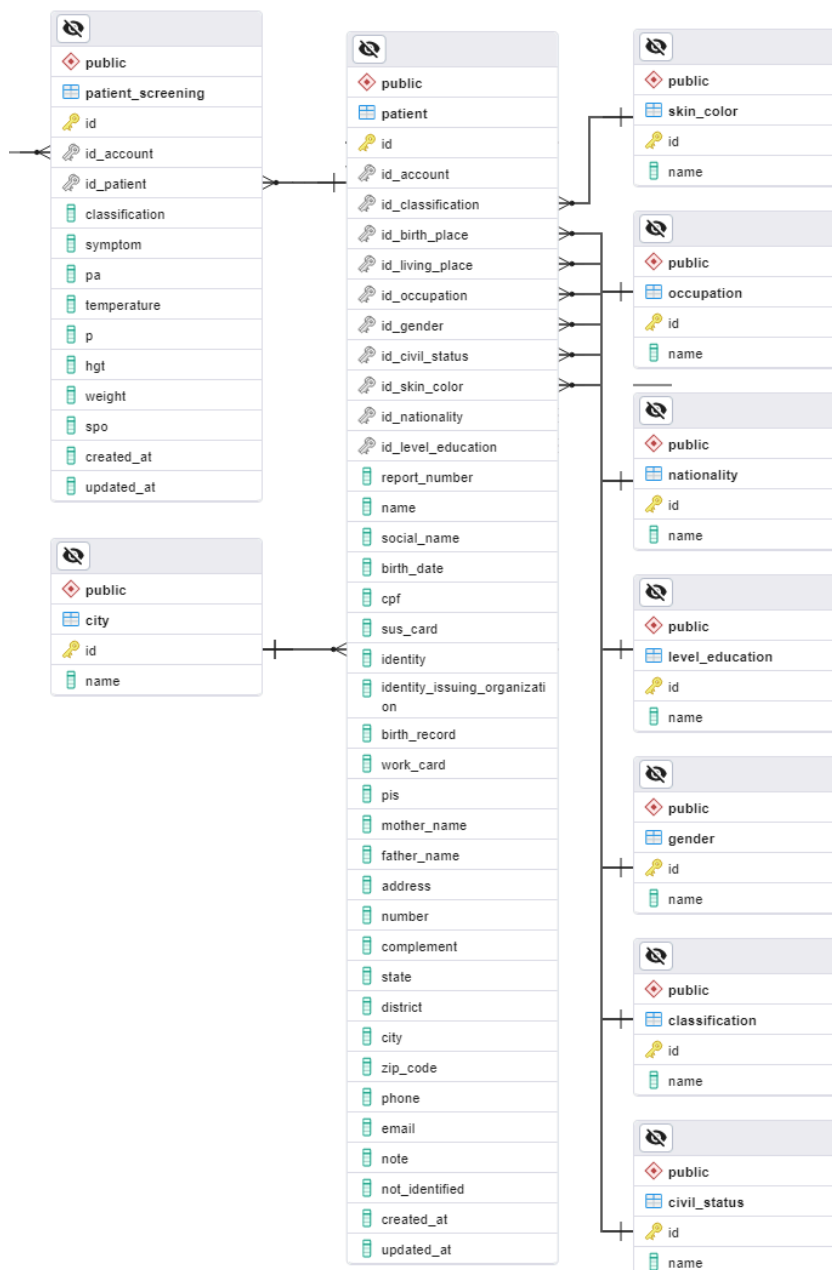


Fonte: Elaborado pelo autor, 2024.

Na etapa de requisitos levantou-se 8 atores no sistema, a forma mais fácil de mapear esses usuários seria criar 8 entidades, cada uma referente a um tipo de perfil com suas respectivas permissões, porém, ao pensar em expandir esse sistema para outros hospitais, notou-se que na maioria das vezes os perfis não iriam ser os mesmos, visto que cada organização tem perfis e funções internas para cada funcionário, outro cenário seria a necessidade de aumentar ou diminuir as permissões de um determinado perfil, logo, em frente a isso foi mapeado uma entidade que representa um usuário genérico, e atrelado a esse usuário existem uma ou mais permissões, trazendo assim uma flexibilidade para esse módulo. Logo, é possível que no hospital X, exista um perfil de gestor que tem as permissões Y e Z, porém no hospital A, existe o mesmo perfil de gestor mas que tem as permissões B e C. Além de trazer benefícios pela flexibilidade, essa medida evita manutenções desnecessárias com mudanças de perfis e permissões.

O próximo módulo, foi o de pacientes, esse módulo contém as funcionalidades de gestão de pacientes e triagens, resultando no DER abaixo:

Figura 8 - DER referente ao módulo de pacientes



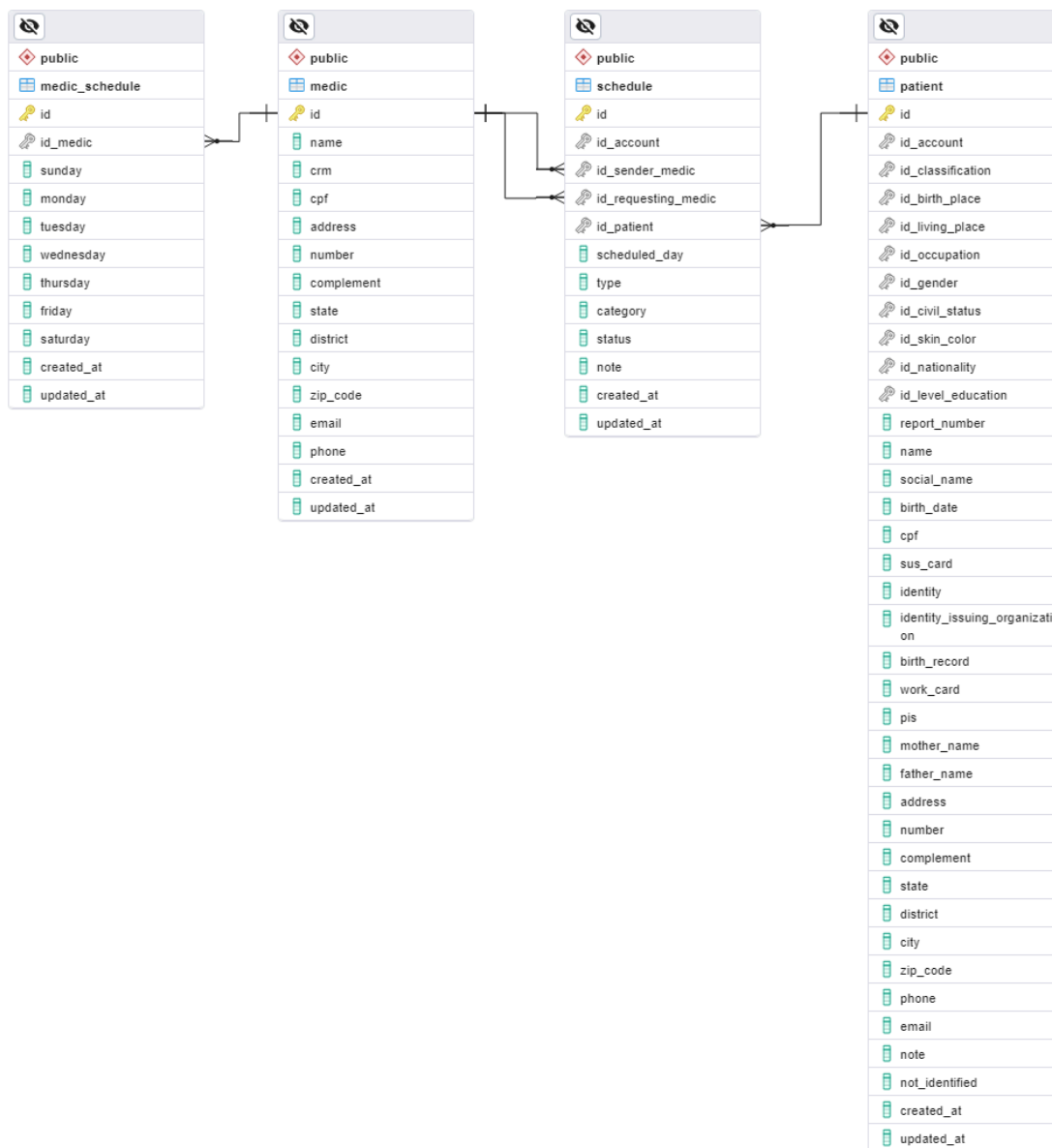
Fonte: Elaborado pelo autor, 2024.

O DER referente ao módulo de pacientes foi desenvolvido com mapeamento de 10 entidades, mapeou-se primeiro a entidade de paciente (*patient*), que contém todos os atributos presentes na ficha cadastral do mesmo, uma das principais necessidades dos *stakeholders*, foi a padronização de informações como sexo, ocupação, cidade, município, entre outros, pois esses dados estavam sendo preenchidos em textos abertos, logo não havia uma padronização dos conteúdos dos textos, por exemplo, no campo de cidade do paciente havia vários registros

referenciando uma mesma cidade com conteúdos diferentes, a cidade “Campina Grande” em algumas fichas apareciam “Campina G.”, “CG”, “campina grande” e isso fere a primeira regra de normalização de banco de dados, onde o objetivo é eliminar grupos repetitivos e garantir que cada valor em uma tabela seja atômico, por esse motivo, identificou-se 8 entidades referente a esses dados, foram elas, cidade (*city*), sexo (*gender*), classificação (*classification*), estado civil (*civil_status*), nível educacional (*level_education*), nacionalidade (*nationality*), emprego (*occupation*) e cor de pele (*skin_color*). Por fim, foi registrado a entidade de triagem (*patient_screening*), apresentando uma relação de muitos para um com a entidade de paciente, onde um paciente pode ter uma ou várias triagens, isso é importante pois após o cadastro do paciente, independente de quantas vezes ele for ao hospital, a organização consegue manter um histórico de todas as triagens realizadas.

Posteriormente, o próximo módulo onde realizou-se o projeto de dados foi o de agendamentos, esse módulo contém os requisitos relacionados aos agendamentos de consultas e exames para os pacientes, seu principal ator é a recepção.

