

Instituto Federal de Educação, Ciência e Tecnologia de Goiás
Câmpus Jataí
Tecnologia em Análise e Desenvolvimento de Sistemas

AquaHaven - um software para gestão de dados no comércio varejista de animais vivos



Autor: Rafael Simão da Costa
Orientador: Prof. Dr. Flávio de Assis Vilela

2023



INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
SISTEMA INTEGRADO DE BIBLIOTECAS

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

- | | |
|--|---|
| ☐ Tese | ☐ Artigo Científico |
| ☐ Dissertação | ☐ Capítulo de Livro |
| ☐ Monografia – Especialização | ☐ Livro |
| ☒ TCC - Graduação | ☐ Trabalho Apresentado em Evento |
| ☐ Produto Técnico e Educacional - Tipo: _____ | |

Nome Completo do Autor: Rafael Simão da Costa

Matrícula: 20211020230211

Título do Trabalho: AquaHaven - um software para gestão de dados no comércio varejista de animais vivos

Autorização - Marque uma das opções

1. (X) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
2. () Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data ____/____/____ (Embargo);
3. () Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção **2 ou 3**, marque a justificativa:

- () O documento está sujeito a registro de patente.
() O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.
() Outra justificativa: _____

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

- i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- ii. obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Jataí - GO , 13/12/2023.

**MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
SISTEMA INTEGRADO DE BIBLIOTECAS**



Documento assinado digitalmente

RAFAEL SIMAO DA COSTA

Data: 13/12/2023 14:56:10-0300

Verifique em <https://validar.itl.gov.br>

Assinatura do Autor e/ou Detentor dos Direitos Autorais

Rafael Simão da Costa

AquaHaven - um software para gestão de dados no comércio varejista de animais vivos

Trabalho de Conclusão de Curso submetida ao curso de graduação em Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de Goiás, câmpus Jataí, como requisito parcial para obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Instituto Federal de Educação, Ciência e Tecnologia de Goiás

Câmpus Jataí

Orientador: Prof. Dr. Flávio de Assis Vilela

2023

Dados Internacionais de Catalogação na Publicação na (CIP)

Costa, Rafael Simão da.

AquaHaven: um software para gestão de dados no comércio varejista de animais vivos [manuscrito]/ Rafael Simão da Costa. - Jataí : IFG, Coordenação dos cursos de Informática – Tecnologia em Análise e Desenvolvimento de Sistemas, 2023.

70 f.; il.

Orientador: Prof. Dr. Flávio de Assis Vilela.

Bibliografias.

1. Animais vivos. 2. Peixes ornamentais. 3. Gerenciamento de dados. 4. *Software*. I. Vilela, Flávio de Assis. II. IFG, Câmpus Jataí. III. Título.

Ficha catalográfica elaborada pela Seção Téc.: Aquisição e Tratamento da Informação.
Bibliotecária – Rosy Cristina Oliveira Barbosa – CRB 1/2380 – Câmpus Jataí. Cód. F058/2023-2.



INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
CÂMPUS JATAÍ

Rafael Simão da Costa

AquaHaven: um software para gestão de dados para o comércio de animais vivos

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de Goiás (IFG), Câmpus Jataí, como parte dos requisitos para a conclusão do Curso e obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Relatório Técnico defendido e aprovado, em 30 de novembro de 2023, pela banca examinadora constituída pelos seguintes membros:

(assinado eletronicamente)

Prof. Dr. Flávio de Assis Vilela

Presidente da banca / Orientador

Instituto Federal de Educação, Ciência e Tecnologia de Goiás – Câmpus Jataí

(assinado eletronicamente)

Prof. Me. Sérgio Henrique de Almeida

Membro

Instituto Federal de Educação, Ciência e Tecnologia de Goiás – Câmpus Jataí

(assinado eletronicamente)

Prof. Romildo Alves de Souza Junior

Membro

Instituto Federal de Educação, Ciência e Tecnologia de Goiás – Câmpus Jataí

Jataí – GO

2023

Documento assinado eletronicamente por:

- Sergio Henrique de Almeida, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 14/12/2023 15:09:54.
- Romildo Alves de Souza Junior, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO, em 14/12/2023 14:53:25.
- Flavio de Assis Vilela, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 14/12/2023 14:48:38.

Este documento foi emitido pelo SUAP em 14/12/2023. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifg.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 490622

Código de Autenticação: e49665c3a1



Instituto Federal de Educação, Ciência e Tecnologia de Goiás

Av. Presidente Juscelino Kubitschek, nº 775, Residencial Flamboyant, JATAÍ / GO, CEP 75804-714
(64) 3514-9511 (ramal: 9511)

Agradecimentos

Agradeço a Deus, a minha família que sempre me incentivaram e apoiaram nos momentos mais difíceis.

Agradeço a todas as pessoas que conheci durante esse tempo que passei na faculdade, pelas amizades que fiz.

Agradeço a todos meus professores por todo apoio recebido nesta jornada.

Agradeço ao meu orientador, professor Flávio pelo apoio que me forneceu durante toda a minha trajetória.

Resumo

Após decretado a permissão de vendas de animais vivos em locais adequados, o comércio varejista teve um grande avanço nas vendas de animais de estimação tais como cachorros, gatos e peixes ornamentais, entre outras espécies. Além disso, as fazendas rurais também contam com o comércio de animais vivos e com a produção de alimentos derivados dos animais, tais como gado de corte e leiteiro, cavalos e suínos. Contudo nas empresas que atuam no comércio de animais vivos cujo foco é em peixes ornamentais, o controle de dados para tomada de decisão ocorre de forma manual, fato este que pode favorecer a falta de controle financeiro, entre outros fatores negativos nas empresas. Para auxiliar a gestão nesse ramo de atividade, foi desenvolvido um *software* que irá auxiliar os profissionais desta área no controle e tomada de decisão da empresa, além de oferecer mais agilidade e rapidez na gestão dos dados da empresa. Para o desenvolvimento, foram utilizadas as abordagens iterativa e incremental, além do *framework* SCRUM.

Palavras-chaves: Animais vivos, Peixes ornamentais, Gerenciamento de dados, *software*.

Abstract

After permission to sell live animals in suitable locations has been approved, the co- The retail market had a big boost in sales of pets such as dogs, cats and ornamental fish, among other species. Furthermore, farms rural areas also rely on the trade of live animals and food production derived from animals, such as beef and dairy cattle, horses and swine. However, in companies that operate in the trade of live animals whose focus is on ornamental fish, the data control for decision making occurs manually, a fact that can favor the lack of financial control, among other negative factors in companies. To assist management in this field of activity, *software* was developed that will assist professionals in this area in controlling and making decisions within the company, in addition to to offer more agility and speed in managing company data. For the development development, iterative and incremental approaches were used, in addition to the *framework* SCRUM.

Key-words: Live animals, Ornamental fish, Data management, *software*.

Lista de ilustrações

Figura 1 – Exemplo do Peixe Platy	16
Figura 2 – Exemplo do Peixe Betta	17
Figura 3 – Exemplo do Peixe Oscar	18
Figura 4 – Exemplo de Caso de Uso	29
Figura 5 – Exemplo de Processos identificação de mudanças e de evolução .	32
Figura 6 – Exemplo de Entrega incremental	34
Figura 7 – Exemplo do SCRUM	35
Figura 8 – Exemplo Planejamento de Sprint	36
Figura 9 – Diagrama de Caso de uso	40
Figura 10 – Diagrama de Classes	41
Figura 11 – Diagrama de Atividade	42
Figura 12 – Diagrama de Sequência	44
Figura 13 – Diagrama de Caso de Uso	47
Figura 14 – Diagrama de Classe	49
Figura 15 – Diagrama de Atividade de Venda	51
Figura 16 – Diagrama de Atividade de Compra	53
Figura 17 – Conexão com o Banco de Dados	54
Figura 18 – Tela de Login	59
Figura 19 – Tela Inicial	60
Figura 20 – Tela de Cadastro de Produto	60
Figura 21 – Tela de Cadastro de Serviço	61
Figura 22 – Tela de Cadastro de Tipo de Produto	62
Figura 23 – Tela de Cadastro de Cliente	62
Figura 24 – Tela de Cadastro de Fornecedor	63
Figura 25 – Tela de Cadastro de Funcionário	64
Figura 26 – Tela de Cadastro de Venda	64
Figura 27 – Tela de Cadastro de Compra	65
Figura 28 – Tela de Cadastro de Registro	66
Figura 29 – Tela de Relatório de Produtos	67
Figura 30 – Tela de Relatório de Vendas	67
Figura 31 – Tela de Relatório de Compras	68

Lista de tabelas

Tabela 1 – Tabela Comparativa de software 25

Lista de abreviaturas e siglas

Web	Rede de computadores interligados na internet.
Mysql	Sistema gerenciador de banco de dados relacional.
Stakeholders	Usuários com conhecimento sobre como a informação é gerida modificada em determinados setores da empresa e que possuem interesse no software.
Framework	É um termo inglês que, em sua tradução direta, significa estrutura. De maneira geral, essa estrutura é feita para resolver um problema específico. Na programação, um framework é um conjunto de códigos genéricos capaz de unir trechos de um projeto de desenvolvimento.
Java	Linguagem de programação orientada a objetos.
sprint	Ciclo de desenvolvimento de incremento do sistema.
UML	Unified Modeling Language

Sumário

1	INTRODUÇÃO	15
1.1	Comercialização de Animais Vivos	15
1.2	Criação de Peixes Ornamentais	16
1.3	Problemas identificados no comércio de animais vivos	19
1.4	Hipótese 1	20
1.5	Hipótese 2	21
1.6	Objetivo geral	21
1.7	Objetivos específicos	21
1.8	Organização do trabalho	22
2	TRABALHOS CORRELATOS	23
2.1	Tabela Comparativa.	25
3	FUNDAMENTAÇÃO TEÓRICA	27
3.1	Atividades fundamentais para a engenharia de software	27
3.1.1	Especificação de software	28
3.1.2	Projeto e Implementação	30
3.1.3	Validação de software	30
3.1.4	Evolução do software	31
3.2	Desenvolvimento ágil	32
3.2.1	Entrega incremental	33
3.2.2	SCRUM	34
3.2.3	1º Fase - Planejamento Geral	37
3.2.4	2º Fase - Sprint	37
3.2.5	3º Fase - Encerramento do Projeto	38
3.3	UML	38
3.3.1	Porque utilizar UML?	39
3.3.2	Diagrama de Caso de Uso	39
3.3.3	Diagrama de Classes	40
3.3.4	Diagrama de Atividade	42
3.3.5	Diagrama de Sequência	43
4	DESENVOLVIMENTO DO SOFTWARE AQUAHAVEN	45
4.1	Definições do Scrum no desenvolvimento do AquaHaven	45
4.2	Definição dos requisitos	45

4.3	Projeto de sistemas	46
4.3.1	Diagrama de caso de uso do sistema AquaHaven	46
4.3.2	Diagrama de Classes do sistema AquaHaven	48
4.3.3	Diagrama de Atividade de Venda do sistema AquaHaven	50
4.3.4	Diagrama de Atividade de Compra do sistema AquaHaven	52
4.4	Operação e manutenção	54
4.5	Implementação e validação	54
4.6	Ferramentas de implementação	56
4.6.1	Mysql	56
4.6.2	Vaadin	56
4.6.3	VSCode	57
4.6.4	Java	58
5	SISTEMA AQUAHAVEN EM FUNCIONAMENTO	59
5.1	Telas do Sistema AquaHaven	59
6	CONCLUSÃO	69
6.1	Trabalhos futuros	69
	REFERÊNCIAS	71

1 Introdução

Neste capítulo será discutido a contextualização sobre o comércio de animais vivos. Na Seção 1.1 será abordada a comercialização de animais vivos. Na Seção 1.2 será falado sobre a criação de peixes ornamentais. Na seção 1.3 fala sobre quais foram os problemas identificados no comércio de animais vivos. Na seção 1.4 será discutido sobre a hipótese 1. Na seção 1.5 será debatido sobre a hipótese 2. Na seção 1.6 será abordado sobre os objetivos gerais. Na seção 1.7 será falado sobre os objetivos específicos. Na seção 1.8 será argumentado sobre a organização do trabalho.

1.1 Comercialização de Animais Vivos

No ano de 1987 foi consolidado o decreto que permite a comercialização de animais vivos, dentre eles estão presentes cachorros, gatos, pássaros, peixes ornamentais dentre outros animais. Segundo (APARECIDA, 2021) "O decreto 5616/1987 especifica que é permitida a comercialização de animais vivos, exclusivamente em estabelecimentos adequados, destinados para tal fim, previamente aprovados pela autoridade sanitária competente".

Desde então, devido ao crescimento da quantidade de animais de estimação, o número de comércio de animais vivos aumentou nos últimos anos, principalmente durante a pandemia, esse fator se deu pelo aumento de usuários que irar começar a trabalhar remotamente. Desse modo, buscaram a companhia de animais de estimação devido ao isolamento social e à necessidade de interação. Segundo (SEBRAE, 2023) "Enquanto o mundo vivia em crises produzidas pela pandemia da covid-19, o mercado pet cresceu 13,5%, gerando R\$ 40,1 bilhões em 2020, colocando o Brasil em segundo lugar no mundo".

No contexto do comércio de animais vivos, é importante mencionar que não se restringe apenas a animais de estimação, como cachorros, gatos, pássaros e peixes ornamentais. O comércio de animais vivos também é uma prática comum no setor pecuário, onde existem animais como gado, suínos e cavalos.

Nesse cenário apresentado sobre a demanda por animais de estimação, é importante destacar que a criação de peixes ornamentais tem se mostrado uma opção interessante. Os peixes ornamentais são apreciados por sua beleza e são populares como animais de estimação, tanto em aquários domésticos quanto em

ambientes comerciais, como lojas especializadas.

1.2 Criação de Peixes Ornamentais

O aquarismo ou aquariofilia é a criação de peixes ornamentais, plantas aquáticas em aquários ou outros recipientes de vidro e acrílico. Segundo MINISTÉRIO DO MEIO AMBIENTE (IBAMA, 2018):

"Os peixes ornamentais são encontrados em seus respectivos habitats sendo que espécies como Platy, Betta, Oscar e outras espécies tem o seu próprio habitat e geralmente são coletados na natureza. Para realizar a coleta comercial de peixes, moluscos, crustáceos e plantas aquáticas com finalidade ornamental ou de aquariofilia, o interessado deverá se registrar junto à Secretaria Especial de Aquicultura e Pesca da Presidência da República - SEAP/PR, na categoria de "pescador profissional."(IBAMA, 2018)

Figura 1 – Exemplo do Peixe Platy



Fonte: (RULIYANDI, 2019)

A Figura 1 apresenta um exemplo de um peixe ornamental popular conhecido como Platy, criado em água doce e muito comum em aquários, por sua facilidade em relação aos cuidados necessários para sua criação.

Segundo Matos:

"Um das espécies de peixe vivíparos mais populares, o Platy tem sua origem na América Central, principalmente na região do México e Honduras. Entretanto, quando falamos da espécie Platy, geralmente estamos nos referindo a duas espécies do gênero *Xiphophorus*, que são: *Xiphophorus maculatus* e *Xiphophorus variatus*."(MATOS, 2022)

"As espécies conseguem procriar entre si, dessa maneira, hoje em dia é até difícil distinguir entre as duas, já que se cruzaram muitas e muitas vezes no hobby do aquarismo. Esse excesso de misturas também acabou gerando uma infinidade de cores e características diferentes nos platys, o que acaba sendo um dos grandes charmes da espécie."(MATOS, 2022)

Além disso, essa espécie possui um comportamento pacífico. Segundo Matos:

"É uma espécie extremamente pacífica e ativa, portanto é muito apropriada para aquários comunitários. Apesar de não ser definido como um peixe de cardume, eles costumam se sentir melhor em aquários com outros peixes da sua espécie. Por ser um peixe pequeno, vale a pena adquirir alguns exemplares para colorir e alegrar seu aquário."(MATOS, 2022)

Figura 2 – Exemplo do Peixe Betta



Fonte: (RULIYANDI, 2020a)

A Figura 2 apresenta um exemplo de um peixe ornamental. O peixe betta é uma espécie de fáceis cuidados desde que as condições adequadas de água, temperatura e alimentação sejam mantidas de forma adequada para sua criação.

