

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE INFORMÁTICA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

HALUAN HAYAMA DA SILVA

SISTEMA GERENCIADOR DE EXPERIMENTOS PARA O MOTOR DE
INDUÇÃO TRIFÁSICO DO LABORATÓRIO DE CONTROLE E
AUTOMAÇÃO – MÓDULO DE EXPERIMENTAÇÃO

TRABALHO DE DIPLOMAÇÃO

CORNÉLIO PROCÓPIO

2013

HALUAN HAYAMA DA SILVA

**SISTEMA GERENCIADOR DE EXPERIMENTOS PARA O MOTOR DE
INDUÇÃO TRIFÁSICO DO LABORATÓRIO DE CONTROLE E
AUTOMAÇÃO – MÓDULO DE EXPERIMENTAÇÃO**

Proposta de trabalho de conclusão de curso de graduação do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Coordenação do curso de Tecnologia em Análise e Desenvolvimento de Sistemas – COADS – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Elias Canhadas Genvigir

CORNÉLIO PROCÓPIO

2013

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me iluminar e abençoar minha trajetória.

Ao meu pai Celso Frederico da Silva, e minha mãe Cristina Tieme Hayama da Silva, pelo apoio e incentivo de vencer e conquistar meus objetivos, por tudo que sempre fizeram por mim, pela simplicidade, exemplo, amizade, e carinho, fundamentais na construção do meu caráter, pois sem eles, jamais teria a oportunidade de ter estudado na Universidade Tecnológica Federal do Paraná.

Aos meus familiares por me ajudarem e me apoiarem.

Ao meu orientador Prof. Dr. Elias Canhadas Genvigir, por me guiar e ter me dado à oportunidade de participar deste projeto, por ajudar com seus ensinamentos, paciência e por sempre colocar caminhos que no qual, poderia trilhar sem medo.

Aos professores, mestres e doutores que lecionaram durante o período do curso.

Aos meus colegas de turma que percorreram esta caminhada ao meu lado e todos os meus amigos que me apoiaram e torceram pelo meu sucesso.

RESUMO

DA SILVA, Haluan Hayama. Sistema gerenciador de experimentos para o motor de indução trifásica do Laboratório de Controle e Automação – Módulo Experimentação. 2012. 34 f. Trabalho de Diplomação (Tecnologia em Análise e Desenvolvimento de Sistemas), Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2012.

Esta proposta descreve um projeto cujo objetivo é o desenvolvimento de um sistema web para o gerenciamento dos experimentos de um motor de indução trifásica. O sistema irá permitir que usuários externos ao laboratório, devidamente cadastrados no sistema, possam realizar experimentos utilizando o equipamento do Laboratório de Controle e Automação. Além de fornecer uma base de dados para o armazenamento dos experimentos realizados. Sendo assim, o desenvolvimento do projeto irá trazer ao processo de experimentação do laboratório critérios como: armazenamento, confiabilidade, extensibilidade e usabilidade.

Palavras-chave: Gerenciamento de experimentos. Motor de indução trifásica. Usuários externos. Armazenamento. Laboratório de Controle de Automação.

ABSTRACT

DA SILVA, Haluan Hayama. System manager experiments for three-phase induction motor Laboratory Automation and Control – Experimentation Module. 2012. 34 f. Trabalho de Diplomação (Tecnologia em Análise e Desenvolvimento de Sistemas), Federal Technology University - Paraná. Cornélio Procópio, 2012.

This proposal describes a project whose goal is the development of a web system for the managing experiments for a three-phase induction motor. The system will allow users outside the laboratory, duly registered in the system, can perform experiments using the equipment of the Laboratory and Automation Control. In addition to providing a database for the storage of experiments performed. Thus, the development of the project will bring to the process of experimentation laboratory criteria as: storage, reliability, extensibility and usability.

Keywords: *Management experiments. Three-phase induction motor. External users. Storage. Laboratory Automation Control.*

LISTA DE ILUSTRAÇÕES

<i>Figura 1 - Diagrama de Pacotes</i>	17
<i>Figura 2 - Diagrama de Subsistemas</i>	19
<i>Figura 3 - Módulo Experimentação</i>	22
<i>Figura 4 - Módulo Exportar Dados</i>	23
<i>Figura 5 - Módulo Backup</i>	24
<i>Figura 6 - Diagrama de Casos de uso</i>	31
<i>Figura 7 - Diagrama de Classes</i>	37
<i>Figura 8 - Diagrama de Atividades Manter Experimento - Visão Usuário</i>	39
<i>Figura 9 - Diagrama de Atividades - Visão Operador</i>	40
<i>Figura 10 - Diagrama de Atividades Manter Experimento Não Linear</i>	41
<i>Figura 11 - Diagrama de Atividades Consultar Log do Experimento</i>	42
<i>Figura 12 - Diagrama de Atividades Backup</i>	43
<i>Figura 13- Diagrama de Atividades Protocolo de Mensagens</i>	44
<i>Figura 14 - Diagrama de Atividades Exportação de Dados</i>	45
<i>Figura 15 - Diagrama de Sequência Manter Log – Visão do Administrador</i>	47
<i>Figura 16 - Diagrama de Sequência Log do Operador</i>	48
<i>Figura 17 - Diagrama de Sequência Log do Usuário</i>	49
<i>Figura 18 - Diagrama de Seqüência Manter Experimento – (Visão Usuário)</i>	50
<i>Figura 19 - Diagrama de Seqüência Manter Experimento Não Linear - Usuário.</i>	51
<i>Figura 20 - Diagrama de Seqüência Manter Experimento – Operador</i>	52
<i>Figura 21 - Diagrama de Seqüência - Manter Experimento Expandir Operador</i>	53
<i>Figura 22 - Diagrama de Seqüência Exportação de Dados</i>	54
<i>Figura 23 - Modelo Relacional</i>	56
<i>Figura 24 – Fase 0 do Processo de Experimentação</i>	57
<i>Figura 25 – Fase 0 do Processo de Experimentação</i>	58
<i>Figura 26 – Fase 0 Cadastro de Experimento Não Linear</i>	59
<i>Figura 27 – Fase 0 Adicionar Variável</i>	60
<i>Figura 28 – Fase 1 do Processo de Experimentação</i>	60
<i>Figura 29 - Fase 2 do Processo de Experimentação</i>	61
<i>Figura 30 – Motivos da Negação de Liberação para o Agendamento</i>	62
<i>Figura 31 - Fase 3 do Processo de Experimentação</i>	63
<i>Figura 32 - Fase 4 do Processo de Experimentação</i>	64
<i>Figura 33 - Log do Experimento - Visão do Operador</i>	65
<i>Figura 34 - Log dos Experimentos - Visão do Usuário</i>	66
<i>Figura 35 - Log dos Experimentos - Visão Administrador</i>	67
<i>Figura 36 – Classe Temporizador</i>	68
<i>Figura 37 - Classe Backup</i>	68
<i>Figura 38 – Arquivo back.bat</i>	69
<i>Figura 39 - Adicionando arquivo dll na raiz do projeto</i>	69
<i>Figura 40 - Criando Vetor de Tarefa</i>	70
<i>Figura 41 - Verifica o tipo da variável</i>	70

<i>Figura 42 - Iniciando a Tarefa</i>	71
<i>Figura 43 – Executando a Tarefa</i>	71
<i>Figura 44 - Listagem de Experimentos para Download</i>	72
<i>Figura 45 - Arquivo .txt com os dados de um Experimento executado</i>	73
<i>Figura 46 - Diagrama de Implantação</i>	75
<i>Figura 47 - Configuração do TCP/IPV4</i>	81
<i>Figura 48 - Bibliotecas do Projeto</i>	83
<i>Figura 49 - Diagrama de Casos de uso</i>	20
<i>Figura 50 - Diagrama de Pacotes</i>	23
<i>Figura 51 - Diagrama de Subsistemas</i>	25
<i>Figura 52 - Módulo Experimentação</i>	28
<i>Figura 53 - Módulo Exportar Dados</i>	29
<i>Figura 54 – UP</i>	30

LISTA DE TABELAS

<i>Tabela 1 - Relação de Requisitos com Casos de uso - Módulo Experimentação.....</i>	<i>21</i>
<i>Tabela 2 - Relação de Requisitos com Casos de uso - Módulo Exportar dados.....</i>	<i>23</i>
<i>Tabela 3 - Requisitos funcionais (contínua).....</i>	<i>27</i>
<i>Tabela 4 - Respectivas funções dos atores no sistema</i>	<i>30</i>
<i>Tabela 5 - Especificação do Caso de uso Manter Experimento (contínua).....</i>	<i>32</i>
<i>Tabela 6 - Especificação do Caso de uso Manter Experimento Não Linear (contínua) 33</i>	
<i>Tabela 7 - Especificação do Caso de Uso Manter Log</i>	<i>34</i>
<i>Tabela 8 - Especificação do Caso de Uso Protocolo de Mensagens</i>	<i>35</i>
<i>Tabela 9 - Especificação do Caso de Uso Exportação de Dados</i>	<i>36</i>

LISTA DE SIGLAS

<i>APF</i>	<i>Análise de Pontos por Função</i>
<i>CPF</i>	<i>Cadastro de Pessoas Físicas</i>
<i>IDE</i>	<i>Integrated Development Environment</i>
<i>JDBC</i>	<i>Java Database Connectivity</i>
<i>JSF</i>	<i>Java Server Faces</i>
<i>RG</i>	<i>Registro Geral</i>
<i>SGBDOR</i>	<i>Sistema de Gerenciamento de Banco de Dados Relacional</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>USB</i>	<i>Universal Serial Bus</i>
<i>VT</i>	<i>Virtual Terminal</i>

LISTA DE ACRÔNIMOS

<i>AJAX</i>	<i>Asynchronous Javascript And XML</i>
<i>CIPECA</i>	<i>Centro Integrado de Pesquisa em Controle e Automação</i>
<i>MATLAB</i>	<i>MATrix LABoratory</i>
<i>MIT</i>	<i>Motor de Indução Trifásica</i>
<i>UI</i>	<i>User Interface</i>

SUMÁRIO

1. INTRODUÇÃO	12
1.1 OBJETIVOS GERAIS.....	12
2. COMPREENDER O DOMÍNIO DA APLICAÇÃO	14
2.1 DESCRIÇÃO DO DOMÍNIO DO NEGÓCIO	14
2.2 DESCRIÇÃO DAS CARACTERÍSTICAS DO NEGÓCIO	14
2.3 COMPREENDER O PROBLEMA A SER RESOLVIDO	15
2.4 DESCRIÇÃO DAS COMPLEXIDADES (RESTRIÇÕES DO SISTEMA).....	16
3. ARQUITETURA INICIAL	17
3.1 DIAGRAMA DE PACOTES	17
3.2 DIAGRAMA DE SUBSISTEMAS	18
3.2.1 Subsistemas a serem desenvolvidos neste trabalho.....	21
4. COMUNICAÇÃO COM A AQUISITORA DE DADOS	25
5. DESENVOLVIMENTO.....	27
5.1 FASE DE CONCEPÇÃO.....	27
5.1.1 Requisitos funcionais.....	27
5.1.2 Requisitos não funcionais	29
5.1.3 Tarefa - Identificar atores e funções	30
5.1.4 Definir casos de uso.....	30
5.2 FASE DE ELABORAÇÃO	32
5.2.1 Especificação dos Casos de Uso.....	32
5.2.2 Diagrama de classes	36
5.3 FASE DE CONSTRUÇÃO.....	38
5.3.1 Diagrama de Atividades.....	38
5.3.2 Diagramas de Seqüência.....	46
5.3.3 Modelo Relacional.....	55
5.4 IMPLEMENTAÇÃO DO SUBSISTEMA DE EXPERIMENTAÇÃO	57
5.4.1 Manter Experimento	57
5.4.2 Protocolo de Mensagens	64
5.4.3 Log do Experimento.....	65
5.5 IMPLEMENTAÇÃO DO SUBSISTEMA DE BACKUP	67
5.6 IMPLEMENTAÇÃO DO SUBSISTEMA DE Exportação DE dados.....	69

5.7	FASE DE TRANSIÇÃO	73
5.7.1	DIAGRAMA DE IMPLANTAÇÃO	74
6.	CONCLUSÃO.....	76
7.	TRABALHOS FUTUROS	77
8.	REFERÊNCIAS	78
	APÊNDICE A – IMPLANTAÇÃO DO SISTEMA	80
	APÊNDICE B – CASOS DE TESTES.....	89
	Caso de Teste Cadastro de Usuário.....	90
	Caso de Teste Login.	90
	Caso de Teste Cadastrar Experimento - Usuário.....	91
	Caso de Teste Solicitar liberação para o agendamento- Usuário	91
	Caso de Teste Liberação Experimento - Operador.....	92
	Caso de Teste Agendamento - Administrador.....	92
	Caso de Teste Agendamento - Usuário.....	93
	Caso de Teste Execução do Experimento - Operador.....	94
	Caso de Teste Download dos dados do Experimento	94
	APÊNDICE C – PROPOSTA DO TRABALHO DE DIPLOMAÇÃO	96

1. INTRODUÇÃO

Esta proposta descreve um projeto cujo objetivo é o desenvolvimento de um sistema Web para o gerenciamento de uma bancada de ensaios com o MIT (Motor de Indução Trifásica) que está situado no Laboratório de Controle e Automação da Universidade Tecnológica Federal do Paraná - Campus Cornélio Procópio.

O sistema proposto é composto por vários módulos que foram desenvolvidos por dois alunos do curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas, ou seja, o sistema terá seu desenvolvimento dividido em partes da seguinte maneira:

- *Módulo de Administração;*
- *Módulo de Agendamento;*
- *Módulo de Experimentação;*
- *Módulo de Exportação de Dados;*
- *Módulo de Backup.*

É responsabilidade desta proposta o desenvolvimento dos módulos de: Experimentação, Exportação de Dados e Backup. Os módulos de Administração e Agendamento foram desenvolvidos pelo aluno Marcus Vinicius Sanches Sant' Anna.

Este trabalho está dividido em capítulos, que são apresentados da seguinte maneira:

- *Capítulo 2 - Compreender o Domínio da Aplicação;*
- *Capítulo 3 - Compreender as Necessidades dos Usuários;*
- *Capítulo 4 - Tecnologias;*
- *Capítulo 5 - Arquitetura Inicial;*
- *Capítulo 6 - Metodologia;*
- *Capítulo 7 - Pontos por Função;*
- *Capítulo 8 - Cronograma;*
- *Capítulo 9 - Referências.*

1.1 OBJETIVOS GERAIS

O sistema a ser desenvolvido possui como objetivos:

- 1) *Permitir o acesso remoto para que usuários externos realizem experimentações.*
- 2) *Possuir uma base de dados que armazene usuários, instituições, variáveis, sensores e experimentos.*
- 3) *Possuir um protocolo de mensagens entre operador e usuário.*
- 4) *Disponibilizar uma agenda para gerenciar os experimentos dos usuários.*
- 5) *Possuir uma base de dados que permita o acesso rápido para a consulta dos dados armazenados.*
- 6) *O sistema deve contemplar itens de confiabilidade e usabilidade.*

2. COMPREENDER O DOMÍNIO DA APLICAÇÃO

Este capítulo apresenta o resultado das atividades de levantamento de requisitos do sistema proposto. Tais atividades iniciam-se pela descrição do domínio do negócio na subseção 2.1. Uma descrição de como ocorre atualmente o processo de experimentação utilizando o MIT e o algoritmo no MATLAB é relatada na subseção 2.2. Os problemas do domínio do negócio a serem resolvidos são apresentados na subseção 2.3. E por fim, a subseção 2.4 mostra quais são as restrições do sistema.

2.1 DESCRIÇÃO DO DOMÍNIO DO NEGÓCIO

O Laboratório de Controle e Automação, situado na sala nove no CIPECA (Centro Integrado de Pesquisa em Controle e Automação), é supervisionado pelo Prof. Dr. Alessandro Goedel e tem como objetivo realizar experimentos com motores e máquinas elétricas.

2.2 DESCRIÇÃO DAS CARACTERÍSTICAS DO NEGÓCIO

O Laboratório possui um MIT, onde os estagiários e alunos que cursam o mestrado de Engenharia Elétrica realizam os experimentos.

Em um único experimento é possível coletar todas as variáveis disponíveis pelo equipamento de experimentação ou selecionar as variáveis que desejar.