Figura 9 - DER referente ao módulo de agendamentos



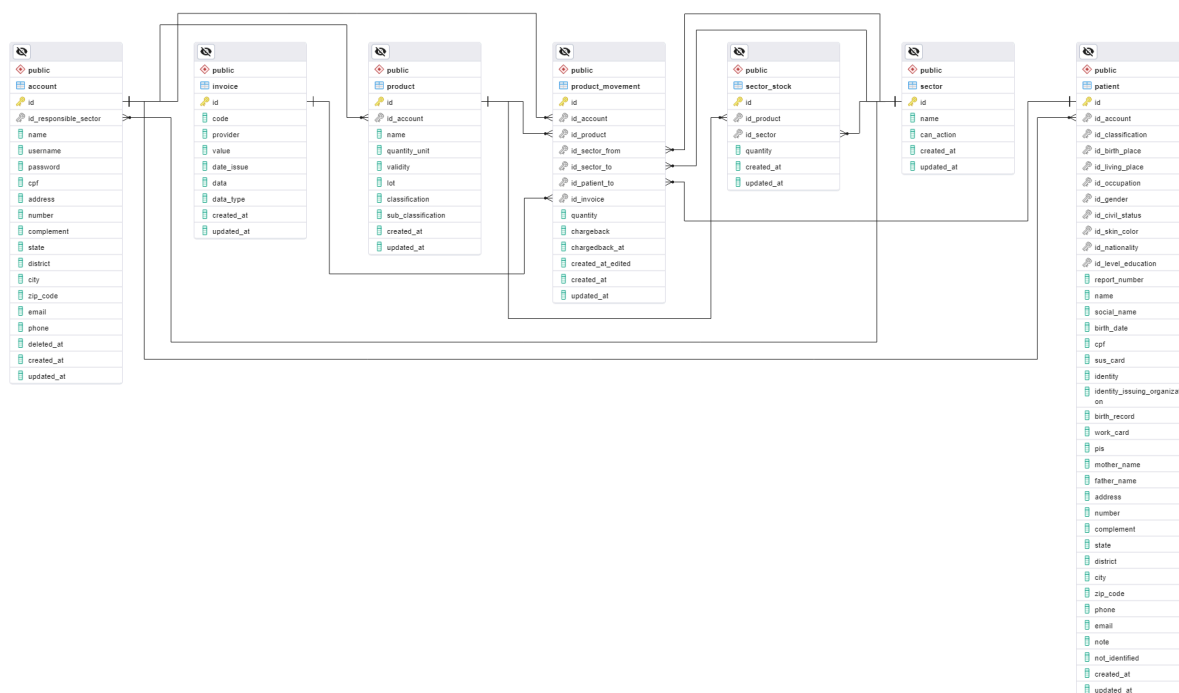
Fonte: Elaborado pelo autor, 2024.

Como resultado do projeto dos dados, essa parte do sistema gerou 4 entidades, sendo que a entidade de paciente (*patient*) já foi identificada no módulo de pacientes, e só será referenciada. A entidade principal foi a de agendamento (*schedule*), essa entidade contém todos os dados do agendamento do paciente, como o tipo de agendamento, status, dia do agendamento, dentre outros, 3 atributos cruciais são referenciados nessa entidade, o médico que pediu o agendamento, o médico que irá atender e o paciente que necessita do agendamento. As outras 2 entidades são relacionadas aos médicos, uma se refere a própria entidade de médico (*medic*) contendo todos os dados do mesmo, como nome, crm, etc, a

última entidade representa a disponibilidade dos médicos para agendamento (*medic_schedule*), essa entidade associa a um médico a uma quantidade de consultas por semana, exemplificando, o médico X oferece toda segunda, terça e quarta-feira 30 consultas, isso é importante pois antes de agendar um paciente, o usuário final precisa saber se naquele dia em específico o médico requerido oferece consultas.

Avançando, partiu-se para um dos módulos mais complexo com relação aos dados da aplicação, a parte de gestão de insumos, onde houve muitas reclamações por parte dos *stakeholders*, nele, funções como controle dos insumos, saída e entrada de novos estoques, gerenciamento de notas fiscais, etc, seriam abordadas.

Figura 10 - DER referente ao módulo de almoxarifado



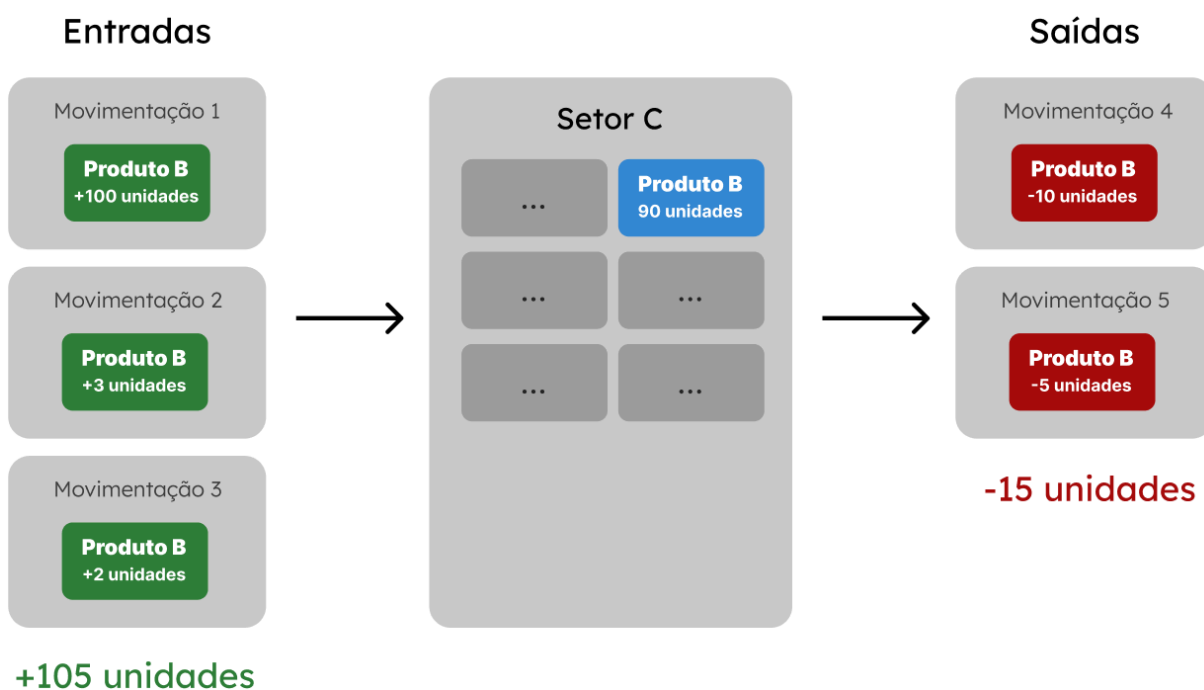
Fonte: Elaborado pelo autor, 2024.

Esse módulo foi desenvolvido com a composição de 7 entidades, sendo que 2 dessas entidades, a de usuário (*account*) e a de paciente (*patient*) foram somente reusadas, pois já tinham sido mapeadas em outros módulos. De forma geral, a idéia elaborada pela equipe foi entender que o almoxarifado é um composto de vários setores, e os suprimentos trafegam entre setores e pacientes, isso implica que cada setor contém um estoque específico de cada produto, por exemplo, o setor X tem o insumo Z em 500 unidades, já o setor A tem o mesmo insumo Z mas agora em 100 unidades. Mapeou-se assim a entidade de insumo (*product*), com

atributos referentes a nome, nota fiscal, classificação, tipo, etc, porém não incluindo o atributo de quantidade, visto que o insumo poderia ter diferentes quantidades se tratando de diferentes setores, vale lembrar que o usuário final também apresentou a necessidade de visualizar as entradas e saídas de cada setor.

A solução foi mapear uma entidade contendo a movimentação de insumos entre setores (*product_movement*), essa entidade contém atributos relacionados a quantidade, setor de origem, setor de destino ou paciente de destino, isso permite ter um estoque flexível, que é calculado por setores, então para saber qual o estoque do produto B no setor C, basta obter todas as movimentações que incluem o produto B, relacionadas ao setor C, realizando uma subtração entre as movimentações que o destino foi o setor C, menos as movimentações que a origem foi o setor C, gerando assim o total de produtos B no setor C. Exemplificando, temos:

Figura 11 - Explicação da entidade de movimentação



Fonte: Elaborado pelo autor, 2024.

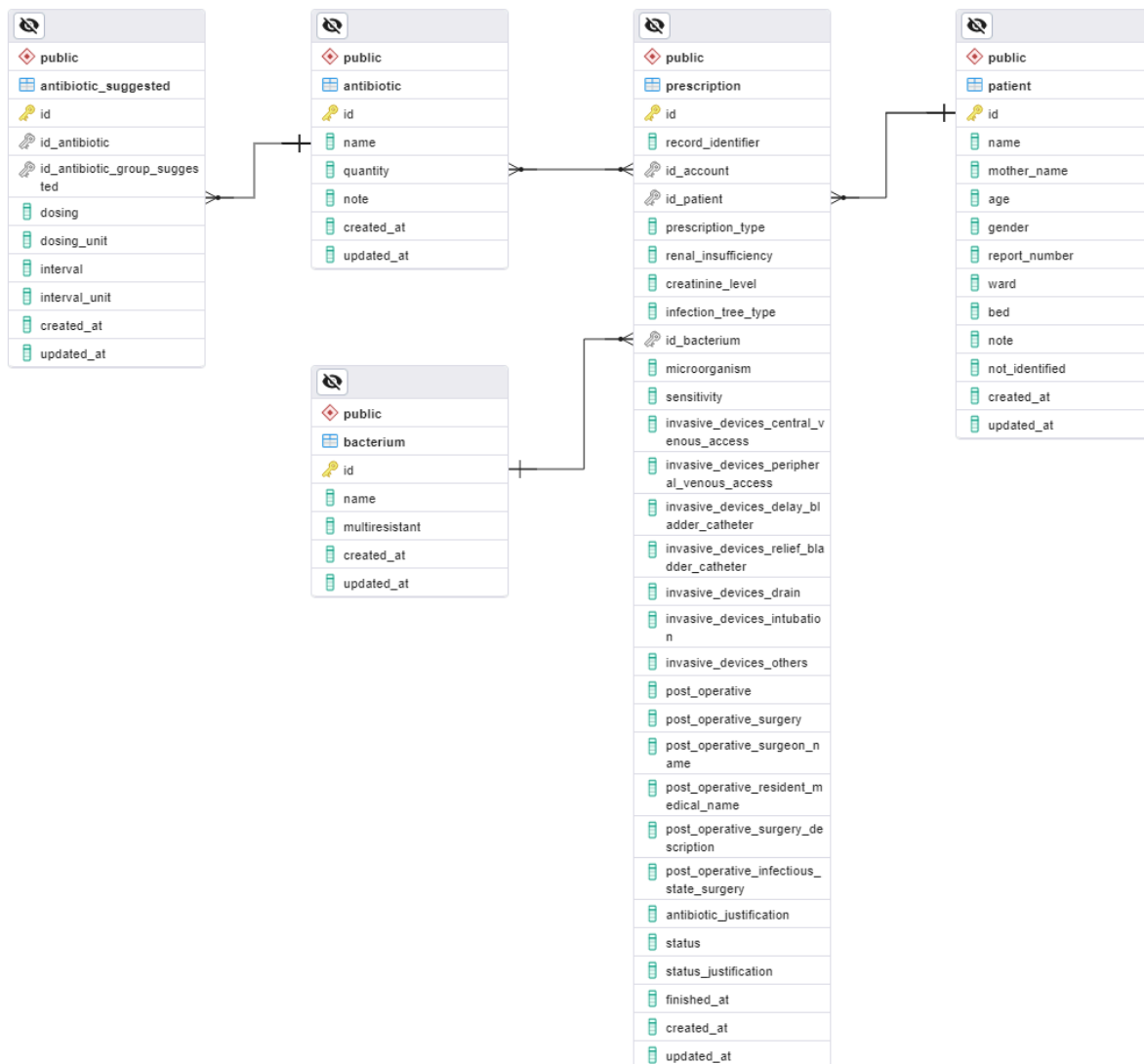
Observa-se que ao mapear todas as movimentações do produto B, entraram no setor C 105 unidades, essas entradas podem ser originadas de um outro setor, ou do cadastro de novos estoques, todavia, as saídas totalizaram 15 unidades, essa movimentação pode acontecer para um outro setor, ou diretamente para um paciente. O estoque real do produto no setor C, é de 90 unidades, visto que é o resultado da subtração das movimentações de entrada (105

unidades) menos as movimentações de saída (15 unidades). Essa abordagem além de controlar o estoque em cada setor, permitiu mapear todas as movimentações de insumos entre os setores e pacientes, agregando confiabilidade ao sistema, pois isso gera logs de auditoria e possibilita também a geração de dados estatísticos para setores administrativos.

