

UNIVERSIDADE ESTADUAL DE GOIÁS  
UNIDADE UNIVERSITÁRIA DE ITABERAÍ  
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Andreone Simeão Lopes

**Software de Gerenciamento de Dados do Sindicato dos Trabalhadores Rurais de  
Taquaral de Goiás**

Itaberaí  
2022

Andreone Simeão Lopes

**Software de Gerenciamento de Dados do Sindicato dos Trabalhadores Rurais de  
Taquaral de Goiás**

Trabalho Conclusão do Curso de Graduação em  
Bacharelado em Sistemas de Informação da Unidade  
Universitária de Itaberaí da Universidade Estadual de  
Goiás como requisito para a obtenção do título de  
Bacharel em Sistemas de Informação.  
Orientador: Prof. Alison Filgueiras

Itaberaí

2022

## Ata de Qualificação do Trabalho de Curso

Às 19 horas e 30 minutos do dia 17 de Março de 2022, em seção remota mediada pelo *Google Meet*, procedeu-se a qualificação do Trabalho de Curso de Bacharelado em Sistemas de Informação, do acadêmico **Andreone Simeão Lopes** com o tema: **Software de Gerenciamento de Dados do Sindicato dos Trabalhadores Rurais de Taquaral de Goiás** sob as orientações do professor: **Alison Carlos Filgueiras**

À oportunidade foi convidado como leitor, a professora **Bruna Vaz Vieira Brandão** e a professora **Juliana Vasconcelos Braga** para fazer parte da Banca Examinadora. Após realizada a apresentação do acadêmico no período mínimo 10 (dez) e máximo 15 (quinze) minutos, foi aberto espaço para as arguições do(a) professor(a) convidado(a). Em seguida, o professor(a) orientador(a), responsável por presidir a Banca Examinadora, reuniu-se com o(a) professor(a) avaliador(a) para atribuir a nota final de **6,0 pontos**, estando o acadêmico **APROVADO**.

[ x ] APROVADO [ ] REPROVADO ou [ ] EXAME FINAL

na disciplina Trabalho de Curso I que representa o pré-requisito para a matrícula na disciplina Trabalho de Curso II e a continuidade na execução do Trabalho de Curso. A banca recomenda a regularização às seguintes ressalvas:

Correções gerais em todo o trabalho. Revisar e reorganizar os capítulos

Correções na especificação dos requisitos.

Sendo esta a expressão da verdade, eu, professor **Alison Carlos Filgueiras**, lavrei a presente Ata, que após lida e achada conforme vai ser assinada por todos e entregue à Secretaria da Unidade Universitária de Itaberaí, para os devidos fins.

Bruna Vaz Vieira Brandão  
Professora Leitora

Juliana Vasconcelos Braga  
Professora Leitora

Alison Carlos Filgueiras  
Professor Orientador

S864s

SIMEÃO LOPES, ANDREONE

SOFTWARE DE GERENCIAMENTO DE DADOS DO SINDICATO DOS  
TRABALHADORES DE TAQUARAL DE GOIÁS / ANDREONE SIMEÃO  
LOPES; orientador ALISON FILGUEIRAS. -- ITABERAÍ, 2022.

39 p.

Graduação - Sistemas de Informação -- Unidade de  
Itaberaí, Universidade Estadual de Goiás, 2022.

1. SOFTWARE DE GERENCIAMENTO. 2. ANÁLISE DE  
REQUISITOS. 3. BANCO DE DADOS . 4. SINDICATO DOS  
TRABALHADORES. 5. PROGRAMAÇÃO ORIENTADA A OBJETO. I.  
FILGUEIRAS, ALISON, orient. II. Título.

Este trabalho é dedicado minha família em especial minha esposa  
Darck Ane e minha filhinha Ester.

O insucesso é apenas uma oportunidade para recomeçar com mais inteligência.

( Henry Ford)

## **AGRADECIMENTOS**

A Deus que me deu vida, ânimo e saúde para vencer todos os desafios e obstáculos desta jornada.

A todos desta Universidade, professores, funcionários e colegas de sala.

Ao meu orientador Professor Alison Filgueiras.

A minha querida família que me incentivou, me apoiou e esteve sempre ao meu lado em todos os momentos.

A todos que de certa forma me apoiaram nessa jornada

## RESUMO

Este projeto tem o intuito de desenvolver e implementar um software de gestão e controle para o Sindicato dos Trabalhadores Rurais de Taquaral de Goiás, ajudando a garantir a representatividade e a defesa dos direitos do trabalhador e da trabalhadora rural, visando agilizar, facilitar e dinamizar o trabalho da entidade. Substituindo o sistema rudimentar atualmente utilizado, implementando um software capaz de reduzir o fluxo de papéis emitidos e o espaço de armazenamento, controlando as mensalidades recebidas, auxiliando a emissão de documentos e melhorando a eficácia da realização da contabilidade e controle das mensalidades. Armazenando dados dos associados, garantindo a integridade, disponibilidade, confiabilidade e segurança, que são pilares importantes da segurança da informação. Para elaboração deste software será realizada engenharia de requisitos coletando as informações do cliente, buscando os requisitos de maneira que o cliente fale à vontade as suas necessidades. As entrevistas sempre serão realizadas fazendo as perguntas e ouvindo o que o cliente está dizendo. A Análise de Requisitos será realizada por partes separando requisitos de usuário, requisitos funcionais, de qualidades, qualidade de sistema e diagramas de caso de uso descritivos para descrever corretamente o que cada funcionalidade do sistema realizará. As ferramentas utilizadas no momento serão o Astah o Netbeans Project, banco de dados MYSQL Workbench e alguns editores do office como o Word e Excel.

**Palavras-chave:** Software. Banco de Dados. Análise de Requisitos.

## ABSTRACT

This project aims to develop and implement a management and control software “Sindicato dos Trabalhadores Rurais de Taquaral de Goiás”, helping to guarantee the representation and defense of the rights of rural workers, aiming to speed up, facilitate and dynamize the entity's work. Replacing the rudimentary system currently used, implementing software capable of reducing the flow of issued papers and storage space, controlling the monthly fees received, assisting in the issuance of documents and improving the effectiveness of accounting and control of monthly fees. Storing associate data, ensuring integrity, availability, reliability and security, which are important pillars of information security. For the elaboration of this software, requirements engineering will be carried out, collecting the customer's information, seeking the requirements so that the customer speaks freely about their needs. Interviews will always be conducted by asking the questions and listening to what the customer is saying. The Requirements Analysis will be performed in parts separating user requirements, functional requirements, quality requirements, system quality and descriptive use case diagrams to correctly describe what each functionality of the system will accomplish. The tools used at the moment will be Astah, Netbeans Project, MYSQL Workbench database and some office editors such as Word and Excel.

