

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA**

**Faculdade de Tecnologia Baixada Santista
Rubens Lara**

**Curso Superior de Tecnologia em
Sistemas para Internet**

**ANDRÉ LUIZ BORGES MOUTINHO
ARTHUR ROSA PASSOS JUNIOR**

**SIG OS
Sistema para Gestão de Ordem de Serviço**

**Santos, SP
2024**

**ANDRÉ LUIZ BORGES MOUTINHO
ARTHUR ROSA PASSOS JUNIOR**

**SIG OS
Sistema para Gestão de Ordem de Serviço**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia Rubens Lara, como exigência para a obtenção do Título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Me. Davi Silvestre Moreira dos Reis

**Santos, SP
2024**

RESUMO

Com o avanço da tecnologia, tornou-se cada vez mais necessário informatizar pequenas e médias empresas para otimizar e automatizar processos. Ao fazer isso, traz benefícios como aumento de serviços e fluxo de vendas, que se concentra em oferecer a melhor solução, bem como a satisfação do cliente. Mas muitos empreendedores acabam não investindo em um sistema informatizado para gestão de negócios, temendo altos custos com manutenção e até mesmo com a própria implementação. Com base nesse problema, o objetivo deste trabalho é desenvolver um aplicativo *web* para gerenciamento de ordens de serviço que auxilie na gestão completa de negócios comerciais e prestadores de serviços. Para o desenvolvimento desta aplicação será utilizado o *framework* Codeigniter 4 com *backend* em linguagem PHP 8.3, ele também terá o uso de banco de dados MySQL. Os resultados realmente refletiram alguma escassez de uma base tecnológica quando o assunto se refere a um *software* de gestão que pode atender às necessidades de forma mais ampla, tanto para a área comercial, e para a área de prestação de serviços; tal indicativo demonstra o quão benéfico e rentável um aplicativo *web* com essas características pode ser.

Palavras-chaves: Gestão. Negócios. Aplicação. Web.

ABSTRACT

With the advancement of technology, it has become increasingly necessary to computerize small and medium businesses in order to optimize and automate processes. By doing so, it brings benefits such as increased services and sales flow, which focuses on delivering the best solution as well as customer satisfaction. But many entrepreneurs end up not investing in a computerized system for business management, fearing high costs with maintenance and even with the implementation itself. Based on this problem, the objective of this paper is to develop a web application for service order management that assists in the complete management for commercial business and service providers. For the development of this application it will be used the framework Codeigniter 4 with backend in PHP 8.3 language, it will also have the use of MySQL database. The results really reflected some scarcity of a technological foundation when the subject refers to a management software that can meet the needs more broadly, both for the commercial area, and for the service provision area; such indicative demonstrates how beneficial and profitable a web application with these features can be.

Keywords: Management. Business. Application. Web.

LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|-----------|
| WEB - World Wide Web..... | 10 |
| SIG OS - Sistema para Gestão de Ordem de Serviço..... | 11 |
| OS - Ordem de Serviço..... | 13 |
| UML- Unified Modeling Language..... | 15 |
| MER - Modelo Entidade Relacionamento..... | 22 |
| SGBD - Sistemas Gerenciadores de Banco de dados..... | 22 |
| SQL - Structure Query Language..... | 23 |
| PHP - Hypertext Preprocessor..... | 23 |
| MVC - Modelo Visão Controlador..... | 23 |
| <i>HTML - HyperText Markup Language.....</i> | <i>23</i> |
| CRUD - Create Read Update Delete..... | 24 |
| URL - Uniform Resource Locator..... | 25 |
| ID - Identity..... | 27 |
| UI - User Interface..... | 28 |
| HMVC - Hierarchical Model View Controller..... | 30 |
| ERP - Enterprise Resource Planning..... | 32 |
| CEP - Código de Endereçamento Postal..... | 33 |

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Ilustração 01 - Visualização de Ordens de Serviço do EOS-Os..... | 13 |
| Ilustração 02 - Visualização de Ordens de Serviço do OnlineOS..... | 13 |
| Ilustração 03 - Visualização de Ordens de Serviço do SistemaSOMA..... | 14 |
| Ilustração 04 - Adicionar Ordem de Serviço | 20 |
| Ilustração 05 - Adicionar Produto | 21 |
| Ilustração 06 - Adicionar Serviço..... | 21 |
| Ilustração 07 - Model OS Adicionar..... | 24 |
| Ilustração 08 - Model OS Editar | 25 |
| Ilustração 09 - Model OS Visualizar | 26 |
| Ilustração 10 - Model OS Excluir | 27 |
| Ilustração 11 - Tela Cadastro de Cliente | 28 |
| Ilustração 12 - Tela de Adicionar Produto | 29 |
| Ilustração 13 - Paleta de Cores | 30 |

SUMÁRIO

| | |
|--|----|
| 1 INTRODUÇÃO | 9 |
| 1.1 OBJETIVO | 10 |
| 1.1.1 OBJETIVO GERAL..... | 10 |
| 1.1.2 OBJETIVOS ESPECÍFICOS..... | 11 |
| 1.2 ESTADO DA ARTE..... | 11 |
| 2 DESENVOLVIMENTO..... | 15 |
| 2.1 ANÁLISE DO SISTEMA..... | 15 |
| 2.1.1 ANÁLISE DE REQUISITOS..... | 15 |
| 2.1.1.1 REQUISITOS FUNCIONAIS..... | 16 |
| 2.1.1.2 REQUISITOS NÃO FUNCIONAIS | 18 |
| 2.1.2 DIAGRAMA DE CASO DE USO | 19 |
| 2.1.3 FLUXO DE EVENTOS | 20 |
| 2.2 BANCO DE DADOS | 22 |
| 2.3 CAMADA DE NEGÓCIO..... | 23 |
| 2.4 CAMADA DE APRESENTAÇÃO | 27 |
| 2.4.1 DESIGN DAS PRINCIPAIS TELAS | 28 |
| 2.4.2 FRONT-END | 30 |
| 3 RESULTADO | 32 |
| 3.1 CONCLUSÃO..... | 33 |
| 3.2 PASSOS FUTUROS..... | 33 |
| REFERÊNCIAS..... | 35 |
| APÊNDICE A – DIAGRAMA DE CASO DE USO | 37 |
| APÊNDICE B – MODELO DE ENTIDADE E RELACIONAMENTO | 38 |
| APÊNDICE C – Questionário de Entrevista..... | 39 |

1 INTRODUÇÃO

Em decorrência do grande avanço de novas tecnologias na área da informática e da globalização, as empresas devem caminhar paralelamente com este mundo expressivo da tecnologia. O mundo empresarial está inserido diretamente dentro deste contexto em que os negócios visam lucro, propósitos sociais, e o retorno dos capitais no menor tempo possível.

Na atualidade, discorre o conceito de empreendedorismo como “qualquer tentativa de criação de um empreendimento como, por exemplo, uma atividade autônoma, uma nova empresa ou a expansão de um empreendimento” (N. Dioná Duarte Cardoso de Brito, 2016).

Atualmente o ambiente comercial, industrial, e de prestação de serviços é altamente competitivo entre si. Desta forma os gestores estão constantemente em busca do aperfeiçoamento dos recursos envolvidos em seus processos, tendo em vista deste modo a melhor qualificação e posição das empresas que administram. Os Sistemas de Informação são um grande trunfo e contribuem no processo de aperfeiçoamento empresarial. (CAETANO, 2013).

De modo geral, José Osvaldo De Sordi e Bernadete de Lourdes Marinho (2017) ressaltam que tais sistemas visam proporcionar, mediante a integração dos dados, resultados significativos em diversos setores de uma mesma empresa, a qual este sistema destina-se. Os controles informatizados traduzem-se em inúmeros benefícios aos meios empresariais, dando um amplo domínio sobre as transações da empresa. O mais evidente nesse processo é que na medida em que a tecnologia evoluiu, foram naturalmente surgindo dentro das empresas, diversificados sistemas com graus variáveis de complexidade possibilitando desta forma atender as suas demandas internas com maior eficácia. Com o propósito de informatizar tal empresa a baixo custo, em muitos casos, os preceitos básicos de um legítimo Sistema de Informação acabam sendo negligenciados. E, quando as condições técnicas são inviabilizadas de forma profissional, os resultados neste processo de informatização podem ser frustrante e não atingir os objetivos esperados até então.

Na era da digitalização, a gestão de negócios informatizados emerge como um pilar essencial para o sucesso empresarial. Como destacado por Maximiano (2011), a administração eficaz nesse contexto requer não apenas compreensão técnica dos

sistemas de informação, mas também habilidades de liderança para alinhar estratégias empresariais com as tecnologias disponíveis. A integração harmoniosa entre a tecnologia da informação e os processos de negócios torna-se crucial para aumentar a eficiência operacional, aprimorar a tomada de decisões e manter a competitividade no mercado atualmente dinâmico e globalizado.

