

UNIVERSIDADE PAULISTA  
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

CARLOS EDUARDO DE OLIVEIRA  
MARCUS VINICIUS NUNES  
PAULO SERGIO BRESSAN JUNIOR  
RAFAEL ANTONIO LOPES ALVES  
SIDNEY DE SOUZA JUNIOR

**PROJETO INTEGRADO MULTIDISCIPLINAR**  
PROJETO E DESENVOLVIMENTO DE UM SISTEMA PARA GERENCIAMENTO  
DE UMA LOJA DE ARTIGOS ESPORTIVOS

RIBEIRÃO PRETO

2018

CARLOS EDUARDO DE OLIVEIRA  
MARCUS VINICIUS NUNES  
PAULO SERGIO BRESSAN JUNIOR  
RAFAEL ANTONIO LOPES ALVES  
SIDNEY DE SOUZA JUNIOR

### **PROJETO INTEGRADO MULTIDISCIPLINAR**

Projeto e desenvolvimento de um sistema para gerenciamento de uma loja de artigos esportivos.

Trabalho de Integração Multidisciplinar  
para a conclusão do terceiro semestre do  
curso de Análise e Desenvolvimento de  
Sistemas apresentado à Universidade  
Paulista.

RIBEIRÃO PRETO

2018

CARLOS EDUARDO DE OLIVEIRA  
MARCUS VINICIUS NUNES  
PAULO SERGIO BRESSAN JUNIOR  
RAFAEL ANTONIO LOPES ALVES  
SIDNEY DE SOUZA JUNIOR

### **PROJETO INTEGRADO MULTIDISCIPLINAR**

Projeto e desenvolvimento de um sistema para gerenciamento de uma loja de artigos esportivos.

Trabalho de Integração Multidisciplinar  
para a conclusão do terceiro semestre do  
curso de Análise e Desenvolvimento de  
Sistemas apresentado à Universidade  
Paulista.

Orientador:

Aprovado em: \_\_\_\_/\_\_\_\_/\_\_\_\_

### **BANCADA EXAMINADORA**

\_\_\_\_\_/\_\_\_\_/\_\_\_\_  
Marcelo Gomes

\_\_\_\_\_/\_\_\_\_/\_\_\_\_  
Alynne

\_\_\_\_\_/\_\_\_\_/\_\_\_\_  
Danielli

## Agradecimento

Agradecemos em primeiro lugar a Deus por ser à base das nossas conquistas. Aos professores, pela dedicação em suas orientações prestadas na elaboração deste trabalho, nos incentivando e colaborando no desenvolvimento de nossas ideias. Aos amigos e familiares pelo apoio e colaboração no trabalho, nos incentivando com dicas preciosas.

## Dedicatória

Dedicamos este trabalho aos nossos familiares e amigos, que compreenderam os momentos de ausência necessários para o desenvolvimento deste trabalho. Muito obrigado a todos pela compreensão e paciência.

## RESUMO

A empresa Intergalact Sports, uma loja de artigos esportivos quer informatizar seu negócio. Atualmente o controle de estoque e vendas é realizado através de anotações em cadernos e este não tem se mostrado eficiente. Para resolver esse problema a empresa procura uma fábrica de software para o desenvolvimento de um novo sistema para controlar suas atividades. O sistema deve contemplar um controle de cadastro de usuários e clientes, assim como registrar as vendas efetuadas e gerir o estoque dos produtos cadastrados. O sistema deve ser acessível para eventuais usuários com necessidades especiais e ser acessível através de dispositivos móveis, sem que haja perdas de desempenho e usabilidade.

**Palavras chave:** Esportivo, Software, ERP.

## **ABSTRACT**

The company Intergalact Sports, a sporting goods store wants to computerize your business. Currently, inventory and sales control is carried out through notebooks and it has not been efficient. To solve this problem the company looks for a software factory to develop a new system to control its activities. The system should include a control of the registration of users and customers, as well as record the sales made and manage the inventory of the products registered. The system must be accessible to potential users with special needs and be accessible through mobile devices without loss of performance and usability.

**Keywords:** Sports, Software, ERP.

## LISTA DE FIGURAS

Figura 1 - Elementos básicos do diagrama de caso de uso .....	20
Figura 2 - Relacionamentos entre classes .....	21
Figura 3 - Exemplo de diagrama de atividades. ....	22
Figura 4 - Exemplo de diagrama de atividades do Trello. ....	28
Figura 5 – Interface Git.....	29
Figura 6 - Diagrama de caso de uso AFSport. ....	36
Figura 7 - Diagrama de classes AFSport. ....	45
Figura 8 - Diagrama de atividade para cadastrar um produto. ....	46
Figura 9 - Diagrama de atividade para cadastrar um cliente. ....	47
Figura 10 - Diagrama de atividade para realizar uma venda.....	48
Figura 11 - Tela de Login. ....	49
Figura 12 - Tela de boas vindas. ....	49
Figura 13 - Tela de menu principal do sistema.....	50
Figura 14 - Tela de pedidos cadastrados. ....	50
Figura 15 - Tela de cadastro de pedido.....	51
Figura 16 - Tela de clientes cadastrados. ....	51
Figura 17 - Tela de cadastro de cliente. ....	52
Figura 18 - Tela de categorias cadastradas. ....	53
Figura 19 - Tela de cadastro de nova categoria. ....	53
Figura 20 - Tela de produtos cadastrados.....	54
Figura 21 - Tela de cadastro de produto. ....	55
Figura 22 - Tela de preços cadastrados.....	56
Figura 23 - Tela de cadastro de preço. ....	56
Figura 24 - Tela de estoques cadastrados. ....	57
Figura 25 - Tela de adicionar estoque.....	57
Figura 26 - Tela de movimentações de caixa.....	58
Figura 27 - Tela de abertura de caixa. ....	58
Figura 28 - Tela de operações cadastradas. ....	59
Figura 29 - Tela de cadastro de operações.....	59
Figura 30 - Tela de movimentações cadastradas.....	60
Figura 31 - Tela de cadastro de movimentação. ....	60
Figura 32 - Tela de estados cadastrados. ....	61

Figura 33 - Tela de cadastro de estado.....61

Figura 34 - Tela de cidades cadastradas. ....62

Figura 35 - Tela de cadastro de cidade.....62

Figura 36 - Tela de usuários cadastrados. ....63

Figura 37 - Tela de cadastro de usuário.....64



## **LISTA DE GRÁFICOS**

Gráfico 1 - Experiência na Utilização do sistema. ....	30
Gráfico 2 - Satisfação com a interface do sistema. ....	31
Gráfico 3 - Recomendaria o sistema AFSport. ....	31

## LISTA DE QUADROS

Quadro 1 - Funcionalidades Systronic Office Comercial. ....	23
Quadro 2 - Funcionalidades Tecnobyte SAC Free.....	24
Quadro 3 - funcionalidades Nex. ....	25
Quadro 4 - Comparativo das funcionalidades. ....	26
Quadro 5 - [RF01] Manter de categoria.....	31
Quadro 6 - [RF02] Manter usuário.....	32
Quadro 7 - [RF03] Gerar relatório. ....	32
Quadro 8 - [RF04] Manter de pedidos. ....	32
Quadro 9 - [RF05] Manter cliente. ....	33
Quadro 10 - [RF06] Manter produto ....	33
Quadro 11 - [RF07] Vender produto.....	33
Quadro 12 - [RF07] Controle de caixa.....	34
Quadro 13 – Caso de uso: Manter um produto. ....	37
Quadro 14 – Caso de uso: Manter uma categoria.....	38
Quadro 15 – Caso de uso: Manter usuário. ....	39
Quadro 16 – Caso de uso: Manter pedidos de produtos. ....	40
Quadro 17 – Caso de uso: Gerar relatório. ....	41
Quadro 18 – Caso de uso: Manter clientes. ....	42
Quadro 19 – Caso de uso: Vender produto.....	43
Quadro 20 – Caso de uso: Controlar caixa. ....	44

## **LISTA DE ABREVIATURAS E SIGLAS**

ASOO – Analise de Sistemas Orientado a Objeto

IHC – Interação Humano Computador

POO – Programação Orientada a Objeto

SQL – Structured Query Language

C# – C-Sharp (Linguagem de Programação)

BD – Banco de Dados

SGBD – Sistema Gerenciador de Banco de Dados

UML – Unified Modeling Language

PDV – Ponto de Venda

SPC – Serviço de Proteção ao Crédito

NF-e – Nota Fiscal Eletrônica

RF – Requisito Funcional

RNF – Requisito não Funcional

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>13</b>
1.1. Objetivo.....	15
<b>2. BASE CONCEITUAL .....</b>	<b>16</b>
2.1. Programação orientada a objetos. ....	16
2.2. Interação humano-computador (IHC).....	17
2.3. Engenharia de Software .....	17
2.5. Banco de Dados .....	18
2.6. UML .....	19
2.7. Trello .....	22
2.8. Ferramenta de controle de versão (Git) .....	22
<b>3. COMPARAÇÃO DE SISTEMAS.....</b>	<b>23</b>
3.1. Software: Systronic Office Comercial .....	23
3.2. Software: Tecnobyte SAC Free.....	24
3.3. Software: Nex - módulo administrador. ....	25
3.4. Comparativo das funcionalidades entre os softwares. ....	26
<b>4. DESENVOLVIMENTO .....</b>	<b>27</b>
4.1. Organização de tarefas (Trello).....	27
4.2. Ferramenta de controle de versão (Git) .....	29
4.3. Linguagem de programação utilizada.....	29
4.4. Interação humano-computador (IHC).....	29
4.4.1. Design do software .....	30
4.4.2. Paleta de cores do sistema .....	30
4.4.3. Avaliação realizada no Workshop .....	30
4.5. Requisitos funcionais .....	31
4.5.1. [RF01] Cadastro de categoria.....	31
4.5.2. [RF02] Manter usuário .....	32
4.5.3. [RF03] Gerar relatório .....	32
4.5.4. [RF04] Manter pedidos de produto .....	32
4.5.5. [RF05] Manter cliente .....	33
4.5.6. [RF06] Manter produto .....	33
4.5.7. [RF07] Vender produto .....	33

4.5.7. [RF07] Controle de caixa .....	34
4.6. Requisitos não funcionais.....	34
4.6.1. [RNF001] Usabilidade .....	34
4.6.2. [RNF002] Manutenibilidade .....	34
4.6.3. [RNF003] Confiabilidade .....	34
4.6.4. [RNF004] Desempenho .....	35
4.6.5. [RNF005] Portabilidade .....	35
4.6.6. [RNF006] Segurança .....	35
4.7. Diagrama de caso de uso .....	36
4.8. Documentação caso de uso .....	36
4.8.1. Manter produto.....	37
4.8.2. Manter categoria.....	38
4.8.3. Manter usuário.....	39
4.8.4. Manter pedidos de produtos. ....	40
4.8.5. Gerar relatório. ....	41
4.8.6. Manter Clientes.....	42
4.8.7. Vender produto.....	43
4.8.8. Controlar caixa. ....	44
4.9. Diagrama de classe.....	44
4.10. Diagrama Atividades.....	46
4.10.1. Cadastrar produto .....	46
4.10.2. Cadastro de cliente. ....	47
4.10.3. Venda .....	48
4.11. Software AFSport.....	48
4.12. Repositório Git dos arquivos .....	64
5. CONCLUSÃO .....	65
5.1. Projetos futuros .....	65
6. REFERÊNCIAS BIBLIOGRÁFICAS .....	66

## 1. INTRODUÇÃO

Segundo LAUDON (2007), a implantação de um sistema de informação em uma empresa pode significar uma vantagem competitiva através do controle de custos, traçando perfil dos clientes e fornecedores. Um sistema bem projetado se torna confiável e permite facilidade de acesso as informações, trazendo agilidade no levantamento de relatórios de vendas e estoque dos produtos.

Com base nas informações levantadas, uma loja de artigos esportivos procurou uma fábrica de software, para que fosse desenvolvido um software visando controlar suas atividades de venda e controle de estoque. O sistema conta também com um módulo de cadastro de clientes e filtros para geração de relatórios. O software possui uma interface intuitiva e de fácil acesso para eventuais usuários com necessidades especiais, além de ser possível seu acesso através de dispositivos móveis.

O sistema teve seu início com a realização do levantamento de requisitos funcionais e não funcionais de acordo com o modelo de negócio da empresa solicitante. Foram criados os diagramas de caso de uso com o intuito de facilitar o entendimento do que seria implementado no sistema, dando uma visão geral para a aprovação do projeto por parte da solicitante. Após a aceitação foram gerados os diagramas de classe, sequência e o Modelo Entidade Relacionamento do banco de dados para nortear o desenvolvimento do sistema.

O sistema teve como base em seu desenvolvimento uma linguagem de programação orientada a objetos (POO) chamada C# (lê-se C-Sharp), onde os objetos do mundo real são modelados para o mundo computacional. O C# é uma linguagem poderosa e simples que herdou muitos recursos do C++ e Microsoft Visual Basic, tornando-a uma linguagem mais limpa e lógica (SHARP, 2014).

Em todo sistema onde se deseja persistência de dados, é necessário o uso de um banco de dados. Na década de 1970 surgiram os bancos de dados relacionais que utilizam a linguagem SQL (*Structured Query Language*) como padrão. Bancos de dados relacionais possuem sua estrutura baseada em registros organizados e relacionados em tabelas, onde as linhas são os registros e as colunas descrevem as características dos registros (FERRARI, 2007).

