INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E TECNOLOGIA DO AMAZONAS

Fernanda Serra e Vinícius Pessoa

GetFood();

Manaus, Amazonas - Brasil

2017

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E TECNOLOGIA DO AMAZONAS

Fernanda Serra e Vinícius Pessoa

GetFood();

**“Trabalho interdisciplinar apresentado à banca examinadora Curso Técnico em Informática do Instituto Federal de Educação, Ciências e Tecnologia do Amazonas – IFAM Campus Manaus - Centro, como requisito para o cumprimento de disciplinas”**

**Jucimar Brito**

03/11 /2017

Manaus, Amazonas

FOLHA DA FICHA CATALOGRÁFICA

**Serra, Fernanda**

**Pessoa, Vinícius**

**Título do Trabalho: GetFood(); / Fernanda Moreira Serra Teixeira, Vinícius Cavalcante de Brito Pessoa - Manaus: IFAM / Campus Manaus Centro, 2017.**

Orientador: Jucimar Brito

Projeto de conclusão de ensino técnico – Instituto Federal do Amazonas,

CMC, 2017.

Gerenciamento de fichas alimentícias.

Ciência da Computação

Brito, Jucimar

Instituto Federal do Amazonas, Campus Manaus Centro (CMC), Curso Técnico em Informática.

GetFood();.

FOLHA DE APROVAÇÃO

Este projeto interdisciplinar do Curso Técnico em Informática do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, Campus Manaus Centro, foi julgado e aprovado pela Banca Examinadora:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof° **TITULAÇÃO** Nome do **PROFESSOR COMPLETO**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof° **TITULAÇÃO** Nome do **PROFESSOR COMPLETO**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof° **TITULAÇÃO** Nome do **PROFESSOR COMPLETO**

AGRADECIMENTOS

# RESUMO

É fato que não existem registros de automatização do processo de controle e venda de fichas alimentícias do Instituto Federal do Amazonas - Campus Manaus Centro. A ausência de automatização deste processo, trouxe diversos aspectos negativos para a vida dos discentes, tais como as filas enormes geradas pela necessidade da coleta de confirmação de obtenção de ficha, atualmente feita pela assinatura do indivíduo.

Nota-se também, a fragilidade do processo de venda de fichas, devido a grande facilidade que o aluno possui de assinar um nome que não seja seu ou obter mais de uma ficha através da compra.

Entretanto, essa ausência de automatização dos processos citados anteriormente, não são prejudiciais apenas para os discentes, pois, os servidores que trabalham para a execução destes processos também são prejudicados, vez que, são responsáveis por contagem de assinaturas e geração de relatórios semanais, os quais, atualmente são feitos manualmente.

Observada esta situação, propõe-se neste projeto, o desenvolvimento de um sistema que irá executar de forma otimizada, os processos de venda e gerenciamento de distribuição, utilizando o cadastro dos alunos existentes no sistema acadêmico.

Para a otimização do processo de venda, é usado a matrícula e foto do respectivo aluno como mecanismo de controle de acesso. Os mecanismos de controle de acesso estão vinculados a verificação de autenticidade, geralmente feito através de logins e senhas, porém, neste caso o usuário(servidor) irá usar o recurso visual da foto para certificar-se da identidade do aluno.

E para a otimização dos processos de gerenciamento de distribuição, usamos ferramentas que possibilitam a geração de relatórios automáticos, as quais, fazem com que a contagem de assinaturas manual seja desnecessária.

**Palavras-Chave:Automatização,fichas,alimentícias,otimização, processo, autenticação.**

# ABSTRACT

It’s a fact that there aren’t registers of automatization of the process of control and sale of food tickets in the Instituto Federal do Amazonas campus Manaus Centro. The absence of the automatization of this process brought some negative aspects to the lives of the students, such as huge lines to being able to confirm that they are in possession of the tickets, currently made by the signature of the individual.

It’s also noticeable the fragility of the process of selling tickets, due to the easiness that the student possess of signing a name that is not his or obtain more than one ticket through a purchase.

However, this absence of automation of the process mentioned above is not only prejudicial to the students, the servers that work to execute theses processes are also impaired because they are responsible for counting of signatures and generations of weekly reports, which are currently done manually.

Observing the situation, it’s proposed in this project, the development of a system that will optimally execute the sales and distribution management processes, using the existing student’s register in the academic system.

To optimize the sales process, the student’s registration and photo are used as the access control mechanism. The access control mechanisms are linked to authenticity checking, usually done through logins and passwords, however, in this case the user(server) will use the visual appeal of the photo to confirm the student’s identity. And for the optimization of distribution management processes, we use tools that enable automatic reporting, which makes manual signature counting unnecessary.

**Keywords:Automatization, tickets, food, optimization, process, authentication.**

LISTA DE FIGURAS

[Figura 1. Diagrama de Caso de Uso. 19](#_41mghml)

[Figura 2. Diagrama de Classe. 33](#_2dlolyb)

[Figura 3. Modelo Entidade-Relacionamento. 34](#_4bvk7pj)

[Figura 4. Diagrama de Sequência - Cadastro de Alunos. 36](#_kgcv8k)

[Figura 5. Diagrama de Sequência - Cadastro de Funcionário. 37](#_1jlao46)

[Figura 6. Diagrama de Sequência - Gerar Relatórios. 37](#_2iq8gzs)

[Figura 7. Diagrama de Sequência - Vender. 38](#_3hv69ve)

[Figura 8. Diagrama de Sequência - Resetar Banco de Dados. 38](#_4h042r0)

[Figura 9. Projeto de Telas - Cadastrar Alunos. 39](#_3vac5uf)

[Figura 10. Projeto de Telas - Cadastrar Funcionário. 39](#_pkwqa1)

[Figura 11. Projeto de Telas - Cadastrar Preço da Ficha. 40](#_1opuj5n)

[Figura 12. Projeto de Telas - Executar Venda. 40](#_2nusc19)

[Figura 13. Projeto de Telas - Gerar Relatório. 41](#_3mzq4wv)

[Figura 14. Projeto de Telas - Venda. 41](#_haapch)

[Figura 15. Modelo Relacional. 42](#_1gf8i83)

[Figura 16. Esquema do Banco de Dados. 43](#_3s49zyc)

# LISTA DE TABELAS

[Tabela 1. Documentação de Caso de Uso - Cadastrar Listagem de Turmas. 20](#_vx1227)

[Tabela 2. Documentação de Caso de Uso - Cadastrar Listagem de Alunos. 21](#_1v1yuxt)

[Tabela 3. Documentação de Caso de Uso - Gerar Relatório. 22](#_2u6wntf)

[Tabela 4. Documentação de Caso de Uso - Vender Fichas de Refeição. 24](#_3tbugp1)

[Tabela 5. Documentação de Caso de Uso - Vender Créditos. 25](#_nmf14n)

[Tabela 6. Documentação de Caso de Uso - Resetar Banco de Dados. 26](#_1mrcu09)

[Tabela 7. Documentação de Caso de Uso - Cadastro de Login. 27](#_2lwamvv)

[Tabela 8. Documentação de Caso de Uso - Cadastro de Listagem de Alunos Beneficiários. 28](#_3l18frh)

[Tabela 9. Documentação de Caso de Uso - Cadastro de Funcionário. 30](#_4k668n3)

[Tabela 10. Documentação de Caso de Uso - Cadastrar Listagem de Cursos. 31](#_1egqt2p)

[Tabela 11. Dicionário de Dados - Login. 42](#_2fk6b3p)

[Tabela 12. Dicionário de Dados - Aluno. 42](#_upglbi)

[Tabela 13. Dicionário de Dados - Funcionário. 43](#_3ep43zb)

[Tabela 14. Dicionário de Dados - Curso. 43](#_1tuee74)

[Tabela 15. Dicionário de Dados - Ficha. 43](#_4du1wux)

[Tabela 16. Dicionário de Dados - Venda. 44](#_2szc72q)

