



INSTITUTO MÉDIO POLITÉCNICO SMARTBITS
COORDENAÇÃO DO CURSO DE INFORMÁTICA
PROVA DE APTIDÃO PROFISSIONAL (PAP)

ALINE KAQUEMA CÂMIA CHIENG

EMANUEL NURIO DE OLIVEIRA CALUNGA

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA O
GERENCIAMENTO DA FARMÁCIA EDACAMBANGO**

LUANDA

2023/2024

ALINE KAQUEMA CÂMIA CHIENGO

EMANUEL NURIO DE OLIVEIRA CALUNGA

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA O
GERENCIAMENTO DA FARMÁCIA EDACAMBANGO**

Trabalho de conclusão de curso apresentado no
Instituto Médio Politécnico Smartbits como
exigência parcial para a obtenção de grau de
Técnico Médio de Informática.

ORIENTADOR: Vidário Antônio.

LUANDA

2023/2024

ERRATA

FOLHA DE APROVAÇÃO

Este projeto foi apresentado em ____/____/2023 para a obtenção do título de técnico médio de informática.

Banca examinadora

Presidente do júri: Assinatura.....

1º vogal: Assinatura.....

2º vogal: Assinatura.....

Aprovado: ____/____/_____

DEDICATÓRIA

Dedicamos este trabalho de abordagem científica
a Deus por nos ter dado vida e saúde a todos
integrantes do grupo.

AGRADECIMENTOS

Ao nosso professor pela orientação do trabalho com muita atenção, disponibilidade, dedicação, perícia metodologia e científica e por todas as oportunidades que nos tem proporcionado o nosso muito e profundo voto de gratidão.

Aos nossos colegas que por mais difíceis que os problemas forem neste grande desafio conseguimos romper barreiras;

Muito obrigada a todos que direita ou indiretamente tornaram possível á realização deste projeto.

RESUMO

Os Sistemas de Gerenciamento são ferramentas desenvolvidas para auxiliar na organização e controle de informações, processos e recursos em diferentes contextos. No caso das farmácias, esses sistemas são utilizados para gerenciar inventário, vendas, receitas médicas e interações com os clientes, visando aumentar a eficiência e melhorar a qualidade dos serviços prestados. A evolução da web remonta ao final do século XX, quando surgiram os primeiros sites estáticos, compostos principalmente por texto e imagens. Com o tempo, a web evoluiu para uma plataforma dinâmica e interativa, permitindo a criação de aplicativos web complexos e sofisticados. Esse progresso foi impulsionado pelo desenvolvimento de tecnologias como HTML, CSS, JavaScript e frameworks de desenvolvimento web. Uma aplicação web é um software projetado para ser acessado e executado em um navegador web. Ao contrário dos aplicativos tradicionais, as aplicações web não precisam ser instaladas no dispositivo do usuário, pois são executadas remotamente em um servidor e acessadas por meio de uma interface web. As aplicações web podem ser classificadas de várias maneiras, dependendo de critérios como complexidade, funcionalidade e arquitetura. Algumas categorias comuns incluem, aplicativos web dinâmicos, aplicativos web estáticos, lojas online, e animados. Uma aplicação server-side é executada em um servidor web, onde o processamento das solicitações do cliente ocorre do lado do servidor. Isso significa que as operações são realizadas no servidor, que interage com um banco de dados para recuperar informações e gerar conteúdo dinâmico para o cliente. As linguagens comuns para o desenvolvimento server-side incluem, HTML, PHP, e JAVA SCRIPT. Um modelo de base de dados é uma representação visual das relações entre os diferentes elementos de um banco de dados. Ele define a estrutura e organização dos dados, incluindo tabelas, campos, chaves primárias e chaves estrangeiras. O modelo relacional é um tipo de modelo de base de dados que organiza os dados em tabelas com linhas e colunas. Ele utiliza chaves primárias e chaves estrangeiras para estabelecer relações entre as tabelas, permitindo consultas e operações complexas de forma eficiente.

Palavras-chave: Sistemas, Web, Sever-side, Aplicações.

ABSTRAT

Management Systems are tools developed to assist in the organization and control of information, processes and resources in different contexts. In the case of pharmacies, these systems are used to manage inventory, sales, prescriptions and interactions with customers, aiming to increase efficiency and improve the quality of services provided. The evolution of the web dates back to the end of the 20th century, when the first static websites appeared, composed mainly of text and images. Over time, the web has evolved into a dynamic and interactive platform, allowing the creation of complex and sophisticated web applications. This progress has been driven by the development of technologies such as HTML, CSS, JavaScript and web development frameworks. A web application is software designed to be accessed and run in a web browser. Unlike traditional applications, web applications do not need to be installed on the user's device, as they are run remotely on a server and accessed through a web interface. Web applications can be classified in several ways, depending on criteria such as complexity, functionality and architecture. Some common categories include, dynamic web apps, static web apps, online stores, and animated. A server-side application runs on a web server, where the processing of client requests takes place on the server side. This means that operations are performed on the server, which interacts with a database to retrieve information and generate dynamic content for the client. Common languages for server-side development include HTML, PHP, and JAVA SCRIPT. A database model is a visual representation of the relationships between the different elements of a database. It defines the structure and organization of data, including tables, fields, primary keys, and foreign keys. The relational model is a type of database model that organizes data into tables with rows and columns. It uses primary keys and foreign keys to establish relationships between tables, allowing complex queries and operations efficiently.

Keywords: Systems, Web, Sever-side, Applications.

LISTA DE ILUSTRAÇÕES

Figura 1:DIAGRAMA DE CASO DE USO.....	21
Figura 2:Diagrama de atividade fazer login.....	25
Figura 3:Diagrama de atividade editar perfil.....	26
Figura 4:Diagrama de atividade alterar senha	27

LISTA DE TABELAS

Tabela 1:Ferramentas e tecnologias utilizadas	18
Tabela 2:Requisitos funcionais	19
Tabela 3:Documentação do caso de uso fazer login	22
Tabela 4:Documentação de caso de uso editar perfil	23
Tabela 5:Documentação de caso de uso alterar senha.....	24

LISTA DE ABREVIATURAS E SIGLAS

HTML	Hyper Text Markup Language
AJAX	Asynchronous JavaScript and XML
ASPs	Application Service Providers
WebApps	Aplicações Web
DNS	Domain Name Server
URL	Uniform Resource Locator
API	Application Programming Interfaces
RUP	Rational Unified Process

INDÍCE

1	INTRODUÇÃO	1
1.1	SITUAÇÃO PROBLEMÁTICA	2
1.2	PROBLEMA DE INVESTIGAÇÃO.....	2
1.3	HIPÓTESE.....	2
1.4	JUSTIFICATIVA	2
1.5	MOTIVAÇÃO DO PROJETO.....	3
1.6	CAMPO DE ACÇÃO	3
1.7	OBJECTIVOS.....	3
1.7.1	Objetivo Geral	3
1.7.2	Objetivos Específicos.....	3
1.8	METODOLOGIA.....	4
1.8.1	TIPO DE PESQUISA.....	4
2	FUNDAMENTAÇÃO TEÓRICA.....	5
2.1	O SURGIMENTO DOS PRIMEIROS SISTEMAS WEB	5
2.2	APLICACÃO WEB.....	6
2.3	TIPOS DE APLICAÇOES WEB	6
2.3.1	Aplicativos web dinâmicos	7
2.3.2	Aplicativos web estáticos	7
2.4	CARACTERÍSTICAS DAS APLICAÇÕES WEB.....	7
2.5	COMO FUNCIONA UMA APLICAÇÃO WEB?	8
2.5.1	Funcionamento das aplicações sever-side.....	9
2.5.2	Integração com client-side	9
2.6	PROCESSO DE DESENVOLVIMENTO DE SOFTWARE.....	10
2.7	FRAMEWORK.....	12
2.8	ARQUITETURA DE SOFTWARE.....	12
2.8.1	Arquitetura de desenvolvimento de software m.v.c	13
2.9	MODELO DE BANCO DE DADO	14
2.9.1	Modelo relacional	14
2.10	O CONCEITO DE FARMÁCIA.....	15

2.10.1	Vantagens da Aplicação Web no Gerenciamento de Farmácia	15
2.10.2	Desvantagens da Aplicação Web no Gerenciamento de Farmácia.....	16
2.10.3	O Impacto de um Sistema de Gerenciamento nas Farmácias	16
2.11	FERRAMENTAS UTILIZADAS	18
2.12	REQUISITOS FUNCIONAIS.....	19
2.13	REQUISITOS NÃO FUNCIONAIS	20
2.14	DIAGRAMAS	21
2.14.1	DIAGRAMA DE CASO DE USO	21
2.14.2	DIAGRAMA DE ACTIVIDADE	25
CONCLUSÃO	1
REFERÊNCIAS BIBLIOGRÁFICAS	2

1 INTRODUÇÃO

Um sistema de gerenciamento web para uma farmácia é uma aplicação baseada na web projetada para facilitar e otimizar várias operações e processos dentro de uma farmácia. Ele oferece uma plataforma centralizada e acessível que permite aos farmacêuticos e outros funcionários gerenciar eficientemente uma ampla gama de atividades, desde a gestão de estoque até a interação com os clientes.