Para auxiliar o desenvolvimento do sistema pelos desenvolvedores, foi utilizado o sistema de controle de versão chamado Git. Este sistema de versão

permite controlar de forma segura as mudanças no código e desfazê-las caso seja necessário sem a necessidade de cópias manuais (PORTAL DEVMEDIA, 2018).

Com a implantação do sistema, espera-se maior agilidade no atendimento dos clientes e controle do fluxo de caixa, assim como estoque de produtos. Estas melhorias podem ser decisivas na fidelização de clientes que buscam realizar suas compras de maneira rápida.

### **1.1. Objetivo**

Este trabalho visa o desenvolvimento de um software de computador, onde o mesmo fará todo o controle de vendas, pedidos de compra, estoque de produtos, cadastro de clientes e geração de relatórios pertinentes a toda movimentação realizada pelo sistema. Assim como o controle de acesso ao sistema por usuários cadastrados, com o intuito de proteger as informações nele armazenadas e manter a rastreabilidade das operações realizadas através do usuário que as realizou.



## **2. BASE CONCEITUAL**

Nesse tópico será abordado uma breve explicação sobre as ferramentas e metodologia que serão utilizadas no desenvolvimento do projeto AFSport. Cada ferramenta utilizada é essencial para o desenvolvimento do projeto e a metodologia abordada tem o objetivo de organizar a equipe e o desenvolvimento do projeto.

### **2.1. Programação orientada a objetos.**

A programação orientada a objetos ao contrário do que a maioria dos profissionais pensa, ela surgiu em 1967 junto com a linguagem Simula. Algum tempo depois nos anos 70 a XEROX criou o SmallTalk, considerada a primeira linguagem puramente orientada a objetos (PORTAL WIKILIVROS, 2018).

Segundo JOYANES AGUILAR (2011), na programação orientada a objetos há alguns conceitos, que juntos moldam suas características e a torna tão atrativa. Entre eles podemos citar:

- **Abstração:**

É como um objeto do mundo real é moldado para o mundo computacional, onde são levantadas suas características relevantes para cada tipo de aplicação.

- **Classe:**

Uma classe descreve a estrutura interna dos objetos que pertencem a ela, é o objeto representado pela abstração, nela estão definidos os atributos e as funcionalidades de cada objeto.

- **Objeto:**

É a instancia de uma classe. Um tipo abstrato de dados, onde seus atributos e métodos são definidos na classe a qual ele pertence.

- **Encapsulamento:**

É utilizado para impedir o acesso direto a estrutura de dados interna de uma classe pelo usuário dela. É a combinação dos dados e os métodos que podem ser executados, garantindo assim o acesso aos dados somente por meio dos próprios métodos do objeto.

- **Herança:**

O conceito de herança pode ser relacionado com a genética humana, sendo assim, um objeto pode transmitir suas propriedades para outros objetos, permitindo reutilizar objetos já definidos anteriormente reduzindo código redundante em

programas. Ter suporte a herança, é um pré-requisito para que uma linguagem seja orientada a objetos.

- Polimorfismo:

É um conceito de grande importância na programação orientada a objetos. Objetos diferentes podem responder de maneira diferente quando ligados através da herança, como por exemplo, classes derivadas de superclasses podem ter métodos parecidos, porém, o mesmo terá comportamento diferente para cada classe derivada.

## **2.2. Interação humano-computador (IHC)**

A interação humano computador trata da forma que as pessoas utilizam uma interface, sendo ela um display de um micro-ondas ou uma tela de software em um computador. Ela se preocupa desde o design, avaliação e a implementação de interfaces onde há interação com o ser humano, buscando a construção de uma interface intuitiva e amigável ao usuário e também promovendo facilidade na aprendizagem e maior produtividade (BARBOSA, 2011).

## **2.3. Engenharia de Software**

O termo surgiu pela primeira vez em 1968 após a crise do software, período em que o desenvolvimento era problemático e defeituoso, resultando em um produto de difícil manutenção em que não seguia padrões.

Segundo Friedrich Ludwig Bauer: “Engenharia de Software é a criação e a utilização de sólidos princípios de engenharia a fim de obter software de maneira econômica, que seja confiável e trabalhe em máquinas reais”.

Engenharia de Software é a área da computação específica para a concepção de projeto de software, documentação, desenvolvimento, manutenção e qualidade utilizando de práticas gerenciais a fim de definir e cumprir objetivos compreendidos e alinhados com toda a equipe envolvida.

Contextualização dos principais aspectos:

Análise de Requisitos. Através da análise de requisito é o momento onde se efetua o conhecimento do problema para desenvolver o software (JALOTE, 2005).

Design do Software. Pelo design do software é o momento que o engenheiro de software realiza o planejamento da solução do problema que foi levantado no documento de requisitos (JALOTE, 2005).

Codificação. A codificação é o momento que pega o problema resolvido no design do software e o transformara em uma linguagem de programação (JALOTE, 2005).

Teste. O teste é uma maneira de medir o controle da qualidade do software durante o desenvolvimento de software (JALOTE, 2005).

## **2.4. Análise de Sistemas Orientados a Objetos**

A Análise Orientada a Objetos surgiu com o intuito de minimizar os problemas de desenvolvimento, em relação ao que é proposto e acordado com o cliente que requer o software. Uma de suas principais características é a divisão de tarefas e responsabilidades, onde deixa claro o que é de competência dos desenvolvedores, o que é uma função que o software deverá ter e como será a execução dos processos. O seu foco é no mapeamento de uma solução sistêmica para algum processo de negócio.

## **2.5. Banco de Dados**

De acordo com MEDEIROS (2007 apud DATE, 2000) “um banco de dados é uma coleção de dados persistentes utilizadas pelos sistemas de aplicação de uma empresa”. Um arquivo de banco de dados é composto por tabelas que através de seus atributos gravam registros que descrevem entidades do mundo real, cujas informações podem ser manipuladas e recuperadas mais tarde por algum sistema de informação.

MEDEIROS (2007), afirma que “é necessário que um BD tenha uma representação eficiente que possibilite acesso a informações corretas, em tempo hábil”. Para um BD ser eficiente, alguns princípios listados abaixo devem ser observados:

- Redundância: Não deve haver duplicidade das informações dentro de um BD, como por exemplo, tabelas distintas contendo campos com informações idênticas.
- Inconsistência: Deve-se ter cuidado quanto ao inserir um dado em um campo respeitando seu tamanho e tipo por exemplo. A redundância pode gerar inconsistência, pois uma mesma informação gravada em tabelas diferentes pode ter a informação atualizada em somente um local.

- Integração: Estabelecer procedimentos para níveis de acesso para os diferentes setores da empresa, visto que atualizações podem comprometer a integridade das informações do BD, evitando que as atualizações realizadas por um setor não comprometam o outro.

Para garantir esses princípios utiliza-se um sistema gerenciador de banco de dados (SGBD), que é um software responsável por administrar o banco de dados e garantir a segurança e a integridade dos dados. São exemplos de SGBD, Oracle, MySQL, Microsoft SQL Server e PostgreSQL. Esses SGBD têm em comum, a linguagem de consulta estruturada SQL.

## 2.6. UML

Com o crescimento na utilização das linguagens orientadas a objetos e a complexidade dos novos projetos, surgiu à necessidade de se criar metodologias que possibilitassem projetar e planejar a construção dos sistemas. Foram criadas metodologias, porém nenhuma atendia totalmente a necessidade dos construtores (FOWLER, KENDAL, 2000).

Surge então em meados dos anos 90 a *Unified Modeling Language* (UML), criada por Grady Booch, Ivar Jacobson e James Rumbaugh, três estudiosos e criadores de métodos orientados a objetos. A UML não é uma metodologia, mais sim uma linguagem de modelagem, pois ela não define quais procedimentos precisam ser seguidos (FURLAN, 1998).

Ainda segundo FOWLER, KENDAL (2000), a UML baseia-se em uma notação principalmente gráfica, utilizada para expressar projetos.

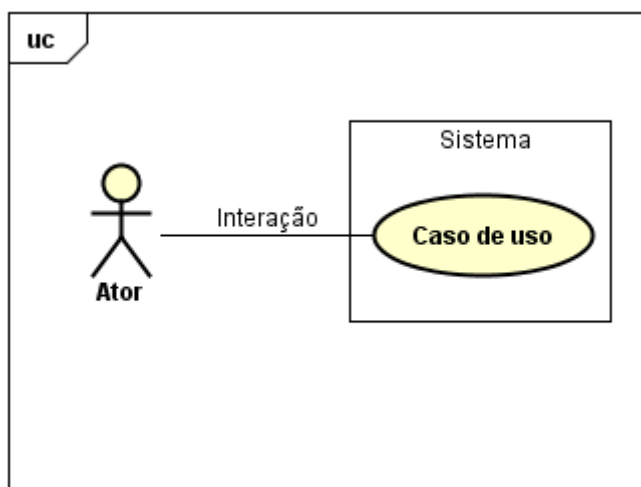
A UML é constituída de vários diagramas, que expressam informações sobre alguns elementos de um modelo. Cada diagrama ilustra o que ocorre em cada parte de um sistema, dando uma visão diferente ao programador (FURLAN, 1998).

Fazem parte da UML os seguintes diagramas: Caso de uso, classes, atividade, interação, estado e implementação.

- Diagrama de caso de uso:

Segundo FOWLER, KENDAL (2000), o caso de uso representa como é a interação do usuário com o sistema, e apesar de ser amplamente utilizado, não é fundamental para a realização de projetos. Ele tem como objetivo descrever quais são os requisitos funcionais para um sistema, tendo como elementos básicos o ator, caso de uso, interação e sistema.

Figura 1 - Elementos básicos do diagrama de caso de uso



Fonte: o autor.

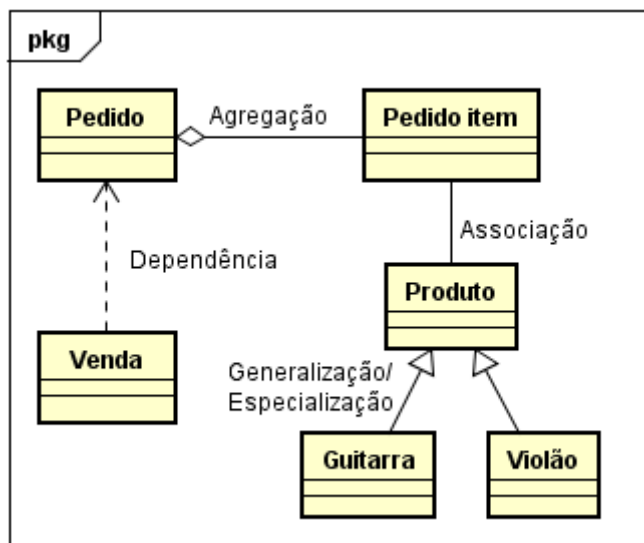
O ator interage com o sistema, podendo ser um usuário, outro sistema, dentre outros. O caso de uso descreve a ação do ator no sistema, já a interação trata do relacionamento entre atores e os casos de uso (FOWLER, KENDAL, 2000).

- Diagrama de classes:

Presente em quase todas as metodologias de análise orientadas a objetos, ele especifica as classes e os relacionamentos entre elas. Representada por um retângulo, uma classe pode ser dividida em até três partes onde na primeira parte está o nome da classe, na segunda parte seus atributos com seus respectivos tipos e visibilidade, e por último seus métodos. Uma classe possui quatro tipos básicos de relacionamentos (FURLAN, 1998). São eles:

1. Agregação: indica que uma classe pertence à outra;
2. Associação: ligação entre classes;
3. Generalização/especificação: Relacionamento de herança entre classes;
4. Dependência: indica que uma classe depende de outra.

Figura 2 - Relacionamentos entre classes

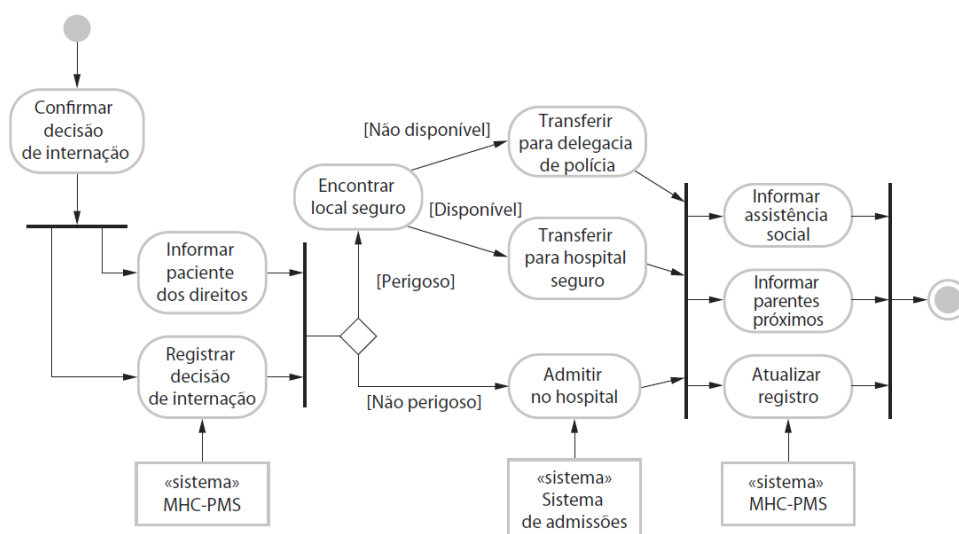


Fonte: o autor.