O procedimento de experimentação ocorre quando o operador regula as variáveis de tensão para cada uma das três fases do MIT através de um painel de controle.

Durante o processo de experimentação é possível alterar a tensão em cada uma das fases. Ao alterar a tensão em uma das fases, ocorrerá uma variação de tensão nas fases restantes.

O operador inicia o software MATLAB e abre um arquivo contendo um algoritmo. Neste algoritmo é modificado a taxa de amostragem e o tempo de aquisição.

A taxa de amostragem é medida em amostras/segundo com um sinal de 60 Hertz e o tempo de aquisição é medido em segundos.

Após a regulagem da taxa de amostragem e do tempo de aquisição, o operador executa o algoritmo e liga o motor através do painel de controle. É necessário aguardar alguns segundos entre essas duas operações para evitar a perda de dados.

Os sensores utilizados no equipamento são os seguintes:

- *Três do modelo LV25-P;*
- *Três do modelo LAH25-NP;*
- *Um Como Torque Type 4700;*
- *Um Varimot V.V -120.*

Os sensores servem para captar respectivamente os valores das variáveis de tensão, corrente, torque e velocidade. Esses valores são enviados para a placa aquisitora que está conectada a um computador.

Ao terminar a execução do experimento, são coletadas as variáveis que o operador designou no início da experimentação. Os dados são transferidos para uma matriz no formato txt e enviados ao MATLAB.

A matriz possui oito colunas, na qual as variáveis são divididas na seguinte maneira:

- *Três de tensão;*
- *Três de corrente;*
- *Uma para o torque;*
- *Uma para a velocidade.*

A quantidade de linhas existentes em uma matriz é definida pelo valor da taxa de amostragem (valores coletados ao longo do tempo).

2.3 COMPREENDER O PROBLEMA A SER RESOLVIDO

Atualmente o processo de experimentação possibilita o acesso ao equipamento apenas para os usuários internos do laboratório. A idéia é permitir que usuários externos a instituição e remotamente localizados possam, através do uso da Internet acessar o equipamento do Laboratório de Controle e Automação e realizar os seus experimentos.

Os usuários devem previamente estar cadastrados no sistema e possuírem direitos de acesso ao sistema de controle de experimentação.

Outro ponto a ser analisado é que os dados dos experimentos são armazenados em uma matriz no formato txt, o que pode dificultar as operações de consulta e controle de experimentos.

O sistema a ser desenvolvido visa estabelecer a comunicação entre a placa de Aquisição National Instruments USB6009 e o computador.

Durante a execução do experimento as seguintes variáveis são coletadas:

- *Tensão;*
- *Corrente;*
- *Torque;*
- *Velocidade.*

Será necessária a criação de uma base de dados para o armazenamento e o gerenciamento das variáveis e dos experimentos executados. A base de dados também proporcionará um controle de usuários, o agendamento de experimentação, além de facilitar as consultas e geração de relatórios.

Outro ponto a ser abordado é que o sistema deve ter a premissa de elementos de: confiabilidade, operação (tanto local quanto remota), interface amigável e um feedback rápido em suas operações.

2.4 DESCRIÇÃO DAS COMPLEXIDADES (RESTRICÇÕES DO SISTEMA)

Podem ser observadas as seguintes complexidades referentes ao desenvolvimento do sistema:

- *A aplicação deverá ser desenvolvida usando tecnologia Web.*
- *A busca de Drivers compatíveis com a placa aquisitora para realizar a comunicação com o sistema.*
- *A criação de uma interface que atenda aos requisitos de confiabilidade, extensibilidade e usabilidade.*

3. ARQUITETURA INICIAL

Neste capítulo é apresentada a arquitetura a ser utilizada no desenvolvimento do sistema. A subseção 3.1 descreve os pacotes a serem utilizados no sistema, a subseção 5.2 apresenta a divisão em módulos e por fim, a subseção 3.2.1 trata dos módulos a serem desenvolvidos neste trabalho conforme a Figura 2.

3.1 DIAGRAMA DE PACOTES

O Diagrama de pacotes descreve os pacotes ou partes do sistema, divididos em agrupamentos lógicos.

Na Figura 1 são apresentados os devidos pacotes do sistema, juntamente com as suas dependências.

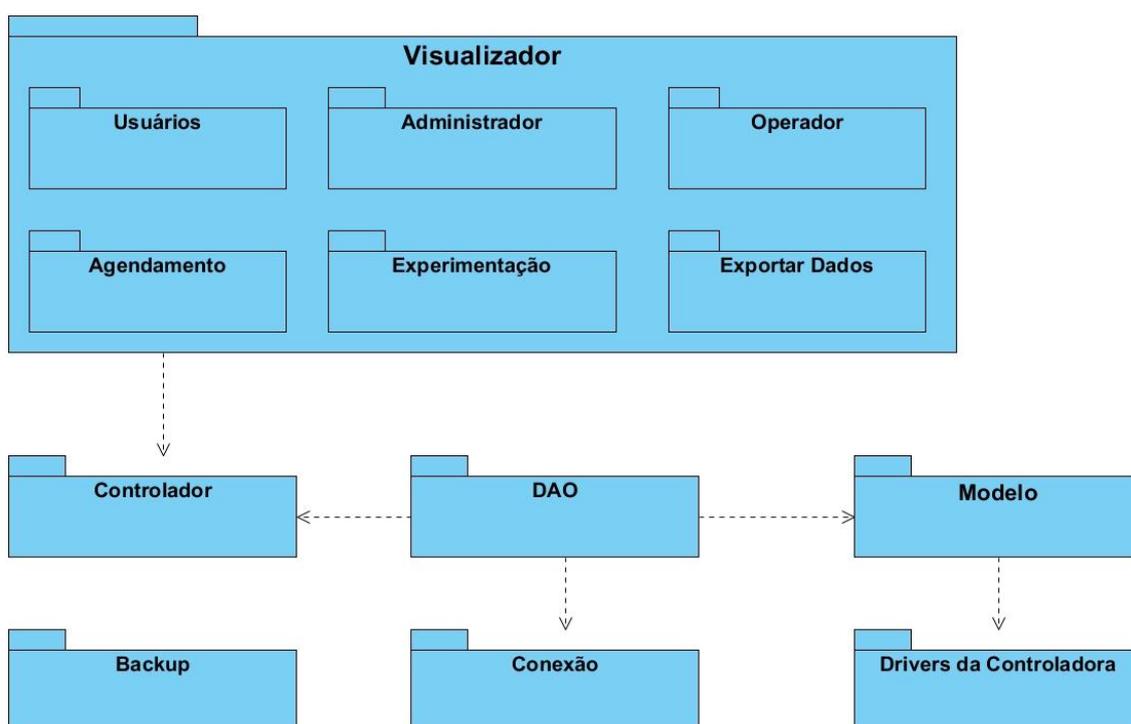


Figura 1 - Diagrama de Pacotes

O diagrama apresentado na Figura 1 exibe os pacotes que são descritos a seguir:

- *DAO: Possui as classes com as regras de negócio do sistema.*

- *Modelo: Possui os beans das entidades (GET e SET).*
- *Conexão: Possui a classe de conexão com o banco de dados.*
- *Backup: Possui classes com a função de realizar o backup.*
- *Controlador: Possui classes com a função de gerenciar o redirecionamento das páginas do sistema.*
- *Pacote Visualizador: Possui páginas xhtml que são armazenadas nos seguintes pacotes:*
 - *Operador;*
 - *Usuário;*
 - *Administrador;*
 - *Agendamento;*
 - *Experimentação;*
 - *Exportar Dados;*
 - *Drivers da Controladora.*

3.2 DIAGRAMA DE SUBSISTEMAS

O diagrama de subsistemas representa os subsistemas envolvidos no sistema. Além de representar as dependências entre os módulos.

A Figura 2 demonstra o comportamento dos subsistemas do sistema em questão, sendo que a parte destacada em vermelho representa os subsistemas a serem desenvolvidos nesta proposta.

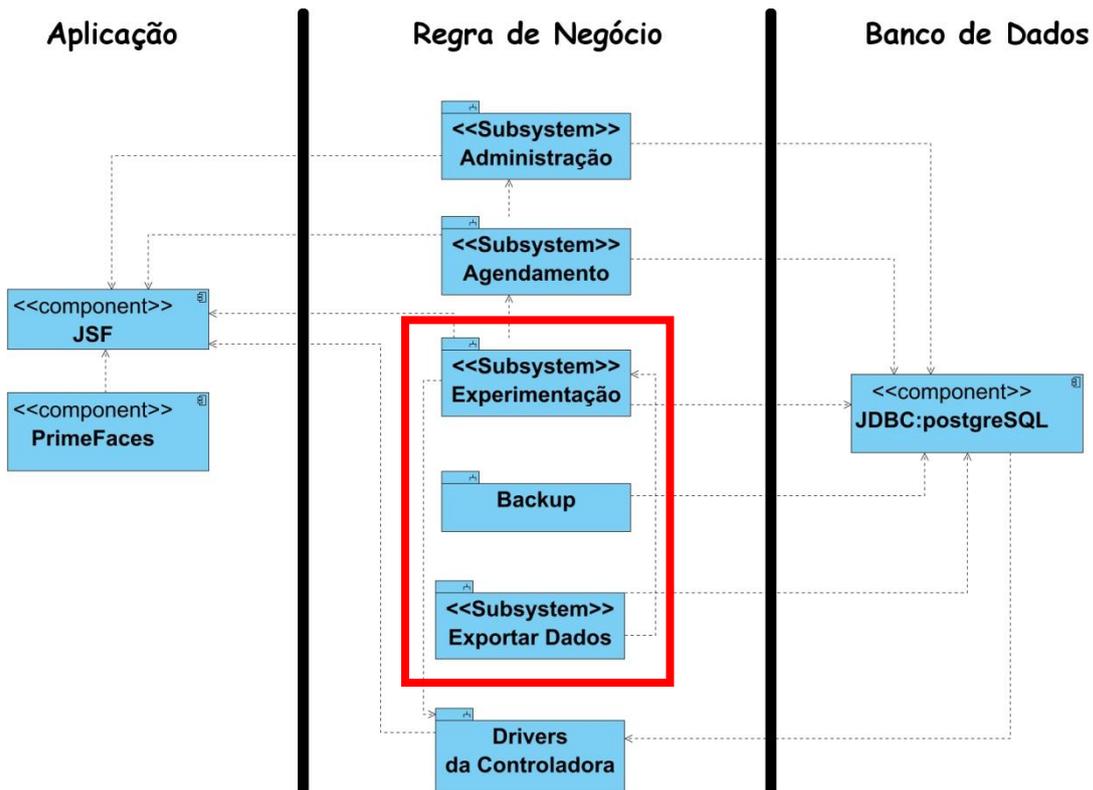


Figura 2 - Diagrama de Subsistemas

O diagrama apresentado na Figura 2 exibe os subsistemas que são descritos a seguir:

- *Administração:* Neste módulo serão implementados os cadastros do sistema exceto o cadastro de experimentação. O módulo é constituído pelos seguintes diagramas de pacotes:

- *Administrador;*
- *Usuário;*
- *Operador;*
- *Bean;*
- *DAO;*
- *Conexão.*

- *Agendamento:* O módulo de agendamento é responsável por gerenciar os dias e horários disponíveis para a realização de experimentos, inclusive a quantidade de experimentos a ser realizado por dia. Este módulo é constituído pelos seguintes pacotes:

- *Agendamento;*

- *Bean;*
 - *DAO;*
 - *Conexão;*
 - *Usuário;*
 - *Operador;*
 - *Administração.*
- *Experimentação: O módulo de experimentação tem a função de cadastrar e gerenciar os experimentos. Este módulo é constituído pelos seguintes pacotes:*
- *Experimentação;*
 - *Operador;*
 - *Usuário;*
 - *DAO;*
 - *BEAN;*
 - *Conexão.*
- *Drivers da Controladora: O módulo de Drivers da controladora possui os drivers para a comunicação com a placa aquisitora. Este módulo é constituído pelos seguintes pacotes:*
- *Controladora;*
 - *Conexão;*
 - *DAO.*
- *Backup: O módulo de Backup tem como função criar uma cópia de segurança dos dados do sistema. Este módulo é constituído pelos seguintes pacotes:*
- *Backup;*
 - *Conexão;*
- *Exportar Dados: O módulo de exportar dados tem como função criar arquivos de saída dos dados dos experimentos e disponibilizá-los no sistema. Este módulo é constituído pelos seguintes pacotes:*
- *Experimentos;*
 - *Exportar Dados.*

3.2.1 Subsistemas a serem desenvolvidos neste trabalho

Esta proposta de trabalho compõe os seguintes módulos a serem desenvolvidos:

- *Experimentação: O módulo de experimentação irá cadastrar e gerenciar os experimentos.*

A realização do experimento é dividida em fases. Inicialmente, o usuário envia uma solicitação de cadastro de experimento ao operador, após a confirmação de solicitação de cadastro será iniciada a fase seguinte que é o agendamento do experimento. E finalmente, na última fase o experimento será executado e os dados coletados serão armazenados na base de dados e disponibilizados no sistema.

A cada fase do processo de experimentação serão armazenados logs com informações como data e horário em que a fase ocorreu. Além de, enviar mensagens automáticas para o e-mail do usuário, informando a situação do pedido do seu experimento.

A relação dos casos de uso da Figura 6 com os requisitos da Tabela 3 que pertencem a este módulo são apresentados na Tabela 1.

Tabela 1- Relação de Requisitos com Casos de uso - Módulo Experimentação

Casos de Uso	Código dos Requisitos									
	RF21	RF22	RF23	RF24	RF25	RF26	RF27	RF28	RF29	RF30
<i>Manter Experimento</i>										
<i>Manter Log</i>	RF36									
<i>Protocolo de Mensagens</i>	RF32	RF35	RF38	RF39	RF40					
<i>Gerar Relatório</i>	RF33	RF34	RF37	RF41	RF42	RF43				

Conforme o diagrama de classes da figura 3 é possível observar uma dependência deste módulo com os módulos de Administração e Agendamento.

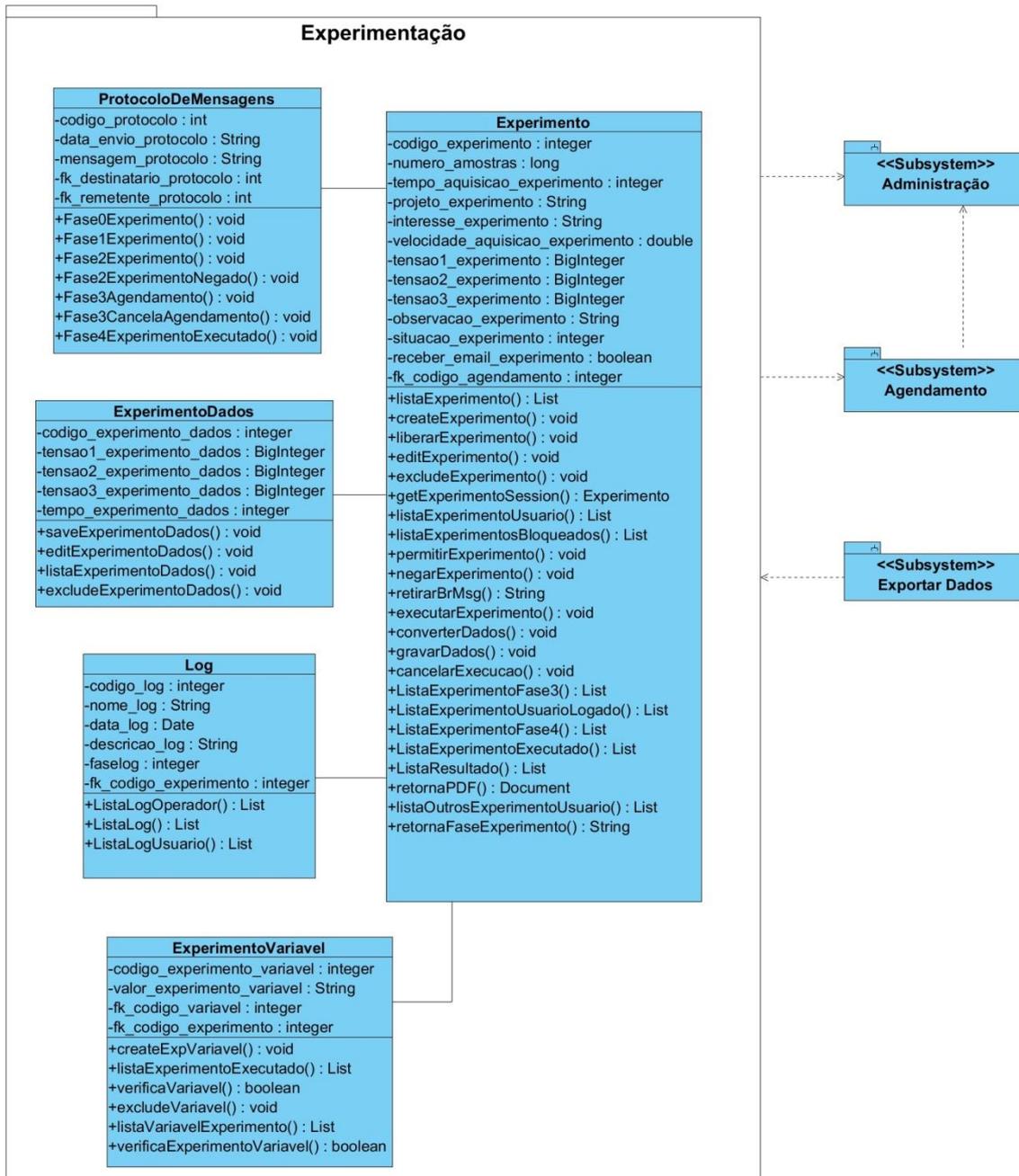


Figura 3 - Módulo Experimentação

- *Exportar dados:* O módulo de Exportação de dados irá capturar os dados do experimento e transformar em um arquivo de saída.

