

**FUNDAÇÃO EDUCACIONAL DE ITUVERAVA
FACULDADE DR.FRANCISCO MAEDA**

Rodolfo da Silva Rodrigues Corrêa

**DESENVOLVIMENTO DE UM SISTEMA PARA GERIR OS PROCESSOS DE
ELEIÇÃO DE DIRETÓRIOS ACADÊMICOS UTILIZANDO A LINGUAGEM JAVA
NO AMBIENTE DE DESENVOLVIMENTO INTEGRADO NETBEANS**

**ITUVERAVA
2016**

RODOLFO DA SILVA RODRIGUES CORRÊA

**DESENVOLVIMENTO DE UM SISTEMA PARA GERIR OS PROCESSOS DE
ELEIÇÃO DE DIRETÓRIOS ACADÊMICOS UTILIZANDO A LINGUAGEM JAVA
NO AMBIENTE DE DESENVOLVIMENTO INTEGRADO NETBEANS**

**Trabalho de Conclusão de Curso apresentado
à Faculdade Dr. Francisco Maeda. Fundação
Educativa de Ituverava para obtenção do
título de Bacharel em Sistemas de Informação.**

**Orientador: Prof. MSc. Fabrício Gustavo
Henrique.**

**ITUVERAVA
2016**

005.133j Corrêa, Rodolfo da Silva Rodrigues
C824d Desenvolvimento de um sistema para gerir os processos de
eleição de diretórios acadêmicos utilizando a linguagem Java
no Ambiente de Desenvolvimento Integrado Netbeans/Rodolfo
da Silva Rodrigues Corrêa – Ituverava: FE/FAFRAM, 2016.
42f.
Trabalho de Conclusão de Curso (Graduação em Sistemas
de Informação - Bacharelado).
Orientador: Fabrício Gustavo Henrique.
1.Instituição de ensino 2.Informatização 3.Processos
acadêmicos

RODOLFO DA SILVA RODRIGUES CORRÊA

**DESENVOLVIMENTO DE UM SISTEMA PARA GERIR OS PROCESSOS DE
ELEIÇÃO DE DIRETÓRIOS ACADÊMICOS UTILIZANDO A LINGUAGEM JAVA
NO AMBIENTE DE DESENVOLVIMENTO INTEGRADO NETBEANS**

**Trabalho de Conclusão de Curso apresentado
à Faculdade Dr. Francisco Maeda. Fundação
Educativa de Ituverava para obtenção do
título de Bacharel em Sistemas de Informação.**

Ituverava, 26 de outubro de 2016.

Orientador (a): _____
Prof. MSc. Fabrício Gustavo Henrique

Examinador (a): _____
Nome do examinador (a)

Examinador (a): _____
Nome do examinador (a)

DEDICATÓRIA

Dedico este trabalho, primeiramente aos meus pais que sempre estiveram e ainda estão ao meu lado nos momentos de dificuldade, aos professores que me ajudaram no desenvolvimento deste projeto e também aos meus amigos que me auxiliaram desde o início do curso.

AGRADECIMENTOS

Sou grato,

A Deus que sempre me fortalece a cada dia e me guia para a justiça.

A minha família no qual é como uma estrutura para mim.

Ao meu orientador que sempre esteve disposto a fazer o possível para o bom resultado deste projeto.

Aos meus amigos que sempre estiveram me auxiliando e ensinando nas partes mais complexas deste curso.

“Se o conhecimento pode criar problemas, não é através da ignorância que podemos solucioná-los.”

Isaac Asimov.

RESUMO

Atualmente, as instituições de Educação Superior vêm informatizando suas rotinas administrativas para redução de tempo e custos operacionais. Nessa situação, esse trabalho descreve uma ferramenta para informatizar todo o processo eleitoral dos diretórios acadêmicos da Faculdade Dr. Francisco Maeda, localizada no município de Ituverava, estado de São Paulo. Para o desenvolvimento, foi utilizada a linguagem JAVA, IDE Netbeans e o sistema de gerenciamento de banco de dados (SGBD) Postgres. A ferramenta possibilitou um controle rigoroso dos dados, segurança para o responsável por esses processos, e a redução do custo operacional.

Palavras-chave: Instituição de ensino. Informatização. Processos acadêmicos.

SUMMARY

Nowadays, the institutions of higher education have been computerizing their administrative procedures to reduce the time and operating costs. In this issue, this project describes a tool to computerize the electoral academic department processes at the Faculty of Dr. Francisco Maeda located in the county of Ituverava, São Paulo State. For the development of this project, it has been used the programming language JAVA, the Integrated Development Environment (IDE) Netbeans and the Data Base Management System (DBMS) Postgres. This tool has been providing a very strict control of data, security for responsible for these procedures and a reduction of operating costs.

Keywords: Educational institution, Computerizing, Academic procedures.

LISTA DE ABREVIATURAS E SIGLAS

DA	DIRETÓRIO ACADÊMICO
FAFRAM	FACULDADE DR. FRANCISCO MAEDA
SGBD	SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS
SQL	STRUCTURED QUERY LANGUAGE
POO	PROGRAMAÇÃO ORIENTADA À OBJETO
OO	ORIENTADA À OBJETO
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
XP	EXTREME PROGRAMMING
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
MER	MODELO DE ENTIDADE-RELACIONAMENTO

LISTA DE FIGURAS

Figura 1 – Caso de uso: Usuário administrador.....	25
Figura 2 – Caso de uso: Usuário normal	25
Figura 3 – Caso de uso: Aluno.....	26
Figura 4 – Diagrama de classe: Funcionário e usuário	26
Figura 5 – Diagrama de classe: Chapas e relacionados	27
Figura 6 – Diagrama de classe: Eleições e relacionados.....	27
Figura 7 – MER	28
Figura 8 – Tela Inicial do usuário tipo: Administrador.....	29
Figura 9 – Tela de funcionários com todas as funções	30
Figura 10 – Código para o preenchimento das tabelas	30
Figura 11 – Código de verificação para exclusão e outros.....	31
Figura 12 – Tela de Chapas com todas as funções.....	31
Figura 13 – Tela de inserção de alunos em chapas	32
Figura 14 – Código de inserção de alunos na chapa	32
Figura 15 – Código da tabela de cargos disponíveis	33
Figura 16 – Tela de Cargos com todas as funções	33
Figura 17 – Tela de inserção e remoção de chapas na eleição	34
Figura 18 – Tela de controle de votação.....	35
Figura 19 – Tela para votar.....	35
Figura 20 – Código da tela de votação.....	36
Figura 21 – Tela de Resultados	36
Figura 22 – Código de finalização de votação	37
Figura 23 – Tela do relatório de resultados.....	37

SUMÁRIO

INTRODUÇÃO.....	111
1 A IMPORTÂNCIA DA INFORMATIZAÇÃO DA INFORMAÇÃO.....	13
2 INFORMATIZAÇÃO NOS DIRETÓRIOS ACADÊMICOS (DA).....	14
3 MATERIAL E METODO.....	15
3.1 Banco de Dados.....	15
3.2 SGDBs e PostgreSQL.....	15
3.2.1 Banco de Dados Relacional	16
3.3 Programação Orientada a Objeto (POO).....	17
3.3.1 Classe	17
3.3.2 Objeto	17
3.3.3 Atributo e Método	18
3.3.4 Linguagem JAVA	18
3.4 Netbeans IDE.....	19
4 DESENVOLVIMENTO.....	20
4.1 Estudo de viabilidade.....	20
4.2 Levantamento de Requisitos.....	21
4.2.1 Técnicas de levantamento de Requisitos	22
4.2.2 Requisitos Funcionais	22
4.2.3 Requisitos Não Funcionais	23
4.3 DIAGRAMAS.....	24
4.3.1 Caso de uso	24
4.3.1.1 Usuário Administrador.....	24
4.3.1.2 Usuário Normal.....	25
4.3.1.3 Aluno.....	26
4.3.2 Diagrama de classe	26
4.3.4. Modelo de Entidade-Relacionamento (MER)	28
5 INTERFACE DO SISTEMA.....	29
CONCLUSÃO.....	38
REFERÊNCIAS.....	39

INTRODUÇÃO

Diversas entidades vem implementando a tecnologia em seus processos, não somente para reduzir o tempo operacional, mas para abater os custos que as organizações sofrem devido ao grande uso de materiais que poderiam ser substituídos por ferramentas adequadas para funções repetitivas ou que requerem uma lógica mais complexa para resolução de problemas.

A tecnologia tem mudado a vida de muitos que necessitam de uma ferramenta para tornar ágeis seus processos e reduzir seus custos materiais. Com esse desenvolvimento rápido, diversas empresas foram atingidas por esta mudança, que até então, já necessitavam de melhorias em seus métodos de trabalho ora manual. Sacilotti (2011, p.12) afirma que com o progressivo avanço tecnológico e a integração, a demanda e o aperfeiçoamento das informações vêm se tornando um dos primordiais objetivos das empresas. Sem este conhecimento, é impossível inovar-se e adaptar-se a esse ambiente.

Podemos ver, através do ambiente em que as organizações estão vivenciando, que o conhecimento e a informação unem-se e se tornam algo que superam as necessidades e perspectivas, onde as organizações e seus coparticipantes obtenham um senso comum relacionado às metas criadas através da troca de interesses. Tendo isso, as organizações permanecerão no mercado por mais tempo. Através destes pontos, a tecnologia da informação vem assumindo uma nova função, de administrar os interesses como um todo (TEÓFILO; FREITAS, 2007, p.2).

Não sendo diferente com as instituições de ensino, a informática se tornou uma ferramenta indispensável, não só para alunos e professores, mas também para o funcionamento da instituição em geral, onde processos que antes eram feitos com papéis e canetas, hoje são totalmente digitais. E é neste tema que o sistema está explicitamente relacionado.