Esses sistemas são projetados para serem acessados através de um navegador da web, o que significa que podem ser utilizados em qualquer dispositivo com acesso à internet, como computadores, tablets ou smartphones. Isso proporciona flexibilidade e conveniência aos usuários, permitindo-lhes acessar o sistema a qualquer momento e de qualquer lugar.

Nesta jornada, exploraremos os passos cruciais envolvidos no desenvolvimento de uma aplicação web para o gerenciamento de farmácias, destacando como essa abordagem pode transformar e aprimorar significativamente os processos farmacêuticos.

1.1 Situação problemática

Em um mercado farmacêutico em constante movimento, onde a saúde e o bem-estar se entrelaçam em cada interação, a farmácia "EDACAMBANGO" enfrenta uma situação problemática significativa. Por não possuir um sistema de gerenciamento, a farmácia enfrenta diversos desafios. O estoque de medicamentos está constantemente em fluxo, com produtos chegando e saindo em um ritmo frenético, sobrecarregando os funcionários e tornando difícil acompanhar as demandas dos clientes. Esta falta de eficiência no gerenciamento também dificulta a identificação de padrões de compra dos clientes, resultando na oferta limitada de serviços personalizados. Sem uma visão clara do estoque e das vendas, a farmácia enfrenta desperdícios de produtos, falta de itens essenciais e perda de oportunidades de vendas.

1.2 Problema de investigação

Como otimizar a eficiência e a segurança no gerenciamento de estoque de uma farmácia por meio de um sistema web, considerando a rastreabilidade de medicamentos, a gestão de datas de validade, e a integração eficaz com fornecedores e clientes?

1.3 Hipótese

H1: A implementação de um sistema web que permite a rastreabilidade de medicamentos pode reduzir significativamente o risco de erros no gerenciamento de estoque de uma farmácia.

H2: A integração efetiva do sistema web com fornecedores e clientes pode aumentar a eficiência na reposição de estoque, reduzindo os períodos de escassez de medicamentos.

H3: A utilização de alertas automáticos para gestão de datas de validade no sistema web pode minimizar perdas financeiras decorrentes da expiração de medicamentos e melhorar a conformidade com as regulamentações sanitárias.

1.4 Justificativa

Nos últimos anos, o avanço da tecnologia tem transformado profundamente diversos setores, e a área da saúde não é exceção. As farmácias desempenham um papel crucial na prestação de serviços de saúde à comunidade, fornecendo medicamentos e orientações importantes aos pacientes. A implementação de um sistema web para uma farmácia oferece uma oportunidade valiosa para reduzir esses desafios e aprimorar as operações internas, bem como a qualidade dos serviços prestados aos pacientes. Pois um sistema web para uma farmácia

não apenas moderniza suas operações, mas também a capacita para atender às demandas em constante evolução do mercado de saúde. Ao integrar tecnologia moderna às práticas tradicionais de farmácia, é possível melhorar significativamente a eficiência, precisão e segurança no fornecimento de medicamentos, a adoção de um sistema web para uma farmácia a capacita para enfrentar futuras demandas, tornando-a um componente vital e inovador na prestação de serviços de saúde à comunidade.

1.5 Motivação do projeto

A motivação para desenvolver este projeto surgiu em função do relato das necessidades da farmácia Edacambango. Desta forma, nasceu a expectativa de desenvolver uma aplicação web que atendesse às necessidades desta mesma farmácia.

1.6 Campo de acção

O projeto será implementado na Farmácia Edacambango localizado na rua direita da cinquentinha.

1.7 OBJECTIVOS

1.7.1 Objetivo Geral

Desenvolver uma aplicação web para gerenciamento da Farmácia Edacambango.

1.7.2 Objetivos Específicos

- Definir as funcionalidades do sistema de gerenciamento;
- Definir a arquitetura de desenvolvimento de software;
- Coletar os dados necessários para modelar a base de dados;
- Modelar a base de dados;
- Fazer a interface do sistema de gerenciamento;
- Implementar o sistema de gerenciamento na farmácia;

1.8 Metodologia

Para Gil (1999), o método científico é um conjunto de procedimentos intelectuais e técnicos utilizados para atingir o conhecimento. Para que seja considerado conhecimento científico, é necessária a identificação dos passos para a sua verificação, ou seja, determinar o método que possibilitou chegar ao conhecimento.

1.8.1 Tipo de pesquisa

Pesquisa é o procedimento racional e sistemático que tem como objectivo favorecer respostas aos problemas que são propostos (GIL, 2017). É desenvolvida mediante os conhecimentos disponíveis e a utilização cuidadosa de métodos e técnicas de investigação (Ibidem.).

➤ Quanto a sua natureza

O tipo de pesquisa escolhido foi: Pesquisa Aplicada, pois, segundo GIL, ela tem como objetivo gerar conhecimentos para aplicação prática, dirigidos à solução de problemas específicos. Envolve verdades e interesses locais.

➤ Quanto a forma de abordagem

O tipo de pesquisa escolhido foi: Pesquisa Qualitativa pois ela considera que há uma relação dinâmica entre o mundo real e sujeito, de modo que a subjetividade do sujeito não pode ser traduzida em números. É descritiva e não requer o uso de métodos e técnicas estáticas. (PRODANOV,2013).

➤ Quanto aos seus objetivos

O tipo de pesquisa utilizada foi a pesquisa: Exploratória porque ela visa a proporcionar maior familiaridade com o problema com vistas a torná-lo explícito ou construir hipótese. Segundo Malhotra (2001), a pesquisa exploratória é usada em casos nos quais é necessário definir o problema com maior precisão.

➤ Quanto aos procedimentos técnicos

O tipo de pesquisa utilizada foi a pesquisa: Bibliográfica Segundo Vergara (2000), a pesquisa bibliográfica é desenvolvida a partir de material já elaborado, constituído, principalmente, de livros e artigos científicos e é importante para o levantamento de informações básicas sobre os aspectos direta e indiretamente ligados à nossa temática.

2 FUNDAMENTAÇÃO TEÓRICA

Há muitos anos, a engenharia de software se limitava à criação de aplicações para funcionar em computadores. Então, estudavam-se formas de lidar com os recursos e, sobretudo, de manter os sistemas atualizados e seguros, mesmo com essa limitação.

Nessa época, softwares eram basicamente transferidos a uma mídia e vendidos como um produto único. Assim, quem comprava a licença pagava uma vez e ficava responsável pelo controle da aplicação.

Recentemente, a computação em nuvem se tornou uma febre e o cenário mudou: entramos na era da aplicação web. Temos sites se tornando plataformas cada vez mais complexas.

Em comparação com sistemas tradicionais, a aplicação web muda completamente a forma de pensar a relação com as tecnologias, sendo uma alternativa poderosa e fácil de acessar. Assim, é uma ótima opção, do ponto de vista de quem desenvolve e de quem usa. Além disso, impulsionou o mercado dos desenvolvedores web.

2.1 O surgimento dos primeiros sistemas web

Na década de 1990, a internet era basicamente composta por páginas HTML estáticas, geralmente expositivas – seu propósito é simplesmente expor informações ao usuário -, que eram quase exclusivamente documentos. A partir de 1995, a linguagem Javascript foi apresentada e implementada, fornecendo mais rapidez para as páginas, além de possibilitar maior dinamicidade.