Outro ponto pensado com base em auditoria, foi a entidade de nota fiscal (*invoice*), pois ao cadastrar um novo produto ou dar entrada em uma nova quantidade de estoque, há uma obrigação de associação dessa movimentação uma nota fiscal, com isso, é possível gerar um rastreamento e evitar movimentações sem nota.

No penúltimo módulo, foi especificado a parte relacionada às prescrições, o desenvolvimento dessa parte surgiu devido a reclamação dos médicos na falta de controle das prescrições dos pacientes, as necessidades incluem gerenciar e acompanhar a evolução do paciente, juntamente com a sugestão de protocolos de medicamentos na hora da prescrição.

Figura 12 - DER referente ao módulo de prescrições



Fonte: Elaborado pelo autor, 2024.

A especificação deste módulo gerou 5 entidades, sendo que a entidade de paciente (patient) já tinha sido mapeadas em fases anteriores, logo a mesma seria reusada. Dito isso, 4 novas entidades surgiram, a mais importante delas, de prescrição (*prescription*), contendo os atributos fornecidos pelo médico-prescritor na hora do cadastro, como informações sobre a doença, nível de creatinina, sensibilidade do paciente, bactéria envolvida caso exista, e a informação mais crucial, o protocolo de tratamento que o paciente será submetido. Em seguida, identificou-se a entidade de antibióticos sugeridos (*antibiotic_suggested*), ela foi necessária para permitir o médico-infetologista cadastrar sugestões de protocolos associados a doenças ou bactérias, logo, o médico-prescritor no momento do cadastro de uma prescrição,

REST, em particular, ganhou ampla aceitação no mercado de trabalho, proporcionando uma abordagem leve e escalável para a troca de dados entre clientes e servidores.

O REST utiliza métodos HTTP padrão, como GET, POST, PUT e DELETE, disponibilizando rotas que podem ser consumidas pelo cliente usando esses métodos, através dessas rotas é possível realizar operações de criação, edição, remoção e listagem dos dados fornecidos, vale salientar que a representação dos dados geralmente ocorre em formatos como JSON (JavaScript Object Notation) ou XML (Extensible Markup Language), facilitando a interoperabilidade entre diferentes sistemas e linguagens de programação. Notou-se que o padrão de API REST, seria a melhor escolha para a disposição e gerenciamento de dados no sistema desenvolvido, visto que esse estilo permite integração de novas aplicações e interfaces sem precisar de nenhuma mudança na parte de dados.

Para desenvolver a Interface de Programação de Aplicação, escolheu-se tecnologias bastante consolidadas no mercado de trabalho, começando com o Node.js, que se trata de um ambiente de execução de código JavaScript do lado do servidor, dessa forma, os desenvolvedores utilizam a linguagem para criar aplicativos server-side, proporcionando um ambiente de execução eficiente e escalável, devido a abordagem assíncrona e baseada em eventos, o processamento de operações de entrada e saída como acesso a bancos de dados, manipulação de arquivos, é extremamente rápido, tornando essa ferramenta referência no mercado, onde grandes empresas como Netflix, Uber, Paypal, IBM, etc, usam essa ferramenta.

No âmbito do gerenciador de banco de dados, a tecnologia escolhida foi o sistema de banco de dados PostgreSQL (PSQL), outra ferramenta muito utilizada globalmente, conhecida por sua robustez, extensibilidade e escalabilidade, ademais o PSQL apresenta código aberto, por essa razão tem uma comunidade ativa com suporte contínuo, que garantem atualizações regulares e a incorporação de novos recursos. Outro ponto que motivou essa escolha foi o fato de que ambientes hospitalares geram muitos dados, e o PSQL se trata de um banco de dados relacional, ou seja, suporta muita carga de dados com possibilidade de execução de consultas complexas.

3.2.3 Projeto de interface

No projeto de interface acontece o planejamento da parte interativa entre o sistema e o usuário final, envolvendo o design visual da interface do usuário (UI) e a experiência do usuário (UX), garantindo uma interação intuitiva e eficiente, o objetivo aqui é criar uma

interface agradável, fácil de usar e que atenda às necessidades dos usuários, isso inclui decisões sobre layout, design gráfico, cores e tipografia, visando uma experiência positiva, usabilidade eficaz e resposta adequada do sistema às ações do usuário. Pressman e Maxim (2016) utilizam uma metáfora interessante comparando a Interface a uma casa, segundo eles a UI é comparada ao conjunto de entradas e saídas, como as portas, janelas e conexões externas, isso permite identificar como as pessoas e objetos fluem para dentro e para fora da casa.

Nessa etapa do medNUTES foi decidido usar a técnica de prototipação, que consiste em criar modelos representativos das telas do sistema, isso foi importante pois as especificações estavam ganhando tamanho, e até o exato momento o sistema já demandava de várias telas, gerando a necessidade de ter um feedback preciso por parte do usuário final antes mesmo de qualquer implementação, essas medidas possibilitaram a identificação de possíveis problemas de usabilidade, validação e eficácia do design, resultando na contribuição para a redução de riscos e custos, pois é muito mais fácil e barato alterar um protótipo do que em um sistema já desenvolvido.

No desenvolvimento dos protótipos, foi usado o Figma, uma ferramenta de design colaborativa baseada na nuvem, utilizada na criação de interfaces de usuário, em conjunto com ela, também foi usado o Material Design, isso se trata de uma série de princípios, padrões e componentes de design desenvolvido pela Google, que tem a ênfase na usabilidade e clareza visual, permitindo a criação de interfaces intuitivas e compreensíveis para os usuários. Lembrando, o medNUTES é um sistema no âmbito hospitalar, logo seria executado em diferentes dispositivos, com diferentes tamanhos de tela, etc, em frente a isso, todos os protótipos foram desenvolvidos de forma responsiva. A princípio, foi definido os layouts da aplicação, conforme abaixo:

Figura 14 - Definição dos layouts da interface

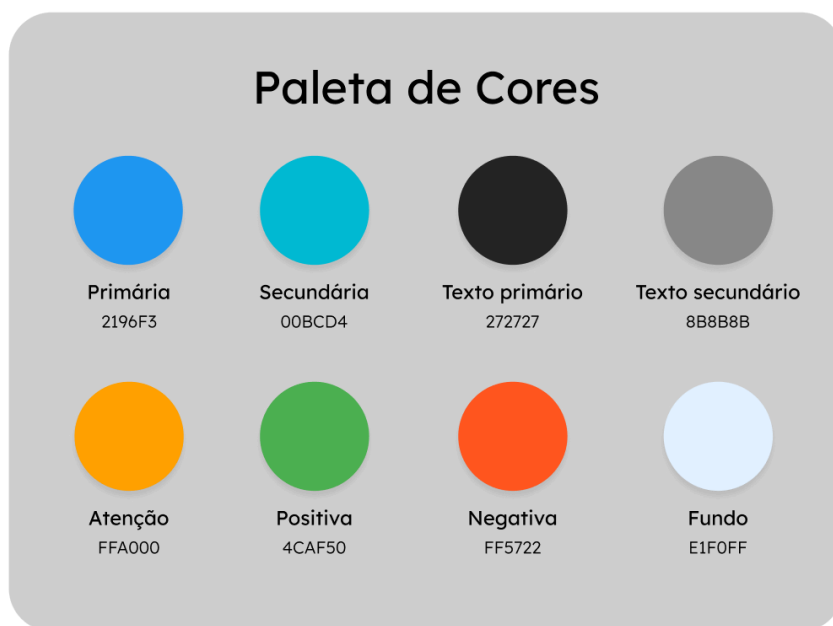


Fonte: Elaborado pelo autor, 2024.

Observa-se que o layout foi desenvolvido basicamente com três componentes principais, o cabeçalho, que é um componente fixo contendo acessos rápidos como, o menu lateral, minha conta, e logout, o segundo componente mapeado foi o menu lateral, que traz ao usuário uma facilidade em acessar todas as páginas da aplicação, reunindo todas opções em um único lugar. Por fim, o último componente trata do conteúdo que será exibido, no caso listagens, botões, entradas de textos, formulários, dentre outros, esse conteúdo aparece centralizado na tela, tanto horizontalmente quanto verticalmente. O layout mobile segue o mesmo padrão, todavia com pequenas mudanças adaptadas para telas menores, como por exemplo o menu lateral, que nesse caso sobrepõe o cabeçalho, o que não acontece no layout *desktop*.

Em paralelo a definição dos layouts, também foi definido a paleta de cores da aplicação, a cor principal escolhida foi o azul, já que não se trata de uma cor muito chamativa, nem clara demais, além disso, como o sistema foi desenvolvido em parceria com o NUTES, a equipe considerou em manter uma cor que remetesse o sistema a essa parceria, sendo o azul a melhor opção. A paleta de cores final do sistema ficou conforme mostrado abaixo:

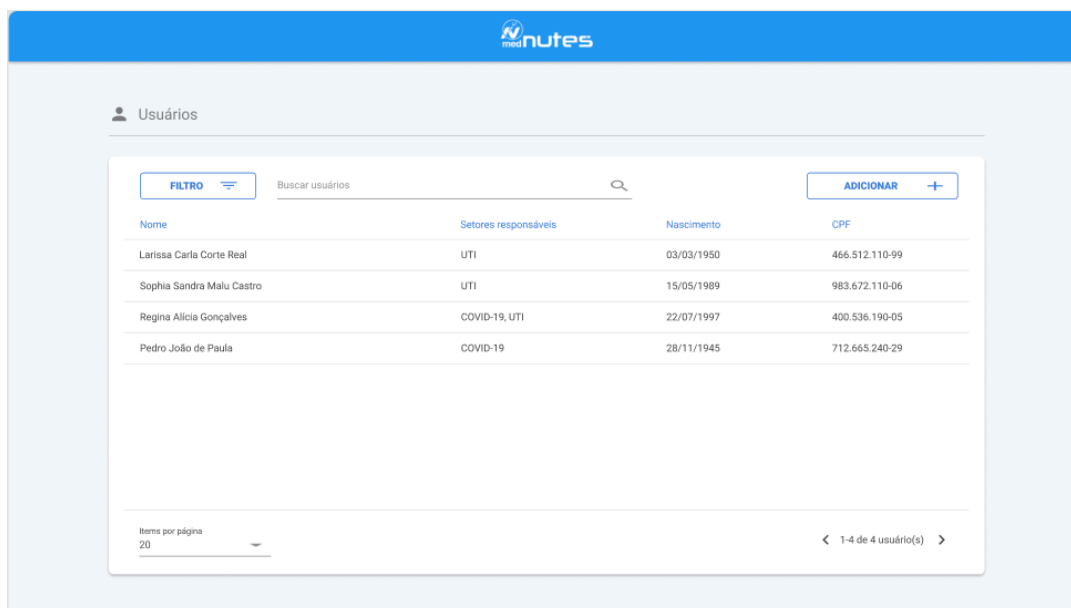
Figura 15 - Paleta de cores



Fonte: Elaborado pelo autor, 2024.