Segundo Santos:

"Sua origem é a região da Indonésia (sudeste da Ásia). São agressivos (machos) e, num mesmo local, podem brigar violentamente até a morte. O seu nome científico é *Betta splendens*. Possui um ritual de acasalamento, sendo que o macho solta na água uma grande quantidade de bolhas para atrair a fêmea. É o macho que fica protegendo

os ovos até o nascimento dos alevinos (filhotes de peixe). É uma espécie de peixe bastante resistente e vive em condições simples dentro de um aquário (sem necessidade de equipamentos)."(SANTOS, 2020)

O peixe Oscar é uma espécie um pouco mais agressiva com outros peixes, principalmente se o aquário for pequeno, é recomendado mantê-lo em um aquário exclusivo ou em um aquário com outras espécies grandes. A Figura 3 apresenta um exemplar desse peixe ornamental.

Figura 3 – Exemplo do Peixe Oscar



Fonte: (RULIYANDI, 2020b)

Segundo Rechi:

"Espécie popular entre aquaristas, porém, devido seu crescimento lento não é apreciado por aquicultores. Possui boca grande com lábios grossos, préopercular primeiro arco branquial sem lobo; rastros branquiais curtos e grossos com diversos dentículos; corpo de cor escura com opérculo no pedúnculo caudal. Podem alterar sua coloração rapidamente de acordo com a necessidade do momento, seja ritual, comportamental ou de combate entre indivíduos da mesma espécie."(RECHI, 2014)

Além dos peixes ornamentais existentes no meio ambiente aquático, há também os corais e outros invertebrados, Segundo MINISTÉRIO DO MEIO AMBIENTE:

"Para realizar a coleta comercial de Invertebrados aquáticos, que não sejam crustáceos ou moluscos, o interessado deverá:

I. Solicitar autorização ao Ibama, especificando as finalidades da coleta, as espécies que serão coletadas, os locais de coleta, os instrumentos que serão utilizados e o esforço de coleta (quantidade/período).

II. Registrar-se junto à Secretaria Especial de Aquicultura e Pesca da Presidência da República - SEAP/PR, na categoria de "pescador profissional". Caso tenha finalidade de formar plantel para aquicultura, o registro junto à SEAP deverá ser na categoria "aquicultor". Uma mesma pessoa pode ter os dois registros."(IBAMA, 2018)

É notável a presença de um maior número de estabelecimentos comerciais especializados em peixes ornamentais nas grandes cidades. Segundo Ribeiro:

"Os mercados consumidores de peixes ornamentais situam-se nos grandes centros urbanos. Os maiores mercados brasileiros situam-se no entorno das cidades de São Paulo e Rio de Janeiro, apesar do mercado também ser intenso em capitais como Belo Horizonte e Curitiba."(RIBEIRO et al., 2008)

1.3 Problemas identificados no comércio de animais vivos

No comércio de animais vivos, especialmente no ramo de peixes ornamentais, é comum o controle das vendas e outras atividades por meio de fichas, caderno-caixa, planilhas e outros métodos de gerenciamento de dados. No entanto, essa abordagem pode ser custosa e apresentar desafios para a gestão eficiente e tomada de decisões estratégicas. Dentre os problemas enfrentados estão a perda de dados, registros errôneos e registros fora do prazo adequado, pagamentos de contas fora do prazo devido ao mal controle da parte financeira e possibilidades de

perda de mercadorias devido a erros na contagem ou falta de acompanhamento adequado.

A criação do sistema AquaHaven se justifica pela necessidade de melhorar a gestão do comércio de peixes ornamentais. Atualmente são utilizados métodos tradicionais e manuais de controle de dados. Com o desenvolvimento dessa solução espera-se que supere as perdas de dados, registros errôneos e proporcione uma gestão mais eficiente e precisa. Além disso, o AquaHaven será capaz de lidar com as especificidades desse setor, como restrições na venda de certas espécies de peixes, fornecendo informações valiosas para a tomada de decisões estratégicas. Portanto, a criação do sistema AquaHaven é justificada pela necessidade de modernizar e aprimorar a gestão do comércio de peixes ornamentais.

A partir do contexto apresentado, é de suma importância desenvolver uma ferramenta de gestão mais moderna que possa superar esses desafios e fornecer um controle detalhado de todas as atividades da empresa. Essa ferramenta deve permitir registrar as vendas de diferentes tipos de peixes ornamentais, controlar as entradas e saídas de caixa, gerenciar compras de produtos e controlar o estoque. Além disso, é necessário que a ferramenta seja capaz de lidar com as especificidades desse domínio de negócio, como restrições na venda de certos tipos de peixes juntamente com outros, e emitir alertas para faturas em atraso, entre outros benefícios.

Ao utilizar uma ferramenta de gerenciamento de dados adequada, as empresas do ramo de animais vivos, em particular peixes ornamentais, podem obter diversos benefícios. Esses incluem um controle mais eficiente das atividades da empresa, registros precisos de vendas, controle de fluxo de caixa, gerenciamento de estoque e a capacidade de lidar com as especificidades do negócio. Essa modernização na gestão de dados possibilita uma base sólida para tomadas de decisão estratégicas e contribui para o crescimento e sucesso da empresa nesse setor.

1.4 Hipótese 1

Será desenvolvido um *software* específico para o ramo de comércio varejista de animais vivos com o foco em peixes ornamentais. Esse mesmo *software* irá auxiliar a tomada de decisão estratégica na obtenção de lucros, pois a forma na qual é feito o controle de gestão desse ramo de atividade atualmente não favorece a gestão precisa e íntegra dos dados produzidos.

1.5 Hipótese 2

Este *software* contará com funcionalidades que auxiliam todo o sistema, tais como gerenciamento de entrada e saída dos produtos, controle de pagamentos e recebimentos e relatórios. Além disso, deve permitir o cadastro de especificidades identificadas no domínio de negócio, cadastro de clientes, fornecedores, forma de pagamento, produtos e serviços oferecidos na empresa.

1.6 Objetivo geral

O objetivo geral deste trabalho é desenvolver uma ferramenta para uma empresa varejista de animais vivos, cujo o foco é em venda de peixes ornamentais, com o intuito de apoiar o processo de tomada de decisão estratégica a partir de dados integrados e consistentes.

1.7 Objetivos específicos

- Estudar o domínio de negócio sobre a comercialização de animais vivos;
- Realizar pesquisa de campo juntamente com um empresário da área de comércio de peixes ornamentais para entender o fluxo da atividade, assim como as necessidades e demandas de dados para tomada de decisão;
- Fazer uma pesquisa bibliográfica sobre ferramentas disponíveis atualmente que forneçam dados para tomada de decisão estratégica no ramo de peixes ornamentais;
- Realizar o levantamento de requisitos em uma empresa real;
- Analisar a viabilidade de fazer entregas parciais da ferramenta;
- Realizar uma pesquisa sobre técnicas e linguagem de programação que ajudem no desenvolvimento;
- Desenvolver o *software* com base nas abordagens iterativa e incremental, e auxiliado pelo *framework* SCRUM;
- Realizar todas as atividades inerentes a validação e testes de *software*;
- Desenvolver relatórios específicos para tomada de decisão estratégica para o ramo de peixes ornamentais.

1.8 Organização do trabalho

Este trabalho está organizado da seguinte forma: no capítulo 2 serão apresentados os resultados da pesquisa bibliográfica referente aos *softwares* encontrados, no Capítulo 3 é abordado a fundamentação teórica acerca dos conceitos básicos para o entendimento do trabalho. No Capítulo 4 é apresentada a metodologia aplicada neste trabalho. No Capítulo 5 são apresentadas as telas do sistema AquaHaven. No capítulo 6 será apresentado a conclusão do trabalho e os trabalhos futuros.

2 Trabalhos Correlatos

Para obter informações sobre o ramo de atividade relacionado ao comércio de animais vivos, foi adotado o método de pesquisa da observação participante. Essa escolha se deu devido à necessidade de obter uma compreensão aprofundada das práticas e processos envolvidos nesse contexto específico. Além da observação direta na loja de aquarismo, também foi realizada uma pesquisa bibliográfica abrangente. Por meio dessa pesquisa, foram identificados outros sistemas relevantes que abordavam *softwares* utilizados no comércio de animais vivos.

Foi adotada a abordagem da observação participante, em uma loja de aquarismo. Assim, foi possível identificar os problemas existentes no ramo de peixes ornamentais, os quais serviram de inspiração para o desenvolvimento do sistema AquaHaven. Durante essa etapa, foi realizada uma interação com os funcionários e clientes da loja, além de observar de perto as atividades e processos relacionados ao controle da empresa, que atualmente são realizados manualmente. Essa imersão no ambiente real permitiu uma compreensão aprofundada das necessidades e experiências dos envolvidos, além de uma avaliação sobre possíveis oportunidades de melhoria que poderão ser exploradas com o desenvolvimento do sistema AquaHaven.

Já a pesquisa bibliográfica desempenhou um papel fundamental ao fornecer informações sobre os fundamentos teóricos e conceituais subjacentes aos sistemas existentes. Ela também contribuiu para a compreensão das funcionalidades, vantagens e desafios associados a esses *softwares*. Durante a pesquisa foram encontrados dois *softwares*, que estão relacionados a área de animais vivos, e um *software*, que está relacionado a área de peixe.

O primeiro *software* é descrito por Lima (LIMA, 2015). Esse mesmo *software* foi desenvolvido com objetivo de auxiliar na gestão de pet shop e clínica veterinária, além disso, foi desenvolvido para auxiliar e automatizar a gestão das empresas no ramo da atividade. Entre algumas funcionalidades presentes nesta *software* são: cadastros, controle de caixa, agendamento de serviços, vendas de produtos e alguns relatórios.

Em sua fase de desenvolvimento foi realizado um levantamento de requisito em conjunto com os usuários e empresários do ramo. Para a implementação foi utilizada linguagem de programação Delphi com banco de dados relacional *Firebird*. Além disso, foi utilizado para a implementação algumas técnicas, tais como

diagramas e fluxogramas.

O segundo *software* encontrado na pesquisa é descrito em Lopes et al (LOPES, 2000), esse mesmo *software* foi desenvolvido com o objetivo de auxiliar os técnicos e produtores na determinação do custo de produção do leite. O *software* permiti o cadastro de todas as despesas e receitas referentes ao sistema de produção de leite, entre outras funcionalidades referente ao sistema.

O sistema foi desenvolvido utilizando a ferramenta Delphi, sendo uma programação orientada para objeto, em microcomputador IBM-PC compatível, com processador 586 e 16 MB de memória RAM. Além disso, será utilizado sobre a plataforma Windows 95 ou superior, não atendendo a plataformas anteriores, como o Windows 3.11.

Ambos os *softwares* auxiliam na gestão e controle da empresa, entre outras funcionalidades inerentes ao domínio de negócio.

Referente ao domínio de peixes ornamentais, ao realizar a pesquisa bibliográfica. Foi encontrado um *software* relacionado a área dessa pesquisa. O software em questão se chama ExpertFish - *software* para diagnóstico de doenças de peixes. (LOPES; TRINDADE; FUJIMOTO, 2016). O objetivo desse *software*, é auxiliar os usuários a diagnosticar doenças de peixes, profilaxia e tratamento. Foi desenvolvido no sistema Expert SINTA para obtenção de resultados rápidos e confiáveis, visto que esta ferramenta utiliza raciocínio lógico, tratamento de probabilidades e estabelecimento de inferências.

2.1 Tabela Comparativa.

Tabela 1 – Tabela Comparativa de software

#	Software	Área de Atuação	Ferramenta para Desenvolvimento	Gestão de dados financeiros?	Plataforma de Desenvolvimento
1	Software PARA GESTÃO DE PET SHOP E CLÍNICA VETERINÁRIA. (LIMA, 2015)	Pet Shop e Clínica Veterinária	Programação Delphi com banco de dados relacional Firebird	Sim	N/I .
2	Custo Leite para Windows: software de Controle de Custos para a Pecuária Leiteira. (LOPES et al., 2000)	Pecuária Leiteira	Programação Delphi	Sim	N/I.
3	EXPERTFISH – software PARA DIAGNÓSTICO DE DOENÇAS DE PEIXES. (LOPES; TRINDADE; FUJIMOTO, 2016)	Diagnóstico de Doenças de Peixes	Expert Sinta	Não	N/I.

4	AquaHaven software para Gerenciamento de Dados e Tomada de Decisão para Comércio Varejistas de Animais Vivos (COSTA, 2023)	Comércio Varejista de Animais Vivos focado em peixes ornamentais	Programação Java, utilizando VSCode com banco de dados MySQL	Sim	Web.
---	--	--	--	-----	------

Fonte: Autoria própria

A Tabela 1 mostra a comparação de três *softwares* já existentes, com o AquaHaven. A tabela possui quatro identificadores, em que no primeiro identificador, mostra o *software* para gestão de pet shop e clínica veterinária (LIMA, 2015), que atua no mercado de Pet Shop e Clínica Veterinária. No segundo identificador, se encontra um *software* de controle de custo para pecuária leiteira (LOPES, 2000), que atua no mercado de Pecuária Leiteira. O terceiro identificador, mostra um *software* para diagnosticar doenças em peixe (LOPES; TRINDADE; FUJIMOTO, 2016), em que é utilizado para Diagnóstico de Doenças de Peixes. E por último é mostrado o AquaHaven, a ferramenta que será desenvolvida neste trabalho, será focada para a área de peixes ornamentais, e será utilizada para gestão de dados e tomada de decisões dentro da empresa. O *software* para gestão de pet shop e clínica veterinária, possui gestão para dados financeiros, mas não apresenta de maneira clara qual seria a plataforma utilizada. O *software* de Controle de Custos para a pecuária leiteira também possui gestão para dados financeiros, mas não apresenta de maneira clara qual seria a plataforma utilizada. O *software* para diagnóstico de doenças de peixes, não possui gestão para dados financeiros e também não apresenta de maneira clara qual seria a plataforma utilizada. Em comparação aos softwares identificados ao longo da pesquisa bibliográfica, o AquaHaven contará com a gestão de dados financeiros, vendas, estoque e demais processos elencados ao longo da pesquisa, os quais serão apresentados no Capítulo 1.5. Além do mais, o AquaHaven será desenvolvido a partir da linguagem Java, a qual permite o desenvolvimento de aplicações Web.

3 Fundamentação Teórica

Para o desenvolvimento deste trabalho, foram utilizadas duas abordagens principais. O modelo de processo de *software* utilizado foi o desenvolvimento ágil. Para a fase de projeto, foram desenvolvidos os diagramas necessários para o entendimento do problema. Para a fase de desenvolvimento, foram utilizadas algumas ferramentas de desenvolvimento para sistemas Web. Dessa forma, neste capítulo serão descritos os fundamentos acerca dos conceitos aplicados neste trabalho.

3.1 Atividades fundamentais para a engenharia de software

Segundo Sommerville (SOMMERVILLE, 2011), há diferentes processos para desenvolvimentos de *software*, todos devem seguir quatros atividades fundamentais que são elas:

1. Especificação de *software*: A funcionalidade do *software* e as restrições a seu funcionamento devem ser definidas.
2. Projeto e implementação de *software*: O *software* deve ser produzido para atender as especificações
3. Validação de *software*: O *software* deve ser validado para garantir que atenda às demandas do cliente.
4. Evolução de *software*: O *software* deve evoluir para atender às necessidades de mudança dos clientes.

Essas quatro etapas são essenciais para o processo de desenvolvimento de *software*. Cada etapa contempla atividades que auxiliam o processo de desenvolvimento, desde a especificação e levantamento de requisitos, até a entrega do *software* desenvolvido. Contudo, existem diferentes tipos de abordagens que podem contemplar novas atividades ou sub atividades conjuntamente com essas quatro etapas listadas.

3.1.1 Especificação de software

Para o desenvolvimento de um *software* ou de um sistema, é necessário entender a necessidade do usuário e para que isso ocorra de maneira adequada, será realizado uma análise de requisitos. Nessa etapa será verificado quais as demandas que o usuário terá. Assim sendo fica claro as necessidades e os problemas que possa ocorrer durante o desenvolvimento. Com isso espera atingir o objetivo esperado pelo sistema.

Segundo Sommerville, (SOMMERVILLE, 2011, p. 65) a especificação de requisitos é:

"A especificação de requisitos é o processo de escrever os requisitos de usuário e de sistema em um documento de requisitos. Idealmente, os requisitos de usuário e de sistema devem ser claros, inequívocos, de fácil compreensão, completos e consistentes."

Na especificação de *software*, é encontrado dois tipos de requisitos, que são os requisitos funcionais e não funcionais, esses dois requisitos devem ser apresentados de maneira clara.

Segundo Sommerville (SOMMERVILLE, 2011, p. 59) Os requisitos de *software* são:

"Os requisitos de *software* são frequentemente classificados como requisitos funcionais e requisitos não funcionais: 1) Requisitos funcionais. São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer. 2) Requisitos não funcionais. São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo."