# LISTA DE ABREVIATURAS E SIGLAS

UML *Unified Modeling Language*

ERD *Entity Relationship Diagram*

DFD Diagrama de Fluxo de Dados

SysML *System Modeling Language*

GSN *Goal Structuring Notation*

BDAS *Database Administrators*

AWT *Abstract Window Toolkit*

SGBD Sistema gerenciador de banco de dados

DAO *Data Access Object*

# 

SUMÁRIO

[**RESUMO**](#_gjdgxs)5

[**ABSTRACT**](#_30j0zll)6

[**LISTA DE FIGURAS**](#_1fob9te) **Erro! Indicador não definido.**

[**LISTA DE TABELAS**](#_2et92p0)8

[**LISTA DE ABREVIATURAS E SIGLAS**](#_3dy6vkm)9

[**CAPÍTULO 1 - INTRODUÇÃO**](#_17dp8vu)12

[1.1 - INTRODUÇÃO](#_3rdcrjn) **Erro! Indicador não definido.**

[1.2 - PROBLEMATIZAÇÃO](#_26in1rg) 12

[1.3 - JUSTIFICATIVA](#_lnxbz9) 12

[1.4 - OBJETIVOS](#_35nkun2) 12

[1.4.1 - OBJETIVO GERAL](#_1ksv4uv) 12

[1.4.2 - OBJETIVOS ESPECÍFICOS](#_44sinio) 12

[1.5 - METODOLOGIA](#_2jxsxqh) 13

[**CAPÍTULO 2 - FERRAMENTAS DE SOFTWARE UTILIZADAS**](#_z337ya)14

[2.1 - INTRODUÇÃO](#_3j2qqm3) 14

[2.2 - ASTAH](#_1y810tw) **Erro! Indicador não definido.**

[2.3 - DRAW.IO](#_4i7ojhp) **Erro! Indicador não definido.**

[2.4 - JAVA](#_2xcytpi) **Erro! Indicador não definido.**

[2.5 - JAVA SWING](#_1ci93xb) **Erro! Indicador não definido.**

[2.6 - NETBEANS](#_3whwml4) **Erro! Indicador não definido.**

[2.7 - SQL](#_2bn6wsx) **Erro! Indicador não definido.**

[2.8 - MYSQL WORKBENCH](#_qsh70q) **Erro! Indicador não definido.**

[2.9 - MYSQL](#_3as4poj) **Erro! Indicador não definido.**

[2.10 - JASPER REPORT](#_1pxezwc) **Erro! Indicador não definido.**

[2.11 - iReport](#_49x2ik5) **Erro! Indicador não definido.**

[**CAPÍTULO 3 - ANÁLISE E MODELAGEM DO SISTEMA**](#_2p2csry)17

[3.1 - INTRODUÇÃO](#_147n2zr) 17

[3.2 - DESCRIÇÃO DO CENÁRIO OU MINI - MUNDO](#_3o7alnk) 17

[3.3 - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS](#_23ckvvd) 17

[3.3.1 - REQUISITOS FUNCIONAIS](#_ihv636) 17

[3.3.2 - REQUISITOS NÃO-FUNCIONAIS](#_32hioqz) 18

[3.4 - CASOS DE USO](#_1hmsyys) 18

[3.4.1 Cadastrar Listagem de Turmas](#_2grqrue) 19

[3.4.2 Cadastrar Listagem de Alunos](#_3fwokq0) 20

[3.4.3 Gerar Relatório](#_4f1mdlm) 22

[3.4.4 Vender Fichas de Refeição](#_19c6y18) 23

[3.4.5 Vender Créditos](#_28h4qwu) 24

[3.4.7 Resetar Banco de Dados](#_37m2jsg) 26

[3.4.8 Cadastro de Login](#_46r0co2) 27

[3.4.9 Cadastro de Listagem de alunos beneficiários](#_111kx3o) 28

[3.4.10 Cadastro de Funcionário](#_206ipza) 29

[3.4.11 Cadastrar Listagem de Cursos](#_2zbgiuw) 30

[3.5 - DIAGRAMA DE CLASSE](#_3ygebqi) 34

[3.6 - MODELO ENTIDADE-RELACIONAMENTO](#_1rvwp1q) 35

[**CAPÍTULO 4 - PROJETO E DESENVOLVIMENTO**](#_2r0uhxc)36

[4.1 - INTRODUÇÃO](#_1664s55) 36

[4.2 - DIAGRAMA DE SEQUÊNCIA](#_3q5sasy) 37

[4.2.1 - Cadastro de listagem de alunos](#_25b2l0r) 37

[4.2.2 - Cadastro de Funcionário](#_34g0dwd) 38

[4.2.3 - Gerar Relatórios](#_43ky6rz) 38

[4.2.4 - Vender](#_xvir7l) 39

[4.2.5 - Resetar Banco de Dados](#_1x0gk37) 40

[4.3 - PROJETO DE TELAS](#_2w5ecyt) 41

[4.3.1 - Cadastrar Alunos](#_1baon6m) 41

[4.3.2 - Cadastrar Funcionário](#_2afmg28) 41

[4.3.3 - Cadastrar Preço da Ficha](#_39kk8xu) 42

[4.3.4 - Executar Venda](#_48pi1tg) 42

[4.3.5 - Gerar Relatório](#_1302m92) 43

[4.3.6 - Venda](#_2250f4o) 43

[4.4 - MODELO RELACIONAL](#_319y80a) 44

[4.5 - DICIONÁRIO DE DADOS](#_40ew0vw) 44

[4.5.1 - Login](#_lsvqsgjmne51) 45

[4.5.2 - Aluno](#_nft22nyxykgr) 45

[4.5.3 - Funcionário](#_x756y4dr8231) 46

[4.5.4 - Curso](#_ynk9vbiarvwu) 46

[4.5.5 - Turma](#_5q98p4ipw9iz) 46

[4.5.6 - Ficha](#_wvx4z1bmqr8e) 47

[4.5.7 - Venda](#_sr3kzy5id3q3) 47

[4.6 - ESQUEMA DO BANCO DE DADOS (BD)](#_184mhaj) 48

[4.7 - LISTAGEM DE CLASSES](#_meukdy) 49

[**CAPÍTULO 5 - CONCLUSÕES OU CONSIDERAÇÕES**](#_36ei31r)50

[5.1 - INTRODUÇÃO](#_1ljsd9k) 50

[5.2 - CONCLUSÕES](#_45jfvxd) 50

[5.3 - DIRECIONAMENTO PARA TRABALHOS FUTUROS](#_2koq656) 50

[5.4 - REFERÊNCIAS](#_zu0gcz) 51

[5.5 - ANEXO(S)](#_3jtnz0s) 52

[5.5.1 - QUESTIONÁRIO DE ELICITAÇÃO DE DADOS](#_1yyy98l) 52

# 

# CAPÍTULO 1 - INTRODUÇÃO

## 1.1 - INTRODUÇÃO

Este projeto tem como intuito facilitar a vida de discentes e servidores dentro do Instituto Federal do Amazonas, Campus Manaus Centro, através da automatização de processos.

## 1.2 - PROBLEMATIZAÇÃO

Os processos necessários para o gerenciamento de fichas alimentícias do Instituto Federal do Amazonas nunca foram automatizados, sendo assim, os mesmos são feitos manualmente através de listas extensas que contêm informações dos seus respectivos alunos.

Atualmente, os servidores que trabalham no âmbito do refeitório, são obrigados a contar centenas de assinaturas manualmente todos os dias para o gerenciamento e futura geração de relatórios sobre as vendas e distribuições de fichas.

## 1.3 - JUSTIFICATIVA

A execução deste projeto foi motivado pela existência de problema que tornou-se habitual na vida dos discentes: a espera para compra de fichas e longas filas durante o curto horário do almoço.

Outro problema que será resolvido é a falta de troco, é comum que no ínicio da semana os alunos possuam notas altas, pois recebem dinheiro para ser usado durante a semana, nossa solução é a disponibilização de compra de créditos(pacotes semanais), ou seja, o aluno poderia deixar pago as refeições da semana ou do mês.

## 1.4 - OBJETIVOS

### 1.4.1 - OBJETIVO GERAL

Automatizar o gerenciamento de fichas alimentícias do Instituto Federal do Amazonas dando recursos para a compra de fichas e controle deste processo com segurança.

### 1.4.2 - OBJETIVOS ESPECÍFICOS

* Administrar os créditos do aluno;
* Cadastrar fotos dos alunos;
* Controlar venda de fichas;
* Otimizar a emissão de relatórios de controle das refeições.

## 1.5 - METODOLOGIA

Para que o desenvolvimento deste software obtivessem os resultados esperados, foi usada a seguinte metodologia :

* Definição de cronograma;
* Estruturação do trabalho;
* Estudo das ferramentas usadas na implementação;
* Orientação feita em curtos períodos de tempo.

# CAPÍTULO 2 - FERRAMENTAS DE SOFTWARE UTILIZADAS

## 2.1 - INTRODUÇÃO

Este capítulo é destinado para a citação e posteriormente explicação das ferramentas e tecnologias utilizadas para o desenvolvimento do sistema apresentado.

## 2.2 - ASTAH

Astah é uma ferramenta de design de software que suporta UML, ERD, DFD, Flowchart, Mind Map, SysML, GSN e muito mais.(Astah[[1]](#footnote-1))

## 2.3 - DRAW.IO

Draw.io é um editor de diagramas on-line completamente gratuito construído em torno do Google Drive, que permite que você crie fluxogramas, UML, relação de entidade, diagramas de rede, etc.(Draw.io[[2]](#footnote-2))

## 2.4 - JAVA

Como qualquer linguagem de programação, a linguagem Java tem sua própria estrutura, regras de sintaxe e paradigma de programação. O paradigma de programação da linguagem Java baseia-se no conceito de OOP, que os recursos da linguagem suportam. [...]

Estruturalmente, a linguagem Java começa com *pacotes*. Um pacote é o mecanismo de namespace da linguagem Java. Dentro dos pacotes estão as classes e dentro das classes estão métodos, variáveis, constantes e mais. (IBM[[3]](#footnote-3))

## 2.5 - JAVA SWING

O [Swing é um framework](http://www.devmedia.com.br/curso/desenvolvendo-uma-aplicacao-completa-utilizando-swing/187) que disponibiliza um conjunto de elementos gráficos para ser utilizado na plataforma Java. O Swing é compatível com o [Abstract Window Toolkit (AWT)](http://www.devmedia.com.br/classes-awt-para-criacao-de-jogos-curso-manipulacao-de-imagens-com-java-parte-6/17870), mas trabalha de forma totalmente diferente. A API Swing, diferente do AWT, não delega a tarefa de renderização ao sistema operacional, ele renderiza os elementos por conta própria. [...] (DevMedia[[4]](#footnote-4))

## 2.6 - NETBEANS

O NetBeans IDE permite que você desenvolva rápida e facilmente aplicativos de desktop, móveis e da Web Java, bem como aplicativos HTML5 com HTML, JavaScript e CSS. O IDE também fornece um ótimo conjunto de ferramentas para desenvolvedores de PHP e C / C ++. É gratuito e de código aberto e tem uma grande comunidade de usuários e desenvolvedores em todo o mundo.(NetBeans[[5]](#footnote-5))

## 2.7 - SQL

É uma linguagem para geração, manipulação e recuperação de dados de um banco de dados relacional. (Aprendendo SQL, Alan Beaulieu)

## 2.8 - MYSQL WORKBENCH

MySQL Workbench é uma ferramenta visual unificada para arquitetos de banco de dados, desenvolvedores e DBAs. O MySQL Workbench fornece modelagem de dados, desenvolvimento de SQL e ferramentas de administração abrangentes para configuração de servidor, administração de usuários, backup e muito mais. O MySQL Workbench está disponível no Windows, Linux e Mac OS X.(MySQL[[6]](#footnote-6))

## 2.9 - MYSQL

MySQL é um SGBD que inicialmente foi desenvolvido para trabalhar com projetos de pequeno e médio porte, com a capacidade de suportar por volta de cem milhões de registros em cada tabela, podendo chegar ao tamanho médio de aproximadamente cem megabytes por tabela, entretanto, esses eram os tamanhos recomendados nas primeiras versões. (DevMedia[[7]](#footnote-7))

## 2.10 - JASPER REPORT

A biblioteca JasperReports é o mecanismo de geração de relatórios de código aberto mais popular do mundo. É inteiramente escrito em Java e é capaz de usar dados provenientes de qualquer tipo de fonte de dados e produzir documentos pixel-perfect que podem ser vistos, impressos ou exportados em vários formatos de documentos, incluindo HTML, PDF, Excel, OpenOffice e Word.(Jaspersoft Community[[8]](#footnote-8))

## 2.11 - iReport

O iReport é o designer gratuito de relatórios de código aberto para JasperReports e JasperReports Server. Crie layouts muito sofisticados contendo gráficos, imagens, sub-relatórios, tabelas cruzadas e muito mais. Acesse seus dados através de JDBC, TableModels, JavaBeans, XML, Hibernate, CSV e fontes personalizadas. Em seguida, publique seus relatórios como PDF, RTF, XML, XLS, CSV, HTML, XHTML, texto, DOCX ou OpenOffice.(Jaspersoft Community[[9]](#footnote-9))

# CAPÍTULO 3 - ANÁLISE E MODELAGEM DO SISTEMA

## 3.1 - INTRODUÇÃO

Para que a implementação do software seja realizada, existe uma atividade a qual precisa ser feita,e que está entre as mais importantes das práticas de engenharia de software: a elicitação de requisitos, onde busca-se o entendimento das necessidades do usuário, que ao desenvolvermos este projeto, foi utilizada a abordagem da entrevista, tendo feito um questionário e devidamente respondido pelos funcionários ,julgados necessários.

Para a implementação deste projeto, foi usado o ciclo de vida clássico, o qual, consiste em uma abordagem sistemática, avançando sequencialmente etapa por etapa, para a obtenção do desenvolvimento de software.

## 3.2 - DESCRIÇÃO DO CENÁRIO OU MINI - MUNDO

Uma instituição dispõe de um refeitório, o qual, fornece refeições para os seus respectivos alunos, porém, antes de receberem a refeição, os alunos devem comprar fichas para que os funcionários do refeitório forneçam sua refeição.

