Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Curso Técnico em Informática

José Victor de Oliveira Teixeira

Lívia dos Santos Lopes

**SISTEMA DE MEMORANDO ELETRÔNICO - Ifmemo**

João Câmara/RN

2015

José Victor de Oliveira Teixeira

Lívia dos Santos Lopes

**SISTEMA DEMEMORANDO ELETRÔNICO - Ifmemo**

Trabalho de Conclusão de Curso apresentado à Diretoria Acadêmica do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – Campus João Câmara, como requisito necessário à obtenção do título de Técnico em informática.

Orientador: Prof. Me. Diego Henrique Oliveira de Souza

João Câmara/RN

2015

José Victor de Oliveira Teixeira

Lívia dos Santos Lopes

**SISTEMA DE MEMORANDO ELETRÔNICO - Ifmemo**

Trabalho de Conclusão de Curso apresentado à Diretoria Acadêmica do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – Campus João Câmara, como requisito necessário à obtenção do título de Técnico em Informática.

Orientador: Prof. Me. Diego Henrique Oliveira de Souza

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

BANCA EXAMINADORA

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

Prof. Me. Diego Henrique Oliveira de Souza

Orientador – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

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

Prof. Me. Nickerson Fonseca Ferreira

Orientador – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

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

Prof. Me. Adorilson Bezerra de Araújo

Coordenador do Curso Técnico Integrado em informática – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

“A imaginação é mais importante que a ciência, porque a ciência é limitada, ao passo que a imaginação abrange o mundo inteiro”.

– Albert Einstein

Dedico este trabalho à mãe, que me ajudou nos momentos que mais precisei e me educou, conforme suas condições e a todos que acreditaram em mim.

**Lívia Dos Santos Lopes**

Dedico este trabalho a meus pais, minha base e ponto de apoio, pela criação e educação. A Tainara Macêdo, pelo incentivo e ajuda. A fonte de minha motivação, meu avô, Paulo Marques. Ao exemplo o qual me inspiro, minha avó, Maria Dulcinete.

**José Victor de Oliveira Teixeira**

**Agradecimentos**

Agradecemos primeiramente a Deus por ter nos proporcionado a oportunidade de nos conhecermos e desenvolvermos este trabalho de conclusão de curso. Também a todos que contribuíram para realização deste, como:

Ao Prof. Me. Diego Oliveira, pela orientação durante o processo deste projeto. Além de ser nosso professor e amigo, sendo parte essencial.

Ao Prof. Me. Adorilson Bezerra, pela ajuda com dúvidas relacionada ao funcionamento do instituto referente ao projeto, no início do mesmo.

Ao Renato Silva, pela participação como cliente em potencial para o sistema.

A todos os demais professores que nos lecionaram e que nos ensinaram tecnologias necessárias para que esse trabalho pudesse ser realizado.

Aos nossos pais, que nos apoiaram e fizeram de tudo para que pudéssemos estudar no instituto e ter uma boa educação.

Aos colegas que dividiram experiências conosco durante esses quatro anos de trajetória.

Aos amigos que nos apoiaram nos momentos em que estávamos prestes a desistir e não nos abandonaram nos momentos em que não demos a atenção que eles precisavam.

**RESUMO**

Quem precisa de tecnologia? No mundo atual, toda a sociedade. O uso desta tem se tornado, nos últimos anos, algo indispensável em empresas, órgãos, instituições, etc. O que antes era visto como máquinas gigantes para apenas algumas instituições para apenas algumas tarefas, hoje seu uso é necessário para praticamente qualquer ação a se realizar. Raramente encontramos lojas, empresas, órgãos, instituições, etc sem o uso de tal. As formas de informação evoluíram, a dependência do uso de tecnologia também. Ela já faz parte do cotidiano das pessoas. O uso desta pode auxiliar no trabalho realizado em muitas atividades. No Instituto Federal do Rio Grande do Norte, mesmo com tantos avanços tecnológicos, alguns setores ainda necessitam do auxilio da tecnologia. Na Secretaria Acadêmica do Campus João Câmara, ao que se refere ao controle de memorandos, ainda existe dificuldades para gerencia-los, principalmente na numeração (que não pode se repetir), que é feita de forma manual. Este trabalho apresenta o desenvolvimento de um sistema de gerenciamento de memorando. Objetiva melhorar o funcionamento da comunicação interna do campus, suprindo as necessidades deste setor.

**Palavras-chave**: Secretaria Acadêmica, Memorando, IFMEMO.

**ABSTRACT**

Who needs technology? In today's world, the whole society. The use of it has become, in recent years, something indispensable in companies What was once seen as giant machines to do a few tasks, today its use is necessary for virtually any action to perform. Rarely we find businesseswithout the use of technology. The forms of information have evolved, the dependence on the use of technology as well. it is already part of the daily lives of people. It’s use can assist in the work done in many activities. At Instituto Federal do Rio Grande do Norte, even with so many technological advances, some sectors still require the assistance of technology. On Campus Academic Secretary João Câmara, when it comes to the control of memos, there are still difficulties to manage them, mainly in the numbering (which cannot repeat), which is done manually. This work presents the development of a management system of memo. Aims to improve the functioning of internal communication on campus, supplying the needs of this sector.

**Keywords**: Academic Secretariat, Memorandum, IFMEMO.

**SUMÁRIO**

[LISTA DE FIGURAS 11](#_Toc413329339)

[LISTA DE TABELAS 12](#_Toc413329340)

[LISTA DE SIGLAS E ABREVIATURAS 13](#_Toc413329341)

[1. INTRODUÇÃO 14](#_Toc413329342)

[1.1. Secretaria Acadêmica 14](#_Toc413329343)

[1.2. Memorando 15](#_Toc413329344)

[1.3. Motivação 18](#_Toc413329345)

[1.4. Objetivo 18](#_Toc413329346)

[1.5. Organização 18](#_Toc413329347)

[2. FUNDAMENTAÇÃO TEÓRICA 20](#_Toc413329348)

[2.1. JAVA 20](#_Toc413329349)

[2.2. JSP 21](#_Toc413329350)

[2.3. SQL 21](#_Toc413329351)

[2.4. MySQL 23](#_Toc413329352)

[3. METODOLOGIA 25](#_Toc413329353)

[4. DESENVOLVIMENTO 27](#_Toc413329354)

[4.1. **Login** 27](#_Toc413329355)

[**4.2.** **Escolha de uma ação** 29](#_Toc413329356)

[**4.3.** **Listagem de todos os memorandos** 30](#_Toc413329357)

[**4.4.** **Pesquisar memorando** 30](#_Toc413329358)

[**4.5.** **Visualizar e imprimir memorando** 32](#_Toc413329359)

[**4.6.** **Criar memorando** 34](#_Toc413329360)

[**4.7.** **Listar todos os funcionários** 36](#_Toc413329361)

[**4.8.** **Pesquisar funcionário** 37](#_Toc413329362)

[**4.9.** **Inserir funcionário** 38](#_Toc413329363)

[**4.10.** **Alterar funcionário** 39](#_Toc413329364)

[**4.11.** **Excluir funcionário** 41](#_Toc413329365)

[5. CONCLUSÃO 43](#_Toc413329366)

[5.1. Dificuldades 43](#_Toc413329367)

[5.2. Pontos positivos 44](#_Toc413329368)

[5.3. Trabalhos futuros 44](#_Toc413329369)

[5.4. Fechamento 44](#_Toc413329370)

[REFERÊNCIAS 46](#_Toc413329371)

[APENDICES 48](#_Toc413329372)

[APENDICE A 48](#_Toc413329373)

[APENDICE B 49](#_Toc413329374)

[APENDICE C 50](#_Toc413329375)

# LISTA DE FIGURAS

[Figura 1 - Modelo oficial de memorando proposto pelo manual de redação da presidência da republica 17](#_Toc413329376)

[Figura 2 - Tela de login 27](#_Toc413329377)

