

**CENTRO UNIVERSITÁRIO UNIOPET
ESCOLA DE ARQUITETURA, ENGENHARIA E TECNOLOGIA
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

ARTHUR CAMARGO MACIEL
ELISANE CORDEIRO DE OLIVEIRA LUZ
LUANA LIMA FIGUEIREDO
NICOLLY SPEGIORIN MOREIRA
VITOR SVIDNICKI

Projeto Integrador: Calendário de Remédios

CURITIBA/PR

2023

ARTHUR CAMARGO MACIEL
ELISANE CORDEIRO DE OLIVEIRA LUZ
LUANA LIMA FIGUEIREDO
NICOLLY SPEGIORIN MOREIRA
VITOR SVIDNICKI

Projeto Integrador: Calendário de Remédios

Versão Corrigida

Trabalho PROJETO INTEGRADOR apresentado ao Departamento de análise e desenvolvimento de sistemas e tecnologia do Centro Universitário Uniopet, Câmpus Curitiba, como parte dos requisitos para obtenção do conhecimento e aprovação nas disciplinas participantes do projeto integrador.
Orientador: Prof. Dra. Simone Dominico

CURITIBA/PR

2023

SUMÁRIO

1. INTRODUÇÃO	04
2. OBJETIVOS GERAIS	05
3. OBJETIVOS ESPECIFICOS E REQUISITOS FUNCIONAIS	06
4. REQUISITOS NÃO FUNCIONAIS	08
5. ARQUITETURA DE SOFTWARE	09
6. ARQUITETURA MONOLITICA	10
7. VISÕES DE ARQUITETURA 4+1	12
7.1 Visão em módulo	12
7.2 Visão em tempo de execução	13
7.3 Visão de implantação	13
7.4 Visão de implementação	14
7.5 Visão de dados	14
8.REPRESENTAÇÃO COM DIAGRAMAS	15
9. DIVISÃO E ORGANIZAÇÃO DO PROJETO	18
9.1 Banco de Dados	18
CONTRIBUIÇÃO	22
CONSIDERAÇÕES FINAIS	23
REFERÊNCIAS	24
GLOSSÁRIO	25

1 INTRODUÇÃO

Em uma pesquisa realizada pela IQVIA, empresa líder no uso de informações sobre saúde, revela que dentre os países da América Latina o Brasil está em primeiro lugar no ranking dos países que mais comprem medicamentos. Isto nos revela que maior parte da população brasileira toma algum tipo de remédio. E dentre estas pessoas estão aquelas que precisam tomar medicamentos controlados e diários, sendo muitas delas idosas e muitos debilitados. Visto que nos últimos anos o número de pessoas que tomam remédios controlados e contínuos tem aumentado drasticamente. As pessoas mais idosas e mais debilitadas geralmente precisam da ajuda de familiares e até de enfermeiros para controlar horários e quantidade de cada medicamento, pensando nisto um projeto foi criado para facilitar o acesso destas pessoas ao sistema para manipular as informações sobre cada medicamento a ser tomado.

Tendo em vista que todo cuidado com medicamentos é necessário, pois devem ser tomados nos dias, dosagem e horários corretos para seu total efeito. Projetamos algo que já existe no mercado, mas que procuramos deixar da maneira mais simples e acessível a qualquer pessoa.

É um sistema com lembrete e informações sobre horário, dosagem e dia que cada medicamento deverá ser tomado, facilitando assim a vida de todas as pessoas que precisam em sua rotina diária usar medicamentos diferentes. Trazendo assim muito mais praticidade para quem toma variedade de remédios e para aqueles que cuidam de pessoas com essa necessidade.

A princípio criamos um sistema onde poderá ser cadastrado o paciente, seus medicamentos e o horário acompanhado de alarme que cada remédio deverá ser tomado. O sistema é simples, de fácil acesso e manipulação para facilitar a vida de pessoas que não tem muita convivência com internet e tecnologia.

2 OBJETIVOS GERAIS

A proposta é desenvolver um sistema de controle de medicamento, para ajudar pessoas que tomam vários medicamentos por dia a se lembrar das medicações. Neste sistema deverá conter informações da dosagem, quantas vezes por dia deverá ser tomado, nome dos remédios e os horários de cada um, com lembrete de esquecimento caso o paciente se esqueça de tomar o medicamento.

3 OBJETIVOS ESPECIFICOS E REQUISITOS FUNCIONAIS

Em engenharia de software, um requisito funcional define uma função de um sistema de software ou seu componente. O requisito funcional representa o que o software faz, em termos de tarefas e serviços. Uma função é descrita como um conjunto de entradas, seu comportamento e as saídas. Os requisitos funcionais podem ser cálculos, detalhes técnicos, manipulação de dados e de processamento e outras funcionalidades específicas que definem o que um sistema, idealmente, será capaz de realizar.

Os requisitos funcionais do sistema serão:

- RF 01 - Realizar login;
- RF 02 - Cadastrar usuário;
- RF 03 - Cadastrar profissional de saúde;
- RF 04 - Cadastrar funcionalidades;
- RF 05 - Cadastrar remédios;
- RF 06 - Cadastrar especificações;
- RF 07 - Cadastrar data.
- RF 08 - Cadastrar horário;
- RF 09 - Cadastrar alarme;
- RF 10 – Realizar consulta.