- Diagrama de atividade:

Segundo SOMMERVILLE (2011), o diagrama de atividades descreve as atividades de um processo ou processamento de dados. Iniciado por um círculo preenchido, possui setas indicando o fluxo do processo onde as atividades são representadas por retângulos de cantos arredondados. Pode haver processos paralelos que são iniciados por uma barra sólida, que também servem para indicar o final de processos paralelos, onde todas as atividades paralelas devem ter sido completadas. Para indicar o final de uma atividade, utiliza-se um círculo preenchido dentro de outro círculo.

Figura 3 - Exemplo de diagrama de atividades.



Fonte: SOMMERVILLE (2011).

## 2.7. Trello

O Trello é um site que pode ser acessado utilizando os navegadores Google Chrome, Mozilla Firefox, Safari e Internet Explorer, além de também possuir aplicativos na Google Play e iTunes. É uma ferramenta de gerenciamento de projetos ou tarefas em forma de listas, ele é muito versátil e pode ser moldado de acordo com as necessidades de cada usuário. É possível criar quadros e definir quem pode ter acesso a eles, ver as atualizações e colaborar com os mesmos. Assim toda a equipe pode acompanhar de perto tudo o que se passa com o projeto.

## 2.8. Ferramenta de controle de versão (Git)

“É um sistema de controle de versão, de código aberto, usado no gerenciamento de código-fonte”, (WIKILIVROS, 2018). Criado por Linus Torvalds, possui arquitetura descentralizada, todo repositório do Git possui um histórico e seguidor de revisões. O usuário faz uma cópia da pasta do repositório central, e depois pode publicar as mudanças para os usuários desse servidor (WIKILIVROS, 2018). O acesso pode ser realizado usando linhas de comando ou por alguns softwares disponíveis para conexão com o GIT, como por exemplo o GitHub.

### 3. COMPARAÇÃO DE SISTEMAS

Realizamos um comparativo com soluções disponíveis no mercado para automação comercial, onde dentre as soluções encontradas listamos três a seguir:

#### 3.1. Software: Systronic Office Comercial

No quadro abaixo estão listadas as principais funcionalidades do software Systronic Office Comercial.

Quadro 1 - Funcionalidades Systronic Office Comercial.

Funcionalidade	Descrição
Manter clientes	Cadastro, alteração, exclusão e pesquisa de clientes.
Manter fornecedores	Cadastro, alteração, exclusão e pesquisa de fornecedores.
Manter produtos	Cadastro, alteração, exclusão e pesquisa de produtos.
Manter vendedores	Cadastro, alteração, exclusão e pesquisa de vendedores.
Manter transportadoras	Cadastro, alteração, exclusão e pesquisa de transportadoras.
Manter movimentações	Vendas (PDV), fechamento de caixa (PDV) e entrada de mercadorias, contas á receber e contas á pagar.
Administrativo	Movimentação bancária.
Gerar relatórios	Relatórios de compra, venda, títulos á receber, títulos á pagar e fluxo bancário
Gerar gráficos	Financeiro, vendas, faturamento, projeção de fluxo financeiro, lucratividade, custos fixos e curva ABC produtos.
Utilitários	Parâmetros do sistema, formato de documentos, organizar telas, editor de texto, calendário, calculadora, configurações regionais, Formatar cartas (Cobrança, Aviso SPC, aniversário, promocional), controle de usuário, informações diárias.
Manter usuários	Cadastro, alteração, exclusão e pesquisa de usuários.

Fonte: O autor.



### 3.2. Software: Tecnobyte SAC Free.

Quadro 2 - Funcionalidades Tecnobyte SAC Free.

Funcionalidade	Descrição
Manter clientes	Cadastro, alteração, exclusão e pesquisa de clientes.
Manter fornecedores	Cadastro, alteração, exclusão e pesquisa de fornecedores.
Manter produtos	Cadastro, alteração, exclusão e pesquisa de produtos.
Manter vendedores	Cadastro, alteração, exclusão e pesquisa de vendedores.
Manter transportadoras	Cadastro, alteração, exclusão e pesquisa de transportadoras.
Manter cartões de crédito	Cadastro, alteração, exclusão e pesquisa de administradoras de cartões.
Manter movimentações	Vendas (PDV), fechamento de caixa (PDV), compras, contas á receber e contas á pagar, orçamentos, perdas, consumo interno, doações, emissor NF-e.
Manter conta bancária	Cadastro, alteração, exclusão e movimentação bancária.
Manter agenda telefônica	Cadastro, alteração, exclusão e pesquisa de telefones.
Gerar relatórios	Relatórios de compra, venda, títulos á receber, títulos á pagar, fluxo bancário, preços.
Ferramentas	Configurações locais, globais e banco de dados, consulta SQL, emissão de recibo, promissória e duplicatas, lembretes, calculadora, abrir gaveta.
Manter usuários	Cadastro, alteração, exclusão e pesquisa de usuários.

Fonte: O autor.

### 3.3. Software: Nex - módulo administrador.

Quadro 3 - funcionalidades Nex.

Funcionalidade	Descrição
Manter clientes	Cadastro, alteração, exclusão e pesquisa de clientes.
Manter fornecedores	Cadastro, alteração, exclusão e pesquisa de fornecedores.
Manter produtos	Cadastro, alteração, exclusão e pesquisa de produtos.
Manter vendedores	Cadastro, alteração, exclusão e pesquisa de vendedores.
Manter transportadoras	Cadastro, alteração, exclusão e pesquisa de transportadoras.
Manter entregadores	Cadastro, alteração, exclusão e pesquisa de entregadores.
Manter movimentações	Vendas (PDV), fechamento de caixa (PDV), compras, contas á receber e contas á pagar, orçamentos, pedidos, troca ou devolução.
Gerar relatórios	Caixa: meios de pagamento e movimentação de produtos; Clientes: ranking de compras; Vendas: Comissão por vendedor, horário de pico, categoria e produto, produto, marca, cliente e categoria.
Manter usuários	Cadastro, alteração, exclusão e pesquisa de usuários.

Fonte: O autor.

### 3.4. Comparativo das funcionalidades entre os softwares.

No quadro abaixo são mostradas as funcionalidades encontradas nos três softwares pesquisados e o nosso software AFSport.

Quadro 4 - Comparativo das funcionalidades.

Funcionalidade	Systronic Office Comercial	Tecnobyte SAC Free	Nex	AFSport
Manter clientes	Sim	Sim	Sim	Sim
Manter fornecedores	Sim	Sim	Sim	Não
Manter produtos	Sim	Sim	Sim	Sim
Manter vendedores	Sim	Sim	Sim	Sim
Manter transportadoras	Sim	Sim	Sim	Não
Manter entregadores	Não	Não	Sim	Não
Manter cartões de crédito	Não	Sim	Não	Não
Manter movimentações	Sim	Sim	Sim	Sim
Manter conta bancária	Não	Sim	Não	Não
Manter agenda telefônica	não	Sim	Não	Não
Movimentação Bancária	Sim	Sim	Não	Não
Gerar relatórios	Sim	Sim	Sim	Não
Gerar gráficos	Sim	Não	Não	Não
Utilitários (configuração)	Sim	Sim	Não	Não
Manter usuários	Sim	Sim	Sim	Sim

Fonte: O autor.

## **4. DESENVOLVIMENTO**

Neste capítulo são abordados os requisitos funcionais e não funcionais, diagrama de caso de uso, diagrama de atividades, e diagrama de classes. Nesse capítulo também constam as telas do software, junto com a especificação das funcionalidades e as tecnologias e metodologias usadas na criação do software AFSport.

### **4.1. Organização de tarefas (Trello)**

No planejamento deste projeto foi definido quatro listas para a organização das tarefas que são: a fazer, em andamento, concluído e base. Dentro dessas listas são adicionados os “cards”, assim chamados as tarefas, que descreve tudo o que precisa ser feito, o prazo para a entrega e quem está responsável por essa atividade.

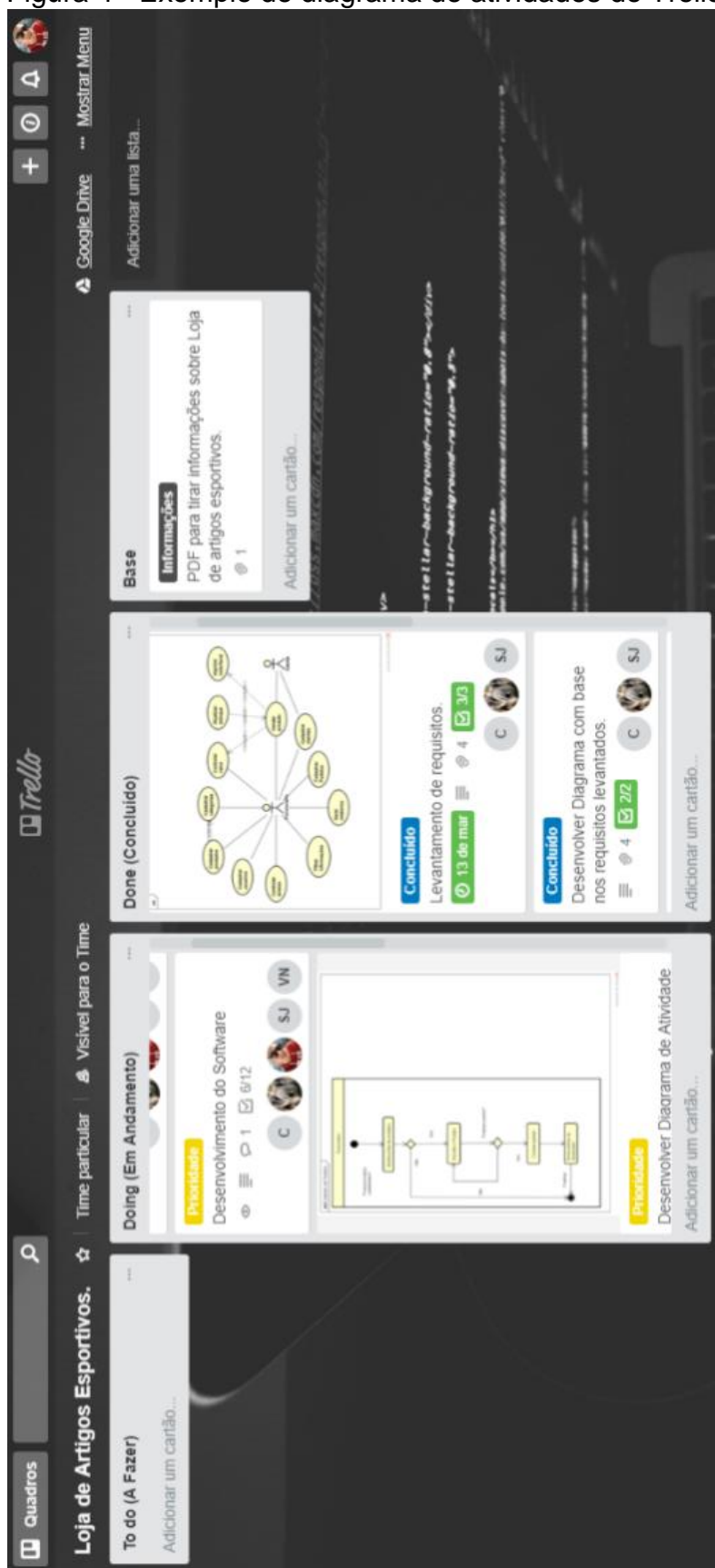
Na primeira lista, “a fazer”, ficam todas as tarefas planejadas para o projeto e que não foram iniciadas.

Na segunda lista, “em andamento”, ficam as tarefas que estão sendo executadas pelos membros da equipe.

Na terceira lista, “concluído”, ficam todas as tarefas finalizadas e certificadas pelos membros da equipe.

Na quarta lista, “base”, foi utilizada para armazenar informações uteis que poderiam ajudar no desenvolvimento do projeto.

Figura 4 - Exemplo de diagrama de atividades do Trello.