[Figura 3 - Tela de erro ao logar. 28](#_Toc413329378)

[Figura 4 - Tela de ações. 29](#_Toc413329379)

[Figura 5 - Tela de listagem de todos os memorandos. 30](#_Toc413329380)

[Figura 6 - Tela de pesquisa de memorando. 31](#_Toc413329381)

[Figura 7 - Tela de resultado de pesquisa de memorando 32](#_Toc413329382)

[Figura 8 - Tela de escolha de memorando para visualizar. 33](#_Toc413329383)

[Figura 9 - Memorando gerado. 34](#_Toc413329384)

[Figura 10 - Tela de criação de um novo memorando. 35](#_Toc413329385)

[Figura 11 - Tela de confirmação de criação de memorando 36](#_Toc413329386)

[Figura 12 - Tela de tela de listagem de todos os funcionários. 36](#_Toc413329387)

[Figura 13 - Tela de pesquisa de funcionários. 37](#_Toc413329388)

[Figura 14 - Tela de resultado de pesquisa de funcionário. 38](#_Toc413329389)

[Figura 15 - Tela de criação de um novo funcionário. 38](#_Toc413329390)

[Figura 16 - Tela de confirmação de inserção de funcionário 39](#_Toc413329391)

[Figura 17 - Tela de alteração de funcionário. 40](#_Toc413329392)

[Figura 18 - Tela de alteração de alteração de funcionário após digitar matricula. 40](#_Toc413329393)

[Figura 19 - Tela de confirmação de atualização de funcionário 41](#_Toc413329394)

[Figura 20 - Tela exclusão de funcionário. 42](#_Toc413329395)

[Figura 21 - Tela de confirmação de exclusão de funcionário 42](#_Toc413329396)

[Figura 22 - Diagrama de Entidade e Relacionamento do Banco de Dados do sistema. 48](#_Toc413329397)

**LISTA DE TABELAS**

[Tabela 1 - Requisitos funcionais do IFMEMO 49](#_Toc413329398)

[Tabela 2 - Requisitos não funcionais do IFMEMO 50](#_Toc413329399)

# LISTA DE SIGLAS E ABREVIATURAS

CSS - Cascating Style Sheets

XHTML - eXtensible Hypertext Markup Language

SGBD - Sistema Gerencial de Banco de Dados

SQL - Structured Query Language

JSP - JavaServer Pages

1. **INTRODUÇÃO**

O Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN), assim como qualquer outra instituição, é uma organização que lida com pessoas. Dentre elas possuem alunos, docentes, técnicos administrativos, Secretário, Diretor, etc. Os servidores são responsáveis por manter a organização da instituição e se esta é capaz de atender as necessidades. Estes devem ser capacitados para simplificar processos e métodos de trabalho, planejar e dirigir, avaliar e controlar as atividades, entender as necessidades dos usuários e trabalhar com eficiência reduzindo os desperdícios.

* 1. Secretaria Acadêmica

No Campus João Câmara, estes servidores estão distribuídos em 20 setores. Dentre os diversos setores existentes no campus, o setor em que o software vai ser implementado, é na Secretaria Acadêmica, pois é o setor responsável pela geração de memorandos, o que vai ser explicado mais tarde. Além da geração de memorandos a Secretária é o setor responsável pelo controle, verificação, registro e arquivamento da documentação da vida acadêmica do aluno (desde o momento de sua entrada até a expedição do diploma).

A Secretaria também realiza outras atividades, tais como: registros e controles de documentos dos acadêmicos; a implantação do Sistema Integrado do Controle Acadêmico; montagem e encaminhamento de processos e diplomas para o setor de registro; receber, redigir e expedir a correspondência que lhe for confiada; manter a organização dos ofícios e demais documentos; encaminhar em tempo hábil, á direção, todos os documentos que devem ser assinado; entre diversas outras atividades.

Além disso, permite que o aluno justifique faltas, realize o cancelamento de matrícula, bem como a renovação, ver o calendário acadêmico, estrutura curricular dos cursos, horários de aulas ou qualquer outra informação referente à vida acadêmica. A Secretaria Acadêmica trabalha em conjunto com a diretoria acadêmica no que se refere à comunicação interna, ou seja, na geração de memorandos. Estes memorandos são gerados na secretária e na diretoria, sendo repassados para a que o gabinete possa assinar.

* 1. Memorando

A secretaria é responsável pela emissão de memorandos, que é um tipo de oficio menos formal. O memorando é um tipo de comunicação eminentemente interna vinculada entre as unidades administrativas, endereçado aos setores. A principal característica e vantagem do memorando é a sua agilidade. Em relação a sua forma, o memorando segue o padrão de oficio. Segundo o Manual de Redação da Presidência da República, este tipo de documento deve ser formado pelas seguintes partes:

1. Tipo e número do expediente, seguido da sigla do órgão que o expede;
2. Local e data em que foi assinado, por extenso, com alinhamento à direita;
3. Assunto: resumo do teor do documento
4. Destinatário: o nome e o cargo da pessoa a quem é dirigida a comunicação. No caso do ofício deve ser incluído também o endereço.
5. Texto: nos casos em que não for de mero encaminhamento de documentos, o expediente deve conter a seguinte estrutura:
   1. Introdução, que se confunde com o parágrafo de abertura, na qual é apresentado o assunto que motiva a comunicação. Evite o uso das formas: "Tenho a honra de", "Tenho o prazer de", "Cumpre-me informar que", empregue a forma direta;
   2. Desenvolvimento, no qual o assunto é detalhado; se o texto contiver mais de uma ideia sobre o assunto, elas devem ser tratadas em parágrafos distintos, o que confere maior clareza à exposição;
   3. Conclusão, em que é reafirmada ou simplesmente reapresentada a posição recomendada sobre o assunto.

Os parágrafos do texto devem ser numerados, exceto nos casos em que estes estejam organizados em itens ou títulos e subtítulos.

Já quando se tratar de mero encaminhamento de documentos a estrutura é a seguinte:

1. introdução: deve iniciar com referência ao expediente que solicitou o encaminhamento. Se a remessa do documento não tiver sido solicitada, deve iniciar com a informação do motivo da comunicação, que é encaminhar, indicando a seguir os dados completos do documento encaminhado (tipo, data, origem ou signatário, e assunto de que trata), e a razão pela qual está sendo encaminhado, segundo a seguinte fórmula:
2. Desenvolvimento: se o autor da comunicação desejar fazer algum comentário a respeito do documento que encaminha, poderá acrescentar parágrafos de desenvolvimento; em caso contrário, não há parágrafos de desenvolvimento em aviso ou ofício de mero encaminhamento.
3. Fecho (ver item “2.2. Fechos para Comunicações” do manual de redação da presidência da república);
4. Assinatura do autor da comunicação; e
5. Identificação do signatário (ver item “2.3. Identificação do Signatário” do manual de redação da presidência da república).

Na Figura 1 é mostrado o modelo oficial de memorando (planalto) representando suas normas.



Figura 1 - Modelo oficial de memorando proposto pelo manual de redação da presidência da republica

* 1. Motivação

No manuseio de memorando no instituto ocorrem erros como a repetição da numeração. Este problema ocorre, pois o memorando é manuseado por duas pessoas em salas distintas e por meio de planilhas eletrônicas. A repetição da numeração dos memorandos pode parecer sem importância, porém pode atrapalhar muito o trabalho e causar situações complicadas de serem resolvidas.

O projeto IFMEMO foi desenvolvido com o intuito de auxiliar o trabalho da Secretaria Acadêmica do campus a resolver problemas como o citado acima. A numeração do memorando pode-se dizer é a parte principal causadora de problemas e uma das razões pela qual o sistema foi desenvolvido. O sistema permite que a cada vez que um memorando é gerado, sua data e hora fiquem armazenadas, assim como o nome de quem o gerou e o mais importante, a numeração do memorando. No próximo memorando que será criado, esta numeração vai ser automaticamente atualizada e não vai se repetir, evitando frustações e garantindo segurança.