A maior mudança ocorreu em 2005, com o surgimento do Ajax, que permitiu maior flexibilidade e interatividade com o usuário, além de tornar a programação mais fácil. Por fim, em 2014 a linguagem HTML foi aprimorada devido ao HTML5, que fornece suporte à diferentes tipos de mídia, permitindo a criação de aplicações independentes do browser e da plataforma utilizada.

Vale ressaltar que, mesmo sendo cunhado em 1999, o termo “*web app*” só foi amplamente utilizado a partir do uso do HTML5 e do desenvolvimento da linguagem Java e Javascript. Com o avanço e disseminação da internet e de aparelhos móveis, as chamadas ***Application Service Providers (ASPs)*** ganharam muito espaço no mercado, fornecendo serviços e acesso web à softwares que anteriormente eram distribuídos como aplicações locais

2.2 Aplicacão web

Ela se refere a um software que opera por meio da internet, em contraposição à sua execução dependente de sistemas operacionais, lidando assim com um sistema plenamente funcional, elaborado com base em requisitos e princípios da engenharia de software.

As aplicações web referem-se a software acessado e utilizado através de navegadores web, sem a necessidade de instalação local. Elas operam remotamente em servidores e interagem com os usuários por meio de interfaces web. Este conceito é fundamentado na arquitetura cliente-servidor, em que o cliente (navegador) solicita serviços ao servidor que processa e retorna os resultados.

Tim O'Reilly, em seu artigo "Web 2.0: Compact Definition?" (2005), destaca a transição das aplicações web tradicionais para uma nova geração, enfatizando a colaboração e interatividade. Essa evolução inclui características como AJAX (Asynchronous JavaScript and XML) para melhorar a experiência do usuário.

No entanto, sua característica notável reside no fato de ser concebido para funcionar de forma nativa na internet.

O que impulsionou o desenvolvimento de plataformas web cada vez mais avançadas foi, precisamente, a tecnologia de computação em nuvem. Essa inovação alterou a abordagem das aplicações, passando de um modelo de produtos para um novo modelo centrado em serviços.

Nesse contexto, as aplicações operam como serviços, proporcionando funções essenciais aos usuários. Exemplificando, citamos o e-mail, as redes sociais e os aplicativos de edição de texto.

Portanto, aplicações web representam uma evolução contínua na forma como interagimos e utilizamos software, influenciando a arquitetura, tecnologias e a experiência do usuário ao longo do tempo.

2.3 Tipos de aplicações web

As aplicações web (WebApps) encontram aplicabilidade em diversas circunstâncias e têm a capacidade de atender às demandas dos clientes com maior agilidade do que um aplicativo nativo. Consequentemente, é possível afirmar que as aplicações baseadas na web têm ganhado maior destaque no mercado.

2.3.1 Aplicativos web dinâmicos

Os aplicativos web dinâmicos fazem uso de uma base de dados para que sua carga seja concluída, resultando em uma atualização constante da aplicação sempre que um usuário a acessa. Devido a essa dinâmica, eles se caracterizam como WebApps de maior complexidade.

Essa categoria de aplicativo viabiliza uma administração mais eficiente, uma vez que o responsável pela aplicação pode efetuar modificações e atualizar o conteúdo do site com facilidade e rapidez. Adicionalmente, essa modalidade de aplicação possibilita a incorporação de uma variedade de recursos.

2.3.2 Aplicativos web estáticos

Amplamente reconhecido entre os utilizadores que procuram partilhar artigos, este é um tipo de aplicação simplificada que apresenta um conteúdo conciso e tem a capacidade de operar em modo offline sem requerer a instalação de outras aplicações.

No contexto dessa categoria de aplicação web, os utilizadores não têm a autonomia para alterar ou administrar o site, uma vez que todo o seu conteúdo é completamente estático.

2.4 Características das aplicações web

Acessibilidade: Podem ser acessadas de qualquer lugar com conexão à internet, proporcionando flexibilidade de uso.

Interface Amigável: Interfaces intuitivas e amigáveis ao usuário, facilitando a navegação e interação.

Atualizações Contínuas: Facilidade em realizar atualizações e implementar novas funcionalidades sem a necessidade de reinstalações nos dispositivos dos usuários.

Compatibilidade Multiplataforma: Funcionam em diversos dispositivos e sistemas operacionais, como computadores, tablets e smartphones.

Integração de Dados: Facilidade em integrar dados de diferentes fontes, promovendo uma visão holística das informações.

Armazenamento na Nuvem: Utilização de serviços de armazenamento na nuvem para facilitar o acesso e a segurança dos dados.

Segurança: Implementação de medidas de segurança, como criptografia, para proteger dados sensíveis e garantir a privacidade dos usuários.

Interatividade: Possibilidade de interação em tempo real, como atualizações automáticas de conteúdo ou chat em tempo real.

Facilidade de Atualização: Atualizações e correções podem ser aplicadas de forma centralizada, garantindo uniformidade e consistência.

Análise de Dados: Incorporação de ferramentas analíticas para gerar insights a partir dos dados coletados, auxiliando em tomadas de decisão.

Personalização: Adaptação da experiência do usuário com base em preferências individuais, oferecendo uma experiência mais personalizada.

Resposta Rápida: Tempo de resposta rápido às interações do usuário, proporcionando uma experiência fluida e eficiente.

Integração de Mídias Sociais: Facilidade em integrar funcionalidades relacionadas a mídias sociais, permitindo compartilhamento e interação social.

Manutenção Simplificada: Manutenção centralizada, permitindo atualizações e correções sem a necessidade de intervenção nos dispositivos dos usuários.

Escalabilidade: Capacidade de expansão para atender a um maior número de usuários e demandas crescentes.

Estas características contribuem para a popularidade e eficácia das aplicações web em diversos contextos, incluindo o gerenciamento de farmácias.

2.5 Funcionamento de uma aplicação web

Uma aplicação web funciona com base na infraestrutura da internet. Por isso, é preciso entender, primeiro, como funciona a web.

O site fica armazenado em um servidor. Quando o usuário solicita acesso por meio de um endereço URL, ele automaticamente consegue conexão com o servidor DNS (servidor com uma lista de domínios) que, por sua vez, acessa o IP que referencia aquele site.

Então, o site envia informações de download para o usuário. Por fim, o usuário pode interagir com a plataforma enviando informações, alterando e salvando novos dados etc.

Tomemos como exemplo quando acessamos o site de uma rede social. Inicialmente, enviamos uma requisição para abrir a página. O servidor encontra o endereço e responde com as informações para download.

Em seguida, nos deparamos com um formulário de login. Ao preencher as informações e enviar, o sistema vai checar o banco de dados em que estão as informações de todos os usuários cadastrados. Uma vez checado, o sistema indica se houve um match com o que já existe ou não.

Caso haja um match, a aplicação permite o acesso e solicita ao servidor as informações da página principal com os dados específicos daquela conta. Logo, esse pacote é enviado para o download do user.

Caso não haja um match, o sistema volta na mesma página do formulário de login e indica um erro. Uma vez que já está logado, o usuário consegue realizar ações, como: publicar uma imagem, interagir em posts dos amigos, adicionar novos amigos, etc. Tudo isso representa interações que podem ser traduzidas em envio de requisições cliente-servidor.

2.5.1 Funcionamento das aplicações sever-side

O server-side, ou lado do servidor, é a parte de uma aplicação web que é executada no servidor. É responsável por processar as solicitações feitas pelos usuários e fornecer as respostas adequadas. Enquanto o client-side, ou lado do cliente, é executado no navegador do usuário, o server-side lida com a lógica de negócios, acesso a banco de dados, autenticação, entre outras funcionalidades.

Quando um usuário acessa um website ou envia uma solicitação para uma aplicação web, essa solicitação é enviada para o servidor. O servidor, por sua vez, processa essa solicitação utilizando a lógica de negócios e os dados armazenados em um banco de dados, se necessário. Após processar a solicitação, o servidor envia uma resposta de volta para o cliente, que pode ser uma página HTML, um arquivo de imagem, um documento PDF, entre outros.

2.5.2 Integração com client-side

Embora o server-side e o client-side sejam partes separadas de uma aplicação web, eles precisam trabalhar em conjunto para fornecer uma experiência completa aos usuários. O server-

side é responsável por fornecer os dados e a lógica necessários para o funcionamento da aplicação, enquanto o client-side é responsável por exibir esses dados e interagir com o usuário. A comunicação entre o server-side e o client-side geralmente é feita por meio de APIs (Application Programming Interfaces).