Um programa de computador não é, necessariamente, um Sistema de Informação, como muitos pensam. Se a modelagem estrutural do banco de dados não for adequada, ou até mesmo, se as operações não reproduzirem à risca o mundo real em que a empresa vive, então é possível que ocorra uma ineficácia deste sistema, que nada mais é do que, um Sistema de Desinformação, ou seja, os resultados que deveriam ser obtidos pelo sistema não são alcançados, tornando-o inapropriado e muitas vezes inútil, pois não teve o devido planejamento para o seu desenvolvimento. (CARLOS, 2015).

Devido à grande burocracia que é se ter um sistema informatizado, muitos proprietários acabam não aderindo aos meios tecnológicos disponíveis no mercado, e com isso recorrem a meios um tanto quanto antigos, como planilhas manuais, anotações em cadernos, ou até mesmo criam inúmeras planilhas de *Excel*, que dependendo do quanto o negócio evoluir, este proprietário acabará se perdendo e desta maneira causará prejuízos para si mesmo.

Consequentemente a tudo que foi descrito anteriormente surgiu a ideia do SIG OS, uma aplicação *web* que serve para amparar e auxiliar pequenas e médias empresas do ramo comercial e de prestação de serviço a potencializar e otimizar os processos de seus negócios, sendo uma solução completa de gerenciamento empresarial, com o enfoque em ordens de serviço.

1.1 OBJETIVO

Esta seção explicita os objetivos que foram traçados para o projeto separando-os em duas etapas, inicialmente há a delimitação de um objetivo geral, e em seguida tem os objetivos específicos, que são estágios tratados antes de concluir um objetivo geral.

1.1.1 OBJETIVO GERAL

A sistematização diz respeito ao alinhamento de uma estrutura baseado em uma regra ou um parâmetro para atingir um objetivo. Portanto, conduzir o desenvolvimento de um sistema que possua a estratégia para gerir um negócio de forma concisa, no qual o usuário, no caso, um empreendedor, tenha de maneira fácil e objetiva uma ferramenta capaz de otimizar e ajudar nas demandas essenciais do seu empreendimento.

A simplicidade é a diretriz que irá conduzir o SIG OS, ou seja, para que o empreendedor de posse do sistema, tenha maior segurança na gestão e concentre em uma só ferramenta, tudo o que é necessário para realizar as demandas que são inerentes ao seu negócio. Com base nisso, a proposta do SIG OS é uma aplicação web de gestão de ordem de serviço, auxiliando de forma completa todo o processo do ciclo de negócio empresarial.

1.1.2 OBJETIVOS ESPECÍFICOS

- Realizar estudo sobre o processo de Ordem de Serviço: Condicionantes, Modelos empresariais, Composição de Custos, Variáveis
- Compreender o conceito de um sistema de OS, para ser compactuante com as funcionalidades presentes no sistema
- Compreender como são cobrados gastos administrativos (custos fixos) de uma Ordem de Serviço para uma prestadora de serviço
- Estudar o básico sobre como gerar uma ordem de serviço do zero
- Entender de maneira mais aprofundada a utilizado do framework Codeigniter

1.2 ESTADO DA ARTE

Segundo Heloiza da Silva (2023), gestão de ordem de serviço nas empresas esbarra em uma série de desafios: erros humanos, falta de tecnologia, criação de relatórios em papel e dificuldades no armazenamento e acompanhamento de dados gerados a partir dos relatórios, as OS's são tarefas e/ou trabalhos para um cliente, que está contida em empresas do ramo comercial e em prestadoras de serviços na qual visa documentar serviços prestados, produtos vendidos, discriminando de maneira formal, tudo o que é realizado, dentro de um determinado estabelecimento.

Uma pesquisa de Luiz Carlos Begosso (2019) destaca que a gestão eficaz de ordens de serviço não se limita apenas à alocação de tarefas. Ela abrange a proteção

de informações confidenciais, como dados do cliente e históricos de serviço, contra ameaças cibernéticas em um ambiente cada vez mais digital. Esse aspecto se torna particularmente crucial à medida que mais dados e informações sensíveis são mantidos eletronicamente.

De acordo com Ribeiro (2020), a automatização é um tema amplamente abordado, onde sistemas de gestão automatizados têm a capacidade de simplificar tarefas como agendamento, atribuição de técnicas e rastreamento em tempo real. A implementação dessas ferramentas resulta em uma redução significativa de erros, melhorando assim a eficiência operacional.

A análise de dados tem se tornado uma prática comum na gestão de ordens de serviço, bem como a coleta e análise de dados detalhados permitem às empresas avaliarem o desempenho, identificar áreas de melhoria e tomar decisões informadas, cuja a satisfação do cliente é, naturalmente, um foco central nesse contexto, destacando que a implementação de sistemas de *feedback* em tempo real, notificações proativas e agendamento conveniente pode aumentar significativamente a satisfação do cliente, resultando em maior fidelização e crescimento do negócio, segundo Jacksson Ozorio (2023).

Francisco Paulo (2009) ressalta que olhando para o futuro da gestão de ordens de serviço apresenta tendências promissoras e inovações em suas pesquisas, exploram o potencial da inteligência artificial na automação de programação de tarefas e o uso da realidade aumentada como uma ferramenta para suporte remoto e treinamento de técnicos.

Olhando para o futuro, a gestão de ordens de serviço apresenta tendências promissoras e inovações. Pesquisas, como as conduzidas por Lee et al. (2020) e Wang et al. (2021), exploram o potencial da inteligência artificial na automação de programação de tarefas e o uso da realidade aumentada como uma ferramenta para suporte remoto e treinamento de técnicos.

Acima da conceitualização, é preciso visualizar interfaces que tentam usar funções de gestão de ordem de serviço, um exemplo é a plataforma web EOS-OS, Ilustração 1, na qual é concebível inserir as OS's dos clientes, cadastrar os produtos que serão necessários para realizar a ordem de serviço e suas respectivas informações adicionais. Todos estes dados são salvos para serem usados na criação e geração da OS ao final da inserção dos dados. Porém a plataforma tem avaliações negativas quanto a instabilidade do sistema.

Ilustração 01 - Visualização de Ordens de Serviço do EOS-OS

| N° O.S. | Vinculada a | Tags | Produto | Serial | Cliente | Garantia | Data entrada | Valor | Situação |
|---------|--------------|------|----------------------|-------------------------|------------------|----------|--------------|--------------|---|
| 722 | | | Teste | 123456 - 08/2017 | Cliente Fictício | 08h47 | 02/04/2019 | R\$ 0,00 | Orçamento pronto |
| 721 | teste.teste2 | | Teste | 123456789 - 01/2019 | Cliente Fictício | 08h44 | 02/04/2019 | R\$ 1.286,47 | Aguardando |
| 720 | | | Produto Marca Demo 6 | 123456 - 10/2018 | Cliente Demo 1 | 18h39 | 18/01/2019 | R\$ 0,00 | Aguardando |
| 719 | | | Fonte 150A Demo 5 | 123 - 10/2017 | Cliente Teste | 15h08 | 27/11/2018 | R\$ 60,44 | Aguardando |
| 718 | | | Teste | 0517003200597 - 10/2018 | Cliente Teste | 14h36 | 31/10/2018 | R\$ 0,00 | Aguardando |
| 717 | | | Fonte Demo 1 | 123 - 10/2018 | Cliente Demo 1 | 21h36 | 17/10/2018 | R\$ 57,00 | Orçamento enviado (pendente de aprovação) |
| 7 | EOS Studio | | Produto 10 | 101003377428 - 10/2018 | Cliente Teste | 20h22 | 03/10/2018 | R\$ 0,00 | Aguardando |
| 716 | | | Produto Marca Demo 6 | S/N - 12/2000 | Cliente Teste | 20h22 | 03/10/2018 | R\$ 0,00 | Aguardando |
| 6 | EOS Studio | | Produto 3 | 1234567 - 07/2018 | Cliente Demo 3 | 02h18 | 01/10/2018 | R\$ 0,00 | Aguardando |

Fonte: <https://eos-os.com.br>

Outro ponto de referência foi OnlineOS, Ilustração 2, com uma interface bem limpa, intuitiva e leve, recebe bons feedbacks, possuindo um ranqueamento alto, e com quase todas as funcionalidades do EOS-OS, apenas faltando a ferramenta de exportação dos históricos das OS's, e disponibilização de imagem (disponível na versão PRO).

Ilustração 02 - Visualização de Ordens de Serviço do OnlineOS

| N. | Cliente | Status | Colaborador | Criado Em | Valor |
|------|------------|----------------------|---------------------|------------------|--------------------|
| 0002 | Cliente 02 | EM ATENDIMENTO | Colaborador 04 | 27/09/2021 09:39 | R\$ 200,00(A Page) |
| 0001 | Cliente 01 | FINALIZADO | Tutoriais Online OS | 27/09/2021 09:38 | R\$ 800,00(Page) |
| 0003 | Cliente 03 | Aguardando APROVAÇÃO | Tutoriais Online OS | 27/09/2021 09:39 | R\$ 200,00(A Page) |
| 0004 | Cliente 04 | APROVADO | Tutoriais Online OS | 27/09/2021 09:40 | R\$ 400,00(A Page) |
| 0005 | Cliente 05 | REPROVADO | Tutoriais Online OS | 27/09/2021 09:40 | R\$ 500,00(A Page) |

Fonte: <https://onlineos.com.br>

E por último, outra base de referência foi o SistemaSOMA, Ilustração 3, com uma interface mais minimalista e focada na gestão interna da empresa em que ele está implantado, possui menos funcionalidades comparado com as duas outras referências (OnlineOS e EOS-OS) apresentadas neste documento, porém possui um bom feedback dos seus usuários.