* 1. Objetivo

O objetivo geral deste trabalho é desenvolver um software de apoio a Secretaria Acadêmica, nomeado Sistema de memorando eletrônico do IFRN/JC (IFMEMO), implementado para a plataforma web e projetado para suprir as demandas internas do cenário, identificadas no campus João Câmara do IFRN.

A fim de chegar ao objetivo geral deste trabalho, alguns objetivos específicos são definidos, como: aplicar a aprendizado sobre os conceitos estudados durante todo o curso; conhecer novas tecnologias no decorrer de cada etapa do projeto.

* 1. Organização

Este trabalho foi desenvolvido com cinco capítulos: a introdução, fundamentação teórica, metodologia, desenvolvimento e conclusão. Em cada uma delas pode-se conhecer um pouco mais sobre o IFMEMO, o sistema de memorandos que será implementado no instituto a partir desse ano de 2015. Ao decorrer deste trabalho são apresentadas todas as ferramentas utilizadas no seu desenvolvimento.

No capítulo inicial do trabalho, a introdução, há uma breve descrição do sistema, o objetivo de seu desenvolvimento e a justificativa.

No segundo capítulo, a fundamentação teórica, encontram-se disponíveis as teorias que o sistema foi baseado. Apresenta um pouco sobre cada conceito utilizado no sistema como Java, MySQL, SQL, JSP explicando o que é cada uma dessas tecnologias, o que permite um entendimento sobre o assunto quando forem citados mais adiante.

No terceiro capítulo, metodologia, fornece as informações de como o sistema foi desenvolvido passo-a-passo desde entrevista realizada com Renato, para o estudo de caso até a implementação dos códigos. Essa explicação é clara e possibilita que outras pessoas observem o passo a passo do trabalho.

O quarto capítulo, desenvolvimento, é o de maior extensão apresenta as imagens das telas, com suas funcionalidades.

Para o desenvolvimento do sistema fez-se o uso da metodologia de pesquisa e de um estudo de caso realizado com Renato Silva, onde se abordou os problemas causados pelo método que os memorandos eram gerados. Esse estudo de caso consistiu em uma entrevista com perguntas simples a respeito das necessidades a serem resolvidas, do tempo disponível para que os testes fossem iniciados.

Por fim, o quinto capítulo contém as conclusões, que relatam os resultados da pesquisa em relação ao desenvolvimento do sistema, dificuldades durante o desenvolvimento, pontos positivos do sistema, sugestões para trabalhos futuros e fechamento.

1. **FUNDAMENTAÇÃO TEÓRICA**

## JAVA

Java é uma linguagem de programação orientada a objetos iniciada em 1991 na empresa Microsystems, por uma equipe de programadores liderada por *James Gosling.* Hoje é disponibilizada pela empresa Oracle, que a comprou em 2009. A linguagem é o padrão global para desenvolvimento e fornecimento de aplicativos para celular, jogos, conteúdo on-line e software corporativo. Praticamente, ela é à base de todos os tipos de aplicativos em rede.

Na década de 90, seus maiores problemas eram ponteiros, gerenciamento de memória, organização, falta de bibliotecas, ter de reescrever parte do código ao mudar de sistema operacional, custo financeiro de usar a tecnologia, entre outros. A linguagem Java supera esses problemas e, o que antes aparecia com frequência, foram atacados pela plataforma Java, uma vez que a principal motivação era que esta linguagem fosse usada em dispositivos pequenos.

Java foi pensado a fim de permitir aplicações de alto desempenho para dispositivos portáteis. Ela permite que os desenvolvedores:

* Gravassem software em uma plataforma e o executem virtualmente em qualquer outra plataforma;
* Criassem programas que podem ser executados dentro de um web browser e acessem web services disponíveis;
* Desenvolvessem aplicações do servidor para fóruns on-line, armazenamentos, pesquisas, processamento de forms HTML;
* Combinassem aplicações ou serviços usando a linguagem Java para criar aplicações ou serviços altamente personalizáveis.
* Criassem aplicações potentes e eficientes para telefones celulares, processadores remotos, micro controladores, módulos sem fio, sensores, gateways, produtos de consumo e praticamente qualquer outro dispositivo eletrônico.
  1. JSP

Java Server Pages é uma linguagem aberta de script utilizada para criação de paginas da internet com conteúdo dinâmico. Esta pagina JSP tem extensão .jsp e compõem marcações HTML com codificações Java, essas inseridas entre as tag’s “<%” e “%>”.

O JSP necessita de um servidor para poder funcionar, assim a requisição do usuário é processada no servidor e retornada e apresentada apenas a codificação HTML. Então, o usuário não consegue ver o código em JSP.

É possível, também, criar conteúdos dinâmicos através de servelets. Entretanto, teremos que criar um servelet sempre que quisermos fazer uma solicitação. Isto toma tempo, trabalho e dificulta na manutenção das paginas ao que se refere à legibilidade do código, pois mistura classes com código Java e marcação HTML. Para fugir destes transtornos, é necessário usar uma tecnologia para permitir a utilização do Java diretamente no HTML. Esta tecnologia, como explicado anteriormente, é a JSP.

Assim, a página JSP é interpretada pelo servidor (Tomcat, por exemplo) e não precisa ser compilada, como aconteceria ao usar servelets. Esta tarefa é realizada em tempo real pelo servidor.

* 1. SQL

Structured Query Language (SQL) É uma linguagem de pesquisa declarativa para banco de dados relacional, que possui alguns elementos como as cláusulas, operadores lógicos e relacionais, funções de agregação e as expressões. Desenvolvida originalmente na década de 70 nos laboratórios da International Business Machines (IBM), é uma linguagem que possui muitas variações e extensões definidas pelos diferentes fabricantes dos Sistemas de Gerenciamento de Banco de Dado (SGBDs). Ela altera, insere e deleta os dados no banco.

A primeira versão da SQL surgiu entre 1976 e 1977, na Califórnia, chamada de SEQUEL (Structured Query English Language). Mais tarde a linguagem foi ampliada e passou a ser conhecida como SQL. Atualmente, Vários SGBD’s utilizam a linguagem SQL como padrão para seus bancos de dados.

A SQL tornou-se cada vez mais utilizada devido a sua forma de manipulação e consulta dentro do banco de dados. Em 1982 a ANSI (American National Standard Institute) tornou a SQL sua linguagem padrão em ambiente relacional.

Através da SQL o usuário pode fazer consultas, sem a necessidade de criação de um programa. Os comandos SQL acessam informações armazenadas em um banco de dados relacional. O administrador do banco de dados pode usar os comandos SQL para a realização de tarefas como consulta, exclusão, inserção. A linguagem possui um controle de acesso que não permite manipulação não autorizada dos dados

A linguagem SQL também é uma linguagem padrão para a manipulação de informações distribuídas. É formada pela Data Definition Linguage (DDL), Data Manipulation Linguage (DML), Data Query Linguage (DQL) e Data Control Linguage (DCL). A linguagem de manipulação de dados (DML) permite ao usuário ou o programa de aplicação, realizar operações como exclusão, inclusão, atualização ou seleção de dados já armazenados no banco. Na linguagem definição de dados (DDL), o usuário realiza a organização dos dados armazenados e da relação entre eles.

A principal vantagem em se utilizar a SQL é que ela possui independência de fabricante, uma vez que a SQL é a linguagem padrão da maioria dos bancos de dados atuais e padronizada pela ANSI, o usuário tem a possibilidade de mudar de SGBD sem a necessidade de modificar os programas de aplicação; portabilidade de plataforma entre hardware e software; redução dos custos com treinamento, por possuir a portabilidade de plataforma, não se faz necessário que haja uma reciclagem da equipe que desenvolveu o projeto; usa como linguagem o inglês de forma simples e de fácil entendimento; permite que o usuário acesso de forma rápida às informações do banco, de forma interativa; múltiplas visões dos dados; o usuário pode de forma dinâmica incluir, alterar com flexibilidade as estruturas dos dados armazenados.