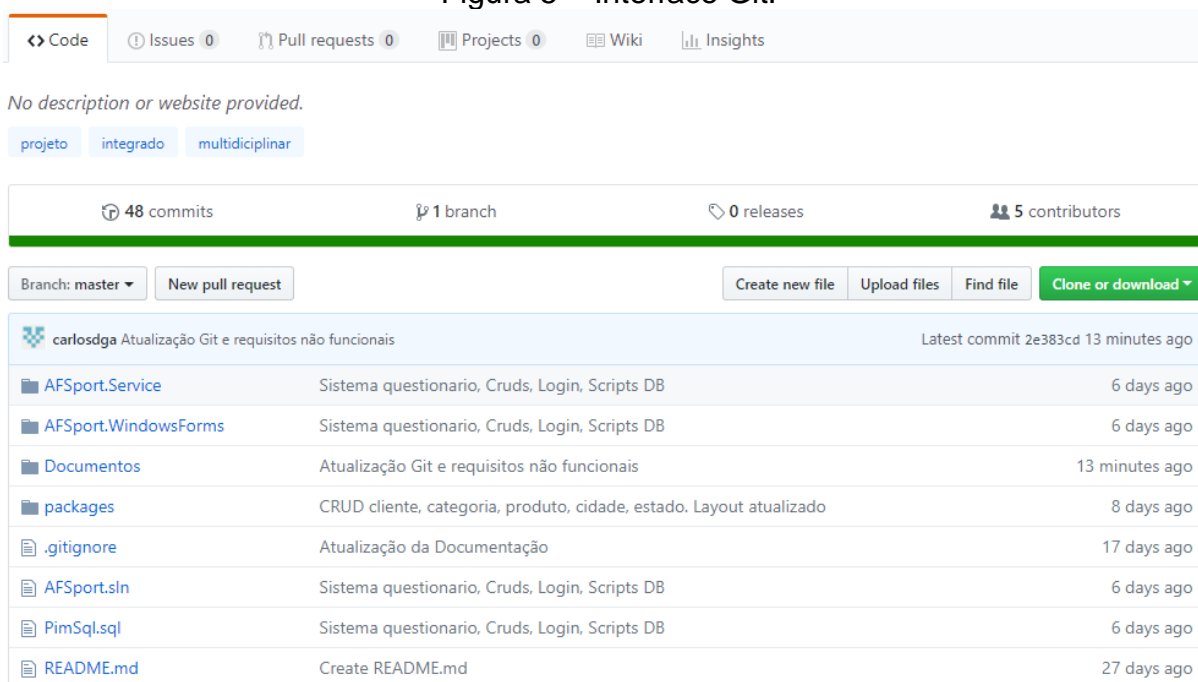


Fonte: O autor

## 4.2. Ferramenta de controle de versão (Git)

Para facilitar a participação de todos os integrantes do grupo no desenvolvimento, criamos um repositório no Git para que todos pudessem trabalhar em cima da mesma base do código do sistema, otimizando tempo e evitando retrabalhos por conta de duas ou mais pessoas estarem realizando a mesma modificação.

Figura 5 – Interface Git.



Fonte: O autor.

## 4.3. Linguagem de programação utilizada

O sistema foi desenvolvido utilizando a linguagem de programação orientada a objetos C-Sharp (C#), utilizando o framework Visual Studio 2017.

## 4.4. Interação humano-computador (IHC)

O sistema AFSport foi desenvolvido com o intuito de maximizar sua utilização de forma simples e intuitiva, tendo como base os seguintes itens:

#### 4.4.1. Design do software

O Software utiliza o Design minimalista para compor suas telas, botões e ícones. Tudo isso para poder entregar uma interface simples e amigável, tendo a experiência de uso como o seu aspecto central e fazendo da usabilidade um ponto forte.

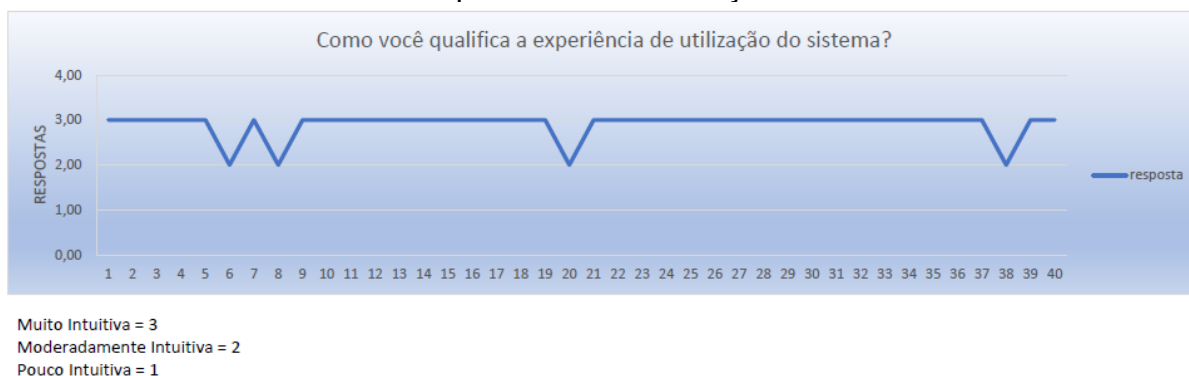
#### 4.4.2. Paleta de cores do sistema

A cor azul presente em todas as telas da interface tem como intuito, transmitir confiança e maturidade, além de proporcionar um ambiente acolhedor e confortável na experiência do usuário.

#### 4.4.3. Avaliação realizada no Workshop

No dia 21 de abril de 2018, realizamos um workshop nas dependências da Universidade Paulista, onde a interface e funcionalidades do sistema foram apresentas aos visitantes. Através de um formulário realizamos uma pesquisa com os visitantes afim de que eles avaliassem o sistema e dessem um feedback. No total foram 38 participantes, onde os resultados obtidos estão expressos nos gráficos a seguir:

Gráfico 1 - Experiência na Utilização do sistema.



Fonte: O autor.

Gráfico 2 - Satisfação com a interface do sistema.



Fonte: O autor.

Gráfico 3 - Recomendaria o sistema AFSport.



Fonte: O autor.

## 4.5. Requisitos funcionais

Neste capítulo estão listados os requisitos funcionais do sistema AFSport.

### 4.5.1. [RF01] Cadastro de categoria

O sistema permite cadastrar, alterar e excluir/inativar uma categoria.

Quadro 5 - [RF01] Manter de categoria.

Parâmetros de Entrada	Nome, Descrição, ativa ou inativa.
Descrição	Dados levantados para cadastro de uma categoria
Restrições	Valida os dados e cadastra, altera ou exclui uma categoria.
Dados de Saída	No caso do sistema encontrar algum erro, uma mensagem será exibida.

Fonte: O autor.



#### 4.5.2. [RF02] Manter usuário

O sistema permite cadastrar, alterar e excluir/inativar um usuário.

Quadro 6 - [RF02] Manter usuário.

Parâmetros de Entrada	Login e senha
Descrição	Dados levantados para cadastro de usuário.
Restrições	Valida os dados e cadastra, altera ou exclui um usuário.
Dados de Saída	No caso do sistema encontrar algum erro, uma mensagem será exibida.

Fonte: O autor.

#### 4.5.3. [RF03] Gerar relatório

O sistema permite gerar relatórios a partir das informações inseridas.

Quadro 7 - [RF03] Gerar relatório.

Parâmetros de Entrada	Data inicial e data final.
Descrição	Dados levantados para criação de relatórios.
Restrições	Valida os dados e busca as informações no banco de dados.
Dados de Saída	Mostra um relatório com as informações solicitadas

Fonte: O autor.

#### 4.5.4. [RF04] Manter pedidos de produto

O sistema permite cadastrar, alterar e excluir um pedido.

Quadro 8 - [RF04] Manter de pedidos.

Parâmetros de Entrada	Produto e quantidade.
Descrição	Dados levantados para cadastro de operações.
Restrições	Valida os dados e cadastra, altera ou exclui um pedido.
Dados de Saída	No caso do sistema encontrar algum erro, uma mensagem será exibida.

Fonte: O autor.

#### 4.5.5. [RF05] Manter cliente

O sistema permite cadastrar, alterar e excluir/inativar um cliente.

Quadro 9 - [RF05] Manter cliente.

Parâmetros de Entrada	Nome, e-mail, rua, número, bairro, cidade e estado.
Descrição	Dados levantados para cadastro de cliente.
Restrições	Valida os dados e cadastra, altera ou exclui um cliente.
Dados de Saída	No caso do sistema encontrar algum erro, uma mensagem será exibida.

Fonte: O autor.

#### 4.5.6. [RF06] Manter produto

O sistema permite cadastrar, alterar e excluir/inativar um produto.

Quadro 10 - [RF06] Manter produto

Parâmetros de Entrada	Nome, descrição, valor de compra, valor de venda e categoria.
Descrição	Dados levantados para cadastro de produto.
Restrições	Valida os dados e cadastra, altera ou exclui um produto.
Dados de Saída	No caso do sistema encontrar algum erro, uma mensagem será exibida.

Fonte: O autor.

#### 4.5.7. [RF07] Vender produto

O sistema permite cadastrar, alterar e excluir uma venda.

Quadro 11 - [RF07] Vender produto.

Parâmetros de Entrada	Selecionar cliente, selecionar produtos e escolher forma de pagamento.
Descrição	Dados levantados para realizar uma venda.
Restrições	Valida os dados e cadastra, altera ou exclui uma venda.
Dados de Saída	No caso do sistema encontrar algum erro, uma mensagem será exibida.

Fonte: O autor.

#### 4.5.7. [RF07] Controle de caixa

O sistema permite gerenciar o fluxo de caixa.

Quadro 12 - [RF07] Controle de caixa.

Parâmetros de Entrada	Selecionar a data do caixa, escolher uma opção do caixa para processar e exibir as informações solicitadas.
Descrição	Dados levantados para realizar o controle de fluxo do caixa.
Restrições	Valida os dados e exibe as informações solicitadas.
Dados de Saída	No caso do sistema encontrar algum erro, uma mensagem será exibida.

Fonte: O autor.

#### 4.6. Requisitos não funcionais

Neste capítulo estão listados os requisitos não funcionais do sistema AFSport.

##### 4.6.1. [RNF001] Usabilidade

O sistema foi desenvolvido de forma que seja de fácil aprendizagem, com um esforço mínimo para aprender e focado na facilidade de utilização com uma interface agradável para a realização das tarefas. Ou seja, em pouco tempo e com esforço mínimo o usuário já irá alcançar um bom desempenho no uso do sistema.

##### 4.6.2. [RNF002] Manutenibilidade

O sistema AFSport foi desenvolvido utilizando a linguagem de programação C-Sharp, que por ser uma linguagem orientada a objetos, permite a modularização do sistema. Sendo assim, erros encontrados em sua utilização, podem ser facilmente solucionados. Caso também seja necessária alguma alteração para inclusão de funcionalidades, por exemplo, a organização do código fonte permite fazê-la com facilidade sem que grandes alterações sejam necessárias.

##### 4.6.3. [RNF003] Confiabilidade

O software AFSport foi testado para garantir que todos os requisitos implementados fossem cumpridos sem ocorrência de erros e mantendo o desempenho esperado diante de cada operação executada.

#### **4.6.4. [RNF004] Desempenho**

O desempenho de um sistema é um fator muito importante, pois ele vai afetar dentre outras coisas a usabilidade do sistema prejudicando o trabalho do usuário e ainda trazer insatisfação por parte dos clientes ao realizarem uma compra. O sistema AFSport foi desenvolvido com a preocupação que seja rápido e que não sobrecarregue os recursos do computador onde está sendo executado, utilizando práticas de programação e programas adequados para este fim.

#### **4.6.5. [RNF005] Portabilidade**

Atualmente a versão disponível do software foi desenvolvida para funcionar nas versões atuais para desktop do sistema Windows®, porém, futuramente poderá receber também uma interface web e para telemóveis. O desenvolvimento do software foi realizado já pensando nessas tendências de mercado permitindo a reusabilidade do código já implementado.

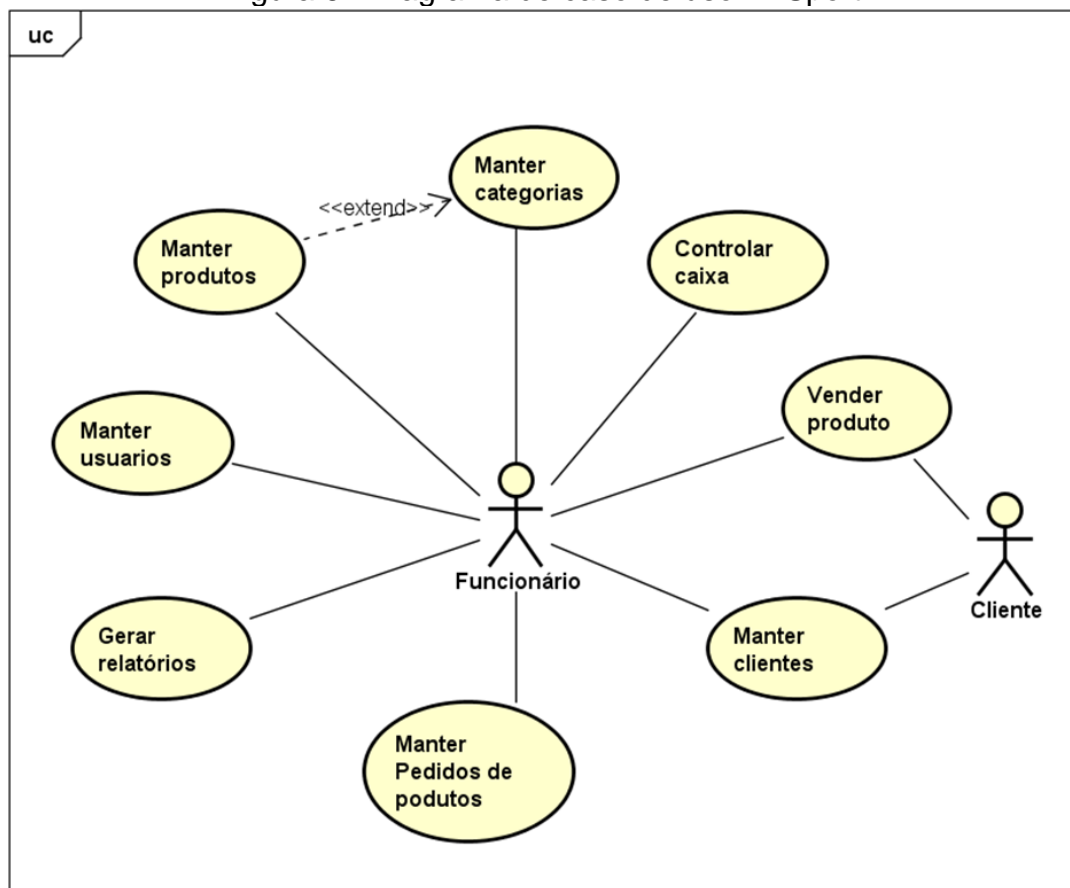
#### **4.6.6. [RNF006] Segurança**

O software AFSport dispõe de um cadastro de usuários, onde o software só pode ser utilizado mediante a confirmação dos dados de login e senha do usuário quando o software é executado. Ou seja, dessa forma a integridade das informações esta protegida de ataques intencionais ou acidentais. O cadastro de usuários deve ser controlado por um administrador, o qual também será orientado quando a importância da realização de cópias de segurança frequentes do sistema.

