

---

**Faculdade de Tecnologia de Americana “Ministro Ralph Biasi”**  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

Daniel Evaldo da Silva

Fernando Pinheiro Alves

Geraldo dos Santos Júnior

Joyce Lima de Arruda

**DESENVOLVIMENTO DE UM SISTEMA DE GERENCIAMENTO  
PARA LOJAS DE INFORMÁTICA E PERIFÉRICOS**

**Americana, SP  
2022**

---

**Faculdade de Tecnologia de Americana “Ministro Ralph Biasi”**

**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

Daniel Evaldo da Silva

Fernando Pinheiro Alves

Geraldo dos Santos Júnior

Joyce Lima de Arruda

**DESENVOLVIMENTO DE UM SISTEMA DE GERENCIAMENTO  
PARA LOJAS DE INFORMÁTICA E PERIFÉRICOS**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do curso de Análise e Desenvolvimento de Sistemas, da Faculdade de Tecnologia de Americana, sob a orientação do Prof. Dr Ivan Menerval da Silva.  
Área de concentração: Análise e Desenvolvimento de Sistemas.

**Americana, SP  
2022**

Daniel Evaldo da Silva

Fernando Pinheiro Alves

Geraldo dos Santos Júnior

Joyce Lima de Arruda

## **DESENVOLVIMENTO DE UM SISTEMA DE GERENCIAMENTO PARA LOJAS DE INFORMÁTICA E PERIFÉRICOS**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do curso de Análise e Desenvolvimento de Sistemas, da Faculdade de Tecnologia de Americana, sob a orientação do Prof. Dr Ivan Menerval da Silva.  
Área de concentração: Análise e Desenvolvimento de Sistemas.

**Americana, 07 dezembro de 2022.**

**Banca Examinadora:**

---

Ivan Menerval da Silva (Presidente)

Doutor

Fatec Americana

---

Eduardo Antonio Vicentini (Membro)

Mestre

Fatec Americana

---

Diógenes de Oliveira (Membro)

Mestre

Fatec Americana

Esta monografia é dedicada aos nossos familiares e ao nosso orientador, que nos apoiou para o seu desenvolvimento.

## **AGRADECIMENTOS**

Agradecemos a Deus, primeiramente, por ter nos dado saúde e a oportunidade de poder fazer este trabalho.

Aos professores que tivemos durante o curso, que contribuíram com o nosso crescimento e conhecimento.

Ao nosso orientador, que nos incentivou e dedicou seu tempo para contribuir com seu conhecimento em nossa monografia.

Aos nossos familiares e amigos, pela paciência e compreensão ao longo desta trajetória, oferecendo todo o apoio necessário para que conseguíssemos concluir esta etapa.

“Aprender é, de longe, a maior recompensa”  
(William Hazlitt)

## RESUMO

Esta monografia enfoca o desenvolvimento de um software para gerenciamento e controle de lojas de manutenção de equipamentos e periféricos de informática. Com base na necessidade do atendimento com mais agilidade e controle todas as atividades realizadas, cadastro de usuário, cadastro de clientes, cadastro de serviços, emissão de ordem de serviços e relatórios gerais. Ao se criar esse sistema os usuários terão um controle de todos os clientes ligados a loja mantendo seu cadastro sempre atualizado podendo realizar alterações sempre que houver necessidade. Os usuários têm a opção de criar ordem de serviço podendo alterá-las ou até mesmo apagá-las ou imprimi-las assim que desejar, podendo ser uma ordem de serviço concreta ou apenas um simples orçamento. O administrador majoritário tem a opção de criar e excluir usuários e ao mesmo tempo limitar a acessibilidade do usuário ao sistema, limitando acesso a informações que sejam mais sigilosas como por exemplo o relatório total de serviços realizados. Para o desenvolvimento deste software foi realizada uma pesquisa de várias ferramentas e metodologias para conseguirmos encontrar as ferramentas e tecnologias necessárias para atender a nossa necessidade. Por fim as seguintes metodologias foram adotadas. O Método de desenvolvimento SCRUM, linguagem de programação Java, o banco de dados MySQL Server.

**Palavras Chave:** Gerenciamento e controle de loja; Manutenção de computadores; Desenvolvimento de *Software*; SCRUM.

## **ABSTRACT**

This monograph focuses on the development of software for the management and control of a maintenance shop for computer equipment and peripherals.

Based on the need for service with more agility and control of all activities carried out, user registration, customer registration, service registration, service order issuance and general reports.

When creating this system, users will have control of all customers connected to the store, Keeping their records always updated, being able to change or even delete them or print them as they wish, which can be a concrete work order or just a simple quote. The majority administrator has the option to create and delete users and at the same time limit the user's accessibility to the system, limiting access to information that is more confidential, such as the total report of services performed.

For the development of this software, research of several tools and methodologies was carried out so that we were able to find the necessary tools and technologies to meet our needs.

Finally, the following methodologies were adopted. The SCRUM development method, Java programming language, MySQL Server database.

**Keywords:** Store management and control; Computer maintenance; Software development; SCRUM.



## **SUMÁRIO**

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	Objetivos	15
1.1.1	Objetivo Geral	16
1.1.2	Objetivo Específicos	16
1.2	Justificativa	16
1.3	Público Alvo	17
<b>2</b>	<b>PROJETO do sistema</b>	<b>18</b>
2.1	Levantamento de Requisitos	18
2.1.1	Requisitos Funcionais	17
2.1.2	Requisitos não funcionais	19
2.2	Regra de Negócio	20
2.3	Software Similares	20
2.4	Recursos e Ferramentas	21
<b>3</b>	<b>MODELAGEM</b>	<b>24</b>
3.1	Casos de Uso	24
3.1.1	Documentação dos Casos de Uso	28
3.2	Diagrama de Classe	35
3.3	Banco de Dados	37
3.3.1	Diagrama de Entidade e Relacionamento	37
<b>4</b>	<b>DESENVOLVIMENTO</b>	<b>40</b>
4.1	Etapas do Desenvolvimento	40
4.1.1	Entrega 1	41
4.1.2	Entrega 2	42
4.1.3	Entrega 3	44
4.1.4	Entrega 4	46
4.1.5	Entrega 5	47
4.2	Interface do Sistema	49
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>54</b>
<b>6</b>	<b>REFERÊNCIAS</b>	<b>55</b>

APÊNDICE A – Dicionário de Dados	59
APÊNDICE B – Código fonte do Sistema	60

## GLÓSSÁRIO

Br: Brasil

PT: Português

RF: Requisito Funcional

RNF: Requisito não Funcional

RN: Regra de negócio

S1: Software 1

S2: Software 2

S3: Software 3

MER: Modelo Entidade-Relacionamento

DER: Diagrama Entidade Relacionamento

ERP: Enterprise Resource Planning

MEI: Micro Empreendedor Individual

SGBD: (Sistema de Gerenciamento de Bancos de Dados)

## **LISTA DE FIGURAS**

<b>Figura 1</b> – Casos de uso do Administrador	25
<b>Figura 2</b> – Casos de uso do Usuário	25
<b>Figura 3</b> – Casos de uso do Cliente	26
<b>Figura 4</b> – Casos de uso da ordem de serviço administrador	26
<b>Figura 5</b> – Casos de uso ordem de serviço do Usuário	27
<b>Figura 6</b> – Casos de uso do administrador adicionar Usuário	27
<b>Figura 7</b> – Diagrama de classe Usuário	35
<b>Figura 8</b> – Diagrama de classe Administrador	36
<b>Figura 9</b> – Diagrama Entidade e Relacionamento	38
<b>Figura 10</b> – Banco de dados	39
<b>Figura 11</b> – Burndown 1º entrega	42
<b>Figura 12</b> – Burndown 2º entrega	44
<b>Figura 13</b> – Burndown 3º entrega	45
<b>Figura 14</b> – Burndown 4º entrega	47
<b>Figura 15</b> – Burndown 5º entrega	48
<b>Figura 16</b> – Tela do sistema Login	49
<b>Figura 17</b> – Tela Principal	50
<b>Figura 18</b> – Tela do Cadastro	50
<b>Figura 19</b> – Tela adicionar Cliente	51
<b>Figura 20</b> – Tela Pesquisa	51
<b>Figura 21</b> – Tela remover Cliente	52
<b>Figura 22</b> – Tela alterar dados do Cliente	52
<b>Figura 23</b> – Tela cliente removido com Sucesso	53
<b>Figura 24</b> – Tela não remoção de Cliente	53
<b>Figura 25</b> – Login Conexão	60
<b>Figura 26</b> – Consulta ao banco de Dados	61

<b>Figura 27</b> – Tela Login	62
<b>Figura 28</b> – Tela Adicionar Clientes	62
<b>Figura 29</b> – Tela Alterar Cliente	63
<b>Figura 30</b> – Pesquisar Cliente	63
<b>Figura 31</b> – Tela Remover Clientes	64
<b>Figura 32</b> – Tela Adicionar Usuários	64
<b>Figura 33</b> – Tela Alterar Usuário	65
<b>Figura 34</b> – Conexão do Usuário	65
<b>Figura 35</b> – Consulta de Usuário Cadastrado	66
<b>Figura 36</b> – Remoção de Usuário	67
<b>Figura 37</b> – Alterar Ordem de Serviço	67
<b>Figura 38</b> – Emitir Ordem de Serviço	68
<b>Figura 39</b> – Excluir Ordem de Serviço	68
<b>Figura 40</b> – Imprimir Ordem de Serviço	69
<b>Figura 41</b> – Limpar Ordem de Serviço	69
<b>Figura 42</b> – Conexão da Ordem de Serviço	70
<b>Figura 43</b> – Pesquisar a Ordem de Serviço	70
<b>Figura 44</b> – Recuperar a Ordem de Serviço	71

## **LISTAS DE TABELAS**

<b>Tabela 1</b> - Requisitos funcionais do projeto	19
<b>Tabela 2</b> - Requisitos não funcionais do projeto	19
<b>Tabela 3</b> - Regra de Negócio	20
<b>Tabela 4</b> - Comparativo de softwares	21
<b>Tabela 5</b> - Caso de uso Login	28
<b>Tabela 6</b> - Caso de uso tela Principal	29
<b>Tabela 7</b> - Caso de uso Usuário	30
<b>Tabela 8</b> - Caso de uso Cliente	31
<b>Tabela 9</b> - Caso de uso Ordem de serviços	32
<b>Tabela 10</b> - Caso de uso acessar relatório	33
<b>Tabela 11</b> - Caso de uso sobre	34
<b>Tabela 12</b> - Planejamento do primeiro entregável	41
<b>Tabela 13</b> - Planejamento do segundo entregável	42
<b>Tabela 14</b> - Planejamento do terceiro entregável	44
<b>Tabela 15</b> - Planejamento do quarto entregável	46
<b>Tabela 16</b> - Planejamento do quinto entregável	47
<b>Tabela 17</b> – Dicionário de dados tabela clientes	59
<b>Tabela 18</b> - Dicionário de dados tabela usuários	59
<b>Tabela 19</b> - Dicionário de dados tabela ordem de serviços	59

## 1 INTRODUÇÃO

Atualmente o mercado consumidor brasileiro principalmente pós pandemia tem buscado adquirir equipamentos de informática para se manter conectado às tendências da tecnologia (MEIRELLES, 2022). No primeiro trimestre de 2020 houve por conta da pandemia uma redução na oferta de serviços de compra, venda e manutenção de computadores. “Em meio à pandemia, mercado de PCs cai 12,3% no primeiro trimestre de 2020“(INSIDE, 2020). Já em 2021 o cenário tem se mostrado favorável apresentando um crescimento considerável na busca por esses serviços como pode ser observado no artigo da IDC BRASIL (IDC, 2022). Alavancado por esse crescimento surge também a necessidade de gerenciamento das empresas ofertantes desses serviços. Cerca de 70% dessas empresas são enquadradas como MEI, as quais não têm capital para investir em software com alto custo para aquisição. Em pesquisas realizadas por software para esta finalidade, nota-se que ou possuem alto custo ou são providas em cloud.

Com base no cenário apresentado, este trabalho propõe o desenvolvimento de uma aplicação (Sistema de Gerenciamento de lojas de informática), utilizando a linguagem Java e demais tecnologias Open Source, que poderá ser utilizada gratuitamente por empresas do setor, provendo cadastro de usuário, cliente, serviços e relatórios diversos.

Para atender tal demanda, este trabalho propõe como objetivo principal o desenvolvimento de um Sistema de Gerenciamento de lojas de informática visando agilizar a demanda no recebimento de equipamentos para manutenção de microcomputadores e periféricos da loja, trazendo a agilidade no recebimento e no andamento do trabalho de manutenção, melhorando o relacionamento com o cliente quanto ao tempo de retorno desde o balcão, com o orçamento de peças a mão de obra dos serviços, a aprovação pelo cliente até a liberação do serviço prestado ao cliente.

Para atingir este objetivo, o grupo utiliza a metodologia Scrum, a linguagem definida foi Java junto ao MySQL para criação do banco de dados.

## 1.1 OBJETIVOS

O objetivo de gerenciar o controle de entrada e saída de equipamentos e periféricos de informática ao mesmo tempo garantir uma comunicação mais assertiva entre as pessoas envolvidas, seja interna ou externa. O documento possibilita o monitoramento de um determinado processo (desde que o armazenamento e arquivamento sejam feitos corretamente, contemplando as seguintes funcionalidades:

- **Cadastro de usuários:** Responsável por conter a inclusão dos nomes, telefones, login de usuários e senha de acesso ao sistema.
- **Cadastro de clientes:** Contém a inclusão de nomes, endereço, telefone e Email dos clientes.
- **Ordem de serviços:** Possui a situação, o equipamento, defeito, serviço, nome do técnico e valor referente ao equipamento que se encontra na assistência técnica.
- **Relatórios:** Gerar relatórios da quantidade de serviços e valores na assistência técnica.
- **Gerência de serviços e valores:** Utilizado pelo administrador do sistema para acompanhar todo o processo de entrada de microcomputadores e periféricos e acesso a relatórios específicos na loja e fazer possíveis ajustes.

Para o desenvolvimento deste trabalho o conteúdo foi organizado em cinco capítulos que abordam as metodologias adotadas, o levantamento de requisitos funcionais e não funcionais, os recursos e ferramentas utilizadas, a modelagem de dados, a documentação de casos de uso, o diagrama de classes, a interface do sistema e o desenvolvimento a partir da metodologia Scrum.



### 1.1.1 Objetivo Geral

Ganho de tempo nas atividades do dia a dia da loja com a manutenção de microcomputadores e periféricos e controle maior de entrada e saída na loja.

Ter um sistema de gerenciamento da loja de informática sem a necessidade de pagamento de mensalidade.

### 1.1.2 Objetivos Específicos

- Identificar a movimentação das peças e dos equipamentos na loja de informática.
- Descrever a rotina de trabalho desde a chegada do equipamento para orçamento, aprovação de reparo e liberação do equipamento.
- Discutir como minimizar o tempo de espera do equipamento no conserto com o uso da tecnologia.

## 1.2 JUSTIFICATIVA

A viabilidade de um serviço informatizado tem um ganho em tempo no processo, desde o momento que o equipamento é recebido no balcão da loja até o final do conserto.

Ganho de tempo para que o técnico informe que o pedido foi concluído e o cliente seja informado o valor do conserto.

O acesso informatizado traz agilidade ao providenciar as peças para o reparo.

Menos tempo gasto pela secretária na devolução do pedido com todos os materiais que serão utilizados para o conserto do equipamento.

O envio do orçamento para o cliente é muito mais fácil e sem perda de tempo.

Ter o maior retorno sobre o investimento no menor tempo possível.

Maior rotatividade de equipamentos aguardando conserto na loja.

O sistema ajuda ao microempreendedor a ter mais eficiência e eficácia em seu negócio.

### 1.3 PÚBLICO ALVO

Empresa de pequeno porte que procura agilizar, dinamizar e organizar a rotina de loja de informática, equipamentos, serviços e relatório que são inseridos diariamente.