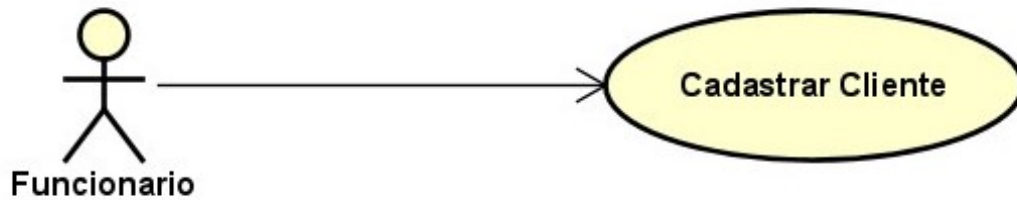
Segundo Pressman (PRESSMAN, 2010, p. 138) O processo de levantamento de requisito e a documentação do *software* é:

"O primeiro passo ao escrever um caso de uso é definir o conjunto de "atores" envolvidos na história. Atores são as diferentes pessoas (ou dispositivos) que usam o sistema ou produto no contexto da função e comportamento a ser descritos. Os atores representam os papéis que pessoas (ou dispositivos) desempenham enquanto o sistema opera. Definido de maneira um pouco mais formal, ator é qualquer coisa que se comunica com o sistema u o produto e que é externa

ao sistema em si. Todo ator possui uma ou mais metas ao usar o sistema."

Um exemplo em que se pode aplicar um diagrama de caso de uso em um sistema que cadastra a entrada de um cliente em uma loja. Dessa forma, um caso de uso seria o cadastro do cliente, como demonstrado na Figura 4:

Figura 4 – Exemplo de Caso de Uso



Fonte: Autoria própria

A Figura 4 mostra um exemplo de diagrama de caso de uso para cadastrar um cliente, em que há um ator Funcionário que é representado por um *stickman*, e um relacionamento que representa o acionamento do caso de uso Cadastrar Cliente, o qual é representado por uma elipse com a descrição do caso de uso.

Conforme mencionado anteriormente, a especificação de *software* aborda como o desenvolvimento será realizado e quais são os segmentos para iniciar o projeto.

3.1.2 Projeto e Implementação

O projeto de *software*, é onde ocorre a demonstração da estrutura que será utilizada para o desenvolvimento do *software*. Onde ocorre a criação dos modelos, estrutura de dados e das criações das interfaces do sistema. E a etapa da implementação é onde ocorre o início do desenvolvimento do que foi planejado na etapa de projeto.

Segundo Pressman (PRESSMAN, 2010, p. 209) um projeto de *software* é:

"O projeto de *software* é um processo iterativo por meio do qual os requisitos são traduzidos em uma "planta" para construir o *software*. Inicialmente, a planta representa uma visão holística do *software*. O projeto é representado em um alto nível de abstração - um nível que pode ser associado diretamente ao objetivo específico do sistema e aos requisitos mais detalhados de dados, funcionalidade e comportamento. À medida que ocorrem as iterações do projeto, refinamento subsequente leva a representações do projeto em níveis de abstração cada vez mais baixos. Estes ainda podem ser associados aos requisitos, mas a conexão é mais sutil."

Na etapa de implementação, onde ocorre o que foi planejado e descrito pelo projeto, nesta fase ocorre o início da codificação do sistema, onde será utilizado alguns *frameworks* para a codificação do sistema, alguns exemplos desses *frameworks* são: o banco de dados *MySQL*, no qual será utilizado para guardar os dados do sistema, para o desenvolvimento *back-end*, será utilizado Java e para o *front-end* será utilizado o Vaadin, para a criação das telas do sistema. Além desses, será utilizado a Linguagem Unificada de Modelagem (*Unified Modeling Language - UML*).

3.1.3 Validação de software

A etapa de validação do *software*, é onde ocorre toda parte de teste do *software*, onde é verificado se o sistema atende as necessidades do cliente, é esperado

que todas as partes seja executada da maneira que foi projetada, espera-se que nesta etapa possa detectar os erros que o sistema possa ter antes de ser entregue ao cliente.

Segundo Sommerville:

"O teste é destinado a mostrar que um programa faz o que é proposto a fazer e para descobrir os defeitos do programa antes do uso. Quando se testa o *software*, o programa é executado usando dados fictícios. Os resultados do teste são verificados à procura de erros, anomalias ou informações sobre os atributos não funcionais do programa. O processo de teste tem dois objetivos distintos: 1. Demonstrar ao desenvolvedor e ao cliente que o *software* atende a seus requisitos. Para *softwares* customizados, isso significa que deve haver pelo menos um teste para cada requisito do documento de requisitos. Para *softwares* genéricos, isso significa que deve haver testes para todas as características do sistema, além de suas combinações, que serão incorporadas ao release do produto. 2. Descobrir situações em que o *software* se comporta de maneira incorreta, indesejável ou de forma diferente das especificações. Essas são consequências de defeitos de *software*. O teste de defeitos preocupa-se com a eliminação de comportamentos indesejáveis do sistema, tais como panes, interações indesejáveis com outros sistemas, processamentos incorretos e corrupção de dados."(SOMMERVILLE, 2011, p.144)

Além disso a etapa de validação, realizar outros diversos testes a fim de concluir que o *software* esteja em funcionalidade adequada. Espera-se que com a etapa de validação possa analisar todo o *software*, a fim de concluir que os requisitos do sistema estejam de acordo com as especificações esperadas.

3.1.4 Evolução do software

A etapa de Evolução do *software*, e a etapa em que ocorre a melhoria ou correção de erros sem afetar as outras funcionalidades do sistema que já foram desenvolvidas ou que estão em produção. É de suma importância que haja o controle por meio de profissionais, que poderá solicitar ou sugerir mudanças no sistema.

Segundo Sommerville, (SOMMERVILLE, 2011, p. 164) a especificação de requisitos é:

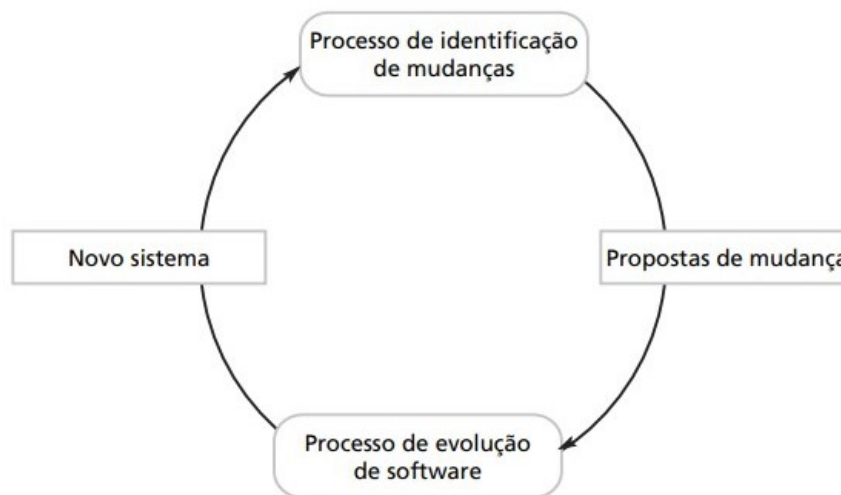
"O desenvolvimento de *software* não é interrompido quando o sistema é entregue, mas continua por toda a vida útil do sistema. Depois que o sistema é implantado, para que ele se mantenha útil é inevitável que ocorram mudanças —, mudanças nos negócios e nas expectativas dos usuários, que geram novos requisitos para o *software*. Partes do *software* podem precisar ser modificadas para corrigir erros encontrados na operação, para que o software se adapte

às alterações de sua plataforma de hardware e *software*, bem como para melhorar seu desempenho ou outras características não funcionais."

Além disso, na etapa de evolução *software*, ocorre alguns passos no processo de evolução. Segundo Sommerville (SOMMERVILLE, 2011, p.166):

"Em todas as organizações, as propostas de mudança no sistema são os acionadores para a evolução. As propostas de mudança podem vir de requisitos já existentes que não tenham sido implementados no release de sistema, solicitações de novos requisitos, relatórios de bugs do sistema apontados pelos *stakeholders* e novas ideias para melhoria do *software* vindas da equipe de desenvolvimento. Os processos de identificação de mudanças e de evolução de sistema são cíclicos e continuam durante toda a vida de um sistema (Figura 5)."

Figura 5 – Exemplo de Processos identificação de mudanças e de evolução



Fonte: SOMMERVILLE (2011)

A Figura 5, representa o ciclo de identificação de mudanças e de evolução com quatro processos, sendo eles: 1) Processo de identificação de mudanças, 2) Propostas de mudança, 3) Processo de evolução de *software*, 4) Novo sistema. A cada etapa ocorre um processo diferente, referente a evolução do *software*.

3.2 Desenvolvimento ágil

A abordagem principal escolhida para o desenvolvimento deste trabalho foi o desenvolvimento ágil, onde será utilizado dois procedimentos metodológicos, onde

o primeiro é a entrega incremental e a segunda o *frameworks* de gerenciamento de projeto SCRUM Sommerville (SOMMERVILLE, 2011) e Roger.S.Pressman (PRESSMAN, 2010).

Segundo Larman (LARMAN, 2007, p. 55) Desenvolvimento ágil é:

"Métodos de desenvolvimento ágil usualmente aplicam desenvolvimento iterativo e evolutivo de tempo limitado, empregam planejamento adaptativos, promovem entrega incremental e incluem outros valores e práticas que encorajam agilidade-resposta rápida e flexível à modificação."

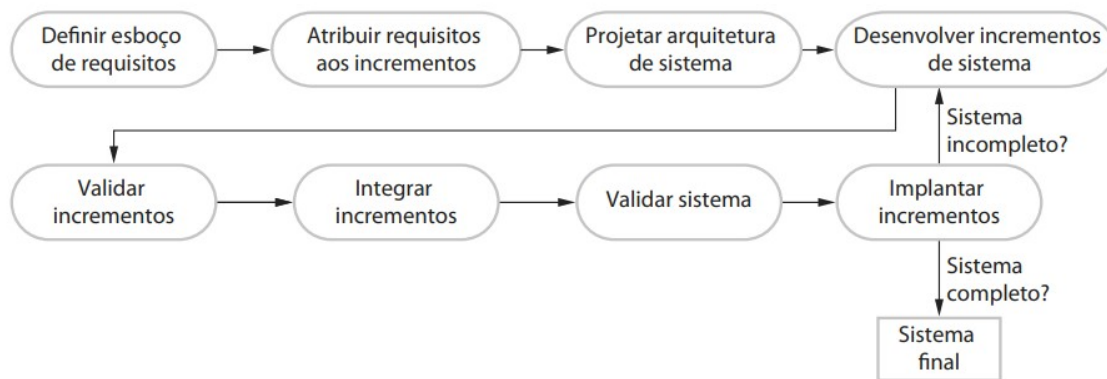
3.2.1 Entrega incremental

Para o primeiro procedimento será utilizado a entrega incremental Sommerville (SOMMERVILLE, 2011).

Que por sua vez oferece múltiplos processos para alcançar tal objetivo. Desse modo fica permitido a entrega de partes funcionais do *software* para o usuário na medida que as restantes das partes são desenvolvidas. Deste modo, a ferramenta poderá estar em funcionamento mesmo quando uma parte dele ainda está em construção.

Um exemplo da aplicação desse processo é permitir que o cliente solicite novos requisitos ao longo do desenvolvimento, e o desenvolvimento desses novos requisitos não impactam os requisitos que já estão em desenvolvimento. Isso porque os requisitos são desenvolvidos de acordo com a prioridade e as entregas de partes do *software* ocorre parcialmente até que todas as partes se integrem em um único sistema.

Figura 6 – Exemplo de Entrega incremental



Fonte: SOMMERVILLE (2011)

A Figura 6, representa cada etapa que ocorre na entrega incremental, é definido no início seu esboço de requisitos, a partir dessa etapa, começa o desenvolvimento, validação, e pôr fim a implementação do sistema. Caso haja mais modificações a serem desenvolvidas e implementada no sistema, essa modificação retornara para a parte de desenvolvimento, onde será feita todo o processo de validação do sistema novamente antes da entrega final.

A entrega incremental é um procedimento de múltiplos processos. Como citada pelo autor Sommerville:

"Uma vez que os incrementos do sistema tenham sido identificados, os requisitos dos serviços a serem entregues no primeiro incremento são definidos em detalhes, e esse incremento é desenvolvido. Durante o desenvolvimento, podem ocorrer mais análises de requisitos para incrementos posteriores, mas mudanças nos requisitos do incremento atual não são aceitas."(SOMMERVILLE, 2011, p.32)

"Quando um incremento é concluído e entregue, os clientes podem colocá-lo em operação. Isso significa aceitar a entrega antecipada de uma parte da funcionalidade do sistema. Os clientes podem experimentar o sistema, e isso os ajuda a compreender suas necessidades para incrementos posteriores. Assim que novos incrementos são concluídos, eles são integrados aos incrementos existentes para que a funcionalidade do sistema melhore com cada incremento entregue."(SOMMERVILLE, 2011, p. 32)

3.2.2 SCRUM

Segundo Pressman (PRESSMAN, 2010) e Sommerville (SOMMERVILLE, 2011) O *framework* de gerenciamento de projeto SCRUM é:

É uma metodologia ágil, o qual é composto por um conjunto de *sprints*, que por sua vez representa um intervalo de tempo defi-

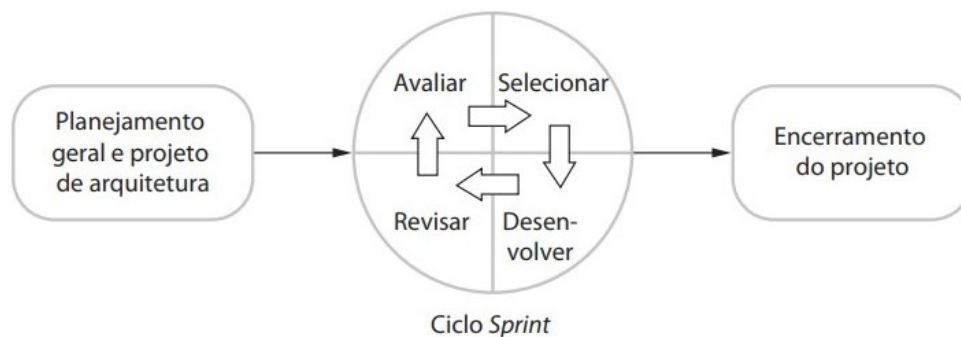
nido para desenvolver partes do sistema. Essa definição de tempo ocorre por meio da priorização de atividades definidas e na seleção de tais atividades a serem desenvolvidas prioritariamente. (SOMMERVILLE, 2011, p. 53)

O *framework* de gerenciamento de projeto SCRUM, é dividido em três fases, sendo elas, a primeira fase é o planejamento geral, logo em seguida ocorre a segunda fase, que é um conjunto de ciclos de *sprint*, e por fim ocorre a terceira fase e última, que ocorre o encerramento do projeto.

Segundo Pressman (PRESSMAN, 2010, p. 95). Os princípios de SCRUM são:

"Os princípios do Scrum são consistentes com o manifesto ágil e são usados para orientar as atividades de desenvolvimento dentro de um processo que incorpora as seguintes metodologias, ocorrem tarefas a realizar dentro de um padrão de processo chamado *sprint*. O trabalho realizado dentro de um *sprint* (o número de *sprints* necessários para cada atividade metodológica varia dependendo do tamanho e da complexidade do produto) é adaptada ao problema em questão e definido, e muitas vezes modificado em tempo real, pela equipe Scrum."