**Keywords:** Software. Database. Requirements Analysis.

## LISTA DE FIGURAS

Figura 1 – Diagrama de Caso de Uso .....	29
Figura 2 – Logar no Sistema.....	29
Figura 3 – Tratar Associado .....	30
Figura 4 – Receber Pagamentos .....	31
Figura 5 – Emitir Relatórios .....	32
Figura 6 – Tratar Logins.....	33
Figura 7 – Diagrama de Classes .....	34
Figura 8 – Diagrama de Entidade e Relacionamento .....	34
Figura 9 – Tela de Login.....	34
Figura 10 – Tela de Cadastro de Associados .....	35
Figura 11– Tela de Cadastro de Login .....	35
Figura 12 – Tela de Receber Pagamentos .....	36
Figura 13 – Tela de Emissão de Pagamentos .....	36

## LISTA DE QUADROS

Quadro 1 – Etapas do projeto .....	18
Quadro 2 – Requisitos funcionais.....	26
Quadro 3 – Requisitos não funcionais.....	27
Quadro 4 – Requisitos normativos gerais.....	27
Quadro 5 – Especificação Logar no Sistema.....	29
Quadro 6 – Especificação Tratar Associados .....	30
Quadro 7 – Especificação Receber Pagamentos.....	31
Quadro 8 – Especificação Emitir Relatório.....	32
Quadro 9 – Especificação Tratar Logins .....	33

## **LISTA DE ABREVIATURAS E SIGLAS**

CNPJ Cadastro Nacional de Pessoas Jurídicas

CONTAG Confederação Nacional dos Trabalhadores na Agricultura

CPF Cadastro de Pessoas Físicas

E-MAIL Eletronic Email

FETAEG Federação dos Trabalhadores Rurais NA Agricultura Familiar do Estado de Goiás

PDF Portable Document Format

POO Programação Orientada a objetos

RNF Requisitos não funcionais

RN Requisitos normativos

RF Requisitos funcionais

SQL Structured Query Language

STR Sindicato dos Trabalhadores Rurais

UML Unified Modeling Language

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>15</b>
1.1	OBJETIVOS .....	15
<b>1.1.1</b>	<b>Objetivo Geral.....</b>	<b>16</b>
<b>1.1.2</b>	<b>Objetivos Específicos .....</b>	<b>16</b>
<b>2</b>	<b>PROBLEMÁTICA .....</b>	<b>16</b>
2.1	JUSTIFICATIVA .....	16
<b>2.1.1</b>	<b>METODOLOGIA.....</b>	<b>17</b>
<b>2.2</b>	<b>FUNDAMENTAÇÃO TEORICA.....</b>	<b>18</b>
<b>2.2.1</b>	<b>POO ( PROGRAMAÇÃO ORIENTADA A OBJETO.....</b>	<b>19</b>
2.2.1.1	<i>Objetos.....</i>	19
2.2.1.2	<i>Classes .....</i>	20
2.2.1.3	<i>Associação .....</i>	20
2.2.1.4	<i>Encapsulamento.....</i>	20
2.2.1.5	<i>Herança .....</i>	21
2.2.1.6	<i>Polimorfismo.....</i>	21
<b>2.3</b>	<b>INTERFACE GRÁFICA.....</b>	<b>22</b>
<b>2.4</b>	<b>BANCO DE DADOS .....</b>	<b>22</b>
<b>2.4.1</b>	<b>MYSQL .....</b>	<b>23</b>
<b>2.5</b>	<b>NETBEANS .....</b>	<b>23</b>
<b>2.6</b>	<b>JAVA .....</b>	<b>24</b>
<b>2.7</b>	<b>ASTAH.....</b>	<b>24</b>
<b>3</b>	<b>MODELAGEM DO SOFTWARE .....</b>	<b>25</b>
<b>3.1</b>	<b>LEVANTAMENTO DE REQUISITOS.....</b>	<b>25</b>
<b>3.1.1</b>	<b>REQUISITOS FUNCIONAIS .....</b>	<b>26</b>
<b>3.1.2</b>	<b>REQUISITOS NÃO FUNCIONAIS .....</b>	<b>27</b>
<b>3.1.3</b>	<b>REQUISITOS NORMATIVOS GERAIS .....</b>	<b>27</b>

<b>3.2</b>	<b>DIAGRAMA DE CASO DE USO .....</b>	<b>28</b>
<b>3.2.1</b>	<b>ESPECIFICAÇÃO DE CASO DE USO .....</b>	<b>29</b>
3.2.1.1	<i>UC01 - Logar no Sistema .....</i>	29
3.2.1.2	<i>UC02 – Tratar Associados .....</i>	30
3.2.1.3	<i>UC03 – Receber Pagamentos.....</i>	31
3.2.1.1	<i>UC04 – Emitir Relatórios .....</i>	32
3.2.1.1	<i>UC05 – Tratar Login.....</i>	32
<b>3.3</b>	<b>DIAGRAMA DE CLASSES .....</b>	<b>33</b>
<b>3.4</b>	<b>DIAGRAMA DE ENTIDADE E RELACIONAMENTO .....</b>	<b>34</b>
<b>3.5</b>	<b>PROTOTIPAÇÃO .....</b>	<b>34</b>
3.5.1	Tela de Login.....	34
3.5.2	Tela de Cadastro de Associados .....	35
3.5.3	Tela de Cadastro de Login.....	35
3.5.4	Tela de Receber Pagamentos .....	36
3.5.5	Tela de Emissão de Relatórios.....	36
<b>4</b>	<b>CONCLUSÃO.....</b>	<b>37</b>
	<b>REFERÊNCIAS.....</b>	<b>38</b>

## **1 INTRODUÇÃO**

O Sindicato dos Trabalhadores Rurais de Taquaral de Goiás é responsável por garantir que os direitos dos trabalhadores rurais sejam cumpridos, combatendo o trabalho escravo infantil, garantindo Educação e Saúde, também representando o trabalhador rural frente ao INSS para auxiliar na aposentadoria dos mesmos. Para isso é necessário que o trabalhador rural seja participante do sindicato e contribua pagando suas mensalidades todo mês.

O software de gestão e controle para o Sindicato dos Trabalhadores Rurais de Taquaral de Goiás, tem como objetivo facilitar, agilizar e dinamizar o trabalho desta entidade, que representa a defesa e direitos do trabalhador rural. Pois essa organização trabalha com armazenamento de dados em papéis, fazendo com que haja o acúmulo dos mesmos, correndo o risco de perdas irreversíveis.

Por essa razão teve-se a necessidade de criar um software de gestão e controle para ajudar, esta instituição no controle de seus dados, substituindo um sistema rudimentar e desatualizado por um programa que irá armazenar de forma segura todos os dados que eles necessitam, além de facilitar o acesso aos mesmos. Dessa forma diminuindo o grande fluxo de armazenamento de papéis e agilizando o trabalho do funcionário.

O software de gestão terá como requisitos, logar no sistema, cadastrar associados, logins, receber pagamentos, emitir relatórios, recibos e realizar consultas.

O programa será em linguagem UML (Unified Modeling Language) ponderando o padrão de projeto DAO (Data Access Object). A área de interação com o usuário será no modelo Desktop, executada com o pacote Swing da Linguagem de programação Java. O Banco de Dados a ser utilizado deverá ser o MySQL. Para a implementação será requerido no mínimo as visões de Diagramas de Caso de Uso, classe e de Entidade e Relacionamentos que represente o Banco de Dados.

Para melhor desenvolvimento e execução deste projeto, será utilizado alguns autores que abordam essas linguagens de programação, entre eles se destacam: Deitel (2017), Pressman (2011) e Pereira (2011).

### **1.1 OBJETIVOS**

Os objetivos desse trabalho são os resultados esperados para a criação desse software.

### **1.1.1 OBJETIVO GERAL**

Este trabalho tem como objetivo o desenvolvimento de um *software desktop* de gestão e controle para o Sindicato dos Trabalhadores Rurais de Taquaral de Goiás, ajudando a garantir a representatividade e a defesa dos direitos do trabalhador e da trabalhadora rural, visando agilizar, facilitar e dinamizar o trabalho da entidade.