A Faculdade Dr. Francisco Maeda (FAFRAM), instituição de Educação Superior mantida pela Fundação Educacional de Ituverava (FE), sediada no município de Ituverava, estado de São Paulo, foi fundada em 1987 para atender a demanda dos vários segmentos da sociedade com ensino de qualidade (COSTA; JABUR, 2010, p.35). Com base nesta grande estrutura, este sistema tem como objetivo informatizar alguns dos processos de eleição de diretórios acadêmicos, reduzindo a utilização de papéis e amortizando os custos operacionais. O uso de papéis é usado constantemente em quase todos os processos relacionados à eleição e votação dos diretórios acadêmicos.

Apesar de algumas funções já serem informatizadas (Cadastro de funcionários, por exemplo), o sistema terá todas as funções incluídas, para que seja possível obter um controle completo, não se tornando um sistema particionado.

O processo de votação é realizado da seguinte maneira: Os alunos assinam seus nomes como forma de comprovação de sua presença na votação e que concretizou o seu voto, então ele escolhe a chapa desejada e insere o papel dentro de uma urna. Após este processo, o responsável pela votação realiza a contagem destes votos papel por papel e, após somar todos os votos, o resultado é alcançado. Para que o consumo e o tempo sejam reduzidos, o sistema armazenará todos os dados necessários para a realização deste processo, desde o armazenamento das chapas e dos seus participantes até o resultado da votação.

Além de todo este controle, o sistema possui outras funções não tão importantes quanto às citadas acima, mas também oferecendo um aprimoramento e maior qualidade ao realizar funções específicas. Sendo assim, o sistema pode realizar um controle de funcionários e usuários, controlar as informações através do cadastro de todas as partes do sistema, facilitar a organização entre alunos e chapas, assim como chapas e cargos, controlar as votações e seus resultados, armazenar resultados e a criação de relatórios para o mesmo e, por final, utilizar o conceito de privilégios no sistema para o real uso da distinção entre usuário administrador e usuário normal, onde se distingue através de opções oferecidas ao administrador e ao usuário.

1 A IMPORTÂNCIA DA INFORMATIZAÇÃO DA INFORMAÇÃO

A informação possui um forte peso de importância no ciclo de vida de qualquer empresa. Sendo informatizada ou não, as empresas de grande ou pequeno porte utilizam de grandes quantidades de informações que, ao se analisar detalhadamente, podem ser consideradas como vida da empresa. Para Aceti (2006, p.40), a informação possui uma definição tão importante que ela se torna o centro de todo o trabalho. A informação é nada menos que um dos patrimônios mais raro e de maior importância.

Além disto, a informatização carrega uma ampla ideia que faz com que administradores e empresários se preocupem ainda mais com seus bens e gastos, ela oferece o conceito de uma nova cultura no local em que é aplicada. Em tempos em que os dados de uma empresa ou instituição se tornam algo valioso, novos investimentos são realizados para que o custo de processos seja abatido de maneira que estes dados permaneçam armazenados de maneira segura e de fácil acesso. Zuboff (1994, p.82) explica que, em geral, administradores realizam seus investimentos em tecnologias mais recentes, pois acreditam que elas lhes trarão maior rapidez e um menor custo ao realizar suas operações.

Com o uso de tais informações, as empresas se dispõem de um apoio nas suas tomadas de decisões, no qual se tornam precisas ao serem analisadas adequadamente, pois a informação é o próprio conhecimento. Quando ela é usada de forma estratégica, um grande leque de oportunidades é aberto e o que estava oculto se manifesta. De acordo com Salgado (2007, p.24), a informação se transforma em conhecimento a partir do momento em que ela proporciona a compreensão, experiência ou capacidade que se aplica a algum indivíduo ou a um grupo, ficando incorporada nas pessoas como um conhecimento interno.

Ao possuir tal conhecimento, a empresa atua na frente que qualquer outra que não se dispõe deste tipo de tecnologia ou cultura. Simplificando, o uso da informação como ferramenta de auxílio em conjunto com a tecnologia podem servir de grande assistência ao gerir uma grande instituição.

2 INFORMATIZAÇÃO NOS DIRETÓRIOS ACADÊMICOS (DA)

Assim como todas as faculdades, os cursos são administrados por Diretórios Acadêmicos (DAs), que por sua vez, possuem uma grande quantidade de informações sobre a parte discente, contábil e toda a gestão de conteúdo para a melhoria e organização dos cursos. Ao pesquisar oportunidades para melhorias da instituição, foi notado que alguns processos na parte de DA necessitavam de melhorias em relação à informatização de métodos utilizados para a realização de tarefas em tempo de eleição.

Após um entendimento com o responsável, no qual é o encarregado por gerir toda a parte que engloba o ambiente DA, foi vista necessidade de informatização. Tendo em vista o conceito de informatizar, criou-se a motivação para o desenvolvimento de um sistema que aprimoraria a gestão dos processos que englobam desde a parte de criação de um DA até a sua votação.

Nestes processos, são utilizados grandes quantidades de papéis e tempos de funcionários para a execução destes. Logo, com o emprego deste sistema, tais problemas seriam amenizados e melhorias seriam criadas.

3 MATERIAL E METODO

3.1 Banco de Dados

Ao se analisar o mercado, é possível observar que todo sistema necessita armazenar seus dados, para que estes dados possam ser consultados ou utilizados de alguma maneira. Segundo Matsumoto (2006, p.46) “O primeiro passo para se obter informações estratégicas para uma gestão eficiente é a formação de bancos de dados consistente”.

Seja um banco na própria máquina ou online, a informação possui um valor imenso e deve ser salva de maneira inteligente e segura. Para Bochner et al. (2012, p.2), na globalização em que habitamos, a informação vem se tornando significativa e necessitada, onde diversos elementos sociais fundamentam-se gradativamente de informação que estão a disposição para a efetuação de suas operações.

Com a utilização de uma base de dados, todo o trabalho de acumular papéis dentro da gaveta é substituído por uma tecnologia simples e de baixo custo (dependendo do gerenciador de banco que se usa). Neste projeto a base de dados utilizada é gerenciada pelo Sistema de Gerenciamento de Banco de Dados (SGBD) PostgreSQL.

3.2 SGDBs e PostgreSQL

Com o uso de um SGDB é possível realizar o controle de todas as suas informações de maneira que o sistema que o utiliza possa não somente consultá-los, mas também realizar todas as funções possíveis, desde a criação de uma nova informação até a sua exclusão ou recuperação.

SGBD é um conjunto de softwares altamente complexos e sofisticados que possibilitam, basicamente, os serviços de armazenamento, recuperação e gerência de dados. Existe uma grande diversidade de arquiteturas de SGBD, no entanto, é possível identificar um conjunto básico de funcionalidades que normalmente consta das mais diferentes arquiteturas (CÔRTEZ, 2001, p.2).

Ao compreender um pouco mais sobre SGBDs é fácil perceber a importância de seu uso tanto quanto a Base de Dados utilizada e, com o uso dos dois, é possível ter um banco gerenciado e bem estruturado podendo realizar operações complexas. Existem diversos SGBDs disponíveis no mercado, entre eles estão o Oracle, SQL Server, MySQL e o utilizado neste projeto chamado PostgreSQL. É claro que existem outros, porém estes são os mais

conhecidos até a data em que este trabalho foi escrito.

O *PostgreSQL* está entre os SGBDs classificado como um *software* livre mais avançado. Ele é um resultado de um trabalho da Universidade de Berkeley, Califórnia. Ele foi criado pelo professor de ciência da computação chamado Michael Stonebraker e foi iniciado na década de 80, onde era chamado no princípio de Postgres (PostgreSQL, 2016).

Por todos os motivos descritos acima, foi utilizado o PostgreSQL, pois é uma ótima ferramenta para armazenamento de dados utilizando diversos recursos para a sua melhor automatização. Além de seu ótimo desempenho, este banco foi escolhido por ser uma ferramenta gratuita e de fácil uso.

3.2.1 Banco de Dados Relacional

O PostgreSQL é um SGBD Objeto-Relacional, o que significa que ele é SGBD que utiliza o modelo orientado à objeto onde é possível realizar diversas funções combinadas de outros modelos já existentes.

O modelo de dados orientado a objeto é baseado no paradigma de linguagem de programação orientada a objeto, muito utilizada atualmente. Herança, identidade de objetos e encapsulamento (ocultação de informações), com métodos para fornecer uma interface para objetos, estão entre os principais conceitos da programação orientada a objeto que encontraram usos na modelagem de dados (SILBERSCHATZ, 1999, p.13).

Os dados são armazenados em tabelas ou entidades. Essas tabelas possuem colunas e linhas onde as colunas são conhecidas como atributos e as linhas como tuplas e, dentro de cada entidade, se encontra um atributo chamado chave primária.

Pode-se entender que a chave primária é a identidade única de cada linha ou tupla dentro de uma tabela e é com essa chave que se torna possível criar a relação dentro do banco de dados. No momento que se cria a relação entre duas tabelas, a chave primária de uma entidade recebe o nome de chave estrangeira na outra tabela que se relacionou. Através disto, é possível evitar redundâncias de informações armazenadas na base de dados.

Para Bonfioli (2006, p.15) o modelo relacional demonstrou-se ser o mais apropriado e ágil ao solver os diversos impasses que se apresentam na criação e implementação da estrutura de dados. A base essencial do modelo relacional é a própria relação. Ela é fundada através de um ou mais atributos que remete o traço de um dado a se armazenar.

3.3 Programação Orientada a Objeto (POO)

Programação Orientada a Objeto (POO) é algo muito utilizado, não apenas por estar presente em diversos sistemas e aplicações, mas também por possuir um conceito bem estruturado e organizado. Pfleeger (2004, p.211) afirma que a orientação a objeto se caracteriza como um modo utilizado na criação de *software* no qual se dispõe de uma organização de problemas e soluções como uma agregação de objetos dissemelhantes. Para Sommerville (2003, p.221), o projeto orientado a objeto é uma técnica em que os desenvolvedores de aplicações consideram em termos de ‘coisas’, ao invés de termos técnicos como operações e funções.