Após a realização do experimento, o sistema irá automaticamente gerar um arquivo de saída no formato txt com os dados coletados na experimentação. Este arquivo será disponibilizado posteriormente no sistema para que o usuário possa fazer download.

A relação dos casos de uso da Figura 6 com os requisitos da Tabela 3 que pertencem a este módulo são apresentados na Tabela 2.

Tabela 2 - Relação de Requisitos com Casos de uso - Módulo Exportar dados

Casos de Uso	Código dos Requisitos
Exportar Dados	RF31

Conforme o diagrama de classes da figura 4 este subsistema depende do módulo de experimentação.

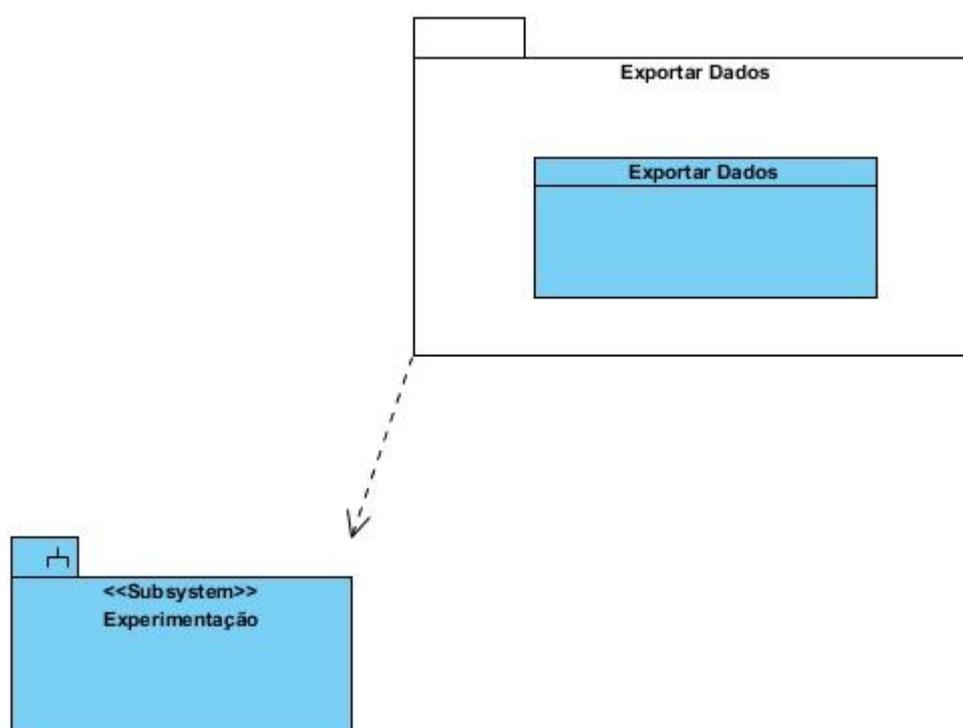


Figura 4 - Módulo Exportar Dados

- *Backup:* O módulo de Backup irá criar uma cópia de segurança dos dados de cadastros e experimentos do sistema. Esta cópia será utilizada caso o sistema esteja corrompido por algum erro.

O sistema irá realizar o backup, em um horário com poucos usuários. Este backup será realizado diariamente.

Conforme o diagrama da figura 5.

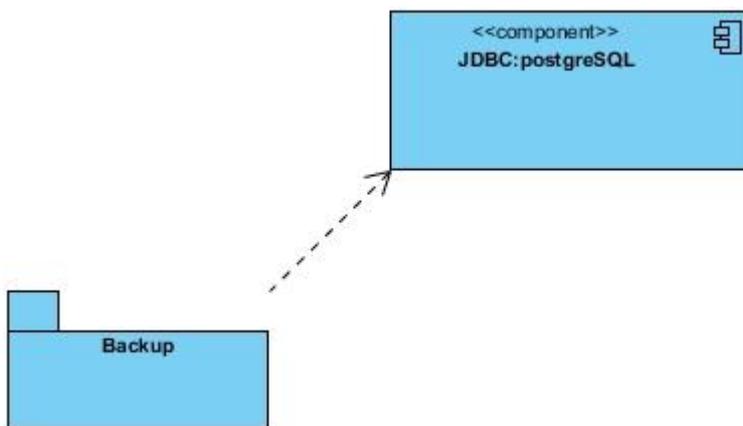


Figura 5 - Módulo Backup

4. COMUNICAÇÃO COM A AQUISITORA DE DADOS

Neste capítulo é apresentada a forma de comunicação com a placa aquisitora NIUSB-6221.

Primeiramente, é necessário explicar o que é o software NI-DAQmx.

O NI-DAQmx proporciona aumento da produtividade e desempenho em instrumentação virtual e computador baseado em aquisição de dados. Esta ferramenta promove:

- Uma interface de programação simples para a programação de entrada e saída analógica, digital e os contadores em centenas de dispositivos de hardware DAQ multifuncionais;*
- Funções nas linguagens Visual Basic, Visual Studio .NET, e C/C++.*
- O NI-DAQmx também oferece a possibilidade de se criar um simulador para a aquisitora. Utilizando o simulador é possível realizar os testes para a experimentação sem a utilização do equipamento do laboratório. O simulador é limitado ao uso de apenas dois tipos de variável (Corrente e Tensão), as variáveis do tipo velocidade e torque não podem ser simulados nesta ferramenta.*

Antes de relatar como foi feita a comunicação com a aquisitora, é necessário explicar as idéias iniciais de como realizar esta tarefa e realçar pontos importantes do sistema utilizando o MATLAB.

O sistema que utiliza o MATLAB conforme explicado na subseção 2.2, executa um algoritmo que realiza a comunicação com o driver da placa. Este algoritmo foi desenvolvido utilizando a linguagem C que é uma das linguagens suportadas pelo driver.

O novo sistema que está sendo implantado no laboratório, utiliza a linguagem JAVA, que não é uma linguagem compatível com o driver.

Inicialmente foi planejado utilizar a tecnologia de Webservices para resolver o problema. O Webservices permite que diferentes aplicações, em diferentes plataformas, possam interagir entre si, disponibilizando serviços. Apesar de esta tecnologia poder solucionar essa complexidade, ela não foi utilizada.

A solução adotada foi a de criar um driver que possa utilizar a linguagem JAVA.

O driver foi criado e testado pelo aluno de graduação em Engenharia da Computação Ítalo Macedo Laino. Esta atividade ocorreu em paralelo com este estágio curricular.

O novo driver foi desenvolvido na IDE Eclipse Indigo 3.7, utilizando a tecnologia JNI (Java Native Interface). Ele executa as mesmas funções do driver original da NIDA Qmx.

5. DESENVOLVIMENTO

Este capítulo apresenta as fases de execução do projeto, juntamente com a modelagem e os testes realizados com os usuários.

5.1 FASE DE CONCEPÇÃO

Na fase de concepção foi realizado o levantamento e a delimitação do escopo do projeto. Além disso, foram desenvolvidas as atividades de refinação dos requisitos que foram levantados durante a proposta deste trabalho e a especificação dos cenários dos casos de uso.

5.1.1 Requisitos funcionais

Os requisitos funcionais apresentam as principais funcionalidades do sistema, ou seja, eles descrevem o que o sistema deve fazer. Todavia, não apresentam como tais itens serão feitos. A tabela 3 apresenta os requisitos do sistema que foram levantados durante entrevistas com o Prof. Alessandro Goedel e os estagiários do laboratório.

Tabela 3 - Requisitos funcionais (contínua)

Código	Descrição
RF01	<i>O sistema deverá permitir a inclusão, exclusão e edição de usuários. O usuário pode ser interno ou externo. Um usuário possui os seguintes dados: nome, CPF, RG, login, senha e instituição.</i>
RF02	<i>Para o manuseio do painel de controle do equipamento o usuário externo precisará de um operador disponível. O usuário interno poderá operar este painel.</i>
RF03	<i>Após o usuário realizar o seu cadastro, será enviado uma solicitação de confirmação de cadastro ao email do administrador. Será necessária a confirmação do administrador para que o usuário tenha acesso ao sistema.</i>
RF04	<i>O administrador poderá alterar e consultar os seus dados. Um administrador possui os seguintes dados: nome, CPF, RG, login e senha.</i>
RF05	<i>O sistema deverá permitir a inclusão, edição e exclusão de instituições. Uma instituição possui os seguintes dados: nome, setor, campus.</i>
RF06	<i>O sistema deverá permitir o relatório de usuários por instituições.</i>
RF07	<i>O sistema deverá possuir um cadastro das variáveis já utilizadas nos experimentos, que são as seguintes: tensão, corrente, torque e velocidade. Uma variável possui os seguintes dados: nome, unidade de medida e sigla.</i>

Tabela 3 – Requisitos funcionais (contínua)

Código	Descrição
RF08	<i>Além de gerenciar as variáveis atuais já utilizadas, o sistema deverá permitir a inclusão de novas variáveis.</i>
RF09	<i>Apenas o administrador terá a permissão para cadastrar, editar e excluir as variáveis.</i>
RF10	<i>Uma variável possui apenas um sensor associado a ela.</i>
RF11	<i>O sistema deverá emitir relatórios de variáveis por sensores.</i>
RF12	<i>O sistema deverá permitir a inclusão, edição e exclusão de sensores. Um sensor possui os seguintes dados: modelo, fabricante, ano de aquisição e descrição.</i>
RF13	<i>O sistema deverá gerenciar os sensores que atualmente o equipamento possui e permitir a inclusão de novos sensores.</i>
RF14	<i>Apenas o administrador terá a permissão para cadastrar, editar e excluir sensores.</i>
RF15	<i>O sistema deverá realizar a inclusão, exclusão e edição dos fatores de escala. Um fator de escala possui os seguintes dados: nome, data, descrição e valor.</i>
RF16	<i>Apenas o administrador poderá cadastrar o fator de escala.</i>
RF17	<i>Um fator de escala está associado apenas a um sensor.</i>
RF18	<i>O sistema deverá realizar a inclusão, exclusão e edição de operadores. Um operador possui os seguintes dados: nome, CPF, RG, login e senha.</i>
RF19	<i>Apenas o administrador terá a permissão para cadastrar um operador.</i>
RF20	<i>Um operador está associado a um ou vários experimentos.</i>
RF21	<i>O sistema deverá permitir o cadastro de experimentos. Um experimento possui os seguintes dados: tempo de aquisição, taxa de amostragem, variáveis, horário, interesse e projeto.</i>
RF22	<i>Durante o cadastro de experimentos, o sistema deverá permitir que o usuário altere o valor de uma ou mais tensões e determinar o tempo em que ocorrerá a alteração.</i>
RF23	<i>O sistema deverá permitir que usuários externos realizem experimentos.</i>
RF24	<i>O sistema deverá permitir que o usuário escolha as variáveis que serão coletadas nos experimentos.</i>
RF25	<i>Um usuário poderá consultar todos os experimentos cadastrados na base de dados.</i>
RF26	<i>Os experimentos a serem realizados pelos usuários externos deverão possuir a confirmação de um operador.</i>
RF27	<i>Um operador poderá consultar todos os experimentos cadastrados na base de dados.</i>
RF28	<i>O administrador poderá consultar todos os experimentos cadastrados.</i>
RF29	<i>O sistema deverá permitir que apenas o administrador possa excluir os experimentos já realizados.</i>
RF30	<p><i>O processo para a realização de um experimento é dividido em cinco fases:</i></p> <ol style="list-style-type: none"> <i>1. Usuário envia solicitação de experimento ao operador.</i> <i>2. Operador envia uma resposta ao usuário. Caso a resposta seja negativa, o operador irá escrever uma justificativa explicando por que o experimento não pode ser realizado.</i> <i>3. Após a aprovação da fase anterior, o operador irá agendar o experimento</i> <i>4. O experimento será executado. Caso haja motivos para o cancelamento, será enviado um e-mail ao usuário explicando os motivos.</i>

Tabela 3 – Requisitos funcionais (conclusão)

Código	Descrição
	<i>5. Após a realização do experimento, os dados coletados serão armazenados na base de dados e disponibilizados no sistema.</i>
<i>RF31</i>	<i>O sistema deverá fornecer um arquivo de saída no formato txt com os dados referentes ao experimento.</i>
<i>RF32</i>	<i>O sistema deverá possuir um protocolo de mensagens. Um protocolo de mensagens possui: código, mensagem, remetente, destinatário e data de envio.</i>
<i>RF33</i>	<i>O sistema deverá emitir um relatório de quantos experimentos um usuário realizou.</i>
<i>RF34</i>	<i>O sistema deverá emitir um relatório de quais instituições realizaram experimentos.</i>
<i>RF35</i>	<i>Durante as fases do processo serão enviadas mensagens automaticamente para o e-mail do usuário, informando a situação do pedido de experimentação.</i>
<i>RF36</i>	<i>Para cada fase do processo, o sistema deverá armazenar um log com as seguintes informações: nome, data e descrição.</i>
<i>RF37</i>	<i>O sistema deverá emitir um relatório com os logs dos experimentos.</i>
<i>RF38</i>	<i>O usuário poderá consultar apenas o seu protocolo de mensagens dos seus experimentos.</i>
<i>RF39</i>	<i>O operador poderá consultar apenas o protocolo de mensagens dos experimentos que estejam associados a ele.</i>
<i>RF40</i>	<i>O administrador poderá consultar todos os protocolos de mensagens dos experimentos.</i>
<i>RF41</i>	<i>O administrador poderá emitir relatórios de todos os experimentos cadastrados.</i>
<i>RF42</i>	<i>O operador poderá emitir relatórios apenas de experimentos associados a ele.</i>
<i>RF43</i>	<i>O usuário poderá emitir relatórios apenas de experimentos associados a ele.</i>
<i>RF44</i>	<i>O sistema deverá proporcionar o agendamento dos experimentos.</i>
<i>RF45</i>	<i>A agenda de experimentação deverá possuir os seguintes dados: anos, meses, dias disponíveis, horários e quantidade de experimentos.</i>
<i>RF46</i>	<i>A agenda de experimentação é gerenciada apenas pelo administrador.</i>
<i>RF47</i>	<i>Caso o experimento esteja fora das cargas nominais, conjugado ou com desbalanceamento de tensão, o sistema deverá proporcionar um intervalo de pelo menos uma hora até o próximo experimento.</i>
<i>RF48</i>	<i>O agendamento de experimentação será realizado por um operador.</i>
<i>RF49</i>	<i>O sistema deverá emitir relatórios dos experimentos agendados por dia, mês e ano.</i>

5.1.2 Requisitos não funcionais

Os requisitos não funcionais descrevem restrições sobre o processo de desenvolvimento. Foram levantados os seguintes requisitos não funcionais do sistema:

- 1) O sistema deverá possuir três níveis de acesso (usuário, operador e administrador).
- 2) Deverá ser implementado um controle de visão ao acesso. Este controle irá restringir o que cada usuário poderá visualizar no sistema.
- 3) O sistema deverá ser confiável e que possibilite a segurança dos dados armazenados.