Descrição dos requisitos funcionais:

-RF 01 -Realizar login: Para que o usuário obtenha acesso para cadastrar seus medicamentos ele precisa realizar login.

- RF 02 - Cadastrar usuário: Para que o usuário tenha acesso ao sistema e possa utilizá-lo é necessário que ele realize um cadastro com informações pessoais.

- RF 03 – Cadastrar profissional de saúde: Todos os usuários do sistema necessitam estar cadastrados para obterem acesso as funcionalidades.

- RF 04 – Cadastrar funcionalidades: Todos os usuários podem cadastrar a funcionalidade que desejar no sistema.

- RF 05 - Cadastro de remédios: Para que o sistema consiga avisar os horários, cada medicamento precisa estar cadastrado no sistema.

- RF 06 – Cadastrar especificações: Cada medicamento cadastrado deve estar definido a quantidade que deverá ser tomada por dia, seguido da especificação de local de armazenamento.

- RF 07 - Cadastro de data: O usuário deve definir quais dias o alarme precisa lembrá-lo.

- RF 08 - Cadastro de horário: Para que seja possível o alarme tocar é necessário que o usuário defina o horário para cada medicamento.

- RF 09 - Cadastro de alarme: para que o alarme avise no momento adequado, o usuário deve definir dias e horas que cada medicamento deve ser tomado, seguido de lembrete caso perca o primeiro alarme.

- RF 10 – Realizar consulta: Cada usuário cadastrado poderá ser visualizado juntamente com seus respectivos medicamentos, horários e dias.

4 REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais são aqueles que não estão diretamente ligados as funcionalidades do sistema, mas com os atributos que este sistema irá possuir, ele diz respeito a forma como o sistema deverá agir para que os requisitos funcionais consigam atingir seus objetivos.

São uma parte importante do desenvolvimento de um sistema, eles ajudam a garantir que o sistema atenda às necessidades do usuário. Assim como os requisitos funcionais definem o que o sistema precisa fazer, os requisitos não funcionais definem como o sistema fará, tratando as questões vitais de qualidade do sistema.

Os requisitos funcionais escolhidos para este projeto especificamente foram estes listados abaixo:

- RNF 01 - Disponibilidade: O sistema deverá estar disponível quando necessário;
- RNF 02 – Manutenção: O sistema deverá ser fácil de manter e atualizar;
- RNF 03 – Confiabilidade: O sistema deverá ser confiável e atender aos requisitos do usuário.
- RNF 04 – Usabilidade: O sistema deverá ser fácil de usar e entender.

5 ARQUITETURA DE SOFTWARE

Todo projeto de software precisa ser modelado através da arquitetura de software, para que todos possam compreender de forma clara como tudo funcionará. Esta consiste na definição dos componentes de software, suas propriedades externas, e seus relacionamentos com outros softwares. O termo também se refere à documentação da arquitetura de software do sistema. A documentação da arquitetura do software facilita: a comunicação entre os stakeholders, registra as decisões iniciais acerca do projeto de alto-nível, e permite o reuso do projeto dos componentes e padrões entre projetos.

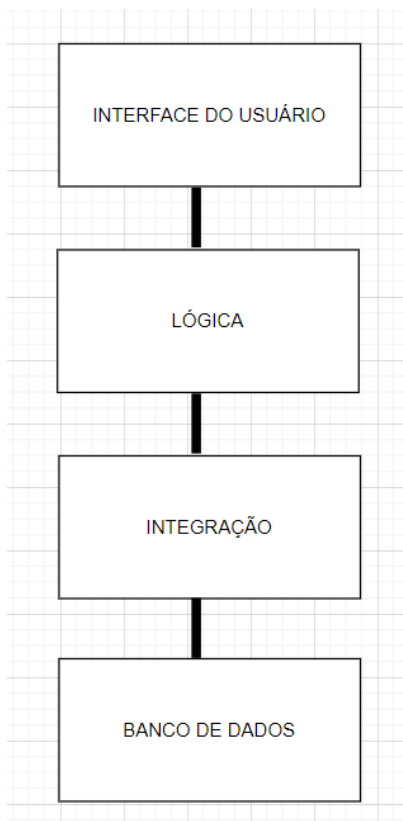
6 ARQUITETURA MONOLITICA

Existem vários tipos de arquiteturas de software, cada uma é escolhida de acordo com a necessidade do sistema. Vários fatores devem ser levados em conta na hora de escolher o tipo de arquitetura que o projeto vai implementar, como a escalabilidade por exemplo que é um grande fator a analisar no momento da decisão, pois se o projeto prevê um grande crescimento ele necessitará de uma arquitetura que suporte este crescimento.

Neste contexto aplicaremos neste projeto a arquitetura Monolítica que é um sistema único, não dividido, que roda em um único processo, uma aplicação de software em que diferentes componentes estão ligados a um único programa dentro de uma única plataforma.

Este foi o estilo de arquitetura escolhida para este projeto uma vez que ele não necessitará de grande escalabilidade pois é um sistema simples e pequeno. Este tipo de arquitetura pode ser mais fácil de testar, já que todos os componentes estão todos no mesmo código.