Neste refeitório trabalham diversos funcionários tais como: nutricionista, cozinheiro e o vendedor, os quais serão descritos por seus nomes, suas matrículas e seus respectivos cargo e, se o funcionário tiver contato direto com o sistema obrigatoriamente ele terá um login e senha.

Um aluno possui nome, matrícula, turma, curso e saldo, e pode ser beneficiário ou não, a única diferença entre os dois será o tipo de venda executado, pois os alunos beneficiários não precisam comprar saldo ou fichas, já que o saldo para a compra das mesmas são providos pelo governo.

A venda é descrita por data, tipo, matrícula do aluno, matrícula do funcionário e seu respectivo valor, podendo ser venda de saldo(créditos), fichas para alunos beneficiários ou alunos pagantes.

## 3.3 - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS

### 3.3.1 - REQUISITOS FUNCIONAIS

3.3.1.1 - Alterar preço da ficha alimentícia;

3.3.1.2 - Cadastro de funcionário;

3.3.1.3 - Cadastro de login;

3.3.1.4 - Cadastrar imagem do aluno;

3.3.1.5 - Cadastrar listagem de alunos beneficiários;

3.3.1.6 - Cadastrar listagem de cursos;

3.3.1.7 - Cadastrar listagem de turmas;

3.3.1.8 - Cadastrar preço da ficha alimentícia;

3.3.1.9 - Cadastrar listagem de alunos;

3.3.1.10 - Gerar relatório;

3.3.1.11 - Vender crédito ao aluno;

3.3.1.12 - Vender ficha alimentícia ao aluno;

3.3.1.13 - Resetar banco de dados.

### 3.3.2 - REQUISITOS NÃO-FUNCIONAIS

3.3.2.1 - Baixo tempo de resposta;

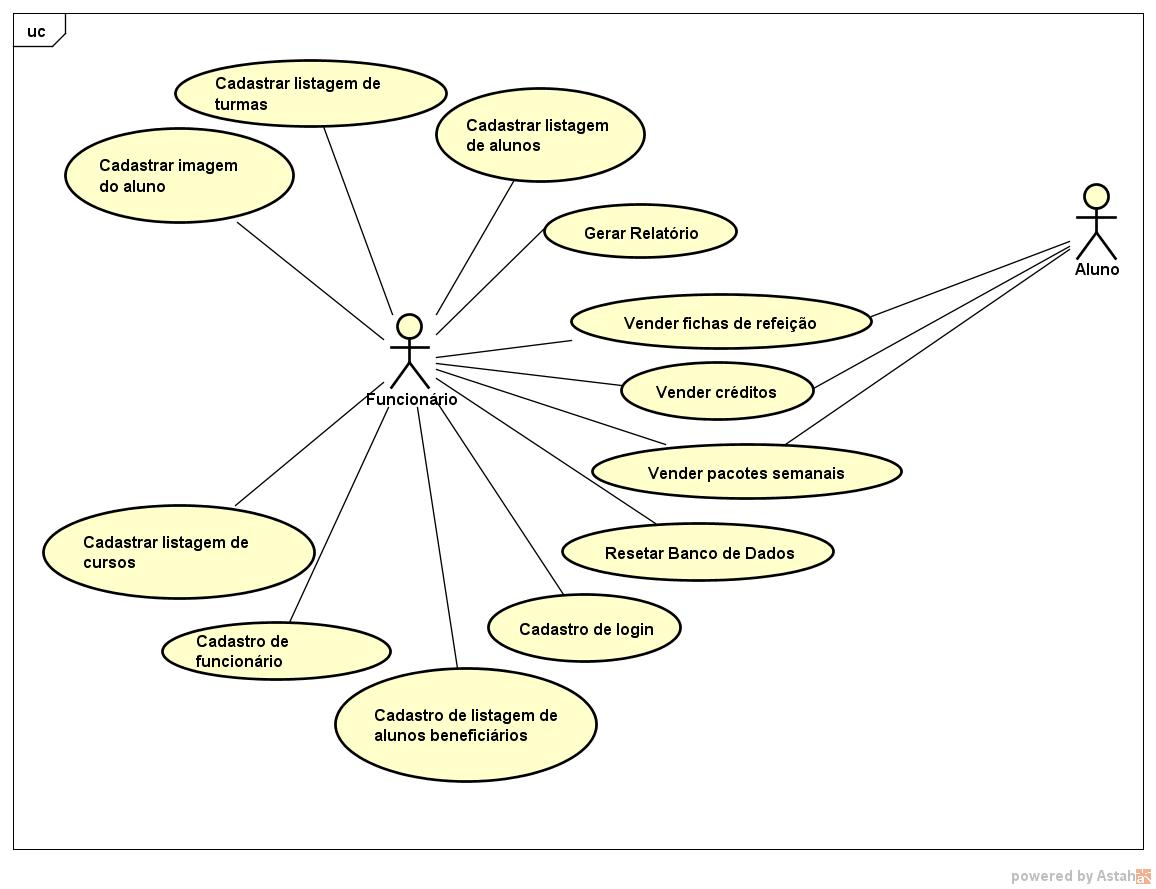
3.3.2.2 - Boa usabilidade;

3.3.2.3 - Compatibilidade com as plataformas Windows e Linux;

3.3.2.4 - Interface de fácil compreensão e navegação;

3.3.2.5 - Tolerância a falhas.

## 3.4 - CASOS DE USO

*Figura 1. Diagrama de Caso de Uso.*

### 3.4.1 Cadastrar Listagem de Turmas

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Cadastrar Listagem de Turmas** |
| Caso de Uso Geral | - |
| Ator Principal | Funcinário |
| Atores Secundários | - |
| Resumo | Este caso de uso descreve as etapas necessárias para que o funcionário cadastre as turmas do instituto no sistema usando um arquivo de extensão “.txt”. |
| Pré-Condições | O funcionário deve estar logado. |
| Pós-Condições | - |
| **Fluxo Principal** | |
| Ações do Ator | Ações do Sistema |
| 1. Entrar na tela de cadastro de turmas. |  |
|  | 1. Carregar tela com o botão para selecionar o arquivo “.txt”. |
| 1. Selecionar o arquivo .txt que contenha as turmas a serem cadastradas. |  |
|  | 1. Mostrar os dados do arquivo selecionado. |
| 5. Clicar no botão “Salvar” para salvar as turmas ao banco de dados. |  |
|  | 1. Salvar os dados do arquivo no banco de dados. |
|  | 1. Perguntar se o funcionário deseja ir para o menu principal ou continuar na tela atual. |
| 1. Escolher opção. |  |
|  | 1. Executar opção escolhida pelo funcionário. |
| **Fluxo Alternativo** | |
| Ações do Ator | Ações do Sistema |
| 1. Clicar no botão para retornar ao menu principal. |  |
|  | 1. Retornar ao menu principal. |
| **Fluxo de Exceção** | |
| Ações do Ator | Ações do Sistema |
| 1. Clicar no botão “Salvar”. |  |
|  | 1. Identificar que não há arquivo selecionado. |
|  | 1. Mostrar aviso para que o funcionário selecione um arquivo. |

Tabela 1. Documentação de Caso de Uso - Cadastrar Listagem de Turmas.

### 3.4.2 Cadastrar Listagem de Alunos

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Cadastrar Listagem de Alunos** |
| Caso de Uso Geral | - |
| Ator Principal | Funcionário |
| Atores Secundários | - |
| Resumo | Este caso de uso descreve as etapas necessárias para que o funcionário cadastre, no sistema, os alunos do instituto por meio do uso de um arquivo de extensão “.txt”. |
| Pré-Condições | O funcionário deve estar logado no sistema. |
| Pós-Condições | - |
| **Fluxo Principal** | |
| Ações do Ator | Ações do Sistema |
| 1. Entrar na tela de cadastro de alunos. |  |
|  | 1. Carregar tela de cadastramento com botão para “Escolher Arquivo” e “Salvar”. |
| 1. Clicar no botão “Escolher Arquivo” e selecionar o arquivo que contenha os alunos a serem cadastrados. |  |
|  | 1. Mostrar os dados do arquivo que o funcionário selecionou. |
| 1. Clicar no botão “Salvar” para salvar os dados dos alunos no banco de dados. |  |
|  | 1. Salvar os dados dos alunos no banco de dados. |
|  | 1. Perguntar se o funcionário deseja ir para o menu principal ou continuar na tela atual. |
| 1. Escolher opção. |  |
|  | 1. Executar opção escolhida pelo funcionário. |
| **Fluxo Alternativo** | |
| Ações do Ator | Ações do Sistema |
| 1. Clicar no botão de voltar ao menu principal. |  |
|  | 1. Voltar ao menu principal. |
| **Fluxo de Exceção** | |
| Ações do Ator | Ações do Sistema |
| 1. Clicar no botão “Salvar”. |  |
|  | 1. Identificar que não há arquivo selecionado. |
|  | 1. Mostrar aviso para que o funcionário selecione um arquivo. |

*Tabela 2. Documentação de Caso de Uso - Cadastrar Listagem de Alunos.*

### 3.4.3 Gerar Relatório

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Gerar Relatório** |
| Caso de Uso Geral | - |
| Ator Principal | Funcionário |
| Atores Secundários | - |
| Resumo | Este caso de uso descreve as etapas necessárias para que o funcionário gere um relatório. |
| Pré-Condições |  |
| Pós-Condições |  |
| **Fluxo Principal** | |
| Ações do Ator | Ações do Sistema |
| 1. Abre a tela de geração de relatório. |  |
|  | 1. Carregar a tela de geração de relatório. |
| 1. Inserir dados necessários (Data inicial, data final e se é um relatório dos beneficiários). |  |
|  | 1. Ler dados inseridos pelo usuário. |
|  | 1. Fazer a recuperação dos dados no banco de dados. |
|  | 1. Montar o relatório. |
|  | 1. Mostrar uma nova janela ao funcionário com o relatório pronto. Na janela, o usuário tem a opção de imprimir ou salvar o relatório. |
| **Fluxo Alternativo** | |
| 1. Funcionário clicar no botão para voltar ao menu principal. |  |
|  | 1. Fechar a tela atual e abrir a tela do menu principal. |
| **Fluxo de Exceção** | |
| Ações do Ator | Ações do Sistema |
| 1. Funcionário não digitar as datas |  |
|  | 1. Avisar ao funcionário que ele deve digitar as datas e impedir que o procedimento continue. |