A orientação a objeto consiste em criar um sistema relacionando ao mundo real utilizando classes, objetos, atributos e métodos. De acordo com Farinelli (2007, p.4), o intento da orientação a objeto é simular, o mais preciso possível, as ocorrências do mundo real nos sistemas computacionais. Nós percebemos como um todo constituído por diversos objetos que associam uns com os outros.

3.3.1 Classe

Em POO é utilizado o termo Classe para mostrar que todos os objetos são instanciados da classe, ou seja, todos os objetos terão as características da classe, que são os atributos e seus métodos. Para Ricarte (2001, p.4), “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”.

Os objetos são originados através de uma declaração de classe de objeto, no qual é visto como um molde para a concepção dos objetos. Essa classe contém declarações de atributos e procedimentos que devem ser vinculados ao objeto pertencente a esta classe (SOMMERVILLE, 2003, p.222).

3.3.2 Objeto

Os objetos, como já explanado, são instâncias da classe mãe e herdará tudo o que nela estiver. Instanciar neste contexto significa reservar ou alocar um espaço na memória do computador. Em poucas palavras, o termo objeto é o termo usado para representar um elemento do mundo real. Nas palavras de Ricarte (2001, p.5), “Um objeto é um elemento que

representa, no domínio da solução, alguma entidade (abstrata ou concreta) do domínio de interesse do problema sob análise”.

O objeto por sua vez, possui o termo “estado” para se referir ao grupo de atributos nos quais são herdados pela classe mãe, e esses estados são utilizados através de seus métodos. De acordo com (SOMMERVILLE, 2003, p.222), um objeto se refere a uma entidade no qual contém funções que são aplicadas de acordo com o estado que ele possui. Este estado é caracterizado pelo conjunto de atributos que o objeto possui.

3.3.3 Atributo e método

Como todo objeto real, os objetos criados através das classes também se dispõem de características e também de funções. As características são conhecidas como atributos e as funções como métodos. Os atributos como já dito, são construídos na classe e são compreendidos como características reais de um objeto criado pela classe. Um exemplo seria uma classe pessoa e seus atributos seriam o nome, idade, endereço, etc. Após a instância, o objeto automaticamente recebe todos os atributos situados na classe.

Os objetos do mundo real possuem propriedades que possuem valores. Estes valores definem o estado do objeto. As propriedades recebem o nome de atributos em OO. Podemos dizer que os atributos dos objetos são “variáveis” ou “campos” que armazenam os diferentes valores que as características dos objetos podem conter (FARINELLI, 2007, p.6).

Métodos por sua vez são as ações que o objeto, assim como acontece com os atributos, herdam da classe mãe. Todo método é entendido como ação na qual o objeto pode realizar. Para Farinelli (2007, p.7), tudo o que o objeto realiza é feito por meio de seus métodos, porquanto o objeto se inter-relaciona com outros objetos através de seus métodos. Nas palavras de Sierra, Bates (2010, p.15) “Os objetos possuem estados e comportamentos. O comportamento reside na classe, mas o estado reside dentro de cada objeto”.

3.3.4 Linguagem JAVA

Para que seja desenvolvido qualquer sistema, a linguagem é um fator muito importante, pois é com ela que todos os comandos, funções como um todo (comportamento do sistema) e a comunicação entre os módulos e o usuário são capazes de realizar as suas funções requeridas pelo usuário. Não existem sistemas sem a linguagem de programação,

sendo que toda a estrutura é montada através dela, pois ela determina um conjunto de regras a qual o sistema deverá seguir. Nas palavras de Pressman (2006, p.677), “As linguagens de programação são veículos de comunicação entre os seres humanos e os computadores”.

De acordo com Fontes, Silva (2008, p.85), nos últimos tempos, as linguagens de programação evoluíram amplamente. Elas oferecem uma extensa quantidade de recursos, que comportam desde a criação de programas simples para computações matemáticas até o desenvolvimento de sistemas computacionais com engenharia avançada, unificação de dados alfanuméricos, geográficos, fotográficos ou sonoros, nos quais podem ser lidos e registrados em qualquer base de dados do planeta. Dentre estas linguagens disponíveis, a linguagem Java é uma das que se destacam grandemente pelo fato de diversas páginas na *web* e aplicações a utilizarem e, conseqüentemente, nós necessitamos dela.

Java é uma linguagem de programação e plataforma computacional que foi lançada pela empresa Sun Microsystems pela primeira vez no ano de 1995. Existem diversas aplicações e sites que não irão funcionar, a não ser que se tenha o Java instalado, e a todo o momento, mais desses são desenvolvidas (JAVA, 2016).

Portanto, este projeto foi desenvolvido utilizando a linguagem Java, pois ela possui um vasto número de recursos que são essenciais em partes específicas e complexas.

3.4 Netbeans IDE

IDE (*Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado, em português) é um ambiente onde é possível criar aplicação, desde sistemas mais simples até os mais avançados, utilizando linguagens que são suportadas por ele, onde tudo o que é escrito é compreendido como instruções a serem compiladas, e assim, após sua depuração, é criada uma aplicação que pode ser entendida como um sistema. Existem hoje diversos IDEs gratuitos disponíveis e entre eles encontra-se o Netbeans.

Netbeans é um espaço de desenvolvimento, ou seja, uma ferramenta que permite aos desenvolvedores escrever, compilar, depurar e executar aplicações. Esta IDE foi totalmente escrita em linguagem Java, porém é possível utilizar qualquer linguagem para desenvolvimento, possuindo uma imensa quantidade de módulos para ampliar suas funcionalidades. O *Netbeans* IDE é também uma ferramenta livre e não possui restrições à sua forma de utilização (NETBEANS, 2016). Com o uso deste IDE foi possível usufruir ao máximo todas as funções necessárias para o melhor desempenho do sistema.

4 DESENVOLVIMENTO

Neste capítulo será descrito as partes que levaram ao desenvolvimento do sistema. Desde a coleta das informações até o seu resultado.

4.1 Estudo de viabilidade

O primeiro passo realizado para a construção da estrutura deste projeto foi realizar o estudo de viabilidade, onde foi possível perceber, não todos, mas alguns cenários possíveis após a sua implementação. O estudo ou análise de viabilidade, como já diz o nome, é o processo que nos permite visualizar se realmente será viável a sua existência no ambiente que será utilizado. Esta análise engloba desde o conhecimento possuído da pessoa que irá utilizar o sistema até custo que será gasto na sua instalação.

Mas não só isso, ela também nos possibilita visualizar alguns dos os fatores positivos e negativos após seu suposto uso. Ou seja, mostra quais serão os benefícios e prejuízos que o projeto poderá trazer através de seu uso. Sommerville (2003, p.103 - 104) diz, “Para todos os sistemas novos, o processo de engenharia de requisitos deve começar com o estudo de viabilidade. A entrada para o estudo de viabilidade é uma descrição geral do sistema e de como ele será utilizado dentro de uma organização”.

Com isso, os benefícios foram:

- Uma grande redução no tempo de realização do processo de votação;
- Redução em uso de papéis;
- Uma melhor organização e estrutura em relação a armazenamento de informação;
- Maior confiabilidade na contagem de votos;
- O sistema poderá criar uma nova cultura no ambiente de eleição;
- A instalação será simples e rentável, pois necessitaria de apenas alguns computadores e uso de aplicações gratuitas.

Os pontos negativos são:

- Para a utilização do sistema, seria necessário realizar um pequeno treinamento para os usuários futuros;
- É necessário transferir todos os dados para o banco de dados que o sistema

usará, ou seja, dados de alunos, cursos, funcionários que utilizarão o sistema e cargos em chapa. Apesar de alguns processos serem feitos apenas uma única vez, ainda assim utilizaria um tempo extenso;

- É necessário que tenha uma pessoa com um conhecimento básico sobre banco de dados e seus comandos para que alterações possam ser feitas na base de dados com o passar do tempo.

Com estes pontos observados, o sistema mostra ser adequado no ambiente em que ele será implementado.

4.2 Levantamento de Requisitos

O levantamento de Requisitos é uma etapa essencial em qualquer desenvolvimento, *desktop* ou *web*, os requisitos são as estruturas do sistema. Eles criam a separação do que é mais e menos importante na aplicação. Pressman (2006, p.231) defende que para o sucesso de um desenvolvimento, o total entendimento dos requisitos é fundamental. Mesmo que o *software* seja bem construído ou bem codificado, se os requisitos forem analisados ou identificados incorretamente, isso desengana o cliente ou usuário e, como consequência, trará problemas para o desenvolvedor.

Uma das ações mais relevantes em um sucesso de um *software* é o nível no qual ele auxilia os propósitos e requisitos para os quais foi desenvolvido. Em geral, a Engenharia de Requisitos de *Software* consiste no procedimento de reconhecer os envolvidos, identificar seus objetivos, precisão e arquivá-los apropriadamente para checar, comunicabilidade e futura implementação (FALBO, 2005, p.28).

É através desta etapa que resultará em um *feedback* não só do próprio sistema, mas também para quem o sistema será feito, pois os requisitos são nada mais do que partículas da aplicação que juntas formam o que foi solicitado pelo próprio usuário.

A análise de requisitos se divide em dois fatores importantes, nos quais são, o levantamento de requisitos funcionais e não funcionais de um sistema. Um sistema ou aplicação é dividido em módulos e esses módulos são desenvolvidos a partir dos requisitos para que, no final, eles se integrem formando um sistema absoluto.

Pfleeger (2004, p.111) explica que cada sistema possui sua funcionalidade, não importando se ela possa ser velha ou não, elas são evidentes em termos de produção do sistema. Um requisito é um aspecto do sistema ou uma especificação de alguma função que o

sistema pode fazer para alcançar seu objetivo.