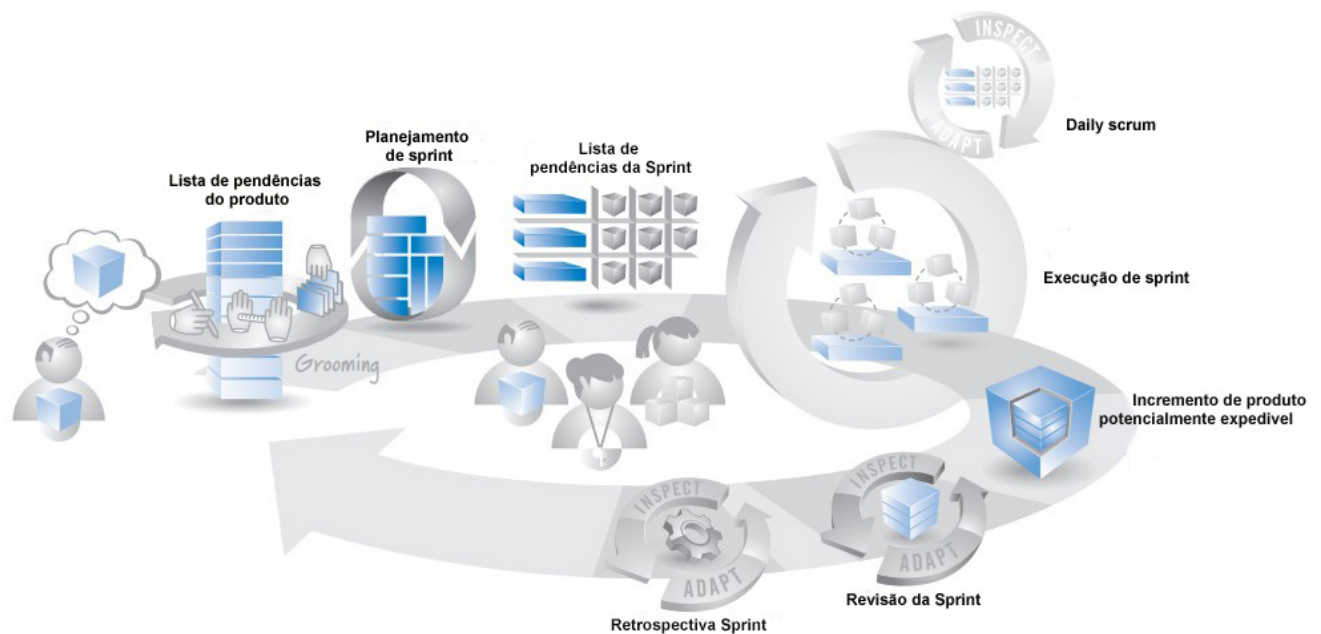
Figura 7 – Exemplo do SCRUM



Fonte: SOMMERVILLE (2011)

A Figura 7, representa o ciclo de *sprint*, que ocorre no SCRUM. Além disso, é mostrado na figura cada fase que ocorre no ciclo, será iniciado pelo planejamento geral e projeto de arquitetura, seguido pelo processo do *sprint*, e logo em seguida será direcionado para o encerramento do projeto.

Figura 8 – Exemplo Planejamento de Sprint



Fonte: Adaptado de (RUBIN, 2012)

A Figura 8, apresenta todo o planejamento do *sprint*. Com a primeira etapa que é a lista de pendências de produto, nesta fase ocorre tudo que deve ser implementado no ciclo da *sprint*, tais como funcionalidades, requisitos, melhorias e correções. Em seguida a etapa planejamento de *sprint*, em que ocorre no início de cada ciclo de trabalho, no qual a equipe irá definir cada objetivos e as tarefas que serão desenvolvidas em cada *sprint*. Logo em seguida acontece a terceira etapa, lista de pendências de *sprint*. Nesta fase ocorre a criação de uma lista de itens, em que a equipe planeja executar durante cada ciclo da *sprint*. Na próxima etapa ocorre a *daily scrum*, esta fase ocorre uma atividade que auxilia a equipe a se auto-organizar e ajuda a executar um trabalho melhor. A próxima etapa é o incremento de produto potencialmente expedível, nesta etapa ocorre a entrega da parte que foi desenvolvida durante o Ciclo *sprint* e deve estar em perfeito funcionamento. Logo em seguida é a etapa revisão da *sprint*, que ocorre no final de cada ciclo onde os programadores recebem *feedback*, para poderem melhorar o resto do sistema que ainda será desenvolvido. E por fim, a etapa retrospectiva *sprint*, ocorre antes de iniciar outro ciclo, nesta etapa os desenvolvedores se reúnem e compartilham como foi a experiência obtida durante o ciclo.

3.2.3 1º Fase - Planejamento Geral

Nessa primeira fase, é realizado uma reunião com a equipe para definirem, o projeto geral e cada *sprint*, nessa fase fica definido o que cada um da equipe irá realizar durante o desenvolvimento do projeto.

Segundo Jeff Sutherland (SUTHERLAND, 2014). O planejamento da *sprint* é:

"Esta é a primeira das reuniões Scrum. A equipe, o Mestre Scrum e o Dono do Produto se reúnem para planejar o Sprint, que sempre tem uma duração definida de tempo menor que um mês. A maioria das pessoas define Sprints de uma ou de duas semanas. As equipes olham para as tarefas no topo das Pendências e estimam o quanto podem fazer naquele Sprint. Se a equipe já está trabalhando a alguns Sprints, ela deve pegar tarefas que totalizem o mesmo número de pontos do Sprint anterior. Esse número é conhecido como a Velocidade da equipe. O Mestre Scrum e a equipe devem tentar aumentar o número de pontos a cada Sprint. Essa é outra chance para a equipe e o Dono do Produto se certificarem que todos entendem como os itens vão satisfazer a visão. Além disso, durante essa reunião todos devem concordar com um Objetivo do Sprint."

E segundo Kenneth S. Rubin (RUBIN, 2012, p.336). O planejamento da *sprint* é:

"O planejamento de sprint se baseia em um conjunto de informações que orientam a equipe de desenvolvimento a determinar o valor que pode ser entregue realisticamente até o final do sprint. A primeira e mais importante informação para o planejamento de sprint é o *backlog* do produto que foi refinado antes do planejamento do sprint, de modo que os itens mais prioritários atendam à definição de pronto da equipe Scrum."

3.2.4 2º Fase - Sprint

Nessa segunda fase, é onde ocorre o ciclo da *sprint*, onde será executado, tudo que foi planejado na fase anterior, nessa etapa ocorre a principal fase do SCRUM.

Segundo Sommerville (SOMMERVILLE, 2011, p. 50) A fase principal é:

"A característica inovadora do Scrum é sua fase central, chamada ciclos de *sprint*. Um *sprint* do Scrum é uma unidade de planejamento na qual o trabalho a ser feito é avaliado, os recursos para o desenvolvimento são selecionados e o *software* é implementado. No fim de um *sprint*, a funcionalidade completa é entregue aos *stakeholders*."

E segundo Kenneth S. Rubin (RUBIN, 2012, p. 338). A fase principal é:

"Durante a primeira etapa, a equipe de desenvolvimento determina sua capacidade de concluir o trabalho e depois prevê os itens da lista de pendências de produto que acredita ser capaz de entregar até o final da sprint. Portanto, se a equipe acredita que pode realizar 40 pontos de história, ela selecionará cerca de 40 pontos de história de trabalho. Durante a segunda etapa, a equipe adquire confiança em sua capacidade de concluir os itens que previu na primeira etapa, criando um plano. A maioria das equipes cria esse plano dividindo os itens da lista de pendências de produto em um conjunto de tarefas e, em seguida, estimando (em horas) o esforço necessário para concluir cada tarefa. A equipe então compara a estimativa de horas das tarefas com sua capacidade, em termos de horas, para ver se seu compromisso inicial foi realista."

3.2.5 3º Fase - Encerramento do Projeto

Na última fase, se encontra o encerramento do projeto, onde é finalizado todo o ciclo da *sprint*, e o projeto já estará pronto para a entrega ao usuário. Nesta etapa toda a equipe avalia tudo que foi aprendido durante o desenvolvimento do projeto.

Segundo Kenneth S. Rubin (RUBIN, 2012). O Encerramento do Projeto é:

"A revisão da sprint oferece a todos que contribuem para o desenvolvimento do produto a oportunidade de inspecionar e adaptar o que foi construído até o momento. Ela proporciona uma visão transparente do estado atual do produto, incluindo quaisquer verdades inconvenientes. É o momento de fazer perguntas, fazer observações ou sugestões e ter discussões sobre como avançar da melhor maneira possível, considerando as realidades atuais."

3.3 UML

A UML é uma linguagem de modelagem Unificada, utilizada para a modelagem de sistema, a partir da criação de diagramas, esses diagramas são desenvolvidos para auxiliar o desenvolvedor na criação e modelagem de todo o sistema.

Segundo Guedes (GUEDES, 2011, p. 19) a UML é:

"A UML— *Unified Modeling Language* ou Linguagem de Modelagem Unificada - é uma linguagem visual utilizada para modelar *softwares* baseados no paradigma de orientação a objetos. É uma linguagem de modelagem de propósito geral que pode ser aplicada a todos os domínios de aplicação. Essa linguagem tornou-se, nos últimos anos, a linguagem-padrão de modelagem adotada internacionalmente pela indústria de engenharia de *software*."

Além disso, a UML é utilizada principalmente em engenharia de *software* para a representação de *software*, que atualmente utilizam requisitos, estruturas,

software que necessitam de padrões mais específicos para que comuniquem um com outro.

3.3.1 Porque utilizar UML?

É de suma importância a criação de diagramas de UML para a modelagem do sistema, a fim de manter um controle do que foi solicitado no início pelo usuário e o que foi solicitado durante o desenvolvimento do sistema.

Ao aplicar a UML para a modelagem de sistemas, deve-se criar alguns dos principais diagramas, a fim de mostrar diferentes perspectivas sobre o *software* a ser desenvolvido. Dentre alguns diagramas, pode-se citar: 1) diagrama de Caso de uso, 2) diagrama de Classe, 3) diagrama de Atividade e 4) diagrama de Sequência. O controle e a modelagem do sistema ficarão mais clara e sutil, no momento em que foi designado para alguma mudança.

Segundo Guedes (GUEDES, 2011, p. 20-21) a modelagem é:

"Na realidade, por mais simples que seja, todo e qualquer sistema deve ser modelado antes de se iniciar sua implementação, entre outras coisas, porque os sistemas de informação frequentemente costumam ter a propriedade de "crescer", isto é, aumentar em tamanho, complexidade e abrangência. Muitos profissionais costumam afirmar que sistemas de informação são "vivos", porque nunca estão completamente finalizados. Na verdade, o termo correto seria "dinâmicos" pois normalmente os sistemas de informação estão em constante mudança. Tais mudanças são devidas a diversos fatores."

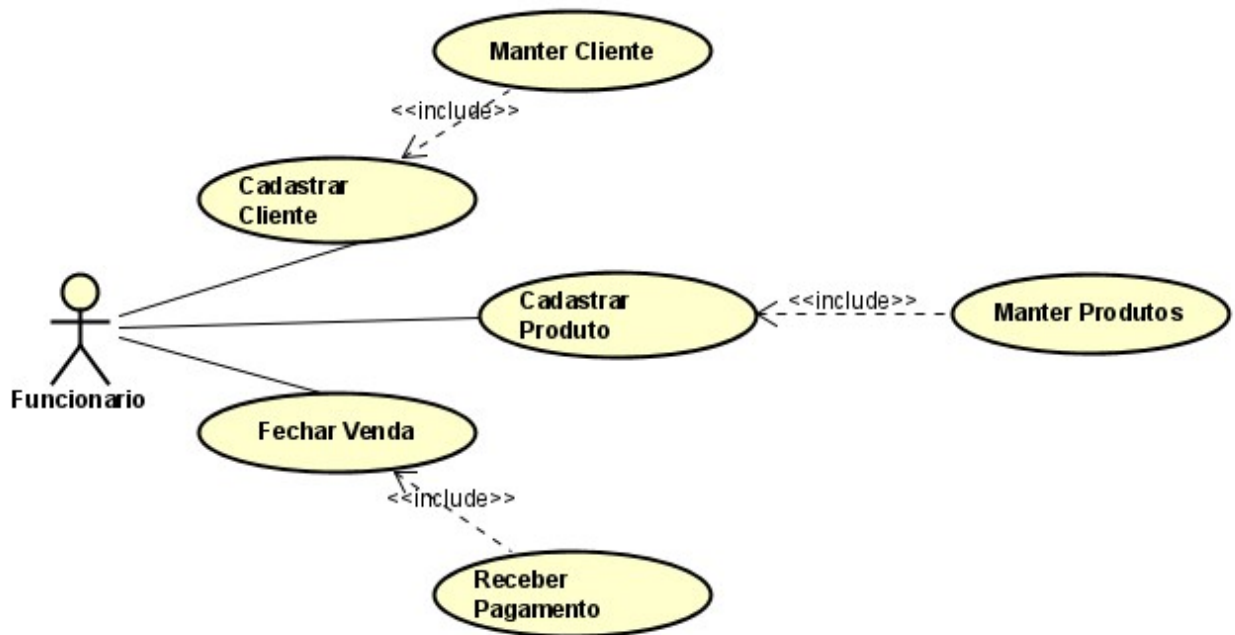
3.3.2 Diagrama de Caso de Uso

O diagrama de caso de uso é amplamente utilizado na fase de levantamento de requisitos, onde são identificadas as funcionalidades do sistema de forma clara. Nesse diagrama, as relações *include* e *extends* são usadas para representar as necessidades do usuário no sistema. A relação *include* indica uma funcionalidade necessária para a execução de outra, enquanto a relação *extends* representa uma extensão opcional de uma funcionalidade existente.

Segundo (GUEDES, 2011, p. 30) o Caso de Uso é:

"O diagrama de casos de uso é o diagrama mais geral e informal da UML, utilizado normalmente nas fases de levantamento e análise de requisitos do sistema, embora venha a ser consultado durante todo o processo de modelagem e possa servir de base para outros diagramas. Apresenta uma linguagem simples e de fácil compreensão para que os usuários possam ter uma ideia geral de como o sistema irá se comportar."

Figura 9 – Diagrama de Caso de uso



Fonte: Autoria própria

A Figura 9 mostrar o exemplo de um diagrama de Caso de Uso. Cada caso de uso representa uma função ou recurso que o sistema oferece, e é representado por uma elipse contendo a descrição do caso de uso, onde há um ator funcionário, onde está relacionado com o acionamento do Caso de Uso de cadastrar cliente, que possui um *include* de manter cliente, outro Caso de Uso de cadastrar produto, onde possui um *include* de manter produtos, e por último um relacionamento de fechar venda, com um *include* de receber pagamento.

3.3.3 Diagrama de Classes

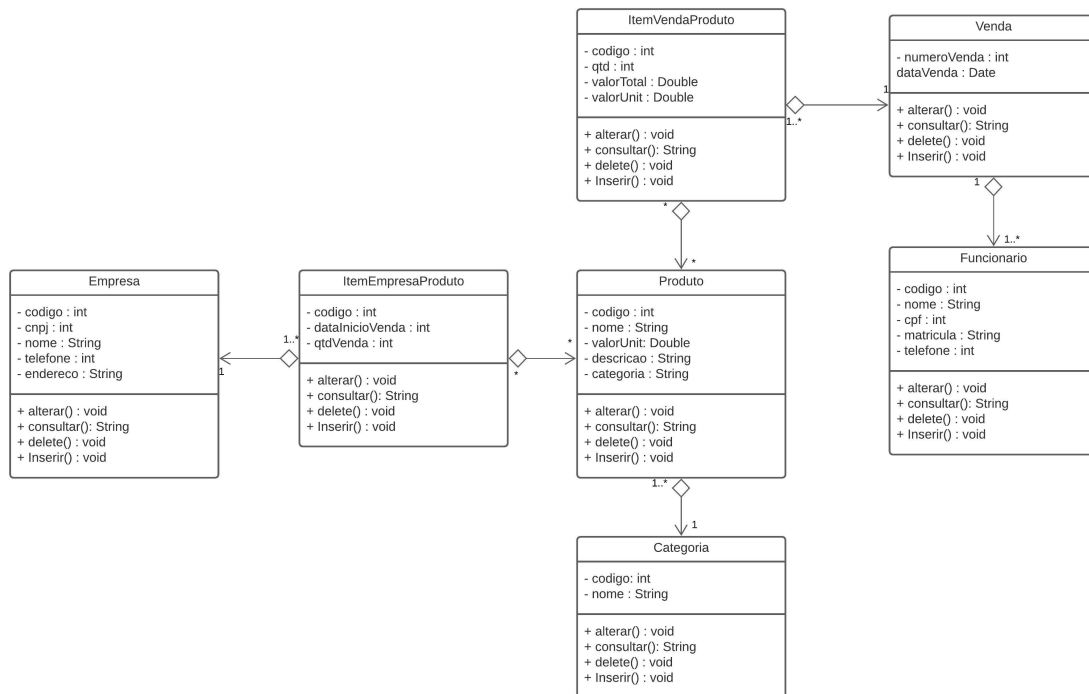
O diagrama de classe é considerado o diagrama principal, a partir dele, o desenvolvedor pode visualizar como será o sistema, pela sua classe, atributos e métodos de forma simples e mais organizada.