5.1.3 Tarefa - Identificar atores e funções

Os atores participam no sistema realizando as suas respectivas funções. A função reflete o papel que o ator desempenha no sistema.

A tabela 4 representa os atores e as suas respectivas funções no sistema.

Tabela 4 - Res pectivas funções dos atores no sistema

Ator	Função
<i>Operador</i>	<ol style="list-style-type: none"> 1. Consultar todos os experimentos cadastrados na base de dados. 2. Consultar apenas os protocolos de mensagens dos experimentos que estejam associados a ele. 3. Confirmar a solicitação de experimentos dos usuários. 4. Realizar o agendamento dos experimentos. 5. Regular as variáveis de tensão no painel de controle, conforme informado no pedido de experimento. 6. Executar os experimentos agendados.
<i>Usuário</i>	<ol style="list-style-type: none"> 1. Gerenciar os seus dados. 2. Enviar solicitações de experimentos. 3. Consultar todos os experimentos cadastrados na base de dados. 4. Consultar apenas os protocolos de mensagens dos experimentos que estejam associados a ele.
<i>Administrador</i>	<ol style="list-style-type: none"> 1. Realizar o cadastro de operadores. 2. Realizar o cadastro de sensores. 3. Realizar o cadastro do fator de escala. 4. Realizar o cadastro de variáveis. 5. Confirmar a solicitação de cadastro do usuário. 6. Criar a agenda de experimentos. 7. Consultar todos os experimentos cadastrados na base de dados. 8. Consultar os protocolos de mensagens de todos os experimentos.

5.1.4 Definir casos de uso

O diagrama de casos de uso descreve um cenário que mostra as funcionalidades do sistema.

Um diagrama de casos de uso possui os seguintes componentes: ator, casos de uso e seus respectivos relacionamentos.

O ator é representado por um boneco. Um ator é um usuário do sistema, que pode ser um usuário humano ou um sistema computacional.

O caso de uso é representado por uma elipse. Um caso de uso define uma funcionalidade do sistema.

Em um relacionamento o ator pode se comunicar com um ou mais casos de uso, da mesma forma um caso de uso pode se relacionar com um ou mais atores.

Na Figura 6 é apresentado o diagrama de caso de uso do sistema.

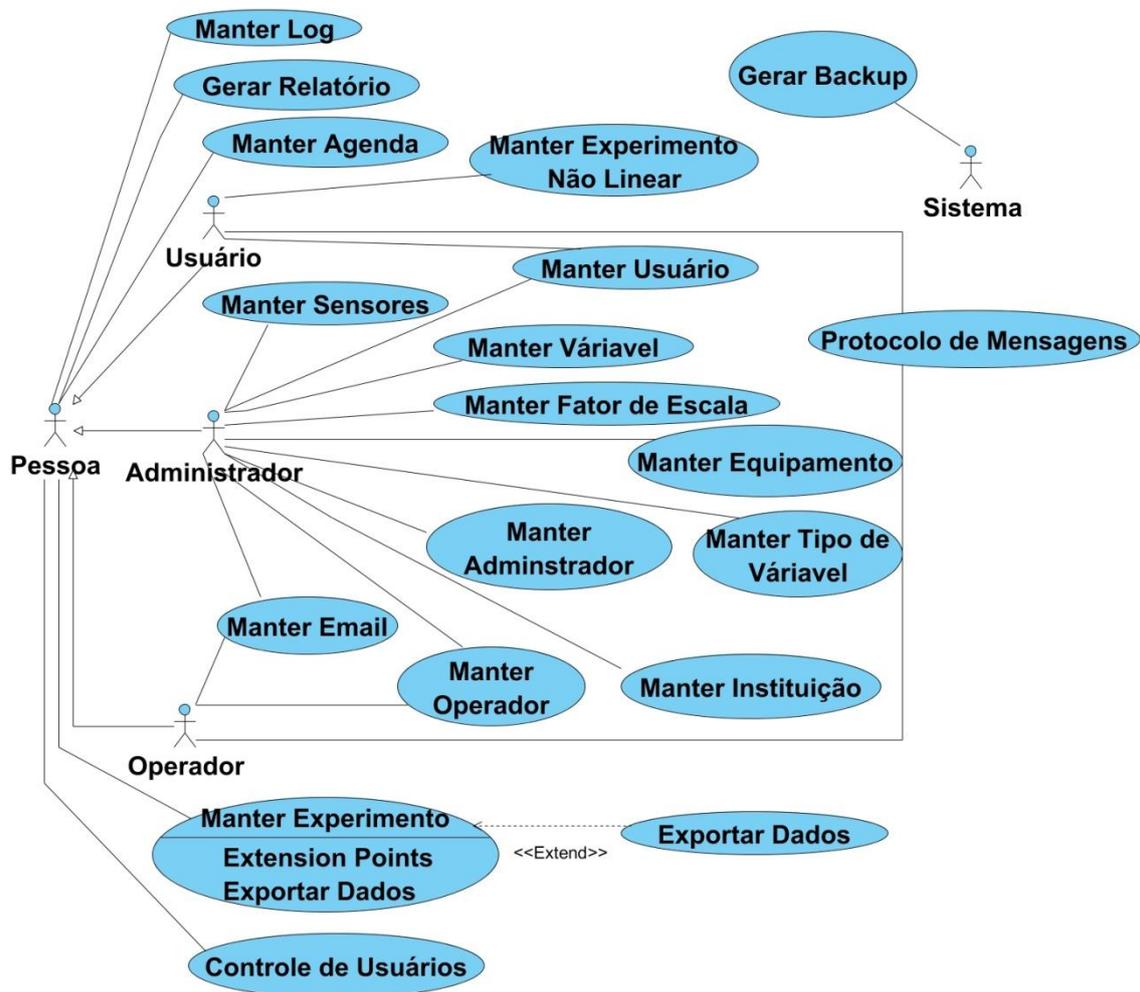


Figura 6 - Diagrama de Casos de uso

5.2 FASE DE ELABORAÇÃO

O objetivo principal da fase de elaboração é criar uma base estável da arquitetura que será o guia para a fase de Construção.

5.2.1 Especificação dos Casos de Uso

A partir da Figura 6 foi elaborada uma descrição para cada um de seus casos de uso.

A seguir serão apresentadas as especificações dos casos de usos dos módulos de Experimentação, Exportação de Dados e Backup.

Tabela 5 - Especificação do Caso de uso Manter Experimento (contínua)

Caso de uso: <i>Manter Experimento</i>
Descrição: <i>O caso de uso Manter Experimento acontece quando o usuário cadastra, altera ou exclui um experimento.</i>
Atores: <i>Operador e Usuário.</i>
Fluxo de eventos:
<p>Fluxo principal:</p> <ol style="list-style-type: none"> 1. <i>O usuário clica em “Experimentos” no menu principal.</i> 2. <i>O sistema exibe a listagem de experimentos.</i> 3. <i>O usuário clica no “+”</i> 4. <i>O sistema exibe o formulário de cadastro de experimentos.</i> 5. <i>O usuário preenche os dados e clica em “Cadastrar”.</i> 6. <i>O sistema apresenta uma mensagem de “Cadastro realizado com sucesso” e atualiza a listagem de experimentos.</i> <p>Fluxo alternativo:</p> <p>FA1. Alterar Experimento:</p> <ol style="list-style-type: none"> 1. <i>O usuário escolhe um experimento listado e clica na “lupa”.</i> 2. <i>O sistema exibe a tela com o formulário com os dados do experimento.</i> 3. <i>O usuário realiza as modificações e clica em “Alterar”.</i> 4. <i>O sistema apresenta uma mensagem de “Dados alterados com sucesso” e lista o experimento com os dados atualizados.</i> <p>FA2. Excluir Experimento:</p> <ol style="list-style-type: none"> 1. <i>O usuário escolhe um experimento listado e clica no “X”.</i> 2. <i>O sistema exibe uma mensagem de confirmação de exclusão.</i> 3. <i>O usuário clica em “OK” para confirmar a exclusão.</i> 4. <i>O sistema apresenta uma mensagem de “Experimento Excluído” e atualiza a listagem dos experimentos.</i>

Tabela 5 - Especificação do Caso de uso Manter Experimento (conclusão)

<p>FA3. Solicitar Liberação do Experimento.</p> <ol style="list-style-type: none"> 1. O usuário escolhe um experimento e clica na lupa. 2. O sistema exibe a tela com o formulário com os dados do experimento. 3. O usuário clica em “Liberar”. 4. O sistema exibe uma mensagem de confirmação de solicitação de liberação. 5. O usuário clica em “OK” para confirmar a solicitação. 6. O sistema apresenta uma mensagem de “Solicitação enviada” e atualiza a listagem dos experimentos. <p>FA4. Permitir Liberação do Experimento.</p> <ol style="list-style-type: none"> 1. O operador escolhe um experimento e clica em “Permitir”. 2. O sistema exibe uma mensagem de confirmação de liberação. 3. O operador clica em “OK” para confirmar a liberação. 4. O sistema apresenta uma mensagem de “Experimento Permitido” e atualiza a listagem dos experimentos. <p>FA5. Negar Liberação do Experimento.</p> <ol style="list-style-type: none"> 1. O operador escolhe um experimento e clica em “Negar”. 2. O sistema exibe uma mensagem de confirmação de liberação. 3. O operador clica em “OK” para confirmar a liberação. 4. O sistema exibe uma área de texto para o operador explicar os motivos da negação do experimento. 5. O operador escreve os motivos e clica em “Enviar”. 6. O sistema apresenta uma mensagem de “Experimento Negado” e atualiza a listagem dos experimentos. <p>FA6. Executar Experimento.</p> <ol style="list-style-type: none"> 1. O operador escolhe um experimento e clica em “Executar”. 2. O sistema importa os dados a serem coletados pela aquisitora. 3. O sistema exibe uma confirmação para gravar os dados no banco. 4. O operador clica em “OK” para confirmar. 5. O sistema apresenta uma mensagem de “Experimento Executado” e atualiza a listagem dos experimentos. <p>FA7. Campos Inválidos.</p> <ol style="list-style-type: none"> 1. O sistema apresenta uma mensagem informando que os campos estão inválidos.
Pré-Condições: Ator logado no sistema
Pós-Condições: Experimento criado.

Tabela 6 - Especificação do Caso de uso Manter Experimento Não Linear (contínua)

Caso de uso: Manter Experimento Não Linear
Descrição: O caso de uso Manter Experimento Não Linear acontece quando o usuário cadastra, altera ou exclui um experimento não linear.
Atores: Usuário.
Fluxo de eventos:
Fluxo principal:

Tabela 6 - Especificação do Caso de Uso Manter Experimento Não Linear (conclusão)

<ol style="list-style-type: none"> 1. O usuário clica em “Experimentos” no menu principal. 2. O sistema exibe a listagem de experimentos. 3. O usuário clica no “Adicionar Não Linear +” 4. O sistema exibe a listagem dos experimentos não lineares referentes ao experimento. 5. O usuário clica no “+”. 6. O sistema exibe o formulário de experimentos não lineares. 7. O usuário preenche os dados e clica em “Cadastrar”. 8. O sistema apresenta uma mensagem de “Cadastro realizado com sucesso” e atualiza a listagem dos experimentos não lineares. <p>Fluxo alternativo:</p> <p>FA1. Alterar Experimento não linear:</p> <ol style="list-style-type: none"> 1. O usuário clica no “Adicionar Não Linear +” 2. O sistema exibe a listagem dos experimentos não lineares referentes ao experimento. 3. O usuário escolhe um experimento não linear listado e clica na “lupa”. 4. O sistema exibe a tela com o formulário com os dados. 5. O usuário realiza as modificações e clica em “Alterar”. 6. O sistema apresenta uma mensagem de “Dados alterados com sucesso” e lista o experimento não linear com os dados atualizados. <p>FA2. Excluir Experimento não linear:</p> <ol style="list-style-type: none"> 1. O usuário clica no “Adicionar Não Linear +”. 2. O sistema exibe a listagem dos experimentos não lineares referentes ao experimento. 3. O usuário escolhe um experimento não linear listado e clica no “X”. O sistema exibe uma mensagem de confirmação de exclusão. 4. O usuário clica em “OK” para confirmar a exclusão. 5. O sistema apresenta uma mensagem de “Experimento não linear Excluído” e atualiza a listagem dos experimentos não lineares.
<p>Pré-Condições: Ator logado no sistema. Experimento cadastrado.</p>
<p>Pós-Condições: Experimento não linear criado.</p>

Tabela 7 - Especificação do Caso de Uso Manter Log

<p>Caso de uso: Manter Log</p>
<p>Descrição: O caso de uso Manter Log acontece quando o ator deseja consultar o log de mensagens referentes a um experimento.</p>
<p>Atores: Administrador, Operador e Usuário.</p>
<p>Fluxo de eventos:</p> <p>Fluxo principal:</p> <ol style="list-style-type: none"> 1. O usuário clica em “Log” no menu principal. 2. O sistema exibe a listagem dos logs experimentos.

Tabela 7 - Especificação do Caso de Uso Manter Log (conclusão)

3. O usuário clica no experimento.
4. O sistema exibe os dados do log com as mensagens referentes ao experimento.
Pré-Condições: Experimento cadastrado.

Tabela 8 - Especificação do Caso de Uso Protocolo de Mensagens

Caso de uso: Protocolo de Mensagens
Descrição: O caso de uso Protocolo de Mensagens ocorre quando o usuário e o operador trocam mensagens entre si a respeito de um experimento.
Atores: Operador e Usuário.
Fluxo de eventos:
Fluxo principal:
<ol style="list-style-type: none"> 1. O usuário libera um experimento para a Fase 2 do processo de Experimentação. 2. O sistema envia uma mensagem ao operador, informando que existe um experimento que requer a liberação para o agendamento.
Fluxo alternativo:
FA1. Fase 2 Experimento Permitido:
<ol style="list-style-type: none"> 1. O operador permite a liberação do experimento para o agendamento. 2. O sistema envia uma mensagem ao usuário, informando que o experimento foi liberado para a Fase de Agendamento.
FA2. Fase 2 Experimento Negado:
<ol style="list-style-type: none"> 1. O operador nega a liberação do experimento para o agendamento. 2. O operador explica os motivos da negação de liberação do experimento. 3. O sistema envia a mensagem ao usuário, informando que o experimento foi negado para a Fase de Agendamento e explicando os motivos da negação.
FA3. Fase 3 Experimento Agendado:
<ol style="list-style-type: none"> 1. O usuário agenda um experimento. 2. O sistema envia uma mensagem ao operador, informando que existe um experimento que foi agendado.
FA4. Fase 3 Agendamento Cancelado:
<ol style="list-style-type: none"> 1. O usuário cancela o agendamento de um experimento. 2. O sistema envia uma mensagem ao operador, informando que existe um agendamento que foi cancelado.
FA5. Fase 4 Experimento Executado:
<ol style="list-style-type: none"> 1. O operador executa o experimento. 2. O sistema envia uma mensagem ao usuário, informando que o experimento foi executado e os dados já foram disponibilizados.
Pré-Condições: Experimento cadastrado.
Pós-Condições: Mensagem enviada

Tabela 9 - Especificação do Caso de Uso Exportação de Dados

Caso de uso: <i>Exportação de Dados</i>
Descrição: <i>O caso de uso Exportação de dados ocorre quando o ator deseja utilizar os dados coletados de um experimento.</i>
Atores: <i>Administrador, Operador e Usuário.</i>
Fluxo de eventos: Fluxo principal: <ol style="list-style-type: none"> 1. <i>O ator clica em “Experimentos” no menu principal.</i> 2. <i>O sistema exibe a listagem dos experimentos executados.</i> 3. <i>O usuário clica no experimento.</i> 4. <i>O sistema retorna o arquivo .txt para o ator</i> Fluxo alternativo: FA1. Consultar Experimentos: <ol style="list-style-type: none"> 1. <i>O ator escreve o nome do projeto e clica na “lupa”.</i> 2. <i>Sistema retorna a nova listagem de experimentos executados.</i>
Pós-Condições: <i>Retornar arquivo .txt</i>

A especificação do caso de uso Gerar Backup é mostrada conforme a tabela xx.