4.2.1 Técnicas de levantamento de requisitos

Ao falar sobre levantamento de requisitos, é essencial que uma técnica seja utilizada para o melhor proveito desta etapa. Existem várias técnicas disponíveis, contudo será descrito neste projeto apenas a que realmente foi usada pelo autor, sendo ela, entrevista. Para Cassettari (2014, p.8), esta é a técnica mais frequentemente usada e compreensível. Resume-se em produzir uma prescrição e realizar a entrevista devidamente com os envolvidos e futuros usuários do sistema.

Esta técnica foi de grande ajuda ao coletar as informações necessárias para a criação do sistema. Primeiro foi feita uma entrevista com o analista de sistemas do local, após isto foi iniciado o desenvolvimento. Porém, a entrevista não tinha acabado. Foram realizadas perguntas que surgiram durante todo o processo, fazendo assim, uma entrevista contínua.

4.2.2 Requisitos funcionais

Os requisitos funcionais são basicamente as partes mais importantes do sistema. Com eles são feitas as funções mais complexas e essenciais de acordo com o que o cliente necessita. Os requisitos funcionais são especificamente o que o sistema deverá fazer diante de situações ocorrentes no sistema. Como e quando se deve fazer são partes interpretadas no levantamento.

Conforme Sommerville (2003, p.83) explica, requisitos funcionais são alegações que o sistema precisa proporcionar ou como o sistema precisa lidar a entradas precisas, como deve responder em certos eventos. Em algumas situações, os requisitos funcionais também são capazes de declarar claramente o que o sistema não deve executar. Pfleeger (2004, p.115) aponta, “Um requisito funcional descreve uma interação entre o sistema e seu ambiente”.

Os requisitos funcionais deste projeto são:

- O sistema deverá ser capaz de realizar as votações de Diretórios Acadêmicos;
- Não deverá haver conexão alguma entre as chapas e o aluno que votou. As chapas apenas deverão receber os votos;
- Para o processo de votação, algumas regras deverão ser implementadas, tais como:
 - Para que se inicie a votação, é necessário que todos os cargos obrigatórios das chapas tenham sido preenchidos.

- Para que se tenha um resultado oficial, é necessário que a votação obtenha 50% + 1 de votos. Este cálculo é feito através do número total de alunos do curso envolvido;
- É necessário fazer a separação entre os alunos que já votaram e os que não;
- A parte dos cargos das chapas deverá possuir a opção de mudar sua obrigatoriedade;
- É necessário criar um módulo chamado Resultados, pois os resultados serão vistos diversas vezes;
- Realizar o cadastro de todos os dados necessários para a realização das funções principais;
- Criar um controle de usuários, normal e administrador, para a separação de privilégios e registro de ações no sistema;
- Criar um usuário administrador oculto como modo de segurança;
- Os privilégios se distinguirão através de ações disponíveis, onde o administrador será capaz de visualizar, cadastrar, editar e excluir os seguintes módulos: Funcionários, usuários, cursos, alunos, chapas e eleições (apenas cadastro e exclusão).

4.2.3 Requisitos não funcionais

Os requisitos não funcionais são requisitos que não são tão necessários para o funcionamento do sistema. Tais elementos são implementados após o desenvolvimento dos requisitos funcionais. Para Pfleeger (2004, p.115), os requisitos não funcionais, ou conhecidos também como restrições, retratam uma restrição no sistema que faz com que as possibilidades de gerar uma solução para os problemas se tornem limitadas.

Com eles, o sistema é incorporado com grandes quantidades de detalhes que, dependendo do nível de precisão do levantamento de requisitos, pode satisfazer principalmente as necessidades do cliente. Os requisitos não funcionais, como propõe o nome, são aqueles que não possuem relação de modo direto com as funcionalidades específicas concedidas pelo sistema. Eles podem estar associados com a característica de sistemas emergentes, como tempo de resposta, confiabilidade e espaço em disco (SOMMERVILLE, 2003, p.85).

Os requisitos não funcionais são:

- Criar telas para cada função do sistema. Por exemplo: A parte usuário possui a de visualização onde todas as funções como, cadastro e outros. Contudo, para projetos futuros, foi solicitada uma tela para cada função caso precise disponibilizar para o usuário tipo normal;
- A votação não deve ser necessariamente quando criado a eleição, podendo ser possível realizar a votação no dia que se achar conveniente;
- Adicionar imagens da FAFRAM na tela principal e seu símbolo como ícone em cada tela;
- O Administrador deverá ter a disponibilidade de alterar o usuário normal para usuário administrador, ficando toda a responsabilidade para o indivíduo que assim o executar;
- Para realizar funções mais sérias no sistema, será necessário inserir a senha do usuário que tenha realizado o *login*;
- Em todas as telas que se tenha um número elevado de informações em tabelas, uma opção de busca deve ser implementada;
- Implementar a opção para criar um relatório do resultado com todos os dados da eleição e votação.

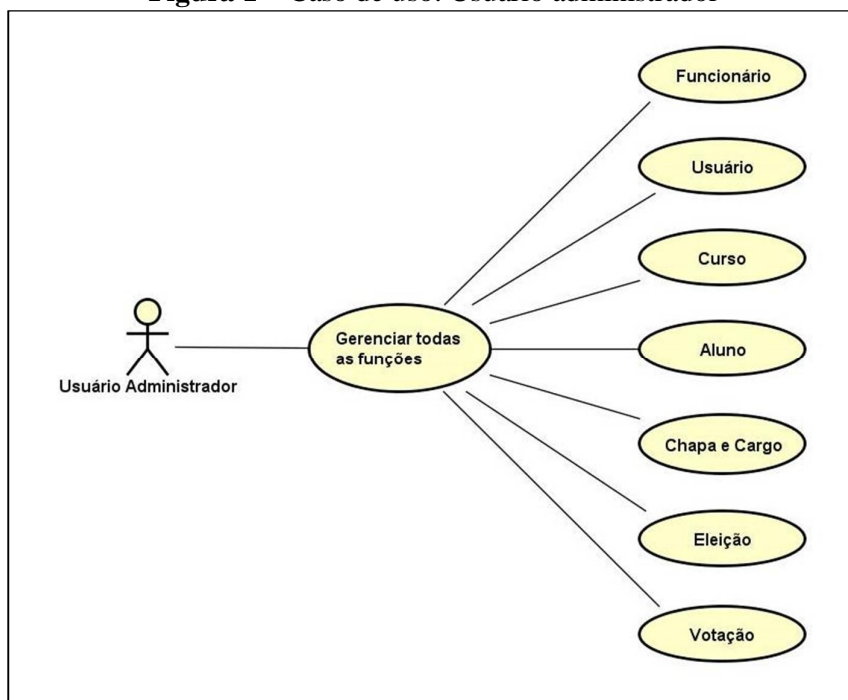
4.3 Diagramas

4.3.1 Caso de uso

Os diagramas de casos de uso possuem a finalidade de explicar ao cliente, em forma visual, como o sistema comportará. Ele mostra as funções que os envolvidos no sistema poderão realizar de forma simples. Neste projeto, os casos de uso foram divididos em três partes, nas quais são: Usuário administrador, usuário normal e aluno.

4.3.1.1 Usuário Administrador

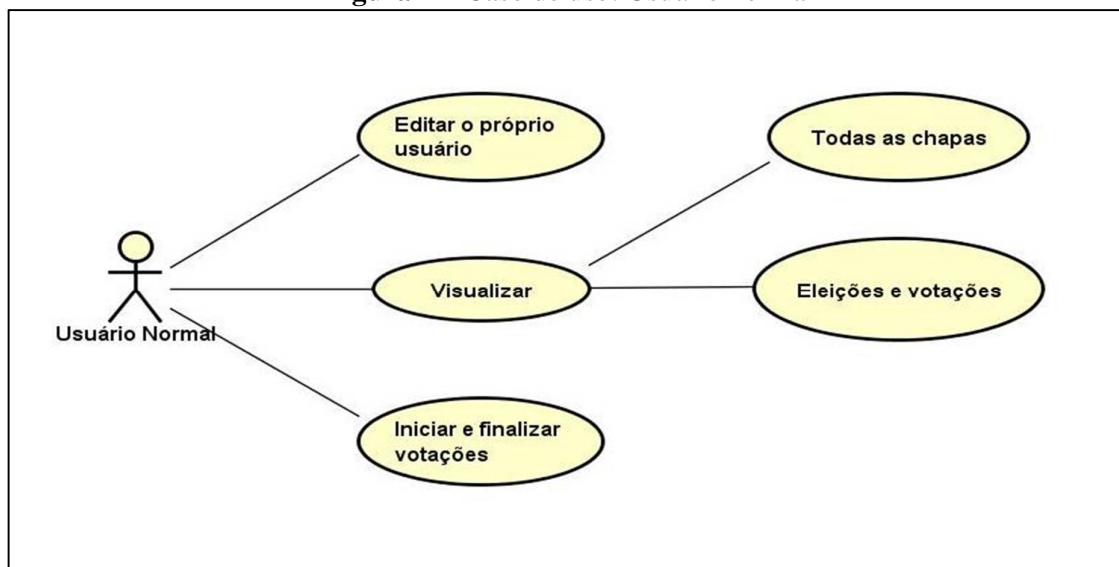
Este diagrama representa as funcionalidades disponíveis ao usuário administrador do sistema, sendo possível enxergar de maneira ilustrativa aos usuários/clientes futuros do sistema. O usuário administrador possui acesso total ao sistema.

Figura 1 – Caso de uso: Usuário administrador

Fonte: Elaborado pelo autor (2016)

4.3.1.2 Usuário Normal

Este diagrama não é diferente do mostrado acima, pois ele representa as funções disponíveis para o usuário, contudo, a imagem abaixo é voltada ao usuário normal.