*Tabela 3. Documentação de Caso de Uso - Gerar Relatório.*

### 3.4.4 Vender Fichas de Refeição

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Vender Fichas de Refeição** |
| Caso de Uso Geral | - |
| Ator Principal | Funcionário |
| Atores Secundários | Aluno |
| Resumo | Este caso de uso descreve as etapas necessárias para que o funcionário realize a venda de fichas de refeição ao aluno. |
| Pré-Condições | O funcionário deve estar logado no sistema e o aluno deve fornecer sua matrícula ao funcionário. |
| Pós-Condições | - |
| **Fluxo Principal** | |
| Ações do Ator | Ações do Sistema |
| 1. Abrir a tela de venda de fichas. |  |
|  | 1. Carregar a tela de venda de fichas com um campo e um botão para verificar a matrícula do aluno. |
| 1. Digitar a matrícula do aluno no campo e clicar no botão “Verificar”. |  |
|  | 1. Recuperar no banco de dados o aluno correspondente a matrícula fornecida pelo funcionário. |
|  | 1. Mostrar os dados e a foto do aluno e disponibilizar os métodos de pagamento. |
| 1. Escolher o método de pagamento e clicar em “Vender” |  |
|  | 1. Executar a venda e mostrar o resultado ao funcionário. |
|  | 1. Perguntar se o funcionário deseja ir para o menu principal ou permanecer da tela atual. |
| 1. Selecionar opção. |  |
|  | 1. Realizar operação selecionada pelo funcionário. |
| **Fluxo Alternativo** | |
| Ações do Ator | Ações do Sistema |
| 1. Clicar no botão de voltar ao menu principal. |  |
|  | 1. Voltar à tela do menu principal |
| **Fluxo de Exceção** | |
| Ações do Ator | Ações do Sistema |
|  | 1. Caso a matrícula informada não pertença a nenhum dos alunos cadastrados, informar ao funcionário e impedir o procedimento da venda. |

*Tabela 4. Documentação de Caso de Uso - Vender Fichas de Refeição.*

### 3.4.5 Vender Créditos

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Vender Créditos** |
| Caso de Uso Geral | - |
| Ator Principal | Funcionário |
| Atores Secundários | Aluno |
| Resumo | Este caso de uso descreve as etapas necessárias para que o funcionário realize a venda de fichas de refeição ao aluno. |
| Pré-Condições | O funcionário deve estar logado no sistema e o aluno deve fornecer a matrícula ao funcionário. |
| Pós-Condições | - |
| **Fluxo Principal** | |
| Ações do Ator | Ações do Sistema |
| 1. Entrar na tela de venda de créditos. |  |
|  | 1. Carregar tela de venda de créditos. |
| 1. Digitar a matrícula do aluno e clicar no botão “Verificar”. |  |
|  | 1. Verificar se a matrícula digitada pertence a algum aluno no banco de dados. |
|  | 1. Desabilitar a edição o campo de matrícula, habilitar o campo da quantia e mostrar os dados do aluno na tela. |
| 1. Digitar a quantia (em reais) a ser vendida. |  |
|  | 1. Realizar a venda e adicionar a quantia de créditos comprados no saldo do aluno. |
|  | 1. Perguntar se o funcionário deseja ir para o menu principal ou permanecer da tela atual. |
| 1. Selecionar opção. |  |
|  | 1. Realizar operação selecionada pelo funcionário. |
| **Fluxo Alternativo** | |
| Ações do Ator | Ações do Sistema |
| 1. Clicar no botão de retornar ao menu principal |  |
|  | 1. Retornar ao menu principal. |
| **Fluxo de Exceção** | |
| Ações do Ator | Ações do Sistema |
|  | 1. Se a matrícula não corresponder à de um aluno no banco de dados, informar ao usuário e impedir a conclusão da venda. |

*Tabela 5. Documentação de Caso de Uso - Vender Créditos.*

### 3.4.7 Resetar Banco de Dados

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Resetar Banco de Dados** |
| Caso de Uso Geral | - |
| Ator Principal | Funcionário |
| Atores Secundários | - |
| Resumo | Este caso de uso descreve as etapas necessárias para que o funcionário realize a venda de fichas de refeição ao aluno. |
| Pré-Condições | O funcionário deve estar logado no sistema. |
| Pós-Condições | - |
| **Fluxo Principal** | |
| Ações do Ator | Ações do Sistema |
| 1. Abrir a tela de reset. |  |
|  | 1. Carregar tela de reset. |
| 1. Clicar no botão “Apagar Dados” |  |
|  | 1. Apagar alunos, vendas, cursos e turmas do banco de dados. |
| **Fluxo Alternativo** | |
| Ações do Ator | Ações do Sistema |
| 1. Clicar no botão para retornar ao menu principal. |  |
|  | 1. Retornar ao menu principal. |
| **Fluxo de Exceção** | |
| Ações do Ator | Ações do Sistema |
| - | - |

*Tabela 6. Documentação de Caso de Uso - Resetar Banco de Dados.*

### 3.4.8 Cadastro de Login

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Cadastro de Login** |
| Caso de Uso Geral | - |
| Ator Principal | Funcionário |
| Atores Secundários | - |
| Resumo | Este caso de uso descreve as etapas necessárias para que o funcionário realize o cadastro do login de outro funcionário. |
| Pré-Condições | O funcionário deve estar logado |
| Pós-Condições | - |
| **Fluxo Principal** | |
| Ações do Ator | Ações do Sistema |
| 1. Abrir a tela de cadastro de login. |  |
|  | 1. Carregar a tela de cadastro de login como o formulário. |
| 1. Preencher o formulário de cadastro e clicar no botão “Cadastrar”. |  |
|  | 1. Verificar os campos e realizar o cadastro. |
|  | 1. Perguntar se o funcionário deseja ir para o menu principal ou resetar a tela atual. |
| 1. Selecionar opção. |  |
|  | 1. Realizar operação selecionada pelo funcionário. |
| **Fluxo Alternativo** | |
| Ações do Ator | Ações do Sistema |
| 1. Clicar no botão para voltar ao menu principal. |  |
|  | 1. Voltar ao menu principal. |
| **Fluxo de Exceção** | |
| Ações do Ator | Ações do Sistema |
|  | 1. Caso algum campo não seja preenchido ou tenha um valor inválido, avisar o funcionário e impedir o cadastramento. |

*Tabela 7. Documentação de Caso de Uso - Cadastro de Login.*

### 3.4.9 Cadastro de Listagem de alunos beneficiários

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Cadastro de Listagem de Alunos Beneficiários** |
| Caso de Uso Geral | - |
| Ator Principal | Funcionário |
| Atores Secundários | - |
| Resumo | Este caso de uso descreve as etapas necessárias para que o funcionário realize o cadastro do login de outro funcionário. |
| Pré-Condições | O funcionário deve estar logado no sistema. |
| Pós-Condições | - |
| **Fluxo Principal** | |
| Ações do Ator | Ações do Sistema |
| 1. Abrir tela de cadastro de listagem de alunos beneficiários. |  |
|  | 1. Carregar tela de cadastro de listagem de alunos beneficiários. |
| 1. Escolher arquivo que contenha os dados dos alunos que se deseja cadastrar. |  |
|  | 1. Carregar arquivo e mostrar os dados contidos nele na tela. |
| 1. Clicar no botão “Salvar”. |  |
|  | 1. Salvar os alunos no banco de dados e informar o resultado na tela. |
|  | 1. Perguntar se o funcionário deseja ir para o menu principal ou resetar a tela atual. |
| 1. Selecionar opção |  |
|  | 1. Realizar a operação selecionada pelo usuário. |
| **Fluxo Alternativo** | |
| Ações do Ator | Ações do Sistema |
| 1. Clicar no botão de voltar ao menu principal. |  |
|  | 1. Voltar ao menu principal. |
| **Fluxo de Exceção** | |
| Ações do Ator | Ações do Sistema |
|  | 1. Caso nenhum arquivo tenha sido selecionado, informar o funcionário e impedir o cadastramento. |

*Tabela 8. Documentação de Caso de Uso - Cadastro de Listagem de Alunos Beneficiários.*

### 3.4.10 Cadastro de Funcionário

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Cadastro de Funcionário** |
| Caso de Uso Geral | - |
| Ator Principal | Funcionário |
| Atores Secundários | - |
| Resumo | Este caso de uso descreve as etapas necessárias para que o funcionário realize o cadastro de outro funcionário. |
| Pré-Condições | O funcionário deve estar logado no sistema. |
| Pós-Condições | - |
| **Fluxo Principal** | |
| Ações do Ator | Ações do Sistema |
| 1. Abrir a tela de cadastro de funcionário. |  |
|  | 1. Carregar tela de cadastro de funcionário com formulário de cadastramento. |
| 1. Preencher o formulário e clicar no botão “Cadastrar”. |  |
|  | 1. Verificar se algum campo não foi preenchido e realizar o cadastro. |
|  | 1. Perguntar se o funcionário deseja ir para o menu principal ou resetar a tela atual. |
| 1. Escolher opção. |  |
|  | 1. Realizar operação escolhida pelo funcionário. |
| **Fluxo Alternativo** | |
| Ações do Ator | Ações do Sistema |
| 1. Clicar no botão de voltar ao menu principal. |  |
|  | 1. Voltar ao menu principal. |
| **Fluxo de Exceção** | |
| Ações do Ator | Ações do Sistema |
|  | 1. Caso algum campo do formulário esteja em branco, avisar ao funcionário e impedir o cadastramento. |

*Tabela 9. Documentação de Caso de Uso - Cadastro de Funcionário.*

### 3.4.11 Cadastrar Listagem de Cursos

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Cadastrar Listagem de Cursos** |
| Caso de Uso Geral | - |
| Ator Principal | Funcionário |
| Atores Secundários | - |
| Resumo | Este caso de uso descreve as etapas necessárias para que o funcionário realize os cadastro de cursos contidos em um arquivo. |
| Pré-Condições | O funcionário deve estar logado no sistema. |
| Pós-Condições | - |
| **Fluxo Principal** | |
| Ações do Ator | Ações do Sistema |
| 1. Abrir tela de cadastramento de curso. |  |
|  | 1. Carregar tela de cadastramento de curso. |
| 1. Clicar em “Escolher Arquivo” e selecionar o arquivo que contenha os dados dos cursos a serem cadastrados. |  |
|  | 1. Exibir os dados contidos no arquivo. |
| 1. Clicar em “Salvar” para salvar os cursos no banco de dados. |  |
|  | 1. Salva os cursos como registros no banco de dados. |
|  | 1. Perguntar se o funcionário deseja ir para o menu principal ou resetar a tela atual. |
| 1. Escolher uma opção |  |
|  | 1. Realizar operação escolhida pelo funcionário. |
| **Fluxo Alternativo** | |
| Ações do Ator | Ações do Sistema |
| 1. Clicar no botão para voltar ao menu principal. |  |
|  | 1. Voltar ao menu principal. |
| **Fluxo de Exceção** | |
| Ações do Ator | Ações do Sistema |
|  | 1. Caso algum campo do formulário não seja preenchido, avisar ao funcionário e impedir o cadastramento. |