Com toda a base de design definida, iniciou-se a prototipação das telas da aplicação, assim como todas as fases anteriores o processo foi feito por módulos, nessa etapa não seria diferente, o primeiro módulo a ser prototipado foi o de usuários, parte responsável pela gestão de usuários na plataforma. Esse módulo foi prototipado em um novo menu na interface, o de usuários, nele há uma listagem referente a todos os usuários já cadastrados, com opções para filtrar e ordenar, além do mais está disponível um botão de ação para cadastrar novo usuários:

Figura 16 - Listagem dos usuários, filtros, ordenação e botão de adicionar



The screenshot displays the 'mednutes' user management interface. At the top, there is a blue header with the 'mednutes' logo. Below the header, the page title is 'Usuários'. The main content area features a search bar labeled 'Buscar usuários' and a 'FILTRO' button. To the right, there is an 'ADICIONAR +' button. Below these elements is a table with the following data:

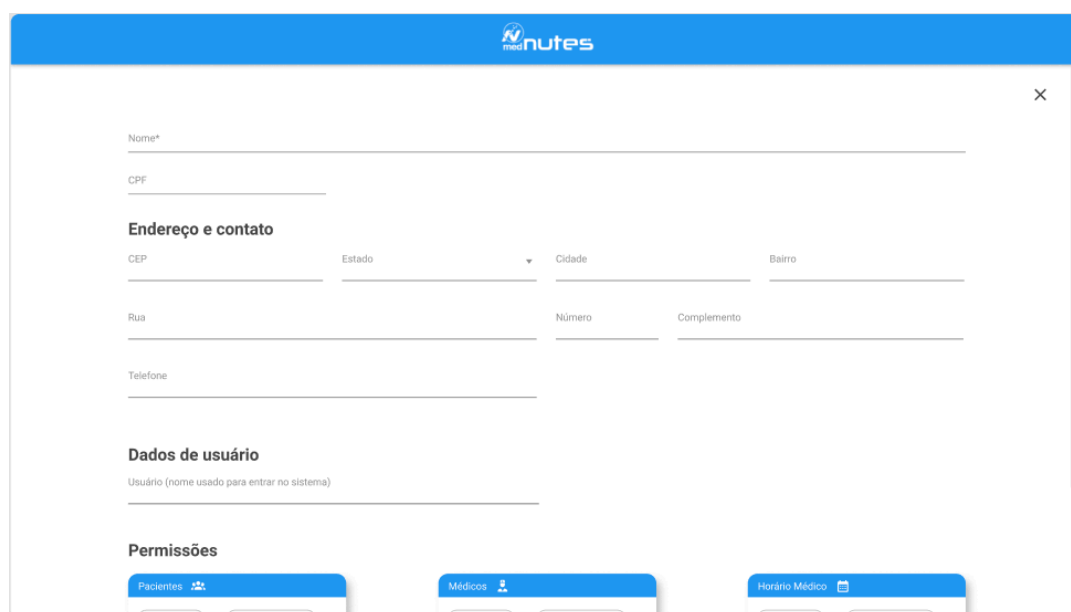
Nome	Setores responsáveis	Nascimento	CPF
Larissa Carla Corte Real	UTI	03/03/1950	466.512.110-99
Sophia Sandra Malu Castro	UTI	15/05/1989	983.672.110-06
Regina Alicia Gonçalves	COVID-19, UTI	22/07/1997	400.536.190-05
Pedro João de Paula	COVID-19	28/11/1945	712.665.240-29

At the bottom of the table, there is a pagination control showing 'Items por página: 20' and a navigation arrow indicating '1-4 de 4 usuário(s)'.

Fonte: Elaborado pelo autor, 2024.

Prosseguindo, foi prototipado o formulário de usuários, a ideia é que ele suporte tanto cadastrar um novo usuário, quanto visualizar e editar um usuário já cadastrado, ao adicionar, ele será exibido com todas as entradas em branco, ao editar ele será exibido com os dados do usuário selecionado na listagem, segue o protótipo:

Figura 17 - Formulário de usuário no modo de cadastro



The screenshot shows the user registration form in the 'mednutes' system. The form is organized into several sections:

- Nome***: A text input field for the user's name.
- CPF**: A text input field for the user's CPF.
- Endereço e contato**: A section containing several input fields: CEP, Estado (with a dropdown arrow), Cidade, Bairro, Rua, Número, and Complemento.
- Telefone**: A text input field for the user's phone number.
- Dados de usuário**: A section with a sub-label 'Usuário (nome usado para entrar no sistema)' and a text input field.
- Permissões**: A section with three buttons: 'Pacientes' (with a person icon), 'Médicos' (with a person icon), and 'Horário Médico' (with a calendar icon). Each button has a 'Visualizar' and 'Adicionar/Editar' sub-button.

Fonte: Elaborado pelo autor, 2024.

Ainda no formulário, foi definido a parte de permissões dos usuários, isso permite suportar os vários tipos de perfis existentes em ambientes hospitalares, tornando as permissões flexíveis quando necessário:

Figura 18 - Parte do formulário com as permissões do usuário



Fonte: Elaborado pelo autor, 2024.

Observa-se que as permissões são separadas por menu, onde quem está cadastrando o usuário vai definir se ele tem acesso ao menu X, Y e Z, além de permitir o acesso a menus específicos há também a possibilidade de definir as permissões dentro dos próprios menus, sendo possível que um usuário acesse o menu X porém somente visualizando os dados, e outro usuário acesse o mesmo menu, mas visualizando, cadastrando e editando os dados.

Ao finalizar a parte de usuários, a equipe prototipou o módulo de pacientes, que foi dividido em dois novos menus dentro da interface, o menu de pacientes e o de agendamentos. O de pacientes foi desenvolvido contendo uma listagem dos pacientes já cadastrados, com opção para buscar um determinado paciente usando os filtros, ou ordená-los por nome, por exemplo. Além do mais, ao clicar no botão de adicionar, o fluxo para adicionar um novo paciente seria acionado:

Figura 19 - Listagem dos pacientes, filtros, ordenação e botão de adicionar

Prontuário	Nome	Nascimento	CPF	CNS
33161277	Maria da Silva Ferreira da Neves	03/03/1950	466.512.110-99	123456789101112
37231277	Victor Cassiano da Silva	15/05/1989	983.672.110-06	786512789101110
48361277	Lucinaldo Florentino da Silva	22/07/1997	400.536.190-05	123013245701117
97861277	Ygor Barbosa Ferreira	28/11/1945	712.665.240-29	173454319101114

Fonte: Elaborado pelo autor, 2024.

O próximo passo ainda nesse menu, foi prototipar o formulário de cadastro ou edição de um paciente, requisitando somente os dados necessários na ficha e padronizando entradas como nome de cidade, ocupação, sexo, dentre outras, permitindo campos de escolha ao invés de texto aberto, vale ressaltar no cadastro esse formulário será exibido com os campos em branco, ao editar, os campos são inicializados com os dados do paciente a ser editado:

Figura 20 - Formulário de pacientes no modo de cadastro

DADOS

Paciente não identificado

Classificação ▼

Nome do paciente* Nome social

CPF Cartão SUS Identidade Orgão expedidor Registro de nascimento Carteira de trabalho

Nascimento Nacionalidade ▼ Naturalidade ▼

Pai Mãe

Sexo ▼ Cor ▼ Escolaridade ▼

Estado civil ▼ Profissão ▼

Observações

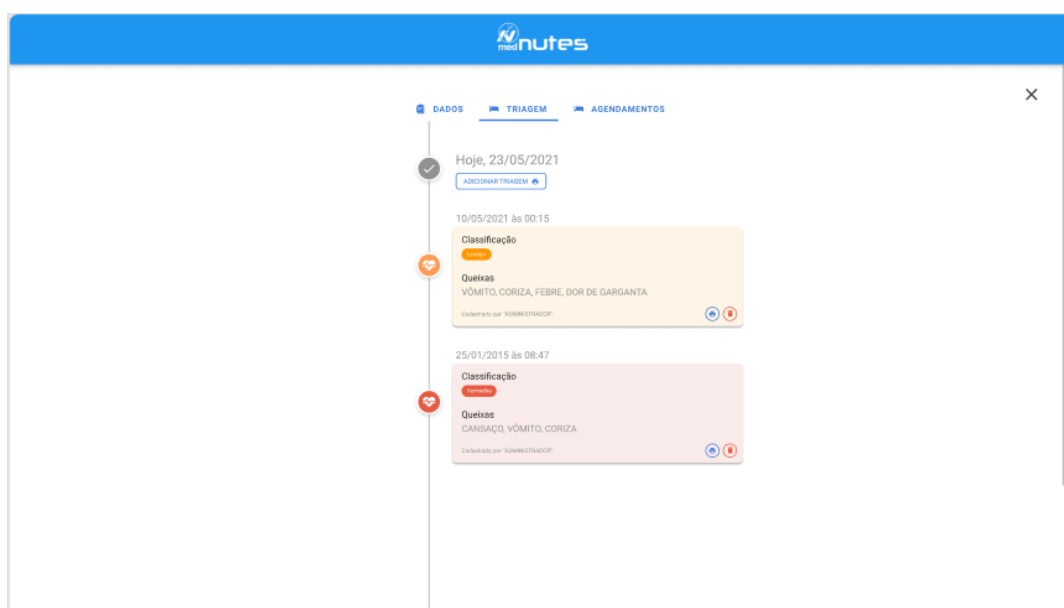
Contato

Telefone Email

Fonte: Elaborado pelo autor, 2024.

Ainda no módulo de pacientes, foi mapeado a parte relacionadas às triagens, a exibição delas acontecem ao abrir a ficha de um paciente e depois selecionar a aba de triagens, listando todas as triagens realizadas em forma de linha do tempo, um botão com ação de adicionar uma nova triagem também foi identificado, como vemos abaixo:

Figura 21 - Listagem das triagens de um paciente em forma de linha do tempo



Fonte: Elaborado pelo autor, 2024.

Para o cadastro de uma nova triagem, foi prototipado um formulário com os dados obtidos nesse processo, como sinais vitais, peso, gravidade do paciente, etc:

Figura 22 - Formulário para cadastrar uma nova triagem

The screenshot shows a web application interface for 'mednotes'. At the top, there is a blue header with the logo. Below the header, there are navigation tabs: 'DADOS', 'TRIAGEM', and 'AGENDAMENTOS'. The current date is 'Hoje, 23/05/2021'. A modal window is open, titled 'Classificação', with a close button (X). The modal contains the following sections:

- Classificação:** Five buttons labeled AZUL, VERDE, AMARELA, LARANJA, and VERMELHA.
- Queixas:** Four buttons labeled TOSSSE, CANSAÇO, VÔMITO, and DIARREIA.
- Queixas adicionais:** A text input field.
- Sinais vitais:** Six input fields labeled PA, T (C), P, HGT (mg/dL), Peso (kg), and SPO (%).
- SALVAR:** A blue button with a checkmark icon.