Segundo (GUEDES, 2011, p. 30) o diagrama de Classe é:

"O diagrama de classes é provavelmente o mais utilizado e é um dos mais importantes da UML. Serve de apoio para a maioria dos demais diagramas. Como o próprio nome diz, define a estrutura das

classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam e trocam informações entre si."

Figura 10 – Diagrama de Classes



Fonte: Autoria própria

A Figura 10 representa o diagrama de classe a seguir, representa a estrutura das classes em um sistema. Cada classe é representada por um retângulo contendo seus respectivos atributos e métodos. Que possui funcionário, venda, itemVendaProduto, produto, itemEmpresaProduto, empresa e categoria que possui seus respectivos atributos e métodos. Além disso, é possível visualizar os relacionamentos entre cada classe. Assim, fica fácil o entendimento do programador sobre a estrutura das classes do sistema.

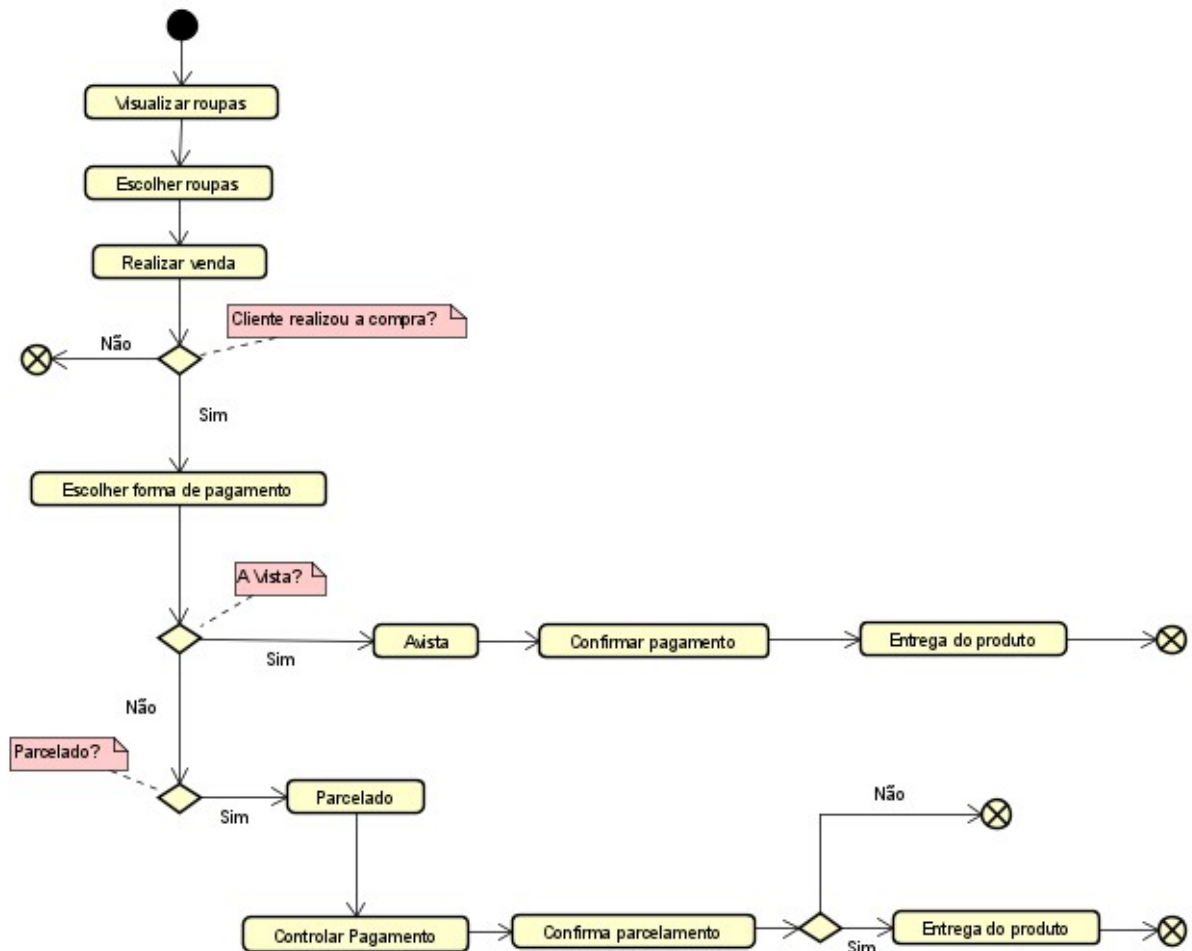
3.3.4 Diagrama de Atividade

O diagrama de atividade é utilizado para demonstrar todos os passos decorridos de um processo, a partir do diagrama, é possível identificar todas as atividades que se relacionam e como as decisões são tomadas dentro do processo.

Segundo (GUEDES, 2011, p. 30) o diagrama de Atividade é:

"O diagrama de atividade preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica, podendo esta ser representada por um método com certo grau de complexidade, um algoritmo, ou mesmo por um processo completo. O diagrama de atividade concentra-se na representação do fluxo de controle de uma atividade."

Figura 11 – Diagrama de Atividade



Fonte: Autoria própria

A Figura 11, é uma representação visual que mostra o fluxo de atividades em um processo ou sistema. Representa o fluxo de atividade de uma loja de roupas. O cliente visualizar as roupas, seleciona as roupas e após isso é realizado a venda, caso o cliente opte por não realizar a compra, a compra é cancelada e finalizado o fluxo, caso resolva comprar, o cliente escolher a forma de pagamento, se for à vista, confirma o pagamento e é entregue o produto e finaliza o fluxo. Caso a compra não for à vista e sim parcelado ele passa pelo controle de pagamento, em seguida confirma o pagamento, caso não confirmado a compra é cancelada, é finalizado o fluxo, e caso for confirmado, é entregue o produto e finalizado a compra.

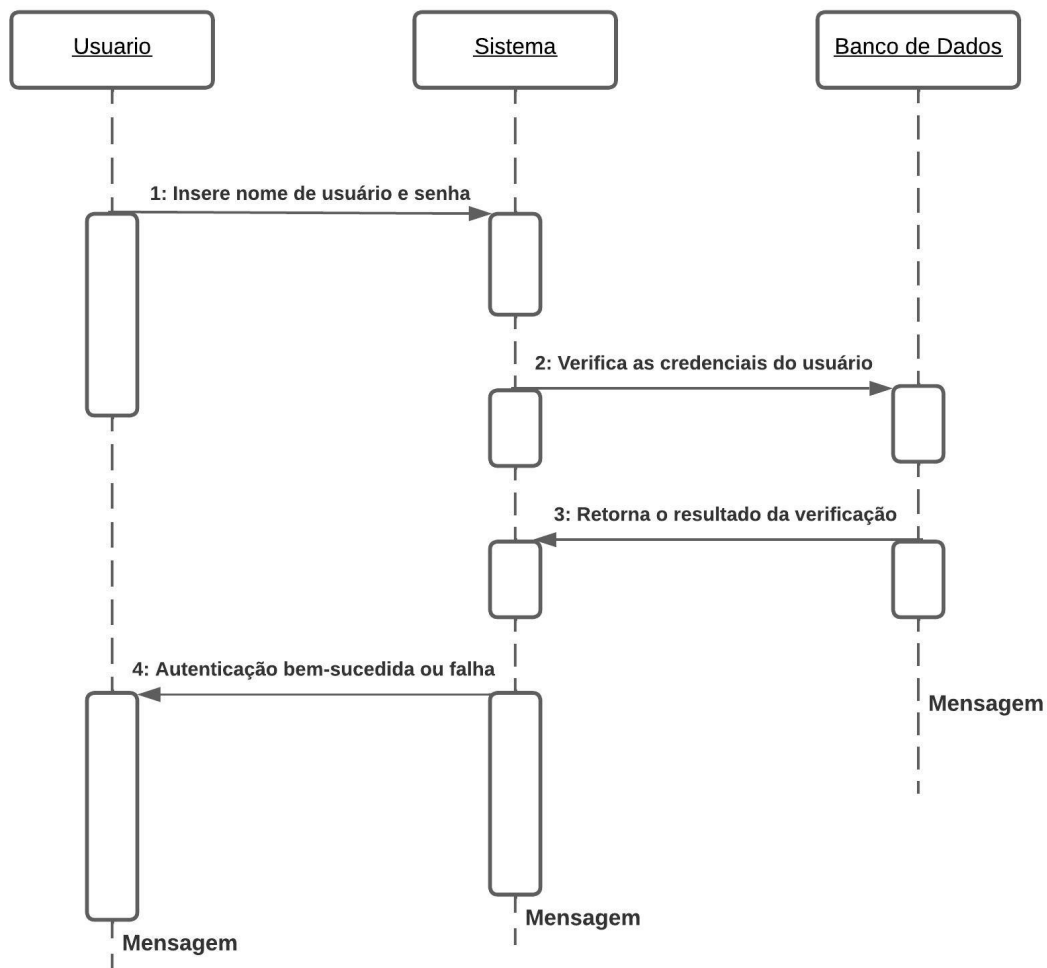
3.3.5 Diagrama de Sequência

O diagrama de Sequência é umas das modelagens da *UML*, utilizada em desenvolvimento de *software*. Esse diagrama tem a finalidade de mostrar qual são as interações dos objetos em um sistema durante um determinado tempo.

Segundo (GUEDES, 2011, p. 33) o diagrama de Sequência é:

"O diagrama de sequência é um diagrama comportamental que preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Em geral, baseia-se em um caso de uso definido pelo diagrama de mesmo nome e apoia-se no diagrama de classes para determinar os objetos das classes envolvidas em um processo. Um diagrama de sequência costuma identificar o evento gerador do processo modelado, bem como o ator responsável por esse evento, e determina como o processo deve se desenrolar e ser concluído por meio da chamada de métodos disparados por mensagens enviadas entre os objetos."

Figura 12 – Diagrama de Sequência



Fonte: Autoria própria

A Figura 12, é uma representação visual que mostra a ordem em que as ações ocorrem em um sistema. Ele apresenta como os objetos interagem entre si ao longo do tempo, mostrando a troca de mensagens entre eles. Esse diagrama representa um sistema de login, o usuário insere login e senha no sistema, que verifica as credenciais no banco de dados, retornando ao sistema o resultado da verificação, no momento em que o sistema retorna ao usuário a autenticação bem-sucedida ou falha.

4 Desenvolvimento do software AquaHaven

No Capítulo 3, foi ressaltado o Desenvolvimento Ágil como um modelo de desenvolvimento de *software* composto por duas abordagens principais que abrangem todo o ciclo de desenvolvimento. Neste capítulo, será mostrado os detalhes sobre a aplicação das etapas do desenvolvimento ágil e do desenvolvimento do *software* no sistema AquaHaven.

4.1 Definições do Scrum no desenvolvimento do AquaHaven

Durante o desenvolvimento do sistema AquaHaven, foi adotado o *framework* Scrum para gerenciar o processo de desenvolvimento de forma ágil. O Scrum é caracterizado pela entrega contínua de valor ao cliente por meio de iterações chamadas de *sprint*. Para este projeto específico, foi estabelecido um total de nove *sprint*, cada uma com duração média de 20 dias cada. Até o momento, foram concluídas quatro *sprint*. Na primeira *sprint*, foi realizada uma pesquisa aprofundada sobre o tema. Na segunda *sprint*, iniciou-se a fase de escrita. Na terceira *sprint*, foram criados os diagramas necessários, enquanto na quarta *sprint*, iniciou-se a criação das primeiras telas do sistema. Restam ainda cinco *sprint*, nas quais ocorrerá o desenvolvimento do software, a validação, os testes e a entrega final do sistema.

A abordagem Scrum adotada no desenvolvimento do sistema AquaHaven possibilita a entrega incremental do produto ao longo das *sprints*. Essa metodologia permite a incorporação contínua de novas funcionalidades e aprimoramentos ao sistema a cada *sprint* concluída, dessa forma, o produto estará em constante evolução. Desse modo, permiti que haja ajustes e adaptações conforme as necessidades identificadas ao longo do processo. Dessa maneira, a entrega incremental contribui para um desenvolvimento mais ágil e satisfatório para o cliente.

4.2 Definição dos requisitos

A metodologia Scrum foi adotada para o desenvolvimento do *software*, e a primeira etapa desse processo é a definição dos requisitos. Para iniciar essa etapa

de especificação do *software*, foi realizada uma reunião com o empresário, em que o mesmo descreveu seu processo de controle que utilizava em sua empresa. Na reunião, percebeu-se que o controle ocorre por meio de fichas, caderno-caixa, planilha e outros meios de gerenciamento de dados, as quais disponibilizam os dados para a tomada de decisão no que diz respeito à empresa. Além disso, verificou-se que após o controle ser realizado, os arquivos eram guardados para controles futuros. A metodologia Scrum será aplicada para desenvolver um *software* que substitua esses métodos de controle tradicionais, oferecendo uma solução moderna e eficiente para a gestão dos dados da empresa.

4.3 Projeto de sistemas

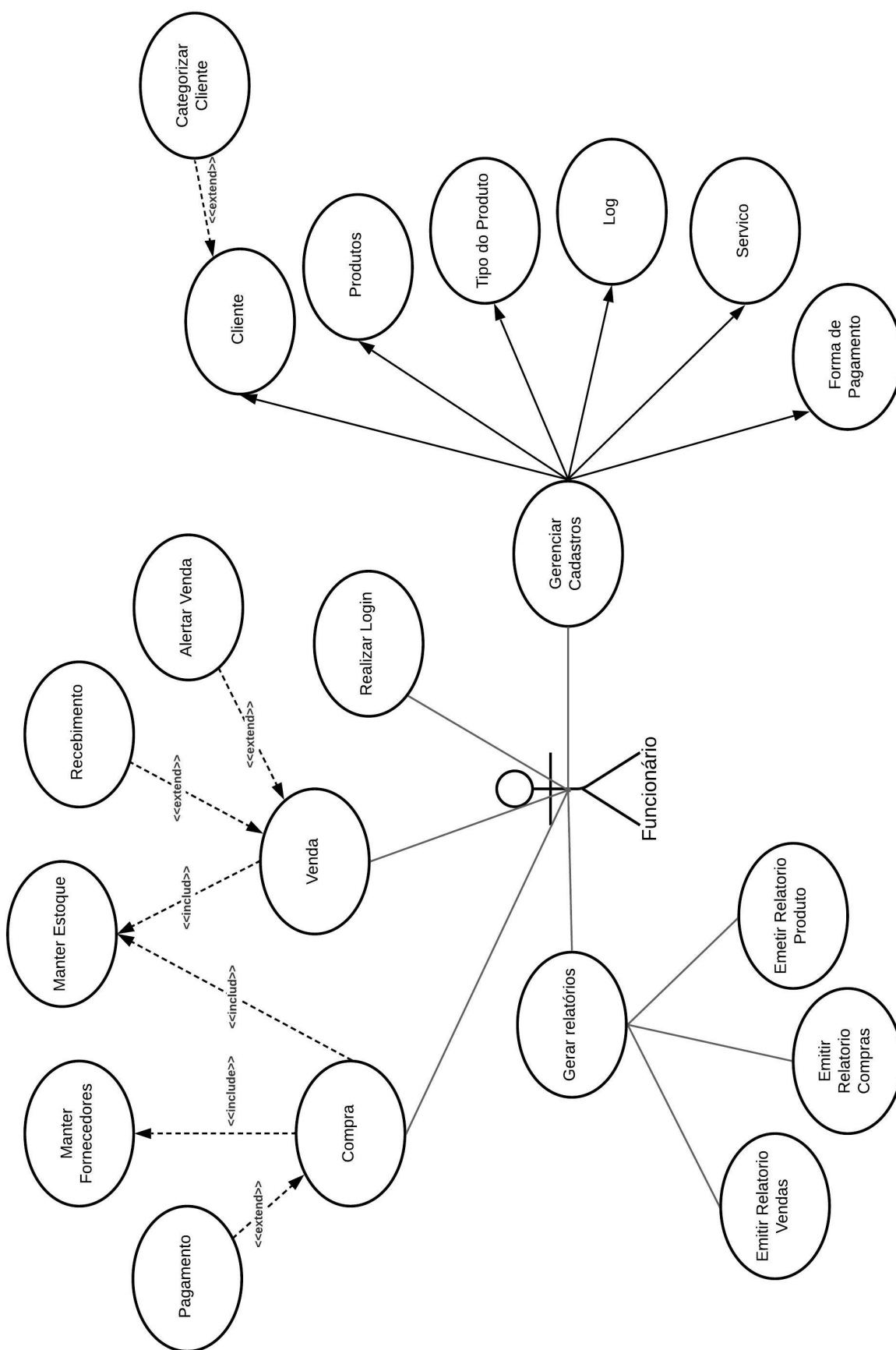
No contexto do desenvolvimento de *software*, foi utilizado a metodologia Scrum, a segunda etapa é a realização do projeto do sistema. Nessa etapa, serão utilizados os princípios e práticas do Scrum para orientar o trabalho. Uma abordagem comum é a criação de diagramas UML, como diagramas de caso de uso, diagramas de classe e diagramas de atividade. Esses diagramas serão desenvolvidos para mapear e visualizar as funcionalidades e interações do sistema AquaHaven ao longo de seu ciclo de desenvolvimento. Com o Scrum, esses diagramas serão constantemente revisados e atualizados durante as *sprints*, isso, permitiu um processo iterativo e adaptativo de projeto e refinamento do sistema.