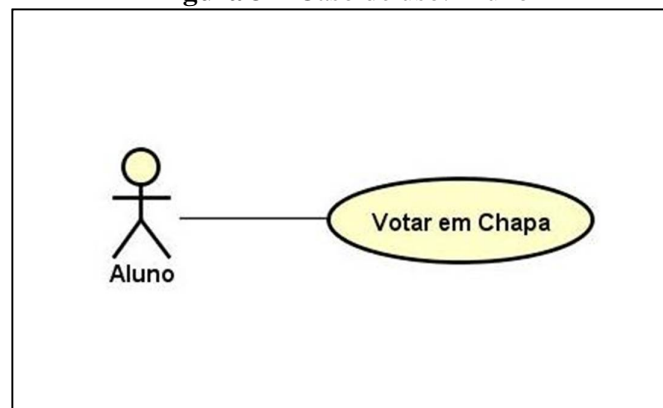
Figura 2 – Caso de uso: Usuário normal

Fonte: Elaborado pelo autor (2016)

4.3.1.3 Aluno

E por fim, o próximo diagrama mostra as funções disponíveis para o usuário aluno, onde possui apenas a funcionalidade de votar na chapa.

Figura 3 – Caso de uso: Aluno

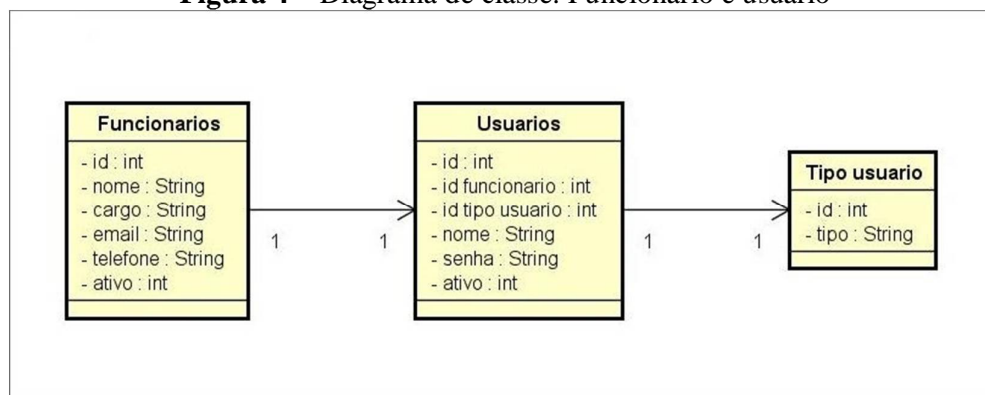


Fonte: Elaborado pelo autor (2016)

4.3.2 Diagrama de classe

Os diagramas de classe são elementos para representação visual de componentes de um sistema, mais especificamente tabelas e entidades do modelo de dados. O diagrama a seguir mostrar como é feito o cadastro de funcionários e usuários. Nesta figura é possível ver que os funcionários podem apenas ter um usuário e um tipo.

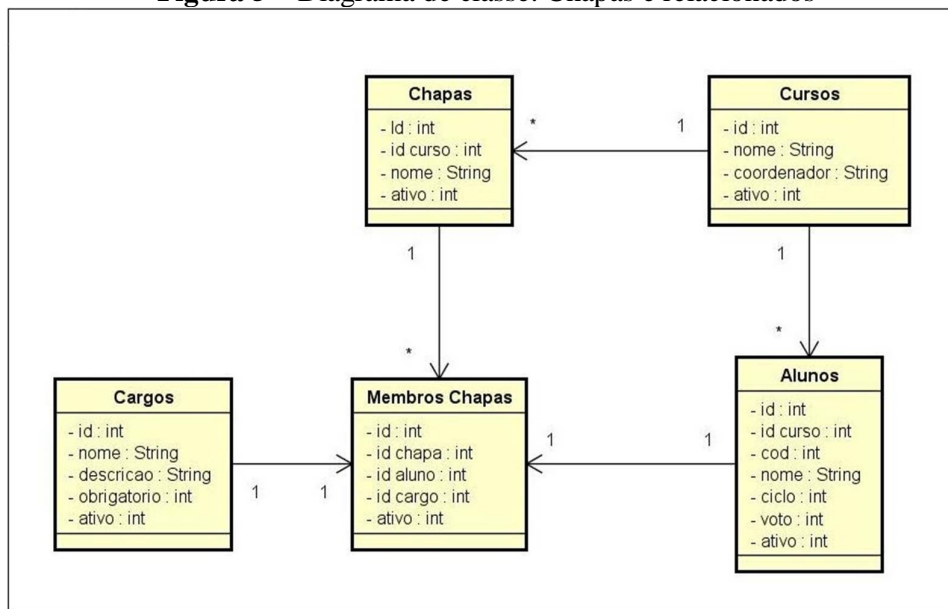
Figura 4 – Diagrama de classe: Funcionário e usuário



Fonte: Elaborado pelo autor (2016)

A figura abaixo representa os processos que envolvem o ambiente de chapas e seus membros. A classe membros chapas representa cada aluno inserido na chapa com seu cargo.

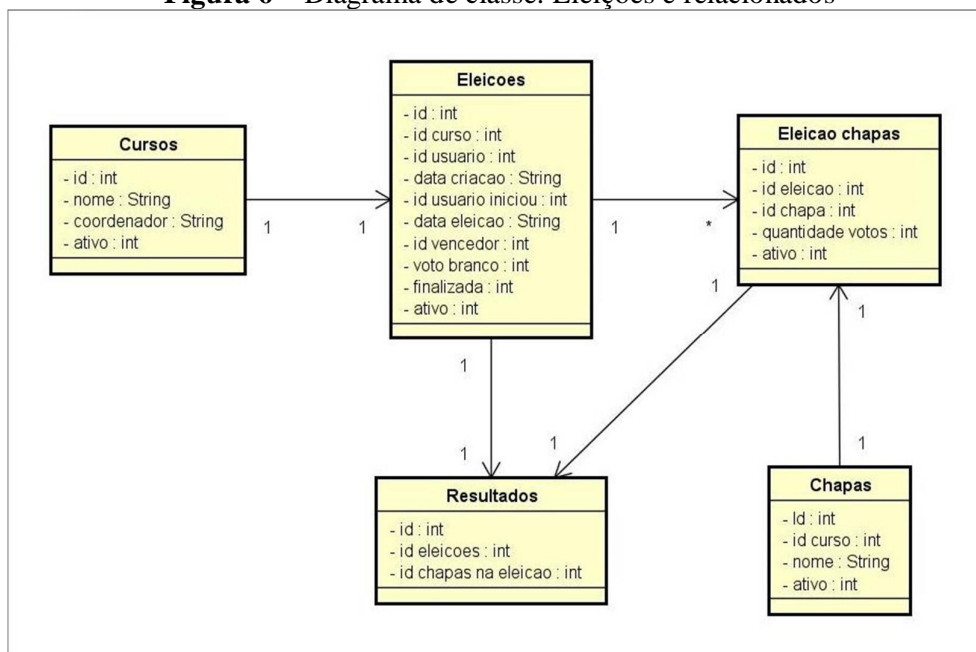
Figura 5 – Diagrama de classe: Chapas e relacionados



Fonte: Elaborado pelo autor (2016)

E por fim, abaixo esta o diagrama que representa o processo de eleição onde a classe Eleição chapas representam as chapas na eleição.

Figura 6 – Diagrama de classe: Eleições e relacionados

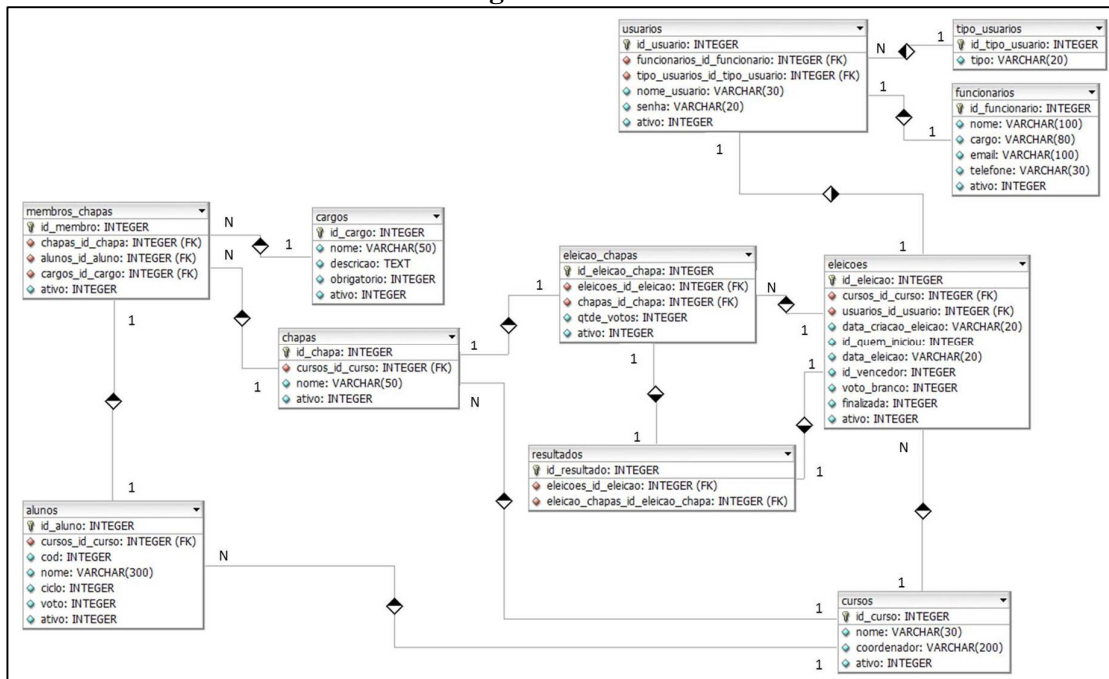


Fonte: Elaborado pelo autor (2016)

4.3.4. Modelo de Entidade-Relacionamento (MER)

Os MER possui apenas o objetivo de representar a estrutura do banco de dados em forma visual, onde as tabelas são conhecidas como entidades, e cada dado nessas tabelas são atributos. A figura seguinte representa como é a base de dados do sistema desenvolvido.