#### 4.7. Diagrama de caso de uso

A figura a seguir representa o diagrama de caso de uso do sistema AFSport.

Figura 6 - Diagrama de caso de uso AFSport.



Fonte: O autor

#### 4.8. Documentação caso de uso

Nesse tópico abordamos uma descrição dos casos de uso, onde são identificados as ações e os atores que interagem junto ao sistema para concluir o requisito.

#### 4.8.1. Manter produto.

O quadro a seguir representa as etapas e o relacionamento ator/sistema para manter um produto.

Quadro 13 – Caso de uso: Manter um produto.

Quadro 10 – Caso de uso: Manter um produto.		
Nome do caso de uso	Manter Produto	
Caso de uso Geral		
Ator Principal	Funcionário	
Ator(res) Secundário(s)		
Resumo	Este caso de uso descreve as etapas percorridas pelo funcionário para cadastrar, alterar e excluir um produto.	
Pré condições	Existir uma Categoria Cadastrada	
Pós condições	Produto cadastrado	
Fluxo Principal		
Ações do Ator		Ações do Sistema
1. Funcionário clica no botão do formulário de cadastro de produto		
		2. Apresenta as opções cadastrar, alterar e excluir
3. Funcionário preenche os dados		
4. Funcionário clica em enviar formulário		
		5. Verifica se os dados estão corretos
		6. Sistema salva os dados no banco de dados
		7. Mostra a tela de sucesso/efetivação

Fonte: O autor.

#### 4.8.2. Manter categoria.

O quadro a seguir representa as etapas e o relacionamento ator/sistema para cadastrar uma categoria.

Quadro 14 – Caso de uso: Manter uma categoria.

Quadro 1.1 - Caso de uso: Manter uma categoria		
Nome do caso de uso	Manter Categoria	
Caso de uso Geral		
Ator Principal	Funcionário	
Ator(res) secundário(s)		
Resumo	Este caso de uso descreve as etapas percorridas pelo funcionário para cadastrar, alterar e excluir uma categoria	
Pré condições		
Pós condições	Categoria cadastrada	
Fluxo Principal		
Ações do Ator		Ações do Sistema
1. Funcionário clica no botão do formulário de cadastro de categoria		
		2. Apresenta a tela de cadastro
3. Funcionário preenche os dados		
4. Funcionário clica em enviar formulário		
		5. Sistema salva os dados no banco de dados
		6. Mostra a tela de sucesso/efetivação

Fonte: O autor.

#### 4.8.3. Manter usuário.

O quadro a seguir representa as etapas e o relacionamento ator/sistema para cadastrar um usuário.

Quadro 15 – Caso de uso: Manter usuário.

Nome do caso de uso	Manter Usuários	
Caso de uso Geral		
Ator Principal	Funcionário	
Ator(res) Secundário(s)		
Resumo	Este caso de uso descreve as etapas percorridas pelo funcionário para cadastrar, alterar e excluir um usuário	
Pré condições		
Pós condições	Usuário cadastrado	
Fluxo Principal		
Ações do Ator		Ações do Sistema
1. Funcionário clica no botão do formulário de cadastro de usuário		
		2. Apresenta a tela de cadastro
3. Funcionário preenche os dados		
		4. Verifica se os dados estão corretos
5. Funcionário clica em enviar formulário		
		6. Sistema salva os dados no banco de dados
		7. Mostra a tela de sucesso/efetivação

Fonte: O autor.



#### 4.8.4. Manter pedidos de produtos.

O quadro a seguir representa as etapas e o relacionamento ator/sistema para cadastrar um pedido.

Quadro 16 – Caso de uso: Manter pedidos de produtos.

Quadro 10 - Caso de uso: Manter pedidos de produtos.	
Nome do caso de uso	Manter Pedidos de Produtos
Caso de uso Geral	
Ator Principal	Funcionário
Ator(res) Secundário(s)	
Resumo	Este caso de uso descreve as etapas percorridas pelo funcionário para cadastrar, alterar e excluir os pedidos de um produto
Pré condições	
Pós condições	Pedido de produtos cadastrados
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Funcionário clica no botão do formulário de cadastro de pedido	
	2. Apresenta a tela de cadastro
3. Funcionário preenche os dados	
	4. Verifica se os dados estão corretos
5. Funcionário clica em enviar formulário	
	6. Sistema salva os dados no banco de dados
	7. Mostra a tela de sucesso/efetivação

Fonte: O autor.

#### 4.8.5. Gerar relatório.

O quadro a seguir representa as etapas e o relacionamento ator/sistema para gerar um relatório.

Quadro 17 – Caso de uso: Gerar relatório.

Quadro 17 – Caso de uso: Gerar Relatórios.		
Nome do caso de uso	Gerar relatórios	
Caso de uso Geral		
Ator Principal	Funcionário	
Ator(res) Secundário(s)		
Resumo	Este caso de uso descreve as etapas percorridas pelo funcionário para gerar os relatórios da loja.	
Pré condições		
Pós condições	Relatório gerado	
Fluxo Principal		
Ações do Ator		Ações do Sistema
1. Funcionário clica no botão de gerar relatórios		
		2. Sistema informa qual tipo de relatório pode ser gerado
3. Funcionário especifica o tipo de relatório que deve ser gerado		
		4. Sistema gera o relatório
		5. Sistema exibe o relatório.

Fonte: O autor.

#### 4.8.6. Manter Clientes

O quadro a seguir representa as etapas e o relacionamento ator/sistema para cadastrar um cliente.

Quadro 18 – Caso de uso: Manter clientes.

Quadro 15 – Caso de uso: Manter Clientes		
Nome do caso de uso	Manter Clientes	
Caso de uso Geral		
Ator Principal	Funcionário	
Ator(res) Secundário(s)		
Resumo	Este caso de uso descreve as etapas percorridas pelo funcionário para cadastrar, alterar e excluir um cliente	
Pré condições		
Pós condições	Cliente cadastrado	
Fluxo Principal		
Ações do Ator		Ações do Sistema
1. Funcionário clica no botão do formulário de cadastro de clientes		
		2. Apresenta a tela de cadastro
3. Funcionário preenche os dados		
		4. Verifica se os dados estão corretos
5. Funcionário clica em enviar formulário		
		6. Sistema salva os dados no banco de dados
		7. Mostra a tela de sucesso/efetivação

Fonte: O autor.

#### 4.8.7. Vender produto.

O quadro a seguir representa as etapas e o relacionamento ator/sistema para vender produto.

Quadro 19 – Caso de uso: Vender produto.

Quadro 16 - Caso de uso: Vender produto.		
Nome do caso de uso	Vender produto	
Caso de uso Geral		
Ator Principal	Funcionário	
Ator(res) Secundário(s)	Cliente	
Resumo	Este caso de uso descreve as etapas percorridas pelo funcionário para gerenciar a venda de um produto.	
Pré condições	Para efetuar a venda de um produto o mesmo deverá estar cadastrado e ter a quantidade necessária mantida sobre o estoque.	
Pós condições	Produto será vendido	
Fluxo Principal		
Ações do Ator		Ações do Sistema
1. Funcionário inicia o sistema		
		2. Apresenta a tela inicial
3. Funcionário lança os produtos e quantidade		
		4. Verifica a disponibilidade no estoque
5. Funcionário verifica a forma de pagamento		
6. Cliente escolhe a forma de pagamento		
7. Funcionário finaliza venda		
		8. Sistema salva as alterações no banco de dados
		9. Mostra a tela de sucesso/efetivação
Restrições/validações		1. Para efetuar a venda de um produto é necessário ter quantidade disponível.
Fluxo de Exceção - Quantidade indisponível		
Ações do Ator		Ações do Sistema
		1.Informa ao funcionário que o produto não possui quantidade disponível
2. Funcionário cancela a venda		

Fonte: O autor.

#### 4.8.8. Controlar caixa.

O quadro a seguir representa as etapas e o relacionamento ator/sistema para controle de caixa.

Quadro 20 – Caso de uso: Controlar caixa.

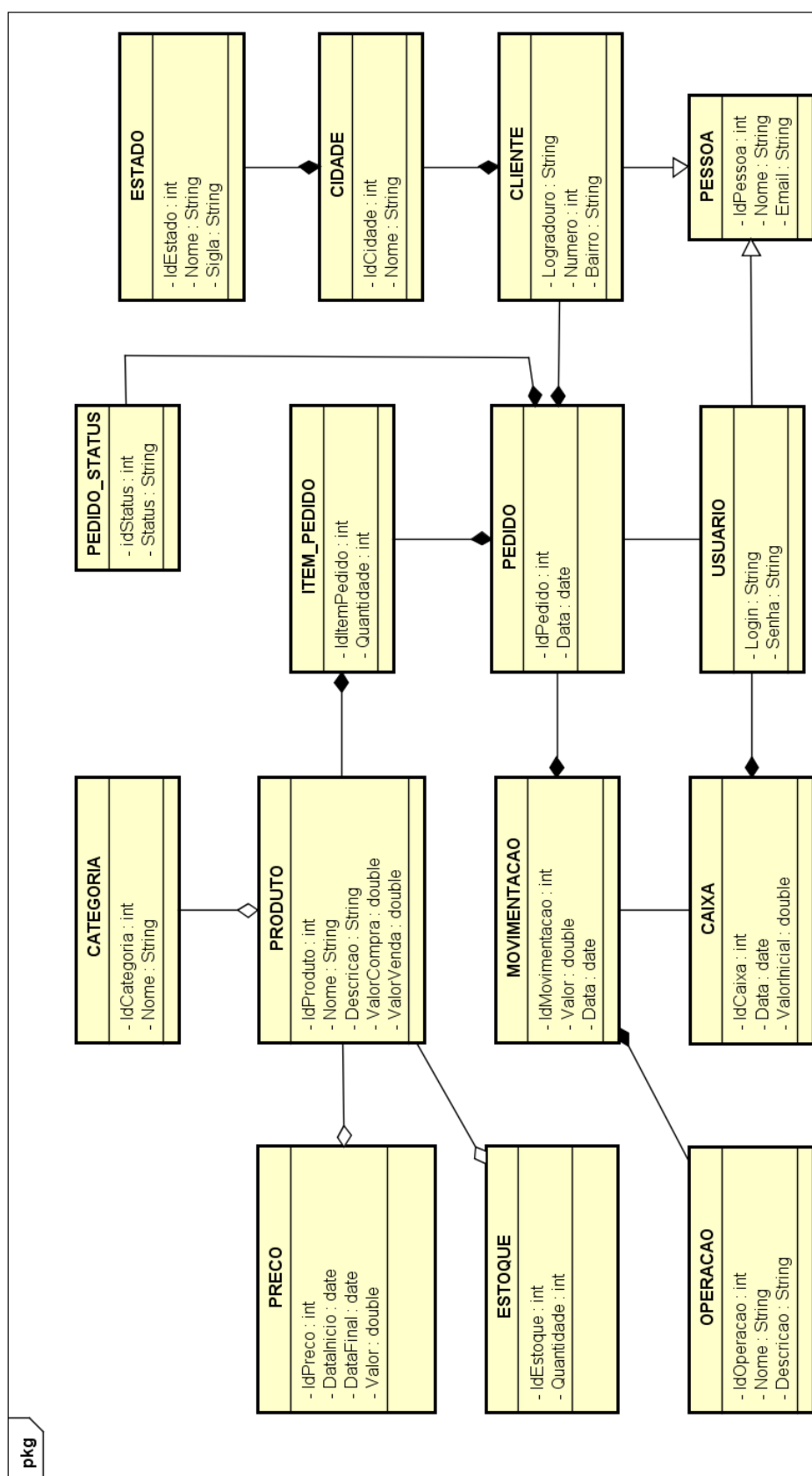
Nome do caso de uso	Controlar caixa	
Caso de uso Geral		
Ator Principal	Funcionário	
Ator(res) Secundário(s)		
Resumo	Este caso de uso descreve as etapas percorridas pelo funcionário para controlar caixa.	
Pré condições		
Pós condições	O fluxo de caixa será controlado	
Fluxo Principal		
Ações do Ator		Ações do Sistema
1. Funcionário clica no botão de relatórios		
		2. Apresenta as opções de relatórios
3. Funcionário escolhe a opção relatório de vendas		
		4. Informa os dados das vendas
5. Funcionário atualiza fluxo de caixa		
		6. Sistema salva os dados no banco de dados
		7. Mostra a tela de sucesso/efetivação

Fonte: O autor.