Na etapa a seguir será apresentado os diagramas.

4.3.1 Diagrama de caso de uso do sistema AquaHaven

O diagrama de caso de uso é uma representação visual que descreve as funcionalidades de um sistema e a interação dos atores com essas funcionalidades. Cada caso de uso é representado por uma elipse com uma descrição. O diagrama a seguir mostra os casos de uso do sistema AquaHaven. Em que o usuário poderá acessar os casos de uso, os quais são representados por uma elipse com a descrição de cada um. Por exemplo há o caso de uso gerenciar cadastro que está relacionado com os demais casos de uso de cadastro, tais como cadastro de cliente, produto, tipo de produto, log, serviço e forma de pagamento. Além desses, há o caso de uso realizar login, venda e compra e os demais casos de usos, como os relatórios. Essa descrição é referente a Figura 13 que apresenta o diagrama de caso de uso do sistema AquaHaven.

Figura 13 – Diagrama de Caso de Uso

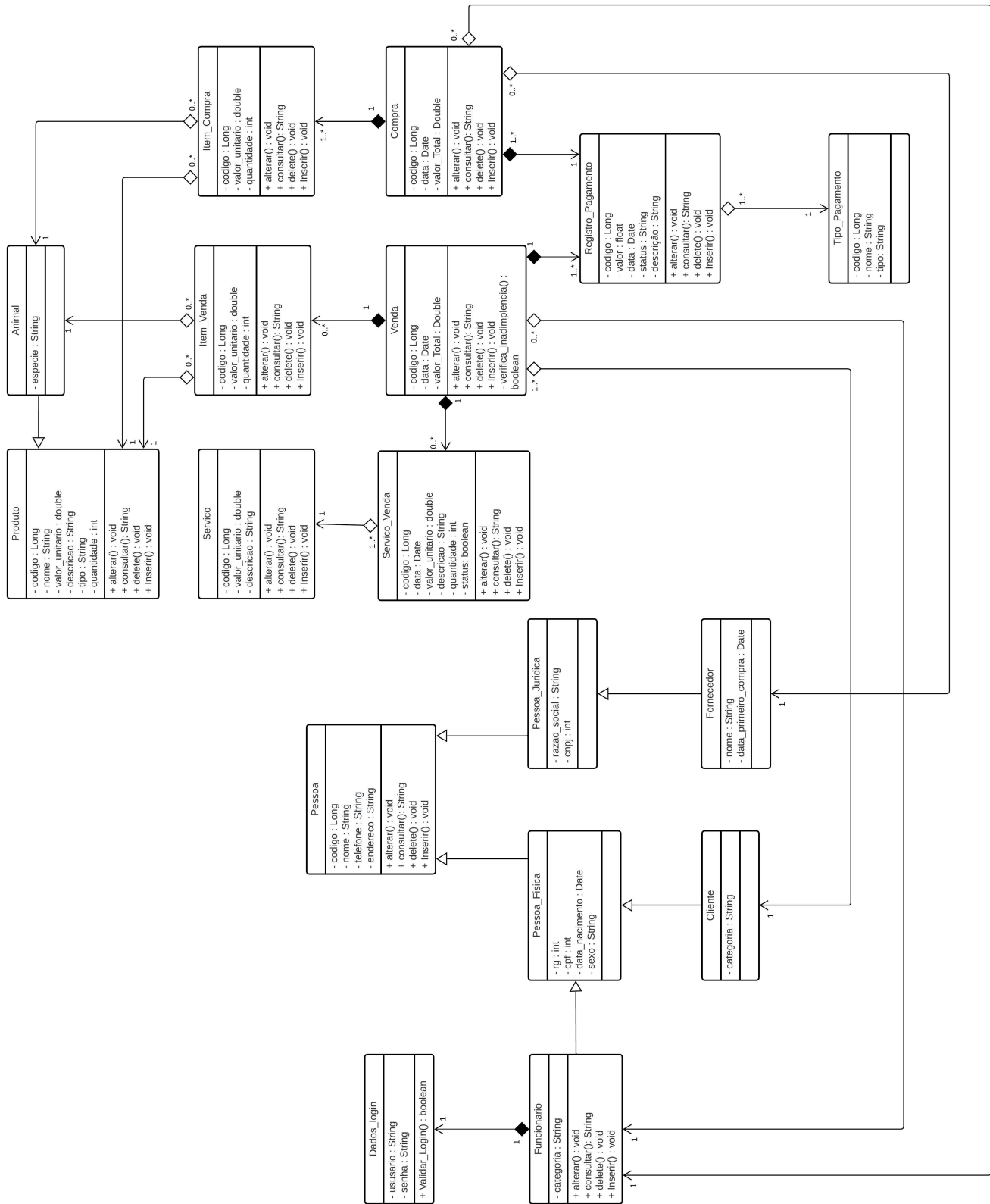


Fonte: Autoria Própria

4.3.2 Diagrama de Classes do sistema AquaHaven

O diagrama de classes descreve a estrutura do sistema, incluindo as classes, seus atributos e métodos, bem como as associações entre as classes. O diagrama de classe é um dos diagramas principais do sistema AquaHaven. Ele apresenta a classe "Login", que é uma composição do funcionário e contém usuário e senha para autenticar a entrada no sistema do funcionário. Além disso, existem as classes "Cliente", "Pessoa Física", "Fornecedor" e "Pessoa Jurídica", que herdam seus atributos e métodos da classe "Pessoa". As classes "Cliente" e "Fornecedor" são composições das classes "Pessoa Física" e "Pessoa Jurídica", respectivamente. A classe "Cliente" está relacionada como uma composição da classe "Vendas", que, por sua vez, está relacionada com as classes "ServiçosVenda" que por sua vez está relacionada com a classe "Serviço" por meio de uma composição e a classe "Venda" está relacionada com "Item Venda" como uma composição. A classe "Item Venda" herda da classe "Produto", sendo que "Animal" por sua vez herda da classe "Produto". Além disso, a classe "Venda" está relacionada com "Pagamento" por meio de uma composição. "Pagamento" possui um tipo de relacionamento de composição com o "Tipo de Pagamento". A classe "Compra" está relacionada por composição com a classe "Item Compra", que herda de "Produto" e "Animal". E Todas as classes possuem seus respectivos atributos e métodos. Essa descrição se refere à Figura 14, que ilustra o diagrama de classe do sistema AquaHaven.

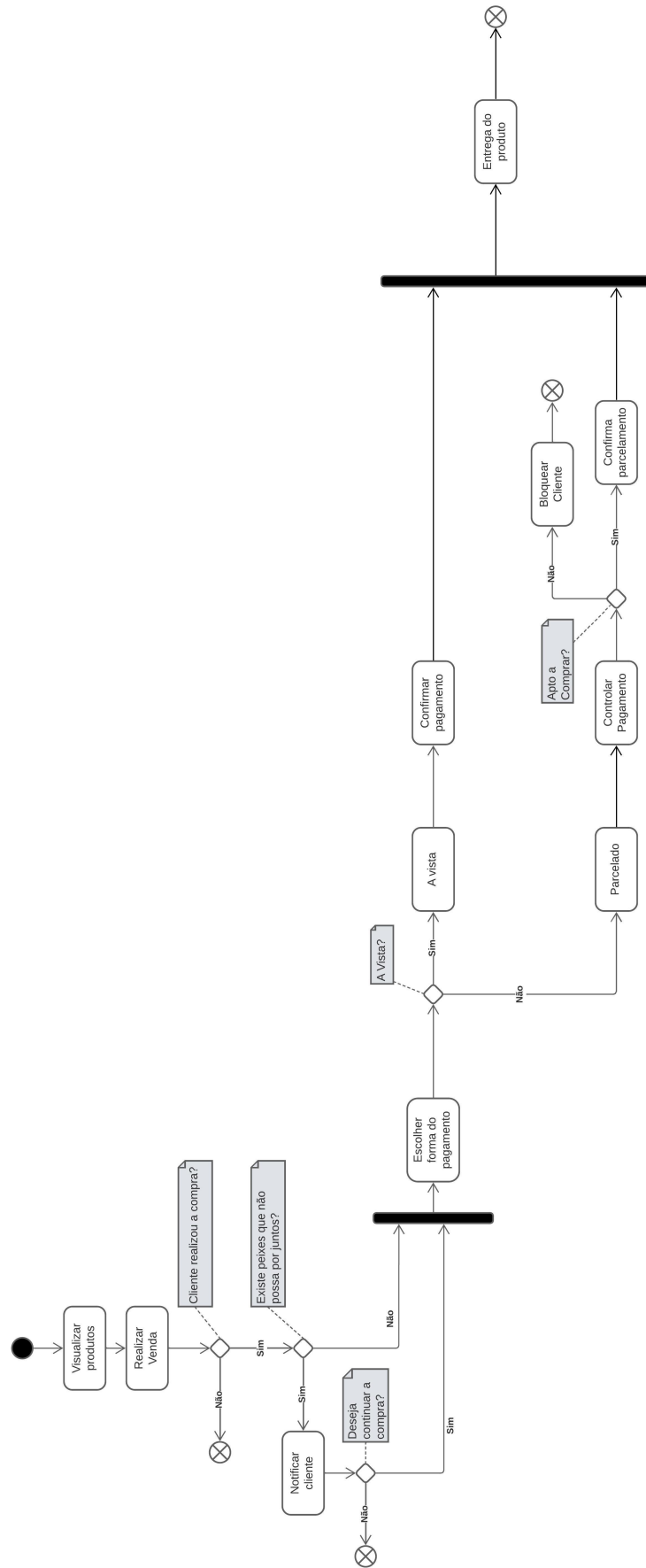
Figura 14 – Diagrama de Classe



4.3.3 Diagrama de Atividade de Venda do sistema AquaHaven

O diagrama de atividade de venda representa o fluxo de atividades envolvidas no processo de venda. Ele é usado para descrever a sequência de passos e decisões que ocorrem em uma determinada atividade. O processo é iniciado com a visualização dos produtos, em seguida é realizado a venda, se o cliente optou por não continuar a compra, o processo é finalizado, e caso ele continue é verificado se a algum peixe na compra que não possa conviver no mesmo ambiente, se sim, o cliente é notificado sobre e poderá escolher se vai continuar ou parar a compra, caso escolha cancelar a compra é finalizado o processo, caso escolha continuar, segue para a escolha do pagamento. Se o pagamento for à vista, confirmamos o pagamento, assim a nota fiscal é emitida, e o produto é entregue ao cliente, após isso é finalizado o processo. Caso o pagamento foi parcelado, será escolhido a forma de pagamento, será passado no controle para verificar se está apto a comprar, caso não esteja o cliente é bloqueado e finalizado o processo, mas se estar apto a comprar, confirma o parcelamento após isso será realizada a emissão da nota fiscal, e após isso é entregue ao cliente junto com a mercadoria e o processo é finalizado. Essa descrição se refere à Figura 15, que ilustra o diagrama de atividade de venda no sistema AquaHaven.

Figura 15 – Diagrama de Atividade de Venda

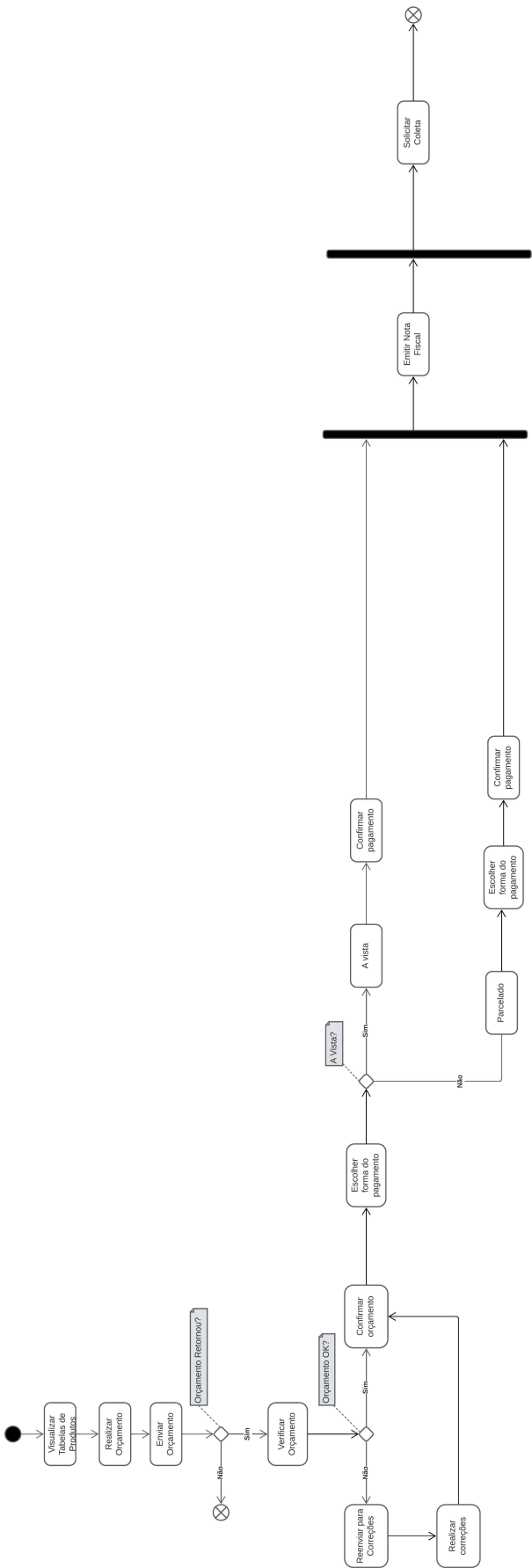


Fonte: Autoria Própria

4.3.4 Diagrama de Atividade de Compra do sistema AquaHaven

O diagrama de atividade da compra representa o fluxo de atividades envolvidas no processo de compras. Ele é usado para descrever a sequência de passos e decisões que ocorrem em uma determinada atividade. O processo é iniciado com a visualização das tabelas de produtos, em seguida é criado o orçamento e enviado para o fornecedor, caso o orçamento não tenha retornado finalizamos o processo, mas caso tenha retornado verificamos o mesmo, e seguimos com a confirmação do orçamento com o fornecedor e seguimos para a escolha da forma de pagamento, se o pagamento for a vista, confirmamos o pagamento e é realizado a emissão da nota fiscal, após e feita a solicitação da coleta do pedido para o envio, após isso é finalizado o processo. Caso o pagamento tenha sido parcelado, será escolhido a forma de pagamento, após a escolha será confirmado o pagamento, e é realizado a emissão da nota fiscal, após e feita a solicitação da coleta do pedido para o envio, após isso é finalizado o processo. Caso não tenha, é solicitado a coleta para o pedido e após isso é enviado, finalizando o processo. Esse processo é referente a figura 16 do sistema AquaHaven.