Figura 7 – MER



Fonte: Elaborado pelo autor (2016)

Neste modelo é possível ver a ligação na tabela resultados, onde todos os dados inseridos nela são apenas chaves estrangeiras da tabela ‘eleição chapas’ e da tabela ‘eleições’. Com isso, o banco de dados não armazena dados repetidos, ou seja, que podem estar em outras tabelas. Também é possível visualizar a tabela ‘membros chapas’ no qual possui três chaves estrangeiras. Cada linha desta tabela representa um aluno e seu cargo pertencente a chapa.

5 RESULTADOS

Com o uso de todas as ferramentas já citadas, foi criado o sistema. Neste capítulo será mostrado, como resultado, o projeto de interface do sistema desenvolvido.

O projeto de interface mostra como sistema é em modo visual e mostra também o contato entre o usuário e a aplicação. De acordo com Sommerville (2003, p.278), um projeto de interface agradável para com o usuário é essencial para o bom êxito de um sistema.

Uma interface que possui uma complexidade em ser manipulada terá no mínimo como efeito, um número elevado de erros proporcionado pelo usuário. É através da interface que o sistema poderá receber as entradas que a pessoa deseja inserir ou as informações que deseja retirar. Com uma interface amigável e organizada, o usuário poderá usufruir ao máximo o que o sistema pode oferecer.

O primeiro contato após o *login* é a *Home*. A figura 8 representa a tela inicial. Esta tela possui um menu com funções desde cadastro até exclusão para cada paleta. Nela mostra também uma mensagem notificando se há votações em andamento ou não.

Figura 8 – Tela Inicial do usuário tipo: Administrador



Fonte: Elaborado pelo autor (2016)

A figura 9 nos mostra a tela de visualização onde é possível realizar as funções de cadastro, edição e exclusão de funcionários, contendo também, a opção de procura na parte superior. Lembrando que cada função possui sua própria tela.

Figura 9 – Tela de funcionários com todas as funções

Visualização de Funcionários

Pesquisar por: Todos

Todos os funcionários:

ID	Nome	Cargo	E-mail	Telefone
2	FUNCIONARIO 1	ANALISTA DE SISTEMAS	func1@hotmail.com	(16) 99999 9999
3	FUNCIONÁRIO 2	PROGRAMADOR	func2@hotmail.com	(16) 99999 9999
4	FUNCIONÁRIO 3	ATENDENTE	func3@hotmail.com	(16) 99999 9999
5	FUNCIONÁRIO 4	PROGRAMADOR WEB	func4@hotmail.com	(16) 99999 9999

Informações:

ID: 2

* Nome: FUNCIONÁRIO 1

* Cargo: ANALISTA DE SISTEMAS

E-mail: func1@hotmail.com

Telefone: (16) 99999 9999

Cadastrar
Editar
Excluir
Voltar

Fonte: Elaborado pelo autor (2016)

Podemos ver na figura 10 o código utilizado para carregar praticamente todas as tabelas do sistema. Neste exemplo, é mostrada a tabela de funcionários.

Figura 10 – Código para o preenchimento das tabelas

```

void carregatb() {
    while (tbFuncionario.getRowCount() > 0) {
        DefaultTableModel tableModel = (DefaultTableModel) tbFuncionario.getModel();
        tableModel.setNumRows(0);
    }
    List<String[]> lista;
    FuncionarioDAO daoPesquisa = new FuncionarioDAO();
    lista = daoPesquisa.pesquisa_geral();
    try {
        model = (DefaultTableModel) tbFuncionario.getModel();
        lista.stream().forEach((lista1) -> {
            model.addRow(lista1);
        });
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

Fonte: Elaborado pelo autor (2016)

A figura 11 representa o código feito para a verificação antes da exclusão de algum dado no sistema. Nela é possível ver a função onde é verificado se a pessoa que requereu a exclusão esta de acordo com a senha inserida e com o usuário ativo no momento.

Figura 11 – Código de verificação para exclusão e outros

```

public int verifica_seSenhaDoAdm(Usuario user){
    String sql = "select id_usuario from usuarios where ativo > 0 and tipo_usuarios_id_tipo_usuario = 2 "
        + "and funcionarios_id_funcionario = "+user.getId_fun()+" and senha = '"+user.getSenha()+"'";
    try{
        Statement con = conexao.createStatement();
        ResultSet rs = con.executeQuery(sql);
        int valor = 0;
        if(rs.next()){
            valor = rs.getInt("id_usuario");
        }
        conexao.close();
        return valor;
    }
    catch(SQLException u){
        throw new RuntimeException( u );
    }
}

```

Fonte: Elaborado pelo autor (2016)

No módulo Chapa, a figura 12 representa a tela de visualização de todas as chapas onde o usuário pode navegar entre os cursos, e ao clicar na chapa, os alunos pertencentes a ela aparecerem na tabela abaixo. Ainda, esta tela disponibiliza a opção para inserir alunos.

Figura 12 – Tela de Chapas com todas as funções

ID	Nome da Chapa
1	S_AGRONOMOS
2	CHAPA DO SERTÃO
3	AGRO+

ID	Código	Aluno	Cargo
27	27	ANASTASIA C. RYAN	PRESIDENTE
213	213	ANTONIETA XABLAU DINIZ	VICE-PRESIDENTE
6	6	ANTHONY L. SNOW	SECRETÁRIO GERAL
1	1	ANTHONY D. SINCLAIR	VICE-SECRETÁRIO
25	25	BETSY A. DOLLAR	TESOUREIRO GERAL
28	28	CASEY K. WHITTINGTON	VICE-TESOUREIRO
20	20	CELESTE E. SANCHEZ	DIRETOR DE ASSUNTOS ACADÊ...
33	33	CAROL C. MATTHEWS	VICE-DIRETOR DE ASSUNTOS A...
40	40	CLAUDIA MENEZES	DIRETOR DE EXTENSÃO

Fonte: Elaborado pelo autor (2016)

Ainda no módulo Chapa, a figura 13 mostra a tela em que os alunos são inseridos na chapa selecionada. Nela é possível ver os dados da chapa, cargos disponíveis, total de alunos disponíveis e se a chapa está completa.

Figura 13 – Tela de inserção de alunos em chapas

Adição e Exclusão de Alunos/Cargos para Chapas

ID: 13 Nº de alunos disponíveis: 35
 Curso: SISTEMAS DE INFORMAÇÃO Cargos disponíveis: 15/15
 Chapa: SÓ SISTEMAS Chapa completa: NÃO

Pesquisar por: Todos

Alunos do curso selecionado:

ID	Código	Nome	Ciclo
193	55555	ALAYNA CHARPENTIER	9
204	55555	ASHLI WEDELL	9
205	55555	BERNARDO CHRISTIANO	9
206	55555	BRENTON LADOUCEUR	9
212	55555	BRIAN WEATHERSPOON	9
183	17283	BRUNO AGUIAR	2
199	55555	CHASE MARLOWE	9
198	55555	COL EMAN HYNEMAN	9

Cargos disponíveis:

ID	Cargo	Descrição	Obrigatório
1	PRESIDENTE	CARGO RESPONSÁVEL...	SIM
2	VICE-PRESIDENTE	CARGO RESPONSÁVEL...	SIM
3	SECRETÁRIO GERAL	CARGO RESPONSÁVEL...	SIM
4	VICE-SECRETÁRIO	CARGO RESPONSÁVEL...	NÃO
5	TESOUREIRO GERAL	CARGO RESPONSÁVEL...	NÃO
6	VICE-TESOUREIRO	CARGO RESPONSÁVEL...	NÃO
7	DIRETOR DE ASSUNTO...	CARGO RESPONSÁVEL...	NÃO
8	VICE-DIRETOR DE ASS...	CARGO RESPONSÁVEL...	NÃO

ID: 212 Obrigatório: Sim Não
 Código: 55555
 Nome: BRIAN WEATHERSPOON
 Ciclo: 9

ID: 1 Obrigatório: Sim Não
 Cargo: PRESIDENTE
 Descrição: CARGO RESPONSÁVEL POR ...

Cancelar Remover Alunos Adicionar na Chapa

Fonte: Elaborado pelo autor (2016)

Abaixo é possível ver os códigos usados para a realização de algumas funções da imagem acima. Na figura 14 está a *query* usada para a inserção dos alunos na chapa juntamente com o cargo escolhido, enquanto a figura 15 mostra o código para o carregamento da tabela dos cargos disponíveis.

Figura 14 – Código de inserção de alunos na chapa

```

public void cadastrarMembros(int id_chapa, int id_aluno, int id_cargo){
    String sql = "insert into membros_chapas (chapas_id_chapa,alunos_id_aluno,cargos_id_cargo,ativo) "
        + "values (?, ?, ?, 1)";

    try{
        try (PreparedStatement stmt = conexao.prepareStatement(sql)) {
            stmt.setInt(1, id_chapa);
            stmt.setInt(2, id_aluno);
            stmt.setInt(3, id_cargo);
            stmt.execute();
        }
        conexao.close();
    }
    catch (SQLException u){
        throw new RuntimeException(u);
    }
}

```

Fonte: Elaborado pelo autor (2016)

Figura 15 – Código da tabela de cargos disponíveis