## 2 PROJETO DO SISTEMA

Para o desenvolvimento deste trabalho foi adotado a Metodologia Ágil Scrum, o projeto divide em ciclos de sete a trinta dias, chamados de “Sprints”, com a interação e ordem as atividades é desenvolvida, o Software é construído através do conjunto de atividades que representa uma caixa de tempo (*Time Box*) mediante o desenvolvimento de um *time Box*, as atividades que deverão ser realizadas são: Reunião para levantamento de requisitos para entendimento do problema, validação de requisitos, elaboração de diagrama de classe, caso de uso, criação de banco de dados, projeto funcional, interface visual do sistema para a aprovação, codificação, entrega da versão beta, testes de aceitação e possíveis evoluções.

Para o desenvolvimento do sistema de gerenciamento de lojas de informática, o cronograma mencionado será seguido para atingir o objetivo de entrega de software. Requisito: Prioriza requisito com base na necessidade do cliente.

### 2.1 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos denominado Engenharia de Requisitos RE sigla em inglês para *Requirements Engineering*, neste processo compreende as necessidades do cliente, documentando todos os requisitos analisados a serem atendidos no desenvolvimento do sistema. Que compreende e define as descrições dos serviços fornecidos pelo sistema e suas restrições operacionais (SOMMERVILLE, 2007). São divididos em requisitos funcionais e requisitos não funcionais.

#### 2.1.1 Requisitos Funcionais

Os requisitos Funcionais do sistema descrevem as funcionalidades do sistema, sendo elas tarefas e serviços oferecidos (SOMMERVILLE, 2011). A Tabela 1 apresenta os requisitos funcionais deste projeto.

**Tabela 1 - Requisitos funcionais do projeto**

Identificação	Requisitos Funcionais	Prioridade
RF001	Tela Login	Essencial
RF002	Tela Principal	Essencial
RF003	Tela Clientes	Essencial
RF004	Tela Ordem de Serviço	Essencial
RF005	Tela Usuário	Essencial
RF006	Tela Relatório	Essencial
RF007	Tela Sobre	Opcional

Fonte: Autoral (2022).

### 2.1.2 Requisitos Não Funcionais

Os Requisitos não funcionais estão relacionados ao funcionamento da usabilidade, escalabilidade e tecnologias envolvidas, é tudo o que está relacionado ao sistema (SOMMERVILLE, 2007). A seguir a tabela 2 apresenta os requisitos não funcionais deste projeto.

**Tabela 2 - Requisitos não funcionais do projeto**

Identificação	Requisitos não Funcionais	Categoria	Prioridade
RNF 01	Interface simples	Formatação	Importante
RNF 02	Gerar senha ao usuário	Segurança	Importante
RNF 03	UX	Interface	Essencial
RNF 04	Banco de Dados desenvolvido no MySQL	Informação	Essencial

Fonte: Autoral (2022).

## 2.2 REGRA DE NEGÓCIO

A regra de negócio mostrar o que não poderá ser realizado no sistema e o que pode ser alterado pelo usuário do sistema.

**Tabela 3 – Regra de negócio**

Identificação	Regra de Negócio
<b>RN 01</b>	O cadastro não poderá ser realizado no caso já exista algum usuário registrado com o mesmo login.
<b>RN 02</b>	Somente usuários cadastrados poderão efetuar login no sistema.
<b>RN 03</b>	O usuário poderá editar somente o seu próprio cadastro.
<b>RN 04</b>	Somente usuários cadastrados poderão realizar consultas no sistema.
<b>RN 05</b>	O usuário comum não poderá se visualizar ou imprimir relatórios
<b>RN 06</b>	Somente os usuários admin podem visualizar relatórios e imprimir relatórios.
<b>RN 07</b>	Somente o usuário admin poderá adicionar um novo usuário.
<b>RN 08</b>	Cliente com ordem de serviço não pode ser excluído.

Fonte: Autoral (2022).

## 2.3 LEVANTAMENTO DE FUNCIONALIDADES COM BASE NOS SOFTWARES SIMILARES

Atualmente existem vários softwares voltado para a área de gerenciamento de lojas de informática com diferentes funcionalidades, foram selecionadas 3 aplicações mais populares dentre a área, sendo estas:

- **S1 - Nuvem Gestor:** aplicação para acesso de onde estiver pelo computador, tablet ou celular a loja de informática e a oficina de eletrônicos de maneira fácil. tomar decisões a partir de relatórios preciso de cada produto vendido ou serviço realizado (sistema Manutenção de Computadores e loja de informática).
- **S2 - QuantoSobra:** um software de gestão para micro e pequenas empresas, focado no comércio varejista (os lojistas). também conhecido como ERP. (O

programa de computador para loja ideal para empresas que precisam de um controle completo)

- **S3 - Actana:** é um sistema ideal para sua loja de manutenção de computador. ERP (Sistema para lojas de manutenção de informática).

- 

**Tabela 4** - Comparativo de softwares

Funcionalidades	S1	S2	S3	Enable Solutions
Login com email e senha	X	X	X	X
Cadastro	X	X	X	X
Recuperar senha	X	X	X	-
Gratuito	-	-	-	X
Interface simples	-	-	-	X
Otimiza o tempo	X	-	-	X
Anúncios	X	X	X	-
Disponível para desktop	-	-	-	X
Disponível em nuvem	X	X	X	-
Controla senha dos usuário	X	X	X	X
Controla cadastro de clientes	X	X	X	X
Faz mapeamento das ordens de serviço	X	X	X	X
Emite relatório de vendas	X	-	-	X

**Fonte:** Autoral (2022).

## 2.4 RECURSOS E FERRAMENTAS

Neste tópico são abordadas as ferramentas utilizadas para o desenvolvimento deste trabalho, destacamos suas características mais relevantes.

- **Microsoft Teams:** é um ambiente que unifica os alunos e professores e time de gestão em um só lugar, além de possibilitar aulas ao vivo é desenvolvido para controlar, organizar e desenvolver o trabalho individual ou grandes e pequenas além de possibilitar as aulas ao vivo, gravação e upload de vídeo, chats interativos e edição de documentos simultaneamente.

- **Apache NetBeans 14:** é um dos ambientes de desenvolvimento integrado de códigos aberto e gratuito para desenvolvimento de aplicativos no sistema operacional Windows, Mac, Linux e Solaris mais utilizados pelos desenvolvedores Java. Mais que um editor de códigos, ele possui um conjunto de ferramentas que auxiliam a programação de tarefas comuns relacionadas à implementação de aplicações na plataforma NetBeans e uma amplo framework Java em que pode basear grandes aplicações desktop.
- **MySQL Workbench:** é uma ferramenta que lhe permite criar diagrama de relação entidade para bases de dados MySQL, desenhar layout da base de dados, documentar ou realizar migrações, administrar, executar consultas SQL (*Structure Query Language* - Linguagem de consulta Estruturada) está disponível para o Windows, Linux e Mac OS, é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados.
- **XAMPP:** é um pacote com os principais servidores de código aberto do mercado, incluindo FTP, banco de dados MYSQL e Apache com suporte às linguagens PHP e Perl. menção ao suporte a diferentes sistemas operacionais (Windows, Mac, Linux e Solaris) é oferecido como Software livre pela Apache Software Foundation.
- **iReport:** é uma Ferramenta front-end para o *JasperReports* um poderoso framework open-source escrito em Java para geração de relatórios. Ele permite gerar dinamicamente relatórios em diversos formatos: entre eles PDF, HTML, CSV E XML.
- **Google Docs:** é um programa de criação edição de texto online que funciona 100% na nuvem. (tem a mesma função do Word que faz parte do pacote Office da Microsoft) “onipresente” através de um dispositivo conecta-se à sua conta google do usuário para poder usar todas as suas funcionalidades desde a criação, edição e armazenamento dos arquivos é salvo automaticamente.
- **Astah Community:** é um software para modelagem UML (*Unified Modeling Language* – Linguagem de Modelagem Unificada) com suporte a UML 2, desenvolvido pela *Change Vision*, Inc e disponível para sistemas operacionais

Windows 64 bits. Anteriormente conhecido por Jude, um acrônimo de Java and UML *Developers Environment* (Ambiente para desenvolvedores UML e Java).

- **Java:** é uma linguagem de programação orientada a objetos desenvolvida na década de 90 por uma equipe de programadores chefiada por *James Gosling*, na empresa Sun Microsystems, que em 2008 foi adquirido pela empresa Oracle Corporation.
- **Whatsapp:** é um aplicativo multiplataforma de mensagens instantâneas e chamadas de voz para smartphones. Além de mensagens de texto, os usuários podem enviar imagens, vídeos e documentos em PDF, além de fazer ligações grátis por meio de uma conexão com a internet.
- **Google Meet:** Serviço de vídeo conferência gratuito permite gravar reuniões por uma hora ou pago suporta até 250 pessoas em uma sala.
- **OpenJDK:** É uma uma implementação open source do Java que provê um Java *Development Kit* totalmente baseado em software livre, assim como a versão oficial da Oracle, o software está na versão 8, lançado em março de 2014.
- **Pacote Office:** É uma suíte de aplicativos que reúne editor de textos, tabelas e apresentações para você utilizar no seu dia a dia, oferecidos pela suíte Microsoft.



### 3 MODELAGEM

A modelagem usa-se para representar o sistema em diferentes perspectivas para o desenvolvimento sendo dividido sua abordagem em dois segmentos, sendo eles:

- **brModelo** – Ferramenta para criar diagrama de entidade e relacionamento é utilizado para representar a modelagem relacional do banco de dados, descrevendo suas tabelas e o relacionamento entre elas (WEB, 2022).
- **Astah Community**: é um software para modelagem UML (Unified Modeling Language – Linguagem de modelagem Unificada) com suporte a uml 2, desenvolvido pela change vision, inc e disponível para sistemas operacionais Windows 64 Bits (COMMUNITY, 2018).

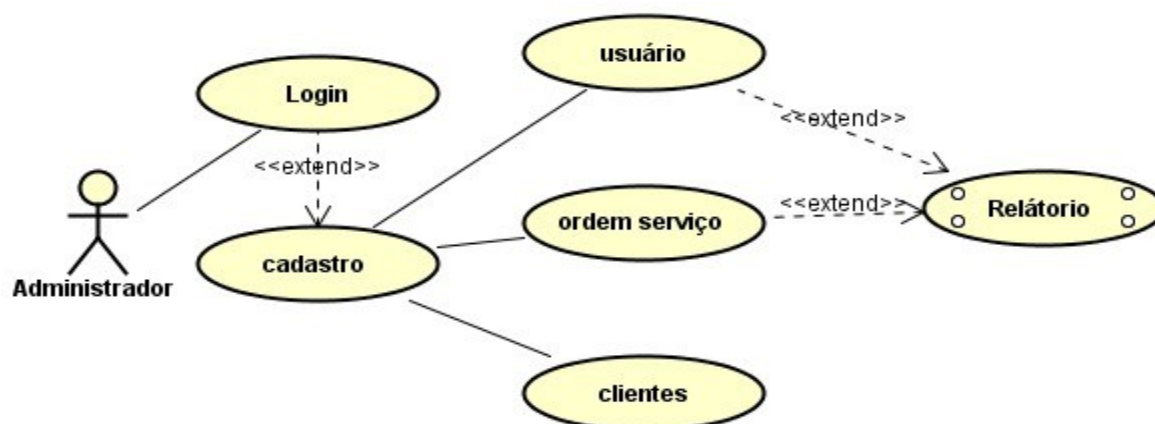
#### 3.1 CASOS DE USO

O diagrama de caso de uso contribui para catalogação de requisitos do sistema, e suas representações de interações com o sistema e suas funcionalidades. na característica deste diagrama os autores são responsáveis por executar determinadas ações do sistema.

- **Administrador**: representa o gestor do sistema; possui autorização para executar quaisquer operações no sistema.
- **Usuário**: representa a interação do usuário com o sistema; possui autorização para atuar em sua área específica

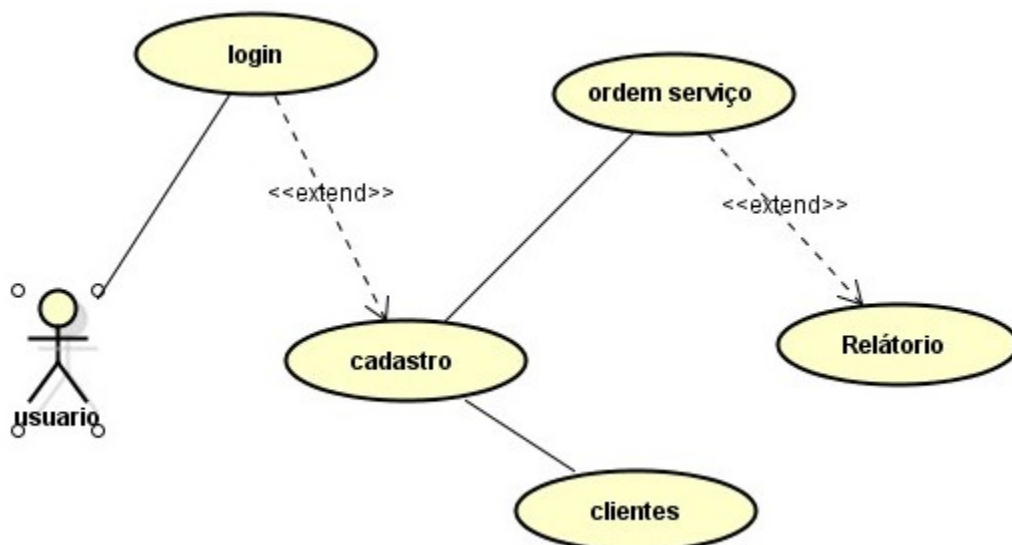
A Figura 1 representa de modo geral o caso de uso do sistema, descrevendo suas ações e seus respectivos atores.

Figura 1 - Casos de uso do administrador



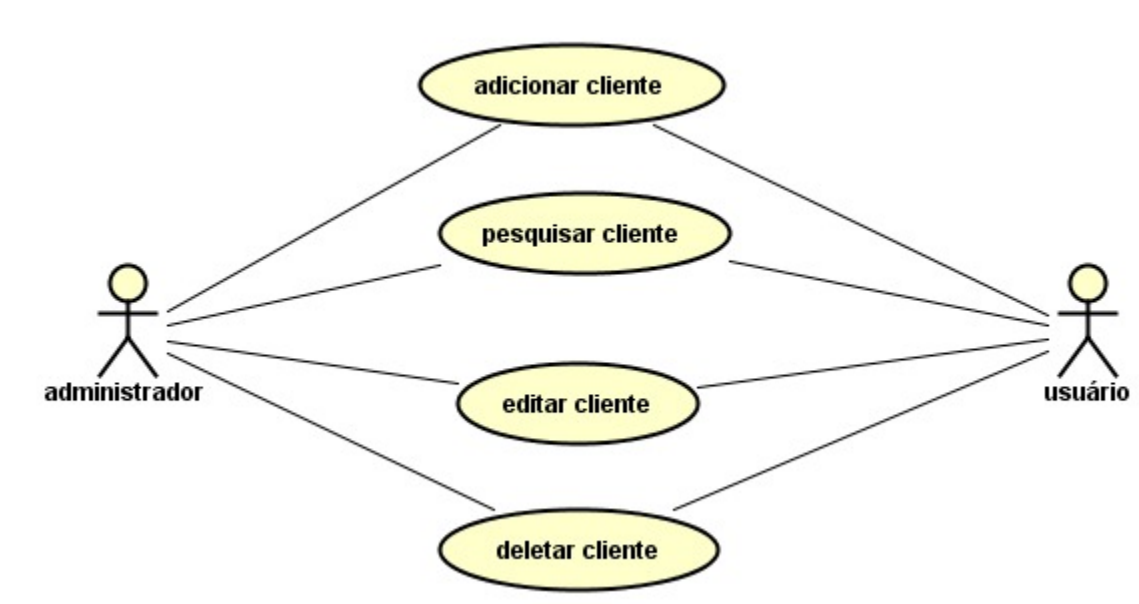
Fonte: Autoral (2022).