5.2.2 Diagrama de classes

Nesta seção será representado o diagrama de classes do Sistema gerenciador de experimentos para o motor de indução trifásica do laboratório de controle e automação. Um diagrama de classes de projeto ilustra as especificações para as classes de software e de interfaces de uma aplicação.

A Figura 7 demonstra o diagrama de classes desenvolvido e utilizado para a conclusão deste projeto.

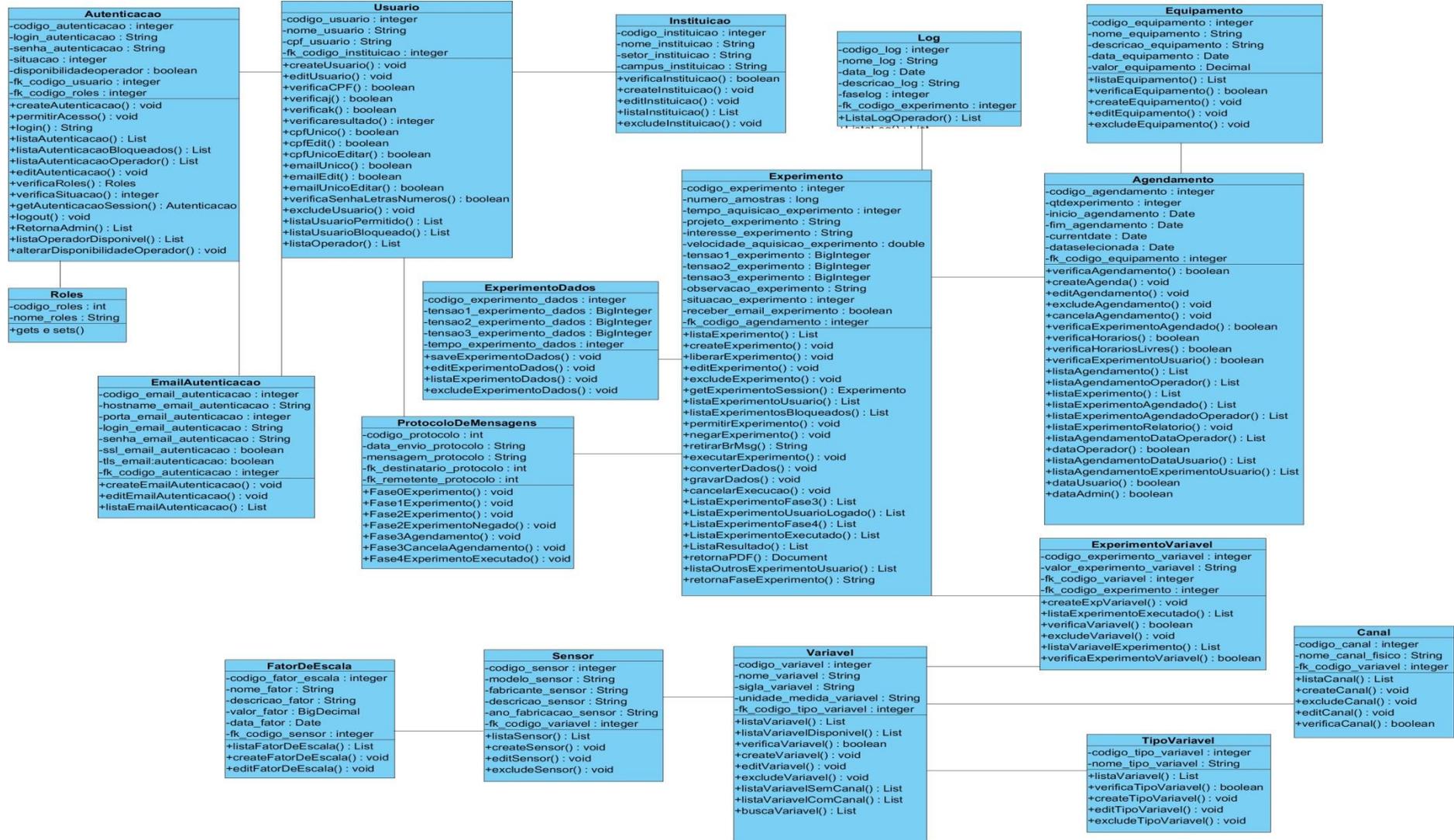


Figura 7 - Diagrama de Classes

5.3 FASE DE CONSTRUÇÃO

O objetivo principal da fase de Construção é a criação de uma versão executável do sistema com base no que foi levantado na fase de Elaboração.

5.3.1 Diagrama de Atividades

Tendo base na especificação dos casos de uso feita no subcapítulo 1.2.2, é possível elaborar os diagramas de atividades.

A seguir são apresentados os diagramas de atividades dos módulos de Experimentação, Exportação de Dados e Backup.