No entanto, segundo o livro “introdução a sistemas de banco de dados” do autor Christopher J. date, a linguagem SQL apresenta algumas desvantagens em seu uso. As críticas do autor são referentes à falta de algumas funções; Discordância com as linguagens hospedeiras; Falta de ortogonalidade nas expressões, funções embutidas, variáveis indicadoras, referência a dados correntes, constante NULL, conjuntos vazios; Definição formal da linguagem após sua criação não dá suporte a alguns aspectos do modelo relacional.

A linguagem SQL apresenta alguns comandos para a execução de determinadas tarefas. O **CREATE** para a criação de tabelas; **DROP** comando responsável pela exclusão das tabelas e o **SELECT** que permite que o usuário realize consultas. Para realizar uma consulta o usuário indica a(s) coluna(s) especifica(s) que deseja consultar. O usuário pode definir condições para essa consulta com o uso da cláusula **where**. Essas tabelas contêm colunas, que podem variar de valores, podendo esses serem nulos. Toda tabela terá um valor único que a identifica chamado chave primária, que não pode receber um valor nulo. Ás vezes torna-se necessário o acesso a varias tabelas simultaneamente, para isso usa-se o JOINS que realiza as junções entre as tabelas para buscar os dados necessários.

Ao executar um programa que realize mudanças no banco de dados, realizam-se transações, ou seja, atualizam o banco. As transições possuem propriedades conhecidas como (ACID). Ela é atômica, ou seja, se não for executada por completa nada é executado; deve manter a consistência do banco de dados; isolamento, a transação não pode ser visível para outras transações até que seja encerrada com sucesso; durabilidade, uma vez executada com sucesso. As atualizações devem persistir mesmo se houver falha no sistema.

A informação é algo muito importante para um sistema, visando garantir a segurança dos dados dos SGBD’s concedem e retiram privilégios aos usuários.

* 1. MySQL

O MySQL nasceu da necessidade, encontrada pelos desenvolvedores do sistema de gerenciamento de banco de dados( SGBD), de encontrar um método de conectar as tabelas feitas em linguagem SQL para determinados fins.

O MySQL é um banco de dados relacional gratuito, eficaz e otimizado para aplicações web, multi-plataformas, desenvolvido e mantido pela empresa MySQL AB, que também oferece uma versão comercial. As tabelas criadas podem atingir até o tamanho de 4GB, além disso, esse SGBD aceita outros tipos de linguagens de programação, tais como C, PHP, JAVA dentre outras.

Essa tecnologia apresenta características como flexibilidade faz uso de tipos de campos variados, podendo esses serem inteiros, float, Double, char, varchar, text, date, time, datetime, timestamp, set e enum, portabilidade sendo multi-plataforma e podendo ser usado por diversos usuários, segurança o sistema possui senha, outras características são conectividade , escalabilidades e limites, ou seja, ele apresenta a capacidade de aumentar seu desempenho, quanto mais recursos são acrescentados a ele.

O MySQL é responsável pela comunicação entre o aplicativo e o banco de dados, no caso do IFMEMO entre o sistema e o banco, ele pega os dados fornecidos pelo usuário e armazena no banco, do mesmo modo se for necessária uma consulta o MySQL busca no banco e apresenta o resultado na tela para o usuário.

1. **METODOLOGIA**

O sistema foi desenvolvido em quatro etapas: Concepção, em que ocorreram os primeiros contatos com o cliente, este foi o momento de planejar o sistema; Elaboração, em que analisamos o problema; a construção e o desenvolvimento do software, em que ocorre a maior parte das codificações; Transição, período em que se mostra o sistema ao cliente para ser testado suas funcionalidades e se atende as necessidades, o resultado deste será mostrado na CONCLUSÃO.

O sistema de software IFMEMO foi desenvolvido a partir de um estudo de caso feito com o Secretário Acadêmico, Renato Silva, em que com base nas informações fornecidas por ele, foi feito o levantamento de requisitos. Os requisitos são uma declaração geral do que o cliente deseja do sistema. Após o levantamento dos requisitos, foram divididos esses requisitos em tarefas para facilitar e priorizamos de acordo com o que o cliente especificou.

No desenvolvimento do sistema o levantamento de requisitos é uma das partes principais, uma vez que permitiu saber o que o cliente desejava em seu sistema, quais funcionalidades, etc. Os requisitos são levantados com base no domínio do problema, ou seja, são levantados com a observação da necessidade do local onde o sistema será implantado.

Na fase de levantamento de requisitos, a partir de uma entrevista, em que o cliente comentará como quer no sistema e quais problemas deseja que sejam resolvidos com a implantação. Neste momento a equipe preocupa-se em entender o que o cliente deseja, se possível é pedido para que ele escreva em um papel.

Os requisitos são subdivididos em funcionais e não funcionais. Os funcionais são o que o cliente deseja que o sistema realize, enquanto os não funcionais são características das funcionalidades, como confiabilidade, portabilidade, segurança e usabilidade. No APENDICE B, é mostrado os requisitos funcionais do sistema. No APENDICE B, é mostrado os requisitos não funcionais do sistema.

No planejamento, logo após identificarmos os requisitos, os dividimos em tarefas, que são um fluxo sequencial de instruções para determinada finalidade. Estas são algumas vezes interdependentes, ou seja, somente podem ser feitas quando a que antecede seja realizada.

A divisão dos requisitos em tarefas facilita o trabalho no momento de implementação. Estas tarefas receberam prioridades que identificam a sequencia em que devem ser feitas, essas priorizações são feitas para que se possa identificar requisitos críticos e ajudar na tomada de decisões referentes ao sistema.

Em outro momento, foram realizados os rascunhos das telas para somente depois serem transformados em códigos. Neste momento a equipe já começa a decidir as cores e o formato como as informações seriam organizadas nas páginas.

Na sequencia, depois de desenhados os rascunhos das páginas e também do modelo padrão de memorando, incrementou-se em HTML e CSS, para que o sistema comece a “ganhar vida”.

Depois de incrementadas as páginas tornou-se necessário conectar com o banco de dados. Para realiza-lo, foi desenvolvido no Netbeans um projeto que recebeu como dependência a biblioteca do MySQL. Nesta biblioteca, configuramos o banco de dados para armazenar todas as informações.

O banco de dados é composto por quatro tabelas (ver APENDICE **A**): Memorando, Servidores, Cargo, Setor. A tabela Memorando está conectada a tabela servidores recebendo a coluna matricula\_servidor como chave estrangeira. A tabela Servidores está conectada as tabelas Cargo e Setor, recebendo a coluna id\_cargo como chave estrangeira da tabela Cargo e a coluna id-setor como chave estrangeira da tabela Setor.

Foi fornecida pela secretaria uma lista com todos os setores do Campus João Câmara. Esses dados foram inseridos na tabela Setor.

Após essa conexão começamos a utilizar os comandos da linguagem SQL para fazermos as consultas, exclusões, atualizações e inserção ao banco.

O projeto foi desenvolvido com o uso de ferramentas como MySQL (um gerenciador de banco de dados que faz uso da linguagem SQL como interface) e Netbeans (ambiente de desenvolvimento integrado, de código aberto e multi plataformas). Faz uso da linguagem JSP, almeja diminuir situações de erros, ou seja, facilitar a comunicação dentro do instituto, como já citado em capítulos anteriores.

1. **DESENVOLVIMENTO**
   1. **Login**

A geração de memorandos ou qualquer alteração que seja em algum funcionário, só pode ser feita pelo Secretário Acadêmico ou Diretor Acadêmico. Então, faz-se necessário que o usuário tenha um cadastro para que possa logar. A Figura 2, demonstra a tela de login do sistema possuindo uma matrícula e uma senha. Nessa página, o usuário tem que informar os dois campos já citados, além de informar.