Figura 2 - Casos de uso do usuário



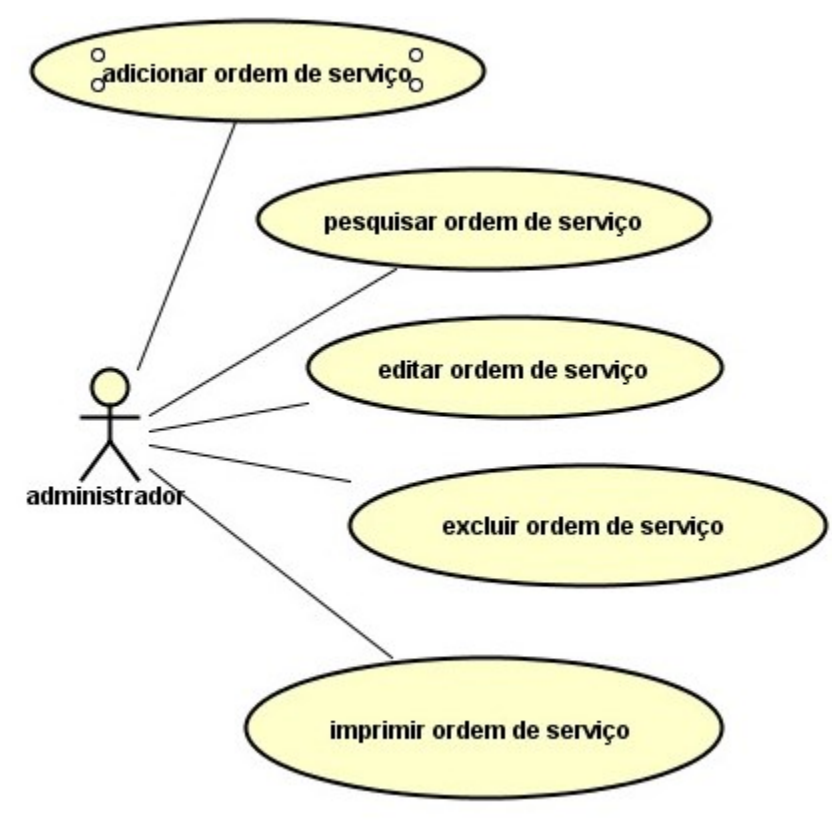
Fonte: Autoral (2022).

Figura 3 - Casos de uso do cliente



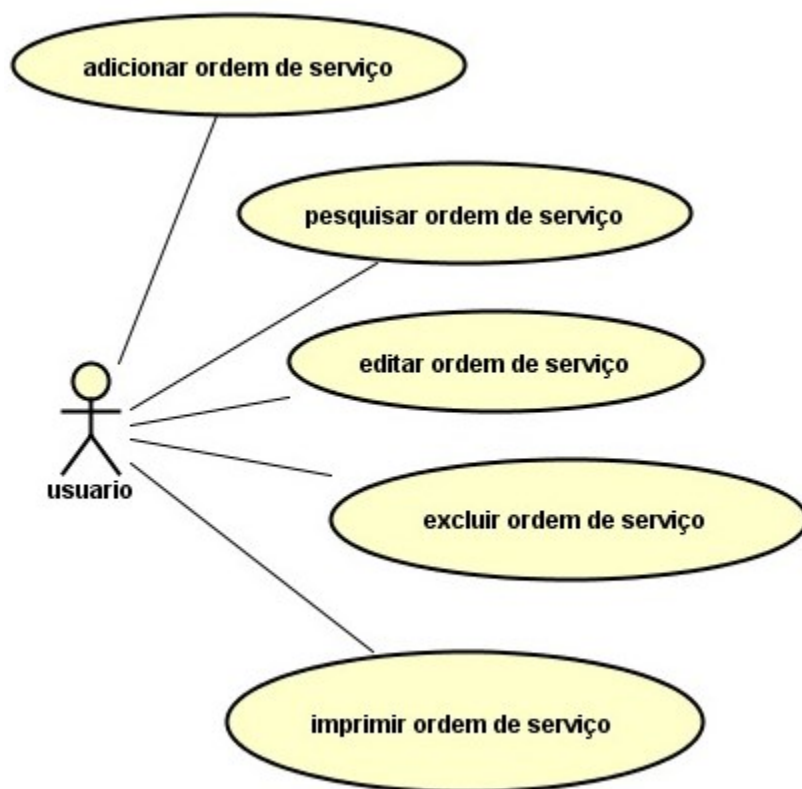
Fonte: Autoral (2022).

Figura 4 - Casos de uso da ordem de serviço administrador



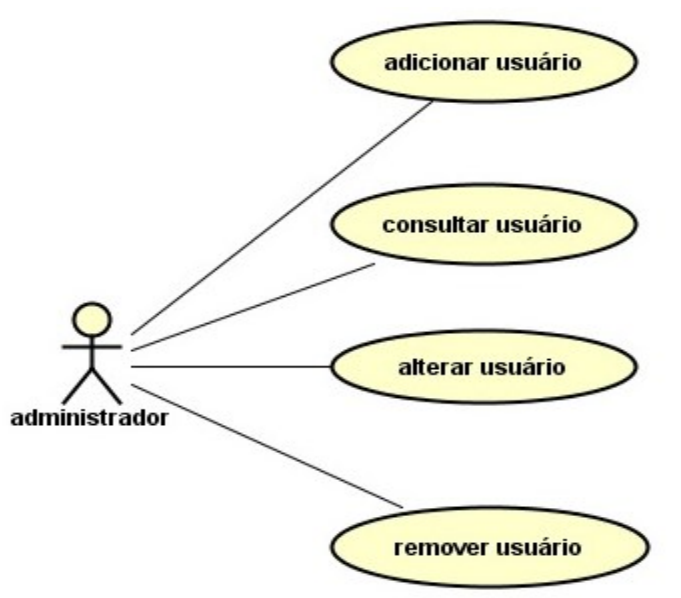
Fonte: Autoral (2022).

**Figura 5** - Casos de uso da ordem de serviço do usuário



Fonte: Autoral (2022).

**Figura 6** - Casos de uso do administrador adicionar usuário



Fonte: Autoral (2022).

### 3.1.1 Documentação do Casos de Uso

A Documentação do caso de uso tem o objetivo a descrição de cada funcionalidade mencionada na Figura 1 a 6, identificando o que é a ação do sistema e ação do ator. tabela 5 até a tabela 11 apresenta a descrição de cada funcionalidade.

Os casos de uso são as ações dos atores do sistema. Sobre estes, afirma Plínio Ventura que “o caso de uso é uma especificação do comportamento de uma funcionalidade. Nele se tem detalhes sobre o funcionamento do sistema, com restrições, premissas e diretrizes pertinentes à funcionalidade do sistema, com restrições, premissas e diretrizes pertinentes à funcionalidade” (VENTURA, 2022). E estão descritas na tabela.

**Tabela 5** - Caso de uso Login

<b>Nome do caso de uso</b>	Efetuar Login
<b>Atores envolvidos</b>	Administrador/Usuário
<b>Objetivo</b>	Este caso de uso descreve o procedimento para o usuário efetuar o login
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do Autor</b>	<b>Ações do Sistema</b>
1.O usuário clica em Login	
2. O usuário insere o user e senha	
	3. O sistema valida o login
4.Iniciar sessão	
	3.B Dados incorretos (mostra mensagem de erro)
<b>Validações</b>	Para que o login seja efetuado, o usuário deve entrar com o seu usuário e senha provido pelo administrador.

Fonte: Autoral (2022).

**Tabela 6 - Caso de uso tela Principal**

<b>Nome do caso de uso</b>	Tela Principal
<b>Atores envolvidos</b>	Administrador
<b>Objetivo</b>	Cadastrar usuário/gerar relatório
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do Autor</b>	<b>Ações do Sistema</b>
1. Cadastro	Cadastrar usuários.
	2.abre a tela com opção cliente, ordem de serviço e usuário.
3.Relatório	
	4.abre opção de relatórios
5. Ajuda	
	6. Mostra opções a realizar
7. Opções	
	8. Sair do sistema.
<b>Validações</b>	Tem acesso somente usuários com login e senha.

fonte: Autoral (2022).

Tabela 7 - Caso de uso Usuário

Nome do caso de uso	Usuário
Atores envolvidos	Usuário
Objetivo	Este caso de uso refere-se a cadastro de usuário, editar usuário e excluir usuário.
Prioridade de desenvolvimento	Essencial
Ações do Autor	Ações do Sistema
1. O administrador clica sobre cadastro	
	2. O sistema mostra a tela para preenchimento dos dados.
3. O administrador digita id, nome do cliente, telefone, login, senha e faz opção do perfil.	
4. O Administrador clica sobre o ícone adicionar	
	5. O sistema adiciona o novo cliente.
6. O administrador digita o id ou o nome e clica em pesquisar.	
	7. O sistema mostra o usuário cadastrado.
8. Na tela de preenchimento o administrador digita os dados a alterar e clicar sobre o ícone editar.	
	9. O sistema faz a alteração dos dados do cliente.
10. O administrador como usuário em tela clica sobre o ícone remover.	
	11. O sistema remove o usuário do sistema.
Validações	Somente o administrador pode remover.

Fonte: Autoral (2022).

Tabela 8 - Caso de uso Cliente

Nome do caso de uso	Cliente
Atores envolvidos	Administrador/User
Objetivo	Este caso de uso refere-se a cadastro cliente, editar cliente e excluir cliente.
Prioridade de desenvolvimento	Essencial
Ações do Autor	Ações do Sistema
1. O usuário clica sobre a opção cliente.	
	2. O sistema mostra a tela para preenchimento dos dados.
3. O usuário tem a opção de <u>inserir</u> nome, endereço, telefone e email do cliente.	
4. O usuário clica sobre o ícone adicionar	
	5. O sistema adiciona o novo cliente.
6. O usuário digita a primeira letra do nome do cliente ou o nome no campo <u>pesquisa</u> .	
	7. O sistema mostra o cliente cadastrado.
8. O usuário clica sobre o nome do cliente que procura.	
	9. O sistema preenche os campos com os dados do cliente.
8. Na tela de preenchimento o usuário digita os dados a <u>alterar</u> e clicar sobre o ícone editar.	
	9. O sistema faz a alteração dos dados do cliente.
10. No campo de pesquisa o usuário digita o nome do cliente que deseja <u>excluir</u> .	
11. O usuário clica sobre o nome na parte superior.	
	12. O sistema preenche os campos com os dados do cliente.
13. O usuário clica no ícone <u>remover</u> .	
	14. O sistema remove o usuário do sistema.
	14.B Cliente com ordem de serviço (o sistema mostra mensagem que não pode excluir cliente)



<b>Validações</b>	Cliente que possui ordem de serviço não pode ser removido.
-------------------	--

Fonte: Autoral (2022).

**Tabela 9 - Caso de uso Ordem de serviços**

<b>Nome do caso de uso</b>	Ordem de serviços/Orçamento
<b>Atores envolvidos</b>	Administrador/Usuário
<b>Objetivo</b>	Formalizar o serviço a ser prestado para um cliente e servir como ponto de partida para a organização do trabalho.
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do Autor</b>	<b>Ações do Sistema</b>
1. O usuário clica em ordem de serviço	
	2. O sistema mostra a tela para preenchimento dos dados.
3. O usuário escolhe a opção orçamento ou OS.	
4. No item Situação o usuário escolhe uma das opções na bancada, entrega realizada, orçamento não aprovado, aguardando aprovação, abandono do cliente, aguardando peças ou retorno	
5. No campo cliente digite o nome do cliente que procura.	
	6. O sistema mostra clientes com os nomes cadastrado
7. O usuário clica sobre o nome que procura.	
	8. O sistema assinala as informações do cliente (fica habilitado para inclusão do orçamento).
9. O usuário clica no botão <u>adicionar</u>	
	10. O sistema adiciona o orçamento ou OS no sistema
11. O usuário clica em <u>pesquisar</u> e digita o número da ordem de serviço.	
	12. O sistema pesquisa e mostra a ordem de serviço se não constar mostra a mensagem não cadastrada.
13. O usuário clica no botão <u>excluir</u> OS.	

	14. o sistema exclui a ordem de serviço/orçamento.
15. Para <u>criar</u> ordem de serviço o usuário deve seguir o passo do item 8 acima.	
16. O usuário escolhe a opção do item 4 e preenche o campo Equipamento, defeito, serviço, técnico, valor e clicar sobre o botão adicionar.	
	17. O sistema adiciona a ordem de serviço
18. Ao clicar em clientes tem a opção de pesquisar, editar, excluir e imprimir ordem de serviço/ orçamento seguindo os passos do item 4 e 8 acima.	
<b>Validações</b>	Somente cria ordem de serviço para Cliente cadastrado

Fonte: Autoral (2022).

**Tabela 10** - Caso de uso acessar relatório

<b>Nome do caso de uso</b>	<b>Acessar Relatório</b>
<b>Atores envolvidos</b>	Administrador
<b>Objetivo</b>	Mostra o procedimento para o administrador acessar os relatórios gerados no sistema (iReport).
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do Autor</b>	<b>Ações do Sistema</b>
1. O Administrador entra com a opção relatório	
	2. O sistema mostra as tabelas gerado automaticamente*
3. O administrador tem acesso às informações	
<b>Validações</b>	O acesso às informações só é permitido através do login do administrador.

Fonte: Autoral (2022).

Tabela 11 - Caso de uso Sobre

<b>Nome do caso de uso</b>	<b>Sobre</b>
<b>Atores envolvidos</b>	Administrador/Usuário
<b>Objetivo</b>	Mostra a licença open source
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do Autor</b>	<b>Ações do Sistema</b>
1.Dar um click sobre Licença GPL	
	2.Mostrar a licença GPL
<b>Validações</b>	O acesso aos desenvolvedores tem somente com login e senha.

Fone: Autoral (2022).

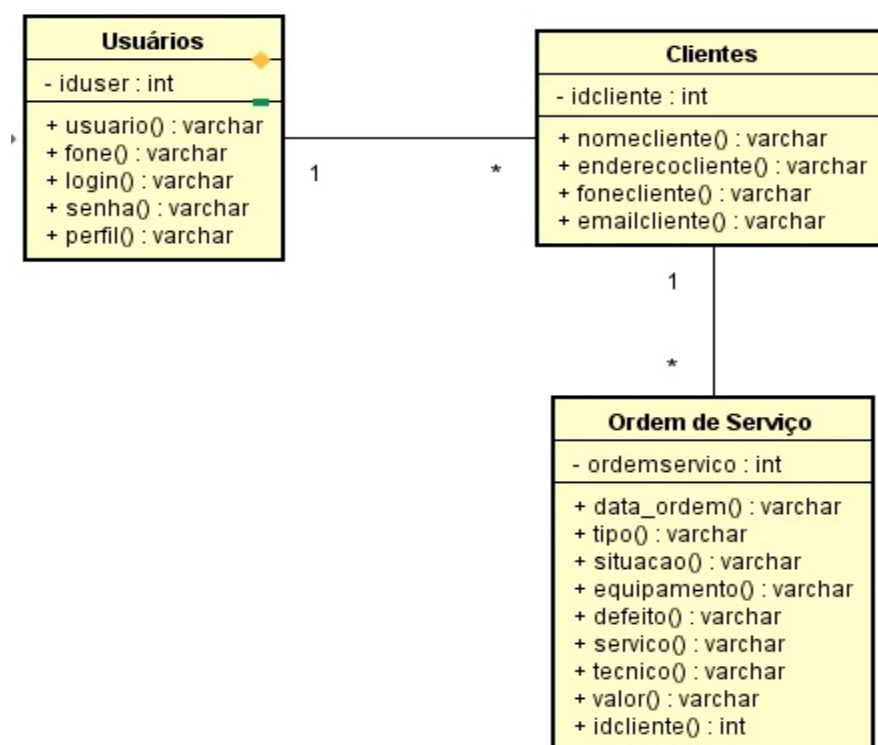
## 3.2 DIAGRAMA DE CLASSE

Uma classe é um módulo de software que fornece uma abstração de algo concreto. A classe fornece similar de objetos, denominados instâncias.

O diagrama de classe é uma ferramenta importante para os requisitos de análise e desenho de sistemas orientados a objetos. Uma classe é abstração de um conjunto de objetos que possuem os mesmos tipos de características e comportamentos, sendo representada por um retângulo que pode possuir até três divisões. A primeira divisão armazena o nome da classe; a segunda, os atributos (características) da classe; e a terceira os métodos (PASCUTTI, 2016).