Fonte: Elaborado pelo autor, 2024.

Após a conclusão do módulo de paciente, a equipe partiu para o módulo de agendamentos, sendo responsável por gerenciar todos os agendamentos do hospital, com opções de listagem, cadastro, edição, confirmar presença ou falta, dentre outras. Ao mapear as funcionalidades em interface, foi gerado um novo menu no sistema, o menu de agendamentos. Um dos pedidos dos *stakeholders* foi a visualização dos agendamentos em função de um calendário, logo a equipe teve a ideia de dividir a tela exibindo dois componentes, um calendário e uma listagem, o fluxo funcionaria da seguinte forma, ao selecionar uma data no calendário, a listagem seria atualizada exibindo os agendamentos para a data selecionada. A listagem contém dados como o nome do paciente, o médico que irá atendê-lo, a data do agendamento e o status do mesmo. Ademais há possibilidade de filtrar os agendamentos por médicos, imprimir a lista de agendamentos e adicionar novo agendamento.

Figura 23 - Formulário para cadastrar uma nova triagem

The screenshot shows the 'mednotes' application interface. At the top, there's a blue header with the 'mednotes' logo. Below it, the page title is 'Agendamentos'. There's a search bar for 'Filtrar médico' with the text 'Nenhum médico selecionado'. The main content area is split into two parts. On the left, there's a calendar for October 2021, with the date 'qui., 21 de out.' highlighted. On the right, there's a sidebar titled 'Agenda 21/10/2021' showing appointment details for 'João Silva Ferreira' and 'Mariane Roseane Dantas' at '25/10/2021 às 08:55'. There are buttons for 'IMPRIMIR' and 'AGENDAR'.

Fonte: Elaborado pelo autor, 2024.

Na parte de cadastro de um novo agendamento, a ideia pensada foi exibir um formulário de cadastro ao clicar na ação “Agendar”, esse formulário também serviria para editar um agendamento existente, onde entradas seriam preenchidas com os valores do agendamento selecionado, já no caso de cadastro, as entradas seriam exibidas em branco, conforme abaixo:

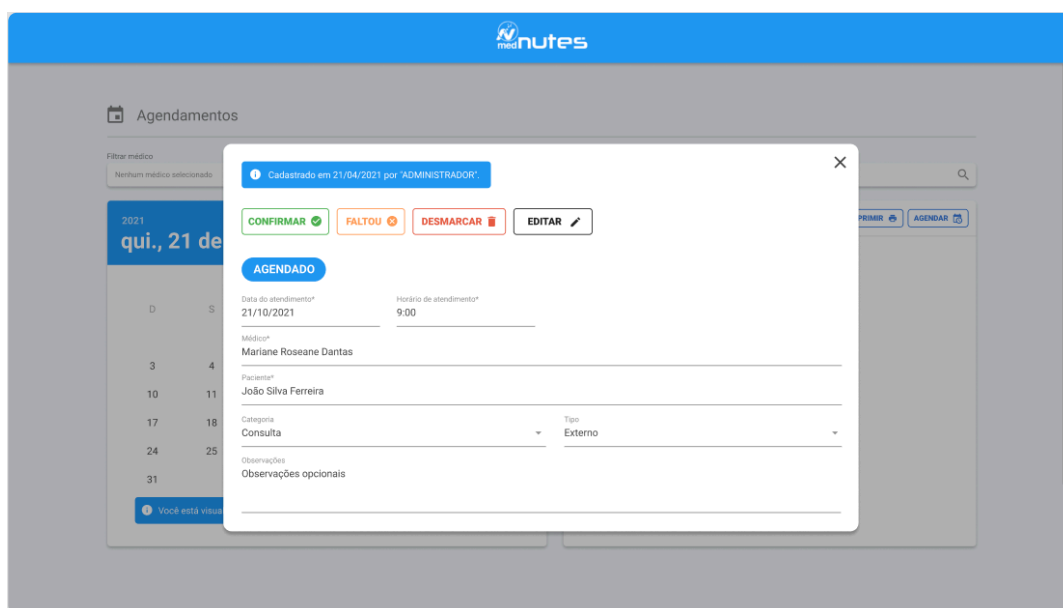
Figura 24 - Formulário para cadastrar um novo agendamento

The screenshot shows the 'mednotes' application interface with a modal form for creating a new appointment. The modal form is centered on the screen and contains the following fields: 'Data do atendimento*', 'Horário de atendimento*', 'Médico*', 'Paciente*', 'Categoria', 'Tipo', and 'Observações'. There are 'CANCELAR' and 'SALVAR' buttons at the bottom of the modal. The background shows the same calendar and appointment details as in Figure 23.

Fonte: Elaborado pelo autor, 2024.

Outra necessidade desse módulo foi a possibilidade de confirmação ou não a presença do paciente, nesse caso, mapeou-se botões de ações para confirmar o paradeiro do paciente e até mesmo desmarcar o recurso, essas ações devem ser exibidas ao selecionar um agendamento já cadastrado, exibindo a seguinte interface:

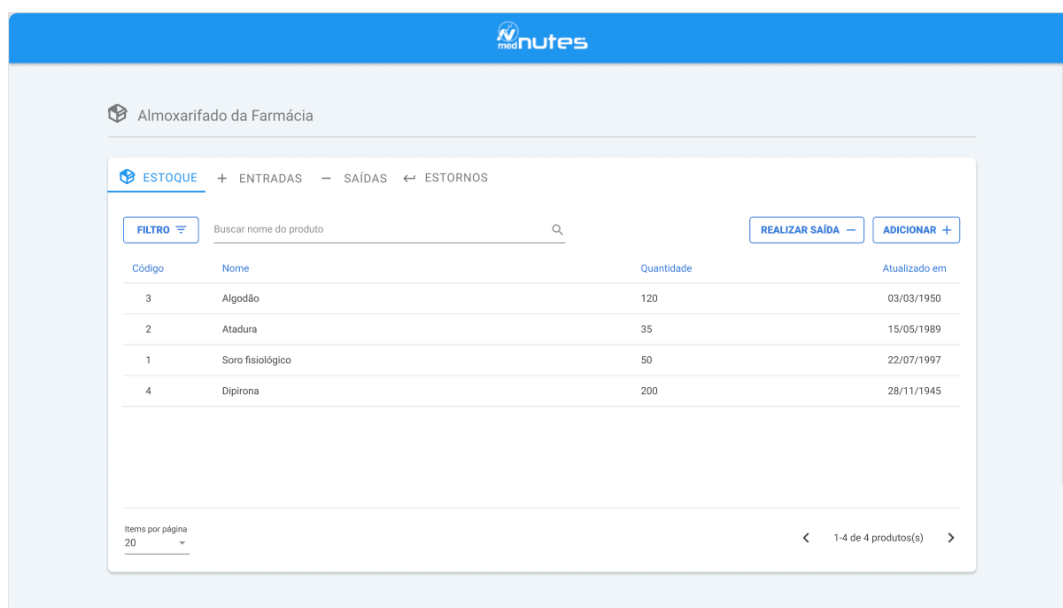
Figura 25 - Ações de confirmar presença, falta e desmarcar o agendamento



Fonte: Elaborado pelo autor, 2024.

Avançando, prototipou-se o módulo de almoxarifado, essa parte teve uma atenção especial da equipe pois englobava funcionalidades extremamente importantes, possibilitando de forma prática a gerência de todos os insumos e seus estoques, rastreando as entradas e saídas entre todos os setores do hospital. Esse módulo gerou um novo menu para a interface, chamado Almoxarifado, composto por 4 abas principais, a primeira apresenta uma listagem dos insumos, junto com ações de filtragem, ordenação, adicionar novo insumo e confirmar saída de insumo. Sendo mapeados da seguinte forma:

Figura 26 - Aba de estoque contendo a listagem dos insumos



Almoxarifado da Farmácia

ESTOQUE + ENTRADAS - SAÍDAS ← ESTORNOS

FILTRO

Código	Nome	Quantidade	Atualizado em
3	Algodão	120	03/03/1950
2	Atadura	35	15/05/1989
1	Soro fisiológico	50	22/07/1997
4	Dipirona	200	28/11/1945

Itens por página: 20

< 1-4 de 4 produtos(s) >

Fonte: Elaborado pelo autor, 2024.

Ao clicar na ação de adicionar, o sistema deverá exibir o formulário de cadastro de um novo insumo, o mesmo foi prototipado e contém entradas referentes a um insumo, como nome, classificação, validade, estoque, etc. Visando a padronização dos dados, a entrada que contém o nome do insumo não deve ser um texto aberto, e sim uma entrada selecionável, onde os valores devem ser obtidos através de tabelas de nomenclatura definidos pelo SUS, evitando que haja vários cadastros com nomes diferentes referenciando um mesmo insumo. Mediante a isso, o resultado do protótipo foi:

Figura 27 - Formulário de cadastro de insumo

Almoxarifado da Farmácia

ESTOQUE + ENTRADAS - SAÍDAS ← ESTORNOS

FILTRO

Nome do produto*

Estoque

Unidade de estoque

Validade DD / MM / AAAA

Classificação NÃO INFORMADO

CANCELAR SALVAR

Atualizado em

03/03/1950

15/05/1989

22/07/1997

28/11/1945

Itens por página 20

1-4 de 4 produtos(s)

Fonte: Elaborado pelo autor, 2024.

Outra ação mapeada na listagem dos insumos foi a de realizar saída de estoque, essa necessidade gerou um novo formulário, apresentando as entradas em forma de um passo-a-passo, sendo necessário informar se o recurso será enviado para um outro setor ou para um paciente, e definir quais insumos juntamente com as quantidades que serão transferidos.

Figura 28 - Formulário de realização de saídas de insumos

Almoxarifado da Farmácia

ESTOQUE + ENTRADAS - SAÍDAS ← ESTORNOS

FILTRO

Selecione o tipo de saída

Selecione o destino

Selecione os produtos e defina a quantidade

FILTRO Buscar nome do produto

Código	Nome	Quantidade	Atualizado em
3	Algodão	120	03/03/1950
2	Atadura	35	15/05/1989
1	Soro fisiológico	50	22/07/1997
4	Dipirona	200	28/11/1945

VOLTAR EFETUAR SAÍDA

Atualizado em

03/03/1950

15/05/1989

22/07/1997

28/11/1945

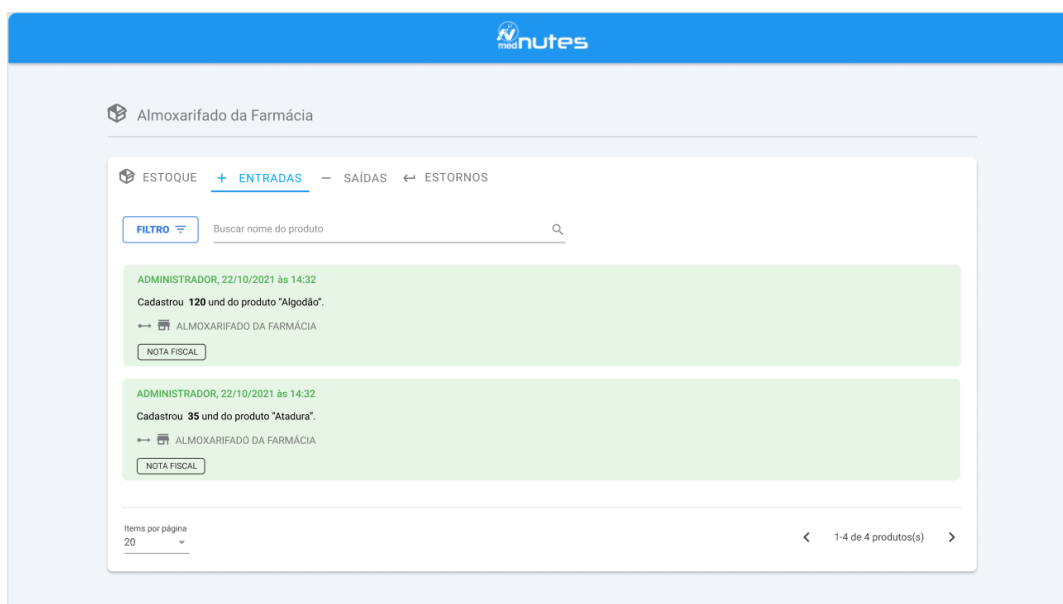
Itens por página 20

1-4 de 4 produtos(s)

Fonte: Elaborado pelo autor, 2024.