### **1.1.2 OBJETIVOS ESPECÍFICOS**

- Pesquisar como é o funcionamento administrativo do Sindicato dos trabalhadores;
- Conhecer como funciona um software desktop;
- Analisar e desenvolver um software desktop que consiga atender as carências mais imprescindíveis do sindicato;
- Elaborar o software criado;
- Testar o software criado.

## **2 POBLEMÁTICA**

A criação desse software se deu devido a problemática que se percebeu, através de uma pesquisa, que destacou o fato que os dados da entidade são armazenados em papéis, causando assim o amontoamento dos mesmos, atrasando o serviço e podendo causar perdas irreparáveis. Dessa forma foi desenvolvido um software desktop de gerenciamento de dados para agilizar, facilitar e dinamizar o trabalho da entidade.

### **2.1 JUSTIFICATIVA**

A tecnologia vem ganhando espaço no mundo, de uma forma positiva ela facilita a vida das pessoas, que usam a mesma como apoio para se organizarem de forma rápida e prática.

Com o desenvolvimento da tecnologia, surge grandes contribuições nas áreas administrativas, em que se destaca a grande importância das tecnologias da informação, que de acordo com França (2013), ajuda a suprir, a controlar, planejar e organizar as funcionalidades da empresa.

O crescimento dos negócios vem acontecendo por causa desta ferramenta que é utilizada para planejar, organizar e controlar os funcionamentos, se utilizados adequadamente. Com isto pode-se minimizar as possíveis falhas que pode ocorrer na empresa. (FRANÇA, 2013).

Nesse sentido o intuito desse projeto é desenvolver um software desktop para o Sindicato dos Trabalhadores Rurais de Taquaral de Goiás, e tem como intuito de ajudar no gerenciamento de dados dos trabalhadores que pertencem a entidade. Dessa forma oferecendo seguridade, facilidade e organização nas contabilidades administrativas.

O software tem a função de logar no sistema, cadastrar associados, logins, receber pagamentos, emitir relatórios, recibos e realizar consultas. Ele é desenvolvido em linguagem UML ponderando o padrão de projeto DAO.

### **2.1.1 METODOLOGIA**

Para a criação desse projeto, foram feitas pesquisas de campos na cidade de Taquaral de Goiás, afim de descobrir qual entidade teria uma maior necessidade de implementar um software de gerenciamento de dados, para melhorar seu sistema administrativo.

Depois de várias pesquisas foi notado que o Sindicato dos Trabalhadores Rurais de Taquaral necessitava urgentemente de um software desktop, pois o mesmo ainda armazenava seus dados em papéis, que poderiam ser facilmente perdidos. Dessa forma foi definido o tema para o projeto: Um Software de Gerenciamento de dados do Sindicato dos Trabalhadores Rurais de Taquaral de Goiás.

No processo de desenvolvimento desse software será aplicada a linguagem de Programação Java, que é uma das mais usuais atualmente utilizada. A plataforma utilizada para a criação será o NetBeans que é uma plataforma onde pode-se desenvolver páginas e aplicações para dispositivos móveis utilizando linguagens Java. O sistema de Banco de Dados utilizado para o desenvolvimento do Software foi o MYSQL.

Para a elaboração desse projeto também foram feitas pesquisas bibliográficas para conhecer cada linguagem que seria utilizada na criação desse software. Depois de conhecer e estudar sobre todos esses aspectos o software desktop foi preparado e criado baseado nos requisitos levantados, tendo como foco as carências do sindicato dos trabalhadores.

Este projeto foi desenvolvido por etapas como apresenta o quadro 1:

<b>Etapa</b>	<b>Responsável</b>	<b>Descrição</b>
1	Andreone	Pesquisas de campos na cidade de Taquaral de Goiás, para descobrir possíveis entidades que necessitava de um software de gerenciamento de dados.
2	Andreone	Pesquisas bibliográficas sobre linguagens de programação e desenvolvimento de software desktop.
3	Andreone	Apresentação dos requisitos, desmembramento de necessidades, construção de diagramas, conclusão das necessidades a serem trabalhadas.
4	Andreone	Criação do Banco de Dados.
5	Andreone	União do Banco de Dados com o software e teste do mesmo.

Quadro 1

## 2.2 FUNDAMENTAÇÃO TEORICA

Para elaboração desse projeto foram utilizados vários recursos que serão descritos nesse capítulo, para melhor compreensão deste trabalho.

### 2.2.1 POO (PROGRAMAÇÃO ORIENTADA A OBJETOS)

A programação orientada a objeto é um modelo de programação, que se baseia na ideia de que objetos possuem dados, que faz a junção do mundo virtual com o mundo real. Na POO o objeto recebe uma classe principal, uma espécie de fôrma, que cria seu aspecto inicial e daí surge seus atributos.

Na programação orientada a objetos um programa de computador é conceituado como um conjunto de objetos que trabalham juntos para realizar uma tarefa. Cada objeto é uma parte do programa, interagindo com as outras partes de maneira específica e totalmente controlada (CADENHEAD; LEMAY, 2005, p. 6).

Ela possui 4 pilares: Encapsulamento, que faz o atributo do objeto ficar invisível, mas continuar funcionando; Herança, há a troca de heranças, onde a classe filha herda os atributos da classe mãe; Polimorfismo, pertence aos modos do objeto, nele diferentes tipos de objetos podem usar um mesmo método; Abstração, é o template, alguns autores relacionam ela como uma combinação de encapsulamento e polimorfismo.

De acordo com (Ascencio e Campos, 2014), a linguagem orientada a objetos tem como foco a percepção dos objetos que integram determinado problema, e seu relacionamento mutuamente. Um objeto possui características que são atributos e o comportamento dos objetos são seus métodos.

Dessa forma pode-se dizer POO é um a forma de representar, o objeto tangível, ou seja, os objetos do mundo real são representados no mundo virtual, através dos programas. Essa nova concepção se fundamenta basicamente em dois princípios: *Classes e Objetos*. Tendo outros conceitos, não menos importantes, mas baseados nos dois anteriores: *Associação, Encapsulamento, Herança e Polimorfismo*.

#### 2.2.1.1 OBJETOS

O objeto é tudo que pode sofrer modificação, ele é uma aproximação da classe. O objeto faz a comunicação do real com o virtual.

“O paradigma de orientação a objeto traz um enfoque diferente da programação estruturada, no sentido de adotar formas mais próximas do mecanismo humano para gerenciar complexidade de um sistema. A programação orientada a objeto possibilita o programador usar conceitos de herança, polimorfismo e encapsulamento.” (MENDES, 2009 apud FERNANDES E MARTINS, 2019, P. 16).

O objeto no POO é o elemento que será criado a partir da definição da classe. O Objeto é algo concreto é uma espécie de elemento que receberá características, que são seus atributos e terá métodos que é seu comportamento.

### 2.2.1.2 CLASSES

A classe é um conjunto de objetos que se destacam por apresentar características comuns. De acordo com RICARTE (2001, p. 3):

A definição de classes e seus inter-relacionamentos é o principal resultado da etapa de projeto de software. Em geral, esse resultado é expresso em termos de alguma linguagem de modelagem, tal como UML. Uma classe é um gabarito para a definição de objetos. Através da definição de uma classe, descreve-se que propriedades — ou atributos — o objeto terá. Além da especificação de atributos, a definição de uma classe descreve também qual o comportamento de objetos da classe, ou seja, que funcionalidades podem ser aplicadas a objetos da classe. Essas funcionalidades são descritas através de métodos. Um método nada mais é que o equivalente a um procedimento ou função, com a restrição que ele manipula apenas suas variáveis locais e os atributos que foram definidos para a classe.