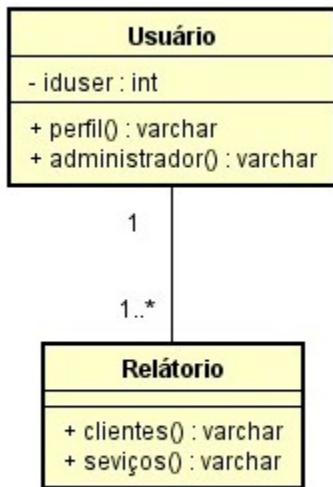
O diagrama de classe é utilizado para representação das classes e suas relações e funcionalidades disponíveis aos seus atores utilizarem. As figuras representam a estrutura das classes do sistema.

**Figura 7** - Diagrama de Classe: Usuário



**Fonte:** Autoral (2022)

**Figura 8** - Diagrama de Classe: Administrador



**Fonte:** Autoral (2022)

### 3.3 BANCO DE DADOS

O banco de dados utiliza-se para armazenamento de dados estruturados ou coleção de informações organizadas que se relacionam para criar um dado que faça sentido, provendo mais eficiência durante uma pesquisa ou análise científica.

O SGBD (Sistema de Gerenciamento de Bancos de Dados) é um sistema que dispõe de recursos que possibilitam a interação do usuário com o banco de dados.

Os principais bancos de dados fazem uso de uma modelagem baseado em linhas e colunas.

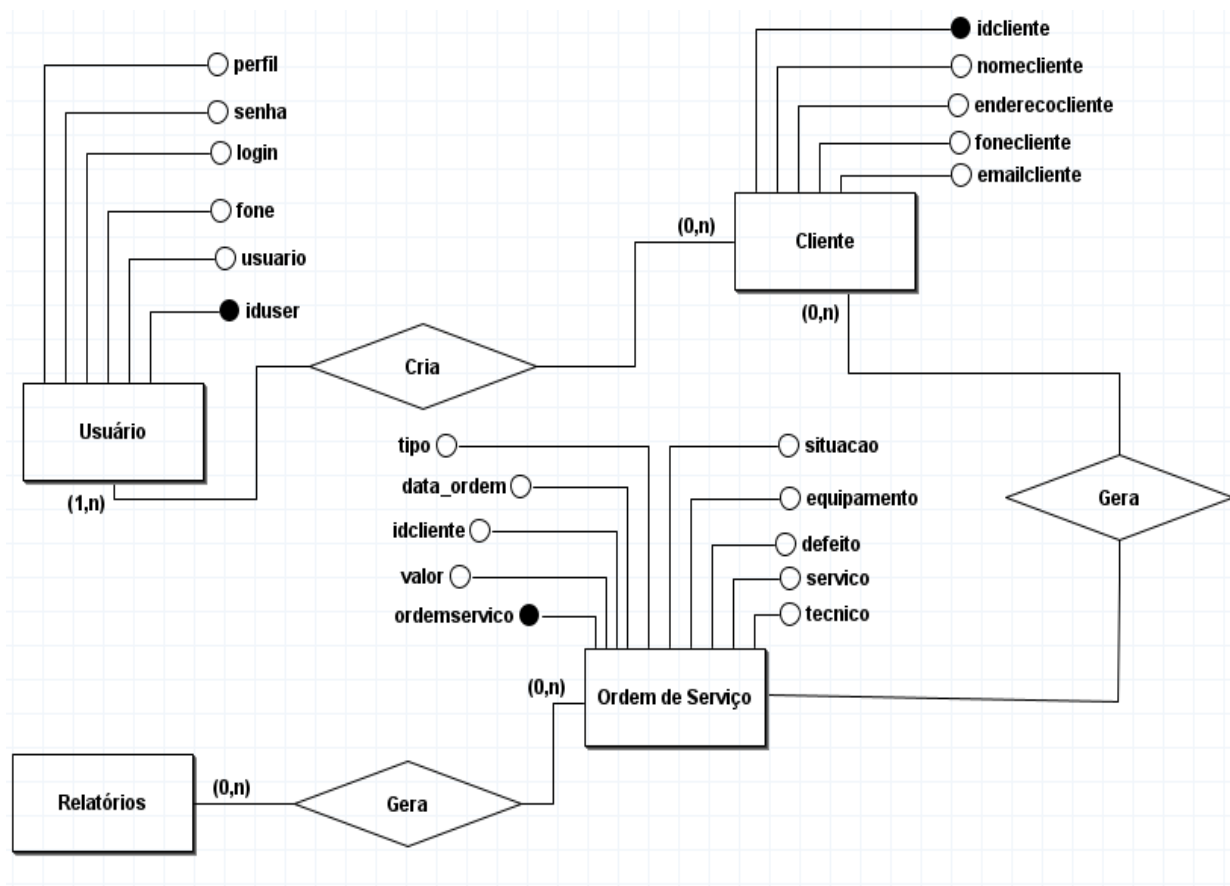
#### 3.3.1 Diagrama de Entidade e Relacionamento

Diagrama de entidade e relacionamento representa a forma que as entidades se relacionam entre si, como estes conjunto de objetos são representados por entidades, atributos e relacionamentos. o diagrama mostra uma visão lógica de como o banco de dados está estruturado com três componentes fundamentais são:

- Entidades: Objetos relacionados ao domínio e que podem armazenar dados.
- Relacionamento: Como as entidades atuam ou são associadas com as demais.
- Atributos: Características das entidades.

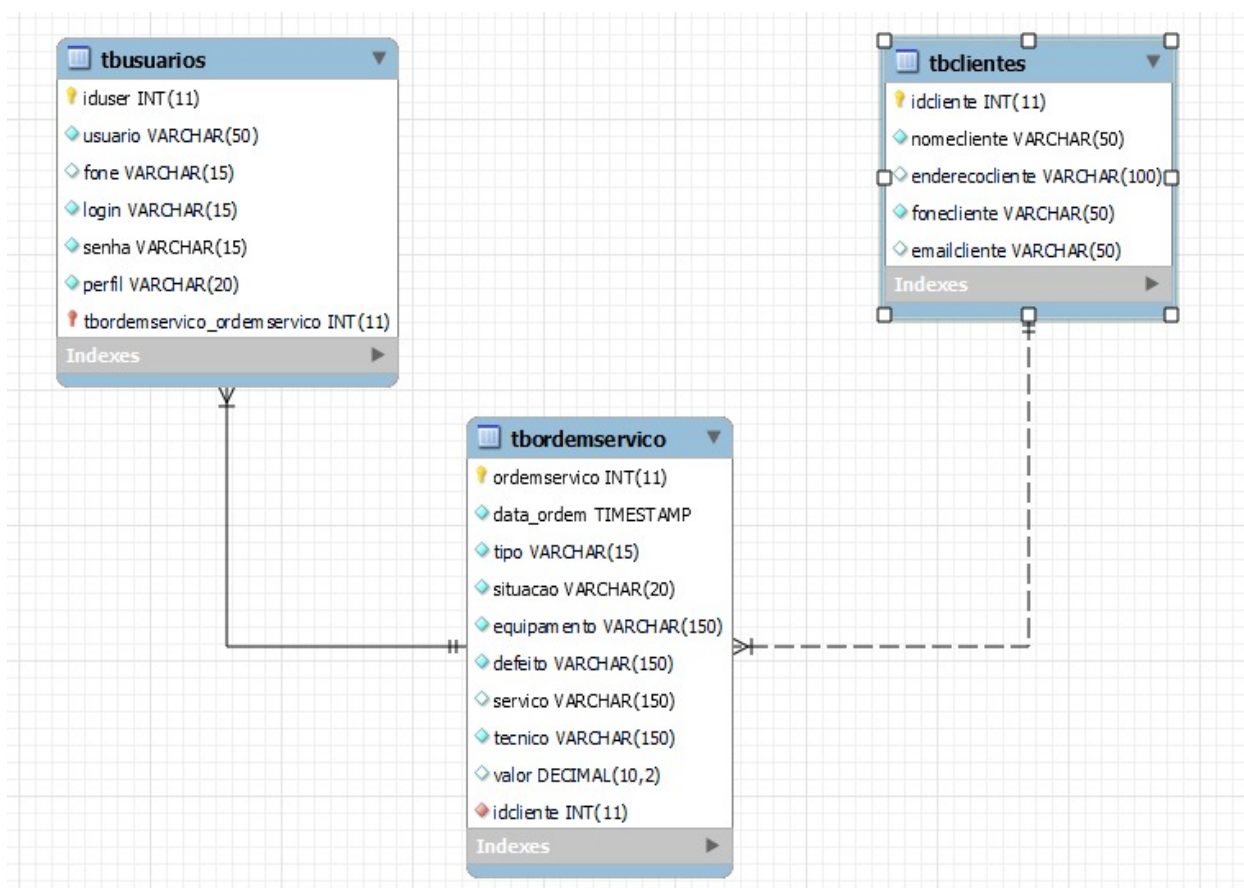
As figuras **9** e **10** a seguir apresentam essa estrutura.

Figura 9 - Diagrama Entidade e Relacionamento:



Fonte: Autoral (2022).

Figura 10 - Banco de dados



Fonte: Autorial (2022).



## 4 DESENVOLVIMENTO

O desenvolvimento utilizado neste projeto a metodologia Scrum já que podem ser feitos ajustes sem que atrapalhe o projeto final, onde os ciclos de desenvolvimento tendo objetivos definidos, houve imprevisto solucionados durante o desenvolvimento e na entrega.

### 4.1 ETAPAS DE DESENVOLVIMENTO

O planejamento para entregas das partes em cada entregável foi realizado de acordo com a evolução do programa que aconteceu em períodos pequenos até formarem os dias planejado para cada uma das entregas que é realizada, com o grau de dificuldades apresentada de início estipulamos os dias conforme o cronograma em dias para cada uma das entregas 16,30,8,29 e 7 dias abaixo: Com as seguintes especificações:

- **Entrega 1:** Discutir o projeto e realizar o estudo de viabilidade do negócio (16 dias).
- **Entrega 2:** Codificação do banco de dados, criação de tabelas e testes (30 dias).
- **Entrega 3:** criação de Usuário restrito, tela de Usuário, tela de cliente e ordem de serviço (8 dias).
- **Entrega 4:** Ajustes testes e correções (29 dias).
- **Entrega 5:** Implementação da impressão de orçamento e ordens de serviços (7 dias).

#### 4.1.1 Entrega 1

Inicialmente houve uma reunião no dia 30/08/2022 na Fatec de Americana onde surgiram ideias para o desenvolvimento do projeto. Em pauta estavam três ideias para projetos, sendo elas um site para salão de beleza, um software para gestão de biblioteca e pôr fim a ideia que mais agradou foi a de desenvolvimento de um sistema gerenciador de lojas de informática e periféricos, na primeira entrega foi realizado um levantamento do grau de dificuldade de cada atividade mapeada para este entregável, representado por pontos, onde o número 1 representa um grau de dificuldade baixo e o 10, alto.

A Tabela 12 representando o mapeamento das atividades, seu tempo de realização em dias e sua respectiva pontuação.

**Tabela 12** – Planejamento do primeiro entregável

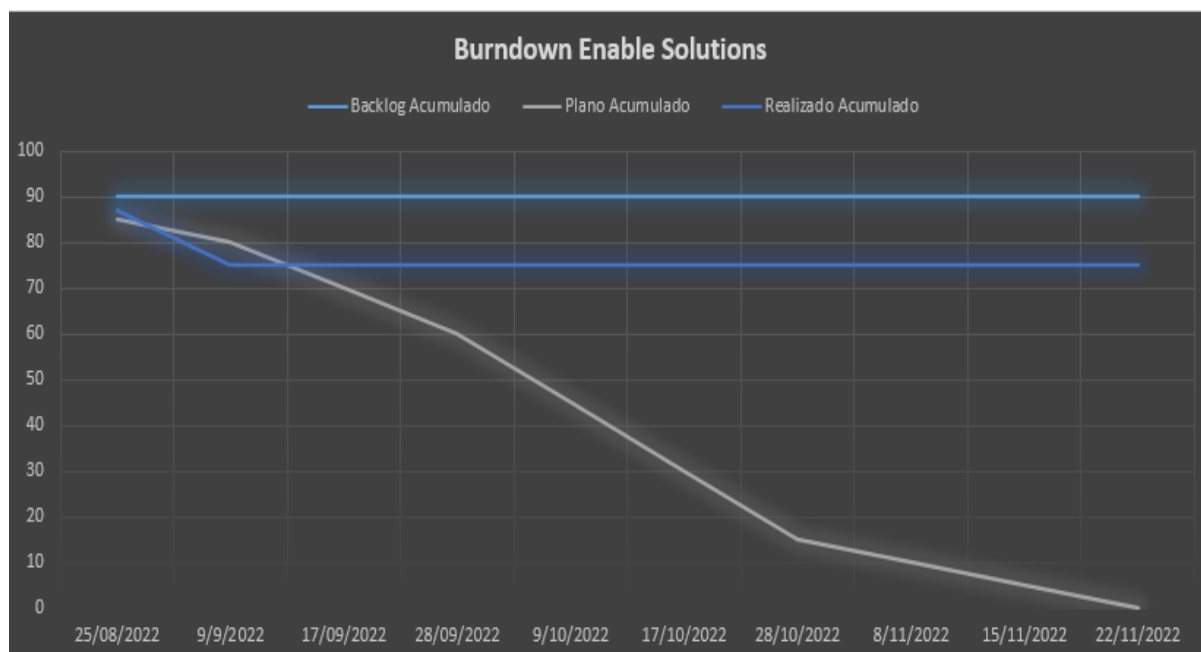
Atividades	Tempo em dias	Pontos
Discutir o projeto	4	6
Viabilidade do projeto	8	10
<b>Total atingido</b>	<b>12</b>	<b>16</b>
Total projetado	16	16

**Fonte:** Autoral (2022).

No andamento do primeiro entregável foi realizado reuniões na escola e cada um fez a pesquisa de viabilidade do projeto de gerenciamento para lojas de informática. No final deu se os pontos ao que deu certo (ou não) e possíveis melhorias:

- **O que deu certo:** Com os desafios da etapa atingimos esta fase com o tempo esperado nas pesquisas tivemos um pouco de dificuldade em relação a estudo de viabilidade para o negócio, mas conseguimos solucionar dentro do tempo.
- **O que deu errado:** As reuniões para discutir em um horário igual para todos não foi possível:
- **Ações de melhorias:** propomos outro meio para nós realizar as reuniões do grupo através do aplicativo WhatsApp e o Google Meet.

A figura 11 representa o gráfico de *burndown* desta entrega.



Fonte: Autoral (2022).

#### 4.1.2 Entrega 2

Conseguimos Solucionar os desafios desta etapa e atingir esta fase com o maior tempo esperado pois já sabíamos da dificuldade e foi projetado um tempo maior onde tivemos que pesquisar na documentação do MySQL por não dominar muito bem esta linguagem criação das tabelas foi um processo bem complexo, um pouco de dificuldade em relação a sintaxe da linguagem;

A Tabela 13 representando o mapeamento das atividades, seu tempo de realização em dias e sua respectiva pontuação.