Concluiu-se assim todos os protótipos relacionados às ações da aba de Estoque. As próximas 3 abas, a equipe mapeou pensando em exibir as movimentações de insumos, gerando abas referente às entradas, saídas e estornos de recursos respectivamente. Cada aba foi mapeada com um componente de listagem, apresentando ações de filtragem e ordenação das movimentações. Segue abaixo a aba de entrada de estoque, além dos filtros e ordenação, foi mapeada uma ação de nota fiscal, que deverá baixar a nota fiscal atrelada a movimentação.

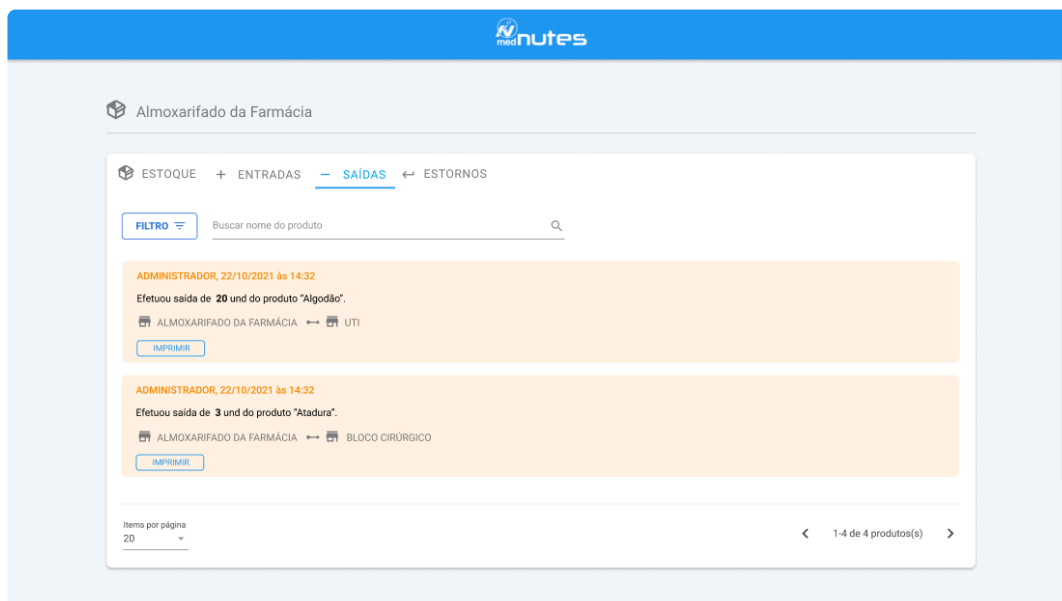
Figura 29 - Listagem de entradas de insumos



Fonte: Elaborado pelo autor, 2024.

A aba de saída foi mapeada com uma listagem de todas as movimentações de saídas, em cada item foi adicionado uma ação de “Imprimir”, essa ação deverá imprimir os dados daquela movimentação específica, sendo usada internamente pelos funcionários do hospital, conforme abaixo:

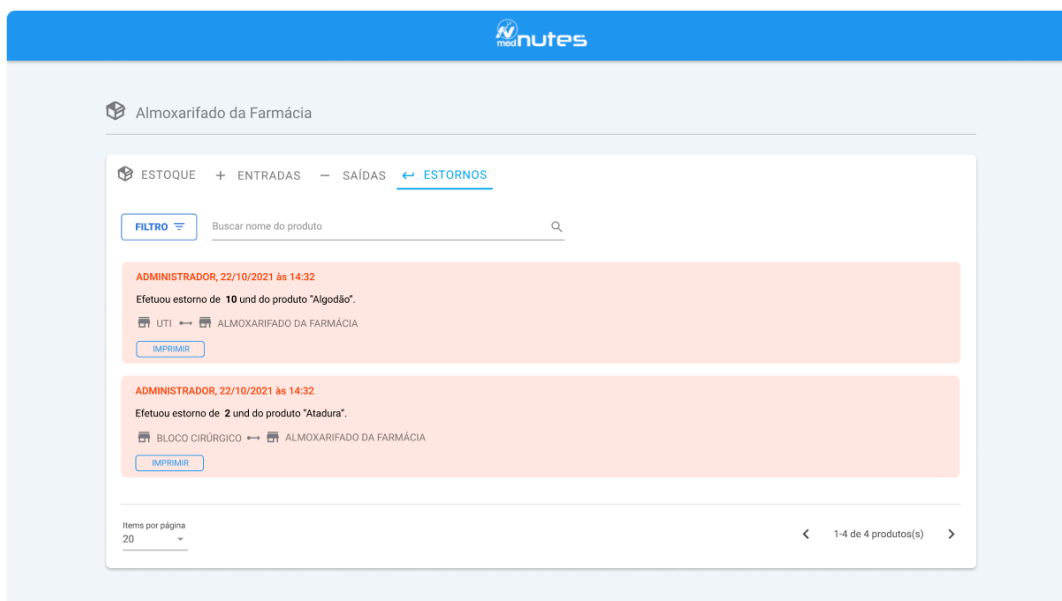
Figura 30 - Listagem de saídas de insumos



Fonte: Elaborado pelo autor, 2024.

A aba de estornos funciona da mesma forma da aba de saída, contendo até a mesma ação de imprimir, a única diferença é o conteúdo da listagem, que nesse caso exhibe as movimentações de estorno, como vimos adiante:

Figura 31 - Listagem de estornos de insumos



Fonte: Elaborado pelo autor, 2024.

Ao finalizar o módulo de almoxarifado, foi iniciado o protótipo do penúltimo módulo, o de prescrições, essa parte do sistema será usado pelos médicos, tanto médico-prescritor quanto médico-infetologista, e onde o objetivo é ter um gerenciamento mais eficiente das prescrições, evitando duplicatas, protocolos de tratamento incorretos, e dar a possibilidade do médico-infetologista sugerir protocolos de tratamento para uma determinada doença. Como resultado na interface, esse módulo gerou 2 novos menus, primeiro abordaremos o de prescrições, esse módulo contém um componente com a listagem de todas as prescrições já cadastradas no sistema, contendo o nome do paciente, o médico responsável, status e data de cadastro da mesma. Além disso mapeou-se ações de filtragem e ordenação, e a opção de adicionar uma nova prescrição com o botão “Adicionar”.

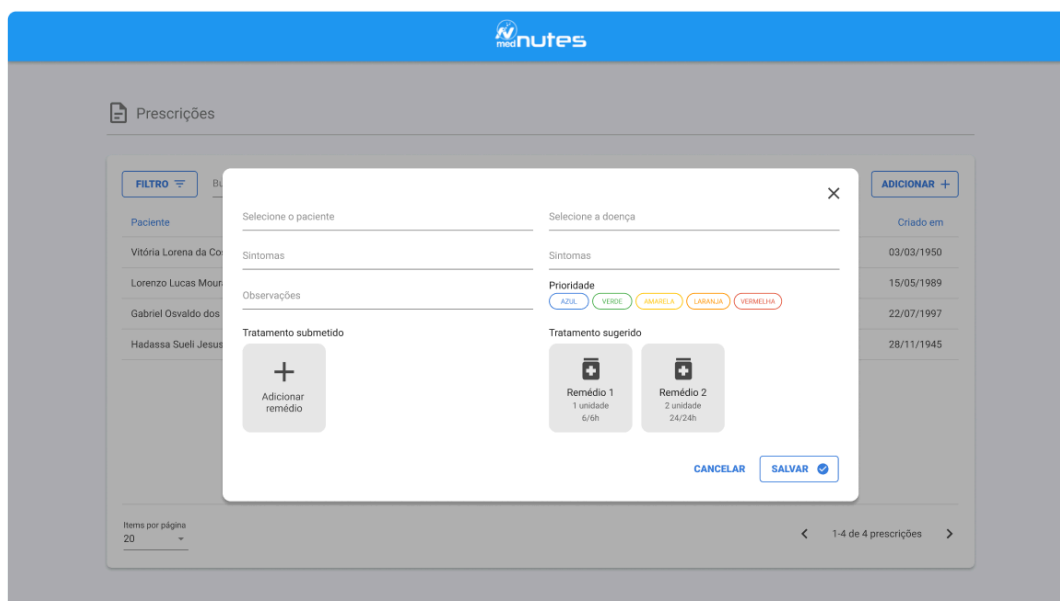
Figura 32 - Listagem de prescrições

Paciente	Médico	CPF do paciente	Status	Criado em
Vitória Lorena da Costa	Patricia Valentina Rosa Fogaça	193.028.230-32	Em tratamento	03/03/1950
Lorenzo Lucas Moura	Isabella Marina Betina Cavalcanti	357.302.005-44	Em tratamento	15/05/1989
Gabriel Osvaldo dos Santos	Patricia Valentina Rosa Fogaça	382.318.533-08	Finalizada	22/07/1997
Hadassa Sueli Jesus	Igor Severino Freitas	562.770.766-63	Finalizada	28/11/1945

Fonte: Elaborado pelo autor, 2024.