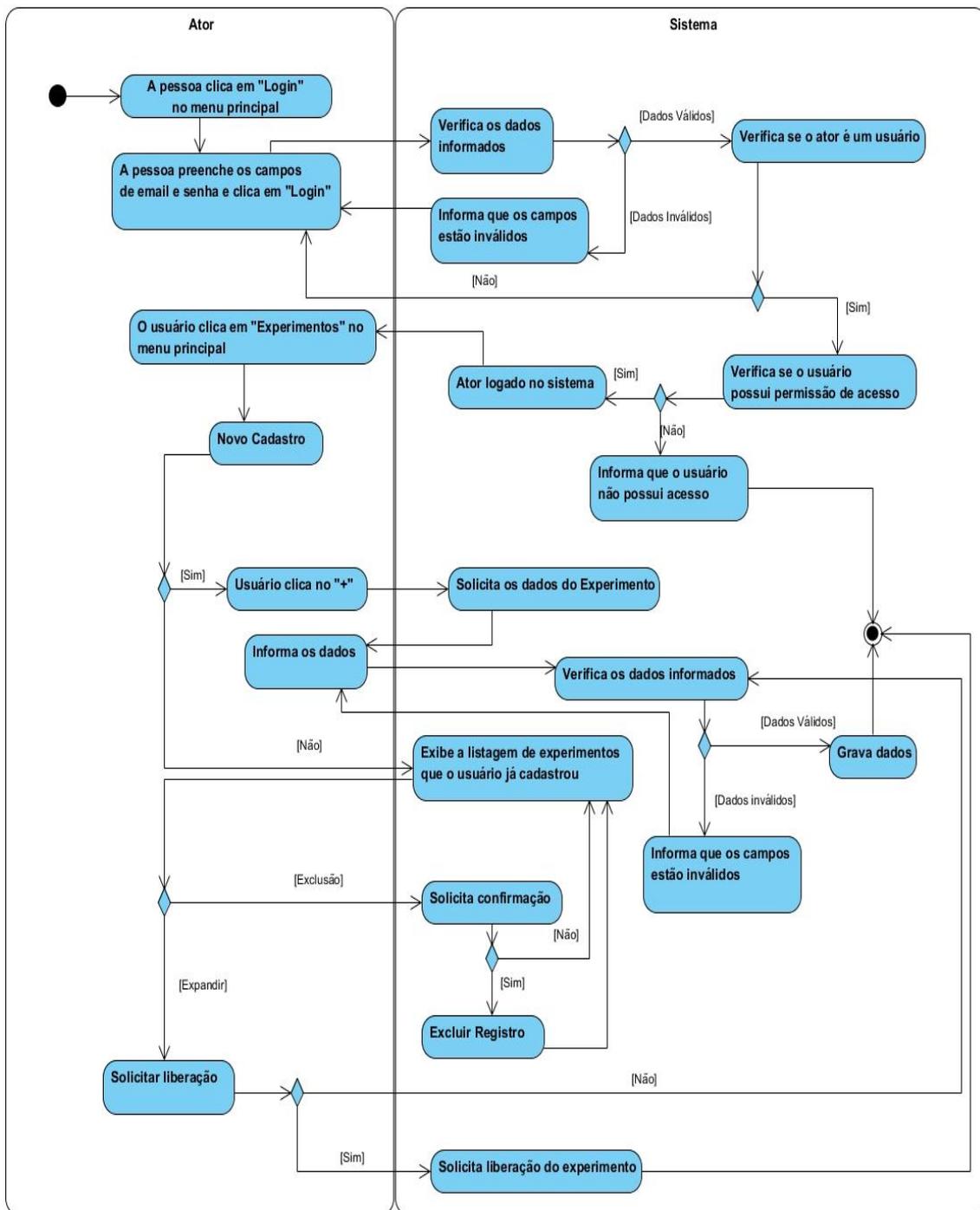


Figura 8 - Diagrama de Atividades Manter Experimento - Visão Usuário

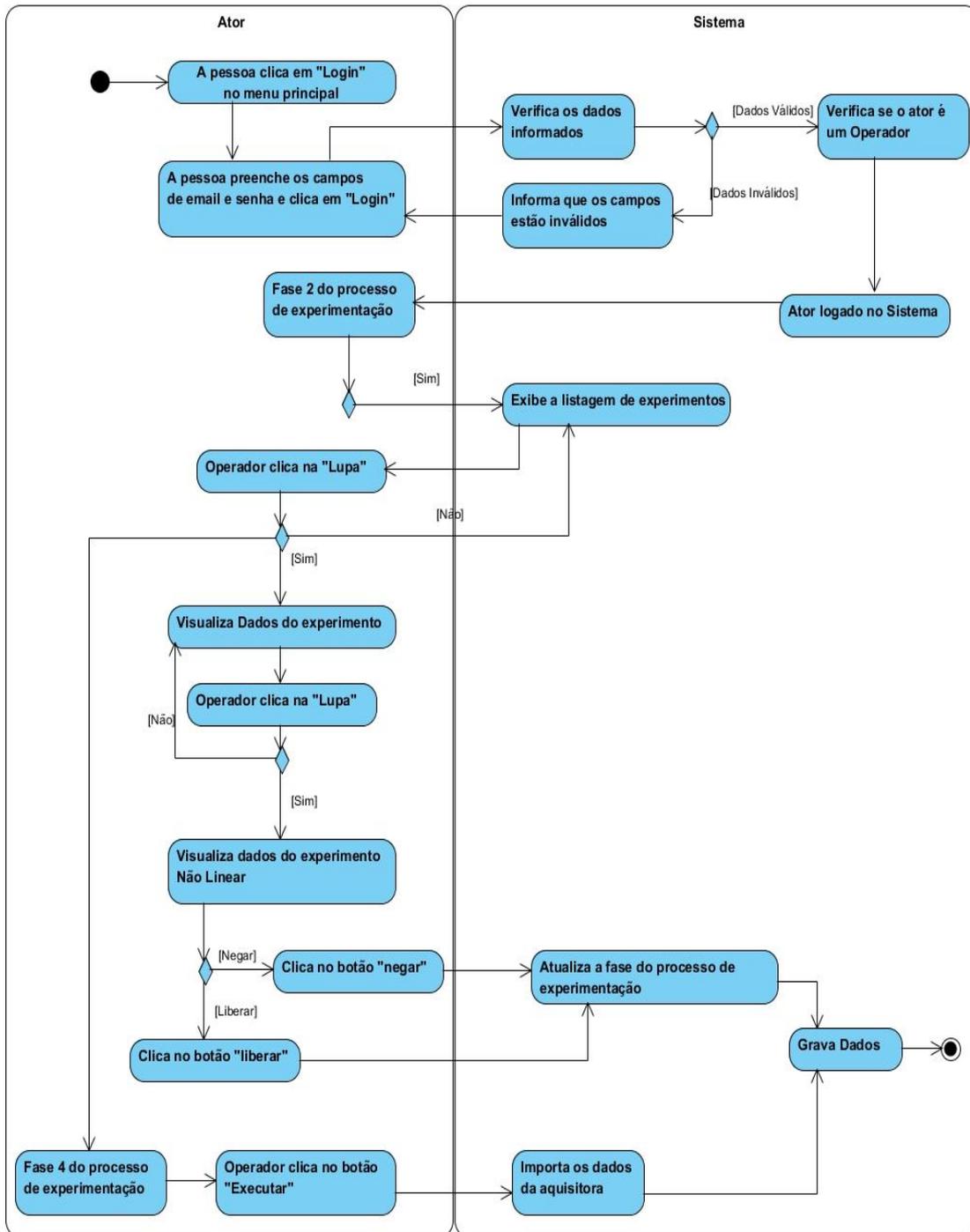


Figura 9 - Diagrama de Atividades - Visão Operador

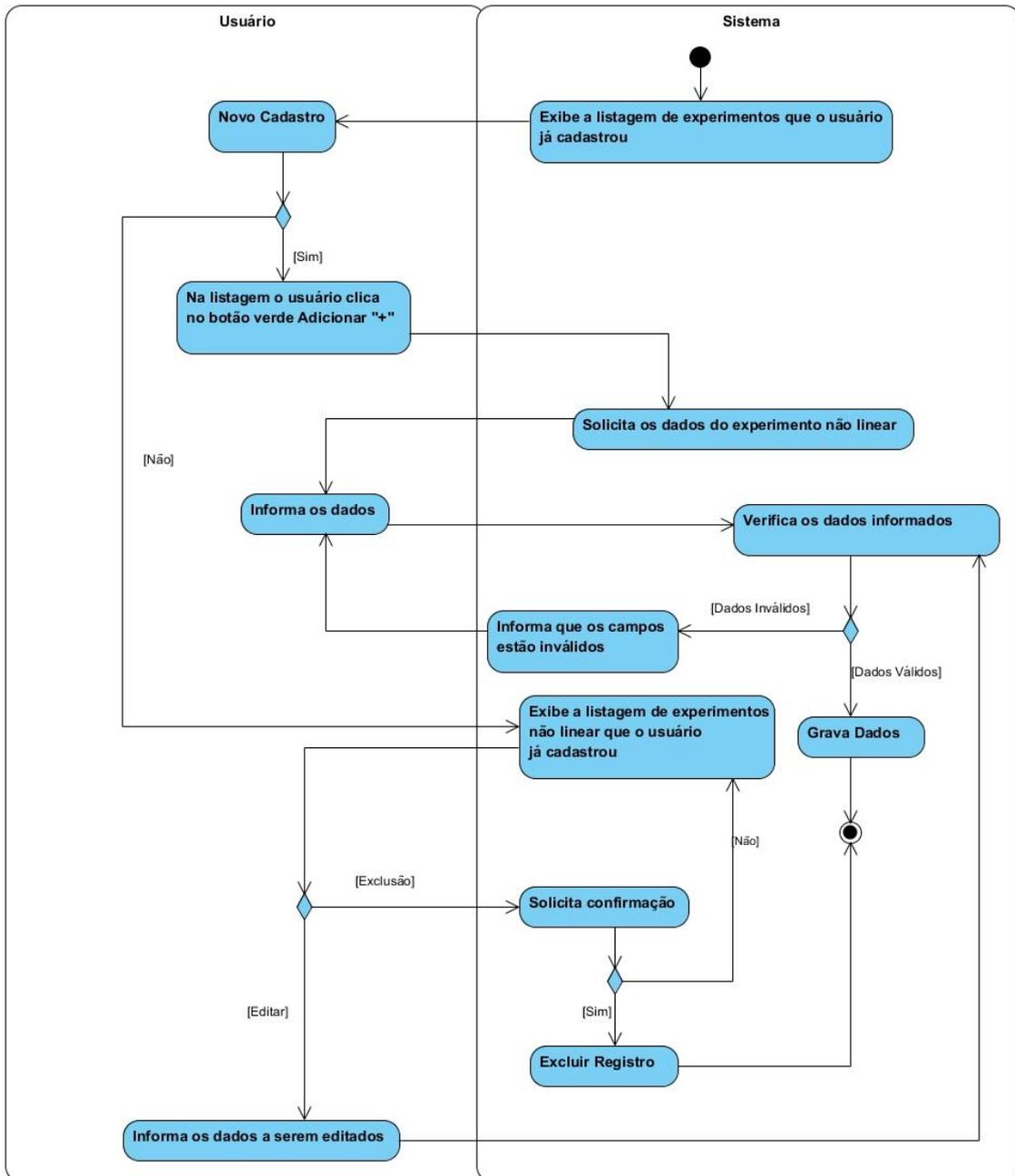


Figura 10 - Diagrama de Atividades Manter Experimento Não Linear

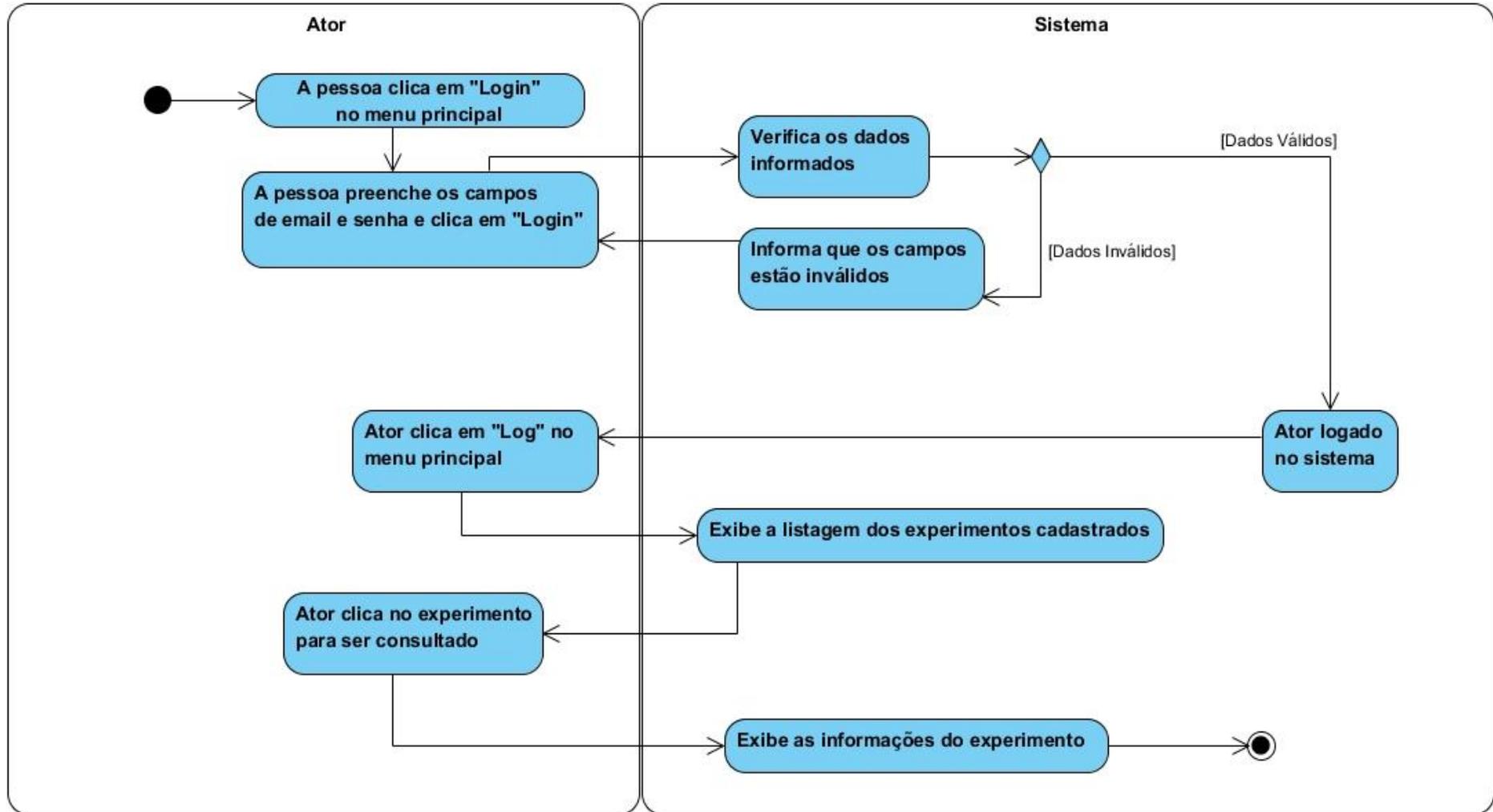


Figura 11 - Diagrama de Atividades Consultar Log do Experimento

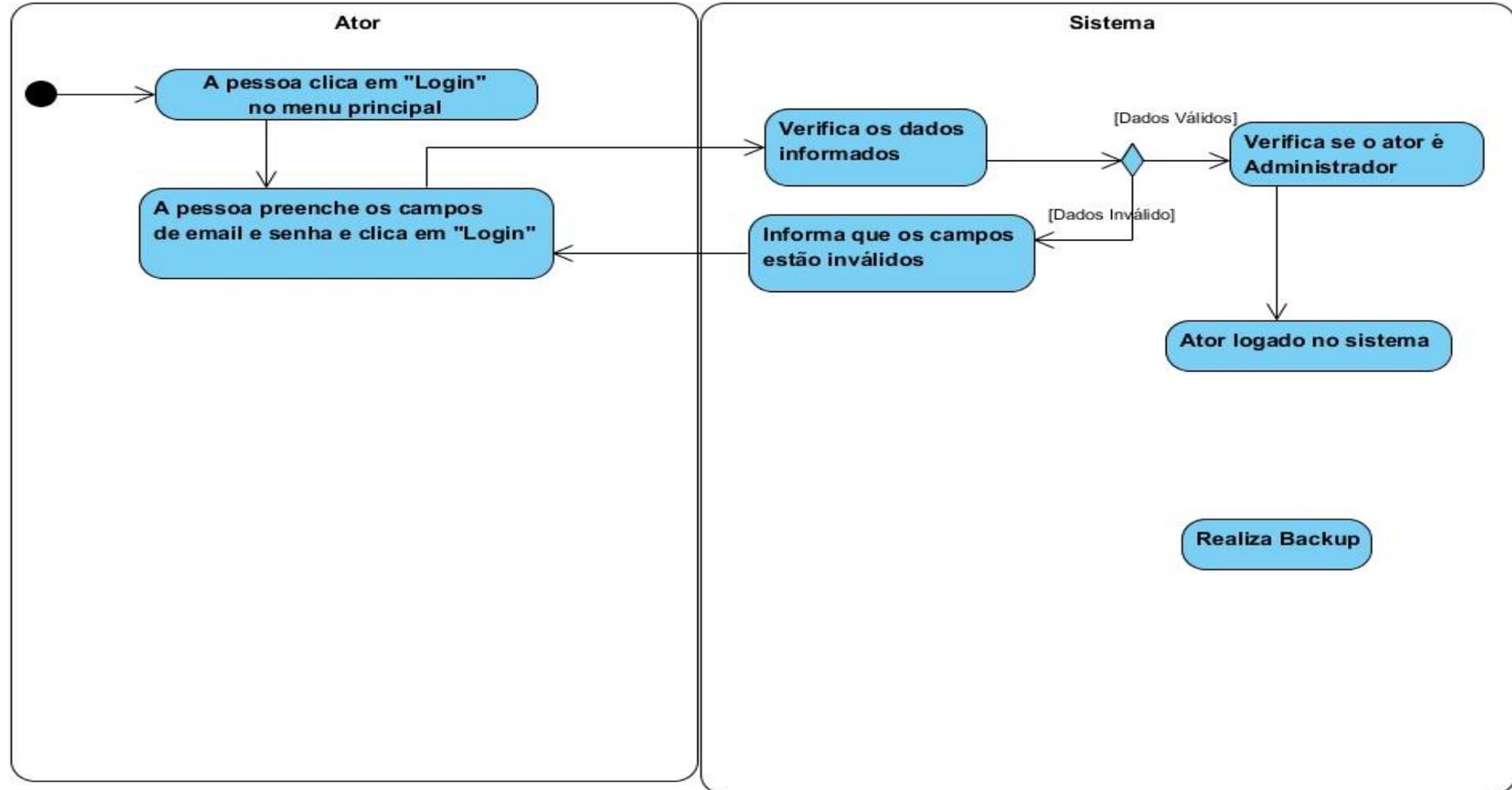


Figura 12 - Diagrama de Atividades Backup

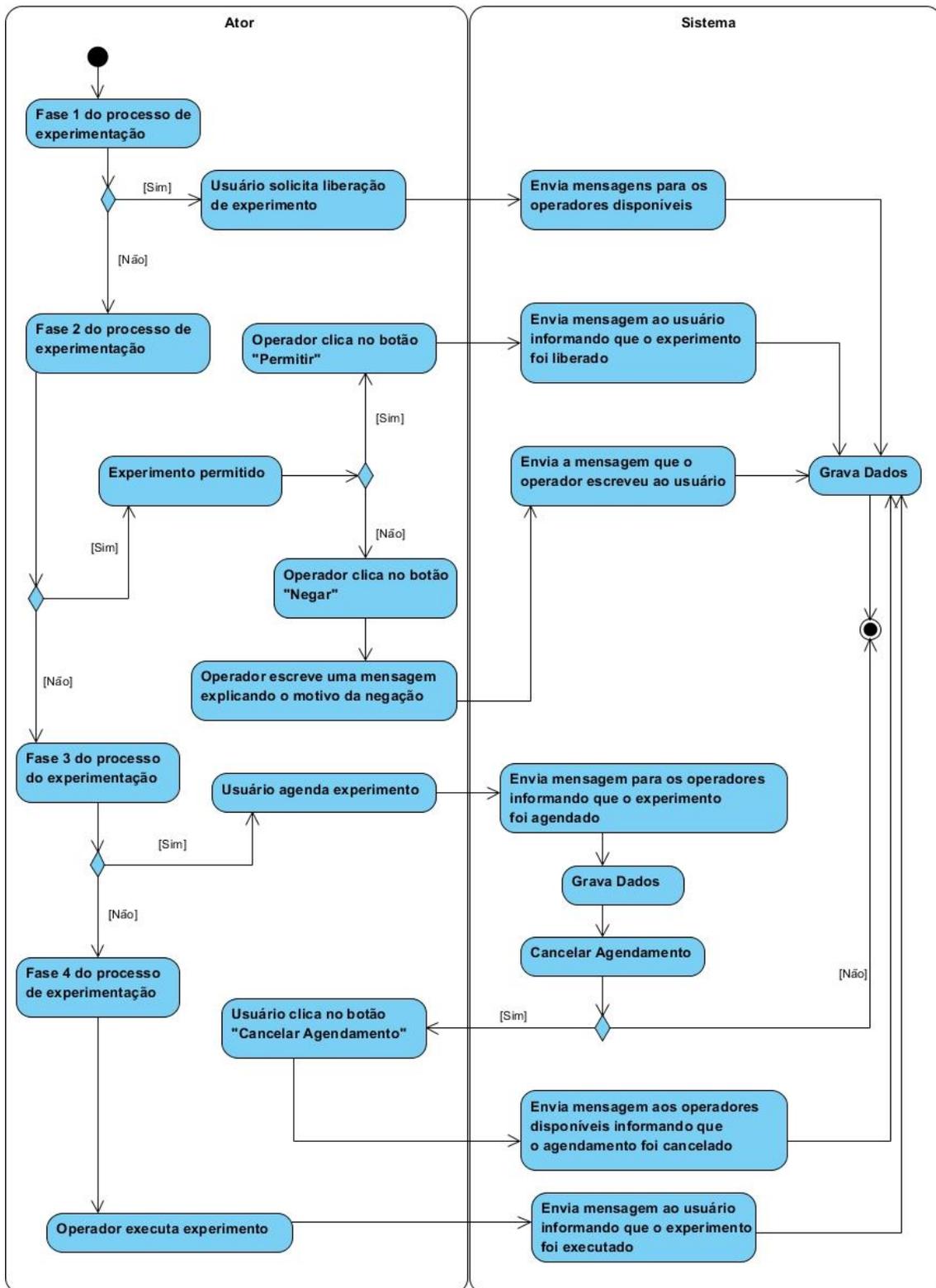


Figura 13- Diagrama de Atividades Protocolo de Mensagens

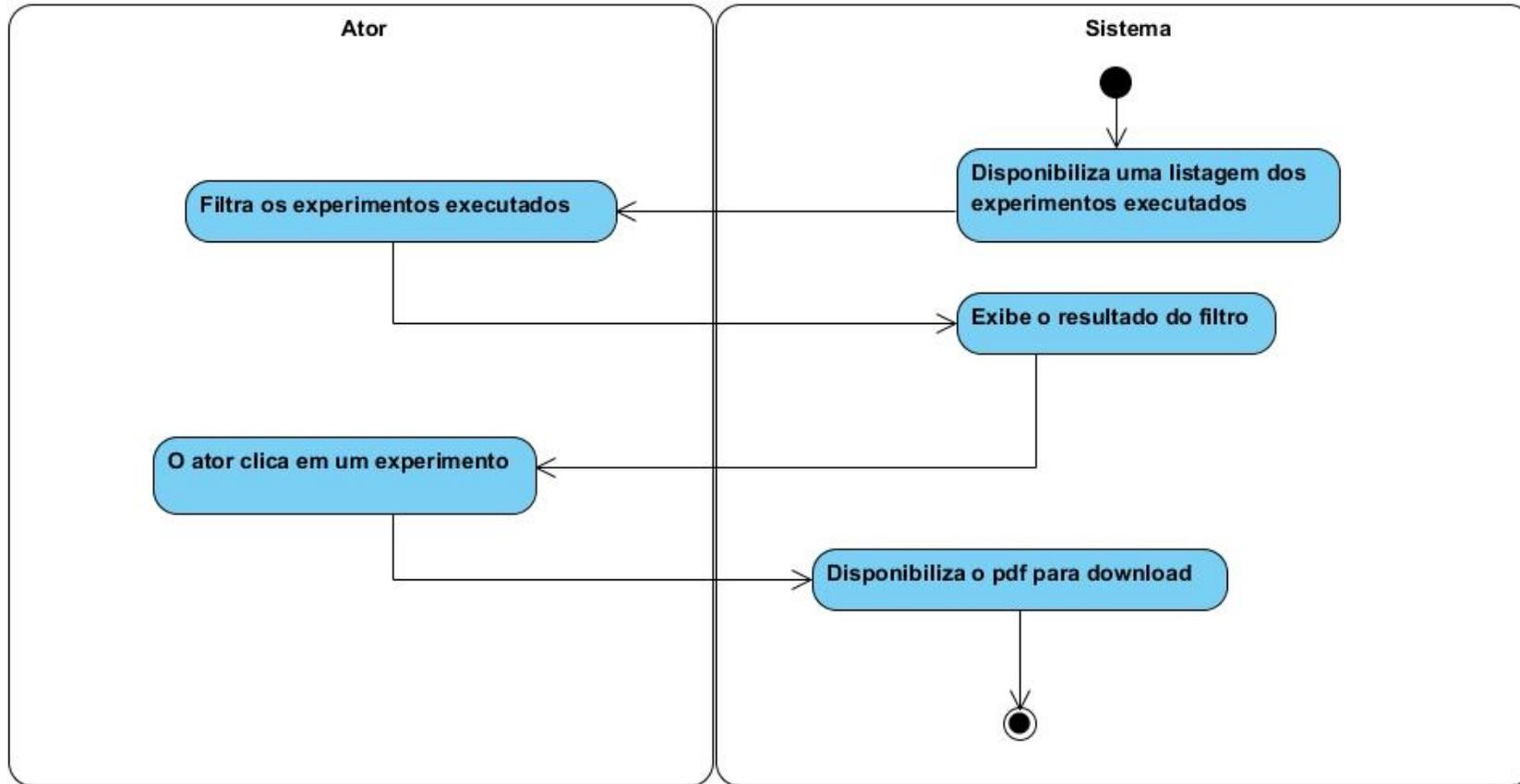


Figura 14 - Diagrama de Atividades Exportação de Dados

5.3.2 Diagramas de Seqüência

Nesta seção, são apresentados os diagramas de seqüência, que são artefatos da atividade de análise. Estes artefatos mostrarão a realização dos casos de uso do sistema desenvolvido neste trabalho em forma de troca de mensagens entre os objetos.

Um diagrama de seqüência procura determinar a seqüência de eventos para a execução de uma determinada tarefa, mostrando os métodos e as suas ações que devem ser disparadas e a seqüência na qual são executadas. O diagrama de seqüência tem como base o diagrama de Casos de Uso, modelando as ações disparadas por um ou mais ator para executar as tarefas de um caso de uso. (GUEDES, 2005).

A seguir serão apresentados os diagramas de seqüência dos módulos de Experimentação, Backup e Exportação de Dados.

Utilizando a página logadmin.xhtml, o administrador tem acesso ao log de todos os experimentos cadastrados no sistema conforme o diagrama da Figura 15.