*Tabela 10. Documentação de Caso de Uso - Cadastrar Listagem de Cursos.*

### 3.4.12 Cadastrar Imagem do Aluno

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Cadastrar Imagem do Aluno** |
| Caso de Uso Geral | - |
| Ator Principal | Funcionário |
| Atores Secundários | - |
| Resumo | Este caso de uso descreve as etapas necessárias para que o funcionário realize os cadastro das imagens dos alunos. |
| Pré-Condições | O funcionário deve estar logado no sistema.  Os arquivos devem estar no seguinte formato: “F#000000000000”. |
| Pós-Condições | - |
| **Fluxo Principal** | |
| Ações do Ator | Ações do Sistema |
| 1. Abrir tela de cadastramento de curso. |  |
|  | 1. Carregar tela de cadastramento de curso. |
| 1. Clicar em “Escolher Arquivo” e selecionar o arquivo que contenha os dados dos cursos a serem cadastrados. |  |
|  | 1. Exibir os dados contidos no arquivo. |
| 1. Clicar em “Salvar” para salvar os cursos no banco de dados. |  |
|  | 1. Insere as imagens nos respectivos registros de alunos. |
| **Fluxo Alternativo** | |
| Ações do Ator | Ações do Sistema |
| 1. Clicar no botão para voltar ao menu principal. |  |
|  | 1. Voltar ao menu principal. |
| **Fluxo de Exceção** | |
| Ações do Ator | Ações do Sistema |
|  | 1. Caso o nome dos arquivos não sigam o padrão de nomes, o informar ao usuário e impedir o cadastramento. |

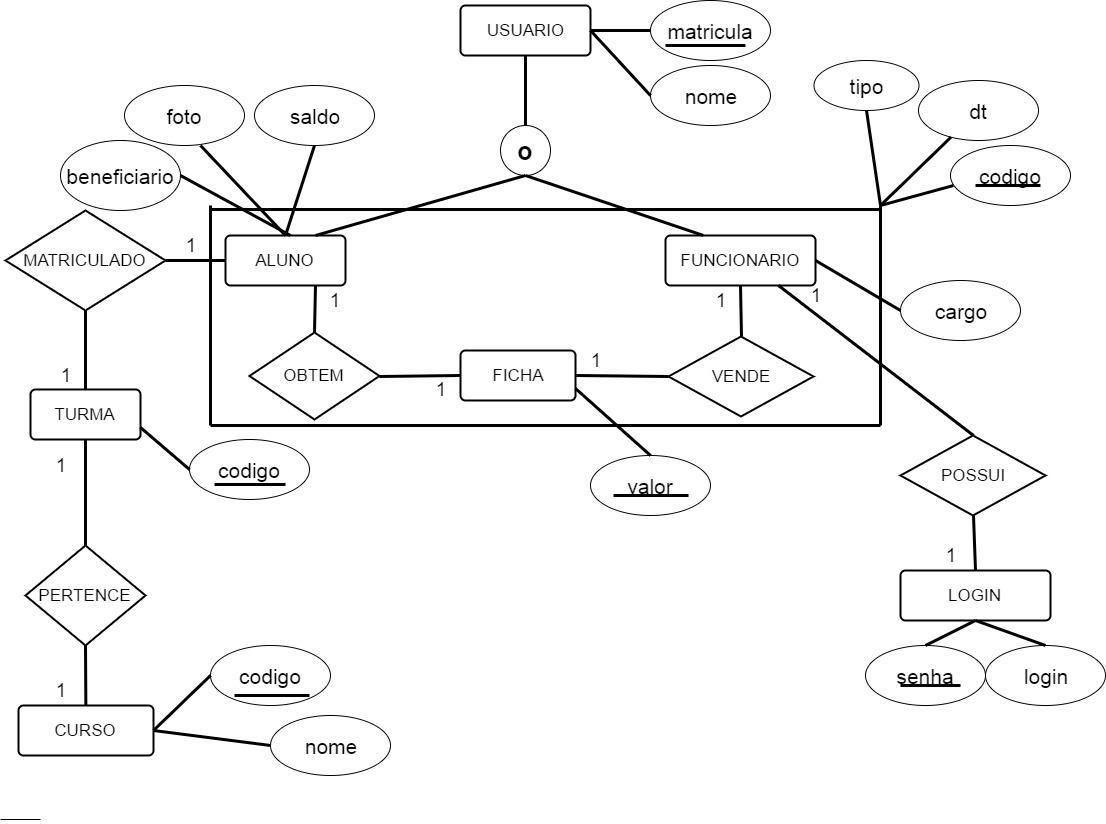
*Tabela 11. Documentação de Caso de Uso - Cadastrar Imagem do Aluno.*

## 3.5 - DIAGRAMA DE CLASSE

*Figura 2. Diagrama de Classe.*

## 

## 3.6 - MODELO ENTIDADE-RELACIONAMENTO



*Figura 3. Modelo Entidade-Relacionamento.*

# CAPÍTULO 4 - PROJETO E DESENVOLVIMENTO

## 4.1 - INTRODUÇÃO

O projeto foi implementado através da linguagem java para sua interface e para processamento do banco de dados foi utilizado a linguagem SQL. O projeto foi implementado na fórmula de módulos.

Seu desenvolvimento foi feito de forma sistemática, sendo usado o padrão de projeto DAO, um padrão de projeto que abstrai e encapsula os mecanismos de acesso a dados, escondendo os detalhes da execução da origem dos dados.

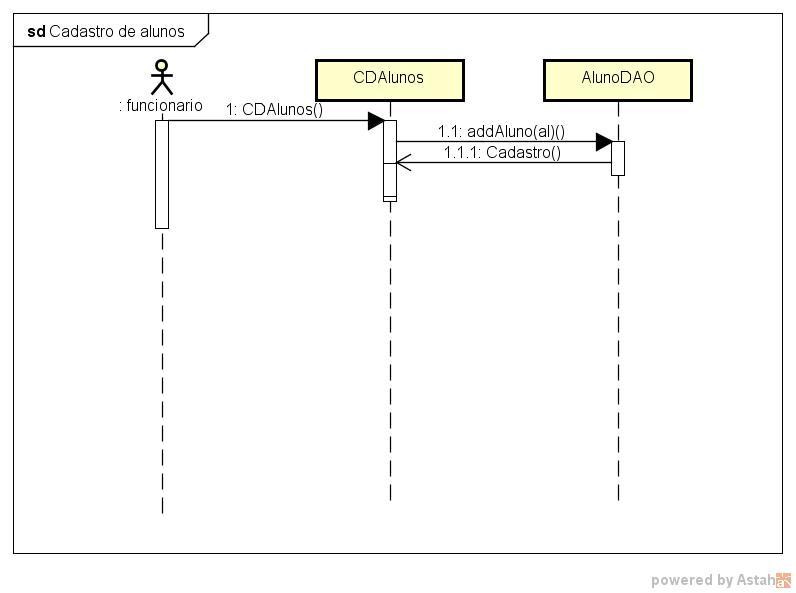
São características de implementações do padrão DAO, o acesso aos dados através das classes DAO, sendo cada instância da DAO, responsável por um objeto de domínio, etc.

Inicialmente, foram desenvolvidas as classes principais, sendo elas : Aluno, Curso, Ficha, Funcionário, Relatório, Turma e Venda. Posteriormente, foram criadas as instâncias(classes) de DAO, porém, as instâncias de DAO não devem ser responsáveis por transações, sessões ou conexões, sendo assim, houve a necessidade de implementarmos uma classe ConnectionFactory.

A classe ConnectionFactory é responsável por abrir uma conexão, ou seja, basta invocar um método e recebemos uma conexão com o banco de dados pronta para uso.

## 4.2 - DIAGRAMA DE SEQUÊNCIA

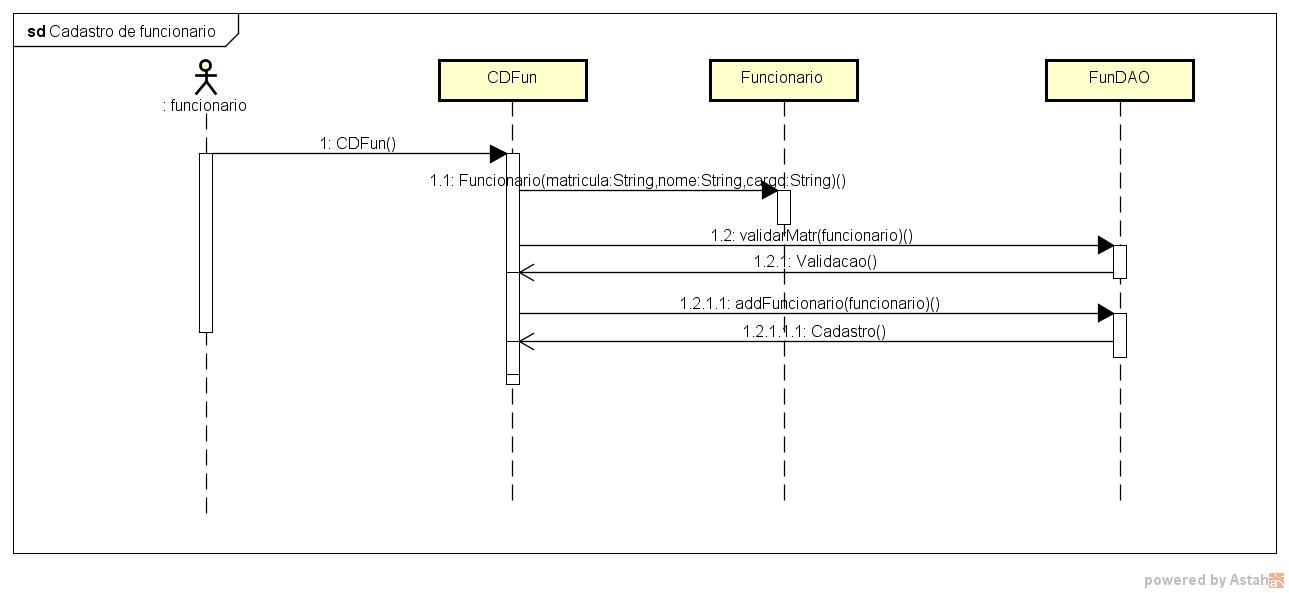
### 4.2.1 - Cadastro de listagem de alunos



*Figura 4. Diagrama de Sequência - Cadastro de Alunos.*

O cadastro de listagem de alunos, só pode ser feito após a lista ser disponibilizada pelo funcionário através do seletor de arquivos e, feito isso, o sistema adicionará todos os dados dos respectivos alunos no sistema.