Para atender a ação do botão de adicionar uma nova prescrição, a equipe desenvolveu um formulário de cadastro, contendo os dados referentes a uma prescrição, como nome do paciente, doença, sintomas, prioridade e tratamento submetido para a doença do paciente. Vale lembrar que ao selecionar a doença, será exibido para o médico-prescritor uma sugestão de tratamento enviada pelo médico-infetologista, dessa forma em casos delicados há uma contribuição mútua entre ambos. O formulário foi mapeado conforme abaixo:

Figura 33 - Cadastro de prescrição com tratamento sugerido pelo médico-infectologista



The screenshot displays the 'mednotes' application interface for prescription management. A modal window is open, allowing the user to add a suggested treatment. The modal contains the following fields and options:

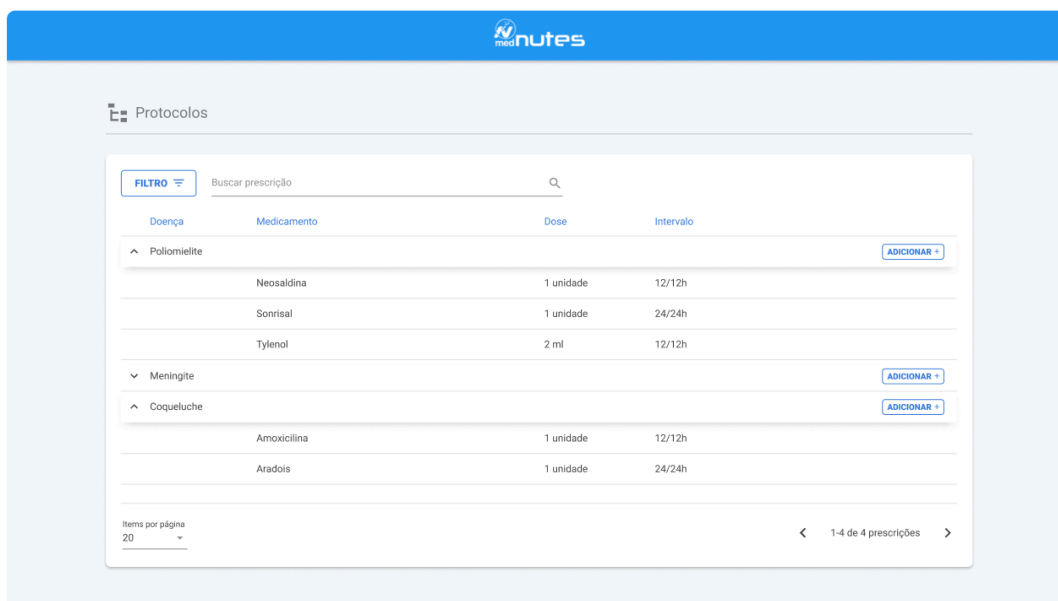
- Seleção de paciente:** A text input field for patient selection.
- Seleção de doença:** A text input field for disease selection.
- Sintomas:** A text input field for symptoms.
- Prioridade:** A set of radio buttons with color-coded labels: AZUL, VERDE, AMARELA, LARANJA, and VERMELHA.
- Tratamento sugerido:** A section with two medication cards:
 - Remédio 1:** 1 unidade, 6/h.
 - Remédio 2:** 2 unidade, 24/24h.
- Buttons:** 'CANCELAR' and 'SALVAR' (with a save icon).

The background interface shows a list of patients and a table of existing prescriptions with columns for 'Criado em' and dates.

Fonte: Elaborado pelo autor, 2024.

O segundo menu gerado na interface por esse módulo, foi o de protocolos, essa parte foi pensada exclusivamente para o médico-infectologista, que tem a liberdade para cadastrar sugestões de tratamento associados a cada doença, isso, como dito anteriormente, será exibido para o médico-prescritor no momento do cadastro de uma prescrição. Na tela de protocolos, foi desenvolvido uma listagem com todos os protocolos e seus respectivos medicamentos, porém diferente de listagens anteriores, cada doença pode ter vários medicamentos associados, logo a exibição foi adaptada, ademais ações de filtragem, ordenação e cadastro de novos medicamentos também foram mapeados:

Figura 34 - Listagem dos protocolos associados a cada doença



Doença	Medicamento	Dose	Intervalo	
Poliomielite	Neosaldina	1 unidade	12/12h	ADICIONAR +
	Sonrisal	1 unidade	24/24h	
	Tylenol	2 ml	12/12h	
Meningite				ADICIONAR +
Coqueluche	Amoxicilina	1 unidade	12/12h	ADICIONAR +
	Aradois	1 unidade	24/24h	

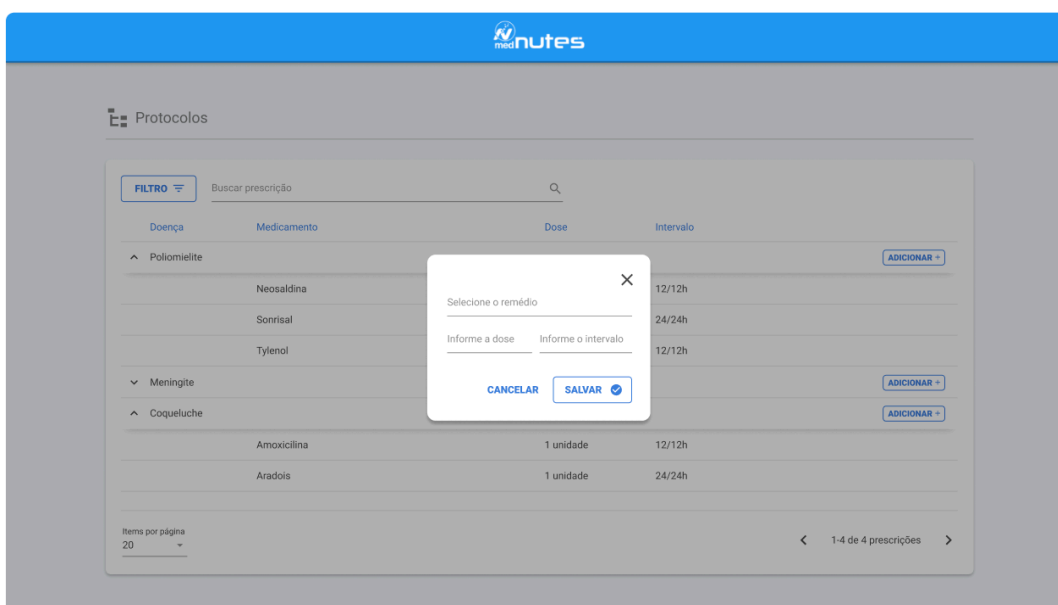
Items por página: 20

1-4 de 4 prescrições

Fonte: Elaborado pelo autor, 2024.

Ao executar a ação de adicionar, será exibido um formulário para cadastro do medicamento, o mesmo será atrelado a doença selecionada, esse formulário foi prototipado com os dados referentes ao tratamento, como o nome do medicamento, dose e intervalo recomendado, conforme abaixo:

Figura 35 - Cadastro de tratamento



Selecione o remédio

Informe a dose

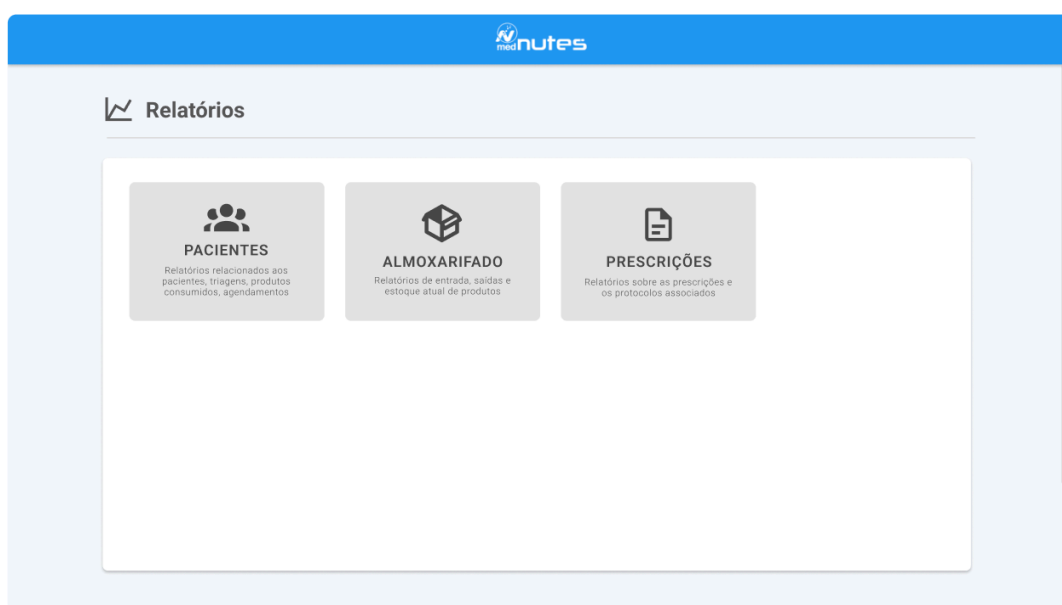
Informe o intervalo

CANCELAR SALVAR

Fonte: Elaborado pelo autor, 2024.

Por fim, foram feitos os protótipos relacionados ao módulo de relatórios, o último do sistema. O resultado final dessa parte gerou um novo menu na interface do medNUTES, o menu de relatórios, as necessidades citadas pelos stakeholders envolvia a geração de relatórios de forma prática, com possibilidade de aplicação de filtros e realizar a impressão do que foi gerado. Buscando atingir o que foi pedido, a equipe decidiu listar os tipos de relatórios de forma clara na tela, mas ainda possibilitando a inclusão de filtros adicionais, resultando no protótipo abaixo:

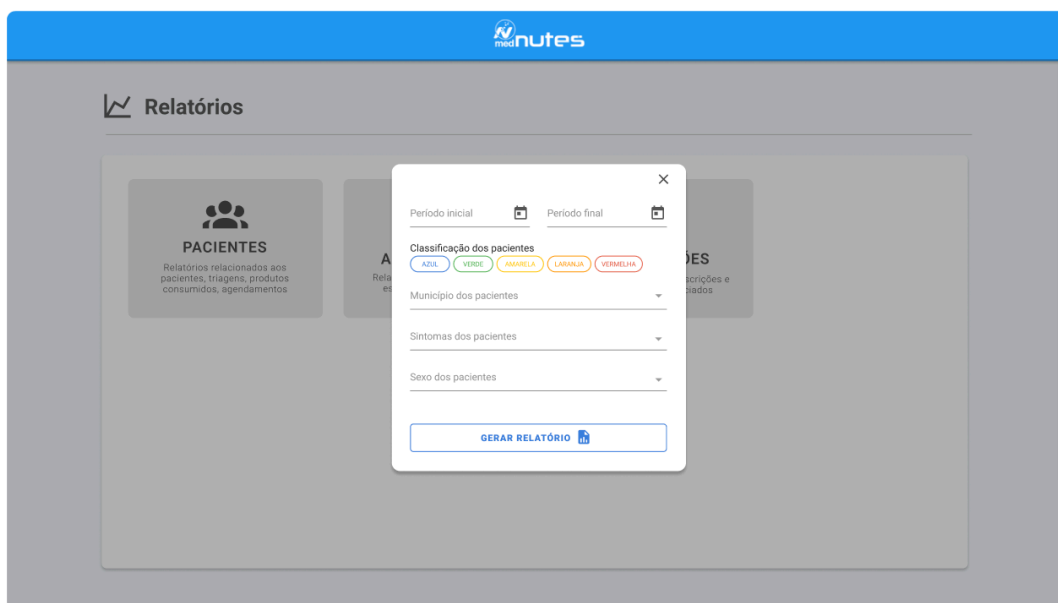
Figura 36 - Escolha de relatórios



Fonte: Elaborado pelo autor, 2024.

A ideia é que ao clicar em qualquer um dos tipos de relatórios, os filtros relacionados serão exibidos, da seguinte forma:

Figura 37 - Filtros do relatório de pacientes



Fonte: Elaborado pelo autor, 2024.