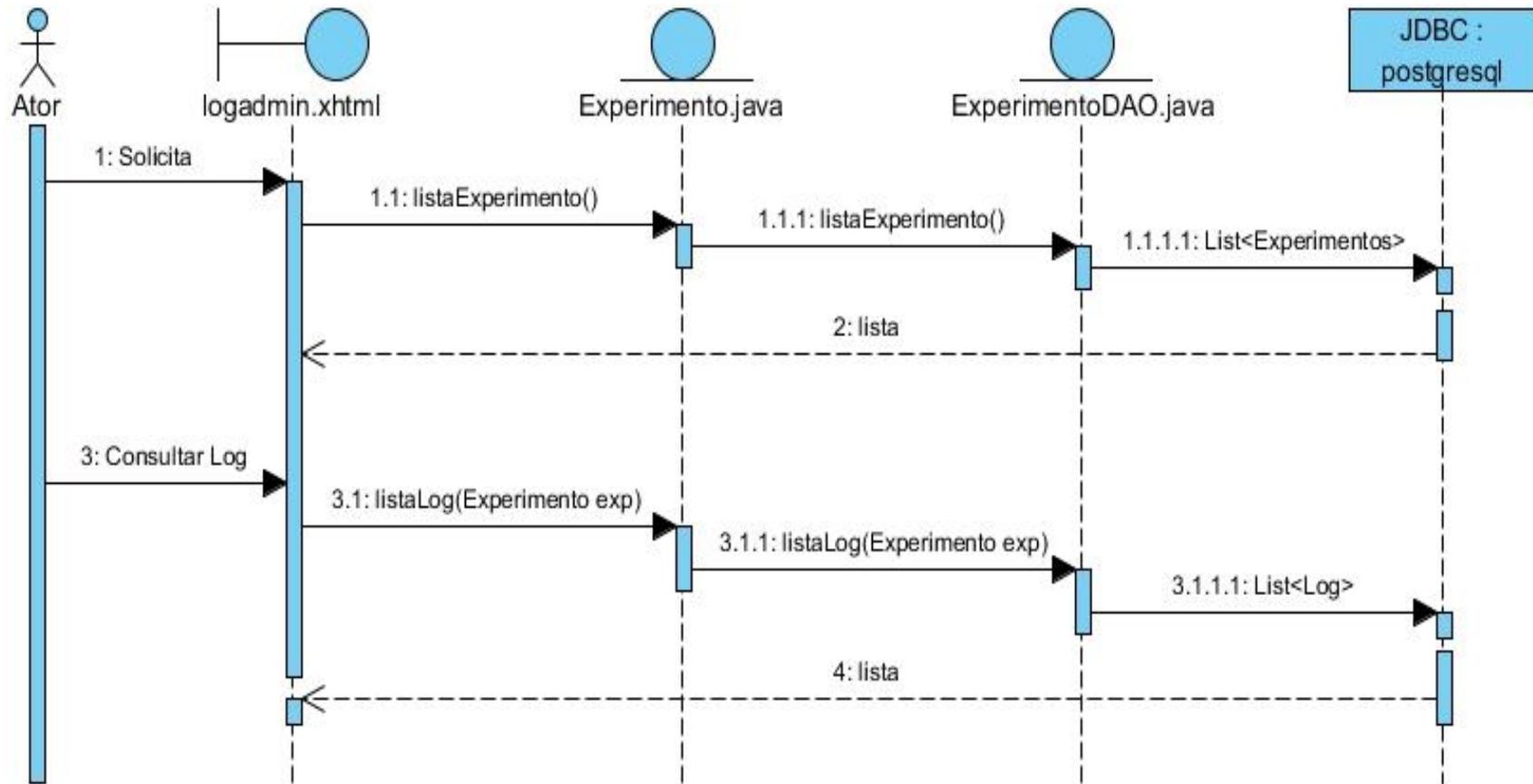


Figura 15 - Diagrama de Sequência Manter Log – Visão do Administrador

Utilizando a página `logoperador.xhtml`, o operador tem acesso ao log de todos os experimentos que solicitaram liberação para o agendamento e os experimentos que foram agendados conforme o diagrama da Figura 16.

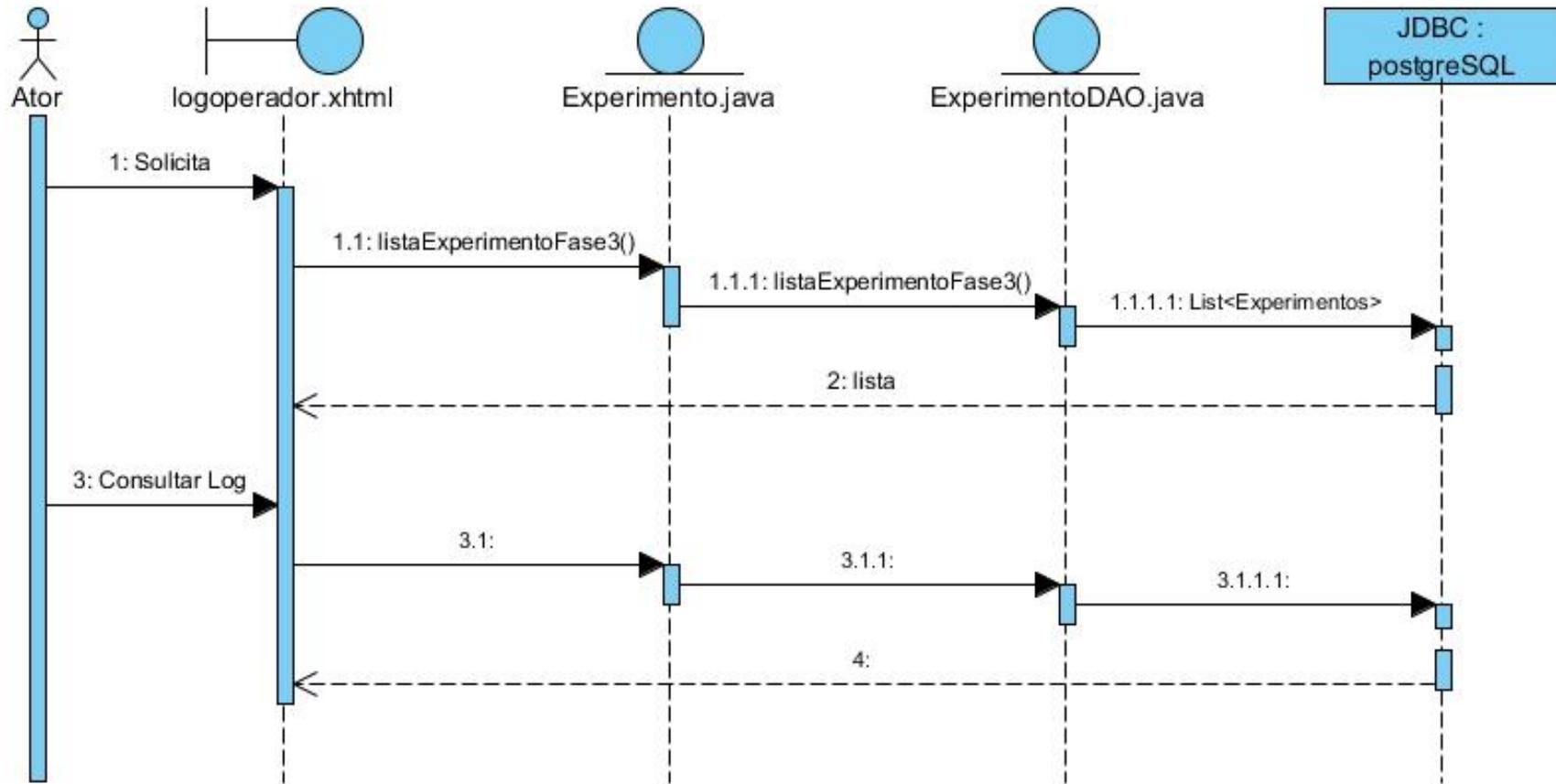


Figura 16 - Diagrama de Sequência Log do Operador

Utilizando a página `informacoedosexperimentos.xhtml`, o usuário pode ter acesso ao log de todos os experimentos relacionados a ele conforme o diagrama da Figura 17.

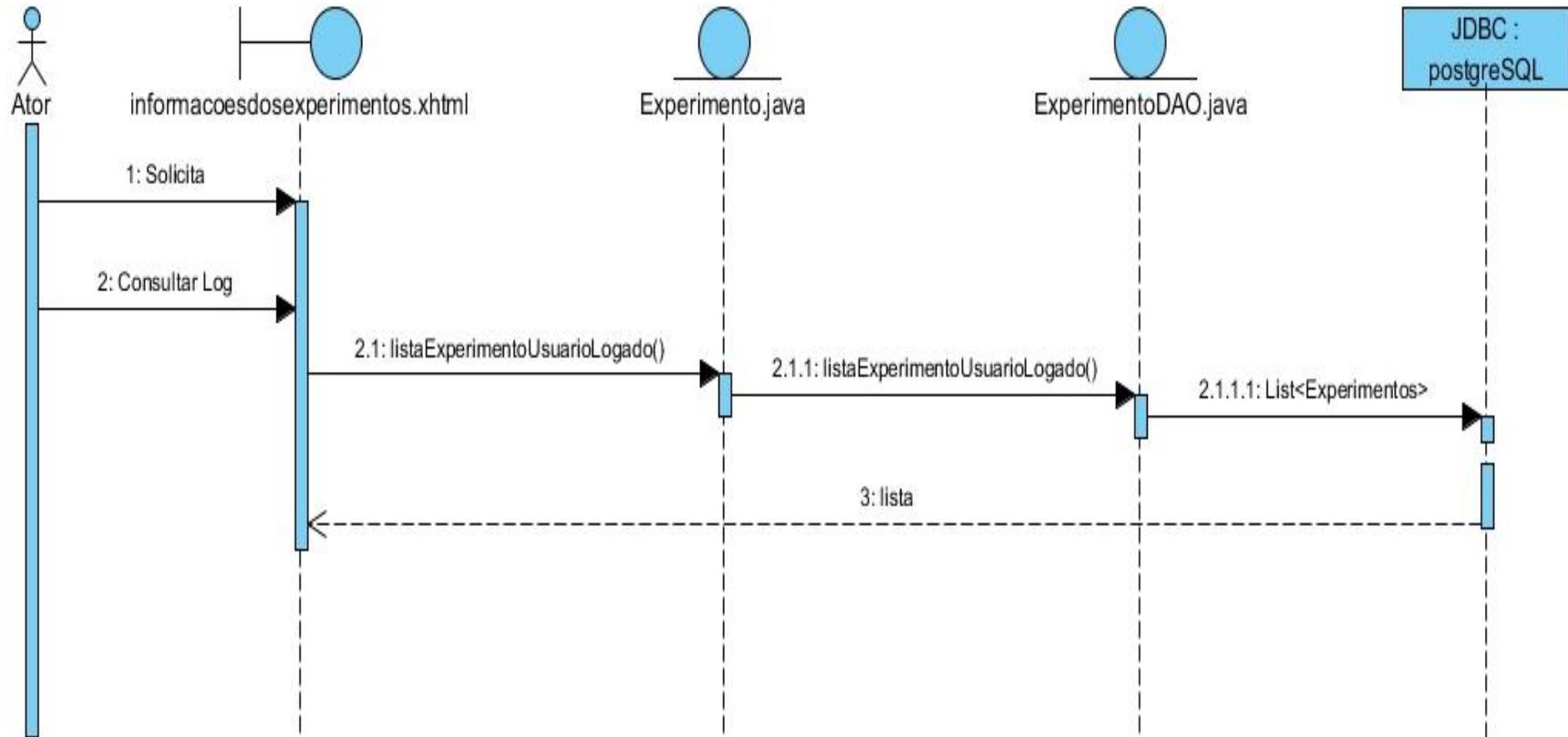


Figura 17 - Diagrama de Sequência Log do Usuário

Utilizando a página *experimento.xhtml*, o usuário pode inserir, alterar e excluir um experimento. Além disso, é possível visualizar os experimentos que já foram cadastrados pelo usuário conforme o diagrama da Figura 18.

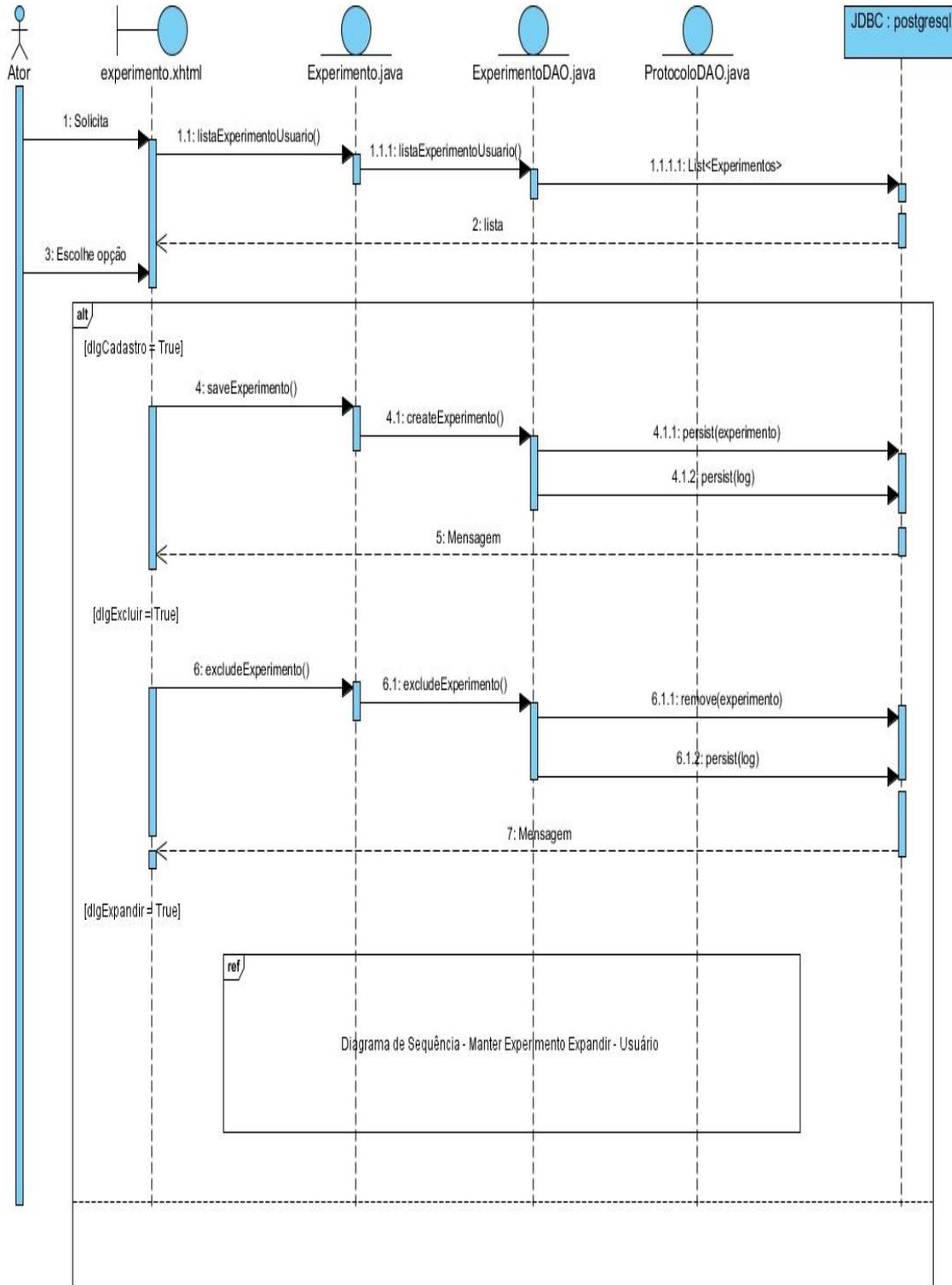


Figura 18 - Diagrama de Sequência Manter Experimento – (Visão Usuário)

Na página `experimento.xhtml`, conforme a Figura 18, é referenciado o Diagrama de Seqüência – Manter Experimento Expandir Usuário. Este diagrama é exibido conforme o diagrama da Figura 19.