Outro motivo pelo qual este tipo de arquitetura foi escolhido, seria porque uma arquitetura monolítica é muito mais fácil de implementar e gerenciar, uma vez que ela roda em um único processo.



(Diagrama de representação visual da arquitetura monolítica do projeto calendário de remédio)

Como demonstra este diagrama, basicamente o projeto ficou representado em quatro camadas que fazem parte do mesmo processo, são eles:

- Interface do usuário: Esta camada permite ao usuário interagir com o sistema onde ele poderá agendar, editar ou cancelar o alarme de remédio.

- Lógica: Esta camada será responsável pela validação e processamento dos dados enviados pelo usuário, também pelo cálculo dos horários dos alarmes, armazena cada informação dos alarmes no banco de dados e será responsável por enviar as notificações dos alarmes programados para o usuário.

- Integração: Esta camada recebe uma API que servirá para enviar os alarmes aos dispositivos móveis dos usuários.

- Banco de dados: Esta camada será responsável pelo armazenamento de todas as informações sobre os medicamentos e seus respectivos alarmes.

7 VISÕES DE ARQUITETURA 4+1

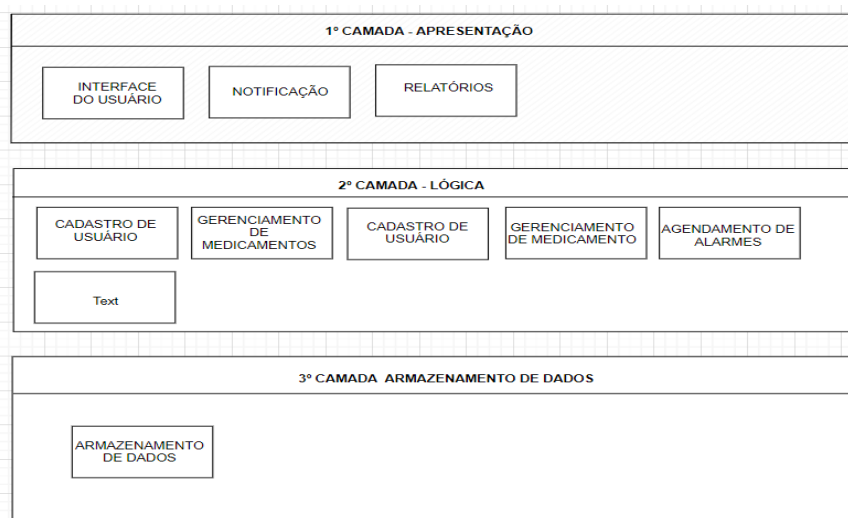
A arquitetura visão modelo 4+1 foi desenvolvida por Philippe Kruchten, 1995, com o objetivo de descrever o funcionamento de sistemas de software e é baseado no uso de múltiplas visões concorrentes.

As visões são usadas para mostrar o sistema sob várias perspectivas como final, desenvolvedores e gerentes de projetos.

7.1 Visão em módulos

Apresenta a estrutura do sistema em termos de unidade de implementação, pode ser representada por caixas e linhas, textos ou tabelas e ainda por diagrama de classe da UML.

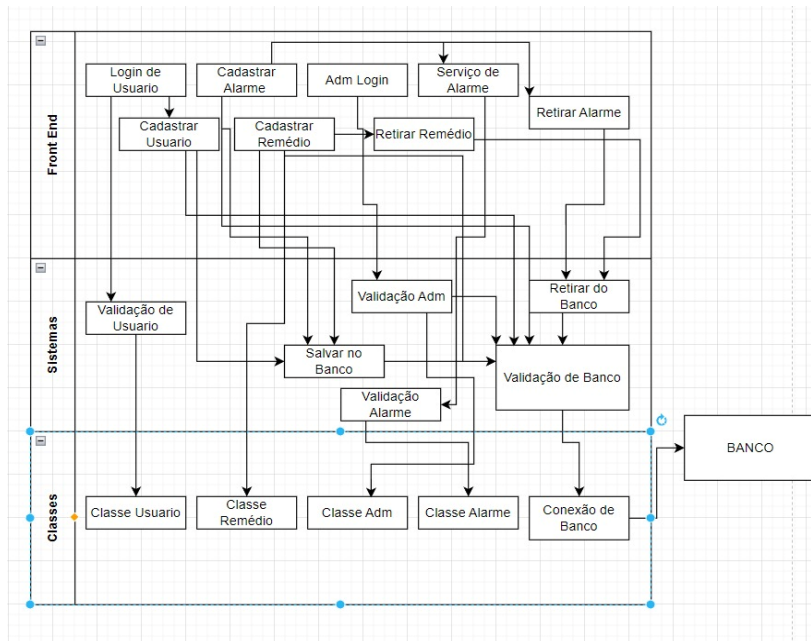
Para representar esta visão usaremos visão por camadas, assim como está representado abaixo.