A Classe é um composto de aspectos e modos que caracterizam o grupo de objetos que fazem parte dessa classe. Ela é usada para detalhar um conjunto de objetos com maneiras e atributos parecidos, que se relacionam entre si.

### 2.2.1.3 ASSOCIAÇÃO

A associação é feita entre classes, ou seja, ela estabelece uma relação entre duas classes, onde os objetos que pertencem a elas podem realizar uma ação no lugar do outro, pois estão conectados.

**Associação:** É uma conexão entre classes, e também significa que é uma conexão entre objetos daquelas classes. Uma associação representa que duas classes possuem uma ligação (link), por exemplo, elas "conhecem uma a outra", "estão conectadas com", "para cada X existe um Y" e assim por diante. (DEVMEDIA, 2006).

### 2.2.1.4 ENCAPSULAMENTO

O encapsulamento é utilizado para manter os dados da classe protegida. Ele oculta esses atributos para que não possam ser modificados ou utilizada de forma errada. Esses dados só poderão ser acessados por meio de seus métodos.

*O encapsulamento é uma das principais técnicas que define a programação orientada a objetos. Se trata de um dos elementos que adicionam segurança à aplicação em uma programação orientada a objetos pelo fato de esconder as propriedades, criando uma espécie de caixa preta.*

*A maior parte das linguagens orientadas a objetos implementam o encapsulamento baseado em propriedades privadas, ligadas a métodos especiais chamados *getters* e *setters*, que irão retornar e setar o valor da propriedade, respectivamente. Essa atitude evita o acesso direto a propriedade do objeto, adicionando uma outra camada de segurança à aplicação. (DEV MEDIA,2014)*

O encapsulamento é escondido atrás de uma interface externa. De acordo com Pressman (2011), ele impede que os dados que não importam a outros módulos sejam acessados pelos mesmos, dispondo somente os elementos que lhe cabem, fazendo com que ocorra menos erros a se espalhar no software.

#### **2.2.1.5 HERANÇA**

A herança na POO é o compartilhamento dos atributos entre as classes, por exemplo uma classe herda as características da outra. *“A Herança possibilita que as classes compartilhem seus atributos, métodos e outros membros da classe entre si. Para a ligação entre as classes, a herança adota um relacionamento esquematizado hierarquicamente.”* (DEV MEDIA, 2010).

Ela é composta de dois tipos de classes, a Classe Base que é a que transmite suas características a outra classe, e a Classe Derivada que é a que recebe essas características.

A herança é uma forma de favorecer a programação, pois ela permite que um código já criado, possa ser reutilizado, informando que o mesmo é herança do código base, poupando assim o tempo de programação.

#### **2.2.1.6 POLIMORFISMO**

No Polimorfismo *“dois objetos, de duas classes diferentes, têm um mesmo método que é implementado de formas diferentes, ou seja, um método possui várias formas, várias implementações diferentes em classes diferentes, mas que possuem o mesmo efeito.”* (Henrique, 2019).

Polimorfismo significa “muitas formas”, na programação orientada a objeto é isso que ela exerce. Por exemplo: Um objeto pode ter o mesmo método, mas pode exercer funções diferentes: O cavalo, o gato e o galo por exemplo, pertence a superclasse animal, todos eles emitem som, porém cada um tem um som particular. Assim também é o Polimorfismo, possui uma Superclasse principal, de onde derivam as subclasses que exercem métodos que são implementados de formas diferentes.

## **2.3 INTERFACE GRÁFICA**

A interface gráfica é a forma de comunicação entre o usuário e os aparelhos digitais, isso é feito através de um programa representado na tela na forma de imagem gráfica visual.

De acordo com Santos (2009, p 32):

A interface é o nome dado a toda porção de um sistema com a qual um usuário mantém contato ao utilizá-lo, tanto ativa quanto passivamente. A interface engloba o software e o hardware (dispositivos de entrada e saída tais como: teclados, mouses, tablets, monitores e impressoras e etc. Uma interface gráfica adequada aproveita ao máximo as potencialidades do computador em questão, desde a CPU aos periféricos, de forma a tornar um programa computacional, mais fácil de usar.

Como foi citado por Santos, é muito importante ter conhecimento sobre os elementos e as necessidades do usuário, afim de criar uma interface de fácil acesso, prática e que supre as necessidades do usuário final.

## **2.4 BANCO DE DADOS**

O Banco de dados é uma série de informações ou dados organizados sobre um mesmo assunto, que são armazenadas pelo sistema computacional.

De uma forma simplista, podemos dizer que um banco de dados consiste em uma coleção de dados estruturados, organizados e armazenados de forma persistente. Um banco de dados não tem, necessariamente, de estar informatizado. Pode perfeitamente consistir em um fichário existente em um escritório com o nome dos fornecedores ou mesmo em um conjunto de Post-it colados em um painel de parede com a informação dos automóveis existentes em estoque para venda. No entanto, é necessário que os dados tenham algum tipo de significado e organização. (DAMAS, 2014, p. 15).

O Banco de dados é muito importante para que o usuário consiga ter acesso a todos os comandos do programa e encontre todas as informações que necessita ao navegar no sistema. Além disso ele mantém o software seguro, pois somente o profissional que desenvolveu o mesmo pode acessá-lo, para atualizar, restaurar e armazenar os dados quando necessário.

O Banco de Dados utilizado na criação do software STR de Taquaral de Goiás foi o **MYSQL**.

#### **2.4.1 MYSQL**

O **MYSQL** é um Banco de Dados relacional, com desenvolvimento inicial em 1995, pela empresa Sueca MYSQL AB, porém em 2008 a empresa foi vendida para a empresa norte-americana Sun Microsystems, dessa forma o Banco de Dados passou a pertencer a mesma. Em 2009 outra empresa norte-americana, a Oracle, compra a Sun Microsystems e o MYSQL, tornando-se dona do Sistema desde então.

O **MYSQL** utiliza como interface a linguagem SQL (Structured Query Language, uma linguagem de Consulta Estruturada) e é um dos sistemas mais utilizados no mundo.

O **Mysql** foi criado na Suécia por dois suecos e um finlandês: David Axmark, Allan Larsson e Michael "Monty" Widenius, que têm trabalhado juntos desde a década de 1980. No dia 16 de Janeiro de 2008, a Mysql AB, desenvolvedora do Mysql foi adquirida pela Sun Microsystems, por US\$ 1 bilhão, um preço jamais visto no setor de licenças livres. No dia 20 de abril de 2009 a Oracle compra a Sun Microsystems e todos os seus produtos, incluindo o Mysql. Após investigações da Comissão Européia sobre a aquisição para evitar formação de monopólios no mercado a compra foi autorizada e hoje a Sun faz parte da Oracle. (TANGANELLI, 2011, p.20)

O **MYSQL** é dos Sistemas mais populares utilizados pelas empresas, por ser um sistema que assegura uma alta execução e velocidade, e possuir versatilidade e segurança.

#### **2.5 NETBEANS**

O NetBeans foi criado pela empresa Sun Microsystems, ele é um espaço de desenvolvimento integrado grátis e livre para uso pelos desenvolvedores e de fácil instalação. Ele é codificado em Java e permite que sejam desenvolvidos aplicativos de desktop Java. Possui ferramentas que facilitam e assegura criar softwares.