#### 4.9. Diagrama de classe.

A figura a seguir representa o diagrama de classes do sistema AFSport.

Figura 7 - Diagrama de classes AFSPort.



powered by Astah

Fonte: O autor.

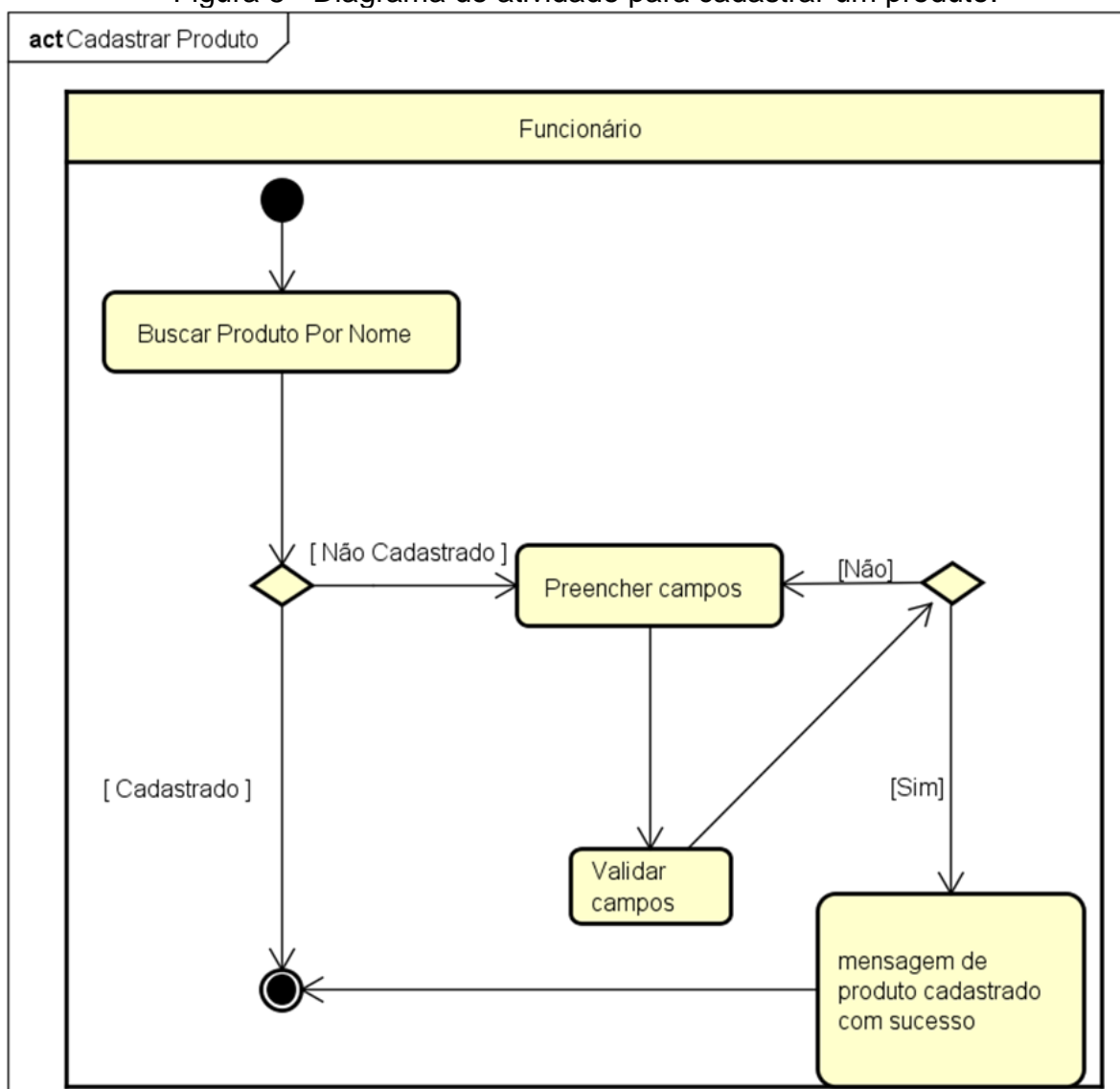
#### 4.10. Diagrama Atividades

Abaixo estão listados os diagramas de atividades para cadastro de produto, cadastro de cliente e venda.

##### 4.10.1. Cadastrar produto

A figura a seguir representa o diagrama com os passos até a conclusão do cadastro de um produto.

Figura 8 - Diagrama de atividade para cadastrar um produto.

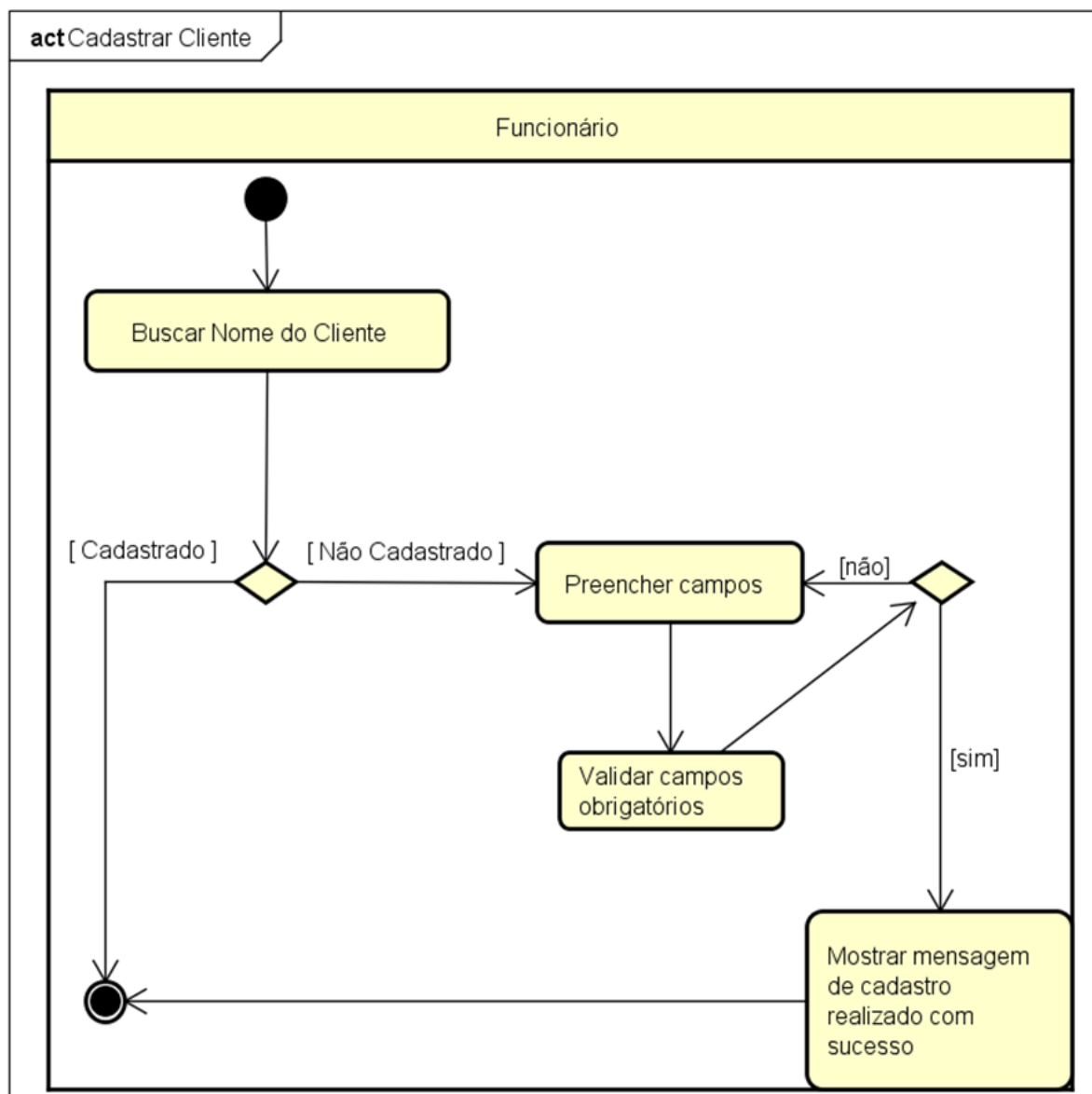


Fonte: O autor.

#### 4.10.2. Cadastro de cliente.

A figura a seguir representa o diagrama com os passos até a conclusão do cadastro de um cliente.

Figura 9 - Diagrama de atividade para cadastrar um cliente.



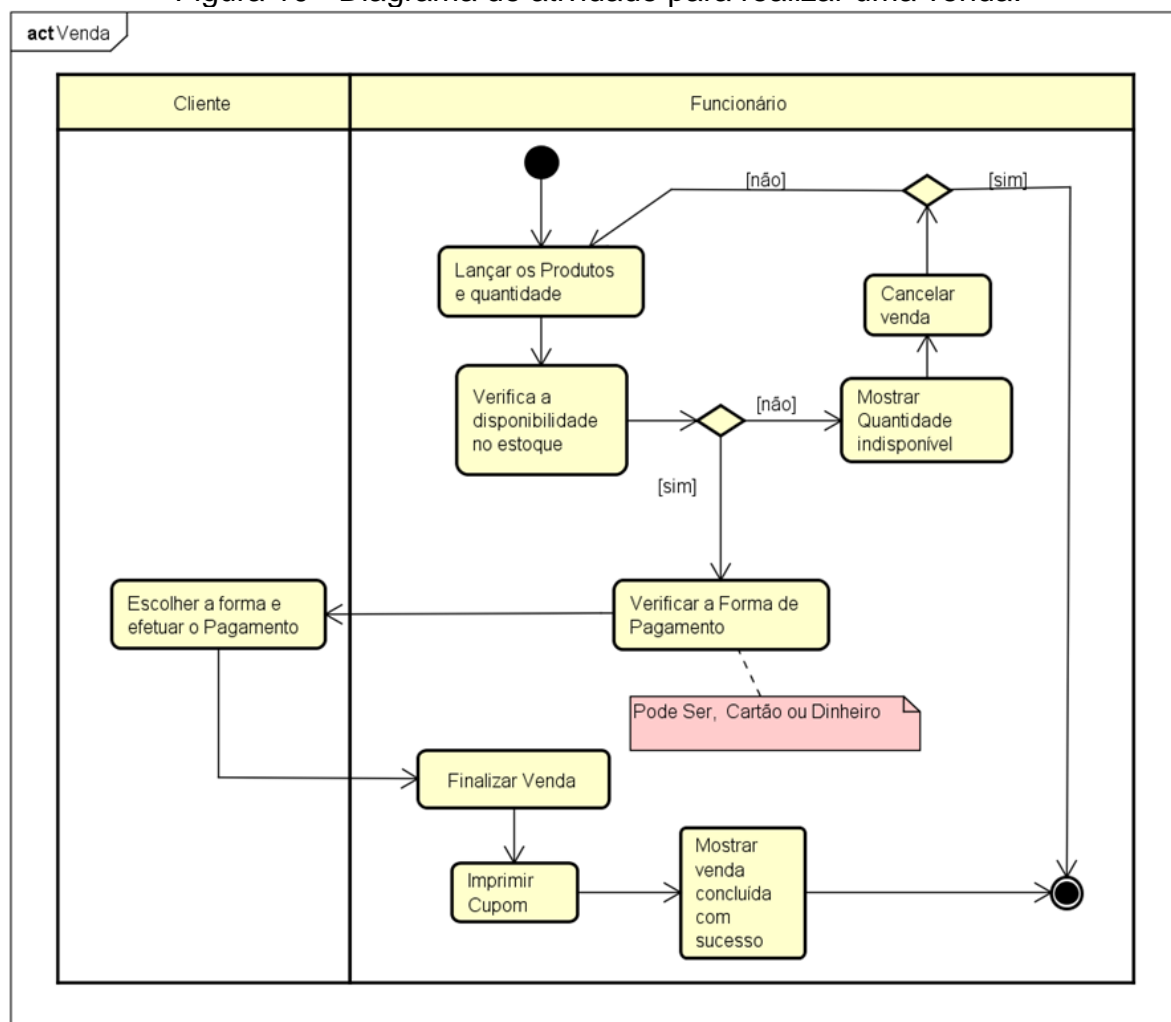
Fonte: O autor.



### 4.10.3. Venda

A figura a seguir representa o diagrama com os passos até a conclusão da venda de um produto.

Figura 10 - Diagrama de atividade para realizar uma venda.



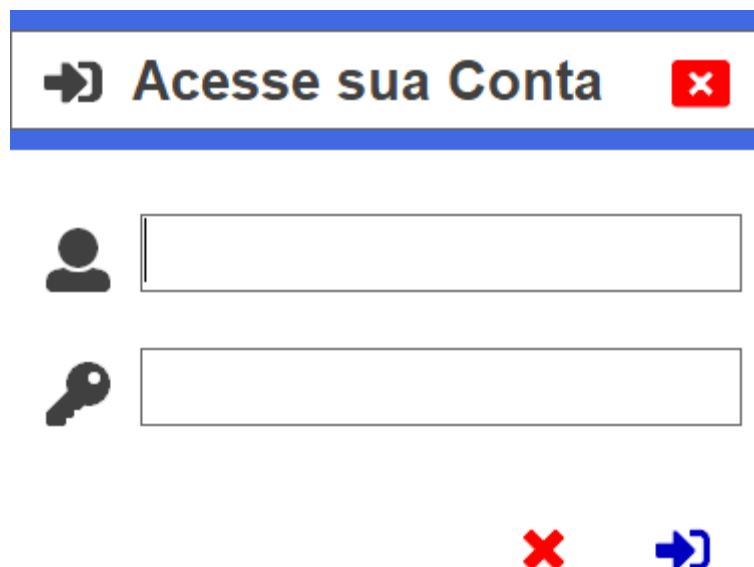
Fonte: O autor.