```

public List<String[]> carregathCargo(int idCurso, int idChapa){
    String sql = "select id_cargo,nome,descricao,obrigatorio from cargos where ativo = 1 and id_cargo not in "
    + "(select distinct mc.cargos_id_cargo from membros_chapas mc inner join "
    + "chapas c on (c.id_chapa = mc.chapas_id_chapa) inner join "
    + "cargos ca on (ca.id_cargo = mc.cargos_id_cargo) inner join "
    + "alunos a on (mc.alunos_id_aluno = a.id_aluno) inner join cursos cu on (cu.id_curso = a.cursos_id_curso) "
    + "where mc.ativo = 1 and c.ativo = 1 and ca.ativo = 1 and cu.id_curso = "+idCurso+" and c.id_chapa = "+idChapa+" ) "
    + "order by id_cargo";
    List <String[]> lista = new ArrayList<>();
    try{
        Statement con = conexao.createStatement();
        try (ResultSet rs = con.executeQuery(sql)) {
            conexao.close();
            int columnCount = rs.getMetaData().getColumnCount();
            while(rs.next())
            {
                String[] row = new String[columnCount];
                for (int i=0; i <columnCount ; i++)
                {
                    row[i] = rs.getString(i+1);
                }
                lista.add(row);
            }
            rs.close();
        }
    }
}

```

Fonte: Elaborado pelo autor (2016)

A interface Cargo visto na figura 16 é de fato uma parte breve do sistema, porém, uma parte essencial. É nele onde o usuário adicionará, editará e excluirá os cargos em chapas. Esta parte possui uma função bem interessante, visto que o usuário pode alterar uma característica muito importante, o atributo “Obrigatório”.

Esta função será usufruída de acordo com a instituição de ensino e é crucial saber usá-la, já que ele só pode ser alterado quando não há votação em andamento. Apenas este módulo não foi feito uma tela para cada função.

Figura 16 – Tela de Cargos com todas as funções

Informações:

* Cargo: PRESIDENTE

* Descrição: CARGO RESPONSÁVEL POR ...

* Obrigatório: Este cargo é obrigatório em todas as chapas.

Cargos cadastrados:

ID	Cargo	Descrição	Obrigatório
1	PRESIDENTE	CARGO RESPONSÁVEL POR ...	SIM
2	VICE-PRESIDENTE	CARGO RESPONSÁVEL POR ...	SIM
3	SECRETÁRIO GERAL	CARGO RESPONSÁVEL POR ...	SIM
4	VICE-SECRETÁRIO	CARGO RESPONSÁVEL POR ...	NÃO
5	TESOUREIRO GERAL	CARGO RESPONSÁVEL POR ...	NÃO
6	VICE-TESOUREIRO	CARGO RESPONSÁVEL POR ...	NÃO
7	DIRETOR DE ASSUNTOS ACA...	CARGO RESPONSÁVEL POR ...	NÃO
8	VICE-DIRETOR DE ASSUNTO...	CARGO RESPONSÁVEL POR ...	NÃO
9	DIRETOR DE EXTENSÃO	CARGO RESPONSÁVEL POR ...	NÃO
10	DIRETOR DE CULTURA E ES...	CARGO RESPONSÁVEL POR ...	NÃO

Fonte: Elaborado pelo autor (2016)

Agora, serão mostrados os módulos de eleição e votação. A figura 17 abaixo nos proporciona uma visão clara de como o sistema trata esta parte. Nela podemos ver que as chapas possuem informações que ajudam o usuário entender o que está acontecendo no momento, mas o foco está na informação “*Status*”.

Os *status* explicam que não é possível adicionar a chapa na eleição, pois a chapa não acomoda todos os cargos obrigatórios naquele momento. Na figura podemos ver que a chapa “EU AMO ANIMAIS” está disponível para ser adicionada, pois abriga os cargos obrigatórios. Além destas informações, podem-se observar abaixo as opções para remover a chapa da eleição ou iniciá-la.

Figura 17 – Tela de inserção e remoção de chapas na eleição

Chapas disponíveis:

ID	Chapa
10	CHAPA ANIMAL
12	EU AMO ANIMAIS

Curso: MEDICINA VETERINÁRIA
ID: 12
Chapa: EU AMO ANIMAIS
Status: Disponível para ser adicionada

Alunos na chapa:

ID	Código	Aluno	Cargo
138	11899	DEBORA FILHO	PRESIDENTE
137	30021	GILBERTO LUNES	VICE-PRESIDENTE
170	99999	DENAE OBRIEN	SECRETÁRIO GERAL
145	66666	JOSÉ JOAQUIM	VICE-SECRETÁRIO
147	88888	JOSÉ LUIZ	TESOUREIRO GERAL
176	99999	JOESPH THIESSEN	VICE-TESOUREIRO
156	99999	ISABELL PERRIER	DIRETOR DE ASSUNTOS A...
149	99999	IVEY CHAPLIN	VICE-DIRETOR DE ASSUNT...
171	99999	KELLIE ROTHMAN	DIRETOR DE EXTENSÃO
159	99999	LAURINDA BUGG	DIRETOR DE CULTURA E E...

Adicionar Chapa

ID da Eleição: 3
Curso: MEDICINA VETERINÁRIA
Criada por: FUNCIONÁRIO 1
Data de criação: 29/07/2016 02:28:49

Chapas adicionadas na eleição:

ID	Chapa
4	DOGS&CIA

Voltar Remover Iniciar Votação

Fonte: Elaborado pelo autor (2016)

Após iniciada a votação, a figura 18 mostra o momento em que o responsável pela votação faz o controle de alunos. O sistema mostra na tabela todos os alunos do curso selecionado. Nesta tela o usuário pode visualizar todos os dados da eleição incluindo as chapas participantes e é nela que a tela para votação é liberada.

Figura 18 – Tela de controle de votação

Dados da Eleição:
 ID: 2
 Curso: SISTEMAS DE INFORMAÇÃO
 Data de Criação: 29/07/2016 02:19:28
 Criada por: FUNCIONÁRIO 1
 Date da Votação: 29/07/2016 02:39:50
 Votação iniciada por: FUNCIONÁRIO 1

Chapas nesta eleição:

ID	Chapas
14	SISTEMÁTICOS
15	EU E JAVA

Alunos que não votaram: Todos

ID	Código	Nome	Ciclo
193	55555	ALAYNA CHARPENTIER	9
204	55555	ASHLI WEDELL	9
205	55555	BERNARDO CHRISTIANO	9
206	55555	BRENTON LADOUCEUR	9
212	55555	BRIAN WEATHERSPOON	9
183	17283	BRUNO AGUIAR	2
199	55555	CHASE MARLOWE	9
198	55555	COLEMAN HYNEMAN	9
211	55555	CRISTIN ARENS	9
203	55555	CYNDI TIGGS	9
209	55555	DELORES HOLLOW	9
200	55555	DEWEY ELEY	9
178	63936	DOUGLAS FERREIRA	6

ID: 203
 Código: 55555
 Nome: CYNDI TIGGS

Buttons: Voltar, Selecionar, Finalizar Votação

Fonte: Elaborado pelo autor (2016)

A figura 19 representa o momento após a seleção do aluno. Esta tela é uma tela dinâmica, ou seja, ela se ajustará de acordo com o número de chapas na votação. Neste exemplo, há apenas duas chapas em votação. O sistema manda esta tela para outro computador, onde estará disponível apenas para o aluno, porém isto foi deixado como trabalhos futuros. Esta tela esta integrada no próprio sistema, no entanto, a votação funciona normalmente.

Figura 19 – Tela para votar

Tela de Votação

Selecione a chapa desejada:

SISTEMÁTICOS

EU E JAVA

Voto em Branco

Button: Votar

Fonte: Elaborado pelo autor (2016)

Na figura 20 é mostrado o código de como a tela acima é construída.

Figura 20 – Código da tela de votação

```
private void criaRadios(){
    try{
        int num = chapas.size(), x = 40, y = 50;
        JRadioButton b;

        for(int i=0;i<num;i++){
            b = new JRadioButton();
            b.setBounds(x, y, 220, 15);
            b.setText(chapas.get(i));
            buttons.add(b);
            y += 30;
            getContentPane().add(b);
            bg.add(b);
        }
        branco = new JRadioButton();
        branco.setBounds(x, y, 220, 15);
        branco.setText("Voto em Branco");
        branco.setName("radioBranco");
        buttons.add(branco);
        getContentPane().add(branco);
        bg.add(branco);
        y += 100;
        setSize(400, y);
        btnVotar.setLocation(320, y-10);
        setResizable(false);
    }
}
```

Fonte: Elaborado pelo autor (2016)

Ao final do processo, o usuário deverá finalizar a votação na opção mostrada na figura 18. Depois de finalizada, o sistema mostra os resultados da votação. Estes resultados são representados pela figura 21 abaixo.

Figura 21 – Tela de Resultados

The screenshot shows a window titled 'Resultados de Votações'. It contains a table of candidates and their vote counts, followed by election details and a list of statistics. At the bottom, there are two buttons: 'Sair' and 'Relatório em TXT'.

Chapas nesta eleição:			
ID	Chapas	Total de Votos	% de votos
15	EU E JAVA	19	54,29
14	SISTEMÁTICOS	10	28,57

Dados da Eleição:
 Curso: SISTEMAS DE INFORMAÇÃO
 Data de criação: 29/07/2016 02:19:28
 Criada por: FUNCIONÁRIO 1
 Date da votação: 29/07/2016 02:39:50
 Iniciada por: FUNCIONÁRIO 1
 Total de Alunos: 35
 Votos em Chapas: 29
 Votos em Branco: 6
 Total de votos: 35 em %: 100,00

Chapa Vencedora: **EU E JAVA**

Sair Relatório em TXT

Fonte: Elaborado pelo autor (2016)

A figura 22 representa parte do código de finalização de uma votação.

Figura 22 – Código de finalização de votação