Ilustração 03 - Visualização de Ordens de Serviço do SistemaSOMA

SOMA Ordem de Serviço

Ordens de Serviço

01/01/2018 31/01/2018 ☐ A Fazer ☐ Pendente ☐ Feito ☐ Em Espera ☐ Cancelado

| STATUS | DATA | DESCRIÇÃO | PRODUTOS / SERVIÇOS | RESPONSÁVEL | CLIENTE | VALOR | OPÇÕES |
|---------------------------|------------|--|---------------------|-----------------|---------------------------|---------------------------|--|
| Em andamento | 22/01/2018 | 1,Jun de Instalação de Câmera, 6,Jun de Câmeras Monitoramento | | João Carlos | São de Camo | R\$ 1025,00 / R\$ 1025,00 | <input type="button" value="Cancelar"/> <input type="button" value="Alterar Status"/> |
| Pendente | 14/01/2018 | 2,Jun de HD 4TB, 2,Jun de Instalação de HD de Armazenagem, 1,Jun de Backup-Servidor Surveillance | | Roberto Pereira | Escadonamento Solto | R\$ 1220,00 / R\$ 1220,00 | <input type="button" value="Cancelar"/> <input type="button" value="Alterar Status"/> |
| Em espera | 13/01/2018 | 6,Jun de Atualização GPS de Monitoramento | | Roberto Pereira | Transportadora Novo Mundo | R\$ 0,00 / R\$ 330,00 | <input type="button" value="Cancelar"/> <input type="button" value="Alterar Status"/> |
| Feito | 11/01/2018 | 1,Jun de Backup-Servidor Surveillance | | Roberto Pereira | Profisara | R\$ 0,00 / R\$ 80,00 | <input type="button" value="Cancelar"/> <input type="button" value="Alterar Status"/> |
| Feito | 13/01/2018 | 1,Jun de Instalação PABX, 1,Jun de PABX | | João Carlos | Agência Digital | R\$ 670,00 / R\$ 670,00 | <input type="button" value="Cancelar"/> <input type="button" value="Alterar Status"/> |
| Cancelado | 02/01/2018 | 1,Jun de Backup-Servidor Surveillance | | Roberto Pereira | Supermercado Central | R\$ 0,00 / R\$ 80,00 | <input type="button" value="Cancelar"/> <input type="button" value="Alterar Status"/> |

Mostrando de 1 até 6 de 6 registros

Anterior Próxima

Fonte: <https://sistemasoma.com.br>

2 DESENVOLVIMENTO

Esta seção consiste em como o sistema foi desenvolvido, detalhando técnicas e recursos dos quais compreendem desde a fase do escopo até o desenvolvimento do sistema. Para tanto, esse capítulo será subdividido nos seguintes itens; análise do sistema, a estrutura do Banco de Dados, as ferramentas utilizadas para o desenvolvimento do *front-end* e as principais rotinas implementadas no *back-end*.

2.1 ANÁLISE DO SISTEMA

Pautando as referências pesquisadas, foram expostas às funções que uma aplicação de gestão para ordem de serviço deve possuir, desse modo as vantagens e desvantagens das plataformas online já existentes nortearam a segmentar uma análise de sistema mais concisa. Então essa seção será mostrada de qual forma que eles devem ser adaptados para trazer uma experiência mais simples e lucida para os usuários marcados.

2.1.1 ANÁLISE DE REQUISITOS

A esquematização de um sistema decorre de diversas etapas, contudo a falta de uma organização e padronização acarreta um exercício muito mais árduo quanto a esse desenvolvimento. Por isso, para projetos que utilizam a orientação a objetos e transcorrem fases como análise de requisitos, análise de sistemas e *design* foi criado a notação *UML* (*Unified Modeling Language*).

A *UML* possibilita construir modelos concisos, precisos, completos e sem ambiguidades, tendo, de maneira geral, as seguintes características: 1 - Modela os aspectos estruturais (estáticos) e comportamentais (dinâmicos) do sistema, ou seja, pode especificar os conceitos do negócio e seus relacionamentos (invariantes com o tempo) e os estados, sequências de atividades e de colaborações (aspectos que contemplam a dimensão temporal, ou seja, que variam conforme o tempo passa). Em outras palavras, a *UML* provê elementos de notação para modelar dados, funções de transformação dos dados e as restrições aplicáveis aos dados e às funções, como regras de negócio, por exemplo. Essas características são necessárias, como já dissemos, à produção de bons modelos. 2 - Provê uma linguagem que permite o entendimento e a utilização por humanos e a leitura por máquinas. Além dos elementos gráficos da notação que são compreensíveis aos humanos (desde que, obviamente, conheçam a linguagem). (Luiz Antônio Pereira de Moraes, 2011, P.21)

Tal padronização recebe elementos e conceitos essenciais para a sua aplicabilidade, tendo início no levantamento de requisitos que são bases para as ações e características que o programa deve seguir, e assim também registrando necessidades do cliente. Há dois tipos de requisitos que compõe um sistema, os requisitos funcionais e não funcionais que serão apresentados na seção a seguir:

2.1.1.1 REQUISITOS FUNCIONAIS

Os requisitos funcionais ganharam importância fundamental no processo de desenvolvimento de *software*, pois quando definidos de forma clara e objetiva exercessem um papel de facilitador para o desenvolvedor. Eles projetam de maneira simples as funções que o sistema deve ter para ser solucionado algum problema do cliente final.

Segundo o acadêmico britânico Ian Sommerville, os requisitos funcionais são importantes elementos textuais dos quais descrevem as funcionalidades que cada tipo de usuário deve ser capaz de acessar no sistema (2011). Baseado nisso, já possuindo o reconhecimento das funcionalidades importantes para o objetivo do programa, foram elencados os requisitos funcionais a seguir:

Tela Clientes:

[RF01] – Cadastrar cliente – O sistema deve permitir que o funcionário cadastre clientes;

[RF02] – Editar cliente – O sistema deve permitir que o funcionário edite os dados cadastrais dos clientes;

[RF03] – Visualizar cliente – O sistema deve permitir que o funcionário visualize os dados cadastrais dos clientes;

[RF04] – Deletar cliente – O sistema deve permitir que o funcionário delete os clientes;

Tela Produtos:

[RF05] – Cadastrar produto – O sistema deve permitir que o funcionário cadastre um produto;

[RF06] – Editar produto – O sistema deve permitir que o funcionário edite as informações do produto;

[RF07] – Visualizar produto – O sistema deve permitir que o funcionário visualize as informações do produto;

[RF08] – Deletar produto – O sistema deve permitir que o funcionário delete o produto;

Tela Serviços:

[RF09] – Cadastrar serviço – O sistema deve permitir que o usuário cadastre um serviço;

[RF10] – Editar serviço – O sistema deve permitir que o usuário edite as informações do serviço;

[RF11] – Deletar produto – O sistema deve permitir que o funcionário delete o produto;

Tela Vendas:

[RF12] – Cadastrar Venda – O sistema deve permitir que o funcionário cadastre uma venda;

[RF13] – Editar Venda – O sistema deve permitir que o funcionário edite as informações da venda;

[RF14] – Visualizar venda – O sistema deve permitir que o funcionário visualize as informações da venda;

[RF15] – Deletar venda – O sistema deve permitir que o funcionário delete a venda;

Tela Ordens de Serviços:

[RF16] – Cadastrar Ordem de Serviço – O sistema deve permitir que o funcionário cadastre uma OS;

[RF17] – Editar Ordem de Serviço – O sistema deve permitir que o funcionário edite as informações da OS;

[RF18] – Deletar Ordem de Serviço – O sistema deve permitir que o funcionário delete a OS;

Tela Garantia:

[RF19] – Cadastrar Garantia – O sistema deve permitir que o funcionário cadastre uma garantia;

[RF20] – Editar Garantia – O sistema deve permitir que o funcionário edite as informações da garantia;

[RF21] – Deletar Garantia – O sistema deve permitir que o funcionário delete a garantia;

Tela Arquivo:

[RF22] – Cadastrar Arquivo – O sistema deve permitir que o usuário cadastre um arquivo;

[RF23] – Editar Arquivo – O sistema deve permitir que o usuário edite o arquivo;

[RF24] – Deletar Arquivo – O sistema deve permitir que o usuário delete o arquivo;

Tela Lançamentos Financeiros:

[RF25] – Cadastrar Lançamento Financeiro – O sistema deve permitir que o usuário cadastre um lançamento financeiro;

[RF26] – Editar Lançamento Financeiro – O sistema deve permitir que o usuário edite um lançamento financeiro;

[RF27] – Deletar Lançamento Financeiro – O sistema deve permitir que o usuário delete um lançamento financeiro;

Tela Usuários:

[RF28] – Cadastrar Usuário – O sistema deve permitir que o administrador cadastre um funcionário;

[RF29] – Editar Usuário – O sistema deve permitir que o administrador edite as informações do funcionário;

[RF30] – Deletar Usuário – O sistema deve permitir que o administrador delete um funcionário;

Tela Área do Cliente:

[RF31] – Visualizar Ordem de Serviço – O cliente poderá visualizar as informações da ordem de serviço que foi aberta no estabelecimento;

[RF32] – Visualizar Compras – O cliente poderá visualizar as informações de compras que tenham sido realizadas por ele no estabelecimento;

[RF33] – Visualizar Minha Conta – O cliente poderá visualizar as informações/dados dele que foram cadastradas no estabelecimento;

2.1.1.2 REQUISITOS NÃO FUNCIONAIS

O minimalismo fica por conta dos requisitos não funcionais, pois se os requisitos funcionais, esboçam as funções do sistema, os não funcionais trata-se de recursos que descrevem como o sistema fará essas funções, determinando suas características e limites.