2.6 Processo de desenvolvimento de software

Um processo de desenvolvimento de software é um conjunto de atividades onde define-se quem faz o que, como e quando com o objetivo final de obter um produto de software. Segundo PRESSMAN e MAXIM (2016), essas atividades constituem um conjunto mínimo para se obter um produto de software.

É uma área de estudo de Engenharia de Software que consiste de boas práticas para se obter um software de qualidade e com os requisitos definidos atingidos. Para desenvolver um projeto de software, é necessário seguir alguns passos, porém deve-se entender que não existe um processo perfeito, sendo recomendado cada empresa aperfeiçoar o seu. SOMMERVILLE et al. (2008) cita que, em casos de sistemas críticos, é necessário um processo bem estruturado, e para sistemas de negócios com requisitos que mudam rapidamente, um processo mais flexível e ágil é mais eficaz.

O primeiro passo consiste na viabilidade econômica, verificando se terá lucros suficientes para o projeto. Após isso, deve-se começar as reuniões com os clientes, para o levantamento de requisitos. Segundo SOMMERVILLE et al. (2008), requisitos são descrições ou restrições que definem as propriedades de um sistema. Com os requisitos é possível visualizar as necessidades do cliente e assim, identificar as regras de negócio, transformando-as em funcionalidades do sistema. Nessa fase, o analista e o usuário devem entender qual é o problema a ser solucionado e quais as prioridades do projeto, para então gerar a especificação do projeto.

A especificação do projeto tem a incumbência de descrever de forma precisa o software em questão, nos mínimos detalhes, construindo modelos a fim de representar o sistema a ser desenvolvimento. Um dos modelos é o projeto de arquitetura, descrito em mais detalhes na Subseção 2.1.1. Em seguida, vem a parte da implementação desta especificação, ou seja, sua codificação de fato. Segundo SOMMERVILLE et al. (2008), “o estágio de implementação do desenvolvimento de software é o processo de conversão de uma especificação de sistema em um sistema executável”. Escolhe-se uma linguagem de programação, banco de dados a utilizar,

arquitetura, padrão de interface gráfica, etc. Com a codificação finalizada, é necessário testar o sistema já pronto, para garantir que todos os pontos da especificação estão aderentes. Dependendo da forma que foi conduzido o processo de desenvolvimento, se faz necessário elaborar o documento do software que será utilizado para entendimento do sistema e para futuras manutenções do mesmo.

Após todas as atividades concluídas, o sistema é liberado para produção assistida com os usuários. É nesta fase que eles fazem o teste do sistema e recebem um treinamento para uso e ao concluir, o sistema está entregue e em produção. Quando um sistema está entregue em produção, futuras modificações entram agora no ciclo de manutenção do software, podendo ser melhorias ou correções de erros.

É importante enfatizar que estas atividades acima descritas podem ser executadas de forma sequencial ou paralela ou em interações, dependendo do modelo de processo de desenvolvimento utilizado. Exemplos de modelos de processo de desenvolvimento de software incluem:

- **Cascata:**

Sendo um dos modelos mais populares em engenharia de software, o modelo cascata ou top down, segundo ROYCE (1987) ocorre de forma linear, agrupadas em tarefas e executadas sequencialmente, onde a saída é entrada para outra.

As fases são: Análise de requisitos, projeto, implementação, testes (validação), integração, e manutenção de software (SOMMERVILLE et al., 2008).

- **RUP:**

É uma especialização do Processo Unificado que, por sua vez, é baseado no modelo incremental. O RUP (do inglês, Rational Unified Process) adota algumas premissas para tentar minimizar riscos e problemas. VIANNA (s.d.) cita as seguintes premissas:

- ❖ Uso de iterações para evitar o impacto de mudanças no projeto;
- ❖ Gerenciamento de mudanças;
- ❖ Abordagens dos pontos de maior risco o mais cedo possível;

As quatro fases do RUP são (VIANNA, s.d.):

Concepção - entendimento da necessidade e visão do projeto;

Elaboração – especificação e abordagem dos pontos de maiores riscos;

Construção - desenvolvimento principal do sistema;

Transição - ajustes, implantação e transferência de propriedade do sistema;

O RUP costuma ser visto como um processo pesado devido ao grande número de atividades e artefatos e à rigidez e controle. No entanto, há uma variação chamada Processo Unificado Ágil, que visa desenvolver apenas um pequeno conjunto de atividades e artefatos, tornando-se um processo mais leve e adaptativo (LARMAN, 2004).

2.7 Framework

Um framework em desenvolvimento de software, é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. Um framework pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação. Ao contrário das bibliotecas, é o framework quem dita o fluxo de controle da aplicação, chamado de **Inversão de Controle**.

O framework atua onde há funcionalidades em comum a várias aplicações, porém para isso as aplicações devem ter algo razoavelmente grande em comum para que o mesmo possa ser utilizado em várias aplicações.

Padrões de projeto de software não se confundem com frameworks, pois padrões possuem um nível maior de abstração. Um framework inclui código, diferentemente de um padrão de projeto. Um framework pode ser modelado com vários padrões de projeto, e sempre possuem um domínio de uma aplicação particular, algo que não ocorre nos padrões de projeto de software.

Framework é um conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação. Frameworks possuem vantagens, tais como: maior facilidade para a detecção de erros, por serem peças mais concisas de software; concentração na abstração de soluções do problema que estamos tratando; eficiência na resolução dos problemas e otimização de recursos.

2.8 Arquitetura de software

A arquitetura de software refere-se à estrutura fundamental de um sistema, abrangendo seus componentes, seus relacionamentos e os princípios que orientam seu design.

A arquitetura de software engloba elementos de software, as propriedades visíveis externamente desses elementos e os relacionamentos entre eles. Essa definição também inclui as diretrizes para a composição de elementos para garantir a integridade do sistema ao longo do tempo. (Richard N. Taylor, Nenad Medvidović, e Eric M. Dashofy).

Segundo (Mary Shaw e David Garlan) A arquitetura de software representa uma configuração significativa de elementos de software selecionados e os relacionamentos entre eles, considerando restrições e princípios.

A arquitetura de software abrange a estrutura do sistema, que compreende componentes de software, as propriedades externas visíveis desses componentes e as relações entre eles, (Grady Booch, Robert C. Martin, e James Rumbaugh).

2.8.1 Arquitetura de desenvolvimento de software m.v.c

O MVC é um padrão de arquitetura de software. O MVC sugere uma maneira para você pensar na divisão de responsabilidades, principalmente dentro de um software web. O princípio básico do MVC é a divisão da aplicação em três camadas: a camada de interação do usuário (view), a camada de manipulação dos dados (model) e a camada de controle (controller).

Com o MVC, é possível separar o código relativo à interface do usuário das regras de negócio, o que sem dúvida traz muitas vantagens que veremos mais à frente. Quando falamos sobre o MVC, cada uma das camadas apresenta geralmente as seguintes responsabilidades

- **Model (Modelo):** A responsabilidade dos models é representar o negócio. Também é responsável pelo acesso e manipulação dos dados na sua aplicação.
- **View (Vista):** A view é responsável pela interface que será apresentada, mostrando as informações do model para o usuário.
- **Controller (Controlador):** É a camada de controle, responsável por ligar o model e a view, fazendo com que os models possam ser repassados para as views e vice-versa.

Não dá para falar do MVC sem citar a importância que ele traz em meio ao desenvolvimento de software.

Uma dessas vantagens é que ele nos ajuda a deixar o código mais manutenível, ou seja, mais fácil de fazer manutenção, já que temos as responsabilidades devidamente separadas. Isso também traz uma facilidade na compreensão do código, além da sua reutilização.

Além disso, você tem um código mais testável, pois ele é mais granular: se você tem uma aplicação onde, por exemplo, na página de listagem de usuários, o nome do usuário está sendo cortado ou não está sendo exibido da maneira correta, é muito mais fácil você fazer um teste que atinja somente as estruturas de views.

Aqui, podemos ver claramente que você tem um problema de apresentação: os models não são responsáveis por aspectos de apresentação, assim como os controllers também não são... Veja que é até mais fácil de identificar que o problema está na view. Por isso, você consegue corrigir somente a view e testá-la de maneira isolada.