O **NetBeans** possui um conjunto de ferramentas capaz de gerar código e arquivos de configuração, auxiliar no empacotamento e instalação de aplicações, renomear elementos do código, entre outras. O **NetBeans** é um dos ambientes de desenvolvimento mais utilizados pelos **desenvolvedores Java**. Mais do que um editor de código, ele possui um conjunto de ferramentas que auxiliam a programação de tarefas comuns relacionadas à implementação de aplicações. (DEV MEDIA, 2011).

## 2.6 JAVA

O Java é uma linguagem de programação que permite criar aplicações divertidas, seguras e confiáveis, que podem funcionar em vários sistemas operacionais, ele foi criado pela Sun Microsystems em 1995.

Java é a linguagem de programação orientada a objetos, desenvolvida pela Sun Microsystems, capaz de criar tantos aplicativos para desktop, aplicações comerciais, softwares robustos, completos e independentes, aplicativos para a Web. Além disso, caracteriza-se por ser muito parecida com C++, eliminando as características consideradas complexas, dentre as quais ponteiros e herança múltipla. (CLARO E SOBRAL, 2008, P. 12).

Atualmente a linguagem Java está presente em vários tipos de aplicações, tais como, aplicativos e jogos instalados em celulares e tablets, aplicações desktops, e até mesmos em soluções corporativas.

## 2.7 ASTAH

O Asth foi criado no Japão através de uma plataforma Java, ele é um sistema de modelagem UML. De acordo Com Neto:

Astah é uma ferramenta CASE (Computer-Aided Software Engineering) vastamente utilizada para a modelagem de soluções de software fazendo uso da UML. Ela dispõe de uma versão free “community” e de uma versão paga “professional”. Astah é desenvolvido na plataforma JAVA e permite que seja modelado soluções de software fazendo uso de uma linguagem que seja mais próxima do pensamento humano. Ela admite que os modelos criados sejam transformados em códigos, conceito conhecido na computação como engenharia à frente. Astah também aceita que códigos já criados, sejam transformados em modelos UML, através da engenharia reversa. (2017, p.03).

As ferramentas presentes no Astah permite criar diagramas de modelagem de forma clara e concisa.

### **3 MODELAGEM DO SOFTWARE**

Para o desenvolvimento do software de Gerenciamento de Dados Do Sindicato dos Trabalhadores de Taquaral de Goiás forma feitas várias etapas.

#### **3.1 LEVANTAMENTO DE REQUISITOS**

Este projeto tem o intuito de desenvolver e implementar um software de gestão e controle para o Sindicato dos Trabalhadores Rurais de Taquaral de Goiás, ajudando a garantir a representatividade e a defesa dos direitos do trabalhador e da trabalhadora rural, visando agilizar, facilitar e dinamizar o trabalho da entidade. Substituindo o sistema rudimentar atualmente utilizado, implementando um software capaz de reduzir o fluxo de papéis emitidos e o espaço de armazenamento, controlando mensalidades recebidas pela entidade, melhorando a eficácia da realização da contabilidade e controle das mensalidades, doações pela instituição.

O STR de Taquaral de Goiás conta com aproximadamente 500 associados, que recebem do mesmo os serviços de Advogados para aposentadoria e pedidos de auxílio, engenheiro agrônomo, médico oftalmologista, além de intermediação de serviços da FETAEG (federal) e CONTAG (estadual).

No Software do STR de Taquaral de Goiás os dados dos associados deverão ser cadastrados com o nome, endereço, número, complemento, bairro, cidade, estado, cep, sexo, cpf, rg, estado civil, data de nascimento, celular e e-mail. No momento de fazer o pagamento o usuário deverá informar: associado, inscrição do associado, mês de referência, donativo, carteira social e data.

O software deverá emitir um comprovante de pagamento (recibo), todas as vezes que o mesmo for efetuado, deverá informar também se o associado tem pagamentos pendentes e se tiver muitos meses em atraso o mesmo deverá oferecer as opções de desconto ou parcelamento.

Além disso o software deverá informar se o associado, está aposentado ou não, o status da sua aposentadoria e se o mesmo já está em idade de se aposentar.

Para acessar o sistema o usuário do sindicato precisará entrar com algumas informações na tela de login como: o nome do usuário e a senha, que deverão estar previamente cadastradas para liberação do acesso às funcionalidades do programa.

Ao final de todo o acesso o programa deverá fazer um backup de todos os dados e todos os documentos dos associados devem ser salvos em formato PDF. E ao final de todo Mês o programa deverá emitir um relatório das mensalidades recebidas.

O projeto do sistema deverá ser baseado em UML, considerando o Padrão de projeto o DAO (Data Access Object). A interface de interação com o usuário deverá ser uma aplicação do tipo desktop implementada com o pacote Swing da Linguagem de programação Java. O banco de dados a ser utilizado deverá ser o MySQL. Para a implementação será requerido no mínimo as visões de diagramas de Caso de Uso, Classe e de Entidade e Relacionamentos que represente o Banco de Dados.

### 3.1.1 REQUISITOS FUNCIONAIS

Os Requisitos funcionais é aquilo que o software deve fazer, ele deve executar essas etapas.

1. Requisito Funcionais Gerais (Casos de Uso)		
Id	Nome	Descrição
RF 001	<u>Tratar Logins</u>	É necessário que haja um módulo para todas ações referente a cadastro de logins, o responsável por tal cadastramento terá que informar usuário, senha e status.
RF 002	<u>Tratar Associados</u>	É necessário que haja um módulo para todas ações referente a cadastro de associados, o responsável por tal cadastramento terá que informar, nome, endereço, número, complemento, bairro, cidade, estado, cep, sexo, CPF, RG, estado civil, data de nascimento, celular e e-mail.
RF 003	<u>Receber pagamentos</u>	É necessário que haja um módulo para todas ações referente a receber pagamentos, o responsável por tal recebimento terá que informar, associado, inscrição do associado, mês de referência, donativo, carteira social e data.

<b>RF 004</b>	<b><u>Emitir relatórios</u></b>	É necessário que haja um módulo para todas ações referente a emissão de relatórios, o responsável por tal deverá informar o período.
---------------	---------------------------------	--

Quadro 2

### 3.1.2 REQUISITOS NÃO FUNCIONAIS

<b>1. Requisitos Não-Funcionais Gerais</b>		
<b>Id</b>	<b>Nome</b>	<b>Descrição</b>
<b>NF 001</b>	<b><u>Design não responsivo</u></b>	O Software será Desktop, e terá uma interface simples e com soluções intuitivas.
<b>NF 002</b>	<b><u>Software livre</u></b>	Por ser uma iniciativa da Universidade, tanto para colaboração quanto para fornecimento a outras IE's.

Quadro 3

### 3.1.3 REQUISITOS NORMATIVOS GERAIS

<b>1. Requisitos Normativos Gerais</b>		
<b>Id</b>	<b>Nome</b>	<b>Descrição</b>
<b>NR 001</b>	<b><u>Gerenciar Backup</u></b>	O sistema deve realizar Backup todas as vezes que o software for finalizado.
<b>NR 002</b>	<b><u>Regra de Acesso do Sistema</u></b>	As seguintes funções: Tratar Associados, Tratar Logins, Receber Pagamentos e Emitir Relatórios só poderá ser executada por um usuário logado no sistema.
<b>NR 003</b>	<b><u>Regra de Pagamento</u></b>	O software deverá emitir um comprovante de pagamento (recibo), todas as vezes que o mesmo for efetuado.
<b>NR 004</b>	<b><u>Regra de informação</u></b>	O software deverá informar também se o associado tem pagamentos pendentes e se tiver muitos meses em atraso o mesmo deverá oferecer as opções de desconto ou parcelamento.