(Representação em módulos do projeto calendário de remédio)

7.2 Visão em tempo de execução

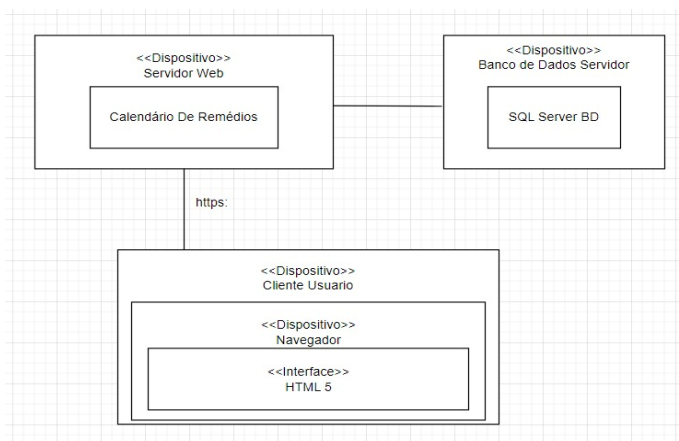
Esta visão mostra o sistema em tempo de execução, possibilita o entendimento do sistema e análise sobre o que está sendo executado. Geralmente representado através de caixas e linhas ou digrama de componentes.



(Representação: Visão em tempo de execução)

7.3 Visão de implantação

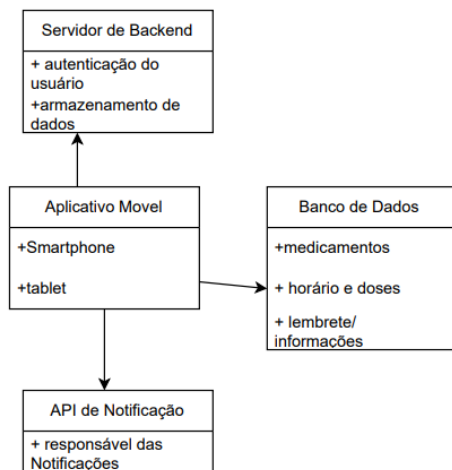
Mostra a estrutura de hardware onde o sistema será executado. Geralmente representado através de diagrama de redes ou diagrama de implantação.



(Visão de implantação)

7.4 Visão de implementação

Mostra a estrutura do software em termos de arquivos organizados em diretórios. Este pode ser representado usando caixas e linhas e arvores, além de UML 2.0.

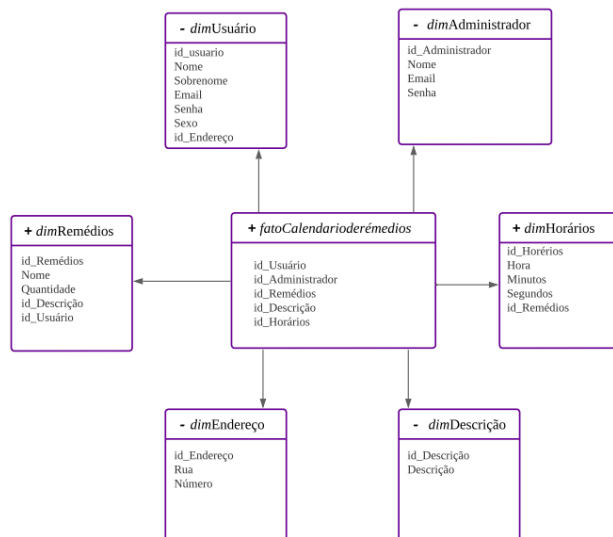


(Representação da visão de implementação)

7.5 Visão de dados

Usada sempre que uma estrutura de banco de dados precisa ser modelada, inicia-se com o modelo lógico. Esta visão será representada através de diagrama de classe UML.

Visão de Dados:



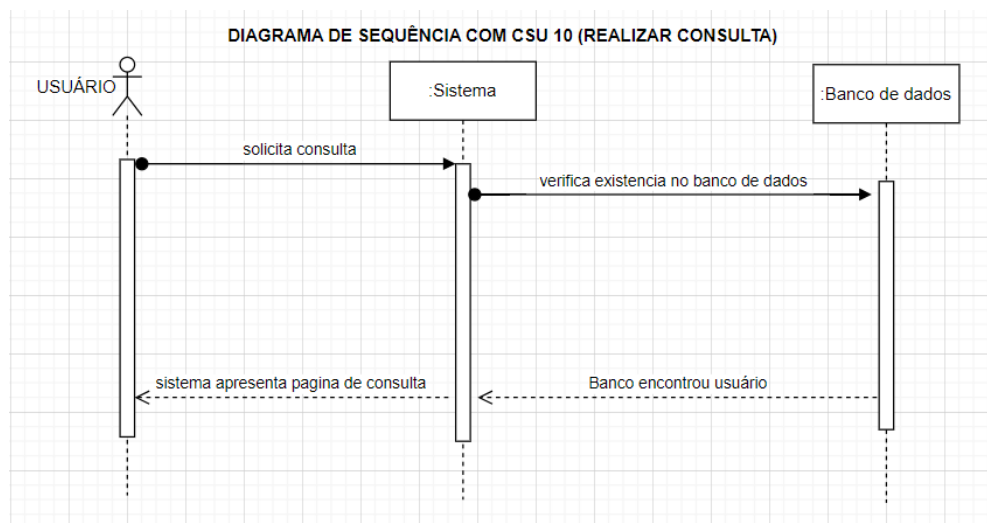
(Visão de dados com diagrama classe)

8 REPRESENTAÇÕES COM DIAGRAMAS

A Linguagem de modelagem unificada (UML) foi criada para estabelecer uma linguagem de modelagem visual comum, semanticamente e sintaticamente rica, para arquitetura, design e implementação de sistemas de software complexos, tanto estruturalmente quanto para comportamentos. A UML também poderia ser compreendida por usuários do mundo dos negócios e qualquer pessoa que queira entender mais sobre um sistema.