**Tabela 13** – Planejamento do segundo entregável

Atividades	Tempo em dias	Pontos
Codificação do banco de dados, criação da tabela de usuários e teste da tabela usuários.	2	4
Criação da tabela clientes, teste da tabela clientes.	4	2
Criação da tabela de ordem de serviços, teste da ordem de serviços.	4	2
Teste da conexão do banco de dados, estabelecendo conexão tela login com o banco de dados.	4	3
Criação da tela login, codificação da tela login, teste da tela login.	4	3
Criação da tela principal, criação da tela usuários e codificação da tela usuário.	4	2

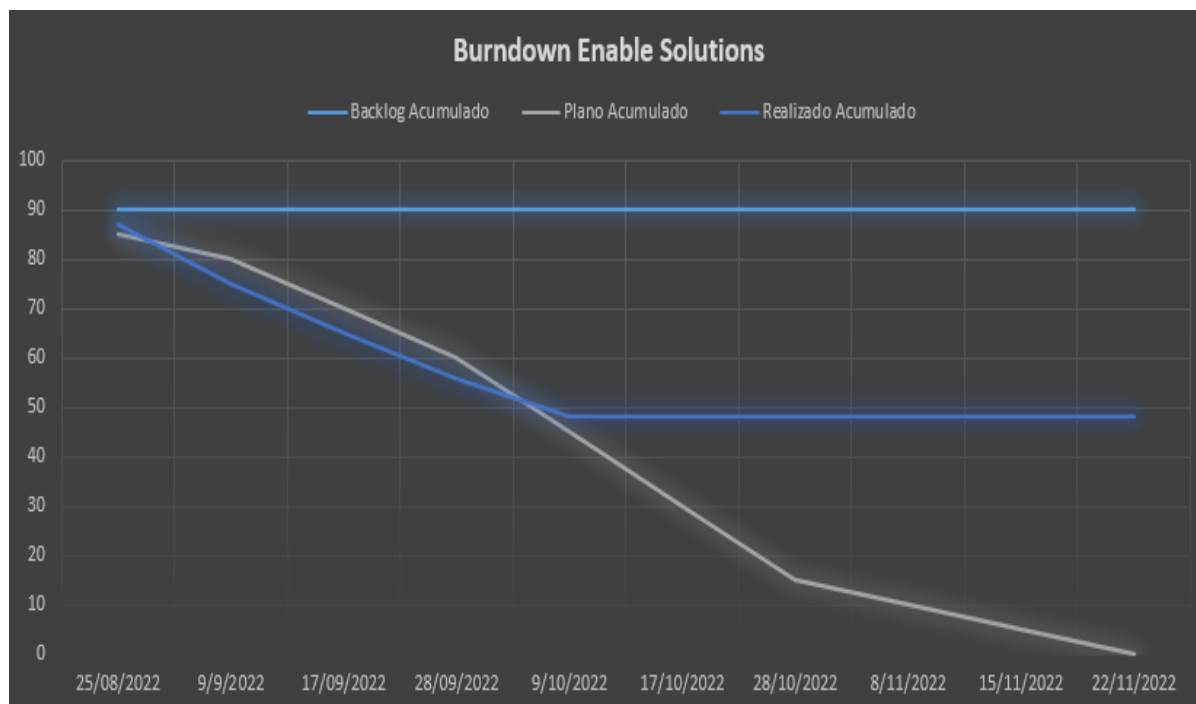
Estabelecendo conexão tela de usuários com o banco de dados e teste da tela usuários.	4	3
Criação da tela clientes, teste da tela cliente e codificação da tela clientes	4	3
Estabelecendo conexão da tela cliente com o banco de dados, teste da tela cliente, criação da tela ordem de serviços e codificação da tela ordem de serviços.	4	3
Estabelecendo conexão tela de ordem de serviços com o banco de dados, teste da tela ordem de serviços, criação da tela de relatórios de serviços e criação da tela de relatórios de clientes.	4	3
Estabelecendo conexão tela de ordem de serviços com o iReport, criação da tela de relatórios no iReport e teste de relatórios no iReport.	7	0
<b>Total atingido</b>	<b>45</b>	<b>28</b>
Total projetado	30	30

**Fonte:** Autoral (2022).

No andamento do segundo entregável foi mantido a comunicação das reuniões pessoalmente e através do WhatsApp e no final deu se os pontos ao que deu certo (ou não) e possíveis melhorias:

- **O que deu certo:** A etapa de criação do Banco foi realizada com sucesso tendo êxitos nas criações das tabelas e as codificando todas elas e fazendo a ligação entre as Banco de dados com a ide NetBeans.
- **O que deu errado:** Relatório não abre automaticamente pela falta de compatibilidade do apache com o iReport, porém a conexão do sistema com o iReport não está funcionando devido à falta de compatibilidade dos pacotes, o iReport está conectado apenas com o banco de dados só não abre, mas mantém as informações do sistema normalmente:
- **Ações de melhorias:** buscar por mais conhecimento na linguagem, procurar outra ferramenta que possa solucionar este problema.

A figura 12 representa o gráfico de *burndown* desta entrega.



Fonte: Autoral (2022).

#### 4.1.3 Entrega 3

Com o Processo de Teste de Software e o seu objetivo atingido estruturou as etapas, as atividades, os papéis e as responsabilidades do teste, permitiu organizar e controle de todo o ciclo do teste, minimizou os riscos e agregando valor ao software objetivo foi atingido.

A Tabela 14 representando o mapeamento das atividades, seu tempo de realização em dias e sua respectiva pontuação.

**Tabela 14** – Planejamento do terceiro entregável

Atividade	Tempo em dias	Pontos
Criando usuários administrador, criando usuário restrito. acesso tela usuário, adicionando usuário, alterando perfil usuário, pesquisando usuário e apagando usuário.	2	2
Criando usuário restrito, acesso tela usuário, adicionando usuário, alterando perfil usuário, pesquisando usuário e apagando usuário.	2	2
Acesso tela cliente, adicionando cliente, alterando cliente. pesquisando cliente, apagando cliente.	2	2

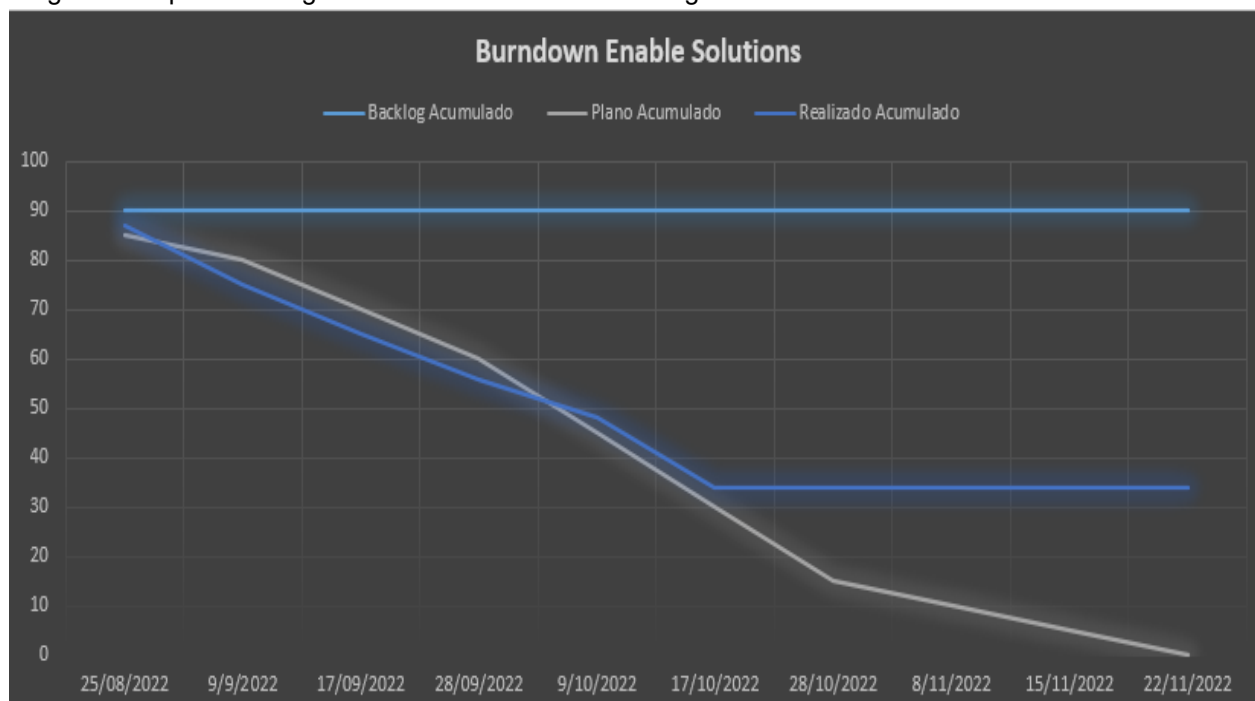
Criando ordem de serviço, criando status da ordem de serviço, criando ordem de serviço, alterando ordem de serviço, pesquisando ordem de serviço, imprimindo ordem de serviço, apagando ordem de serviço.	2	2
<b>Total atingido</b>	<b>8</b>	<b>8</b>
Total projetado	8	8

Fonte: Autoral (2022).

Esse terceiro entregável é consideravelmente pequeno com relação aos demais, foi realizado os testes de todo o projeto, obtendo os seguintes resultados, no final deu se os pontos ao que deu certo (ou não) e possíveis melhorias:

- **O que deu certo:** Criação de usuário, o acesso a tela do usuário adicionar alterar, pesquisar e apagar usuário.
- **O que deu errado:** o iReport não teve a conexão com o sistema impedindo a exibição automática do relatório.
- **Ações de melhorias:** Procurar novos pacotes compatível para a conexão entre o NetBeans e o iReport.

A figura 13 representa o gráfico de burndown desta entrega.



Fonte: Autoral (2022).

#### 4.1.4 Entrega 4

No andamento do quarto entregável foi realizado reuniões através do WhatsApp, esta entrega consiste em ajustes de pacote, porém sem êxito em encontrar a emissão do relatório direto na tela. refez o e teste do projeto e a revisão.

A Tabela 15 representando o mapeamento das atividades, seu tempo de realização em dias e sua respectiva pontuação.

**Tabela 15** – Planejamento do quarto entregável

<b>Atividades</b>	<b>Tempo em dias</b>	<b>Pontos</b>
Ajuste	12	12
Teste do projeto	7	7
Revisão no projeto	10	10
<b>Total atingido</b>	<b>29</b>	<b>29</b>
Total projetado	29	29

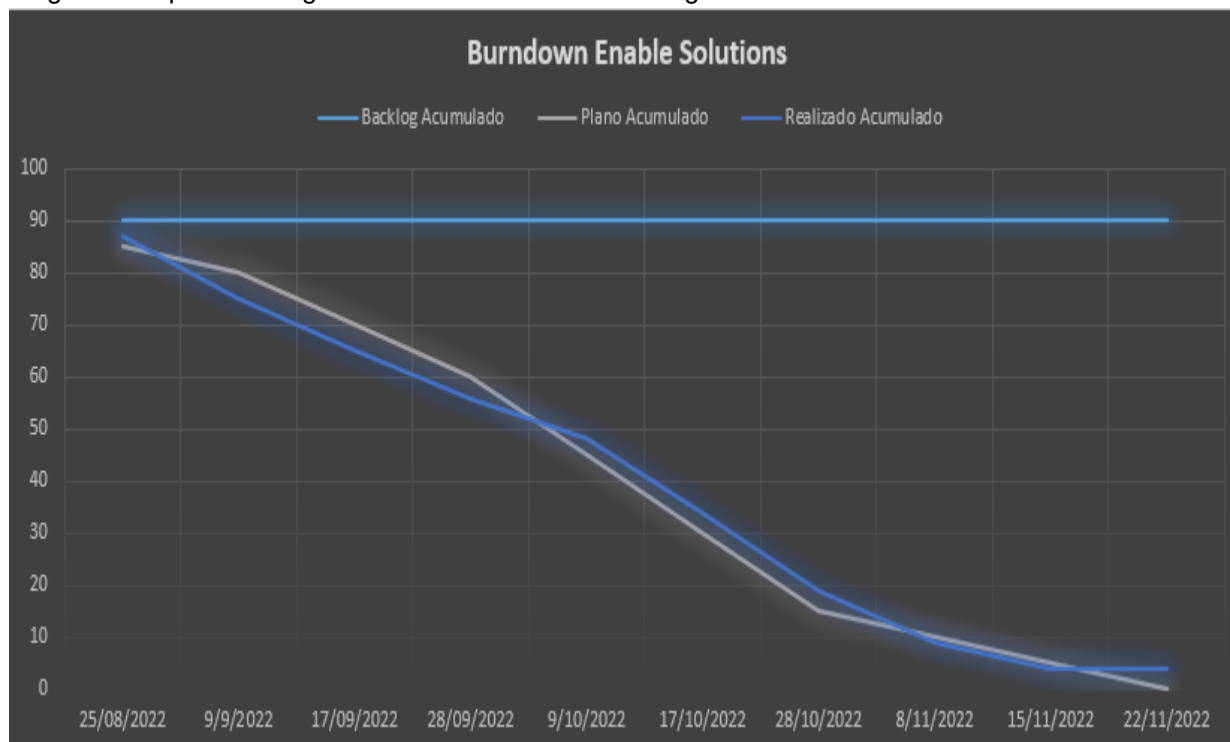
**Fonte:** Autoral (2022).

Foi necessário ajuste na codificação da tela devido o erro apresentado por pacote incompatível, não teve êxito com pacote mostrou a incompatibilidade. Refez o teste do Sistema e revisão do projeto.

No final deu se os pontos ao que deu certo (ou não) e possíveis melhorias:

- **O que deu certo:** tivemos êxito ao fazer ajuste na codificação com o pacote correto.
- **O que deu errado:** As reuniões para discutir em um horário igual para todos não foi possível:
- **Ações de melhorias:** propomos outro meio para nós realizar as reuniões do grupo através do aplicativo WhatsApp e Pessoalmente.

A figura 14 representa o gráfico de burndown desta entrega.



Fonte: Autoral (2022).

#### 4.1.5 Entrega 5

Nesta fase do projeto faz as devidas verificações analisando o que está no conforme e o que deu certo e o que alcançou neste projeto ou precisa ser melhorado e evitar erros futuros alcançando valor ao sistema

**Tabela 16** – Planejamento do quinto entregável

Atividades	Tempo em dias	Pontos
Revisão final do projeto	7	7
<b>Total atingido</b>	<b>7</b>	<b>7</b>
Total projetado	7	7

Fonte: Autoral (2022).

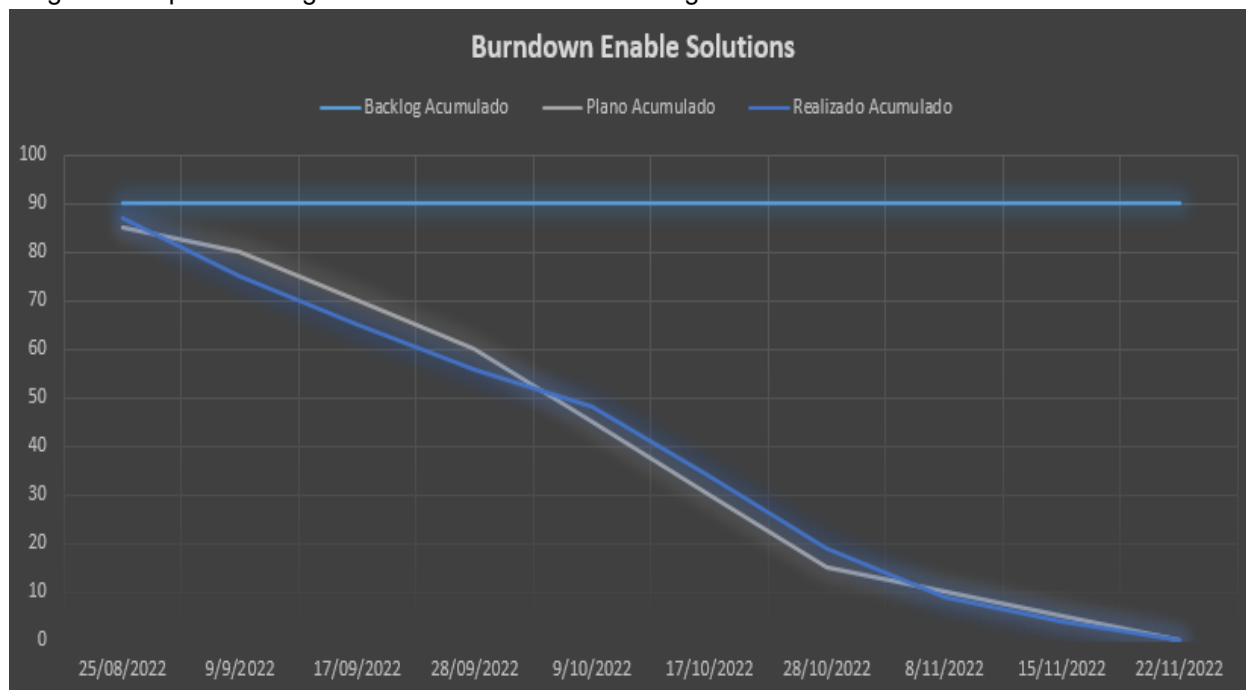
No andamento do quinto entregável foi realizado reuniões na escola realizado revisão final com testes. No final deu se os pontos ao que deu certo (ou não) e possíveis melhorias:

- **O que deu certo:** Funcionando todas as funcionalidades do sistema.