Vale ressaltar que os filtros são opcionais, logo o usuário decide preencher ou não. Por fim, é necessário executar a ação de gerar relatório e o mesmo será gerado e baixado automaticamente, nota-se que a modelagem da interface permitiu que com 2 cliques o usuário tenha um relatório em mãos, atingindo assim as necessidades dos stakeholders de realizar essa atividade de forma prática e rápida.

Ao final da prototipação das telas ainda restava um último passo no projeto de interface, escolher as tecnologias que seriam usadas na etapa de implementação do front-end. A equipe já tinha desenvolvido outros sistemas antes do medNUTES, devido a isso havia uma experiência com a *framework* de front-end chamada Vue.js, essa ferramenta tem o objetivo de construir interfaces modernas e confiáveis usando linguagem JavaScript ou Typescript juntamente com tecnologia SPA (Single Page Application), essa tecnologia permite que toda a interface seja carregada uma só vez, atualizando apenas o conteúdo principal de forma assíncrona, impactando positivamente no desempenho e eficiência da aplicação. Vale ressaltar que o Vue.js é extremamente consolidado a nível de mercado, onde várias empresas globais como Facebook, Netflix, Xiaomi, Adobe, dentre outras, fazem o uso dela levando a decisão da equipe de usar a ferramenta para implementar toda a interface do sistema.

Assim como citado anteriormente, a construção do design do sistema seguiu regras e diretrizes definidos pelo Material Design da Google, dessa forma foi feito uma busca por *frameworks* que fornecessem componentes prontos e que seguissem essas especificações,

aumentando exponencialmente a velocidade de codificação, maior agilidade e menor espera para conclusão da implementação. Após algumas pesquisas a equipe decidiu usar o Vuetify, uma estrutura de projeto para o Vue.js que contém componentes que foram criados seguindo os conceitos estabelecidos pelo MD, oferecendo recursos de responsividade e sendo prontos para uso.

3.3 IMPLEMENTAÇÃO

A fase de implementação é aquela em que os programadores, desenvolvedores e engenheiros começam a redigir, efetivamente, o código de programação do software, utilizando como referência os planos concebidos nas etapas anteriores. Neste ponto, cada um dos módulos que foram mapeados na fase de projeto entra em operação, por esse motivo esta pode ser uma das fases mais extensas e desafiadoras “*O estágio mais crítico desse processo é, naturalmente, a implementação do sistema, estágio em que você cria uma versão executável do software.*” (SOMMERVILLE, 2019, p.135).

Como mencionado em etapas anteriores, foi escolhido a metodologia de desenvolvimento ágil Scrum para gerenciar as atividades no processo de desenvolvimento, onde o mesmo se dividiu em vários ciclos chamados de sprints, cada uma contendo duração de 21 dias com 120 horas de carga de trabalho, ao início de cada novo ciclo a equipe levantava as atividades necessárias até atingir a capacidade máxima de horas da sprint, após isso, o ciclo era iniciado. Para conclusão da aplicação foram realizadas 8 sprints no total, sendo que as 7 sprints iniciais foram alocadas para a codificação de funcionalidades novas, e a última focada em mudanças sugeridas pelos *stakeholders*. Durante toda a semana aconteciam reuniões internas rápidas com o objetivo de acompanhamento das atividades inseridas na sprint, permitindo uma maior comunicação e alinhamento sobre o que estava sendo desenvolvido, dessa forma nenhum membro ficava ocioso aguardando atividades de outros para continuar.

Como resultado dessa metodologia de desenvolvimento, ao final de cada ciclo uma versão executável do sistema era entregue ao usuário final, permitindo que o mesmo validasse o que foi produzido, verificando a coerência dos fluxos e se os resultados gerados estavam conforme o que foi pedido, mas, para que isso acontecesse o entregável deveria ser minimamente operável, apresentando um fluxo completo desde da interface do sistema até o processamento e armazenamento dos dados, em frente a isso, cada sprint mesclava tarefas relacionadas ao back-end e front-end, no back-end as atividades giravam em torno do

gerenciamento do banco de dados, mecanismos de segurança, criação de rotas na API possibilitando o consumo de recursos, etc, já no front-end acontecia o desenvolvimento das interfaces, onde o usuário iria interagir consumindo as rotas e recursos disponibilizadas pela API.

A divisão de módulos presente nos tópicos apresentados anteriormente, também foi aplicada para a implementação, onde módulos mais independentes como, usuários e pacientes foram priorizados na parte inicial da codificação, e módulos que apresentavam uma maior dependência de outros como, almoxarifado e prescrições foram deixados para o final. Isso influenciou positivamente nas últimas sprints, pois ao começar desenvolver um módulo mais denso e que dependia de outros módulos, essas dependências já teriam sido resolvidas e implementadas.

4 CONSIDERAÇÕES FINAIS

Ao término deste projeto dedicado ao desenvolvimento de um sistema focado no gerenciamento hospitalar, foi possível produzir uma solução comprometida em atender a maior parte dos problemas existentes nesse tipo de organização. A implementação bem-sucedida do módulo de usuários foi extremamente importante para a organização e segurança no sistema, possibilitando a proteção das informações em diferentes níveis e permissões. Já o módulo de pacientes e consultas permitiu ao cliente final um controle muito mais detalhado sobre os pacientes, viabilizando o acompanhamento de triagens, insumos consumidos, gerenciamento de consultas e priorização nas filas de atendimento, outro ponto positivo foi a padronização de informações associadas a ficha do paciente, evitando dados incorretos, duplicados e faltantes. Na área de gerenciamento de estoque, houve uma otimização nos custos e logística, visto que esse módulo permitiu aos *stakeholders* rastrear qualquer entrada e saída de insumos entre os setores do hospital, acompanhando de onde veio, para onde foi, qual o insumo e qual quantidade movimentada.

Acerca do controle de prescrições, a aplicação permitiu ao cliente final gerir de forma mais inteligente e monitorado esse processo, onde foi possível o médico-prescritor receber sugestões de tratamento registrados por médicos-infecionologistas, diminuindo o abuso de doses de medicamentos, tratamentos não eficazes e falta de comunicação entre médicos, todas essas vantagens geraram uma maior eficácia na gestão das prescrições, inclusive na redução de custos e medicamentos. Por fim, o medNUTES também possibilitou o levantamento de relatórios acerca de todos os dados manipulados no sistema, onde essas informações foram cruzadas de forma a levantar indicadores extremamente importantes para as áreas gerenciais, influenciando positivamente em decisões desse cunho.

O uso de padrões na aplicação também trouxe vantagens para os usuários como, a uniformização de entradas, evitando a duplicidade de dados e ganho de tempo ao interagir com o sistema, vale ressaltar o fato de permitir filtros e ordenações em vários menus, trazendo uma maior produtividade aos funcionários, ajudando-os a encontrar e organizar as informações requeridas. A estruturação da interface também seguiu padrões para disposição dos layouts e componentes, resultando em uma curva de aprendizado muito rápida por parte dos usuários, visto que tudo foi pensado com foco na praticidade e usabilidade.

Em conclusão, o software desenvolvido neste projeto não é apenas uma conquista técnica, mas uma ferramenta destinada a transformar positivamente a vida das pessoas, contribuindo para uma prestação de serviços de saúde mais eficiente.

Para trabalhos futuros percebeu-se pontos de melhora no sistema, um deles foi referente a questão dos filtros, o cliente reportou que seria interessante aplicar filtros múltiplos, por exemplo filtrar os pacientes por cidade e por data de nascimento ao mesmo tempo, essa estrutura até o momento no sistema ainda não é suportada. Outra futura melhora na aplicação seria permitir outros tipos de saída de insumos no módulo de almoxarifado, atualmente ao realizar a saída de um determinado insumo só existem 2 classificações, saídas de um insumo para outro setor ou saída para um determinado paciente, porém em um cenário real, outros tipos de saídas também acontecem.

REFERÊNCIAS

AL-SAQQA, Samar; SAWALHA, Samer; ABDELNABI, Hiba. Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, v. 14, n. 11, 2020.

DAĞLI, Duygu. Streamlining a Hospital Information System. Ankara: the Degree of MBA Unpublished a Thesis Submitted to Middle East Technical University, 2011.

DAVIS, W. Michael. *Computerizing Healthcare Information*. USA: Irwin Professional Publishing, 1994.

DEACON, John. Model-view-controller (mvc) architecture. [2009]. Disponível em: <http://www.jdl.co.uk/briefings/MVC.pdf>. Acesso em: 10 mar. 2006.

DEMIREL, Demokaan. (2018). Hospital Management Information Systems in Health Sector and Development in Turkey.

HOMAYOUNFAR, Payam. Process mining challenges in hospital information systems. In: *PROCEEDINGS OF THE FEDERATED CONFERENCE ON COMPUTER SCIENCE AND INFORMATION SYSTEMS - FEDCSIS*, Wroclaw, Poland, 2012, p. 1135–1140.

ISKANDAR, Taufan Fadhilah et al. Comparison between client-side and server-side rendering in web development. In: *IOP CONFERENCE SERIES: MATERIALS SCIENCE AND ENGINEERING*, 801. 2020. Anais... IOP Publishing, 2020.

LEAVITT, Mark. Electronic Medical Records (EMR). In: *Healthcare Information Systems*. Auerbach Publications, 2002. p. 269-298.

LIU, Zheng et al. Reliability Modeling and Analysis of Hospital Information System Based on Microservices. In: *2021 IEEE International Conference on Progress in Informatics and Computing (PIC)*. IEEE, 2021. p. 313-318.

ÖZCAN, Tünay; KOCAK, Semra; BRUNE, Philipp. Agile Software Development with Open Source Software in a Hospital Environment–Case Study of an eCRF-System for Orthopaedical Studies. In: WEB ENGINEERING: 13TH INTERNATIONAL CONFERENCE, ICWE 2013, Aalborg, Dinamarca. Anais... Springer Berlin Heidelberg, 2013. p. 439-451.

BALARAMAN, KOSALRAM, "E –Hospital Management & Hospital Information Systems – Changing Trends", International Journal of Information Engineering and Electronic Business(IJIEEB), vol.5, no.1, pp.50-58, 2013.

PRESSMAN, Roger S. Pressman; MAXIM, Bruce R. Maxim. Engenharia de Software Uma Abordagem Profissional. 8. ed. - Porto Alegre: AMGH, 2016.

SAHA, Tushar et al. Hospital pharmacy management system and future development approaches in Bangladeshi hospital. Bangladesh Pharmaceutical Journal, v. 20, n. 2, p. 180-187, 2017.

SHAW, C. How can hospital performance be measured and monitored? Copenhagen, WHO Regional Office for Europe, 2003. (Health Evidence Network report.)

SOMMERVILLE, Ian. Engenharia de Software. 9. ed. Tradução Kalinka Oliveira; Ivan Bosnic. São Paulo: Persson, 2007.

VIDAL, Carvalho; ROCHA, Abreu. Maturity of hospital information systems: Most important influencing factors. Health Informatics J. 2019 Sep;25(3):617-631.

VOLLE, Adam. Web application. Encyclopedia Britannica, 6 Out. 2022. Disponível em: <https://www.britannica.com/topic/Web-application>. Acesso em: 1 Jul. 2023.