A tela de login consiste em receber a matrícula e a senha do usuário que deseja entrar no sistema e verificar se o mesmo tem permissão de realizar o login. Para que o usuário possa entrar no sistema é necessário que o cargo deste esteja atribuído como administrador. Se este usuário tiver permissão de logar no sistema e a senha esteja de acordo com a cadastrada no banco de dados, ele é redirecionado a tela inicial para que possa realizar a ação desejada. Caso o usuário não tenha permissão de entrar no sistema ou a senha esteja errada, ele é redirecionado para a tela de erro e convidado a tentar novamente, como na Figura 3. Caso o usuário tenha permissão e a senha esteja correta, ele é redirecionado para a tela de ações (Figura 4), que é a tela inicial onde ele vai escolher o que deseja realizar.

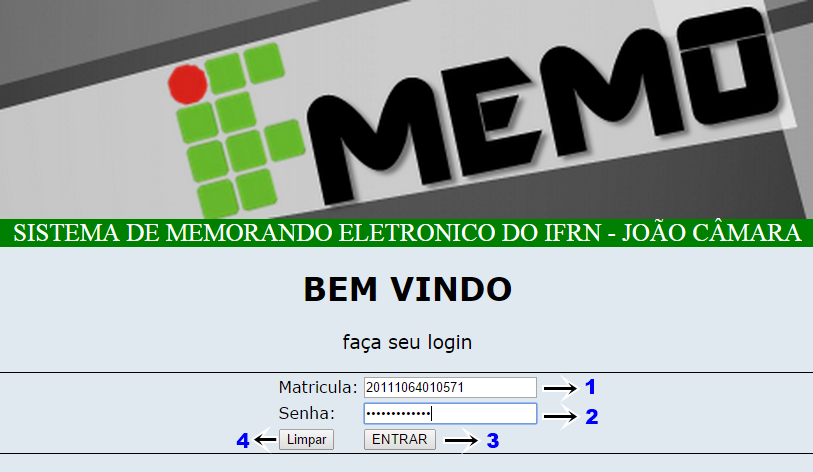


Figura 2 - Tela de login

Segue abaixo uma lista das funções que cada campo ou botão realiza, de acordo com a numeração indicadora da Figura 2:

1. Campo que receberá a matricula do usuário que deseja logar. esta matricula vai ser usada para fazer uma consulta no banco de dados na tabela servidores a fim de obter os dados das demais colunas deste servidor;
2. Campo que receberá a senha do usuário que deseja logar. Esta senha vai ser comparada com a senha cadastrada no banco de dados;
3. Ao ser clicado, os dados são enviados e comparados com os dados cadastrados no banco de dados. O usuário é redirecionado a tela inicial do IFMEMO se este usuário tiver permissão para entrar no sistema (Figura 4);
4. Ao ser clicado todos os campos do formulário são restaurados.



Figura 3 - Tela de erro ao logar.

Segue abaixo uma lista das funções que cada campo ou botão realiza, de acordo com a numeração indicadora da Figura 3:

1. Retorna a tela de login para o usuário tentar logar novamente.
   1. **Escolha de uma ação**

Após serem verificados os dados do login (matricula, senha e cargo) e confirmado que o usuário possa entrar no sistema, é exibido como tela inicial às opções para que o cliente escolha a ação que o usuário deseja realizar, como mostra a Figura 4.

Na tela de ação é mostrado um menu organizado em duas colunas, memorando e funcionário, com as opções de todas as funcionalidades do sistema. Para escolher, o usuário deve clicar nos links. Quando um link é clicado o usuário é redirecionado a pagina referente ação escolhida.



Figura 4 - Tela de ações.

Segue abaixo uma lista das funções que cada campo ou botão realiza, de acordo com a numeração indicadora da Figura 4:

1. Redireciona para pagina de listagem de todos os memorandos (Figura 5);
2. Redireciona para pagina de pesquisa de memorando (Figura 6);
3. Redireciona para pagina de visualização de memorando, seguida da impressão do mesmo (Figura 8, Figura 9);
4. Redireciona para pagina de criação de um novo memorando (Figura 10);
5. Redireciona para pagina de listagem de todos os funcionários (Figura 12);
6. Redireciona para pagina de pesquisa de funcionário (Figura 13);
7. Redireciona para pagina de inserção de um novo funcionário (Figura 15);
8. Redireciona para pagina de alteração de um funcionário (Figura 17, Figura 19);
9. Redireciona para pagina de exclusão de funcionário (Figura 20);
10. Ao ser clicado, é realizado o logout do usuário.
    1. **Listagem de todos os memorandos**

A listagem de todos os memorandos consiste em listar todos os memorando existentes. Os dados são mostrados numa tabela com os valores do memorando, como mostrado na Figura 5. Cada linha da tabela contém um memorando. As colunas especificam o tipo de dado do memorando, como numero, matricula do destinatário, assunto, setor do destinatário e data. Os dados preenchidos na tabela são buscados no Banco de Dados.



Figura 5 - Tela de listagem de todos os memorandos.

* 1. **Pesquisar memorando**

Volta e meia é preciso consultar um memorando criado a fim de saber sua numeração, ou data de criação, etc. Para realizar tal tarefa, o usuário pode escolher a opção de pesquisar memorando no menu de opões e será redirecionado para pagina de pesquisa (Figura 6).

A pesquisa do memorando é realizada através do assunto do mesmo. O usuário digita o assunto no campo especifico e o valor deste campo é enviado ao banco de dados para fazer uma consulta e retornar os demais dados: numero, matricula do destinatário, assunto e data. Os dados são mostrados numa tabela com os valores do memorando, como mostrado na Figura 7. Cada linha da tabela contém um memorando. As colunas especificam o tipo de dado do memorando.



Figura 6 - Tela de pesquisa de memorando.

Segue abaixo uma lista das funções que cada campo ou botão realiza, de acordo com a numeração indicadora da Figura 6:

1. Campo que recebe o assunto do memorando para que seja pesquisado. Este valor é enviado ao banco de dados para fazer a consulta, quando o usuário clicar em pesquisar;
2. Ao ser clicado o usuário o assunto é enviado ao banco de dados para fazer a consulta e o usuário é redirecionado a tela de resultado da pesquisa do memorando (Figura 7);

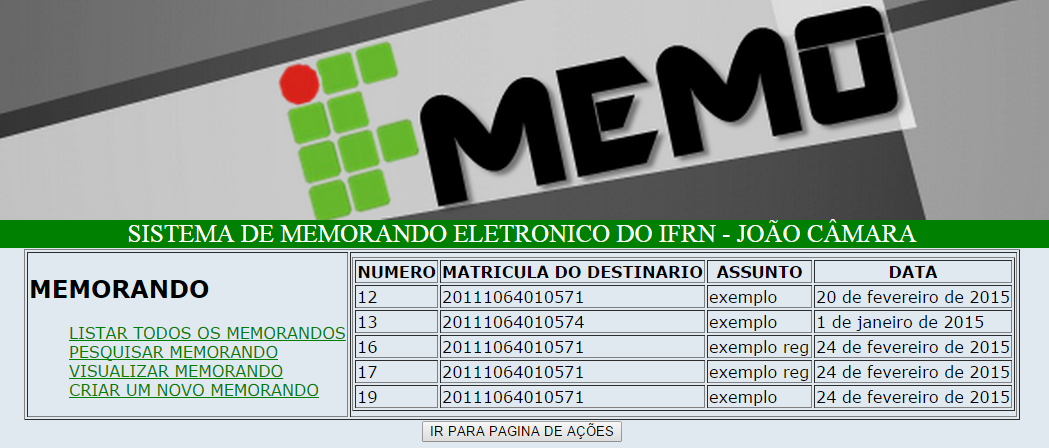


Figura 7 - Tela de resultado de pesquisa de memorando

* 1. **Visualizar e imprimir memorando**

A principal funcionalidade do IFMEMO é visualização do memorando. É nesta que é gerado o memorando é mostrado com todos os dados preenchidos recuperados do Banco de dados.