Figura 16 – Diagrama de Atividade de Compra



Fonte: Autoria Própria

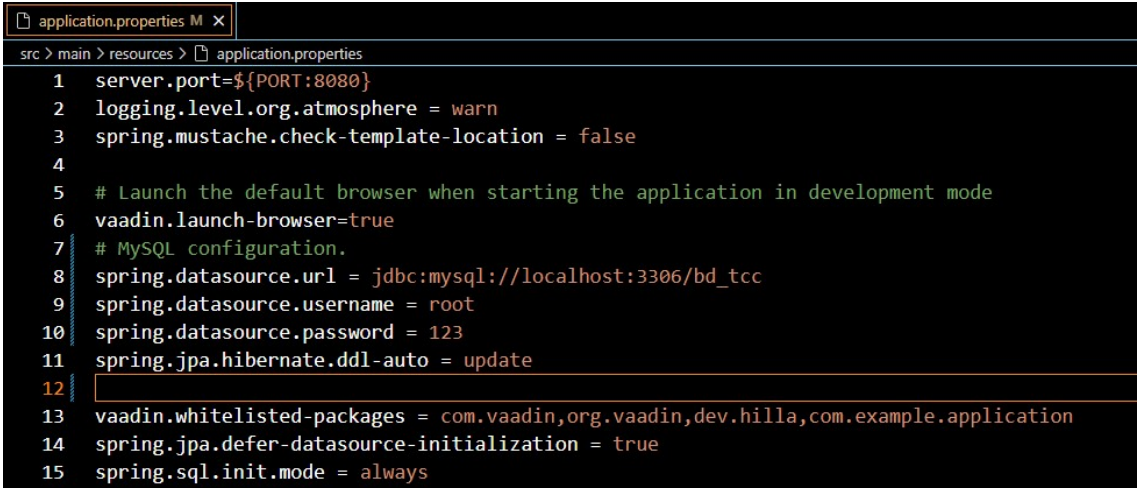
4.4 Operação e manutenção

Durante a etapa de desenvolvimento ágil, o sistema foi dividido em iterações menores, permitindo entregas incrementais e a manutenção contínua do sistema. Na fase de entrega será realizada uma reunião com o empresário a fim de fornecer orientações sobre a utilização do sistema, e caso haja alguma sugestão de aprimoramento, será analisada de forma adequada para a realização da melhoria no sistema, garantindo que não haja interferência nas funcionalidades já existentes.

4.5 Implementação e validação

Em conformidade com os princípios do desenvolvimento ágil, chegamos à etapa de implementação. Durante essa fase, foram selecionadas as seguintes ferramentas de desenvolvimento: a linguagem de programação escolhida foi Java, a IDE utilizada foi o Visual Studio Code, o banco de dados adotado foi o MySQL e o *framework* Vaadin foi utilizado para a criação das interfaces. Além disso, optamos por utilizar o *Spring Boot* como base do projeto, que inclui o Hibernate como uma estrutura ORM para mapear os códigos Java com o banco de dados.

Figura 17 – Conexão com o Banco de Dados

A screenshot of a code editor window titled 'application.properties M'. The editor shows the following configuration for a Spring Boot application:

```
1 server.port=${PORT:8080}
2 logging.level.org.atmosphere = warn
3 spring.mustache.check-template-location = false
4
5 # Launch the default browser when starting the application in development mode
6 vaadin.launch-browser=true
7 # MySQL configuration.
8 spring.datasource.url = jdbc:mysql://localhost:3306/bd_tcc
9 spring.datasource.username = root
10 spring.datasource.password = 123
11 spring.jpa.hibernate.ddl-auto = update
12
13 vaadin.whitelisted-packages = com.vaadin,org.vaadin,dev.hilla,com.example.application
14 spring.jpa.defer-datasource-initialization = true
15 spring.sql.init.mode = always
```

Fonte: Autoria Própria

A Figura 17 mostra um exemplo de código no qual faz a conexão com o banco de dados MySQL, a partir da configuração do Hibernate.

A validação é uma parte crucial do processo. A validação garante que o sistema atenda aos requisitos e expectativas estabelecidos durante a fase de planejamento. Visa assegurar que os objetivos do cliente foram compreendidos e implementados de maneira precisa.

Durante a etapa de validação, o cliente foi introduzido ao *software* e instruído sobre o seu funcionamento. Nesse período de teste e validação, o cliente teve a oportunidade de explorar as funcionalidades do *software*, assim teve como proporcionar *feedback* valioso sobre sua experiência. Esse período de avaliação, embora breve, permitiu uma compreensão eficaz das percepções do cliente em relação ao *software*. Com base nisso, foram identificadas áreas de aprimoramento e houver solicitações específicas para melhorias que serão implementadas na próxima atualização do *software*.

4.6 Ferramentas de implementação

As ferramentas de implementação é a fase em qual transforma-se os requisitos de *software* em programa executável. Nessa fase ocorre a aplicação de conceitos da ferramenta de desenvolvimento e em seguida tem-se o produto final desenvolvido. As ferramentas utilizadas para o desenvolvimento do *software* foram:

4.6.1 Mysql

O *Mysql* é um dos principais gerenciadores de banco de dados, sua linguagem utilizada é a SQL (Structured Query Language), onde os dados são organizados em tabelas com linhas e colunas.

segundo Vinícius carvalho (CARVALHO, 2015, p.3), Banco de dados é:

"Apesar de banco de dados ser um termo técnico, a maioria das pessoas nos dias de hoje tem contato direto com ele. De fato, grande parte da população atualmente tem acesso a equipamentos, cuja função (principal ou secundária) é o armazenamento de informações."

Ainda segundo Vinícius carvalho (CARVALHO, 2015, p.5), linguagem SQL é:

"SQL significa *Structured Query Language* e é a linguagem padrão utilizada pelos banco de dados relacionais. Os principais motivos disso resultam de sua simplicidade e facilidade de uso."

A escolha de utilizar o *Mysql* como sistema de gerenciamento de banco de dados no sistema se justifica pelo fato de sua ampla compatibilidade com diferentes linguagens de programação e *frameworks* e também tive a oportunidade de trabalhar e me familiarizar com o *Mysql* em diversos projetos.

4.6.2 Vaadin

A ferramenta Vaadin é um *framework* para desenvolvimento web de código aberto baseado em Java. Além disso, permite que o desenvolvedor possa desenvolver uma interface mais rica e elegante.

Segundo Marlon, Vaadin é:

"Apesar de parecer novato no cenário Java, o Vaadin é um velho conhecido e existe desde o ano 2000. Contudo, nesta época, seu nome era IT Mill Toolkit, produzido pela empresa IT Mill. Passados

nove anos desde sua concepção, a ferramenta foi rebatizada e passou a se chamar Vaadin. Mas não foi apenas o nome que mudou. Ocorreram mudanças internas, sendo a troca do motor de exibição de componentes a mais importante: sai um mecanismo próprio e entra o GWT."(MARLON, 2011)

Ainda segundo Marlon, a função da ferramenta é:

"Esta ferramenta permite focar-se no principal, que é a construção da aplicação. Os detalhes de comunicação cliente/servidor são praticamente abstraídos, sobrando raríssimas situações em que serão necessárias intervenções desta natureza."(MARLON, 2011)

A escolha de utilizar o Vaadin no desenvolvimento do sistema AquaHaven se justifica pelo fato de que o Vaadin é um *framework* web que me permite desenvolver interfaces ricas e responsivas de forma eficiente. Além disso, o Vaadin possui compatibilidade com Java, que será a linguagem de programação escolhida para o sistema AquaHaven.

4.6.3 VSCode

O VSCode (Visual Studio Code) é uma IDE (Integrated Development Environment) de desenvolvimento completo que combina um editor de código-fonte, um compilador ou interpretador, ferramentas de depuração, um navegador de arquivos, e outras ferramentas para facilitar o desenvolvimento do sistema.

Segundo Edson, o VSCode é:

"Em 2015 foi lançado pela Microsoft um editor de código destinado ao desenvolvimento de aplicações web chamado de Visual Studio Code, ou simplesmente VSCode. Anunciada durante o Build, evento voltado a desenvolvedores que ocorre nos Estados Unidos anualmente, trata-se de uma ferramenta leve e multiplataforma que está disponível tanto para Windows, quanto para Mac OS e Linux e atende a uma gama enorme de projetos, não apenas ASP.NET, como também Node.js. Adicionalmente, o editor possui suporte à sintaxe de diversas linguagens como Python, Ruby, C++."(EDSON, 2016)

A escolha de utilizar o VSCode como IDE no sistema se justifica pelo fato de possui uma interface de usuário limpa e intuitiva. Além disso, o VSCode é conhecido por sua inicialização rápida e baixo consumo de recursos do sistema. Isso permite que os desenvolvimentos do sistema ocorram de forma mais eficiente, sem atrasos ou interrupções causados por longos tempos de carregamento ou travamentos. Por esses motivos a escolha do VSCode para o sistema, proporcionou uma experiência de desenvolvimento fluida e eficiente.

4.6.4 Java

O Java é uma linguagem de programação orientada a objetos e amplamente utilizada em domínios, como desenvolvimento web, aplicativos móveis, entre outros. Desenvolvida pela Sun Microsystems (agora pertencente à Oracle), a linguagem Java foi lançada em 1995 e desde então se tornou uma das linguagens mais populares do mundo.

Segundo Herbert, o Java é:

"Java foi concebida por James Gosling . Patrick Naughton , Chris Warth , Ed Frank e Mike Sheridan na Sun Microsystems , em 1991. No início , a linguagem se chamava "Oak ", mas foi renomeada como "Java "em 1995. Surpreendentemente , a motivação original para a criação de Java não foi a Internet ! A principal motivação foi a necessidade de uma linguagem independente de plataforma que pudesse ser usada na criação de software embutido em vários dispositivos eletrônicos domésticos , como torradeiras , fornos de micro - ondas e controles remotos . Como era de se esperar , muitos tipos de CPUs são usados como controladores . O problema era que (na época) a maioria das linguagens era projetada para ser compilada para um destino específico . Por exemplo , considere C ++."(SCHILDT, 2015)

A escolha do Java para o sistema se justifica pela vasta comunidade de desenvolvedores e recursos disponíveis. Além disso o Java oferece bibliotecas, *frameworks* e ferramentas que otimizam o desenvolvimento. Além do mais, o Java oferece recursos como multiplataforma, apoio a orientação a objetos e recursos de mapeamento objeto-relacional. Por esses motivos, essa foi a escolha do Java como linguagem principal no desenvolvimento do sistema AquaHaven.

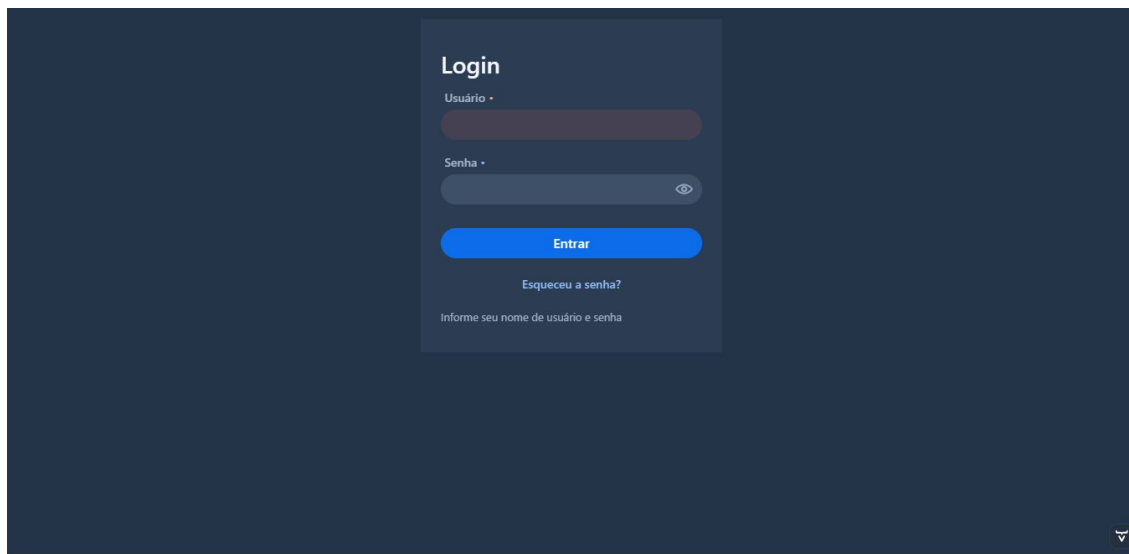
5 Sistema AquaHaven em funcionamento

Após a explicação sobre a fundamentação teórica no Capítulo 3, apresentar os conceitos e a aplicação da metodologia no Capítulo 4, neste capítulo são mostradas as telas do sistema AquaHaven.

5.1 Telas do Sistema AquaHaven

A Figura 18, representa a tela de login do sistema AquaHaven, onde mostra o campo de "usuário" e "senha", além de haver o "entrar" e "esqueceu a senha".

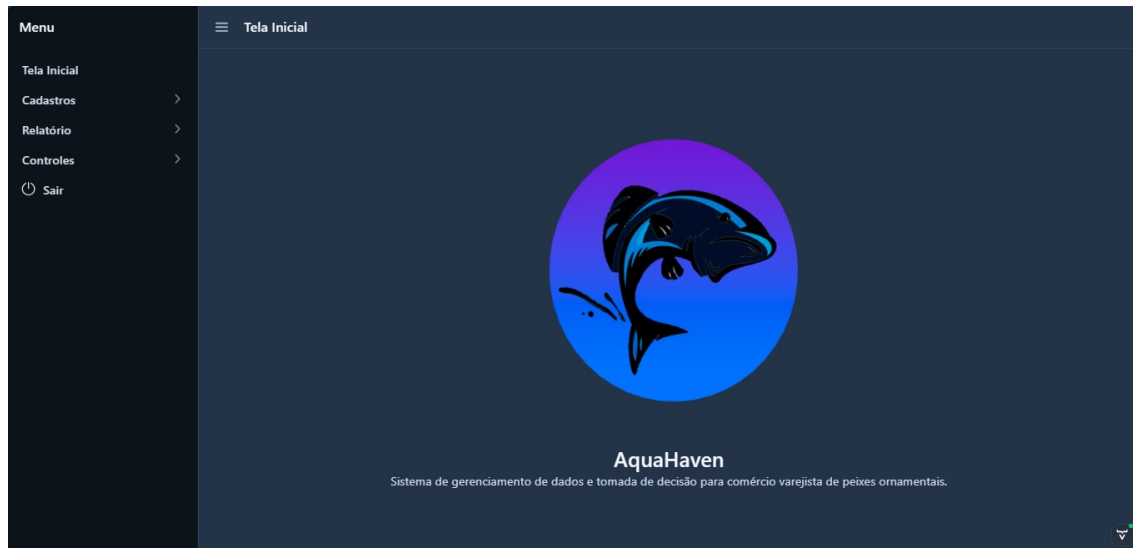
Figura 18 – Tela de Login



Fonte: Autoria Própria

A Figura 19, representa a tela inicial do sistema AquaHaven, onde mostra o nome do sistema e sua descrição, no menu lateral apresenta as opções disponíveis que o usuário irá ter acesso, às funções são de cadastros, relatório e controle.

Figura 19 – Tela Inicial



Fonte: Autoria Própria

A Figura 20, mostra a tela de cadastro da AquaHaven, em que é possível cadastrar o produto, contendo três botões sendo eles de cadastro, excluir e novo, o botão de cadastro permite o usuário cadastrar um novo Produto ou ao selecioná-lo ele pode alterar as informações. O segundo permite a excluir os dados de cadastro. Por fim o terceiro a limpa os dados. E ainda é permitido a visualização dos cadastros feitos.

Figura 20 – Tela de Cadastro de Produto

Id	Nome	Valor Unitário	Descrição	Tipo Produto	Estoque
1	Betta pote	10.0	Ração	Alimento	100

Fonte: Autoria Própria

A Figura 21, mostra a tela de cadastro da AquaHaven, onde é possível cadastrar o Serviço, contendo três botões sendo eles de cadastro, excluir e novo, o botão de cadastro permite o usuário a cadastrar um novo Serviço ou ao selecioná-lo ele pode alterar as informações. O segundo permite a excluir os dados de cadastro. Por fim terceiro a limpa os dados. Na tela ainda é permitido a visualização dos cadastros feitos.

Figura 21 – Tela de Cadastro de Serviço

ID	Data	Valor Unitario	Descricao	Quantidade	Status
1	2023-10-10	60.0	Limpeza de aquario	1	true
2	2023-10-04	60.0	limpeza de aquario	1	false

Fonte: Autoria Própria

A Figura 22, mostra a tela de cadastro da AquaHaven, em que é possível cadastrar o Tipo de Produto, contendo três botões sendo eles de cadastro, excluir e novo, o botão de cadastro permite o usuário a cadastrar um novo Tipo de Produto ou ao selecioná-lo ele pode alterar as informações. O segundo permite a excluir os dados de cadastro. Por fim terceiro a limpa os dados. Na tela ainda é permitido a visualização dos cadastros feitos.

Figura 22 – Tela de Cadastro de Tipo de Produto

ID	Nome
1	Alimento
2	Animal

Fonte: Autoria Própria

A Figura 23, mostra a tela de cadastro da AquaHaven, onde é possível cadastrar o cliente, contendo três botões sendo eles de cadastro, excluir e novo, o botão de cadastro permite o usuário a cadastrar um novo cliente ou ao selecioná-lo ele pode alterar as informações. O segundo permite a excluir os dados de cadastro. Por fim terceiro a limpa os dados. Na tela ainda é permitido a visualização dos cadastros feitos.