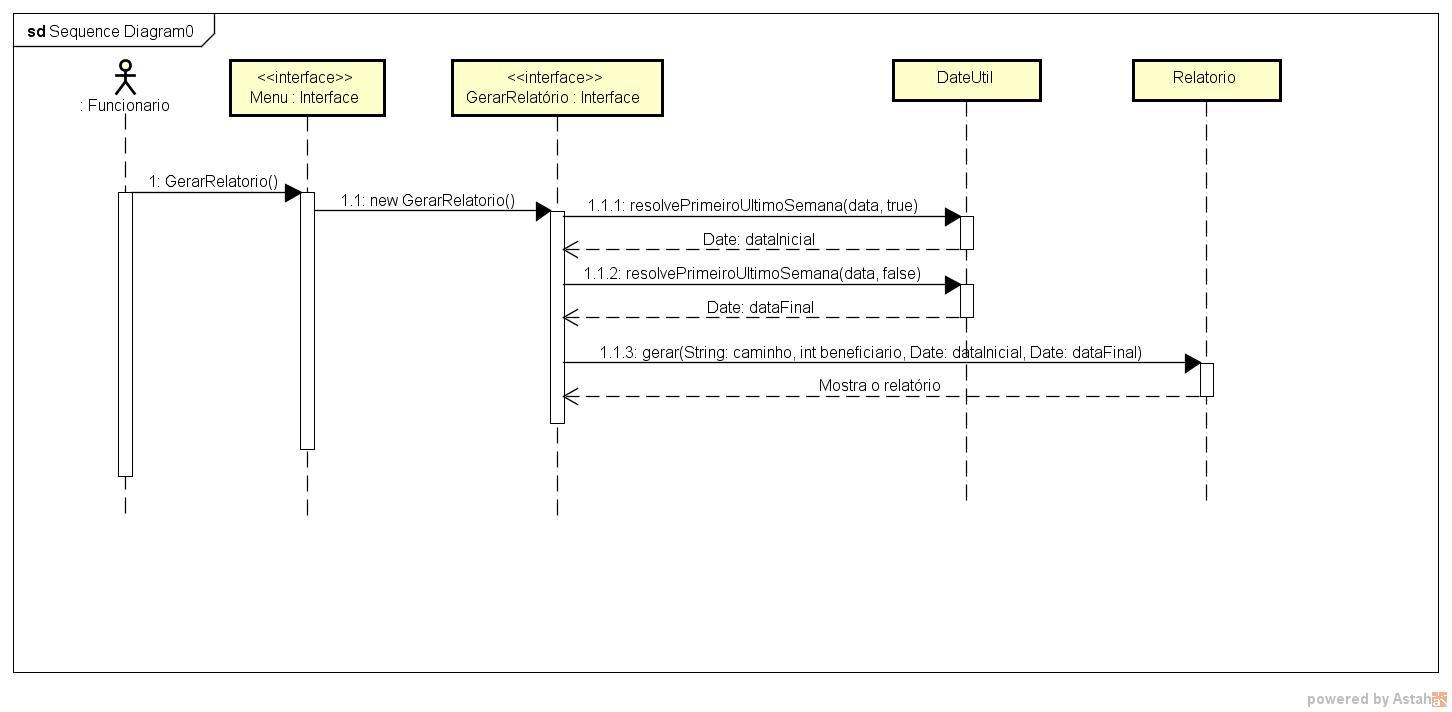
### 4.2.2 - Cadastro de Funcionário



*Figura 5. Diagrama de Sequência - Cadastro de Funcionário.*

O cadastro de funcionário é feito a partir das informações fornecidas pelo funcionário que deve ser cadastrado, tais como: sua matrícula, nome e seu respectivo cargo. Feito isso, o sistema verifica se já existe registro do mesmo e, caso exista, o funcionário não é cadastrado, caso não haja o funcionário é cadastrado.

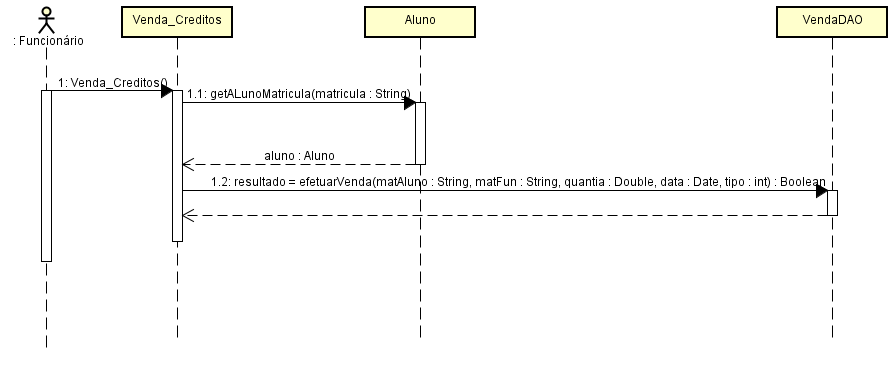
### 4.2.3 - Gerar Relatórios



*Figura 6. Diagrama de Sequência - Gerar Relatórios.*

Para gerar um relatório, o funcionário deve inserir a data inicial e final do período do relatório. Ele pode optar por usar o botão “Esta Semana”, que preenche os dados automaticamente com as datas de início e fim da semana atual. O funcionário também escolhe se o relatório vai ser dos alunos beneficiários ou não. Por fim, clica no botão “Gerar Relatório”. O sistema recupera os dados no banco de dados, cria o relatório e mostra-o ao funcionário, que tem a opção de salvar ou imprimir o arquivo.

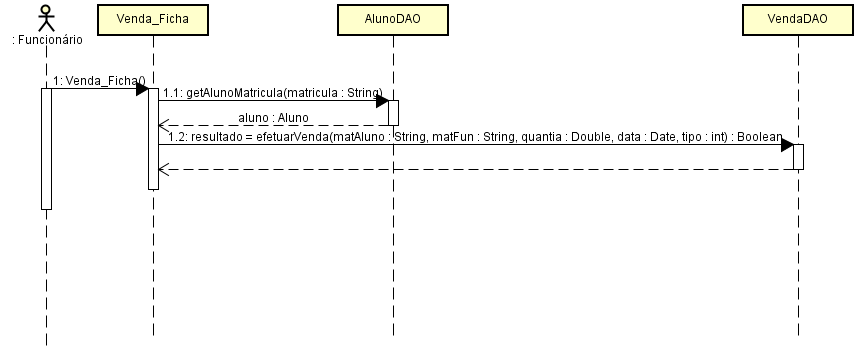
### 4.2.4 - Vender Créditos



*Figura 7.1. Diagrama de Sequência - Vender Créditos*

Para realizar a venda de créditos ao aluno, o funcionário deve entrar na tela de venda, digitar a matrícula do aluno, verificar se a matrícula corresponde à um aluno cadastrado no sistema, escolher a forma de pagamento (Dinheiro, crédito ou benefício) e efetuar a venda.

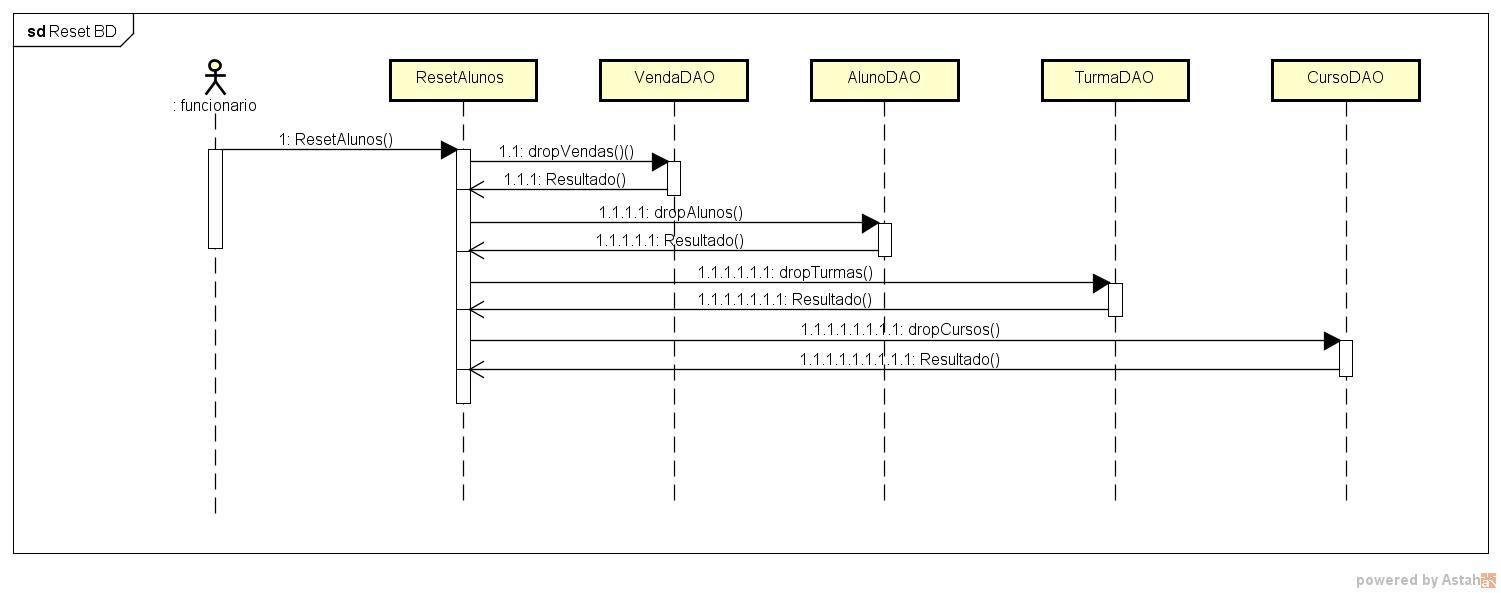
### 4.2.5 - Vender Fichas



*Figura 7.2. Diagrama de Sequência - Vender Fichas*

Para realizar a venda de créditos ao aluno, o funcionário deve entrar na tela de venda, digitar a matrícula do aluno, verificar se a matrícula corresponde à um aluno cadastrado no sistema, digitar a quantia, em reais, de créditos e efetuar a venda.

### 4.2.6 - Resetar Banco de Dados

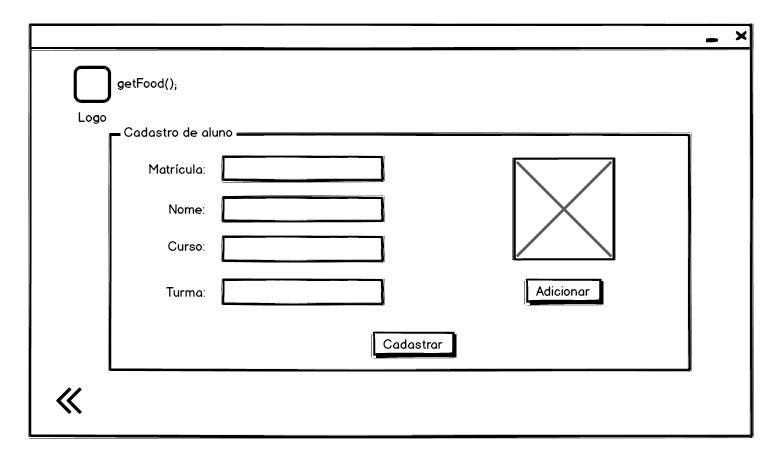


*Figura 8. Diagrama de Sequência - Resetar Banco de Dados.*

Para o banco de dados ser resetado completamente, exige algumas etapas, são elas a exclusão dos dados das vendas feitas anteriormente, exclusão dos dados de todos os alunos cadastrados no sistema, exclusão dos dados das turmas cadastradas no sistema e exclusão de todos os cursos cadastrados, caso alguma destas etapas não obtenha sucesso o banco de dados não é resetado completamente, porém se todas as etapas obtiverem sucesso, finalmente, o banco de dados é resetado completamente.