NR 005	<b><u>Regra de informação</u></b>	O software deverá informar se o associado, está aposentado ou não, o status da sua aposentadoria e se o mesmo já está em idade de se aposentar.
NR 006	<b><u>Regra de informação</u></b>	O software deverá informar se o associado já tem idade suficiente para aposentar.
NR 007	<b><u>Regra de Documentação</u></b>	Os documentos da empresa devem ser emitidos e armazenados em formato PDF.

Quadro 4

### 3.2 DIAGRAMA DE CASOS DE USO

O Diagrama de Caso de Uso é um dos mais utilizados e serve como suporte para a produção de outros diagramas UML. Ele é um diagrama comportamental da UML, utilizado principalmente nas fases iniciais da modelagem do sistema, Levantamento e análise de requisitos. Seus elementos principais são: os atores, os casos de uso e os relacionamentos.

Assim afirma Machado 2016:

O diagrama de casos de uso é um diagrama comportamental da UML, sendo o mais utilizado entre os diagramas comportamentais. Além disso, ele serve como base para a construção de outros diagramas da UML. O diagrama de casos de uso é utilizado no início da modelagem do sistema, geralmente nas fases de levantamento e análise de requisitos. Ele desempenha um papel importante, pois possibilita a especificação, a visualização e a documentação de funcionalidades e características do sistema. Os principais elementos do diagrama de casos de uso são os atores, os casos de uso e os relacionamentos. (P. 28-29).

Observa-se a seguir o Diagrama de Caso de Uso do Software STR de Taquaral de Goiás.

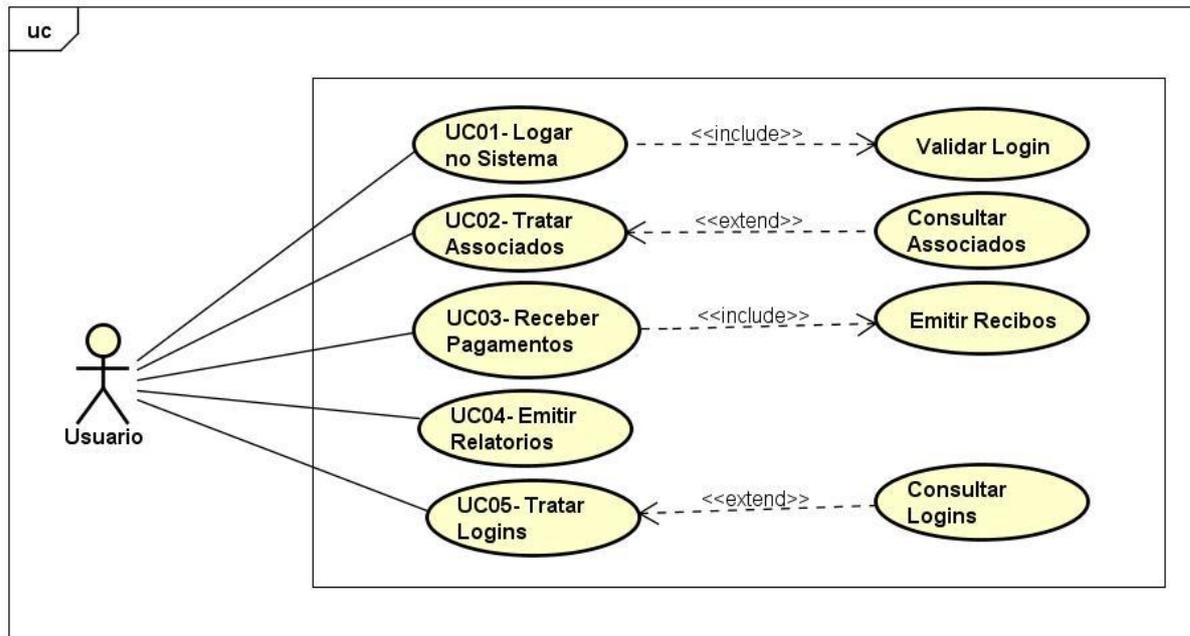


Figura 1

### 3.2.1 ESPECIFICAÇÕES DE CASO DE USO

#### 3.2.1.1 UC01 – LOGAR NO SISTEMA

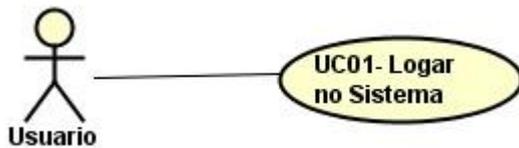


Figura 2

Funcionalidade/Objetivo	Permite ao Usuário, fornecer informações para Logar no sistema.
Ator	Usuário
Pré-Condição	O Usuário deverá estar Cadastrado no sistema.
Cenário Principal	<ol style="list-style-type: none"> <li>1 – O sistema solicita os dados necessários para Logar no sistema.</li> <li>2 – O Usuário informa os dados de acordo com os campos a serem preenchidos.</li> <li>3 – O sistema solicita os dados para executar a função. [A1]</li> <li>4 – O Usuário informa os dados necessários.</li> <li>5 – O Usuário seleciona a opção “Entrar”.</li> <li>6 - O sistema valida os dados de autenticação.</li> </ol>

	7 – O sistema emite a mensagem “Logado com Sucesso”.
Cenário Alternativo	A1 – O Usuário poderá cancelar o processo durante a movimentação.
Casos Teste	5.1 – O sistema verifica se os campos foram preenchidos corretamente. 5.2 – O sistema não confirma o login e emite uma mensagem de erro. 5.3 – O sistema cancela a operação.

Quadro 5

### 3.2.1.2 UC02 – TRATAR ASSOCIADOS

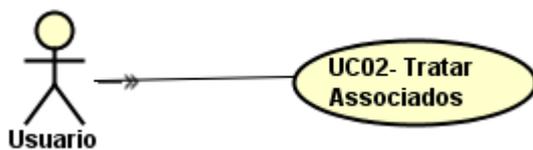


Figura 3

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar Associado.
Ator	Usuário
Pré-Condição	O Usuário deverá estar autenticado no sistema.
Cenário Principal	1 – O sistema solicita os dados necessários para o cadastro do Associado. 2 – O Usuário informa os dados de acordo com os campos a serem preenchidos. 3 – O Usuário informa os dados necessário. [A1] 4 – O Usuário seleciona a opção “Salvar”. 5 – O sistema emite a mensagem “Associado Cadastrado com Sucesso”. 6 – O sistema cadastra o Associado.
Cenário Alternativo	A1 – O Usuário poderá cancelar o processo durante o cadastro.
Casos Teste	4.1 – O sistema verifica se os campos foram preenchidos corretamente.