Visualizar memorando consiste em mostrar o memorando pronto para o usuário. Na Figura 8 é mostrado a tela que o usuário deve escolher o memorando numa caixa de seleção que exibe o numero do memorando, a matricula do destinatário e o assunto do memorando. Após o usuário escolher o memorando desejado e clicar em visualizar, é feito uma busca através do número do memorando (chave primaria) e ele é redirecionado para uma tela em que mostra o memorando gerado e preenchido de acordo com os dados inseridos no Banco de Dados, como na Figura 9. Na pagina de visualização do memorando, o usuário pode imprimir ou salvar como PDF o memorando pressionando **CTRL+ P**.



Figura 8 - Tela de escolha de memorando para visualizar.

Segue abaixo uma lista das funções que cada campo ou botão realiza, de acordo com a numeração indicadora da Figura 6:

1. Campo de seleção em que o usuário escolhe o memorando que deseja visualizar;
2. Ao ser clicado o usuário é redirecionado para visualização do memorando selecionado.

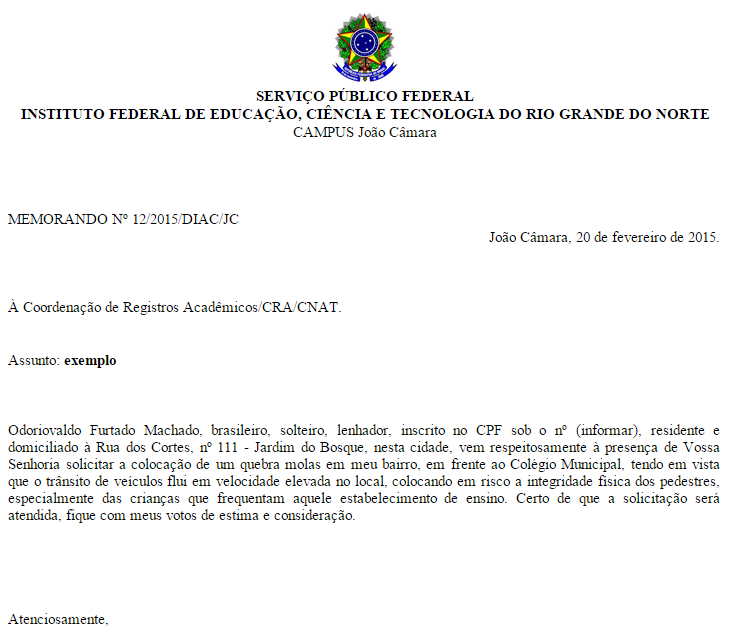


Figura 9 - Memorando gerado.

* 1. **Criar memorando**

Esta tela é usada para criação de um memorando. Na Figura 10 é mostrado a tela de criação do memorando. Nela o usuário deve inserir os dados do memorando como instituição, assunto, local, corpo, setor destinatário, matricula e data. Esses dados são inseridos no Banco de Dados quando o usuário clica no botão criar. O campo do corpo está limitado de 2000 caracteres, para que o memorando não ultrapasse uma pagina.

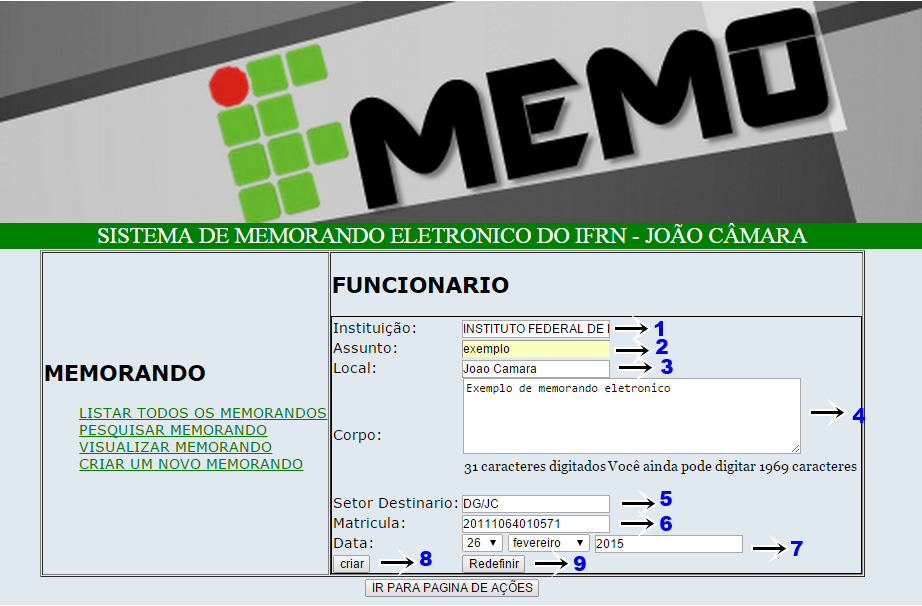


Figura 10 - Tela de criação de um novo memorando.

Segue abaixo uma lista das funções que cada campo ou botão realiza, de acordo com a numeração indicadora da Figura 10:

1. Campo pré-definido que receberá a instituição remetente do memorando;
2. Campo que receberá o assunto do memorando;
3. Campo pré-definido que receberá o local remetente do memorando;
4. Campo, limitado de 2000 caracteres, que receberá o corpo do memorando;
5. Campo que receberá o setor do destinatário do memorando;
6. Campo que receberá a matricula do destinatário do memorando;
7. Conjunto com três campos, dois campos de seleção e um campo de texto pré-definido, que receberá a data de criação do memorando;
8. Ao ser clicado os dados do formulário são enviados e o usuário é redirecionado a tela de confirmação de criação de memorando (Figura 11);
9. Ao ser clicado todos os campos do formulário são restaurados.



Figura 11 - Tela de confirmação de criação de memorando

* 1. **Listar todos os funcionários**

A listagem de todos os funcionários consiste em listar todos os funcionários existentes. Os dados são mostrados numa tabela com os valores do funcionário. Cada linha da tabela contém um funcionário. As colunas especificam o tipo de dado do funcionário como nome, matricula, cargo e setor. Os dados preenchidos na tabela são buscados no Banco de Dados.



Figura 12 - Tela de tela de listagem de todos os funcionários.

* 1. **Pesquisar funcionário**

A pesquisa do funcionário é realizada através do nome do mesmo. O usuário digita o nome do funcionário no campo especifico e o valor deste campo é enviado ao banco de dados para retornar os demais dados: matricula, nome, cargo e setor. Os dados são mostrados numa tabela com os valores do funcionário. Cada linha da tabela contém um funcionário. As colunas especificam o tipo de dado do funcionário.

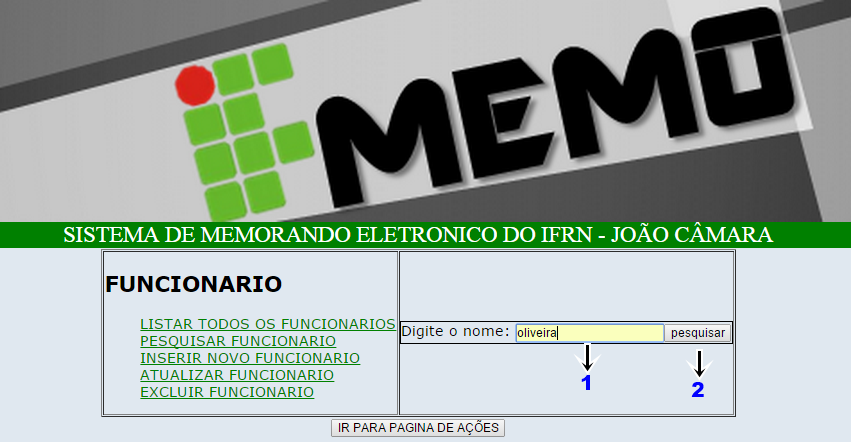


Figura 13 - Tela de pesquisa de funcionários.