Os usuários do sistema podem, geralmente, encontrar maneiras de contornar uma função do sistema que realmente não atenda a suas necessidades. No entanto, deixar de atender a um requisito não funcional pode significar a inutilização de todo o sistema. Por exemplo, se um sistema de aeronaves não cumprir seus requisitos de confiabilidade, não será certificado como um sistema seguro para operar; se um sistema de controle embutido não atender aos requisitos de desempenho, as funções de controle não funcionarão corretamente (SOMMERVILLE, 2011, p. 76).

Com tamanha relevância de um sistema, os requisitos não funcionais do SIG OS foram definidos com base em satisfazer atributos como segurança, qualidade e ambiente. Sendo possível visualizá-los abaixo:

[RNF01] – A base de dados deverá ser protegida para acesso somente a usuários autorizados e devidamente autenticados.

[RNF02] – Utilização do sistema em rede local ou *web* (internet).

[RNF03] – *Design* responsivo nas interfaces gráficas.

[RNF04] – Compatibilidade com sistemas operacionais *Windows* e *Linux*.

2.1.2 DIAGRAMA DE CASO DE USO

No contexto da linguagem *UML* apesar da textualização dos requisitos serem essenciais, eles não podem servir como parâmetro único, pelo fato de abrir brechas para ambiguidade, e prejudicar a proposta final da aplicação. Logo para evadir mais obstáculos a idealização também tende de ser visualizada de modo visual, para tal ocorre a elaboração dos requisitos funcionais por intermédio do diagrama de casos de uso.

Segundo Alberto Silva e Carlos Videira, o diagrama de caso de uso permite uma ótica do usuário desempenhando tarefas com o *software*, ele passa pela identificação dos atores e por estruturar os requisitos em casos lógicos (2001). Nele serão retratados essa comunicação entre atores externos com o sistema, nesse caso o ator, será o administrador e o funcionário.

O diagrama de caso de uso desse sistema pode ser visualizado no Apêndice A desta monografia.

2.1.3 FLUXO DE EVENTOS

Na *UML*, as atividades são elementos de contêiner que descrevem o nível mais alto do comportamento em um diagrama de atividades. As atividades contêm diversos nós de atividades e linhas de atividades que representam a sequência de tarefas em um fluxo de trabalho resultando em um comportamento (*IBM*, 2021). Desse modo, aparado a relação que a gente tem com a aplicação, faz se necessário desmembrar suas atividades, apresentadas no diagrama de uso, para gerir e captar seu comportamento, e assim determinar de modo mais eficiente as condições de entrada e saída para que isso ocorra.

Pode determinar um fluxo básico como a função principal de um caso de uso, detalhando o momento em que nenhum desvio é tomado. Já os fluxos alternativos descrevem desvios pré-definidos do fluxo básico. Então, com essas funções relatadas, serão aprofundadas o fluxo dos principais requisitos funcionais a seguir.

Ilustração 04 - Adicionar Ordem de Serviço

| Ator: | Funcionário |
|-------------------------------|--|
| Fluxo de eventos principal: | 1- O funcionário clica na aba Ordens 2- O funcionário insere informações como nome do cliente, técnico ou responsável, status, data inicial e data final para inserção da Ordem de Serviço 4- O funcionário é direcionado para adicionar as informações pertinentes a OS, e adiciona os produtos e serviços que serão executados 5- O sistema armazena as informações |
| Fluxo de eventos alternativo: | 3.a – Sem conteúdo no campo, se nenhum campo obrigatório for preenchido na tela inicial de abertura de OS, o sistema não deixará prosseguir. |

Fonte: Autores (2024)

Na Ilustração 04 – Adicionar Ordem de Serviço, é retratado o processo de fluxo principal que o funcionário executará ao realizar a inclusão de uma OS. Bem como igualmente é descrito o processo de fluxo alternativo, caso algo durante o fluxo principal para adicionar uma ordem de serviço de errado.

Ilustração 05 - Adicionar Produto

| Ator: | Funcionário |
|-------------------------------|--|
| Fluxo de eventos principal: | 1- O funcionário clica na aba Produtos 2- O funcionário insere informações como descrição do produto, tipo de movimento, preço de compra, preço de venda, unidade, estoque e estoque mínimo para inserção do produto 3- O funcionário clica em adicionar 4- O sistema armazena as informações |
| Fluxo de eventos alternativo: | 3.a – Sem conteúdo no campo, se nenhum campo obrigatório for preenchido na tela de adicionar produto, o sistema não deixará prosseguir. |

Fonte: Autores (2024)

Na Ilustração 05 – Adicionar Produto, é retratado o processo de fluxo principal que o funcionário executará ao realizar a inclusão de um produto. Bem como igualmente é descrito o processo de fluxo alternativo, caso algo durante o fluxo principal para adicionar um produto de errado.

Ilustração 06 - Adicionar Serviço

| Ator: | Funcionário |
|-------|-------------|
|-------|-------------|

| | |
|-------------------------------|---|
| Fluxo de eventos principal: | 1- O funcionário clica na aba Serviços 2- O funcionário insere informações como nome do serviço, valor do serviço e descrição do serviço para inserção do serviço 3- O funcionário clica em adicionar 4- O sistema armazena as informações |
| Fluxo de eventos alternativo: | 3.a – Sem conteúdo no campo, se nenhum campo obrigatório for preenchido na tela de adicionar serviço, o sistema não deixará prosseguir. |

Fonte: Autores (2024)

Na Ilustração 06 – Adicionar Serviço, é retratado o processo de fluxo principal que o funcionário executará ao realizar a inclusão de um serviço. Bem como igualmente é descrito o processo de fluxo alternativo, caso algo durante o fluxo principal para adicionar um serviço de errado.

2.2 BANCO DE DADOS

Expressado o contato do usuário com o sistema e suas correspondências quanto a esses atos, a seguinte etapa do processo de desenvolvimento de um software gira em torno a exploração da estrutura dos dados. Ou seja, observar e identificar as entidades, relacionamentos, atributos, chaves etc.

A modelagem de dados é viabilizada por modelos, destes o modelo físico é provido através do MER (Modelo Entidade-Relacionamento) “MER é um modelo de alto nível, independente do SGBD (Sistemas Gerenciadores de Banco de dados), que representa o problema a ser modelado” (Araújo, M.A.P, 2008). A modelagem física é dita, segundo o analista Carlos Alberto Dornelles como um estágio mais avançado da modelagem lógica, pois já utiliza alguma tecnologia de banco de dados, porém considerando os requisitos funcionais e não funcionais.

Um banco de dados bem elaborado desempenha um papel fundamental no processo de desenvolvimento de sistemas, servindo como o alicerce sobre o qual toda a aplicação é construída. Ele não apenas armazena dados de forma organizada, mas

também fornece uma estrutura flexível para manipulação e recuperação eficiente de informações.

Como ressalta Claudio Leonis Bazzi (2013), um banco de dados adequado permite a integração de diferentes componentes do sistema, facilitando a manutenção, garantindo a consistência dos dados e promovendo a escalabilidade da aplicação. Além disso, um projeto de banco de dados bem planejado contribui para a segurança dos dados, a redução de redundâncias e a melhoria da performance do sistema como um todo.

O MER (Modelo de Entidade e Relacionamento) desse sistema pode ser visualizado no Apêndice B desta monografia.

2.3 CAMADA DE NEGÓCIO

De modo a buscar uma maior escalabilidade e aplicabilidade para criação desta aplicação *web* foi utilizada a linguagem de programação *PHP*, juntamente com o *framework Codeigniter*.

A linguagem de programação *PHP*, acrônimo para "Hypertext Preprocessor," é uma linguagem de script amplamente utilizada para desenvolvimento *web* dinâmico. Criada por Rasmus Lerdorf em 1995, o *PHP* é uma linguagem de código aberto que se destaca pela sua facilidade de uso e integração com *HTML*. De acordo com o livro "*PHP and MySQL Web Development*" de Luke Welling e Laura Thomson (2008), *PHP* permite a criação de páginas *web* interativas, dinâmicas e personalizadas, sendo altamente popular no desenvolvimento de sistemas *web*.