2.9 Modelo de banco de dado

Um modelo de banco de dados mostra a estrutura lógica de um banco de dados, incluindo as relações e restrições que determinam como os dados podem ser armazenados e acessados. Modelos de banco de dados individuais são projetados com base nas regras e nos conceitos do modelo de dados mais abrangente que os designers adotam. A maioria dos modelos de dados pode ser representada por um diagrama de banco de dados acompanhante.

2.9.1 *Modelo relacional*

O modelo mais comum, o modelo relacional, classifica dados em tabelas, também conhecidas como relações, cada uma das quais consiste em colunas e linhas. Cada coluna lista um atributo da entidade em questão, como preço, código postal ou data de nascimento. Juntos, os atributos em uma relação são chamados de domínio. Um determinado atributo ou combinação de atributos é escolhido como uma chave primária que pode ser consultada em outras tabelas, quando é chamada de chave estrangeira.

Cada linha, também chamada de tupla, inclui dados sobre uma instância específica da entidade em questão, como um determinado colaborador. O modelo também explica os tipos de relações entre essas tabelas, incluindo relações uma para uma, uma para muitas e muitas para muitas.

Dentro do banco de dados, as tabelas podem ser normalizadas ou levadas a cumprir as regras de normalização que tornam o banco de dados flexível, adaptável e redimensionável. Quando normalizado, cada dado é atômico, ou dividido em pequenos pedaços úteis.

2.10 O conceito de farmácia

O termo "farmácia" refere-se a um estabelecimento que lida com a preparação, dispensação e venda de medicamentos, além de oferecer serviços relacionados à saúde. O conceito de farmácia é multifacetado, abrangendo áreas como assistência farmacêutica, atenção farmacêutica e gestão de medicamentos.

Segundo a "Encyclopédia Britânica", a farmácia é uma disciplina da saúde que envolve a aplicação de conhecimentos científicos na produção, preparação e dispensação de medicamentos.

Dessa forma, o conceito de farmácia vai além do simples comércio de medicamentos, englobando práticas profissionais voltadas para a promoção da saúde e garantia do uso seguro e eficaz dos medicamentos.

2.10.1 Vantagens da Aplicação Web no Gerenciamento de Farmácia

- **Acessibilidade Remota:** Acesso aos dados da farmácia de qualquer local, facilitando o gerenciamento mesmo fora do estabelecimento.
- **Atualizações em Tempo Real:** informações sobre estoque, vendas e transações são atualizadas instantaneamente, proporcionando uma visão em tempo real da situação.
- **Integração de Dados:** Facilidade em integrar dados de diferentes departamentos, como estoque, vendas e gestão de clientes, promovendo uma abordagem mais holística.
- **Facilidade de Backup e Recuperação:** Sistemas automáticos de backup asseguram a proteção dos dados, permitindo rápida recuperação em caso de falhas ou perdas.
- **Análise de Dados Avançada:** Ferramentas analíticas integradas fornecem insights valiosos sobre padrões de vendas, comportamento do cliente e eficiência operacional.
- **Melhoria na Experiência do Cliente:** Maior eficiência no atendimento ao cliente devido a processos mais rápidos e personalizados.
- **Adaptação a Mudanças:** Facilidade em realizar atualizações e personalizações no sistema para se adaptar a mudanças no mercado ou nas regulamentações.
- **Integração de Canais de Vendas:** Capacidade de integrar vendas online e offline, ampliando as opções de compra para os clientes

2.10.2 Desvantagens da Aplicação Web no Gerenciamento de Farmácia

- **Dependência da Conectividade com a Internet:** A necessidade de uma conexão estável à internet pode ser uma limitação, especialmente em áreas com infraestrutura precária.
- **Questões de Segurança:** Ineribilidades à segurança online, exigindo medidas robustas para proteger dados sensíveis contra ameaças cibernéticas.
- **Custo Inicial e Manutenção:** A implementação inicial e os custos contínuos de manutenção podem ser significativos, especialmente para pequenas farmácias.
- **Curva de Aprendizado:** Necessidade de treinamento dos funcionários para utilização eficaz, podendo gerar resistência inicial.
- **Personalização Limitada:** Alguns sistemas podem ter limitações na personalização, o que pode ser um desafio para atender necessidades específicas.
- **Problemas Técnicos:** Possibilidade de enfrentar problemas técnicos, como falhas no sistema, que podem impactar as operações cotidianas.

2.10.3 O Impacto de um Sistema de Gerenciamento nas Farmácias

A implementação de um sistema de gerenciamento em uma farmácia pode resultar em transformações significativas, redefinindo a eficiência operacional e melhorando substancialmente a qualidade dos serviços prestados. Este impacto é observado em diversas áreas-chave:

- **Controle de Estoque Aprimorado:** Um sistema de gerenciamento possibilita um controle preciso do estoque, evitando excessos ou escassez, otimizando as operações e reduzindo perdas financeiras.
- **Eficiência nas Transações:** A automação de processos, como vendas e gestão de receitas, acelera as transações, resultando em atendimento mais rápido aos clientes e redução de filas.
- **Rastreabilidade de Produtos:** A Capacidade de rastrear produtos desde a aquisição até a venda proporciona maior segurança e qualidade, sendo vital em casos de recalls ou garantias.
- **Gestão de Clientes Aprimorada:** Sistemas de gerenciamento permitem o registro e acompanhamento do histórico de clientes, facilitando atendimentos personalizados e estratégias de fidelização.

- **Relatórios Analíticos:** Ferramentas analíticas integradas proporcionam insights valiosos sobre padrões de vendas, comportamento do cliente e desempenho financeiro, auxiliando em decisões estratégicas.
- **Segurança de Dados:** A implementação de medidas de segurança robustas protege informações sensíveis, garantindo conformidade com regulamentações e aumentando a confiança do cliente.
- **Integração de Canais de Vendas:** Sistemas modernos facilitam a integração de vendas online e offline, ampliando o alcance da farmácia e proporcionando conveniência aos clientes.
- **Aprimoramento da Experiência do Cliente:** Processos eficientes, menor tempo de espera e atendimento personalizado contribuem para uma experiência positiva do cliente, promovendo a fidelização.
- **Adaptação a Mudanças no Mercado:** A capacidade de personalizar e atualizar o sistema permite que as farmácias se adaptem rapidamente a mudanças no mercado, mantendo-se competitivas.
- **Redução de Erros:** A automação reduz a probabilidade de erros humanos em transações e registros, garantindo precisão e conformidade com regulamentações.

Em suma, a implementação de um sistema de gerenciamento nas farmácias não apenas moderniza suas operações, mas também eleva a qualidade do serviço, contribuindo para um ambiente farmacêutico mais eficiente, seguro e centrado no cliente.

2.11 Ferramentas utilizadas

FERRAMENTAS E TECNOLOGIAS USADAS	DESCRIÇÃO	VERSÃO
CSS (Cascading Style Sheets)	Foi utilizado para deixar a nossa página mais interativa com uma visualização, mas agradável ou estilos mais interativos desde as cores até as molduras.	3
HTML(Hypertext Markup Language)	Foi utilizado para a criação dos formulários, estruturar e formatar conteúdo da aplicação web.	5
JAVASCRIPT	usado para tornar a aplicação mais dinâmica e melhorar a interatividade com o usuário	ES6
PHP	Foi utilizado para desenvolver a parte do back-end da aplicação para facilitar a conexão entre o servidor e a interface do usuário	7.2
Jquery	Utilizamos para criar animações e efeitos visuais existentes dentro da aplicação web	3.7.1
Mysql	Utilizamos para gerenciar o nosso banco de dados	8.0.21

Tabela 1:Ferramentas e tecnologias utilizadas. fonte: autor próprio.