Segue abaixo uma lista das funções que cada campo ou botão realiza, de acordo com a numeração indicadora da Figura 13Figura 3:

1. Campo que receberá o nome do funcionário para que seja pesquisado;
2. Ao ser clicado o usuário é redirecionado para pagina de resultado da pesquisa de funcionário.



Figura 14 - Tela de resultado de pesquisa de funcionário.

* 1. **Inserir funcionário**

Esta tela é usada para inserir de um novo funcionário. Nela o usuário irá inserir os dados do funcionário como matricula, nome, cargo e setor. Esses dados são inseridos no Banco de Dados quando o usuário clica no botão INSERIR DADOS.

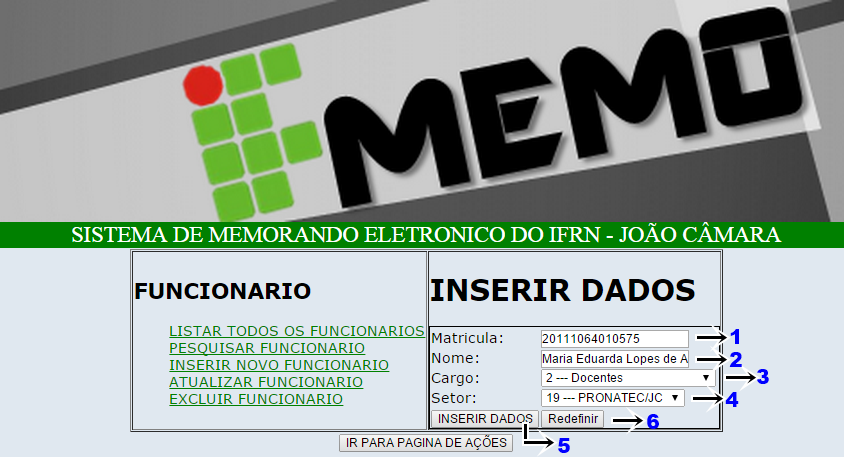


Figura 15 - Tela de criação de um novo funcionário.

Segue abaixo uma lista das funções que cada campo ou botão realiza, de acordo com a numeração indicadora da Figura 15:

1. Campo que receberá a matricula do funcionário;
2. Campo que receberá o nome do funcionário;
3. Campo de seleção que receberá o cargo do funcionário;
4. Campo de seleção que receberá o setor do funcionário.
5. Ao ser clicado os dados são inseridos no Banco de dados e o usuário é redirecionado para tela de confirmação de inserção de funcionário (Figura 16).
6. Ao ser clicado todos os campos do formulário são restaurados.



Figura 16 - Tela de confirmação de inserção de funcionário

* 1. **Alterar funcionário**

Esta tela é usada para atualizar de um funcionário existente. Nela o usuário irá inserir a matricula do funcionário que deseja alterar. Ao clicar em alterar é feito uma busca no banco de dados daquela matricula e se ela existir é mostrado um novo formulário pré-definido com os demais dados do funcionário para que seja alterado, como matricula, nome, cargo e setor. Esses dados são alterados no Banco de Dados quando o usuário clica no botão ATUALIZAR DADOS.



Figura 17 - Tela de alteração de funcionário.

Segue abaixo uma lista das funções que cada campo ou botão realiza, de acordo com a numeração indicadora da Figura 17:

1. Campo que receberá a matricula do funcionário que vai ser alterado;
2. Ao ser clicado todos os campos do formulário são restaurados;
3. Ao ser clicado o usuário é redirecionado para a pagina de alteração dos dados do funcionário selecionado pela matricula digitada no campo 1.



Figura 18 - Tela de alteração de alteração de funcionário após digitar matricula.

Segue abaixo uma lista das funções que cada campo ou botão realiza, de acordo com a numeração indicadora da Figura 18:

1. Campo pré-definido e inalterável que receberá a matricula do funcionário;
2. Campo pré-definido que receberá o nome alterado do funcionário;
3. Campo de seleção que receberá o cargo alterado do funcionário;
4. Campo de seleção que receberá o setor alterado do funcionário;
5. Ao ser clicado os dados são alterados e o usuário é redirecionado a pagina de confirmação de alteração de dados do funcionário;
6. Ao ser clicado todos os campos do formulário são restaurados.



Figura 19 - Tela de confirmação de atualização de funcionário

* 1. **Excluir funcionário**

Esta tela é usada para excluir um funcionário existente. Nela o usuário irá inserir a matricula do funcionário que deseja excluir. Ao clicar em alterar é feito uma busca no banco de dados daquela matricula e se ela existir é mostrado um novo formulário pré-definido todos os dados deste funcionário são excluídos do Banco de Dados.



Figura 20 - Tela exclusão de funcionário.

1. Campo que receberá a matricula do funcionário que deseja excluir;
2. Ao ser clicado o usuário é redirecionado a pagina de confirmação de exclusão de funcionário.



Figura 21 - Tela de confirmação de exclusão de funcionário

1. **CONCLUSÃO**

No Instituto havia um problema na geração de memorandos, pois estes eram realizados de forma manual por duas pessoas (Secretário Acadêmico e Diretor Acadêmico). O método de controle da numeração dos memorandos era anotá-los numa planilha eletrônica e cada vez que um memorando era criado, atualizava o número nesta planilha. O problema é que, estas pessoas trabalham em salas diferentes e, consequentemente, computadores diferentes e muitas vezes, na geração de um novo memorando, estas planilhas não estavam sincronizadas. Inevitavelmente, por consequência da não sincronização da numeração, são gerados memorandos com numeração repetida. Se um memorando já estiver registrado, não é permitido o registo de outro memorando com a mesma numeração, então um tem que ser descartado e feito um novo memorando com a numeração correta. Como cada memorando tem que ser assinado pelo gabinete, ao gerar outro memorando (devido à numeração repetida), faz-se necessário uma nova assinatura. Muitas vezes a pessoa responsável do gabinete não está no campus (seja por folga, feriado, férias, recesso, etc.), fazendo com que atrase a comunicação, contrariando a principal característica do memorando, a agilidade.

A fim de evitar estes problemas, frustrações e atrasos, foi desenvolvido um sistema – IFMEMO - capaz de gerar uma numeração automática a cada vez que um memorando é criado. Com isso, consequentemente, não há existência de erro na sincronização da numeração, pois esta será armazenada no banco de dados e atualizada automaticamente quando um memorando é gerado por qualquer que seja o usuário (Secretário Acadêmico ou Diretor Acadêmico). Sem a ocorrência de repetição, graças ao IFMEMO, evita trabalho dobrado na geração de um novo memorando, devido a numeração repetida, assim como evitar atrasos devido assinatura. Além disso, a realização do sistema IFMEMO reforçou e ajudou o aprendizado em programação orientada a objetos.

* 1. Dificuldades

Usar programação orientada a objeto no trabalho de conclusão de curso de técnico em informática, não foi uma escolha simples. Existe uma dificuldade na maioria dos alunos, inclusive dos autores, nesta área. A escolha não foi feia porque gostam de programar ou tem facilidade de programar, mas sim como uma forma de aprender melhor a programar e extinguir a dificuldade de programar em Java, além de suprir uma necessidade do campus.

Na escolha de fazer um sistema, ao invés de pesquisa, partiu-se de um auto desafio de programar em Java. Obviamente, que durante o processo de aprendizagem ocorreram vários erros, momentos de não saber o que fazer diante do erro. No entanto, após concluir o sistema, essas dificuldades foram extintas. O sentimento ao concluir o sistema IFMEMO e ajudar o campus numa necessidade que se tornava um problema sério é de ter o poder em suas mãos.

* 1. Pontos positivos

Além da geração de memorandos eletrônicos, o IFMEMO também oferece ao usuário as opções de funcionário. Podem ser utilizadas como uma melhor forma de organizar seus funcionários para consulta de dados necessários para criação do memorando.