O *CodeIgniter* é um popular *framework* de desenvolvimento *web* em *PHP* amplamente utilizado devido à sua simplicidade e eficiência. Este *framework* segue o padrão de projeto Modelo-Visão-Controlador (MVC), permitindo uma clara separação das camadas de aplicação. *CodeIgniter* é reconhecido por sua arquitetura leve e ágil, o que o torna uma escolha preferida para projetos *web* de pequeno a médio porte. De acordo com uma pesquisa de Luqman Hakim e Muhammad Afnan Surya, publicada no "*International Journal of Computer Applications*" em 2017, o *CodeIgniter* oferece uma estrutura bem definida para o desenvolvimento ágil de aplicações *web*, simplificando tarefas comuns e reduzindo o tempo de desenvolvimento.

Uma linguagem robusta, com anos de suporte e adotada largamente pelo mercado de desenvolvimento de *software* profissional é o *PHP*. O *PHP* é uma linguagem de programação moderna, robusta e orientada a objetos que está em constante evolução e adoção de profissionais no mundo inteiro. Com essa

linguagem de programação é possível criar *softwares* para muitos tipos de aplicativos tais como sites, aplicações *web*, sistemas de gerenciamento de conteúdo e muito mais (*PHP para iniciantes*, 2023, P. 20).

A seguir será apresentada e brevemente explicada as classes que compõem o *CRUD* de execução deste projeto, presentes na pasta *Model* desta aplicação. A entidade OS, possui classes que são diretamente responsáveis pelo processamento das ordens de serviço. Nesta entidade, são recebidos os dados, como: data inicial, data final, descrição do produto, defeito, *status*, observações, laudo técnico, valor total, desconto, tipo de desconto, nome do cliente e usuário responsável. Após inserção de todas as informações de OS, elas são processadas e armazenadas no banco de dados da aplicação.

Ilustração 07 - Model OS Adicionar

```
public function adicionar()
{
    if (!$this->permission->checkPermission($this->session->userdata('permissao'), 'aOs')) {
        $this->session->set_flashdata('error', 'Você não tem permissão para adicionar O.S.');
```

```
        redirect(base_url());
    }
    $this->load->library('form_validation');
    $this->data['custom_error'] = '';
    if ($this->form_validation->run('os') == false) {
        $this->data['custom_error'] = (validation_errors() ? true : false);
    } else {
        $dataInicial = $this->input->post('dataInicial');
        $dataFinal = $this->input->post('dataFinal');
        $termoGarantiaId = $this->input->post('termoGarantia');
        try {
            $dataInicial = explode('/', $dataInicial);
            $dataInicial = $dataInicial[2] . '-' . $dataInicial[1] . '-' . $dataInicial[0];

            if ($dataFinal) {
                $dataFinal = explode('/', $dataFinal);
                $dataFinal = $dataFinal[2] . '-' . $dataFinal[1] . '-' . $dataFinal[0];
            } else {
                $dataFinal = date('Y/m/d');
            }
            $termoGarantiaId = (! $termoGarantiaId == null || ! $termoGarantiaId == '')
                ? $this->input->post('garantias_id')
                : null;
        } catch (Exception $e) {
            $dataInicial = date('Y/m/d');
            $dataFinal = date('Y/m/d');
        }
        $data = [
            'dataInicial' => $dataInicial,
            'clientes_id' => $this->input->post('clientes_id'), //set_value('idCliente'),
            'usuarios_id' => $this->input->post('usuarios_id'), //set_value('idUser'),
            'dataFinal' => $dataFinal,
            'garantia' => set_value('garantia'),
            'garantias_id' => $termoGarantiaId,
            'descricaoProduto' => set_value('descricaoProduto'),
            'defeito' => set_value('defeito'),
            'status' => set_value('status'),
            'observacoes' => set_value('observacoes'),
            'laudoTecnico' => set_value('laudoTecnico'),
            'faturado' => 0,
        ];
    }
}
```

Fonte: Autores (2024)

Na ilustração 07 – Modelo OS Adicionar é realizada a inserção de uma OS, parte fundamental para o funcionamento adequado de todo o sistema, está sendo verificada a permissão de usuário do sistema para que seja adicionada uma ordem de serviço. Caso o usuário não possua permissão suficiente, ele será redirecionado para a página inicial, informando via mensagem que não tem permissão suficiente. Se o usuário estiver com a permissão correta, são carregadas as bibliotecas de validação de formulário e armazenagem de possíveis erros personalizados, para caso em algum momento a validação do formulário falhar, se isso não ocorrer os dados são processados. Os dados inseridos neste formulário de inclusão de OS, são organizados em um *array* associativo, após finalizado, a OS estará pronta e adicionada.

Ilustração 08 - Model OS Editar

```
public function editar()
{
    if (!($this->uri->segment(3) || is_numeric($this->uri->segment(3))) {
        $this->session->set_flashdata('error', 'Item não pode ser encontrado, parâmetro não foi passado corretamente.');
```

```
        redirect('mapos');
    }
    if (!($this->permission->checkPermission($this->session->userdata('permissao'), 'eOs')) {
        $this->session->set_flashdata('error', 'Você não tem permissão para editar OS.');
```

```
        redirect(base_url());
    }
    $this->load->library('form_validation');
    $this->data['custom_error'] = '';
    $this->data['texto_de_notificacao'] = $this->data['configuration']['notifica_whats'];

    $this->data['editavel'] = $this->os_model->isEditable($this->input->post('idOs'));
    if (!($this->data['editavel'])) {
        $this->session->set_flashdata('error', 'Esta OS já e seu status não pode ser alterado e nem suas informações atualizadas. Por favor abrir uma nova OS.');
```

```
        redirect(site_url('os'));
    }
    if ($this->form_validation->run('os') == false) {
        $this->data['custom_error'] = (validation_errors() ? '<div class="form_error">' . validation_errors() . '</div>' : false);
    } else {
        $dataInicial = $this->input->post('dataInicial');
        $dataFinal = $this->input->post('dataFinal');
        $termoGarantiaId = $this->input->post('garantias_id') ?: null;
        try {
            $dataInicial = explode('/', $dataInicial);
            $dataInicial = $dataInicial[2] . '-' . $dataInicial[1] . '-' . $dataInicial[0];
            $dataFinal = explode('/', $dataFinal);
            $dataFinal = $dataFinal[2] . '-' . $dataFinal[1] . '-' . $dataFinal[0];
        } catch (Exception $e) {
            $dataInicial = date('Y/m/d');
        }
        $data = [
            'dataInicial' => $dataInicial,
            'dataFinal' => $dataFinal,
            'garantia' => $this->input->post('garantia'),
            'garantias_id' => $termoGarantiaId,
            'descricaoProduto' => $this->input->post('descricaoProduto'),
            'defeito' => $this->input->post('defeito'),
            'status' => $this->input->post('status'),
            'observacoes' => $this->input->post('observacoes'),
            'laudoTecnico' => $this->input->post('laudoTecnico'),
            'usuarios_id' => $this->input->post('usuarios_id'),
            'clientes_id' => $this->input->post('clientes_id'),
        ];
        $os = $this->os_model->getId($this->input->post('idOs'));
    }
}
```

Fonte: Autores (2024)

Na ilustração 08 – *Model OS Editar*, é onde ocorre a edição de uma OS, segue o mesmo nível de importância do código de adição de OS. É verificado se um

parâmetro numérico é passado corretamente na *URL* para identificar o item a ser editado, em seguida é feita a conferência de usuário, se o usuário possui privilégios para essa ação de editar uma OS, após isso, são carregadas as bibliotecas de validação, é obtido informações sobre se a ordem de serviço é editável ou não, com base em sua condição atual, caso a ordem de serviço não seja editável, uma mensagem será mostrada ao usuário e redirecionará ele. Os dados são processados e organizados em um *array* associativo para a atualização de OS.