- **O que deu errado:** não emite relatório automático ao clicar no botão relatórios.
- **Ações de melhorias:** buscar novas tecnologias que exerce a mesma função para emissão automática do relatório.

A figura 15 representa o gráfico de burndown desta entrega.

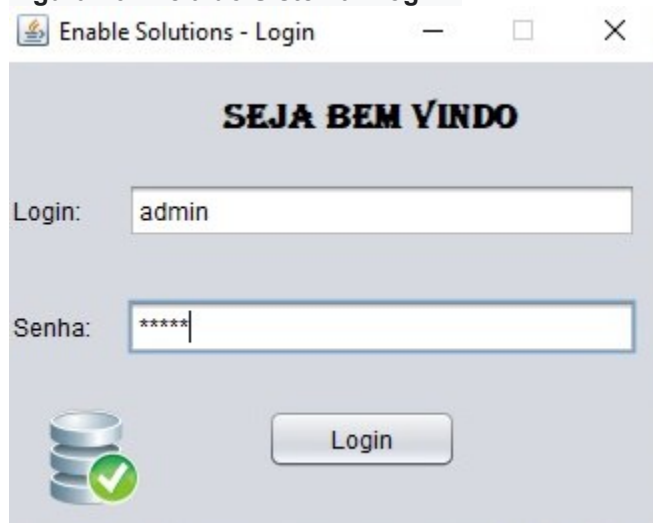


Fonte: Autoral (2022).

## 4.2 INTERFACE DO SISTEMA

A figura 16 abaixo refere-se à tela de login do sistema. A realização de autenticação no sistema foi configurada junto ao banco de dados MYSQL do servidor Windows, para o usuário ter o acesso ao sistema ele necessita ter o servidor mysql local ou remoto. O administrador de TI cria usuários para ter permissão quando requisitado ao sistema.

**Figura 16 - Tela do Sistema: Login**



The screenshot shows a login window titled "Enable Solutions - Login". At the top, it says "SEJA BEM VINDO". Below that, there are two input fields: "Login:" containing the text "admin" and "Senha:" containing five asterisks. A "Login" button is positioned below the password field. To the left of the button is an icon representing a database with a green checkmark.

Fonte: Autoral (2022).

Na Tela Principal do sistema no menu superior possui todas as funcionalidades. No menu cadastro realiza a base do sistema, cliente, ordem de serviço e usuários, conforme a figura 17.

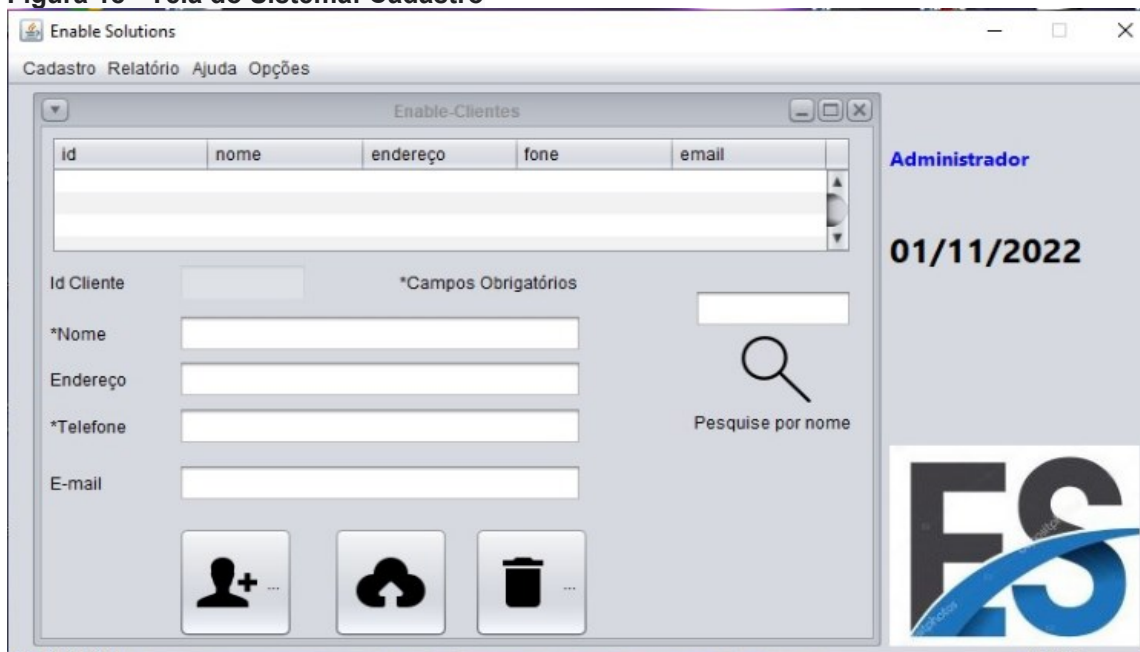
**Figura 17 - Tela do Sistema: Principal**



Fonte: Autoral (2022).

Ao selecionar a tela Cadastro de clientes tem os campos para ser preenchido, conforme a figura 18:

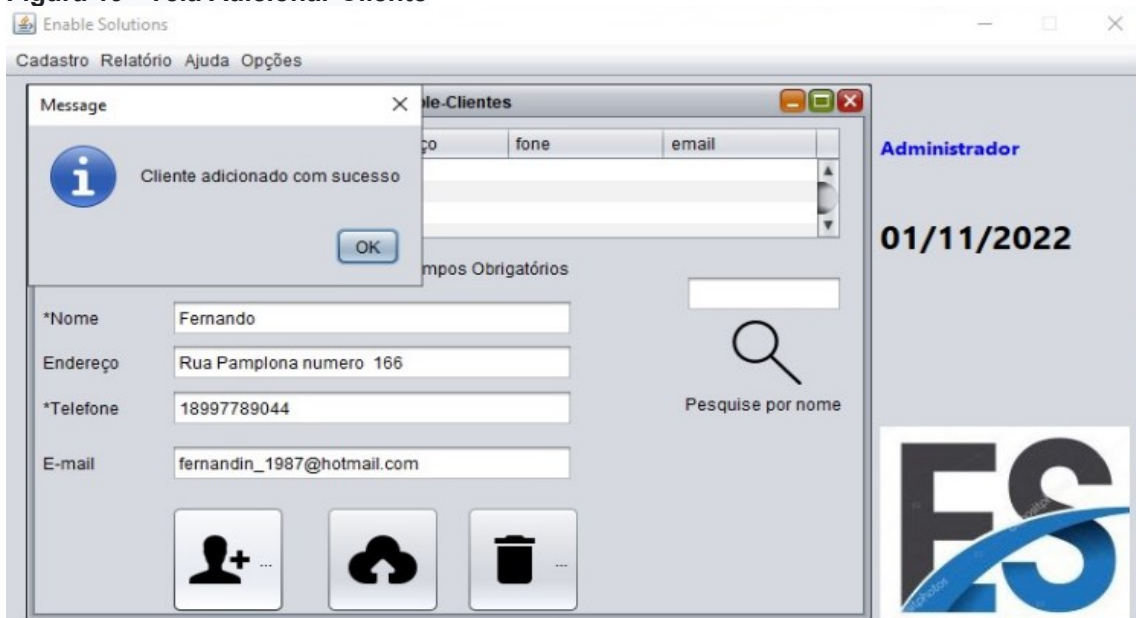
**Figura 18 - Tela do Sistema: Cadastro**



Fonte: Autoral (2022).

Com todos os campos preenchidos vá no ícone adicionar o novo cliente ao sistema, conforme a figura 19:

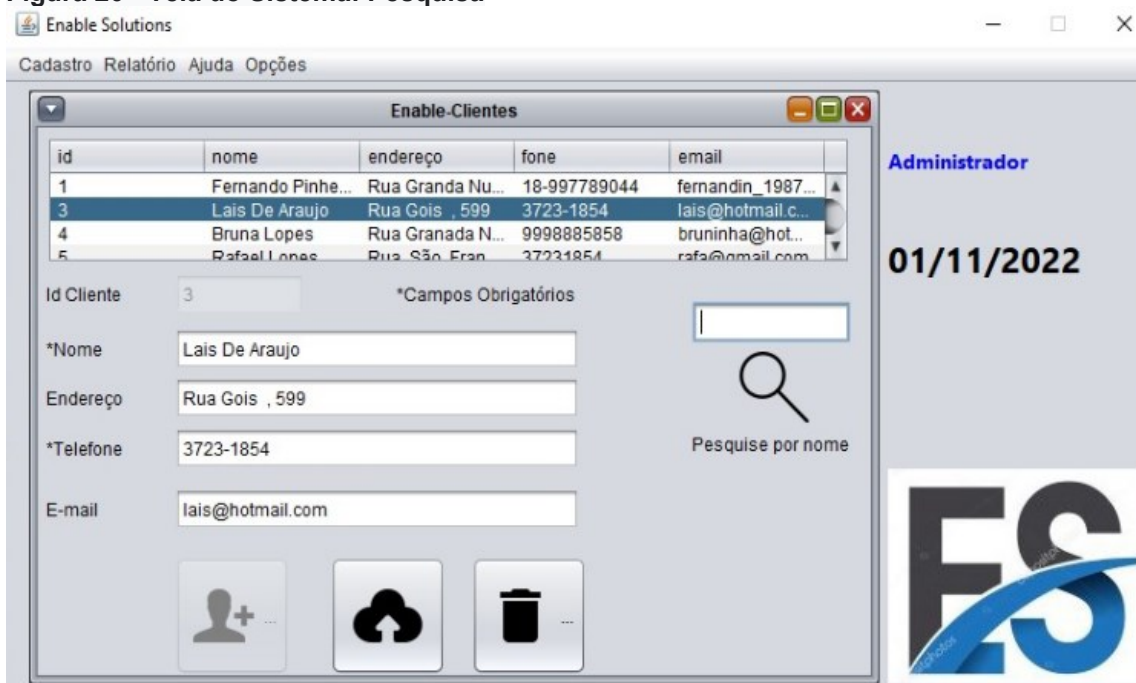
**Figura 19 - Tela Adicionar Cliente**



Fonte: Autoral (2022).

Nesta tela faz a pesquisa de cliente que já tem cadastro (com uma letra) mostra o nome do cliente, conforme figura 20:

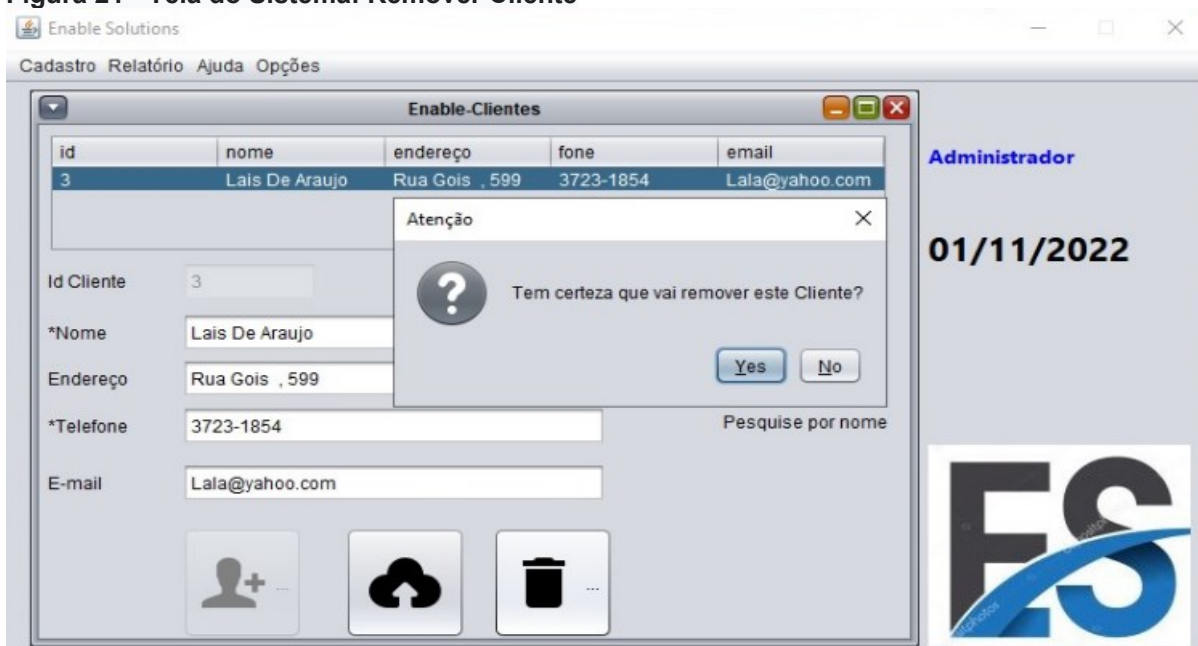
**Figura 20 - Tela do Sistema: Pesquisa**



Fonte: Autoral (2022).

Com o cliente selecionado no ícone remover após confirmação o cliente é removido, conforme figura 21:

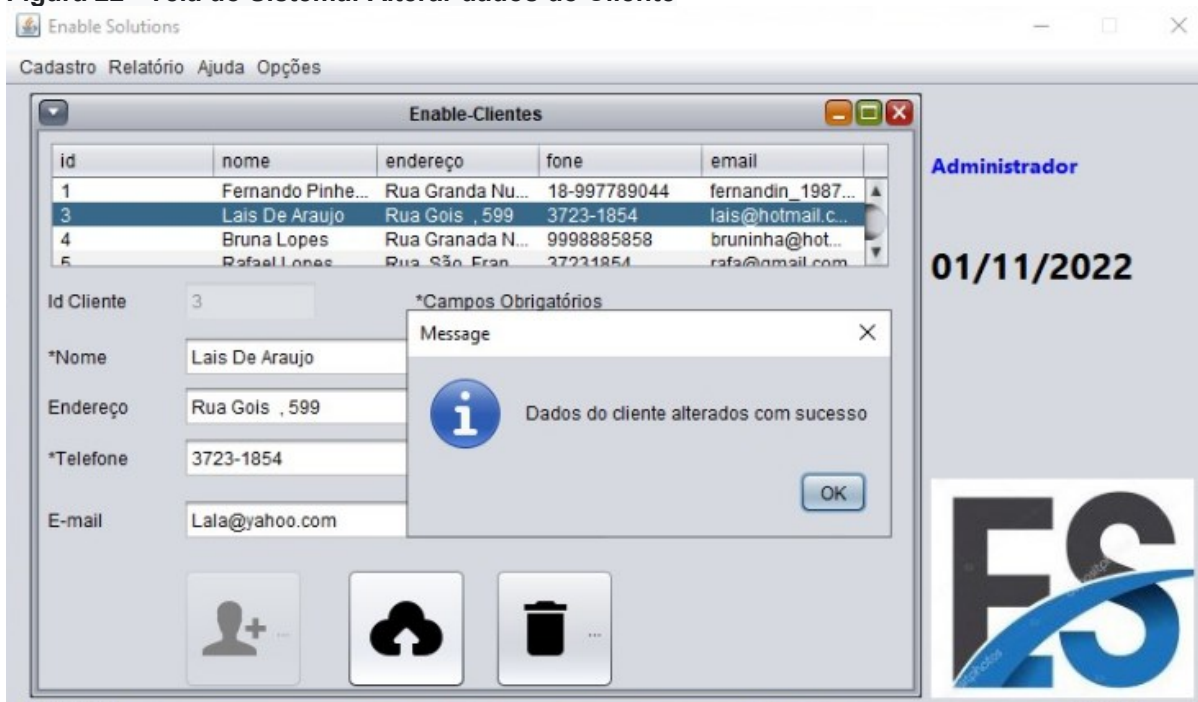
**Figura 21 - Tela do Sistema: Remover Cliente**



Fonte: Autoral (2022).