```
public Boolean finalizaVotacao(int id_eleicao, int id_curso){
    PreparedStatement updateEleicao = null;
    PreparedStatement updateAluno = null;
    String sqlEleicao = "update eleicoes set finalizada = 1 where id_eleicao = "+id_eleicao;
    String sqlAluno = "update alunos set voto = 0 where cursos_id_curso = "+id_curso;
    try {
        conexao.setAutoCommit(false);
        updateEleicao = conexao.prepareStatement(sqlEleicao);
        updateAluno = conexao.prepareStatement(sqlAluno);
        updateEleicao.executeUpdate();
        updateAluno.executeUpdate();
        conexao.commit();
        return true;
    }
    catch (Exception e) {
        if(conexao != null){
            try {
                conexao.rollback();
            }
            catch(SQLException excep) {
                throw new RuntimeException(excep);
            }
        }
        return false;
    }
}
```

Fonte: Elaborado pelo autor (2016)

Outra função que o sistema também possui é a criação de relatórios. Após salvo, o sistema cria o arquivo na pasta do projeto com o nome desejado, neste projeto ele é criado na pasta de projetos Netbeans. A figura 23 mostra como são os relatórios.

Figura 23 – Tela do relatório de resultados

```
----- Dados da Eleição -----
Curso_____ : SISTEMAS DE INFORMAÇÃO
Data da Criação_____ : 29/07/2016 02:19:28
Criada por_____ : FUNCIONÁRIO 1
Data da Votação_____ : 29/07/2016 02:39:50
Iniciada por_____ : FUNCIONÁRIO 1
Total de alunos no curso__ : 35
Total de votos em chapas__ : 29
Total de votos em branco__ : 6
Total de votos_____ : 35 Total em %: 100,00

----- Chapas e votos -----
Chapa: EU E JAVA Votos: 19
Chapa: SISTEMÁTICOS Votos: 10
-----

*Chapa Vencedora_____ : EU E JAVA
```

Fonte: Elaborado pelo autor (2016)

CONCLUSÃO

Ao analisar constantemente o uso do sistema através dos testes feitos pelo autor e desenvolvedor, foi possível ver alguns pontos em que o sistema trará o resultado desejado e outros em que ainda poderão ser feitos, nos quais são: a utilização de dispositivos móveis para a realização dos votos e criação de um sistema web para melhor praticidade, porém serão deixados como projetos futuros.

O sistema possui um grande potencial para gerenciamento não só da votação em si, mas também em toda a parte necessária para que ela seja feita, sendo um *software* que não exigirá um conhecimento avançado em informática por ser autoexplicativo e não demandará um custo avançado para sua instalação.

Integrando os resultados do sistema com os resultados da análise de viabilidade e os requisitos levantados, o sistema de gestão de Diretórios Acadêmicos mostra-se apto a cumprir o que foi proposto pelo cliente e resultará em pontos positivos para a instituição onde for implementado.

REFERÊNCIAS

- ACETI, P. A. Z. **Desenvolvimento de um plano de ações para informatização usando software livre em empresas de pequeno porte de um cluster regional**. 2006. 121 f. Dissertação (Pós-Graduação) - Curso de Engenharia de Produção, Faculdade de Engenharia, Arquitetura e Urbanismo, da Universidade Metodista de Piracicaba – Unimep, Santa Bárbara D’oeste, 2006. Disponível em: <https://www.unimep.br/phpg/bibdig/pdfs/docs/28112012_213122_patricia_aparecida_zibordi_aceti.pdf>. Acesso em: 30 ago. 2016.
- BOCHNER, R. et al. **Qualidade da informação: A importância do Dado Primário, o princípio de tudo**. In: Encontro nacional de pesquisa em ciência da informação, 12., 2011, Distrito Federal. Comunicação Oral. Brasília: Enancib, 2011. p. 3526 – 3538. Disponível em: <<http://inseer.ibict.br/ancib/index.php/tpbci/article/viewFile/58/95>>. Acesso em: 07 set. 2016.
- BONFIOLI, G. F. **Banco de dados relacional e objeto-relacional: Uma comparação usando PostgreSQL**. 2006. 62 f. Trabalho de Conclusão de Curso (Graduação) - Curso de Ciência da Computação, Universidade Federal de Lavras, Lavras, 2006. Disponível em: <http://novo.more.ufsc.br/tese_dissert/insserir_tese_dissert>. Acesso em: 14 ago. 2016.
- CASSETTARI, P. B. **Banco de questões para levantamento de requisitos de sistema financeiro orçamentário público**. 2014. Disponível em: <https://www.softplan.com.br/tratadoaprendizagem/wp-content/files_mf/1417173374TCC_PatrickBorgesCassettariArtigo.pdf>. Acesso em: 28 jul. 2016.
- CÔRTEZ, S. C.; LUCENA, C. J. P. **Um framework de regras ativas para sistemas de gerência de banco de dados distribuído**. 2001. Disponível em: <ftp://ftp.inf.puc-rio.br/pub/docs/techreports/01_16_cortes.pdf>. Acesso em: 26 jul. 2016.
- COSTA, A.D.M. da; JABUR, A.M. R. T. **Maçonaria e Educação: A experiência da Fundação Educacional de Ituverava – SP**. 2010. **Nucleus**. v.7, n.1. Ituverava, SP. Disponível em: <<http://www.nucleus.feituverava.com.br/index.php/nucleus/article/view/339/411>>. Acesso em: 21 ago. 2016.
- FAFRAM – FACULDADE DR. FRANCISCO MAEDA. **Fundação Educacional de Ituverava completa 41 anos. 2015**. Disponível em: <www.fafram.com.br/index.php/campus/4-fundacao-educacional-de-ituverava-completa-41-anos>. Acesso em: 15 Mar. 2016.
- _____. **Histórico**. 2016. Disponível em: <www.fafram.com.br/index.php/campus/4-fundacao-educacional-de-ituverava-completa-41-anos>. Acesso em: 26 jul. 2016.
- FALBO, R. de A. **Engenharia de Software**. 2005. Disponível em: <<http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/NotasDeAula.pdf>>. Acesso em: 14 ago. 2016.

FARINELLI, F. **Conceitos básicos de programação orientada a objetos**. 2007. Disponível em: <http://www.marcelohsantos.com.br/aulas/downloads/apostilas/Apostila_Java_B.pdf>. Acessado em: 27 jul. 2016.

FONTES, C.R.; SILVA, F. W. O. da. O ensino da disciplina linguagem de programação em escolas técnicas. **Ciências & Cognição**, Belo Horizonte, v. 13, n. 2, p.84-98, jun. 2008. Disponível em: <<http://www.cienciasecognicao.org/revista/index.php/cec/article/view/221/120>>. Acesso em: 27 jul. 2016.

JAVA. **O que é Java?**. Disponível em: <https://www.java.com/pt_BR/download/faq/whatis_java.xml>. Acesso em: 28 ago. 2016.

MATSUMOTO, C. Y. A importância do banco de dados em uma organização. **Maringá Management: Revista de Ciências Empresariais**, Maringá, v. 3, n. 1, p.45-55, jun. 2006. Disponível em: <<http://www.maringamanagement.com.br/novo/index.php/ojs/article/view/58/30>>. Acesso em: 19 nov. 2016.

NETBEANS. **History**. Disponível em: <https://netbeans.org/index_pt_PT.html>. Acesso em: 27 jul. 2016.

PFLEEGER, S. L. **Engenharia de Software: Teoria e prática**. 2. ed. São Paulo: Pearson, 2004. Dino Franklin.

PostgreSQL. **History**. 2013. Disponível em: <<https://www.postgresql.org/about/history/>>. Acesso em: 27 jul. 2016.

PRESSMAN, R. S. **Engenharia de Software**. São Paulo: Pearson, 2006.

RICARTE, I. L. M. **Programação Orientada a Objetos: Uma Abordagem com Java**. 2001. Disponível em: <<http://www.dca.fee.unicamp.br/cursos/PooJava/Aulas/poojava.pdf>>. Acesso em: 27 jul. 2016.

SACIOTTI, A. C. **A importância da tecnologia da informação nas micro e pequenas empresas: Um estudo exploratório na região de Jundiaí**. 2011. 116 f. Dissertação (Mestrado) - Curso de Administração, Faccamp, Campo Limpo Paulista, 2011. Disponível em: <http://www.faccamp.br/site/arq/pdf/mestrado/Documentos/producao_discente/2011/04abril/AdaniCusinSaciloti/dissertaCAo.pdf>. Acesso em: 19 nov. 2016.

SALGADO, C. M. M. **Uso da informação no desenvolvimento do território turístico de Bonito-MS**. 2007. 109 f. Tese (Doutorado) - Curso de Desenvolvimento Local, Universidade Católica Dom Bosco, Campo Grande, 2007. Disponível em: <<http://www.cpap.embrapa.br/teses/online/DST39.pdf>>. Acesso em: 02 set. 2016.

SIERRA, K.; BATES, B. **Use a Cabeça! Java**. Rio de Janeiro: Alta Books, 2010.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de banco de dados**. 3. ed. São Paulo: Makron Books, 1999.

SOMMERVILLE, I. **Engenharia de Software**. 6. ed. São Paulo: Pearson, 2003. André Maurício de Andrade Ribeiro.

TEÓFILO, R. B.; FREITAS, L. S. de. **O uso de tecnologia da informação como ferramenta de gestão**. 2007. 12f. Curso de Tecnologia da Informação, Associação Educacional Dom Bosco - Aedb, Resende, 2007. Disponível em: <http://www.aedb.br/seget/arquivos/artigos07/652_SEGET%20roro.pdf>. Acesso em: 19 nov. 2016.

ZUBOFF, S. Automatizar/informatizar: As duas faces da tecnologia inteligente. **Revista de Administração de Empresas**, São Paulo, v. 34, n. 6, p.80-91, nov. 1994. Disponível em: <<http://www.scielo.br/pdf/rae/v34n6/a09v34n6.pdf>>. Acesso em: 31 ago. 2016.