	<p>4.2 – O sistema não confirma o cadastro e emitir uma mensagem de erro.</p> <p>4.3 – O sistema cancela a operação.</p>
--	--

Quadro 6

### 3.2.1.3 UC03- RECEBER PAGAMENTOS

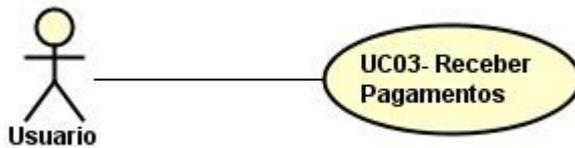


Figura 4

Funcionalidade/Objetivo	Permite ao Usuário, fornecer informações para Receber Pagamentos.
Ator	Usuário
Pré-Condição	O Usuário deverá estar autenticado no sistema.
Cenário Principal	<p>1 – O sistema solicita os dados necessários para movimentar recebimento.</p> <p>2 – O Usuário informa os dados de acordo com os campos a serem preenchidos.</p> <p>3 – O sistema solicita os dados para o cadastro da função. [A1]</p> <p>4 – O Usuário informa os dados necessários.</p> <p>5 – O Usuário seleciona a opção “Salvar”.</p> <p>6 – O sistema emite a mensagem “Operação Realizada com Sucesso”.</p>
Cenário Alternativo	A1 – O Usuário poderá cancelar o processo durante a movimentação.
Casos Teste	<p>5.1 – O sistema verifica se os campos foram preenchidos corretamente.</p> <p>5.2 – O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>5.3 – O sistema cancela a operação.</p>

Quadro 7

### 3.2.1.4 UC04- EMITIR RELATÓRIOS

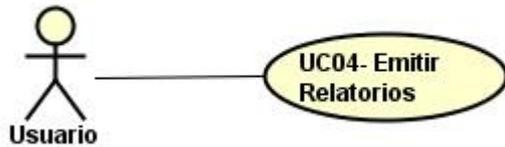


Figura 5

Funcionalidade/Objetivo	Permite ao funcionário emitir relatório recebimentos.
Ator	Usuário
Pré-Condição	O Usuário deverá estar autenticado no sistema.
Cenário Principal	<p>1 – O sistema disponibiliza os dados necessários para o relatório.</p> <p>2 – O Usuário seleciona a aba “Relatório”. [A1]</p> <p>3 – O Usuário seleciona o botão “Imprimir”.</p> <p>4 – O sistema imprime o relatório com sucesso.</p>
Cenário Alternativo	A1 – O Usuário poderá visualizar o relatório e não imprimir.
Casos Teste	4.1 – O Usuário cancela a operação.

Quadro 8

### 3.2.1.5 UC05- TRATAR LOGINS

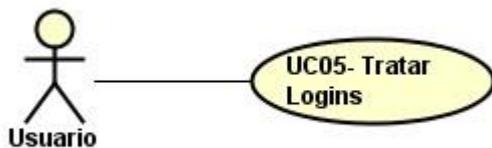


Figura 6

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar Login.
Ator	Usuário
Pré-Condição	O Usuário deverá estar autenticado no sistema.

Cenário Principal	<p>1 – O sistema solicita os dados necessários para o cadastro do Login.</p> <p>2 – O Usuário informa os dados de acordo com os campos a serem preenchidos.</p> <p>3 – O Usuário informa os dados necessário. [A1]</p> <p>4 – O Usuário seleciona a opção “Salvar”.</p> <p>5 – O sistema emite a mensagem “Login Cadastrado com Sucesso”.</p> <p>6 – O sistema cadastra o Login.</p>
Cenário Alternativo	A1 – O Usuário poderá cancelar o processo durante o cadastro.
Casos Teste	<p>4.1 – O sistema verifica se os campos foram preenchidos corretamente.</p> <p>4.2 – O sistema não confirma o cadastro e emitir uma mensagem de erro.</p> <p>4.3 – O sistema cancela a operação.</p>

Quadro 9

### 3.3 DIAGRAMA DE CLASSES

Diagrama de Classes é um diagrama que mapeia de forma clara e precisa as classes que serão utilizadas na criação do software.

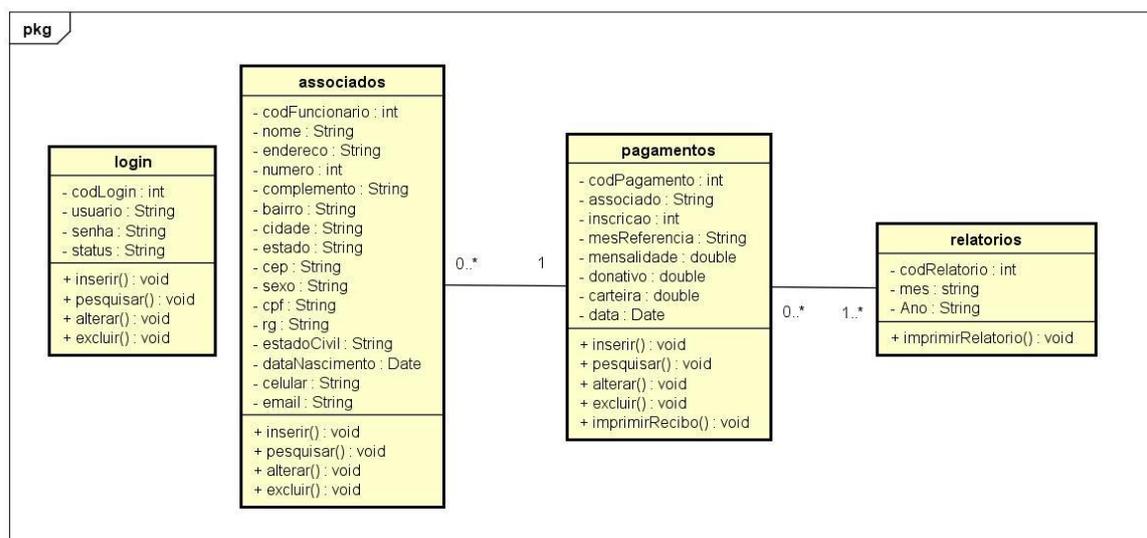


Figura 7

### 3.4 DIAGRAMA DE ENTIDADE E RELACIONAMENTO

O Diagrama de Entidade e Relacionamento é um diagrama que projeta como as entidades se relacionam entre si em um sistema.