Na aba cliente com o cliente selecionado após as devidas mudanças e confirmar os dados do cliente será alterado com sucesso, conforme figura 22:

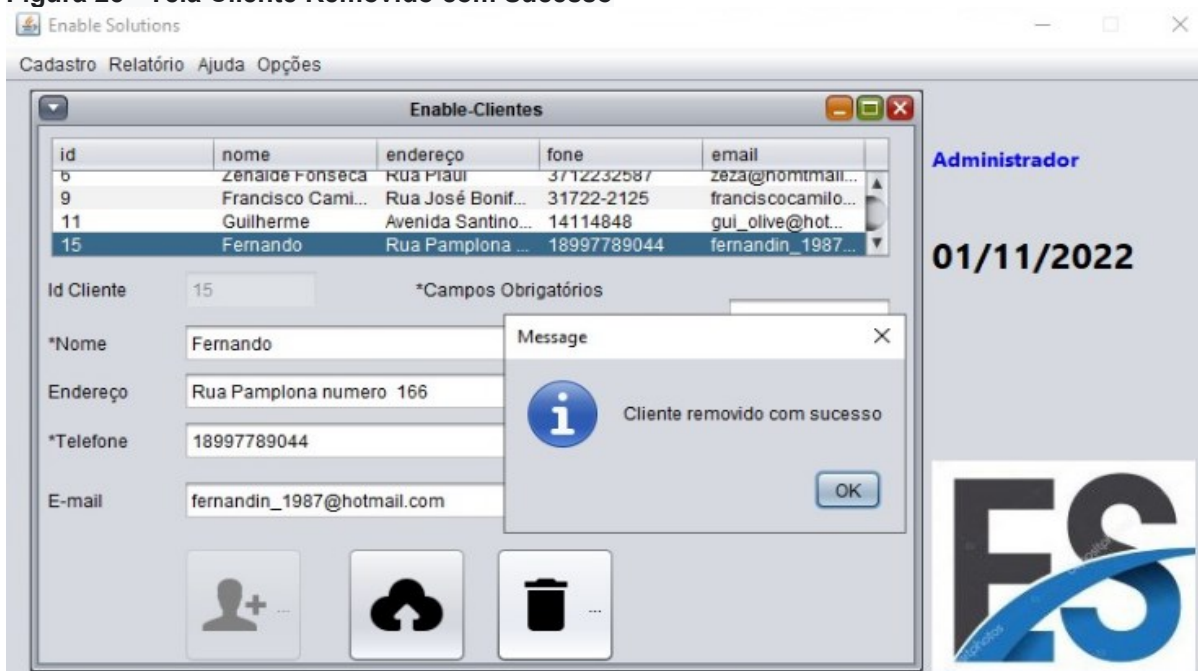
**Figura 22 - Tela do Sistema: Alterar dados do Cliente**



Fonte: Autoral (2022).

Nesta aba cliente com o cliente selecionado e apertar o botão remover o cliente será removido com sucesso.

**Figura 23 - Tela Cliente Removido com Sucesso**



Fonte: Autoral (2022).

O cliente que só poderá ser removido se não tiver nenhuma ordem de serviço.

**Figura 24 - Tela não Remoção de Cliente**



Fonte: Autoral (2022).

## 5 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo final, obtenção de nota para aprovação no projeto de conclusão de curso, para finalização da graduação no curso de análise e desenvolvimento de sistemas.

O trabalho inicialmente teve a ideia de realizar uma aplicação que auxiliasse no trabalho do dia a dia da loja de manutenção de microcomputadores e periféricos, que propôs o desenvolvimento de um sistema de gerenciamento para loja de informática que auxilia na gestão quanto entradas de equipamentos, controle e saída, visa gerar maior facilidade no interior da loja quanto as informações de clientes, ordem de serviços e tramites das mesmas diminuindo tempo de resposta no levantamento de informações solicitadas por usuário do sistema ou o gestor.

Para tanto, desenvolveu-se um sistema informatizado que acarretou a toda as vantagens citadas acima, com a utilização de tecnologias Open Source foi de grande importância pois apesar de existirem sistemas á venda no mercado, onde o empreendedor deste ramo como o micro empreendedor que realiza no brasil mais de 70% destes consertos não ter grandes recursos financeiros para adquirir e ter uma solução aplicada as características da mesma.

Porém ao decorrer o tempo foi notado que poderia alcançar mais com esta aplicação que foi possível visualizar uma oportunidade de negócios como prestar assistência com o software aos MEI na aplicação de determinada loja com a sua logomarca e podendo ser aplicado em outros comércios que use este sistema ou até mesmo realizando aplicação de melhorias já que a linguagem escolhida roda em qualquer plataforma.

Nesta sequência, é relatado todo o processo desde o começo que surgiu a ideia, foi definido cada fase com o que é bom e o que é ruim sendo acompanhado pelo nosso orientador responsável e apontando melhorias.

Sobre o sistema, tivemos várias dificuldades e partimos para as buscas ao conhecer, e achar a solução teve partes que precisamos refazer para vencer etapas, mas que nos trouxe aprendizados que podemos levar para nossa vida profissional, conseguimos finalizar com êxito, um projeto que nos deu satisfação com a conclusão.

O Software cumpre o que promete, mesmo com todos os desafios que passou em aplicar suas funcionalidades usamos fontes com bases solidas e confiáveis para conseguirmos ter êxito e alcançar o objetivo para entregar o sistema final de conclusão do curso.

Apresentado um sistema que surgiu a partir do pós-pandemia, com a ideia de atender a demanda nas manutenções de microcomputadores que foi adquirido para a suprir as necessidades dos trabalhos e aulas em home office fazendo o uso de desktops para acesso à internet.

Com desafio do projeto proposto, encontramos através da tecnologia várias opções a seguir através do curso de Análise e desenvolvimento de Sistemas, usamos uma linguagem de programação Java, o software que atendeu a necessidade apontada no início que resultou em um sistema real, que pode como propostas futuras realizar implementações de algumas funcionalidades para facilitar ainda mais o trabalho no dia a dia dos serviços prestado como implementação de cupom fiscal, envio de e-mail e o envio de SMS, que é possível serem analisadas e implementadas.

Conclui-se com este trabalho a percepção da importância da existência de sistemas que facilite e agilize a vida das pessoas cada vez mais.



## 6 REFERÊNCIAS

COMMUNITY, Astah Uml. Diagrama. **diagrama**, 2018. Disponível em: <https://astah.net/products/astah-community/>. Acesso em: 12 set. 2022.

DESENVOLVIMENTOAGIL. **Scrum**. 2013/2014. Disponível em: <http://www.desenvolvimentoagil.com.br/scrum/>. Acesso em: 05 set. 2022.

IDC, BRASIL. **Estudo da IDC Brasil aponta que mercado brasileiro de PCs cresceu 37% em 2021**, 2022. Disponível em: <https://www.idc.com/getdoc.jsp?containerId=prLA49041522>. Acesso em: 10 ago. 2022.

INSIDE, TI. **Em meio à pandemia, mercado de PCs cai 12,3% no primeiro trimestre de 2020**, 2020. Disponível em: <https://tiinside.com.br/20/04/2020/em-meio-a-pandemia-mercado-de-pcs-cai-123-no-primeiro-trimestre-de-2020/>. Acesso em: 24 ago. 2022.

MEIRELLES, FERNANDO S. **Pesquisa Uso da TI – Tecnologia de Informação nas Empresas. Resultados da Pesquisa e Tendências de Software**, São Paulo, 33ª Edição, 2022, FGV EAESP, 1.6-1.7, agosto 2022.

ORG, NORMAS ABNT. **Regras ABNT 2022**. pré-textuais, textuais e pós-textuais, 2022. Disponível em: <https://www.normasabnt.org/>. Acesso em: 20 nov. 2022.

PASCUTTI, Márcia Cristina; FREITAS, Janaina Aparecida; GASPAROTTI, Talita Tonsic. **Engenharia de Software**. Maringá: Patras, 2016. 136-138 p.

PRESSMAN, Roger S. **Engenharia de Software: Uma abordagem Profissional**. 7. ed. SÃO PAULO: Makron Books, 2011.

SOMMERVILLE, Ian. **Engenharia de software**: 8ed. SÃO PAULO: Person Addison Wesley, 2007.

SOMMERVILLE, Ian. **Engenharia de Software**. 9ª. ed. São Paulo: Pearson Prentice Hall, 2011. p. 2-159.

VENTURA, Plínio. **O que é Requisito Funcional**. Até o Momento, 2022. Disponível em: <https://www.ateomomento.com.br/diagramas-uml/>. Acesso em: 10 set. 2022.

WEB, Br Modelo. Ferramenta de modelagem. **ferramenta de modelagem de banco de dados online**, 2022. Disponível em: <https://www.brmodeloweb.com/lang/en/index.html>. Acesso em: 11 set. 2022.

WORKBENCH, MySQL. Oracle. **Windows (x86, 32-bit), MSI Installer**, 2022. Disponível em: <https://dev.mysql.com/downloads/workbench/>. Acesso em: 13 set. 2022.

## APÊNDICES

## APÊNDICE A – Dicionário de Dados

Nas Tabelas de 17 a 19 estão o dicionário de dados das tabelas do sistema.

Tabela	tbclientes			
Descrição	Armazena as informações do cliente			
Observação				
Campos				
Nome	Descrição	tipo de dados	Tamanho	Restrições do dominio
idcliente	codigo de identificação do cliente	int	11	PK/auto_increment
nomecliente	nome do cliente	varchar	50	not null
endereccliente	endereço do cliente	varchar	100	
fonecliente	telefone do cliente	varchar	50	not null
emailcliente	email do cliente	varchar	50	

Fonte: Autoral (2022). Tabela 17 – Dicionário de dados da tabela clientes

Tabela	tbusuário			
Descrição	Armazena as informações do usuário			
Observação				
Campos				
Nome	Descrição	tipo de dados	Tamanho	Restrições do dominio
idusuario	codigo de identificação do usuário	int		
nomeusuario	nome do usuário	varchar	50	
endusuario	endereço do usuário	varchar	100	
foneusuario	telefone do usuário	varchar	50	
emailusuario	email do usuário	varchar	50	

Fonte: Autoral (2022). Tabela 18 – Dicionário de dados da tabela usuários

Tabela	tbordem de serviços			
Descrição	Armazena as informações da tabela ordem de serviços			
Observação				
Campos				
Nome	Descrição	tipo de dados	Tamanho	Restrições do dominio
ordemservico	codigo de identificação da ordem de serviço	int	11	PK/auto_increment
data_ordem	data de emissão da ordem de serviço	timestamp		current_timestamp
tipo	define se ordem de serviço ou orçamento	varchar	15	
situacao	definição do estado do produto na loja	varchar	20	
equipamento	especifica o equipamento que entrou na loja	varchar	150	
defeito	especifica o problema do equipamento	varchar	150	
servico	especifica serviço a ser realizado	varchar	150	
tecnico	nome do responsavel pela manutenção	varchar	150	
valor	valor total do serviço realizado ou orçado	decimal	10,2	not null, foreign key
idcliente	nome do cliente	int	11	foreign key(cliente)

Fonte: Autoral (2022). Tabela 19 – Dicionário de dados da tabela ordem de serviços

## APÊNDICE B – Código fonte do Sistema

A Figura 25 mostra o código fonte da tela login.

```
//Contrutor
public TelaLogin() {
    initComponents();
    conexao = ModuloConexao.conector();

    // A linha abaixo serve de apoio ao status de conexão se o banco esta ou não conecatdo com o apache
    //System.out.print(conexao);
    if (conexao != null) {
        lblStatus.setIcon(new javax.swing.ImageIcon(getClass().getResource("/br/com/enable/icones/on.banco.
png")));
    } else {
        lblStatus.setIcon(new javax.swing.ImageIcon(getClass().getResource("/br/com/enable/icones/off.banco.
png")));
    }
}
}
```

Fonte: Autoral (2022). Figura 25 – Tela Login.

A Figura 26 mostra o código fonte da consulta ao banco de dados.

```
// Aqui realiza a consulta ao banco de dados(query)
rs = pst.executeQuery();
//Se tiver usuario correspondente
if (rs.next()) {
    //Obtem o conteudo do campo perfil da tabela tbusuarios
    String perfil = rs.getString(6);
    //System.out.println(perfil);
    //Realizando a estrutura de acesso ao nível do perfil
    if (perfil.equals("admin")) {
        //Abrindo tela principal
        TelaPrincipal principal = new TelaPrincipal();
        principal.setVisible(true);
        TelaPrincipal.menRel.setEnabled(true);
        TelaPrincipal.MenCadUsu.setEnabled(true);
        TelaPrincipal.lblUsuario.setText(rs.getString(2));
        TelaPrincipal.lblUsuario.setForeground(Color.blue);
        this.dispose();
    }else{
        TelaPrincipal principal = new TelaPrincipal();
        principal.setVisible(true);
        TelaPrincipal.lblUsuario.setText(rs.getString(2));
        TelaPrincipal.lblUsuario.setForeground(Color.red);
        this.dispose();
    }
    // conexao.close();
} else {
    JOptionPane.showMessageDialog(null, "usuário e/ou senhas inválidos(s)");
}
```

Fonte: Autoral (2022).      Figura 26 – Consulta ao banco de dados

A Figura 27 mostra o código fonte da tela login.

```
public class TelaLogin extends javax.swing.JFrame {

    //frameworks do sql
    Connection conexao = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    //metodo login
    public void logar() {
        String sql = "select * from tbusuarios where login=? and senha=?";
        try {
            // Recolhendo dados dos campos de texto para o efetuar login
            pst = conexao.prepareStatement(sql);
            pst.setString(1, txtUsuario.getText());
            String captura = new String(txtSenha.getPassword());
            pst.setString(2, captura);
        }
    }
}
```

Fonte: Autoral (2022). Figura 27 – Tela login.

A Figura 28 mostra o código fonte da tela adicionar clientes.

```
28 //Criando método adicionar clientes
29 private void adicionar() {
30     String sql = "insert into tblclientes(nomecliente,enderecocliente,fonecliente,emailcliente) values(?, ?, ?, ?)";
31     try {
32         pst = conexao.prepareStatement(sql);
33         pst.setString(1, txtCliNome.getText());
34         pst.setString(2, txtCliEndereco.getText());
35         pst.setString(3, txtCliFone.getText());
36         pst.setString(4, txtCliEmail.getText());
37
38         //Validação dos Campos obrigatórios
39         if ((txtCliNome.getText().isEmpty()) || (txtCliFone.getText().isEmpty())) {
40             JOptionPane.showMessageDialog(null, "Preencha todos os campos obrigatórios");
41         } else {
42
43             // confirmando a inserção de dados na tabela do banco
44             // atualizando as tabelas do banco de dados com os dados inseridos no formulário
45             int adicionado = pst.executeUpdate();
46             if (adicionado > 0) {
47                 JOptionPane.showMessageDialog(null, "Cliente adicionado com sucesso");
48                 limpar();
49             }
50         }
51     } catch (Exception e) {
52         JOptionPane.showMessageDialog(null, e);
53     }
54 }
```

Fonte: Autoral (2022). Figura 28 – Tela Adicionar Clientes

A Figura 29 mostra o código fonte da tela alterar clientes.

```

86 //Criando o método alterar dados dos clientes
87 private void alterar() {
88     String sql = "update tbclientes set nomecliente=?,enderecocliente=?,fonecliente=?,emailcliente=?where idcliente=?";
89     try {
90         pst = conexao.prepareStatement(sql);
91         pst.setString(1, txtCliNome.getText());
92         pst.setString(2, txtCliEndereco.getText());
93         pst.setString(3, txtCliFone.getText());
94         pst.setString(4, txtCliEmail.getText());
95         pst.setString(5, txtCliId.getText());
96
97         if ((txtCliNome.getText().isEmpty() || txtCliFone.getText().isEmpty())) {
98             JOptionPane.showMessageDialog(null, "Preencha todos os campos obrigatórios");
99         } else {
100
101             // confirmando a alteração de dados na tabela do banco
102             // alterando as tabelas do banco de dados com os dados inseridos no formulário
103             int adicionado = pst.executeUpdate();
104             if (adicionado > 0) {
105                 JOptionPane.showMessageDialog(null, "Dados do cliente alterados com sucesso");
106                 limpar();
107                 btnAdicionar.setEnabled(true);
108             }
109         }
110     } catch (Exception e) {
111         JOptionPane.showMessageDialog(null, e);
112     }
113 }
114 }

```

Fonte: Autoral (2022). Figura 29 – Tela Alterar Cliente.