2.12 Requisitos funcionais

Nº	REQUISITOS FUNCIONAIS	DESCRIÇÃO
RF01	Fazer login	Este caso de uso consiste na verificação dos dados do usuário, afim de permitir o acesso ao sistema.
RF02	Editar perfil	Permite com que o usuário edite os seus dados pessoais como: e-mail, telefone etc.
RF03	Alterar senha	Neste caso de uso permite que o usuário atualize ou altere a sua senha.
RF04	Visualizar Catálogo	Este caso de uso permite que o usuário veja os produtos a serem comercializados pela farmácia.
RF05	Pesquisar produto	Este caso de uso permite ao usuário rastrear de modo rápido um produto.
RF06	Registrar venda de produto	Permite ao usuário armazenar ou registrar as vendas dos produtos a serem realizadas.
RF07	Encerrar sessão	Consiste no encerramento de sessão do sistema.
RF08	Cadastrar técnico	Permite ao admin cadastrar técnicos.
RF09	Eliminar produto	Consiste na eliminação de produtos cadastrados.
RF10	Cadastrar produto	Este caso de uso permite o cadastro de produtos no sistema para facilitar no controle do stock.
RF11	Listar vendas realizadas	Consiste na listagem das vendas realizadas para aprimorar nas tomadas de decisão da farmácia.
RF12	Cadastrar fornecedores	Este caso de uso permite o cadastro no sistema das empresas que fornecem os produtos na farmácia.
RF13	Adicionar atributos do produto	Este caso de uso permite adicionar atributos dos produtos como: descrição, percentagem etc.
RF14	Atualizar atributo dos produtos	Este caso de uso permite atualizar as informações dos atributos dos produtos cadastrado no sistema.
RF15	Eliminar fornecedores	Consiste na eliminação das empresas fornecedoras dos produtos da farmácia.
RF16	Visualizar técnicos	Permite ao admin visualizar todos os técnicos cadastrados no sistema.
RF17	Visualizar fornecedores	Permite ao admin visualizar todos os fornecedores cadastrados no sistema.

Tabela 2:Requisitos funcionais

2.13 Requisitos não funcionais

Desempenho: O sistema deverá ter velocidade de resposta, do tempo de carregamento de páginas, tempo de processamento de transações.

Segurança: A segurança é essencial para proteger os dados e impedir acessos não autorizados. Isso pode incluir criptografia de dados, controle de acesso baseado em funções, auditoria de atividades e conformidade com padrões de segurança, como GDPR, HIPAA, entre outros.

Usabilidade: O sistema garantirá facilidade de uso para seus usuários. Isso pode envolver uma interface intuitiva, design responsivo, acessibilidade para pessoas com deficiência e suporte para múltiplos idiomas.

Confiabilidade: O sistema deverá ter uma ampla disponibilidade, para atender as exigências da farmácia.

Escalabilidade: O sistema crescerá de acordo com o aumento da demanda.

Manutenção e Suporte: o sistema será de fácil manutenção, atualizações de software, correção de bugs e suporte técnico contínuo.

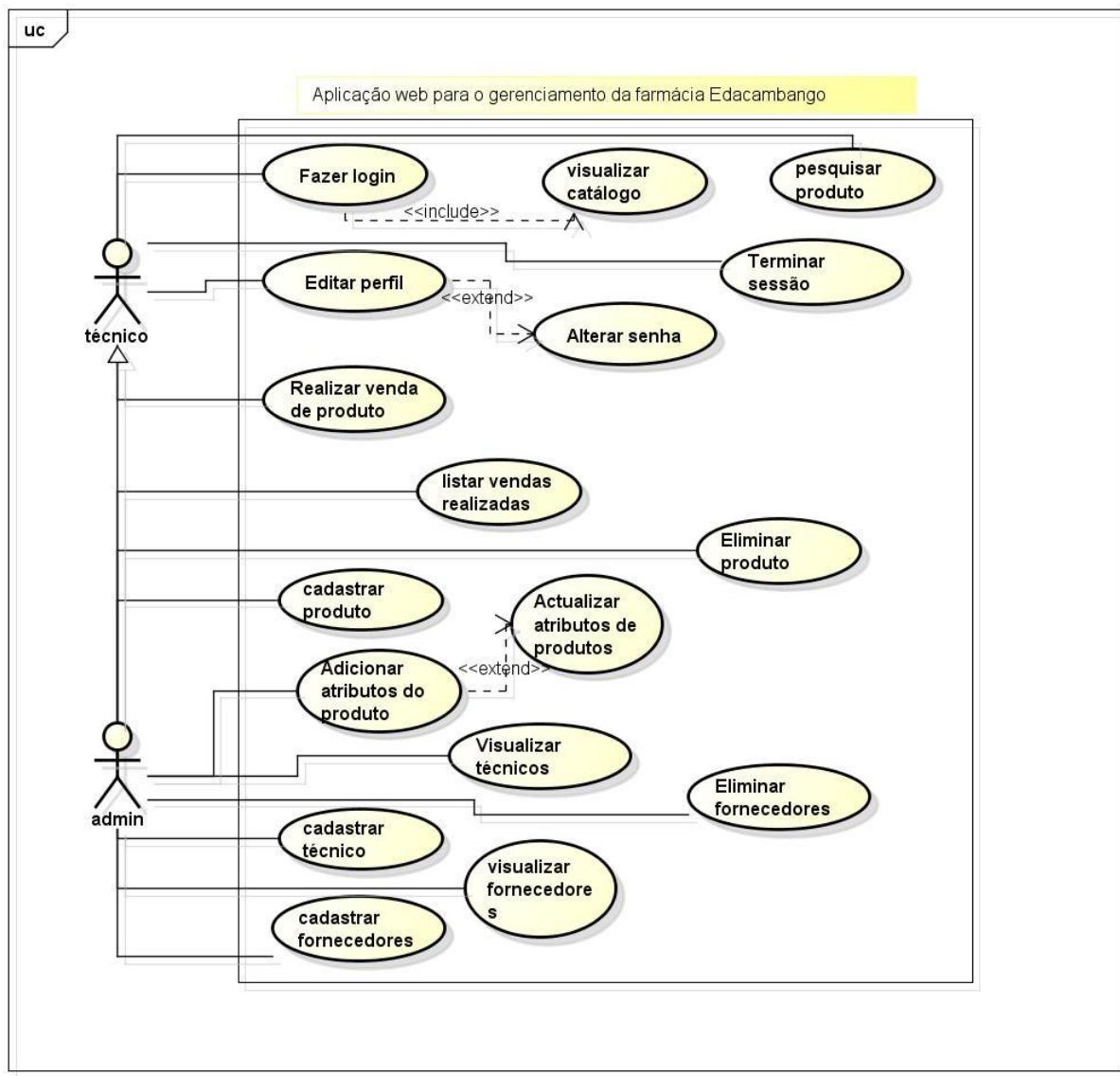
Compatibilidade: O sistema deverá ser compatível com diferentes dispositivos, navegadores, sistemas operacionais.

Eficiência: O sistema deverá utilizar os recursos de hardware de forma eficiente, como memória, armazenamento e capacidade de processamento, para minimizar custos operacionais.

2.14 Diagramas

2.14.1 Diagrama de caso de uso

Figura 1:DIAGRAMA DE CASO DE USO



powered by Astah

FONTE: AUTOR PRÓPRIO.

2.14.1.1 Documentação de caso de uso "Fazer login"

NOME	Fazer login
AUTORES	Administrador, técnico
DESCRIÇÃO	Este caso de uso consiste na verificação dos dados do usuário, afim de permitir o acesso ao sistema.
PRÉ-CÓDIÇÃO	O usuário deve estar registrado no sistema
PÓS-CÓDIÇÃO	Ter o login feito com sucesso
FLUXO NORMAL	
ACÇÃO DO AUTOR	ACÇÃO DO SISTEMA
1º o usuário deve solicitar a tela de fazer login	2º o sistema disponibiliza a tela de login 5º o sistema deve verificar os dados do usuário 6º o sistema deve validar os dados
3º o usuário deve preencher o formulario de fazer login	7º o sistema deve levar o usuário para página inicial
4º submeter os dados	
FLUXO ALTERNATIVO	
No 6º ponto: o sistema não valida os dados 1º Alternativa O sistema informa o erro ocorrido e retorna no ponto 2º	

Tabela 3:Documentação do caso de uso fazer login. Fonte: autor próprio.