Ilustração 09 - Model OS Visualizar

```
public function visualizar()
{
    if (!$this->uri->segment(3) || !is_numeric($this->uri->segment(3))) {
        $this->session->set_flashdata('error', 'Item não pode ser encontrado, parâmetro não foi passado corretamente.');
```

```
        redirect('mapos');
```

```
    }

    if (!$this->permission->checkPermission($this->session->userdata('permissao'), 'vOs')) {
        $this->session->set_flashdata('error', 'Você não tem permissão para visualizar O.S.');
```

```
        redirect(base_url());
    }

    $this->data['custom_error'] = '';
    $this->data['texto_de_notificacao'] = $this->data['configuration']['notifica_whats'];

    $this->load->model('mapos_model');
    $this->data['result'] = $this->os_model->getId($this->uri->segment(3));
    $this->data['produtos'] = $this->os_model->getProdutos($this->uri->segment(3));
    $this->data['servicos'] = $this->os_model->getServicos($this->uri->segment(3));
    $this->data['emitente'] = $this->mapos_model->getEmitente();
    $this->data['anexos'] = $this->os_model->getAnexos($this->uri->segment(3));
    $this->data['anotacoes'] = $this->os_model->getAnotacoes($this->uri->segment(3));
    $this->data['editavel'] = $this->os_model->isEditable($this->uri->segment(3));
    $this->data['qrCode'] = $this->os_model->getQrCode(
        $this->uri->segment(3),
        $this->data['configuration']['pix_key'],
        $this->data['emitente']
    );
    $this->data['modalGerarPagamento'] = $this->load->view(
        'cobrancas/modalGerarPagamento',
        [
            'id' => $this->uri->segment(3),
            'tipo' => 'os',
        ],
        true
    );
    $this->data['view'] = 'os/visualizarOs';

    if ($return = $this->os_model->valorTotalOS($this->uri->segment(3))) {
        $this->data['totalServico'] = $return['totalServico'];
        $this->data['totalProdutos'] = $return['totalProdutos'];
    }

    return $this->layout();
}
```

Fonte: Autores (2024)

Na ilustração 09 – *Model OS Visualizar* código abaixo, refere-se à visualização de OS, após as verificações de permissão de usuário, e de possíveis erros, é carregado o “*mapos_model*”, que recupera as informações específicas da OS para que sejam visualizadas, como os detalhes de produtos e serviços incluídos na OS. É

obtida as informações do emitente de OS, verificado se a ordem de serviço é editável. É então definida a visualização da página como a *view* “os/visualizarOs”, e retorna a renderização da página através do layout definido.

Ilustração 10 - Model OS Excluir

```
public function excluir()
{
    if (!$this->permission->checkPermission($this->session->userdata('permissao'), 'dOs')) {
        $this->session->set_flashdata('error', 'Você não tem permissão para excluir O.S.');
```

```
        redirect(base_url());
    }

    $id = $this->input->post('id');
    $os = $this->os_model->getIdCobrancas($id);
    if ($os == null) {
        $os = $this->os_model->getId($id);
        if ($os == null) {
            $this->session->set_flashdata('error', 'Erro ao tentar excluir OS.');
```

```
            redirect(base_url() . 'index.php/os/gerenciar/');
        }
    }

    if (isset($os->idCobranca) != null) {
        if ($os->status == "cancelado") {
            $this->os_model->delete('cobrancas', 'os_id', $id);
        } else {
            $this->session->set_flashdata('error', 'Existe uma cobrança associada a esta OS, deve cancelar e/ou excluir a cobrança primeiro!');
```

```
            redirect(site_url('os/gerenciar/'));
        }
    }

    $osStockRefund = $this->os_model->getId($id);
    //Verifica para poder fazer a devolução do produto para o estoque caso OS seja excluída.
    if (strtolower($osStockRefund->status) != "cancelado") {
        $this->devolucaoEstoque($id);
    }

    $this->os_model->delete('servicos_os', 'os_id', $id);
    $this->os_model->delete('produtos_os', 'os_id', $id);
    $this->os_model->delete('anexos', 'os_id', $id);
    $this->os_model->delete('os', 'idOs', $id);
    if ((int)$os->faturado == 1) {
        $this->os_model->delete('lancamentos', 'descricao', "Fatura de OS - #{$id}");
    }

    log_info('Removeu uma OS. ID: ' . $id);
    $this->session->set_flashdata('success', 'OS excluída com sucesso!');
```

```
    redirect(site_url('os/gerenciar/'));
}
```

Fonte: Autores (2024)

Na ilustração – 10 *Model OS Excluir*, refere-se a exclusão de uma OS. Após as verificações de permissão de usuário, para saber se ele tem privilégio para realizar tal exclusão, é obtido o *ID* da ordem de serviço a ser excluída a partir dos dados recebidos via método *POST*, registra a ação de exclusão no *log* do sistema, define uma mensagem de sucesso para o usuário, e redireciona o usuário de volta à página de gerenciamento de OS após a exclusão ter sido bem-sucedida.

2.4 CAMADA DE APRESENTAÇÃO

Aqui são apresentadas as telas principais e quais as características técnicas do seu desenvolvimento e seu respectivo comentário.

Dentre os estágios da elaboração da aplicação, essa etapa condiz com a construção da interface (*front-end*), ou seja, a parte visual do site, o que o usuário vê.

Desse modo, foi dada em duas seções, a primeira seção diz respeito a origem do esquema de cores, formatos e *design* da aplicação, com alguns breves comentários a respeito da tela em si. Posteriormente é detalhado as ferramentas utilizadas para o desenvolvimento desta etapa e suas as razões de empregá-las.

2.4.1 DESIGN DAS PRINCIPAIS TELAS

A fomentação de um *design* não é um processo simples, tornando se fundamental para a relação do usuário com a aplicação que está sendo desenvolvida, pois através dele exploram se as sensações dos utilizadores de maneira indireta.

Então para um *design*, que originasse uma interpretação concisa e fácil para o usuário compreender a proposta da aplicação. Usou se a psicologia, por meio do ramo conhecido como Gestalt, nesse segmento trabalhe se a percepção humana, sendo baseado em 7 princípios, de proximidade, similaridade, continuidade, fechamento, figura-fundo, região comum e ponto focal. Tais princípios são bastante utilizados em projetos de *UI (User Interface) design*.

A Gestalt entende que é de suma importância a disposição em que são apresentados à percepção os elementos unitários que compõem o todo. Uma de suas formulações bastante conhecidas é a de que “o todo difere da soma das partes”. Ou seja, a percepção que temos de um todo não é o resultado de um processo de simples adição das partes que o compõem (Uma introdução ao estudo de psicologia. São Paulo: Saraiva, 2004. pág. 50).

Ilustração 11 - Tela Cadastro de Cliente

Boa tarde, Andre Moutinho

Início Clientes Adicionar

Cadastro de Cliente

CPF/CNPJ Buscar(CNPJ) CEP

Nome/Razão Social* Rua

Contato: Número

Telefone Complemento

Celular Bairro

Email Cidade

Senha Estado

Tipo de Cliente ☐ Fornecedor

Salvar Voltar

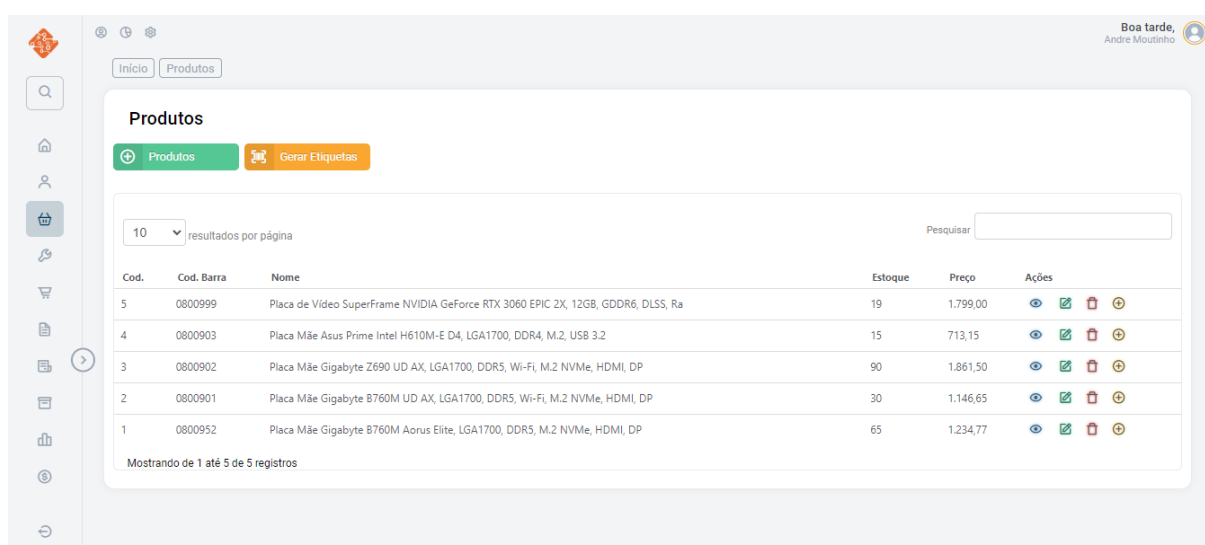
Fonte: Autores (2024)

Na tela de cadastro de cliente demonstrada na ilustração 11 – Tela Cadastro de Cliente, o princípio de proximidade, adquire grande destaque com os *labels* próximos aos campos, dando ao usuário funcionário a sensação de serem o mesmo elemento, assim facilitando no momento de preenchimento de cadastro do cliente.

Na maior parte das telas também é tido o conceito de similaridade, onde os campos de inserção de conteúdo nos campos, botão de salvar têm formas semelhantes, entretanto outros botões, como o de voltar possuem uma cor diferente, para o usuário ter a percepção que suas funções são distintas. Como foi apresentado por exemplo, na tela de cadastro de cliente.