* 1. Trabalhos futuros

Embora o sistema já esteja concluído, o IFMEMO, assim como desenvolvimento de qualquer software, pode ser sujeitado a melhorias. A navegação é tida em dois menus, memorando e funcionário. Ao navegar nos menus de memorando, faz-se necessário para o usuário o acesso de informação dos funcionários, para que preencha alguns campos do memorando, como a matricula. Uma possível melhoria seria a disponibilização rápida destas informações quando o usuário precisar, para evitar que ele volte ao menu de funcionário, consulte a matricula daquele funcionário e anotar ou decorar para poder preencher.

Além disso, pode melhorar também na forma de tratamento, dependendo do setor do funcionário. Outra melhoria, quando o sistema já estiver com mais tempo de uso, os memorandos mais antigos continuarem registrado, mas sem acesso rápido. O acesso pode ser permitido através de outro local, arquivo, por exemplo. Assim, evita que na caixa de seleção de memorandos para visualizar (Figura 8 - Tela de escolha de memorando para visualizar.) haja um excesso de opções.

* 1. Fechamento

A idéia de desenvolver um software de apoio a Secretaria Acadêmica implementado para a plataforma web e projetado para melhorar a comunicação interna no campus João Câmara do IFRN, foi testada para comprovação de resultados. Depois de concluído o software foi mostrado ao cliente, Renato Silva (Secretário Acadêmico), para avaliação se as funcionalidades atende as necessidades. Segundo Renato, com auxilio da opinião de Devison Lopes, a ideia é boa e o sistema atende as necessidades da secretaria ao que se refere ao controle de memorandos, porem como algumas melhorias.

# REFERÊNCIAS

1. SOMMERVILLE, I. **Engenharia de Software.**8. ed. Pearson Addison-Wesley, ISBN 978-85-88639-28-7, 2007.
2. PRABHAKAR, M. **Páginas JavaServer™ (JSP)**. Tradução e revisão técnica: Jorge Duarte Pires Valério. Rio de Janeiro/RJ: LTC; 2010.
3. OLIVEIRA, C. H. P. **SQL: Curso pratico.** 7ª ed. São Paulo/SP: Novatec; 2011.
4. DATE, C. J. **Introdução a sistemas de banco de dados**. 8ª ed. Rio de Janeiro:Campus; 2004.
5. **APOSTILA JAVA E ORIENTAÇÃO A OBJETOS**. Disponível em: <<http://www.caelum.com.br/apostila-java-orientacao-objetos/o-que-e-java/>.> Acesso em: 30 de outubro de 2014.
6. **JAVA.** Disponível em:

<<http://www.oracle.com/br/technologies/java/overview/index.html>.> Acesso em: 30 de outubro de 2014.

1. **OBTENHA INFORMAÇÕES SOBRE A TECNOLOGIA JAVA**. Disponível em: <<https://www.java.com/pt_BR/about/>.> Acesso em: 30 de outubro de 2014.
2. **APOSTILA JAVA PARA DESENVOLVIMENTO WEB**. Disponível em: <<http://www.caelum.com.br/apostila-java-web/javaserver-pages/#6-1-colocando-o-html-no-seu-devido-lugar>.> Acesso em: 30 de outubro de 2014.
3. **INTRODUÇÃO AO JAVA SERVER PAGES - JSP**. Disponível em: <<http://www.devmedia.com.br/introducao-ao-java-server-pages-jsp/25602>.> Acesso em: 30 de outubro de 2014.
4. **MANUAL DE REDAÇÃO DA PRESIDÊNCIA DA REPÚBLICA.**  Disponível em: <<https://www.planalto.gov.br/ccivil_03/manual/manual.htm>> Acesso em: 26 de fevereiro de 2014.
5. **MEMORANDO**. Disponível em: <<http://suporte.ufsj.edu.br/manuais/index.php/Protocolo>> Acesso em: 20 de janeiro de 2015.
6. **SECRETARIA ACADEMICA**. Disponível em: <<http://portal.ifrn.edu.br/campus/novacruz/setores/secretaria-academica.html>> Acesso em: 20 de janeiro de 2015.
7. **RESPONSABILIDADES E ATRIBUIÇÕES ACADEMICAS**. Disponível em: <<http://www.vtohenrique.seed.pr.gov.br/redeescola/escolas/23/2880/301/arquivos/File/ResponsabilidadeseAtribuicoesdoSecretarioEscolar.pdf>> Acesso em: 20 de janeiro de 2015.
8. **SGBD MySQL**. Disponível em: <<https://fit.faccat.br/~jonis/Artigo_mySQL.pdf>> Acesso em: 27 de fevereiro de 2015.
9. **LEVANTAMENTO DE REQUISITOS: VOCÊ SABE O QUE É?**. Disponível em: <<http://www.profissionaisti.com.br/2011/06/levantamento-de-requisitos-voce-sabe-o-que-e/>> Acesso em: 2 de março de 2015.
10. **Apostila SQL**. Disponível em: <http://sistemas.riopomba.ifsudestemg.edu.br/dcc/materiais/833034638\_apostila-sql.PDF> Acesso em: 20 de janeiro de 2015.

# APENDICES

## APENDICE A

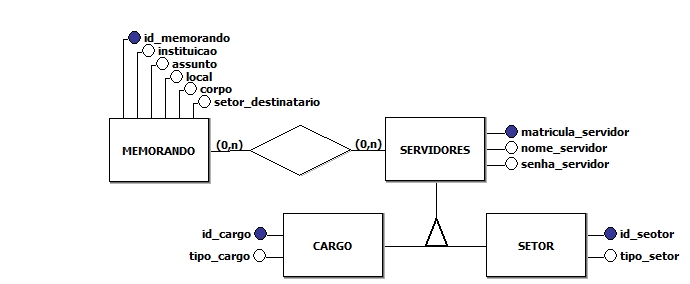
****

Figura 22 - Diagrama de Entidade e Relacionamento do Banco de Dados do sistema.

## APENDICE B

Tabela 1 - Requisitos funcionais do IFMEMO

|  |  |  |
| --- | --- | --- |
| **Nome** | **Descrição** | **Categoria** |
| Autenticação | Autenticação fornecendo login e senha para entrar no sistema. | Evidente |
| Listar memorandos | Mostra uma lista com todos os memorandos criados. | Evidente |
| Pesquisar memorando | O usuário do sistema poderá pesquisar um memorando existente. | Evidente |
| Visualizar e imprimir memorando | O usuário do sistema poderá visualizar um memorando criado anteriormente e imprimi-lo. | Evidente |
| Criar memorando | Cria um novo memorando | Evidente |
| Listar funcionários | Mostra uma lista com todos os funcionários cadastrados. | Evidente |
| Pesquisar funcionário | O usuário do sistema poderá pesquisar um funcionário cadastrado. | Evidente |
| Inserir funcionário | Cria um novo funcionário | Evidente |
| Atualizar funcionário | Atualiza um funcionário existente | Evidente |
| Excluir funcionário | Exclui um funcionário existente | Evidente |

APENDICE C

Tabela 2 - Requisitos não funcionais do IFMEMO

|  |  |  |
| --- | --- | --- |
| **Nome** | **Descrição** | **Categoria** |
| Prazo | O sistema deve estar pronto no final de fevereiro, tendo o tempo de aproximadamente quatro (06) meses para o seu desenvolvimento. | Obrigatório |
| Usabilidade | Boa diagramação dos componentes das páginas Web dinâmicas | Obrigatório |
| Segurança | Deve haver confidencialidade das informações | Obrigatório |
| Robustez | O sistema deve atender ao objetivo de confiabilidade do sistema e executar sem erros recorrentes. | Desejável |
| Compatibilidade | Deve ser compatível com os navegadores Internet Explorer 9, Mozilla Firefox,  Google Chrome | Desejável |
| Base de dados | MySQL | Obrigatório |
| Tecnologia de desenvolvimento | Deve ser desenvolvido usando JSP | Obrigatório |