2.14.1.2 Documentação de caso de uso ”Editar perfil”

NOME	Editar perfil
AUTORES	Administrador, técnico
DESCRIÇÃO	Permite com que o usuário edite os seus dados pessoais como: e-mail, telefone etc.
PRÉ-CÓDIÇÃO	O usuário deve estar logado no sistema
PÓS-CÓDIÇÃO	Ter o perfil editado com sucesso
FLUXO NORMAL	
ACÇÃO DO AUTOR	ACÇÃO DO SISTEMA
1º o usuário deve solicitar a página de editar perfil	2º o sistema disponibiliza a página de editar perfil 5º o sistema deve validar os dados 6º o sistema deve registar os dados editados
3º o usuário deve preencher os campos dos quais deseja editar os seus dados	7º o sistema deve informar ao usuário perfil editado com sucesso
4º submeter os dados	
FLUXO ALTERNATIVO	
No 5º ponto: o sistema não valida os dados 1º Alternativa O sistema informa o erro ocorrido e retorna no ponto 2º	

Tabela 4: Documentação de caso de uso editar perfil. Fonte: autor próprio

2.14.1.3 Documentação de caso de uso ”Alterar senha”

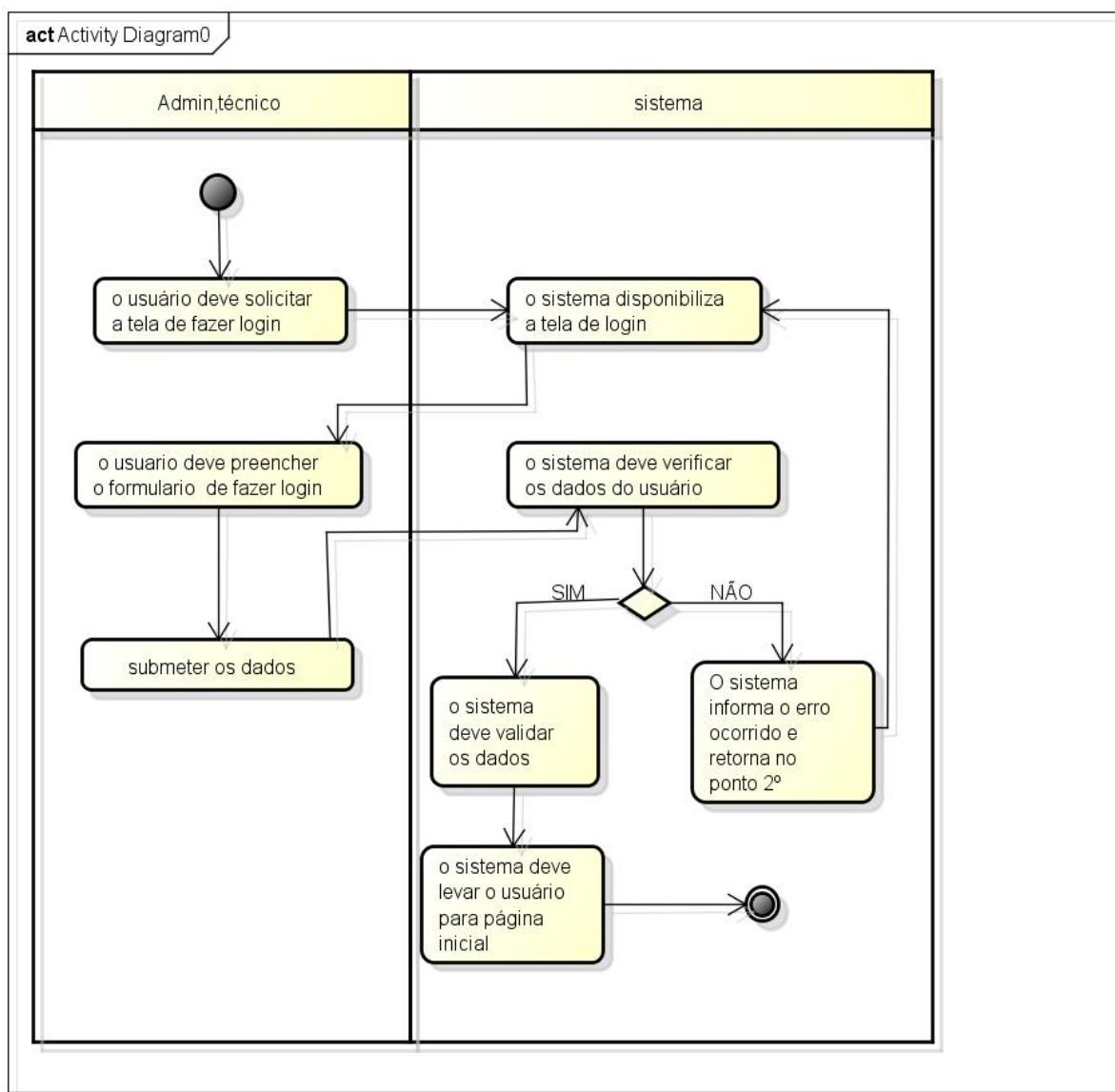
NOME	Alterar senha
AUTORES	Administrador, técnico
Descrição	Neste caso de uso permite que o usuário atualize ou altere a sua senha.
PRÉ-CÓDIÇÃO	O usuário deve estar logado no sistema
PÓS-CÓDIÇÃO	Ter a senha alterada com sucesso
FLUXO NORMAL	
ACÇÃO DO AUTOR	ACÇÃO DO SISTEMA
1º o usuário deve solicitar a página de editar perfil	2º o sistema disponibiliza a página de editar perfil 5º o sistema deve verificar a senha 6º o sistema deve registar a senha
3º o usuário deve preencher o formulário de alterar senha	7º o sistema deve informar ao usuário senha alterada com sucesso
4º submeter os dados	
FLUXO ALTERNATIVO	
No 5º ponto: o sistema não verifica a senha 1º Alternativa O sistema informa o erro ocorrido e retorna no ponto 2º	

Tabela 5:Documentação de caso de uso alterar senha. Fonte: autor próprio.