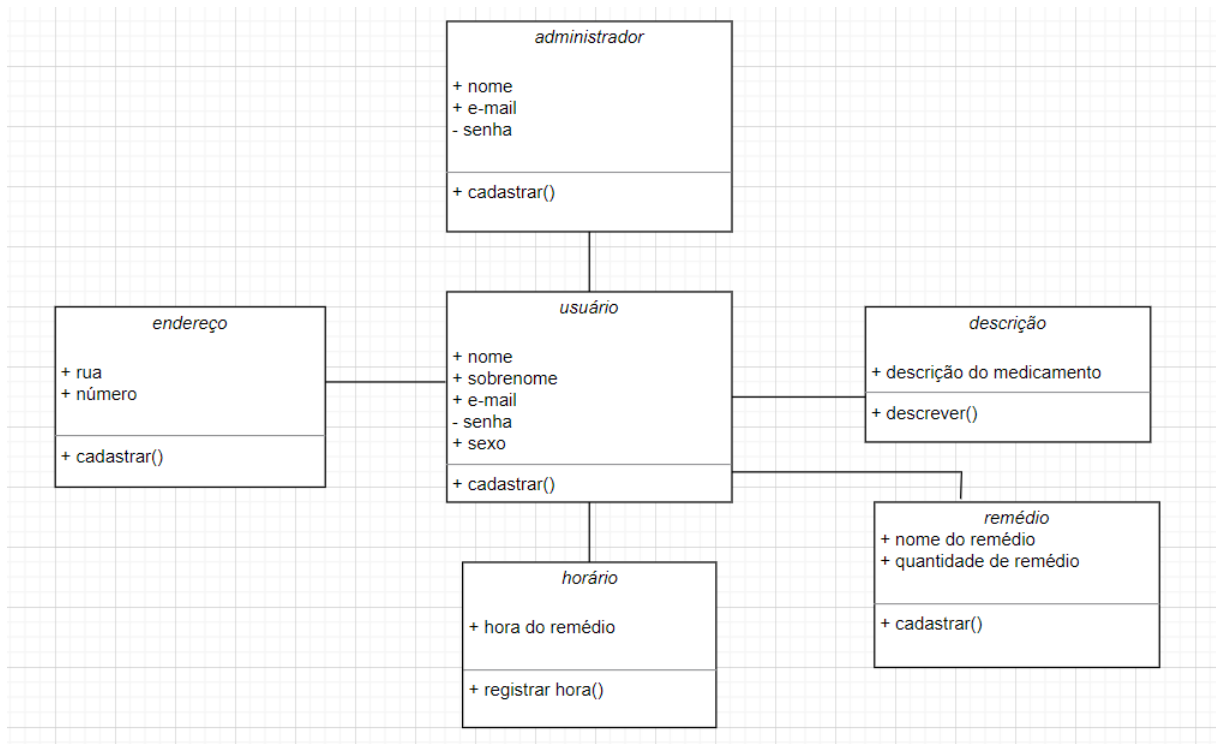
A UML não é uma linguagem de programação, mas existem ferramentas que podem ser usadas para gerar código em várias linguagens por meio de diagramas UML. A UML tem uma relação direta com a análise e o design orientados a objetos.

Diagrama de sequência: Este diagrama mostra o fluxo de eventos de caso de uso de um sistema. O diagrama que vamos representar mostra o fluxo do caso de uso número 10 deste projeto (Realizar consulta).



(Diagrama de Sequência)

Diagrama de classe: Bastante usado por engenheiros de software para documentar arquiteturas de software, os diagramas de classes são um tipo de diagrama da estrutura porque descrevem o que deve estar presente no sistema a ser modelado. Cada tabela criada no banco de dados deve ser descrita como uma classe e precisa estar acompanhada de seus atributos. Todas as classes estão conectadas em um relacionamento de associação.