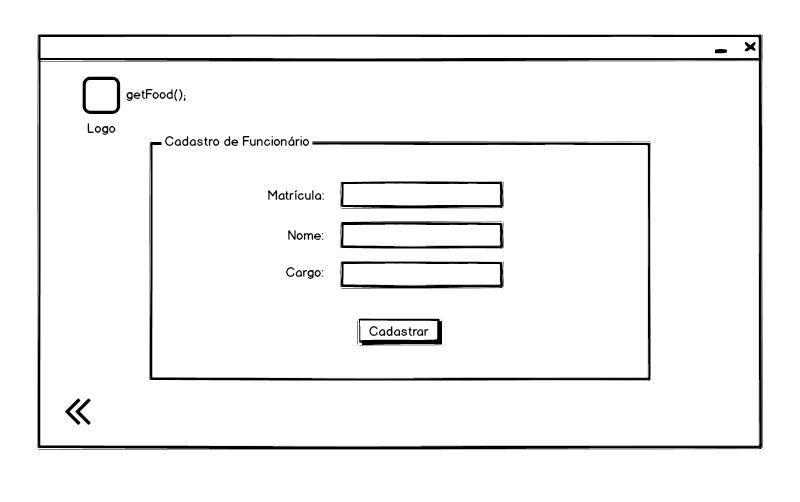
## 4.3 - PROJETO DE TELAS

### 4.3.1 - Cadastrar Alunos



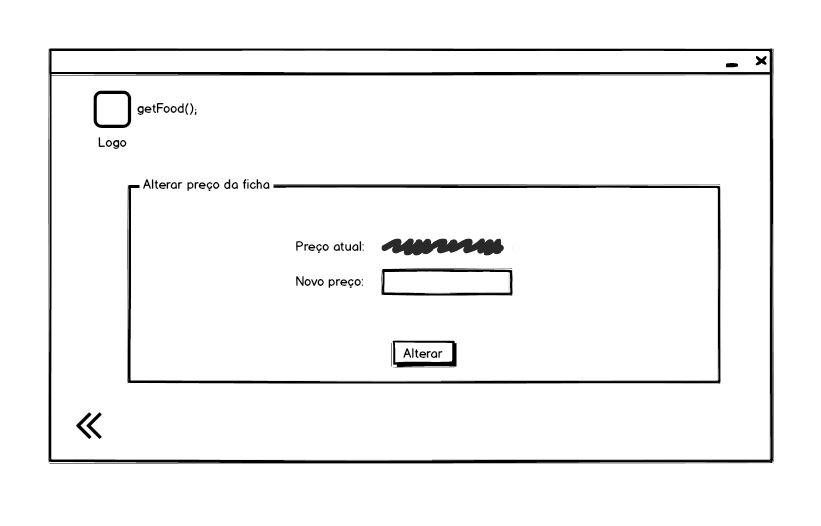
*Figura 9. Projeto de Telas - Cadastrar Alunos.*

### 4.3.2 - Cadastrar Funcionário



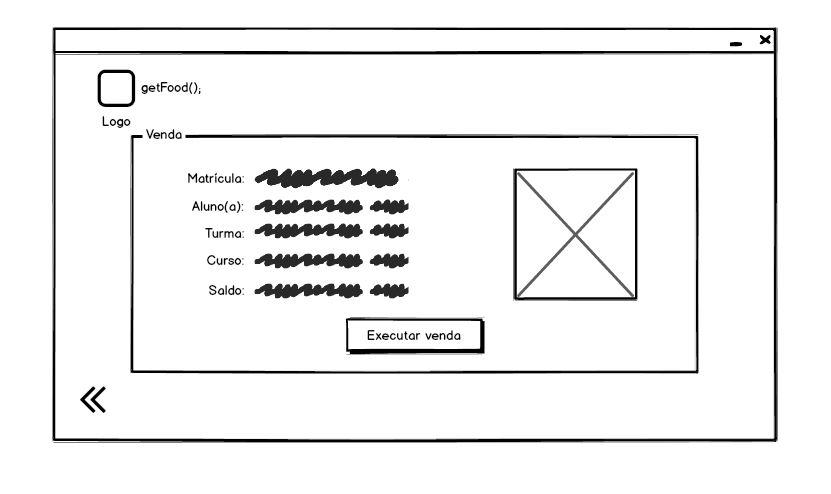
*Figura 10. Projeto de Telas - Cadastrar Funcionário.*

### 4.3.3 - Cadastrar Preço da Ficha



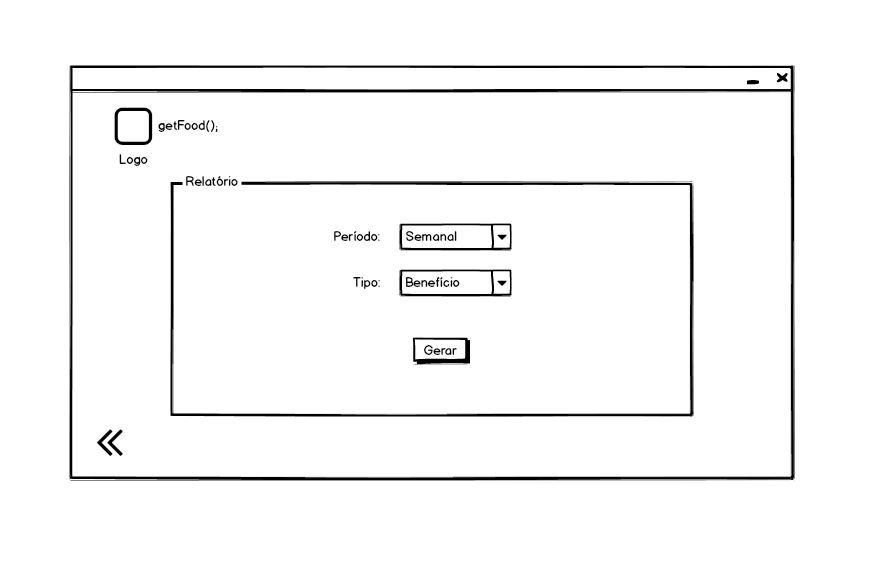
*Figura 11. Projeto de Telas - Cadastrar Preço da Ficha.*

### 4.3.4 - Executar Venda



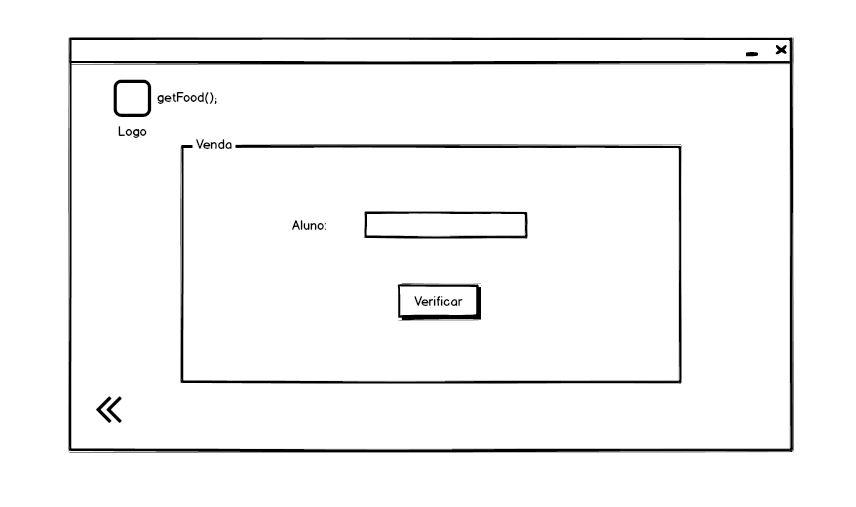
*Figura 12. Projeto de Telas - Executar Venda.*

### 4.3.5 - Gerar Relatório



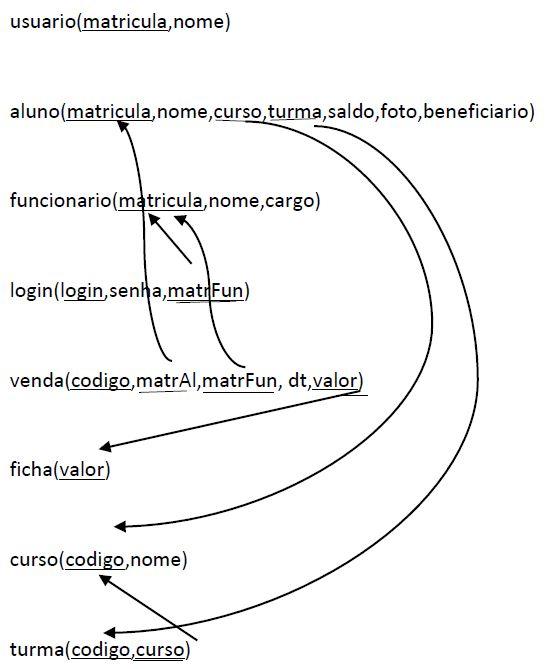
*Figura 13. Projeto de Telas - Gerar Relatório.*

### 4.3.6 - Venda



*Figura 14. Projeto de Telas - Venda.*

## 4.4 - MODELO RELACIONAL



*Figura 15. Modelo Relacional.*

## 4.5 - DICIONÁRIO DE DADOS

Legenda: Chave primária

Chave estrangeira

### 4.5.1 - Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atributo** | **Tipo** | **Tamanho** | **Nulo** | **Descrição** |
| login | varchar | 20 | Não | Id para login no sistema. |
| senha | varchar | 255 | Não | Senha para executar login no sistema. |
| matrFun | varchar | 12 | Não | Matrícula do funcionário à qual o login pertence. |

*Tabela 11. Dicionário de Dados - Login.*

### 4.5.2 - Aluno

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atributo** | **Tipo** | **Tamanho** | **Nulo** | **Descrição** |
| matricula | varchar | 12 | Não | Código de matrícula do aluno na instituição. |
| nome | varchar | 50 | Não | Nome do aluno. |
| saldo | float | - | Sim | Saldo que o aluno possui para comprar fichas registrado em sua matrícula. |
| curso | varchar | 25 | Não | Nome do curso em que o aluno está atualmente matriculado. |
| turma | varchar | 25 | Não | Turma do respectivo curso em que o aluno está matriculado. |
| foto | longblob | - | Sim | Foto de identificação do aluno. |

*Tabela 12. Dicionário de Dados - Aluno.*

### 4.5.3 - Funcionário

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atributo** | **Tipo** | **Tamanho** | **Nulo** | **Descrição** |
| matricula | varchar | 12 | Não | Número de matrícula do funcionário na instituição. |
| nome | varchar | 50 | Não | Nome do funcionário. |
| cargo | varchar | 25 | Não | Cargo que o funcionário exerce dentro da instituição. |

*Tabela 13. Dicionário de Dados - Funcionário.*

### 4.5.4 - Curso

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atributo** | **Tipo** | **Tamanho** | **Nulo** | **Descrição** |
| codigo | varchar | 25 | Não | Código do curso. |
| nome | varchar | 50 | Não | Nome do curso. |

*Tabela 14. Dicionário de Dados - Curso.*

### 4.5.5 - Turma

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atributo** | **Tipo** | **Tamanho** | **Nulo** | **Descrição** |
| codigo | varchar | 25 | Não | Código da turma. |
| curso | varchar | 25 | Não | Código do curso qual a turma pertence. |