Figura 23 – Tela de Cadastro de Cliente

Nome	Cpf	Rg	Telefone	Data Nasc...	Endereco	Genero	Categoria
Arley	05644823654	6245473	(64) 99325...	1980-05-15	Rua 1	Masculino	Pessoa Física

Fonte: Autoria Própria

A Figura 24, mostra a tela de cadastro da AquaHaven, onde é possível ca-

dastrar o Fornecedor, contendo três botões sendo eles de cadastro, excluir e novo, o botão de cadastro permite o usuário a cadastrar um novo Fornecedor ou ao selecioná-lo ele pode alterar as informações. O segundo permite a excluir os dados de cadastro. Por fim terceiro a limpa os dados. Na tela ainda e permitido a visualização dos cadastros feitos.

Figura 24 – Tela de Cadastro de Fornecedor

Nome	Endereco	Telefone	Cnpj	Razao Social	Data Primeira Co...
Felipe	Rua 145	66985143625	24693258000140	Andrade	2011-11-17

Fonte: Autoria Própria

A Figura 25, mostra a tela de cadastro da AquaHaven, onde é possível cadastrar o Funcionário, contendo três botões sendo eles de cadastro, excluir e novo, o botão de cadastro permite o usuário a cadastrar um novo Funcionário ou ao selecioná-lo ele pode alterar as informações. O segundo permite a excluir os dados de cadastro. Por fim terceiro a limpa os dados. Na tela ainda e permitido a visualização dos cadastros feitos.

Figura 25 – Tela de Cadastro de Funcionário

Nome	Cpf	Rg	Telefone	Data Nasc...	Endereco	Genero	Categoria
João	05644823654	6245473	(64) 99325-...	1980-05-15	Rua 1	Masculino	funcionário

Fonte: Autoria Própria

A Figura 26, mostra a tela de cadastro da AquaHaven, onde é possível cadastrar a venda, contendo três botões sendo eles de cadastro, excluir e novo, o botão de cadastro permite o usuário a cadastrar uma nova venda ou ao selecioná-lo ele pode alterar as informações. O segundo permite a excluir os dados de cadastro. Por fim terceiro a limpa os dados da tela. Na tela ainda é permitida a visualização dos cadastros feitos.

Figura 26 – Tela de Cadastro de Venda

Id	Data	Cliente	Funcionario	Pagamento
1	2023-11-16	Arley	Pedro	Cartão Débito

Fonte: Autoria Própria

A Figura 27, mostra a tela de cadastro da AquaHaven, onde é possível cadas-

trar a Compra, contendo três botões sendo eles de cadastro, excluir e novo, o botão de cadastro permite o usuário a cadastrar uma nova compra ou ao selecioná-lo ele pode alterar as informações. O segundo permite a excluir os dados de cadastro. Por fim terceiro a limpa os dados da tela. Na tela ainda e permitido a visualização dos cadastros feitos.

Figura 27 – Tela de Cadastro de Compra

Id	Data	Fornecedor	Funcionario	Pagamento
1	2023-11-16	Felipe	Pedro	Dinheiro

Fonte: Autoria Própria

A Figura 28, mostra a tela de registro de usuário, onde após o cadastro do funcionário é necessário cadastrar o login e senha para acessar o sistema é também realizar as alterações do login e senha, na tela é apresentado os campos nome de usuário que deverá ser preenchido o login de acesso ao sistema, campo senha que deverá ser preenchido a nova senha, também é mostrado os botões cadastrar que após ser preenchido o campo será salvo o login e senha do usuário e o botão excluir caso resolva excluir algum cadastro já salvo e o botão novo onde limpa os campos.

Figura 28 – Tela de Cadastro de Registro

Menu

Tela Inicial

Cadastros

- Produto
- Serviço
- Tipo Produto
- Funcionario
- Cliente
- Fornecedor
- Venda
- Compra
- Registro**

Relatório

Sair

Registro

Nome de Usuário

Senha

Cadastrar Excluir Novo

Username	Password
Rafael	\$2a\$10\$UXVBz.G07vGt/VCamXpa6ef529cTmestnm3VFbyVEaw...

Fonte: Autoria Própria

A Figura 29, mostra a tela de relatório, onde é possível ver todos dados de produtos, que foram cadastrados no sistema.

Figura 29 – Tela de Relatório de Produtos

Nome	Quantidade	Valor Unitario	Descricao	Tipo
Betta pote	1	16.0	ração	consumível
betta	1	10.0	azul	peixe

Fonte: Autoria Própria

A Figura 30, mostra a tela de relatório de Vendas, onde é possível ver todos dados cadastrados na venda. Na tela possui o filtro onde é possível filtrar os resultados do relatório.

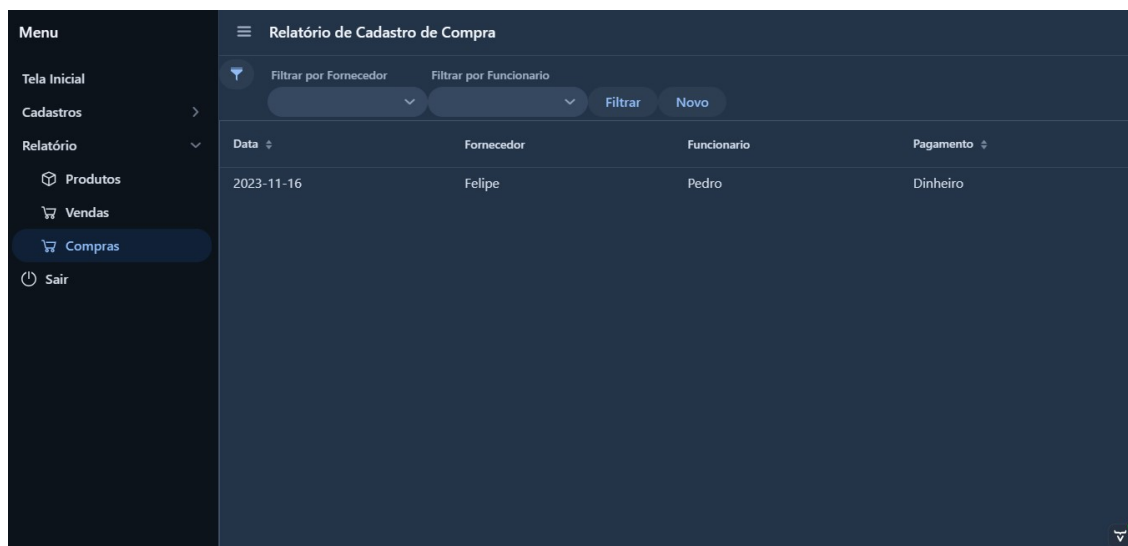
Figura 30 – Tela de Relatório de Vendas

Data	Cliente	Funcionario	Pagamento
2023-11-16	Arley	Pedro	Cartão Débito

Fonte: Autoria Própria

A Figura 31, mostra a tela de relatório de Compras, onde é possível ver todos dados cadastrados na compra. Na tela possui o filtro onde é possível filtrar os resultados do relatório.

Figura 31 – Tela de Relatório de Compras



Fonte: Autoria Própria

O capítulo 5 deste trabalho apresentou as telas do sistema AquaHaven, com essas telas e possível fornecer uma ilustração do seu design. Essas telas ilustraram a *interface* intuitiva do sistema e suas funcionalidades implementadas.

6 Conclusão

Diante o crescimento do comércio de animais vivos, e dos desafios enfrentados na gestão de dados desse domínio de negócio, este trabalho apresentou uma solução moderna e eficiente para apoiar a tomada de decisão estratégica no comércio de animais vivos, sobretudo em empresas que lidam com peixes ornamentais. Após um extenso levantamento bibliográfico, Foi identificado que, atualmente, são usados métodos tradicionais de controle, como fichas, planilhas. Desse modo, essa forma de controle de dados apresenta limitações, tais como perda de dados, registros errôneos, e outros tipos de lançamentos manuais.

Por meio do desenvolvimento do AquaHaven, foi possível solucionar os problemas identificados, e proporcionar melhor controle de entrada e saída de produtos, gerenciamento financeiro, controle de estoque. Além disso, o sistema foi capaz de lidar com as particularidades desse setor, tais como cadastro de diferentes tipos de produtos, permitir o registro de vendas de diferentes tipos de peixes ornamentais, entre outros.

Com a conclusão do desenvolvimento do sistema e a validação conduzida pelo cliente, foi possível validar a hipótese inicial. O *software* demonstrou efetivamente aprimorar a gestão da empresa, com melhorias em dados precisos e no suporte a tomadas de decisões estratégicas. Esses resultados confirmam o valor agregado pelo *software* ao ambiente empresarial.

Portanto, é de suma importância o desenvolvimento de um *software* específico, sobretudo para segmentos de mercado que tem requisitos muito específicos, que é o caso do comércio de peixes ornamentais, além de diversos outros. Com este *software*, torna-se possível proporcionar benefícios notáveis, como um controle mais eficaz, registros precisos e uma base sólida para a tomada de decisões estratégicas das empresas.

6.1 Trabalhos futuros

Os trabalhos futuros elencados para o AquaHaven são:

- Desenvolver versões do sistema compatíveis com dispositivos móveis, como smartphones e tablets;
- Realizar uma área de acesso para o cliente;

- Funcionalidades de suporte ao cliente;
- Relatórios impressos;
- Controle de mortalidade dos animais (peixes).

Referências

APARECIDA, Z. *Possíveis riscos relativos à venda de animais vivos no Mercado Central foram debatidos*. 2021. Acessado em [04/10/2022]. Disponível em: <<https://cmbhweb.cmbh.mg.gov.br/comunica%C3%A7%C3%A3o/not%C3%ADcias/2021/08/poss%C3%ADveis-riscos-relativos-%C3%A0-venda-de-animais-vivos-no-mercado-central#:~:text=Ela%20explicou%20que%20o%20Decreto%205616%2F1987%20especifica%20que,com%20a%20documenta%C3%A7%C3%A3o%20necess%C3%A1ria%2C%20a%20venda%20%C3%A9%20legal>>. Citado na página 15.

CARVALHO, V. *MySQL Comece com o principal banco de dados open souce do mercado*. Rua Vergueiro, 3185 - 8º andar, 04101-300 – Vila Mariana – São Paulo – SP – Brasil: Casa do Código, 2015. © Casa do Código. Todos os direitos reservados e protegidos pela Lei nº9.610, de 10/02/1998. Nenhuma parte deste livro poderá ser reproduzida, nem transmitida, sem autorização prévia por escrito da editora, sejam quais forem os meios: fotográficos, eletrônicos, mecânicos, gravação ou quaisquer outros. Citado na página 56.

EDSON. *Introdução ao Visual Studio Code*. 2016. Acessado em [25/05/2023]. Disponível em: <<https://www.devmedia.com.br/introducao-ao-visual-studio-code/34418>>. Citado na página 57.

GUEDES, G. T. A. *OML 2: uma abordagem prática*. 2. ed. São Paulo: Novatec Editora, 2011. Bibliografia. ISBN 978-85-7522-281-2. Citado 5 vezes nas páginas 38, 39, 40, 42 e 43.

IBAMA, M. D. M. A. *Coleta na natureza*. 2018. Acessado em [20/10/2022]. Disponível em: <<https://www.ibama.gov.br/conteudo-do-menu-superior/28-menu-superior-perguntas-frequentes/600-peixes-ornamentais>>. Citado 2 vezes nas páginas 16 e 19.

LARMAN, C. *Utilizando UML e Padrões*. Bookman, 2007. Acessado em [27/04/2023]. ISBN 9788560031528. Disponível em: <<https://books.google.com.br/books?id=ZHtcynS03DIC>>. Citado na página 33.

LIMA, P. A. D. *Software para gestão de pet shop e clínica veterinária*. UNIVERSIDADE FEDERAL DO PARANÁ, p. 64, 2015. Citado 2 vezes nas páginas 23 e 26.

LOPES, J.; TRINDADE, M.; FUJIMOTO, R. *Expertfish: Software para diagnóstico de doenças de peixes*. CONGRESSO BRASILEIRO DE INTELIGÊNCIA COMPUTACIONAL. Fortaleza, Ceará: SBIC, p. 6, 2016. Citado 2 vezes nas páginas 24 e 26.

LOPES, M. A. e. a. *Custo leite para windows: software de controle de custos para a pecuária leiteira*. *Revista Brasileira de Zootecnia*, v. 29, n. 5, p. 1504–1510, 2000. Citado 2 vezes nas páginas 24 e 26.

MARLON. *Vaadin: Programação para a web com jeito de desktop* - Revista Java Magazine 95 - Parte 1. 2011. Acessado em [25/05/2023]. Disponível em: <<https://www.devmedia.com.br/vaadin-programacao-para-a-web-com-jeito-de-desktop-revista-java-magazine-95-parte-1/22979>>. Citado na página 57.

MATOS, R. *Peixe Plati/Platy: Guia com cuidados, fatos e características da espécie*. 2022. Acessado em [20/04/2023]. Disponível em: <<https://peixeseaquarismo.com/peixe-plati-platy-guia-com-cuidados-fatos-e-caracteristicas-da-especie/>>. Citado 2 vezes nas páginas 16 e 17.

PRESSMAN, R. S. *Software engineering: a practitioner's approach*. 7th. [S.l.]: McGraw-Hill Higher Education, 2010. Citado 5 vezes nas páginas 28, 30, 33, 34 e 35.

RECHI, E. *Oscar, Apaiari (Astronotus ocellatus)*. 2014. Acessado em [20/04/2023]. Disponível em: <<http://www.aquarismopaulista.com/astronotus-ocellatus/>>. Citado na página 19.

RIBEIRO, F. d. A. S. et al. Comércio brasileiro de peixes ornamentais. *Panorama da Aquicultura*, v. 18, n. 110, p. 54–59, 2008. Citado na página 19.

RUBIN, K. S. *Essential Scrum: a practical guide to the most popular agile process*. Upper Saddle River, NJ: Addison-Wesley, 2012. ISBN 978-0-13-704329-3. Citado 3 vezes nas páginas 36, 37 e 38.

RULIYANDI. *PEIXE-AMADOR*. 2019. Acessado em [18/04/2023]. Disponível em: <<https://www.fishhobbyist.net/2019/04/types-of-platys-disease-and-how-to.html>>. Citado na página 16.

RULIYANDI. *PEIXE-AMADOR*. 2020. Acessado em [18/04/2023]. Disponível em: <<https://www.fishhobbyist.net/2020/03/how-to-treat-sick-betta-fish.html>>. Citado na página 17.

RULIYANDI. *PEIXE-AMADOR*. 2020. Acessado em [18/04/2023]. Disponível em: <<https://www.fishhobbyist.net/2020/08/steps-for-spawning-oscar-fish.html>>. Citado na página 18.

SANTOS, E. *Peixe-beta*. 2020. Acessado em [20/04/2023]. Disponível em: <[https://www.suapesquisa.com/mundoanimal/peixe_beta.htm#:~:text=-%20%C3%89%20uma%20esp%C3%A9cie%20de%20peixe%20bastante%20resistente,labirintos%20que%20usam%20para%20respirar%20o%20ar%20atmosf%C3%A9rico](https://www.suapesquisa.com/mundoanimal/peixe_beta.htm#:~:text=-%20%C3%89%20uma%20esp%C3%A9cie%20de%20peixe%20bastante%20resistente,labirintos%20que%20usam%20para%20respirar%20o%20ar%20atmosf%C3%A9rico.)>. Citado na página 18.

SCHILDT, H. *Java para iniciantes*. [S.l.]: Bookman Editora, 2015. Citado na página 58.

SEBRAE, S. *A expansão e a força do mercado pet*. 2023. Acessado em [25/03/2023]. Disponível em: <<https://www.sebrae.com.br/sites/PortalSebrae/sebraeaz/o-mercado-pet,6897934a41fc0710VgnVCM1000004c00210aRCRD>>. Citado na página 15.

SOMMERVILLE, I. *Engenharia de software. 9th.* [S.l.]: São Paulo: Pearson Prentice Hall, 2011. Citado 8 vezes nas páginas 27, 28, 31, 32, 33, 34, 35 e 37.

SUTHERLAND, J. *Scrum: a arte de fazer o dobro do trabalho na metade do tempo.* São Paulo: LeYa, 2014. Título original: Scrum: The art of doing twice the work in half the time. ISBN 9788544100882. Citado na página 37.