A Figura 30 mostra o código fonte da tela pesquisar cliente.

```

56 //Criando método para pesquisar Clientes pelo nome
57 private void pesquisar_cliente() {
58     String sql = "select idcliente as id, nomecliente as nome, enderecocliente as endereço, "
59         + "fonecliente as fone,emailcliente as email from tbclientes where nomecliente like ?";
60     try {
61         pst = conexao.prepareStatement(sql);
62         //passando conteúdo da caixa de pesquisa para o ?
63         //Atenção ao "%" que é a continuação da String sql
64         pst.setString(1, txtCliPesquisar.getText() + "%");
65         rs = pst.executeQuery();
66         //usando a biblioteca rs2xml para preencher a tabela
67         tblClientes.setModel(DbUtils.resultSetToTableModel(rs));
68     } catch (Exception e) {
69         JOptionPane.showMessageDialog(null, e);
70     }
71 }

```

Fonte: Autoral (2022). Figura 30 – Pesquisar Cliente.



A Figura 31 mostra o código fonte da Tela remover cliente.

```

116 //Criando método remover clientes
117 private void remover() {
118     //Perguntando se confirma a remoção
119     int confirma = JOptionPane.showConfirmDialog(null, "Tem certeza que vai remover este Cliente?", "Atenção", JOptionPane.YES_NO_OPTION);
120     if (confirma == JOptionPane.YES_NO_OPTION) {
121         String sql = "delete from tblclientes where idcliente=?";
122         try {
123             pst = conexao.prepareStatement(sql);
124             pst.setString(1, txtCliId.getText());
125             int apagado = pst.executeUpdate();
126             if (apagado > 0) {
127                 JOptionPane.showMessageDialog(null, "Cliente removido com sucesso");
128                 limpar();
129                 btnAdicionar.setEnabled(true);
130             }
131         } catch (Exception e) {
132             JOptionPane.showMessageDialog(null, e);
133         }
134     }
135 }
136
137 }

```

Fonte: Autoral (2022). Figura 31 – Tela Remover Clientes

A Figura 32 mostra o código fonte da Tela adicionar usuário.

```

private void adicionar() {
    String sql = "insert into tbusuarios(iduser,usuario,fone,login,senha,perfil) values(?, ?, ?, ?, ?, ?)";
    try {
        pst = conexao.prepareStatement(sql);
        pst.setString(1, txtUsuId.getText());
        pst.setString(2, txtUsuNome.getText());
        pst.setString(3, txtUsuFone.getText());
        pst.setString(4, txtUsuLogin.getText());
        pst.setString(5, txtUsuSenha.getText());
        pst.setString(6, cboUsuPerfil.getSelectedItem().toString());
        //Validação dos Campos obrigatórios
        if ((txtUsuId.getText().isEmpty()) || (txtUsuNome.getText().isEmpty()) || (txtUsuLogin.getText().
isEmpty()) || (txtUsuSenha.getText().isEmpty())) {
            JOptionPane.showMessageDialog(null, "Preencha todos os campos obrigatórios");
        } else {

            // confirmando a inserção de dados na tabela do banco
            // atualizando as tabelas do banco de dados com os dados inseridos no formulário
            int adicionado = pst.executeUpdate();
            if (adicionado > 0) {
                JOptionPane.showMessageDialog(null, "Usuário adicionado com sucesso");
                txtUsuId.setText(null);
                txtUsuNome.setText(null);
                txtUsuFone.setText(null);
                txtUsuLogin.setText(null);
                txtUsuSenha.setText(null);
            }
        }
    }
}

```

Fonte: Autoral (2022). Figura 32 – Tela adicionar usuário.

A Figura 33 mostra o código fonte da Tela alterar usuário.

```
private void alterar() {
    String sql = "update tbusuarios set usuario=?,fone=?,login=?,senha=?,perfil=? where iduser=?";
    try {
        pst = conexao.prepareStatement(sql);
        pst.setString(1, txtUsuNome.getText());
        pst.setString(2, txtUsuFone.getText());
        pst.setString(3, txtUsuLogin.getText());
        pst.setString(4, txtUsuSenha.getText());
        pst.setString(5, cboUsuPerfil.getSelectedItem().toString());
        pst.setString(6, txtUsuId.getText());

        if ((txtUsuId.getText().isEmpty() || txtUsuNome.getText().isEmpty() || txtUsuLogin.getText().
isEmpty() || txtUsuSenha.getText().isEmpty())) {
            JOptionPane.showMessageDialog(null, "Preencha todos os campos obrigatórios");
        } else {

            // confirmando a alteração de dados na tabela do banco
            // alterando as tabelas do banco de dados com os dados inseridos no formulário
            int adicionado = pst.executeUpdate();
            if (adicionado > 0) {
                JOptionPane.showMessageDialog(null, "Dados do usuário alterados com sucesso");
                txtUsuId.setText(null);
                txtUsuNome.setText(null);
                txtUsuFone.setText(null);
                txtUsuLogin.setText(null);
                txtUsuSenha.setText(null);
            }
        }
    }
}
```

Fonte: Autoral (2022). Figura 33 – Tela alterar usuário.

A Figura 34 mostra o código fonte da conexão do usuário.

```
public class TelaUsuario extends javax.swing.JInternalFrame {

    //frameworks do sql
    Connection conexao = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    //Construtor
    public TelaUsuario() {
        initComponents();
        conexao = ModuloConexao.conector();
    }
}
```

Fonte: Autoral (2022). Figura 34 – Conexão do usuário.

A Figura 35 mostra o código fonte consulta de usuário cadastrado.

```
private void consultar() {
    String sql = "select * from tbusuarios where iduser=?";
    try {
        pst = conexao.prepareStatement(sql);
        pst.setString(1, txtUsuId.getText());
        rs = pst.executeQuery();

        if (rs.next()) {
            txtUsuNome.setText(rs.getString(2));
            txtUsuFone.setText(rs.getString(3));
            txtUsuLogin.setText(rs.getString(4));
            txtUsuSenha.setText(rs.getString(5));
            //refere-se ao ComboBox
            cboUsuPerfil.setSelectedItem(rs.getString(6));
        } else {
            //Emite a mensagem para o usuário
            JOptionPane.showMessageDialog(null, "Usuário não cadastrado");
            //Limpa os campos para o usuário fazer um novo cadastro
            txtUsuNome.setText(null);
            txtUsuFone.setText(null);
            txtUsuLogin.setText(null);
            txtUsuSenha.setText(null);
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

Fonte: Autoral (2022). Figura 35 – Consulta de usuário cadastrado.

A Figura 36 mostra o código fonte da remoção de usuário.

```

private void remover() {
    //Perguntando se confirma a remoção
    int confirma=JOptionPane.showConfirmDialog(null,"Tem certeza que vai remover este usuário","Atenção",
JOptionPane.YES_NO_OPTION);
    if (confirma ==JOptionPane.YES_NO_OPTION){
        String sql="delete from tbusuarios where iduser=?";
        try {
            pst=conexao.prepareStatement(sql);
            pst.setString(1, txtUsuId.getText());
            int apagado = pst.executeUpdate();
            if(apagado>0){
                JOptionPane.showMessageDialog(null,"Usuário removido com sucesso");
                txtUsuId.setText(null);
                txtUsuNome.setText(null);
                txtUsuFone.setText(null);
                txtUsuLogin.setText(null);
                txtUsuSenha.setText(null);
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null,e);
        }
    }
}

```

Fonte: Autoral (2022). Figura 36 – Remoção de usuário.

A Figura 37 mostra o código fonte de alterar a ordem de serviço.

```

private void alterar_os() {
    String sql = "update tbordemservico set tipo=?,situacao=?,equipamento=?,defeito=?,servico=?,tecnico=?,valor=?
where ordemservico=?";

    try {
        pst = conexao.prepareStatement(sql);
        pst.setString(1, tipo);
        pst.setString(2, cboOsSit.getSelectedItem().toString());
        pst.setString(3, txtOsEquip.getText());
        pst.setString(4, txtOsDef.getText());
        pst.setString(5, txtOsServ.getText());
        pst.setString(6, txtOsTec.getText());
        pst.setString(7, txtOsValor.getText().replace(",",".")); // substitui a virgula pelo ponto
        pst.setString(8, txtOs.getText());

        // validando campos obrigatórios
        if ((txtCliId.getText().isEmpty() || (txtOsEquip.getText().isEmpty()) || (txtOsDef.getText().isEmpty())
|| cboOsSit.getSelectedItem().equals(" ")) {
            JOptionPane.showMessageDialog(null, "Preencha todos os campos obrigatórios!");
        } else {
            int adicionado = pst.executeUpdate();
            if (adicionado > 0) {
                JOptionPane.showMessageDialog(null, "Ordem de Serviço alterada com sucesso!");
                limpar();
            }
        }
    }
}

```

Fonte: Autoral (2022). Figura 37 – Alterar ordem de serviço.

A Figura 38 mostra o código fonte de emitir ordem de serviço.

```

private void emitir_os() {
    String sql = "insert into tbordemservico (tipo,situacao,equipamento,defeito,servico,tecnico,valor,idcliente)
values (?, ?, ?, ?, ?, ?, ?, ?) ";
    try {
        pst = conexao.prepareStatement(sql);
        pst.setString(1, tipo);
        pst.setString(2, cboOsSit.getSelectedItem().toString());
        pst.setString(3, txtOsEquip.getText());
        pst.setString(4, txtOsDef.getText());
        pst.setString(5, txtOsServ.getText());
        pst.setString(6, txtOsTec.getText());
        pst.setString(7, txtOsValor.getText().replace(",", ".")); // substitui a virgula pelo ponto
        pst.setString(8, txtCliId.getText());

        // validando campos obrigatórios
        if ((txtCliId.getText().isEmpty() || txtOsEquip.getText().isEmpty() || txtOsDef.getText().isEmpty())
|| cboOsSit.getSelectedItem().equals(" ")) {
            JOptionPane.showMessageDialog(null, "Preencha todos os campos obrigatórios!");
        } else {
            int adicionado = pst.executeUpdate();
            if (adicionado > 0) {
                JOptionPane.showMessageDialog(null, "Ordem de Serviço emitida com sucesso!");
                //recuperar o id da ordem de serviço
                recuperar_os();
                btnOsAdicionar.setEnabled(false);
                btnOsPesquisar.setEnabled(false);
                btnOsImprimir.setEnabled(true);
            }
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

Fonte: Autoral (2022). Figura 38 – emitir ordem de serviço.

A Figura 39 mostra o código fonte de excluir ordem de serviço.

```

private void excluir_os() {
    int confirma = JOptionPane.showConfirmDialog(null, "Tem certeza que deseja excluir esta Ordem de Serviço?",
"Cuidado", JOptionPane.YES_NO_OPTION);
    if (confirma == JOptionPane.YES_OPTION) {
        String sql = "delete from tbordemservico where ordemservico=?";
        try {
            pst = conexao.prepareStatement(sql);
            pst.setString(1, txtOs.getText());
            int apagado = pst.executeUpdate();
            if (apagado > 0) {
                JOptionPane.showMessageDialog(null, "Ordem de Serviço excluída com Sucesso");
                limpar();
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }
}

```

Fonte: Autoral (2022). Figura 39 – excluir ordem de serviço.

A Figura 40 mostra o código fonte de imprimir ordem de serviço.

```
private void imprimir_ordemservico() {
    // imprimindo uma ordem de serviço
    int confirma = JOptionPane.showConfirmDialog(null, "Confirma a impressão desta ordem de serviço?", "Atenção",
    JOptionPane.YES_NO_OPTION);
    if (confirma == JOptionPane.YES_OPTION) {
        // Emitindo relatório através do framework JasperReports
        try {
            // Classe HashMap para criar filtro
            HashMap filtro = new HashMap();
            filtro.put("os", Integer.parseInt(txtOs.getText()));
            //Utilizando a classe JasperPrint para organizar a impressão de um relatório
            JasperPrint print = JasperFillManager.fillReport("C:\\relatorios\\os.jasper", filtro, conexao);
            //Exibindo relatório através da classe JasperViewer
            JasperViewer.viewReport(print, false);
        } catch (JRException e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }
}
```

Fonte: Autoral (2022). Figura 40 – Imprimir ordem de serviço.

A Figura 41 mostra o código fonte de limpar ordem de serviço.

```
public void limpar() {
    txtOs.setText(null);
    txtData.setText(null);
    txtCliPesquisar.setText(null);
    ((DefaultTableModel) tblClientes.getModel()).setRowCount(0);
    cboOsSit.setSelectedItem(" ");
    txtCliId.setText(null);
    txtOsEquip.setText(null);
    txtOsDef.setText(null);
    txtOsServ.setText(null);
    txtOsTec.setText(null);
    txtOsValor.setText(null);
    //Habilitando os botões(objetos) para emitir nova ordem de serviço
    btnOsAdicionar.setEnabled(true);
    btnOsPesquisar.setEnabled(true);
    txtCliPesquisar.setEnabled(true);
    tblClientes.setVisible(true);
    //Desabilitar os botões
    btnOsAlterar.setEnabled(false);
    btnOsExcluir.setEnabled(false);
    btnOsImprimir.setEnabled(false);
}
```

Fonte: Autoral (2022). Figura 41 – Limpar ordem de serviço.

A Figura 42 mostra o código fonte da conexão da ordem de serviço.

```

public class TelaOrdemServico extends javax.swing.JInternalFrame {

    //criando conexão com o banco de dados
    Connection conexao = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    //Criando variável para armazenamento de um texto de radion button selecionando
    private String tipo;

    //construtor
    public TelaOrdemServico() {
        initComponents();
        conexao = ModuloConexao.conector();
    }

    private void pesquisar_cliente() {
        String sql = "select idcliente as Id,nomecliente as Nome ,fonecliente as Fone from tblclientes
where nomecliente like ?";
        try {
            pst = conexao.prepareStatement(sql);
            pst.setString(1, txtCliPesquisar.getText() + "%");
            rs = pst.executeQuery();
            tblClientes.setModel(DbUtils.resultSetToTableModel(rs));
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }
}

```

Fonte: Autoral (2022). Figura 42 – Conexão da ordem de serviço.

A Figura 43 mostra o código fonte de pesquisar a ordem de serviço.

```

private void pesquisar_os() {

    // Criando uma caixa de dialogo do tipo JOptionPane
    String num_os = JOptionPane.showInputDialog("Numero da Ordem de Serviço");
    String sql = "select ordemservico, date_format(data_orden,'%d/%m/%Y - %H:%i'),tipo,situacao,equipamento,
efeito,servico,tecnico,valor,idcliente from tbordemservico where ordemservico=" + num_os;
    try {
        pst = conexao.prepareStatement(sql);
        rs = pst.executeQuery();
        if (rs.next()) {
            txtOs.setText(rs.getString(1));
            txtData.setText(rs.getString(2));
            //Setando os radio buttons
            String rbtTipo = rs.getString(3);
            if (rbtTipo.equals("OS")) {
                rbtOs.setSelected(true);
                tipo = "OS";
            } else {
                rbtOrc.setSelected(true);
                tipo = "Orçamento";
            }
        }
        cboOsSit.setSelectedItem(rs.getString(4));
        txtOsEquip.setText(rs.getString(5));
        txtOsDef.setText(rs.getString(6));
        txtOsServ.setText(rs.getString(7));
        txtOsTec.setText(rs.getString(8));
        txtOsValor.setText(rs.getString(9));
        txtCliId.setText(rs.getString(10));
    }
}

```

Fonte: Autoral (2022). Figura 43 – Pesquisar a ordem de serviço.

A Figura 44 mostra o código fonte recuperar ordem de serviço.

```
private void recuperar_os() {  
    String sql = "select max(ordemservico) from tbordemservico";  
    try {  
        pst = conexao.prepareStatement(sql);  
        rs = pst.executeQuery();  
        if(rs.next()){  
            txtOs.setText(rs.getString(1));  
        }  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(null, e);  
    }  
}
```

Fonte: Autorial (2022).      Figura 44 – Pesquisar a ordem de serviço.