### 4.11. Software AFSPORT

Diante os requisitos levantados, o sistema AFSPORT foi projetado e desenvolvido visando ser intuitivo de fácil aprendizagem e memorização. Logo abaixo adicionamos as telas do sistema e uma breve descrição delas.

Abaixo está a tela de login, onde o usuário fornece o nome de usuário e senha para poder utilizar o sistema.

Figura 11 - Tela de Login.

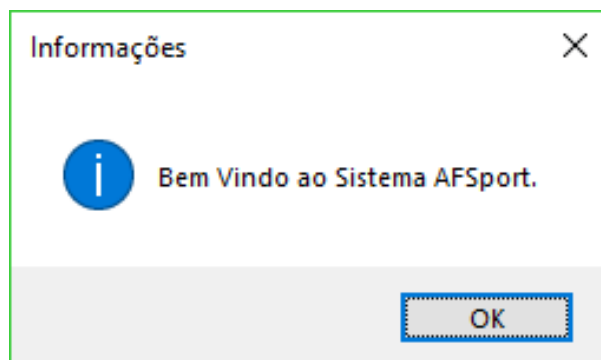


The login screen features a blue header bar with a white background. Inside the header, there is a blue icon of a key with an arrow pointing right, followed by the text "Acesse sua Conta" in bold black font, and a red square button with a white 'X' icon. Below the header, there are two input fields. The first field is preceded by a black user icon, and the second field is preceded by a black key icon. At the bottom right of the screen, there are two icons: a red 'X' and a blue key icon with an arrow pointing right.

Fonte: O autor.

Após preencher os campos, caso as informações estejam corretas o sistema exibe uma mensagem de “Bem vindo”.

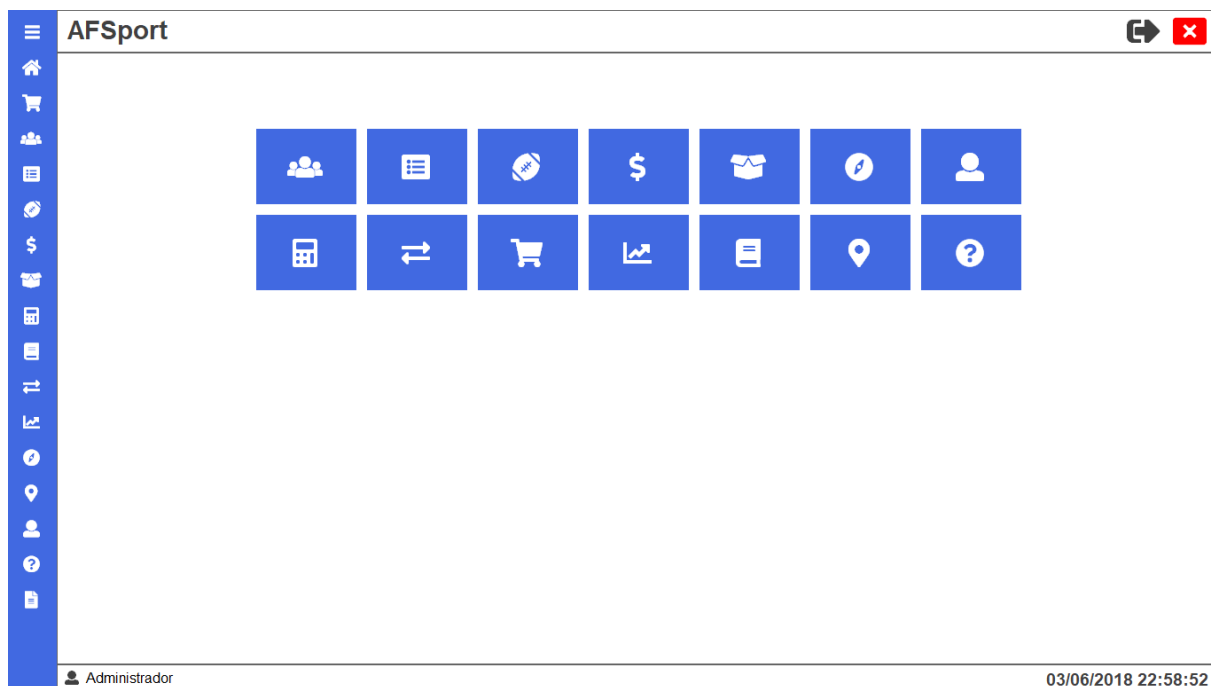
Figura 12 - Tela de boas vindas.



Fonte: O autor.

Na próxima figura temos o menu principal do sistema com os botões da barra central no estilo Windows 8 e 10.

Figura 13 - Tela de menu principal do sistema.



Fonte: O autor.

Figura 14 - Tela de pedidos cadastrados.



Fonte: O autor.

Figura 15 - Tela de cadastro de pedido.

**Cadastro de Pedido**

0

**Produtos Adicionados**

Produto	Qtd.
---------	------

**Seleção de Cliente**

Nome : -  
Endereço : -

**Seleção de Produtos**

Nome Categoria Valor

1

0,00

1

Fonte: O autor.

Figura 16 - Tela de clientes cadastrados.

**AFSport**

**Clientes**

Código	Nome	Email	Logradouro	Nº	Bairro	Cidade	Ativo
1	Rafael Antonio Lopes Alves	rafaelopes.01@hotmail.com	Imigrantes japoneses	490	Parque Ribeirão ...	Ribeirão Preto-SP	<input checked="" type="checkbox"/>

Administrador 03/06/2018 23:05:22

Fonte: O autor.

Figura 17 - Tela de cadastro de cliente.

 **Cadastro de Clientes** 

0

Nome

Email

Logradouro

Numero

0

Bairro

Estado

SP

Cidade

Ribeirão Preto

Fonte: O autor.

Figura 18 - Tela de categorias cadastradas.

AFSport

Categorias de Produtos

Código	nome	Descrição	Ativo
--------	------	-----------	-------

Administrador 03/06/2018 23:06:36

Fonte: O autor.

Figura 19 - Tela de cadastro de nova categoria.

Nova Categoria

0

Nome

Descrição

☒ Ativo

Save Delete

Fonte: O autor.

Figura 20 - Tela de produtos cadastrados.



AFSport

Produtos

Código	Nome	Descrição	Categoria	Valor de Compra	Valor de Venda	Ativo
--------	------	-----------	-----------	-----------------	----------------	-------

Administrador 03/06/2018 23:08:13

Fonte: O autor.

Figura 21 - Tela de cadastro de produto.

 **Cadastro de Produtos** 

0

Nome

Descrição

Valor Compra

0

Valor Venda

0

Categoria

☒ Ativo

Fonte: O autor.



Figura 22 - Tela de preços cadastrados.



Fonte: O autor.

Figura 23 - Tela de cadastro de preço.

The screenshot shows the 'Preço' (Price) registration form. The form has a blue header with the title '\$ Preço' and a red close button. The main area contains the following fields:

- ID**: A large text input field.
- Valor**: A text input field with the value '0'.
- Data Inicio**: A date input field with the value '03/06/2018' and a calendar icon.
- Data Final**: A date input field with the value '03/06/2018' and a calendar icon.
- Produto**: A text input field with a dropdown arrow.

At the bottom, there are two buttons: a blue save button and a red delete button.

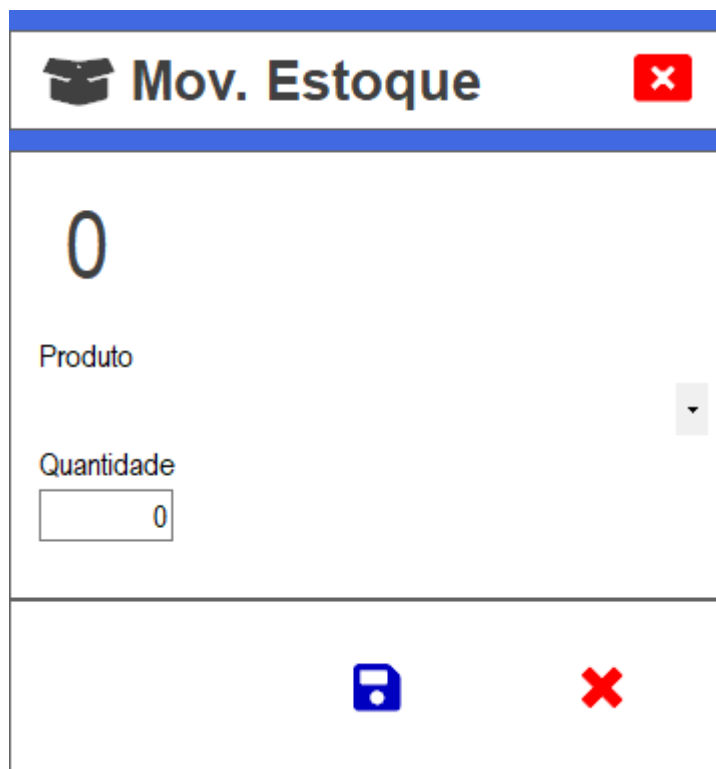
Fonte: O autor.

Figura 24 - Tela de estoques cadastrados.



Fonte: O autor.

Figura 25 - Tela de adicionar estoque.



Fonte: O autor.

Figura 26 - Tela de movimentações de caixa.

AFSport

Caixas

+    ✎    ✖

Código	Valor Inicial	Usuário	Data
--------	---------------	---------	------

Administrador 03/06/2018 23:12:02

Fonte: O autor.

Figura 27 - Tela de abertura de caixa.

Caixa

0

Valor Inicial

0

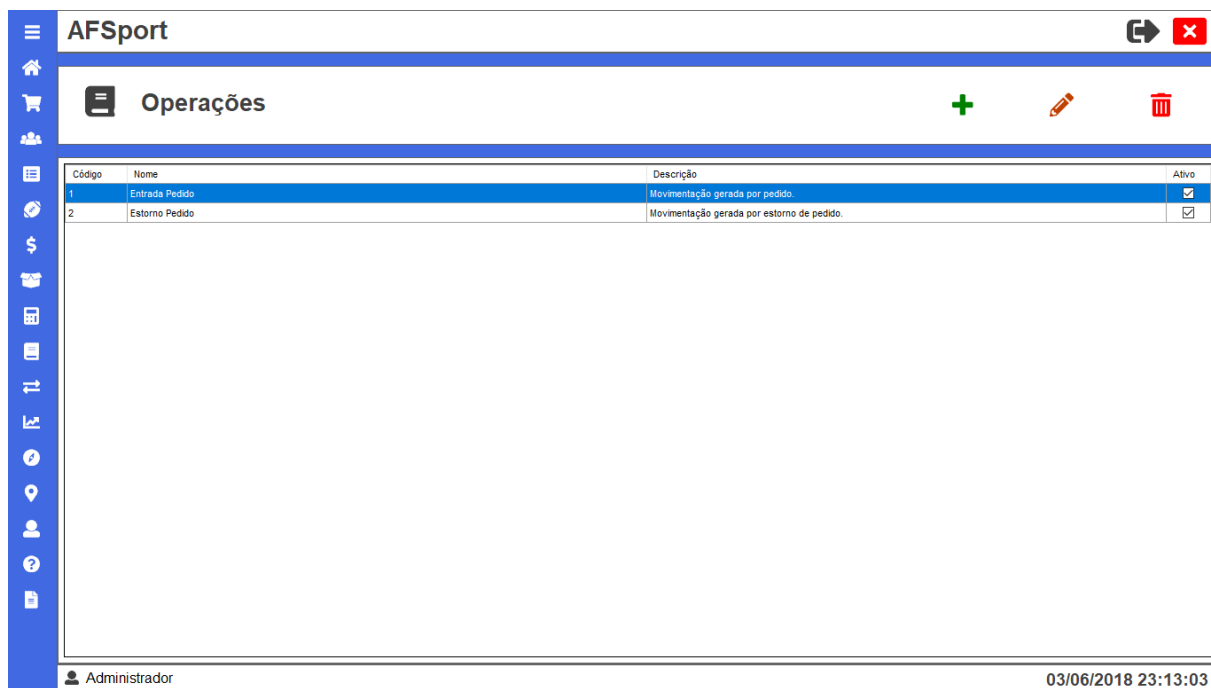
Data

3/06/2018

💾 ✖

Fonte: O autor.