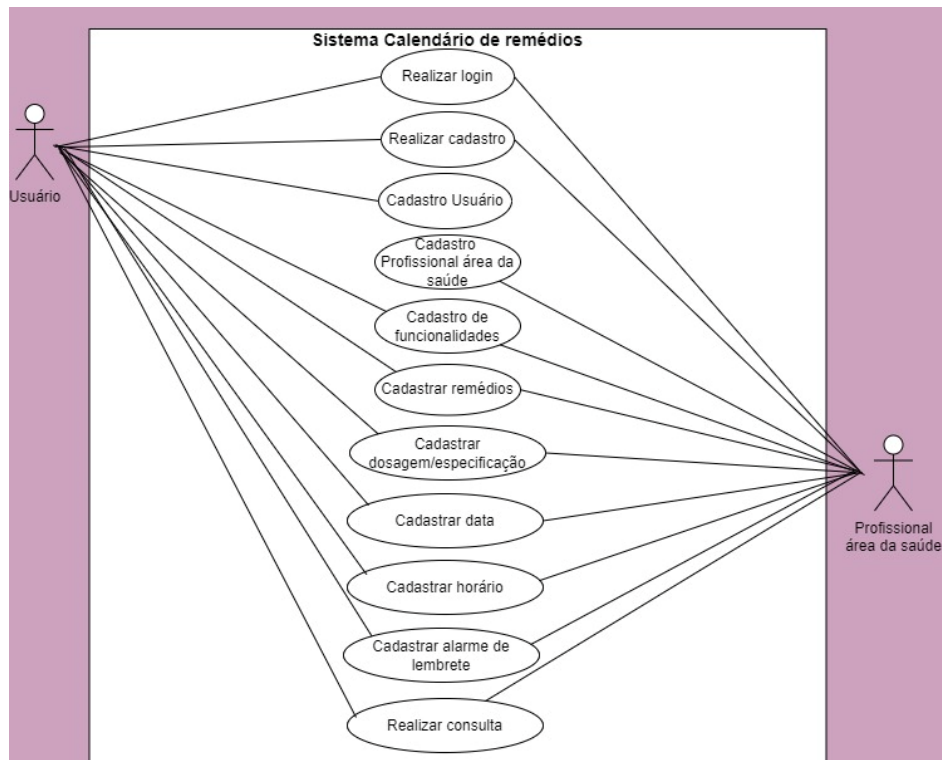


(Diagrama de Classe)

Diagrama de caso de uso: O diagrama de caso de uso descreve a funcionalidade proposta para um novo sistema que será projetado, é uma excelente ferramenta para o levantamento dos requisitos funcionais do sistema. Um caso de uso descreve a sequência de eventos de um ator que usa um sistema para completar um processo. Representa uma unidade discreta da interação entre um ator (humano, dispositivo ou outro software) e o sistema.

Descrição dos casos usos utilizados:

- CSU 01- realizar login;
- CSU 02- cadastrar usuário;
- CSU 03- cadastrar profissional da saúde;
- CSU 04- cadastrar funcionalidades;
- CSU 05- cadastrar remédios;
- CSU 06- cadastrar especificações;
- CSU 07- cadastrar data;
- CSU 08 – cadastrar horário;
- CSU 09 – cadastrar alarme de lembrete;
- CSU 10 – realizar consulta.



(Diagrama de caso de uso)

Deste diagrama foi escolhido um caso de uso representar o fluxo de evento, o processo de interação entre o usuário e o sistema. O caso de uso escolhido foi o CSU 10 (realizar consulta).

FLUXO DE EVENTO CSU 10 (REALIZAR CONSULTA)	
SUMÁRIO:	Usuário consulta seu cadastro e os medicamentos cadastrados em seu nome
ATOR PRIMÁRIO:	Usuário
ATOR SECUNDÁRIO:	Sistema
PRÉ CONDIÇÕES:	Para consultar seu cadastro o usuário precisa estar cadastrado previamente no sistema e estar logado.
FLUXO PRINCIPAL:	
1. Usuário solicita consulta de cadastro.	
2. Sistema verifica se o usuário está cadastrado no banco de dados.	
3. Após receber retorno do banco de dados o sistema apresenta ao usuário a página de consulta de cadastro..	
REGRAS DE NEGÓCIO:	N/A

9 DIVISÃO E ORGANIZAÇÃO DO PROJETO

Este projeto contará com algumas tecnologias que irão formar e interligar cada parte do sistema, assim sendo:

A Home Page, usando a base do HTML e o CSS, optamos por utilizar um framework chamado Bootstrap para facilitar no desenvolvimento do projeto, a partir dele podemos ter um esqueleto pré-definido, o que torna muito mais prática a aplicação.

A próxima parte é o banco de dados, ele será usado para registrar os usuários, os remédios, os horários, as datas, local armazenado e o alarme.

Haverá na parte dos remédios onde será colocado o tipo de remédio e a dosagem certa pra cada pessoa cadastrada, uma parte extra das descrições do remédio, onde deverá ser descrito a cor bem como onde estão localizados cada um deles.

O sistema contará com páginas de login e formulário de cadastro, onde haverá um link na página que direciona o usuário a página de cadastro, ali poderão ser colocadas todas as informações pessoais solicitadas. Caso já tenha efetuado o cadastro ela deverá clicar no link login, então será direcionado para tela de acesso ao sistema, na página de login ela deverá acessar com seu e-mail e senha criados no cadastro.

Haverá também a parte do perfil do usuário, onde poderá ser adicionado foto de sua preferência, nome, remédios, datas e os devidos horários estipulados.

9.1 Banco de Dados

Um banco de dados é uma coleção organizada de informações ou dados estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (DBMS). Juntos, os dados e o DBMS, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados.

Os dados nos tipos mais comuns de bancos de dados em operação atualmente são modelados em linhas e colunas em uma série de tabelas para tornar o processamento e a consulta de dados eficientes. Os dados podem ser facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados. A maioria dos bancos de dados usa a linguagem de consulta estruturada (SQL) para escrever e consultar dados.

Neste projeto o banco de dados será utilizado para armazenar informações do paciente, dos medicamentos e o horário que cada medicamento deverá ser tomado, contando ainda com o alarme que avisa cada horário cadastrado a hora certa de tomar o medicamento.