2.14.2 Diagrama de actividade

2.14.2.1 Diagrama de atividade “Fazer login”

Figura 2:Diagrama de atividade fazer login

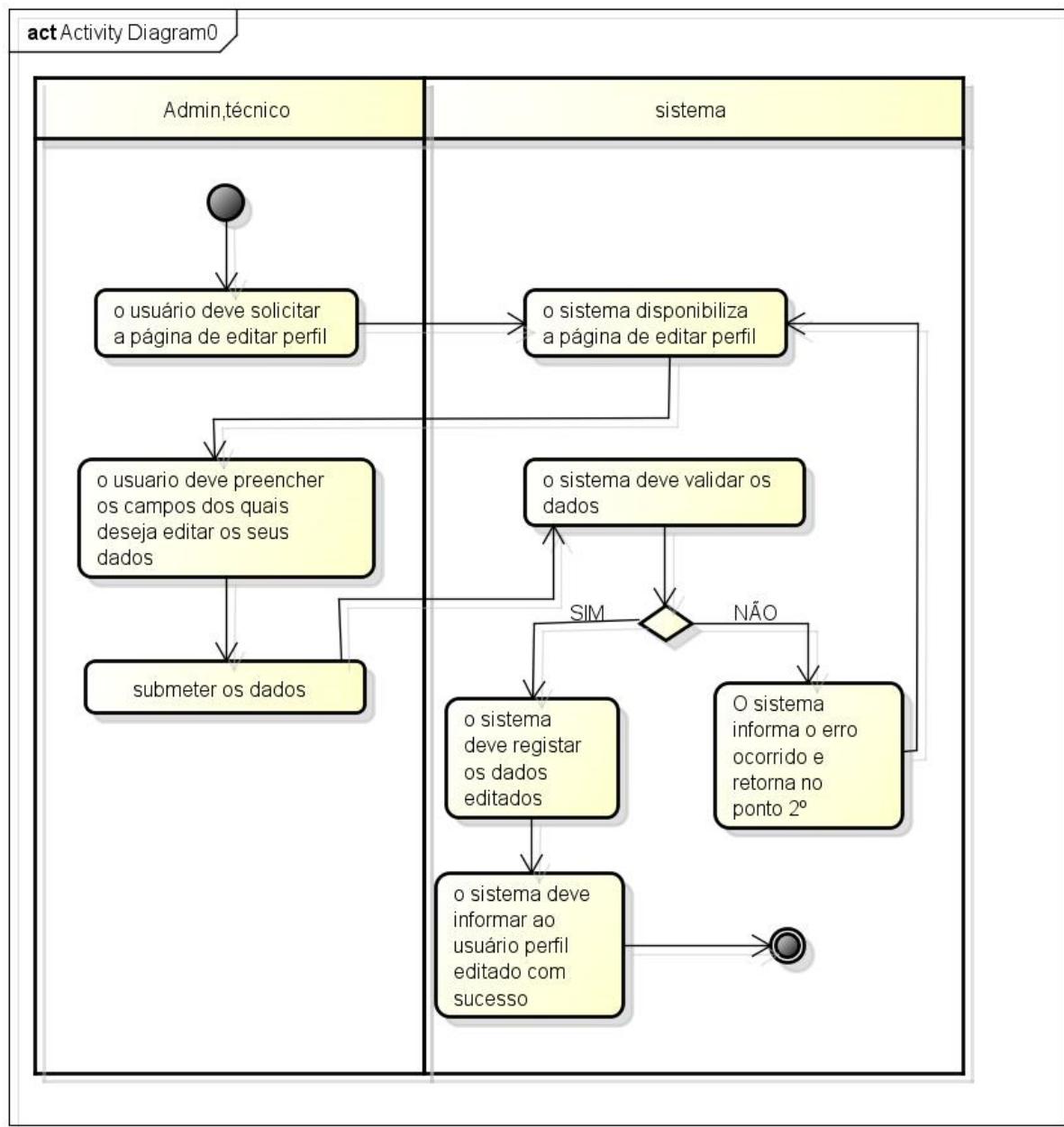


powered by Astah

Fonte: autor próprio

2.14.2.2 Diagrama de atividade “Editar perfil”

Figura 3:Diagrama de atividade editar perfil

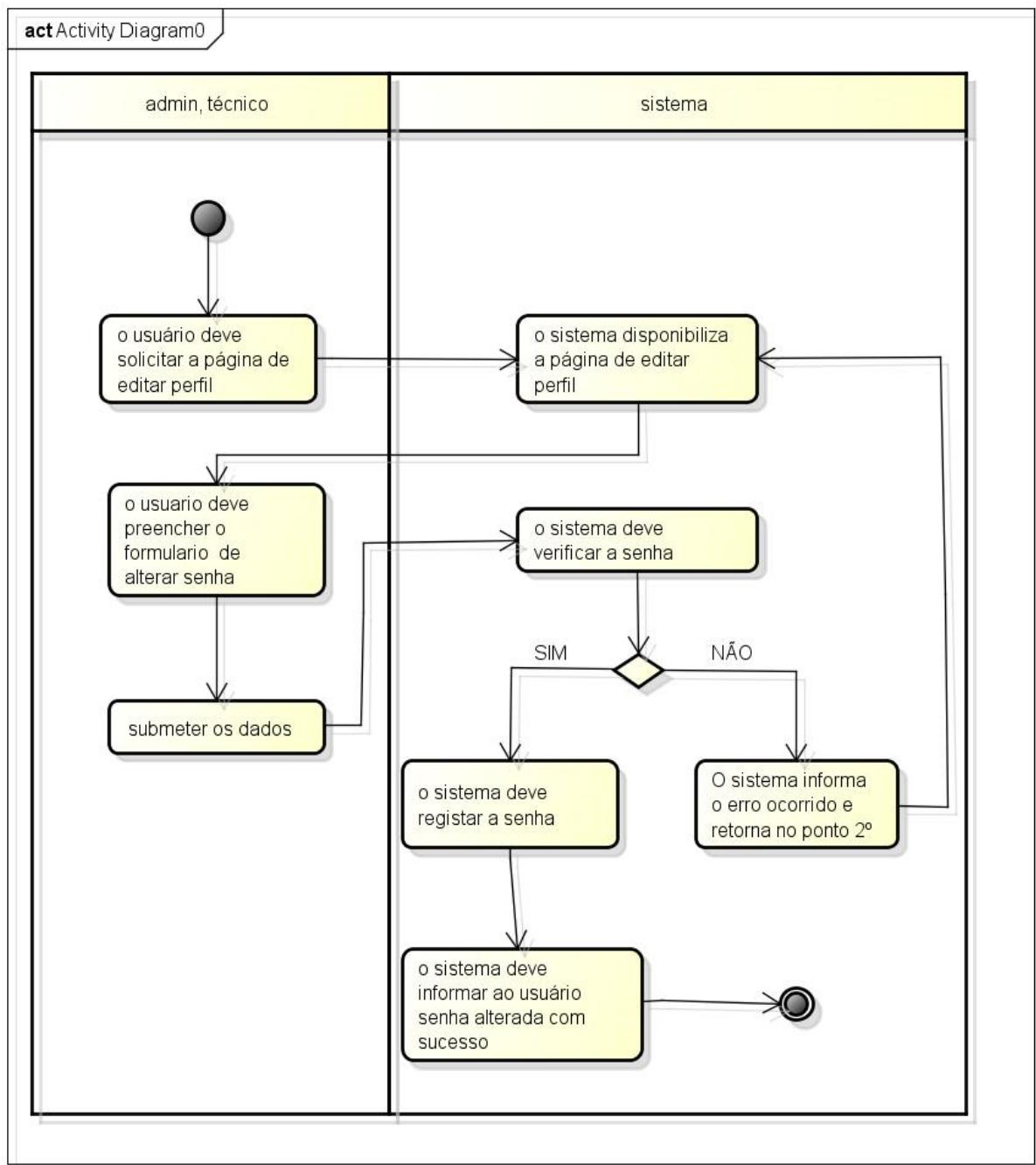


powered by Astah

Fonte: autor próprio

2.14.2.3 Diagrama de atividade “Alterar senha”

Figura 4:Diagrama de atividade alterar senha



powered by Astah

Fonte: Autor próprio

CONCLUSÃO

Concluímos que o desenvolvimento de uma aplicação web voltada para o gerenciamento de uma farmácia é uma abordagem inovadora e essencial na era digital, oferecendo uma solução eficaz para a complexidade das operações farmacêuticas. Em um ambiente onde a agilidade, precisão e segurança são cruciais, pois, uma aplicação web bem projetada pode revolucionar a maneira como as farmácias conduzem suas atividades diárias. Esta iniciativa não apenas simplifica processos, como o controle de estoque e vendas, mas também proporciona uma experiência aprimorada para os clientes. Ao integrar tecnologias modernas, segurança robusta e funcionalidades personalizadas, a aplicação não apenas otimiza a gestão interna, mas também impulsiona a eficiência operacional e a tomada de decisões estratégicas da farmácia.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABBATE, Janet. **Inventando a Internet**. MIT Press, 1999.
- BERNERS-LEE, Tim. Tecendo a Web: **O Projeto Original e o Destino Final da World Wide Web por Seu Inventor**. HarperBusiness, 2000.
- HAFNER, Katie; LYON, Matthew. Onde os Magos Ficam Acordados Até Tarde: **As Origens da Internet**. Simon & Schuster, 1996.
- PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software**. 8. ed. McGraw Hill Brasil, 2016.
- SOMMERVILLE, Ian; ARAKAKI, Renata; MELNIKOFF, Sonia S. S. **Engenharia de software**. Pearson Prentice Hall, 2008.
- ROYCE, Winston W. Managing the development of large software systems: **concepts and techniques**. In: Proceedings of the 9th international conference on Software Engineering. IEEE Computer Society Press, 1987.
- GARLAN, David. Software architecture: a roadmap. In: **Proceedings of the Conference on the Future of Software Engineering**, p. 91-101. ACM, 2000.
- LARMAN, Craig. Applying UML and Patterns: **An Introduction to Object-Oriented Analysis and Design and Iterative Development**. 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2004.
- HARTLEY, Frank; KRANTZ, John C. **Pharmacy**. Enciclopédia Britânica, ed. 1, Londres, 1963.
- VIANNA, Marcelo. Conheça o Rational Unified Process (RUP). Disponível em: <<http://www.linhadecodigo.com.br/artigo/79/conheca-o-rational-unified-processrup.aspx>>. Acesso em: 12 mar. 2024.
- "Aplicação web: entenda o que é, como funciona e a diferença para site." Rock Content. Disponível em: <<https://rockcontent.com/br/talent-blog/aplicacao-web/>>. Acesso em: 12 mar. 2024.
- MATTAR, F. N. **Pesquisa de marketing**. 3. ed. São Paulo: Atlas, 2001.
- GIL, Antônio C. **Métodos e técnicas de pesquisa social**. 5. ed. São Paulo: Atlas, 1999.
- VERGARA, Sylvia C. **Projetos e relatórios de pesquisa em administração**. 3. ed. Rio de Janeiro: Atlas, 2000.