### 4.5.6 - Ficha

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atributo** | **Tipo** | **Tamanho** | **Nulo** | **Descrição** |
| valor | float | - | Não | Preço da ficha. |

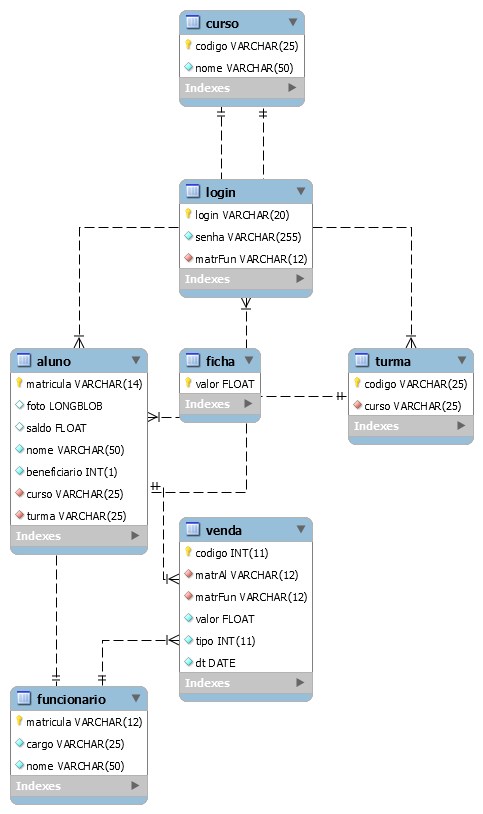
*Tabela 15. Dicionário de Dados - Ficha.*

### 4.5.7 - Venda

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atributo** | **Tipo** | **Tamanho** | **Nulo** | **Descrição** |
| codigo | int | - | Não | Código da venda efetuada. |
| matrAl | varchar | 12 | Não | Matrícula do aluno que efetuou a compra. |
| matrFun | varchar | 12 | Não | Matrícula do funcionário que efetuou a compra. |
| valor | float | - | Não | Valor da compra efetuada. |
| dt | date | - | Não | Data que a venda foi efetuada. |

*Tabela 16. Dicionário de Dados - Venda.*

## 4.6 - ESQUEMA DO BANCO DE DADOS (BD)



*Figura 16. Esquema do Banco de Dados.*

## 4.7 - LISTAGEM DE CLASSES

* Aluno;
* AlunoDAO;
* ConnectionFactory;
* Curso;
* CursoDAO;
* Ficha;
* FichaDAO;
* Funcionario;
* FuncionarioDAO;
* LoginDAO;
* Relatório;
* Turma;
* TurmaDAO;
* Venda;
* VendaDAO.

# CAPÍTULO 5 - CONCLUSÕES OU CONSIDERAÇÕES

## 5.1 - INTRODUÇÃO

A proposta deste projeto foi a automatização de um processo problemático, o qual, não possui histórico de automatização, porém, é fato que seria de fácil automatização, uma vez que o processo fosse estudado e otimizado.

## 5.2 - CONCLUSÕES

Este projeto foi desenvolvido por alunos que sofreram com a falta de automatização do processo problematizado, sendo o mesmo, um exemplo de tarefa, a qual, atualmente vem sendo realizada manualmente, porém, deveria ser automatizada.

Processos que são de fáceis automatizaçoes, e o processo dos mesmos sejam de grande repercussão ou não, trazem diversos benefícios como a baixa taxa de erros, eficiência, segurança e precisão.

Este projeto foi realizado, visando solucionar uma problematização presente na vida de todos os discentes da instituição, o gerenciamento de fichas alimentícias e os processos abrangidos por este gerenciamento. Por exemplo, atualmente para um discente conseguir uma ficha alimentícia, o mesmo tem que assinar uma lista, a qual, é usada para o controle da distribuição de fichas e posteriormente, essa assinatura é usada para a contagem e geração de relatórios, tendo em vista essa informação, ainda não fica claro como esse processo pode se tornar problemático.

A problematização deste pequeno processo surge ao sabermos que o Instituto Federal do Amazonas - Campus Manaus Centro, dispõe de 5 cursos de nível técnico integrado ao ensino médio e cada um desses 5 cursos, possuem 3 turmas ao total, exceto um dos cursos que possuem 6 turmas, ou seja, ao total são 18 turmas de aproximadamente 35 alunos com aproximadamente 630 alunos, os quais, precisam assinar uma única lista e conseguirem almoçar em apenas uma hora.

Finalmente, nossa proposta de solução foi um sistema de simples navegação e interpretação que executasse as principais funções necessárias para que a distribuição e seu gerenciamento fossem feitos de formas instantânea, simples e eficaz.

## 5.3 - DIRECIONAMENTO PARA TRABALHOS FUTUROS

Durante o desenvolvimento deste projeto, não foi possível implementar uma função que permitiria o aluno obter fichas alimentícias através do uso da biometria digital, especificamente, através da leitura da digital do aluno.

A ideia principal desta função seria que ao invés do aluno fornecer sua matrícula ao funcionário encarregado de vender ou distribuir fichas, ele usasse o leitor de biometria, o qual, seria disponibilizado e o sistema se encarregaria de certificar-se de que o aluno possui crédito, caso o mesmo seja pagante.

Essa função não pode ser implementada devido ao fato de que os leitores biométricos disponibilizados pela instituição não possuem compatibilidade com os sistemas operacionais atuais usados na instituição.

## 5.4 - REFERÊNCIAS

Google Drive, “AULA1\_ANÁLISE DE SISTEMAS “. Disponível em : <https://drive.google.com/file/d/0B-O4V2c0z7IPU1FYZmpJM0Rxajg/edit> . Acessado em 01/08/2017.

[Reflectz’s Blog](https://brunobrum.wordpress.com/), “Principais Técnicas de Levantamento de Requisitos de Sistemas”. Disponível em : <https://brunobrum.wordpress.com/2011/04/27/principais-tecnicas-de-levantamento-de-requisitos-de-sistemas/> . Acessado em 01/08/2017.

NetBeans, “NetBeans IDE - Overview”. Disponível em : <https://netbeans.org/features/index.html> . Acessado em 02/08/2017.

EnaComp, “Evolução e Principais Caracterísitcas do IDE Eclipse”. Disponível em : <http://www.enacomp.com.br/2010/cd/artigos/completos/enacomp2010_23.pdf> . Acessado em 02/08/2017.

Até o Momento, “Caso de Uso - Fluxo Alternativo”. Disponível em : <http://www.ateomomento.com.br/caso-de-uso-fluxo-alternativo/> . Acessado em 02/09/2017.

Até o Momento. “Caso de Uso - Fluxo Principal”. Disponível em : http://www.ateomomento.com.br/caso-de-uso-fluxo-alternativo/ . Acessado em 02/09/2017.

Até o Momento, “Caso de uso - Fluxo de Exceção”. Disponível em : <http://www.ateomomento.com.br/caso-de-uso-fluxo-de-excecao/> . Acessado em 02/09/2017.

Jaspersoft Community, “JasperReports Library”. Disponível em : <https://community.jaspersoft.com/project/jasperreports-library> . Acessado em 03/11/2017.

Jaspersoft Community, “iReport Designer”. Disponível em : <https://community.jaspersoft.com/project/ireport-designer> . Acessado em 03/11/2017.

IBM, “Fundamentos da linguagem Java”. Disponível em : <https://www.ibm.com/developerworks/br/java/tutorials/j-introtojava1/index.html> . Acessado em 03/11/2017.

DevMedia, “SQL e Programação de Banco de Dados”. Disponível em : <https://www.devmedia.com.br/sql-e-programacao-de-banco-de-dados/3139> . Acessado em 03/11/2017.

DevMedia, “Introdução ao MySQL”. Disponível em : <https://www.devmedia.com.br/introducao-ao-mysql/27799> . Acessado em 03/11/2017.

DevMedia, “JavaSwing: Conheça os componentes JTextField e JFormatted TextField”. Disponível em : <https://www.devmedia.com.br/java-swing-conheca-os-componentes-jtextfield-e-jformattedtextfield/30981> . Acessado em 03/11/2017.

Caelum, “Banco de dados e JDBC”. Disponível em : <https://www.caelum.com.br/apostila-java-web/bancos-de-dados-e-jdbc/> . Acessado em 04/11/2017.

Java Building, “DAO”. Disponível em : <http://www.javabuilding.com/academy/patterns/dao.html>. Acessado em 04/11/2017.

Google Drive, “AULA2\_Elicitação de Requisitos\_MANAUS.pdf”. Disponível em : <https://drive.google.com/file/d/0B-O4V2c0z7IPanRyVTBtSTZnWVE/edit> . Acessado em 04/11/2017.

Draw.io, “*Flowchart maker and online diagram software*”. Disponível em : <https://www.draw.io/> . Acessado em 04/11/2017.

## 5.5 - ANEXO(S)

### 5.5.1 - QUESTIONÁRIO DE ELICITAÇÃO DE DADOS

1. Em algum momento já existiu algum projeto para a automatização deste projeto?
2. Existe alguma função que gostaria que implementássemos?
3. Na sua opinião, o uso de fotos para a identificação do aluno parece bom?
4. As fichas do almoço são disponibilizadas para todos os alunos?
5. Na sua opinião, o aluno deveria ter a opção de crédito?
6. Caso o aluno não pague esse crédito, o aluno deve ter que pagar uma multa ou o impedimento de comprar a ficha?
7. Atualmente os alunos só podem obter uma ficha por dia, na sua opinião isso deveria continuar?
8. Qual a sua opinião na venda de pacotes semanais?
9. A geração de relatórios é feita semanalmente ou mensalmente?
10. O relatório deve ser dividido em quais critérios?

1. astah.net/ [↑](#footnote-ref-1)
2. [www.draw.io/](https://www.draw.io/) [↑](#footnote-ref-2)
3. [www.ibm.com/developerworks/br/java/tutorials/j-introtojava1/index.html](https://www.ibm.com/developerworks/br/java/tutorials/j-introtojava1/index.html) [↑](#footnote-ref-3)
4. [www.devmedia.com.br/java-swing-conheca-os-componentes-jtextfield-e-jformattedtextfield/30981](https://www.devmedia.com.br/java-swing-conheca-os-componentes-jtextfield-e-jformattedtextfield/30981) [↑](#footnote-ref-4)
5. [netbeans.org/features/index.html](https://netbeans.org/features/index.html) [↑](#footnote-ref-5)
6. [www.devmedia.com.br/introducao-ao-mysql/27799](https://www.devmedia.com.br/introducao-ao-mysql/27799) [↑](#footnote-ref-6)
7. [www.devmedia.com.br/introducao-ao-mysql/27799](https://www.devmedia.com.br/introducao-ao-mysql/27799) [↑](#footnote-ref-7)
8. [community.jaspersoft.com/project/jasperreports-library](https://community.jaspersoft.com/project/jasperreports-library) [↑](#footnote-ref-8)
9. [community.jaspersoft.com/project/ireport-designer](https://community.jaspersoft.com/project/ireport-designer) [↑](#footnote-ref-9)