Banco de dados e suas aplicações no projeto

O banco de dados será utilizado para armazenar informações do paciente, dos medicamentos e o horário que cada medicamento deverá ser tomado, contando ainda com o alarme que avisa cada horário cadastrado a hora certa de tomar o medicamento.

Scripts do banco de dados utilizado no projeto:

Script do cadastro de remédio

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Ação
<input type="checkbox"/>	1 id_rmd	int(11)			Não	Nenhum		AUTO_INCREMENT	Muda Eliminar Mais
<input type="checkbox"/>	2 nome_rmd	varchar(50)	utf8mb4_general_ci		Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	3 quantidade_rmd	varchar(50)	utf8mb4_general_ci		Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	4 id_desc	int(11)			Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	5 id_usuario	int(11)			Não	Nenhum			Muda Eliminar Mais

Script do cadastro de horas

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Ação
<input type="checkbox"/>	1 id_hr	int(11)			Não	Nenhum		AUTO_INCREMENT	Muda Eliminar Mais
<input type="checkbox"/>	2 hora_hr	time(4)			Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	3 id_rmd	int(11)			Não	Nenhum			Muda Eliminar Mais

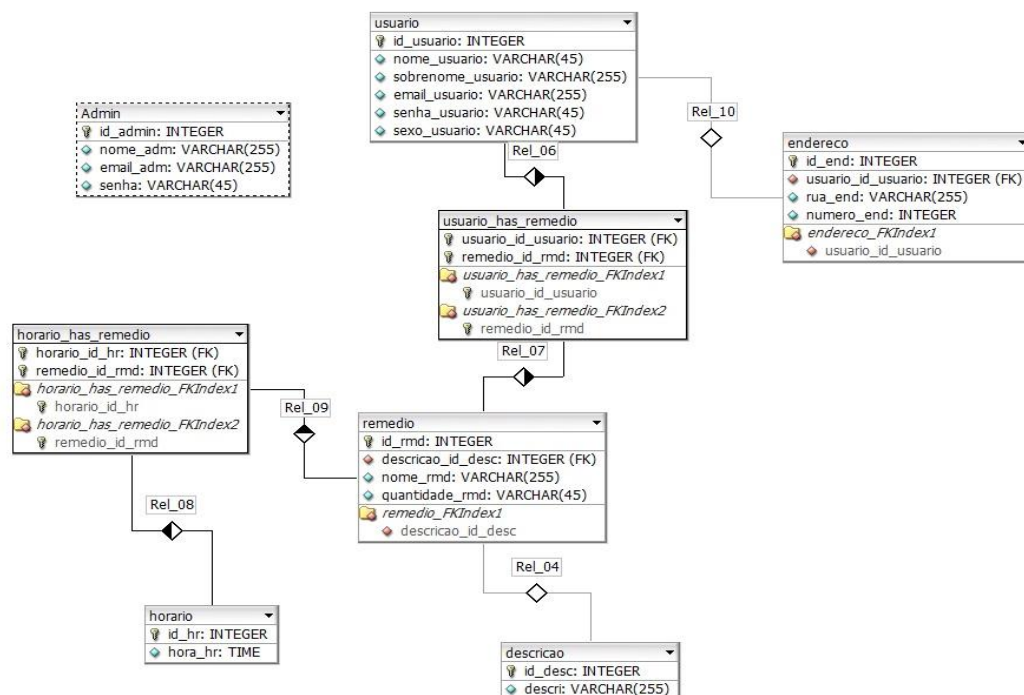
(Script do cadastro do usuário)

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Ação
<input type="checkbox"/>	1 id_usuario	int(11)			Não	Nenhum		AUTO_INCREMENT	Muda Eliminar Mais
<input type="checkbox"/>	2 nome_usuario	varchar(30)	utf8mb4_general_ci		Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	3 sobrenome_usuario	varchar(50)	utf8mb4_general_ci		Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	4 email_usuario	varchar(100)	utf8mb4_general_ci		Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	5 senha_usuario	varchar(50)	utf8mb4_general_ci		Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	6 sexo_usuario	varchar(12)	utf8mb4_general_ci		Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	7 id_end	int(11)			Não	Nenhum			Muda Eliminar Mais

(Banco de dados modelo lógico)

O modelo lógico descreve como os dados serão armazenados no banco e seus relacionamentos. Ele é abstração no nível do usuário do SGBD (sistema gerenciador de banco de dados).

Segue abaixo modelo lógico do projeto calendário de remédios.



PHP

O PHP (um agrônomo recursivo para PHP: Hypertext Preprocessor) é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML.

Originalmente se chamava Personal Home Page (Página Inicial Pessoal).

A princípio todo projeto realizado no primeiro semestre está utilizando a linguagem PHP em toda sua estrutura, desde as validações de senhas e logins bem como abertura de sessão e conexão dos arquivos com o banco de dados.

HTML

HTML é uma Linguagem de Marcação de Hipertexto, é uma linguagem de computador que compõe a maior parte das páginas da internet e dos aplicativos online. É composta por uma série de marcações que dizem para os servidores da web qual é o estilo e a estrutura de um documento.