É importante ressaltar também a implementação do conceito de fechamento visto em várias telas, onde os elementos são agrupados criando grupos diferentes sem ter uma divisória, porém mesmo sem uma divisão é possível entender que suas funções são diferentes. Isso é feito para evitar uma poluição visual para os usuários. Por exemplo, na tela de adicionar produto representado na ilustração 12 – Tela de Adicionar Produto, pode-se verificar três agrupamentos de tela, em um grupo há a visualização dos produtos já existentes cadastrados, no outro é composto pelo botão de adicionar um produto e no outro um botão de gerar etiquetas, mesmo sem uma divisão é perceptível que os dois não são um conjunto.

Ilustração 12 - Tela de Adicionar Produto



Fonte: Autores (2024)

A questão da coloração representada na ilustração 13 – Paleta de Cores, foi eleita por meio do site Adobe color, nela foi realizada uma pesquisa de cores que tivessem alguma relação com prestadoras de serviços, então foi escolhido a paleta que mesclasse as cores que remetem a esse intuito. Além das cores também simbolizarem credibilidade, confiança e profissionalismo.

Ilustração 13 - Paleta de Cores



Fonte: Adobe Color

2.4.2 FRONT-END

A inclusão do *front-end* na aplicação ocorre através do *framework* desenvolvido por EllisLab originalmente e atualmente é mantida por British Columbia Institute, o *Codeigniter*, plataforma para a construção da interface de aplicações na linguagem de programação *PHP*. De acordo com Rasmus Lerdorf, o *Codeigniter* foi desenvolvido sobre um paradigma da programação orientada a objetos sob o padrão de arquitetura de *software* MVC, mas também já dando suporte à metodologia HMVC, um nível hierárquico acima, absorvendo a tríade MVC, tornando-a modular e, até certo ponto, independente das demais. (2016).

O *framework Codeigniter 4* foi criado justamente para a construção de aplicações *web*, principalmente por conta de um conjunto de ferramentas simples para a criação de aplicativos *web* completos.

Esse *framework* oferece componentes prontos em *PHP* facilitando e otimizando o tempo na hora de desenvolver uma aplicação. Criada pela EllisLab, o *Codeigniter* foi desenvolvido no intuito de agilizar o processo de desenvolvimento de *software*, reduzindo as linhas de códigos desenvolvidas do projeto, tornando o desenvolvimento mais produtivo e ágil (Desvendando o CodeIgniter 4. 2020. P.3)

Rasmus Lerdorf, criador do *PHP*, expressou que gostava do *CodeIgniter* "porque é rápido, ligeiro e parece pouco um entorno". Também como forma de completo ao processo de desenvolvimento, foi incluída uma biblioteca dos correios, deste modo, realiza o preenchimento do endereço digitado, apenas informando o CEP.

3 RESULTADO

O SIG-OS é um ERP comercial e tem como seu principal intuito a gestão de ordens de serviços, voltadas a pequenas e médias empresas prestadoras de serviços. Após a finalização do seu processo de desenvolvimento, a aplicação foi implantada em uma pequena empresa voltada a segurança eletrônica, para que deste modo fosse possível realizar a validação da aplicação e se as necessidades elencadas pelo proprietário no questionário de entrevista haviam sido atendidas com êxito. O questionário de entrevista realizada com o proprietário pode ser visualizado no Apêndice C desta monografia.

As respostas dadas pelo proprietário na etapa inicial, o questionário, foram de extrema relevância, para que ficassem claras as problemáticas que ele enfrentava, e o que fosse mais adequado para melhorar sua experiência diante de uma aplicação voltada a gestão, já que ele estava ambientado com outras soluções, mas que não o atendiam de maneira tão satisfatória.

O proprietário da empresa foi convidado a operacionalizar por completo toda a aplicação, e sua participação ativa nesta etapa permitiu avaliar se todas as necessidades mencionadas no questionário de entrevista haviam sido concluídas de forma satisfatória. As contribuições do proprietário foram cruciais para aprimorar e otimizar a gestão das OS's, bem como a experiência dele diante da aplicação.

Uma das necessidades apresentadas pelo proprietário, foi uma área voltada ao cliente, para que esses clientes pudessem ter acesso as OS's abertas, o andamento delas, bem como a todos os seus dados cadastrados na empresa. Tal necessidade foi expressa em uma das respostas dadas através do questionário.

Após esta etapa de validação, o proprietário compartilhou observações positivas dentre elas destacando a rapidez da aplicação, bem como elogiando a automatização dos campos referente ao CEP durante o cadastro do cliente, que na solução anterior que ele possuía, onde os dados relacionados ao CEP tinham de ser inseridos manualmente. Um cliente escolhido pelo proprietário, para testar a área do cliente, enfatizou a interface intuitiva e com boa fluidez, além de mencionar o fácil manuseio em diversas telas (computador, celular e *tablet*), o que facilita o entendimento de diversos clientes, sejam eles mais experientes ou menos com tecnologia.

3.1 CONCLUSÃO

O objetivo do SIG-OS, um ERP comercial voltado para a gestão de ordens de serviços em pequenas e médias empresas prestadoras de serviços, foi plenamente atingido. Após a conclusão do desenvolvimento, a aplicação foi implantada em uma empresa de segurança eletrônica para validação. O processo incluiu a participação ativa do proprietário da empresa, que forneceu insights cruciais através de um questionário inicial que detalhou as problemáticas enfrentadas e as melhorias desejadas em uma solução de gestão.

A operacionalização completa do SIG-OS pelo proprietário permitiu avaliar se as necessidades levantadas foram atendidas de maneira satisfatória. Entre as funcionalidades destacadas, a área voltada ao cliente foi uma das principais demandas atendidas. Esta funcionalidade permitiu que os clientes acessassem informações sobre as ordens de serviço abertas, o andamento delas e os dados cadastrados, proporcionando maior transparência e eficiência.

As observações pós-implantação foram positivas, com o proprietário destacando a rapidez da aplicação e a automatização dos campos relacionados ao CEP durante o cadastro do cliente, eliminando a necessidade de inserção manual, uma limitação presente na solução anterior. Além disso, um cliente escolhido para testar a área do cliente elogiou a interface intuitiva e a fluidez, destacando o fácil manuseio em diversas telas, o que é essencial para a acessibilidade de usuários com diferentes níveis de familiaridade tecnológica.

Esses *feedbacks* positivos, tanto do proprietário quanto do cliente final, indicam que o SIG-OS não só atendeu às necessidades iniciais, mas também aprimorou significativamente a gestão das ordens de serviço, confirmando que o objetivo principal da aplicação foi atingido com êxito.

3.2 PASSOS FUTUROS

Com base nos resultados positivos obtidos durante a validação do SIG-OS, planejamos uma série de ações futuras para expandir e aprimorar ainda mais o projeto, garantindo que continue atendendo às necessidades demandadas, seguindo os seguintes passos:

Aplicativo para dispositivos Móveis: Desenvolvimento de um aplicativo para Android e iOS, para que os clientes possam ter a comodidade em acessar via

aplicativo (e não apenas pelo navegador) a visualização das ordens de serviço, bem como o status dela, além das informações de dados cadastrais. Desta maneira buscando a fidelização do cliente de maneira contínua.

Expansão de Funcionalidades: Planejamos adicionar novas funcionalidades ao SIG-OS para aumentar sua capacidade de gestão e atender a uma gama mais ampla de necessidades. Isso inclui a integração com sistemas de pagamento online, aprimoramento do módulo de gestão financeira para controle de despesas e receitas que já existe.

Aprimoramento da Experiência do Usuário: Continuaremos a coletar feedback dos usuários para melhorar a interface e a usabilidade do SIG-OS. Planejamos realizar atualizações regulares com base nas sugestões dos usuários, focando em tornar a interface ainda mais intuitiva e acessível para pessoas com diferentes níveis de familiaridade tecnológica.

Implementação de Inteligência Artificial: Planejamos incorporar tecnologias de inteligência artificial para análise de dados e previsões. Isso permitirá que as empresas usuárias possam tomar decisões mais informadas com base em insights gerados pelo sistema, como a previsão de demandas e a identificação de padrões de problemas recorrentes.

Suporte e Treinamento: Vamos ampliar nosso suporte ao cliente e oferecer programas de treinamento mais abrangentes para garantir que todas as empresas possam utilizar o SIG-OS ao máximo de seu potencial. Isso incluirá tutoriais online, webinars, e um serviço de suporte técnico dedicado.

Parcerias Estratégicas: Procuraremos estabelecer parcerias estratégicas com outras empresas de tecnologia e consultorias para promover o SIG-OS e ampliar sua adoção. Essas parcerias nos ajudarão a atingir um público maior e a incorporar as melhores práticas do mercado em nosso desenvolvimento contínuo.

Com esses planos futuros, estamos confiantes de que o SIG-OS continuará a evoluir e a oferecer soluções inovadoras e eficientes para a gestão de ordens de serviço, atendendo de forma cada vez mais eficaz as necessidades das pequenas e médias empresas prestadoras de serviços.

REFERÊNCIAS

Adobe Color, 2023. Disponível em: <https://color.adobe.com>. Acesso em: 27 de maio de 2023.