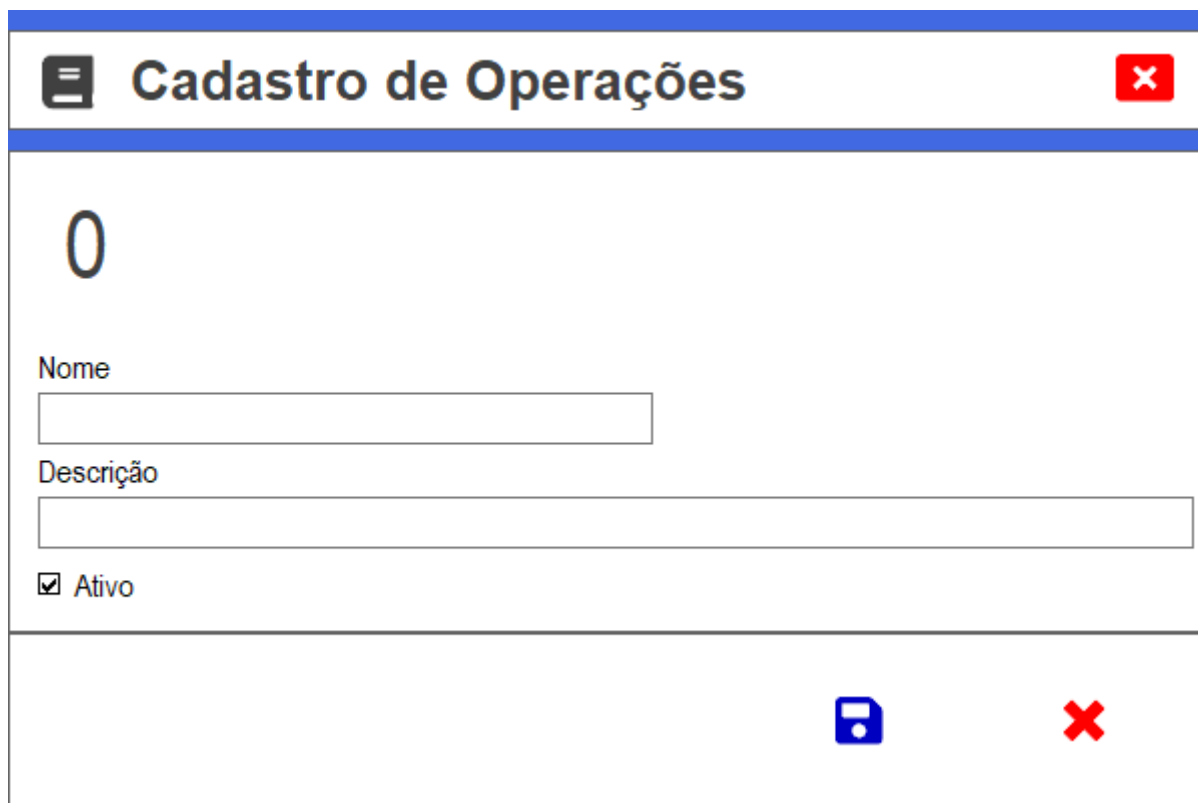
Figura 28 - Tela de operações cadastradas.



Código	Nome	Descrição	Ativo
1	Entrada Pedido	Movimentação gerada por pedido.	<input checked="" type="checkbox"/>
2	Estorno Pedido	Movimentação gerada por estorno de pedido.	<input checked="" type="checkbox"/>

Fonte: O autor.

Figura 29 - Tela de cadastro de operações.



**Cadastro de Operações**

0

Nome

Descrição

☒ Ativo

Fonte: O autor.

Figura 30 - Tela de movimentações cadastradas.

AFSport

Movimentações

Código	Usuário	Operação	Valor	Data
--------	---------	----------	-------	------

Administrador 03/06/2018 23:13:57

Fonte: O autor.

Figura 31 - Tela de cadastro de movimentação.

Mov.

0

Valor

0

Operação

Save Close

Fonte: O autor.

Figura 32 - Tela de estados cadastrados.

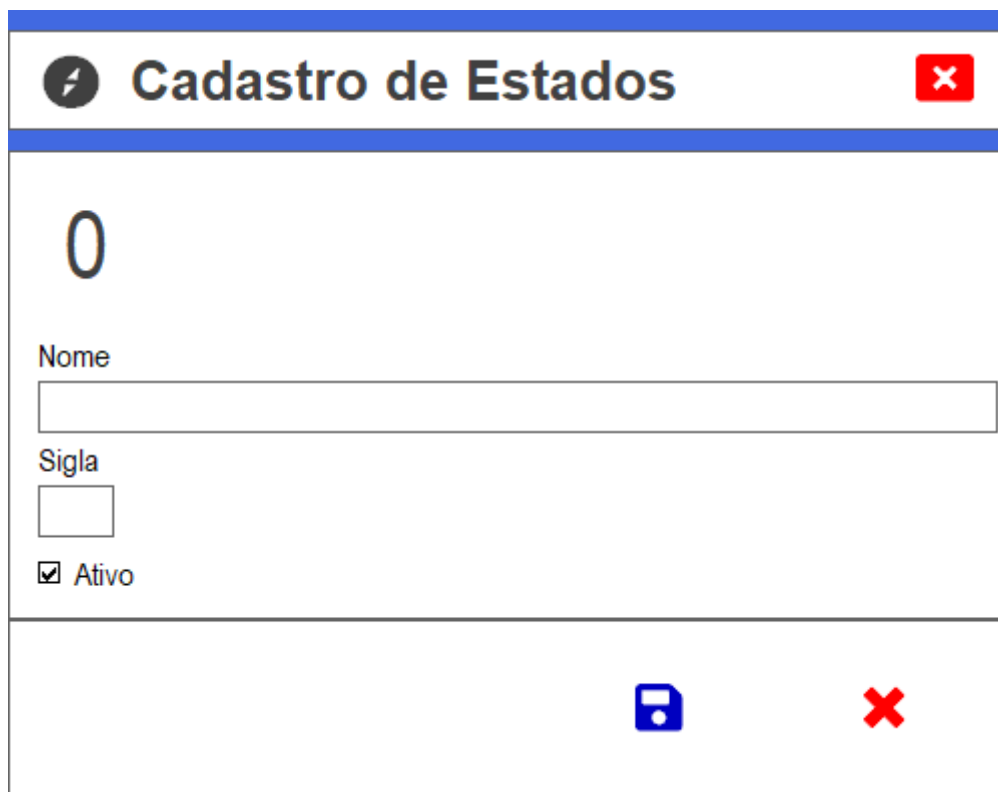


Código	Nome	Sigla	Ativo
1	São Paulo	SP	<input checked="" type="checkbox"/>

Administrador 03/06/2018 23:14:57

Fonte: O autor.

Figura 33 - Tela de cadastro de estado.





**Cadastro de Estados**

0

Nome

Sigla

☒ Ativo

Fonte: O autor.

Figura 34 - Tela de cidades cadastradas.



Código	Nome	Estado	Ativo
1	Ribeirão Preto	São Paulo - SP	<input checked="" type="checkbox"/>

Fonte: O autor.

Figura 35 - Tela de cadastro de cidade.



**Cadastro de Cidades**

0

Nome

Estado

☒ Ativo

Fonte: O autor.

Figura 36 - Tela de usuários cadastrados.

The screenshot displays the 'Usuários' (Users) management page in the AFSport application. The page features a blue sidebar with various icons, a top header with the AFSport logo, and a main content area with a table of users. The table has columns for 'Código', 'Nome', 'Email', 'Login', and 'Ativo'. There is one user listed: 'Administrador' with ID 1, email 'admin', and status 'Ativo'. The interface also includes a top bar with a search icon and a close button, and a bottom status bar showing the user is 'Administrador' and the time is '03/06/2018 23:17:30'.

Código	Nome	Email	Login	Ativo
1	Administrador		admin	<input checked="" type="checkbox"/>

Fonte: O autor.



Figura 37 - Tela de cadastro de usuário.



A interface de usuário para o cadastro de um novo usuário. No topo, há uma barra azul com o ícone de uma pessoa, o título "Cadastro de Usuários" e um botão vermelho com um "X" para fechar. Abaixo, um grande número "0" indica o total de usuários cadastrados. O formulário contém campos para "Nome", "Email", "Login" e "Senha", todos com retângulos de entrada de texto. Abaixo dos campos, há uma opção "Ativo" com uma caixa de seleção marcada. Na base da tela, há dois botões: um azul com um ícone de disco (salvar) e um vermelho com um ícone de "X" (cancelar).

Fonte: O autor.

#### 4.12. Repositório Git dos arquivos

Os arquivos fonte do sistema e instruções para sua utilização estão armazenados no repositório Git abaixo:

<https://github.com/paulobressan/AFSport>

## **5. CONCLUSÃO**

O Desenvolvimento do software de controle e gestão de uma loja de artigos esportivos nos trouxe a uma experiência mais do que simplesmente acadêmica, utilizamos práticas, procedimentos e ferramentas profissionais, as mesmas utilizadas em pequenas e grandes empresas. Ferramentas como o GitHub e o Trello, nos garantiram controle e consistência em nossos recursos para o desenvolvimento, versionamento e cronogramas, com prazos e metas pré-estabelecidas, tornando assim a equipe sincronizada e ciente das etapas já concluídas e das que ainda estavam por vir.

No decorrer do trabalho, foram aplicadas práticas de Análise Orientada a Objetos, onde com os diagramas ficou claro o que foi proposto e quais seriam as formas a serem desenvolvidas as funções e implementações do software, uma etapa de suma importância onde foi possível alinhar a equipe de desenvolvimento para o objetivo proposto no início.

As práticas aplicadas de Orientação a Objetos tornaram o desenvolvimento mais ágil e padronizado, além de simplificar substancialmente alterações no software quando necessário, o reaproveitamento de código possibilitou que mais funções fossem implementadas com um esforço menor e uma economia de tempo considerável.

Os conceitos e sínteses abordados em IHC nos levaram a criação de um software de fácil utilização e agradável aos olhos, onde pudemos receber um feedback positivo no Workshop e acatarmos muitas sugestões interessantes.

Foi um trabalho complexo e exaustivo, com muitos detalhes a serem implementados, mas com o resultado final a equipe se sente orgulhosa e satisfeita e também mais preparada para lidar com os próximos desafios e adversidades.

### **5.1. Projetos futuros**

Futuramente o sistema AFSport contará com uma interface desenvolvida para acesso web e também móbil, e também melhorias na interface e na interação diante do levantamento realizado no workshop.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

BARBOSA, Simone D. J. (Simone Diniz Junqueira). **Interação humano-computador** [recurso eletrônico] / Simone Diniz Junqueira Barbosa, Bruno Santana da Silva. - Rio de Janeiro: Elsevier, 2011.

CASTELLI, Ian. **Trello: como esta ferramenta pode ajudar você a organizar a sua vida.** TECMUNDO. Disponível em: <<https://www.tecmundo.com.br/organizacao/75128-trello-ferramenta-ajudar-voce-organizar-vida.htm>> Acesso em 12/05/2018.

DATE, C. J. Introdução a sistemas de bancos de dados. 7. ed. Rio de Janeiro: Campus, 2000.

FERRARI, Fabrício Augusto. **Crie banco de dados em MySQL** / Fabrício Augusto Ferrari. – São Paulo: Digerati Books, 2007.

PORTAL WIKILIVROS. **Git**. Disponível em: <[https:// https://pt.wikibooks.org/wiki/Git](https://pt.wikibooks.org/wiki/Git)> Acesso em 29 de abril de 2018.

FOWLER, Martin; KENDAL, Scott. **UML Essencial: um breve guia para a linguagem padrão de modelagem de objetos**. Trad. Vera Pezerico e Christian Thomas Price. 2. ed. Porto Alegre: Bookman, 2000.

FURLAN, José Davi. **Modelagem de Objetos através da UML – the Unified Modeling, Language**. São Paulo: Makron Books, 1998.

JOYANES AGUILAR, LUIS. **Fundamentos de programação: algoritmos, estruturas de dados e objetos** / Luis Joyanes Aguilar; tradução: Paulo Heraldo Costa do Valle; revisão técnica: Flávio Soares Corrêa da Silva. – 3. Ed. – Dados eletrônicos. – Porto Alegre: AMGH, 2011.

LAUDON, Kenneth C., 1994- **Sistemas de Informações Gerenciais** - Kenneth C. Laudon e Jane P. Laudon; tradução Thelma Guimarães; revisão técnica Belmiro N. João. – 7. Ed. – São Paulo: Pearson Prentice Hall, 2007.

MEDEIROS, Luciano Frontino de. **Banco de dados: princípios e prática** / Luciano

Frontino de Medeiros – Curitiba: Ibpex, 2007. PORTAL DEVMEDIA. **Git e Github**. Disponível em: <<https://www.devmedia.com.br/guia/git-e-github/37585>> Acesso em 28 abril de 2018.

PORTAL WIKILIVROS. **Programação orientada a objetos**. Disponível em: [https://pt.wikibooks.org/wiki/Programação\\_Orientada\\_a\\_Objeto/Introdução](https://pt.wikibooks.org/wiki/Programação_Orientada_a_Objeto/Introdução) Acesso em 29 de abril de 2018.

SHARP, JOHN., Microsoft Visual C# 2013: **passo a passo [recurso eletrônico]**; tradução: João Eduardo Nóbrega Tortello; revisão técnica: Daniel Antonio Callegari. – Porto Alegre: Bookman, 2014.

SOMMERVILLE, IAN. **Engenharia de Software** / Ian Sommerville; tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hiram. — 9. ed. — São Paulo: Pearson Prentice Hall, 2011.

TRELLO. Disponível em: < <https://trello.com/about>> Acesso em 12/05/2018.