O HTML não é considerado uma linguagem de programação, já que ele não pode criar funcionalidades dinâmicas. Ao invés disso, com o HTML, os usuários podem criar e estruturar seções, parágrafos e links usando elementos, tags e atributos.

Neste contexto do projeto ele foi utilizado para criar formulários de cadastro.

CSS

O CSS é uma sigla para Cascading Style Sheets, ou em uma tradução para português, Folhas de Estilo em Cascatas. Ele serve para dar estilos ao código criado em HTML, XHTML ou XML. De uma maneira mais prática, o que o CSS faz é a personalização e estilização do conteúdo visível pelo usuário.

Foi utilizado no projeto inicialmente nos formulários de HTML na customização das cores, fontes e tamanhos do formulário.

CONTRIBUIÇÃO

Este projeto foi criado baseado no trabalho de extensão que foi realizado no segundo semestre de 2022. Onde discutimos a respeito dos vários tipos de doenças, vacinas e como a Brasil lida com estas situações. Neste contexto surgiu a ideia de criarmos um sistema para facilitar a vida destas pessoas que possuem a necessidade de tomar muitos medicamentos por dia, um software que controle a quantidade e o medicamento correto para se tomar em cada dia.

O Brasil está entre os países que mais usam medicamentos na América Latina, e muitas das pessoas precisam tomar variedade de remédios todos os dias, estas pessoas encontram dificuldade na hora de controlar a quantidade de remédios a serem tomados por dia. Pensando em contribuir para facilitar a vida destas pessoas desenvolvemos um projeto capaz de ajudar no controle do uso de medicamentos.

Já possuem alguns sistemas parecidos no mercado, mas queremos oferecer algo que possa ser mais simples e acessível mesmo que o usuário seja leigo ele possa ter facilidade para acessar, desta forma eles serão beneficiados pela praticidade que o sistema oferece.

CONSIDERAÇÕES FINAIS

Desde que iniciado este projeto no segundo semestre de 2022, alguns passos já foram dados na execução dele. O banco de dados já foi criado juntamente com login e formulários de cadastro, todos já estão funcionando, necessitando ainda de alguns ajustes.

Neste próximo semestre, daremos continuidade ao projeto, adicionando seus requisitos funcionais e requisitos não funcionais, a documentação de requisitos, será adicionado também o tipo de arquitetura de software que será utilizado no projeto.

Trabalhar em equipe sempre é um grande desafio, pois cada um possui uma forma diferente de trabalhar, mas no geral podemos concluir que desenvolvemos várias habilidades, como a capacidade de se comunicar de forma clara e eficaz com a equipe, capacidade de organizar tarefas, recursos e cronogramas para garantir o cumprimento dos prazos estabelecidos, capacidade de entender as tecnologias e ferramentas necessárias para o desenvolvimento do projeto.

O maior desafio encontrado foi em relação a conciliação do tempo, pois devido a demanda de outras atividades acadêmicas, ficamos com pouco tempo para atuarmos juntos com a equipe para compartilharmos de forma efetiva todas as partes do projeto.

Ainda podemos citar como desafio, o fato de não conhecermos algumas tecnologias a fundo, a ponto de dominar cada uma delas, precisando muitas vezes pesquisar por conta como cada uma funciona, não obtendo muitas vezes êxito total no entendimento delas.

REFERÊNCIAS

BALLERINI, Rafaella. **HTML, CSS e Javascript, quais as diferenças?** 2023. Alura. Disponível em: <https://www.alura.com.br/artigos/html-css-e-js-definicoes>. Acesso em: 20 mar. 2023. www.basef.com.br/old/uml/204-arquitetura-visao-modelo-41

COUTINHO, Thiago. Aprenda como o CSS consegue melhorar a qualidade do seu website. **2022. Voitto.** Disponível em: <https://www.voitto.com.br/blog/artigo/css>. Acesso em: 25 abr. 2023.

COWBURN, Peter. **O que é o PHP?** 2023. Php. Disponível em: https://www.php.net/manual/pt_BR/index.php. Acesso em: 20 mar. 2023.

L., Andrei. **HTML abr 12, 2023 Andrei L. 7min de leitura O Que é HTML? Guia Básico Para Iniciantes.** 2023. Hostinger. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos>. Acesso em: 25 mar. 2023. www.lucidchart.com/pages/pt/o-que-e-uml

WIKIPÉDIA. **Arquitetura de software.** 2020. Wikipedia. Disponível em: https://pt.wikipedia.org/wiki/Arquitetura_de_software. Acesso em: 20 abr. 2023.

WIKIPÉDIA. Diagrama de caso de uso. **2020. Wikipédia.** Disponível em: https://pt.wikipedia.org/wiki/Diagrama_de_caso_de_uso. Acesso em: 15 abr. 2023.

WIKIPÉDIA. **Requisito funcional.** 2023. Wikipédia. Disponível em: https://pt.wikipedia.org/wiki/Requisito_funcional. Acesso em: 20 abr. 2023.

GLOSSÁRIO

CSU: Caso de uso.

HTML: Linguagem de marcação de hipertexto.

PHP: Pré-Processador de hipertexto, é uma linguagem de programação.

RF: Requisito funcional.

RNF: Requisito não funcional.

UML: Linguagem de modelagem unificada.