BAZZI, Claudio. **Introdução a banco de dados**. Apostila: Universidade Tecnológica Federal do Paraná, 2013. e-ISBN: 978-85-7014-114-9.

BEGOSSO, Luiz Carlos. (2010). Gestão de ordens de serviço: proteção de informações confidenciais em ambientes digitais. Disponível em: <https://cepein.femanet.com.br/BDigital/arqTccs/0811060056.pdf>. Acesso em: 05/09/2023.

CARLOS, Antonio. Sistemas da Informação x Tecnologia da Informação. IFBA, Bahia, 25 de out. de 2015. Disponível em: http://www.ifba.edu.br/professores/antoniocarlos/index_arquivos/siti.pdf. Acesso em: 10 de set. de 2023.

DA SILVA, Heloiza. Uso da UML na Especificação do Sistema de Informação da Área de Infra-Estrutura da UFRGS. Porto Alegre, 12 de dez de 2003. Disponível em: <https://www.produttivo.com.br/blog/gestao-de-ordem-de-servico/>. Acesso em: 15/10/2023.

De Sordi, J. O., & Marinho, B. L. (2017). Integração de sistemas de informação: análise dos impactos nos processos empresariais. Disponível em: <https://dialnet.unirioja.es/descarga/articulo/7760595.pdf>. Acessado em: 15 março de 2023

EOS-OS. **EOS-Os: Sistema de Gestão de Ordem de Serviço**, c2023. Página inicial. Disponível em: <https://eos-os.com.br>. Acesso em: 16 de set. de 2023.

LERDORF, Rasmus. **Programming PHP, 2nd Edition**. 2.º edição. Nova York, 2006.

MAXIMIANO, Antonio Cesar Amaru. **Administração para empreendedores: fundamentos da criação e gestão de novos negócios**. Pearson, 2011.

NETO, Francisco Paulo de Melo. (2009). Gestão de Operações de Serviço. Disponível em: <https://canal.cecierj.edu.br/012016/12231a5840587ead20bc834b3e1b6a31.pdf>. Acesso em: 25/11/2023.

OZORIO, Jacksson. (2023). Aumentando a Produtividade e a Eficiência da Gestão de Ordens de Serviço com a Tecnologia. Disponível em: <https://n1sistemas.com.br/aumentando-a-produtividade-e-a-eficiencia-da-gestao-de-ordens-de-servico-com-a-tecnologia/>. Acesso em: 05/09/2023.

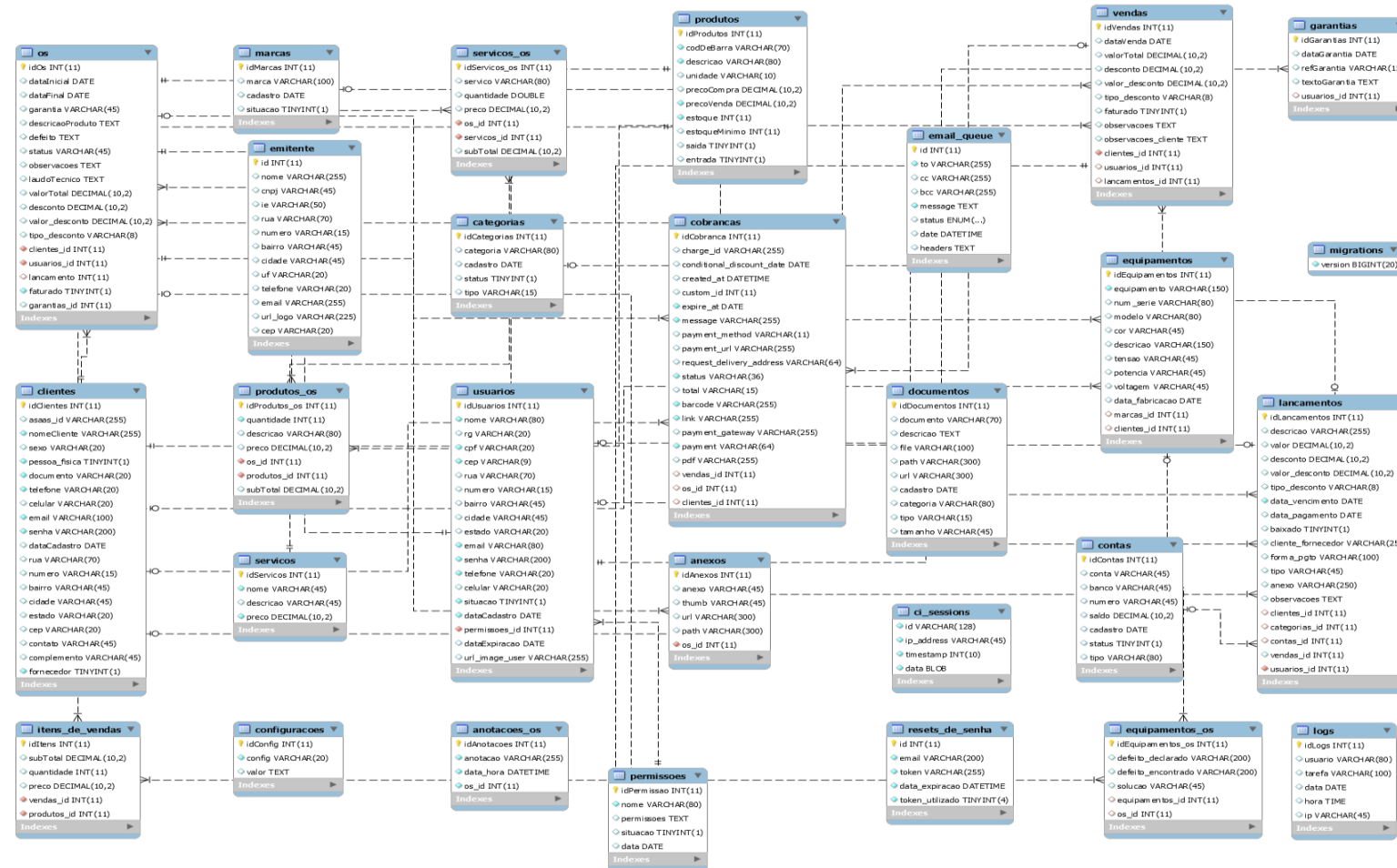
PEREIRA, Luiz. **Análise e Modelagem de Sistemas com a UML: Com Dicas e Exercícios Resolvidos**. 1.º edição. Rio de Janeiro, 2011. ISBN 978-85-911695-0-4.

SILVA, Alberto; VIDEIRA, Carlos. **UML, Metodologias e ferramentas CASE: Linguagem de Modelação UML, Metodologias e Ferramentas CASE na Concepção e Desenvolvimento de Software**. 1.º edição. Inova, abril 2001. ISBN: 972-8426-36-4.

RIBEIRO, L. E-book sobre PCM - Planejamento Programação e controle de manutenção. Academia de Manutenção. Disponível em: <http://www.academiademanutencao.com/materiais>. Acesso em: 26/08/20.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011. ISBN 978-85-7936-108-1.

APÊNDICE B – Modelo de Entidade e Relacionamento



Fonte: Autores (2024)

APÊNDICE C – Questionário de Entrevista



Questionário de Entrevista

1) A empresa já utilizou algum sistema semelhante ao SIG-OS?

Não. O sistema que é utilizado atualmente, é instalado no próprio computador.

2) Quais são os problemas enfrentados com o sistema atual?

O sistema é lento, algumas vezes trava com o manuseio de outros programas do computador, algumas funções não fazem de forma satisfatória uma ação.

3) Quais são as funcionalidades que o sistema atual não possui e que seria interessante ter?

Uma área de clientes, registro de receitas e despesas, estatísticas financeiras com um gráfico, um calendário grande, um balanço mensal do ano com gráfico.

4) O sistema atual é instalado no computador ou é um sistema web (acessado via internet)?

O sistema atual é instalado na máquina, sem acesso à internet.

5) O sistema atual oferece relatórios? Se sim, como você avalia a qualidade desses relatórios?

O sistema oferece relatórios, mas apenas do dia, e não por um período especificado.

6) O sistema atual possui uma área dedicada ao cliente? Seria importante a área do cliente?

Não. Seria importante, para que os clientes, acompanhassem o status das OS's.

7) O sistema atual possui alguma funcionalidade automatizada? Se sim, como você avalia sua eficácia?

Não. O sistema não possui funcionalidade automatizada, tudo tem de ser digitado.

8) Quais funcionalidades automatizadas você considera essenciais no novo sistema?

Preenchimento automático de endereço pelo CEP durante o cadastro, busca inteligente de produtos, serviços, clientes.

9) Quão importante é a personalização do sistema para atender às necessidades específicas da sua empresa?

A personalização da folha da ordem de serviço, contendo os dados da minha empresa junto da logo da empresa e as informações do cliente, da OS, do serviço realizado, de algum produto utilizado para a realização do serviço e o valor do serviço.

10) Como você avalia a eficiência do sistema atual em termos de tempo de resposta e desempenho?

O sistema atual não possui um bom tempo de resposta e nem de desempenho nos computadores em que é executado.