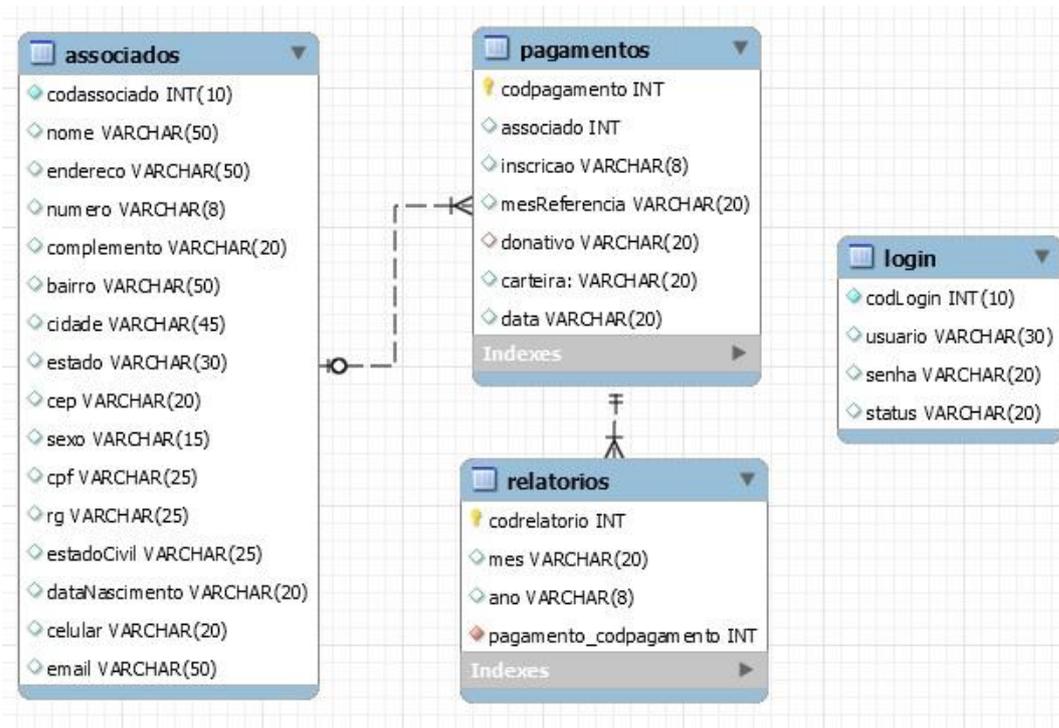


Figura 8

### 3.5 PROTOTIPAÇÃO

#### 3.5.1 TELA DE LOGIN



Figura 9

### 3.5.2 TELA DE CADASTRO DE ASSOCIADOS

Relatórios Login Pagamentos **Associaados**

Pesquisar por

Associado

Código	Associado	Celular
1	Andreone	62995559496
2	Manoel	62929944567
3	Eduardo	62995643783
4	Miguel	62997647858

Dados do Associado

Código  RG  CPF  Data de Nascimento  EX: 01/01/2016

\* Nome  Sexo  Estado Civil

Endereço  Número  Complemento

Bairro  Cidade  Estado  CEP

Celular  \* Email

Figura 10

### 3.5.3 TELA DE CADASTRO DE LOGIN

Relatórios **Login** Associaados Pagamentos

Pesquisar por

Nome

Código	Login	Status
01	andreone	Ativo

Dados da Usuario

Código  Nome  Status

Senha

Figura 11

### 3.5.4 TELA DE RECEBER PAGAMENTOS

Relatórios Login Associados Pagamentos

Pesquisar por

Associado

Código	Associado	Celular
1	Andreone	62995559496
2	Manoel	62929944567
3	Eduardo	62995643783
4	Miguel	62997647858

Dados do Pagamento

Código  Inscrição  Mês de Referência

Associado  Data  EX: 01/01/2016

Mensalidade  Donativo  Carteira Social

Figura 12

### 3.5.5 TELA DE EMISSÃO DE RELATÓRIOS

Relatórios Login Associados Pagamentos

Dados do Relatório

Mês  Ano

Figura 13

## 4 CONCLUSÃO

O Software de Gerenciamento de Dados, para o Sindicato dos Trabalhadores Rurais de Taquaral de Goiás, será de grande relevância para o mesmo, pois irá auxiliar na administração da entidade de forma a organizar de forma segura os dados da instituição, guardando suas informações em um Banco de Dados, que poderá ser acessado pela entidade em uma plataforma Desktop.

O Software será criado com base no cadastramento de logins, associados, no recebimento de pagamentos e emissão de relatórios.

Com a Introdução desse software conclui-se que a entidade conseguirá acessar de forma rápida e segura todos os dados que estarão armazenados no programa, adquirindo um maior controle de todas as atividades exercidas pelo Sindicato.

O Software terá seu desenvolvimento continuado, afim de melhorar seu processo de organização, para que assim a empresa possa conseguir um controle ainda melhor de suas informações.

## REFERÊNCIAS

ASCENCIO, Ana, F. G. e CAMPOS, Edilene, A. V. (2014). **Fundamentos da Programação de Computadores: Algoritmos, Pascal e C++**. 3.ed. São Paulo: Editora Pearson.

APOSTILA CAELUM. **Java e Orientação a Objetos**. Disponível em: [https://www1.univap.br/bruno.pera/uploads/INFORMATICA/BANCODEDADOS/Apostila\\_de\\_Banco\\_de\\_Dados\\_V3.pdf](https://www1.univap.br/bruno.pera/uploads/INFORMATICA/BANCODEDADOS/Apostila_de_Banco_de_Dados_V3.pdf) Acesso em: 10-01-2022.

CADENHEAD, Rogers; LEMAY, Laura. **Aprenda em 21 dias Java 2**. 4. ed. São Paulo: Campus, 2005.

CLARO, Daniela Barreiro. SOBRAL, João Bosco Manguieira. **Programação em Java**. Florianópolis, 2008. Disponível em: <https://www.facterj-rio.edu.br/downloads/bbv/0031.pdf> Acesso em : 10-03-2022.

Damas, Luís, **SQL, Structured Query Language**. Luís Damas; tradução Elvira Maria Antunes Uchôa. - [Reimpr.]. - Rio de Janeiro: LTC, 2014.

DEVMEDIA. **4 Pilares da Programação Orientada a Objetos**. 2014. Disponível em: <https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264> Acesso em: 20-03-2022.

DEVMEDIA. **Implementando UML em java**. 2006. Disponível em: <https://www.devmedia.com.br/implementando-uml-em-java/4042> Acesso em: 20-03-2022.

DEVMEDIA. **Conceitos e exemplos – Herança: Programação Orientada a Objetos**. 2010. Disponível em: [devmedia.com.br/conceitos-e-exemplos-heranca-programacao-orientada-a-objetos-parte-1/18579](https://www.devmedia.com.br/conceitos-e-exemplos-heranca-programacao-orientada-a-objetos-parte-1/18579) Acesso em: 20-03-2022.

FRANÇA, Daniel. **Tecnologia nas empresas**. [S.l.]: 2013.

FERNANDES, TACILIANO AUGUSTO, MARTINS. ELBA FRANCIÉRIKA ROCHA. **Two Style: Um Software De Agendamento E Gerenciamento**. Rio Grande do Norte, 2019. Disponível em: <http://memoria.ifrn.edu.br/handle/1044/1941>. Acesso: 10-01-2022.

HENRIQUE, João. **Formação Java e Orientação a Objetos**. 2019. Disponível em: <https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos?gclid> Acesso em:20-03-2022.

NETO, Moacyr Franco. **Tutorial de Ferramenta de Modelagem**. 2017. Disponível em: <https://www.uaberta.unisul.br/sgc/downloadArquivoConteudo.processa?ead=1.502323005503715E121520391051183&arquivoId=41148&comunidadeId=44> Acesso em: 20-01-2022.

PRESSMAN, Roger S. **Engenharia de software – Uma Abordagem Profissional**. 7. Ed. Porto Alegre. AMGH Editora Ltda, 2011.

RICARTE, Ivan Luiz Marques. **Programação Orientada a Objetos: Uma Abordagem com Java**. Campinas, 2001. Disponível em: <https://www.dca.fee.unicamp.br/cursos/PooJava/Aulas/poojava.pdf> Acesso em: 20-01-2022.

SANTOS, Sérgio Leandro dos. **Concepção e Desenvolvimento de uma Interface Gráfica para Interação Tridimensional**. Porto Alegre, 2019. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/17631/000721396.pdf?sequence=1> Acesso em: 20-01-2022.

TANGANELLI, Claudio. **Desenvolvimento de Software para desktop, usando a plataforma jse, para uma aplicação de gerenciamento de aluguel de caçambas para entulhos**. Fundação Educacional do Municipal de Assis - FEMA - Assis, 2011.