O usuário também pode adicionar, editar e excluir um experimento não linear. Além disso, é possível visualizar os experimentos não lineares cadastrados.

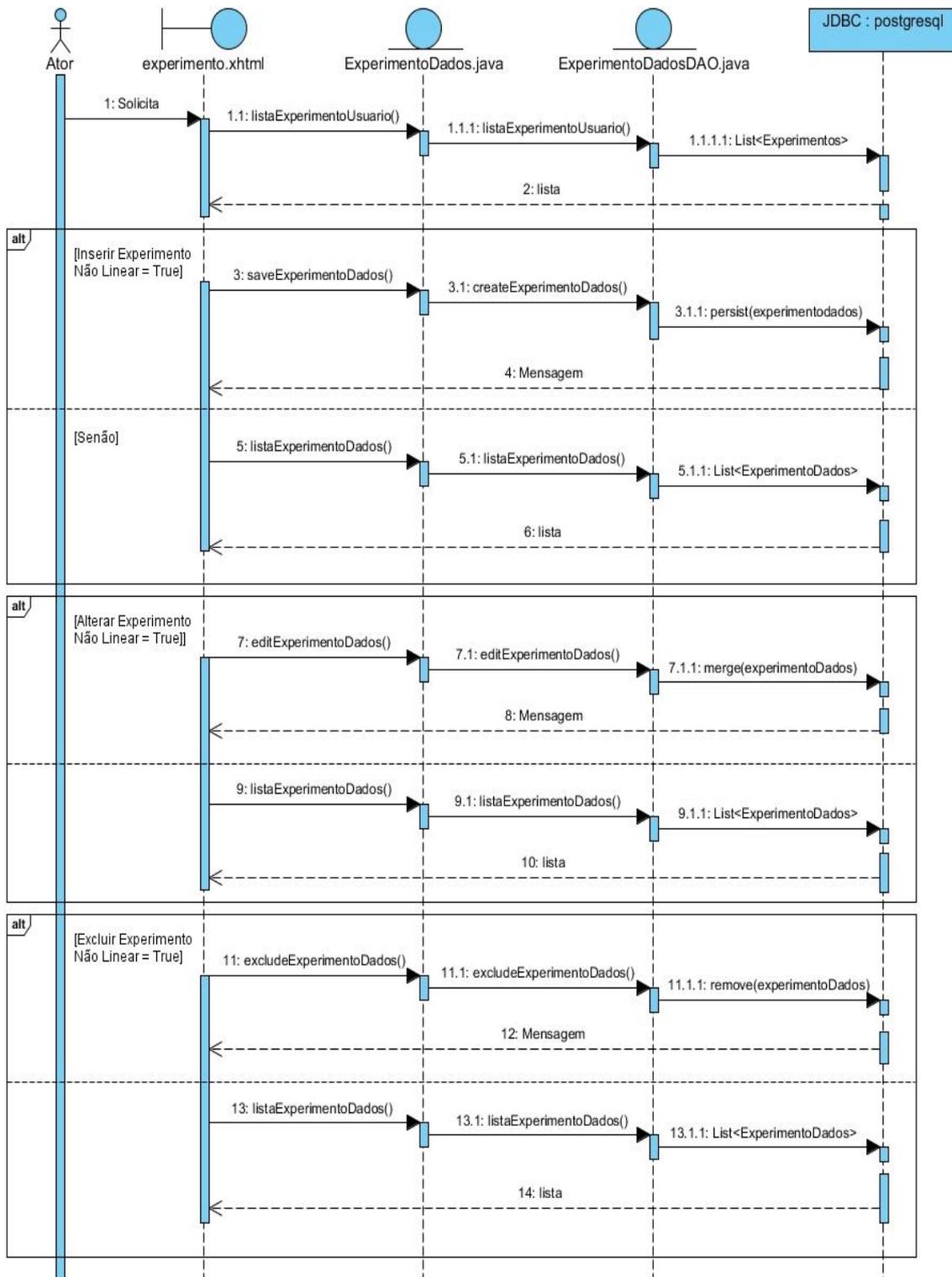


Figura 19 - Diagrama de Seqüência Manter Experimento Não Linear - Usuário.

Utilizando a página `liberarexperimento.xhtml`, o operador pode visualizar os dados do experimento e do experimento não linear. Se o experimento estiver dentro dos limites que o equipamento do laboratório suporta, o operador libera o experimento para prosseguir para a fase de Agendamento. Caso contrário, o operador nega a solicitação de agendamento e o experimento volta para fase anterior do processo de experimentação conforme a Figura 20.

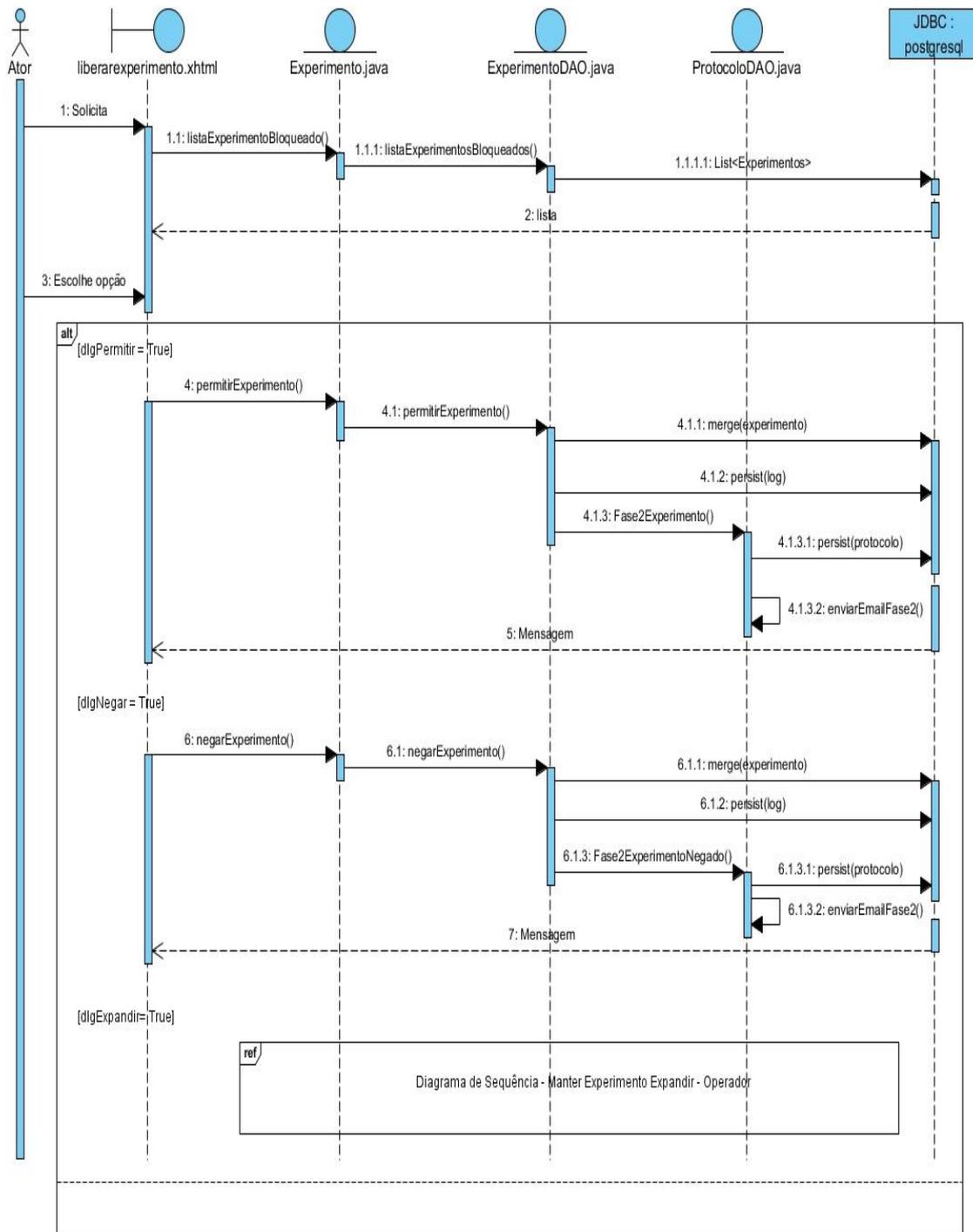


Figura 20 - Diagrama de Seqüência Manter Experimento – Operador

Na página `liberarexperimento.xhtml`, conforme a Figura 20, é referenciado o Diagrama de Seqüência – Manter Experimento Expandir Operador. Este diagrama é exibido conforme o diagrama da Figura 21.

O operador pode visualizar os dados do experimento e dos seus experimentos lineares antes de decidir permitir ou não a solicitação de liberação para o agendamento.

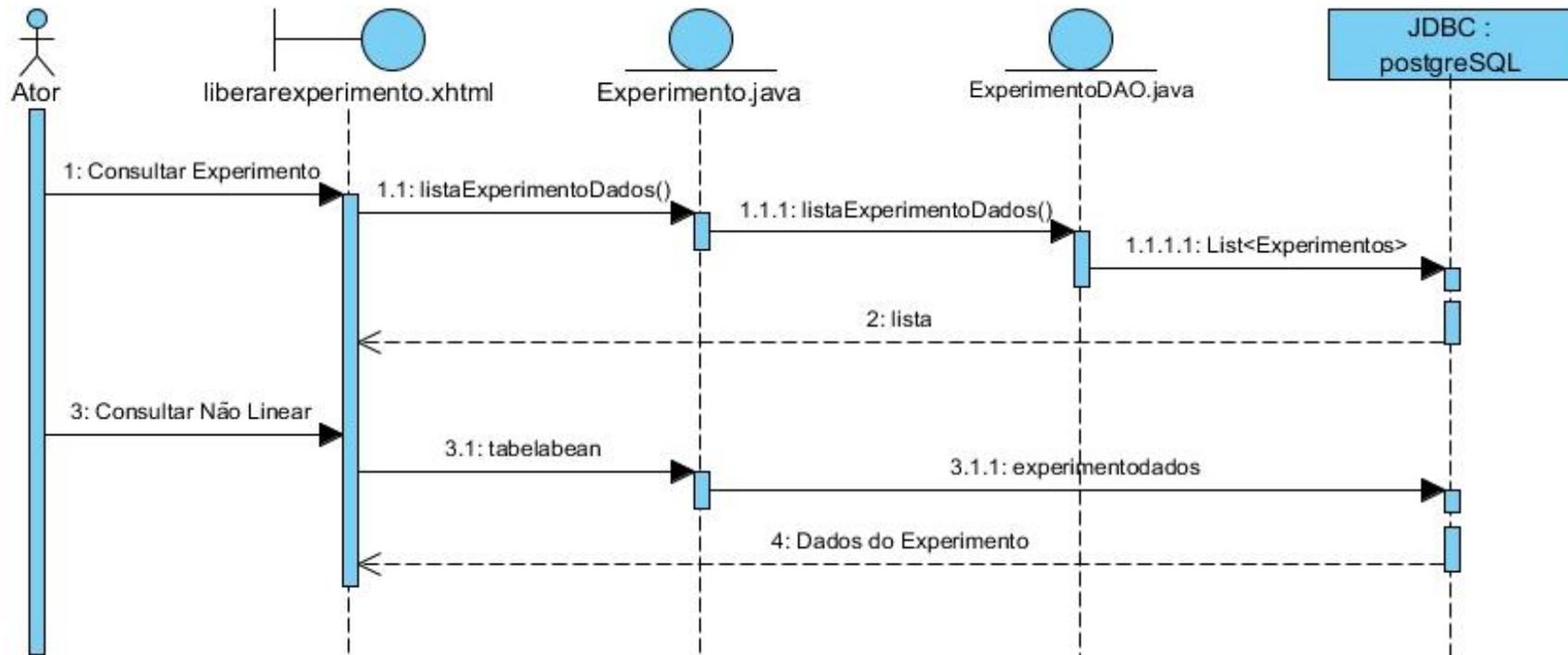


Figura 21 - Diagrama de Seqüência - Manter Experimento Expandir Operador

Utilizando a página `listaexperimentos.xhtml`, é possível realizar o download dos dados dos experimentos conforme a Figura 22.

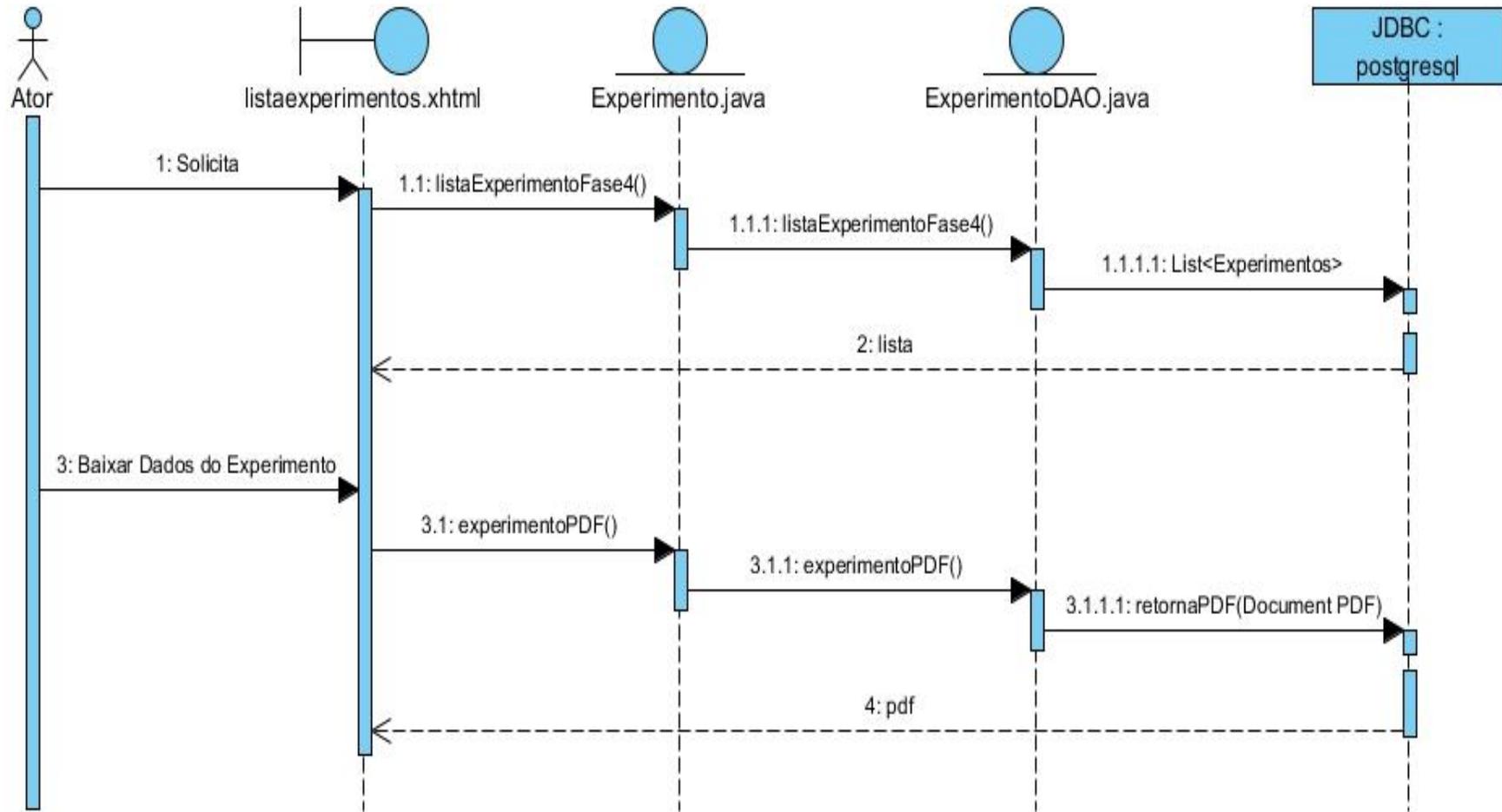


Figura 22 - Diagrama de Seqüência Exportação de Dados

5.3.3 Modelo Relacional

Um Modelo Relacional é um conjunto de conceitos que podem ser usados para descrever a estrutura de uma base de dados. Por estrutura de uma base de dados entendem-se os tipos de dados, relacionamentos e restrições pertinentes aos dados. (CRUZ T., 2013).

A Figura 23 apresenta o modelo relacional construído no desenvolvimento do projeto apresentado.

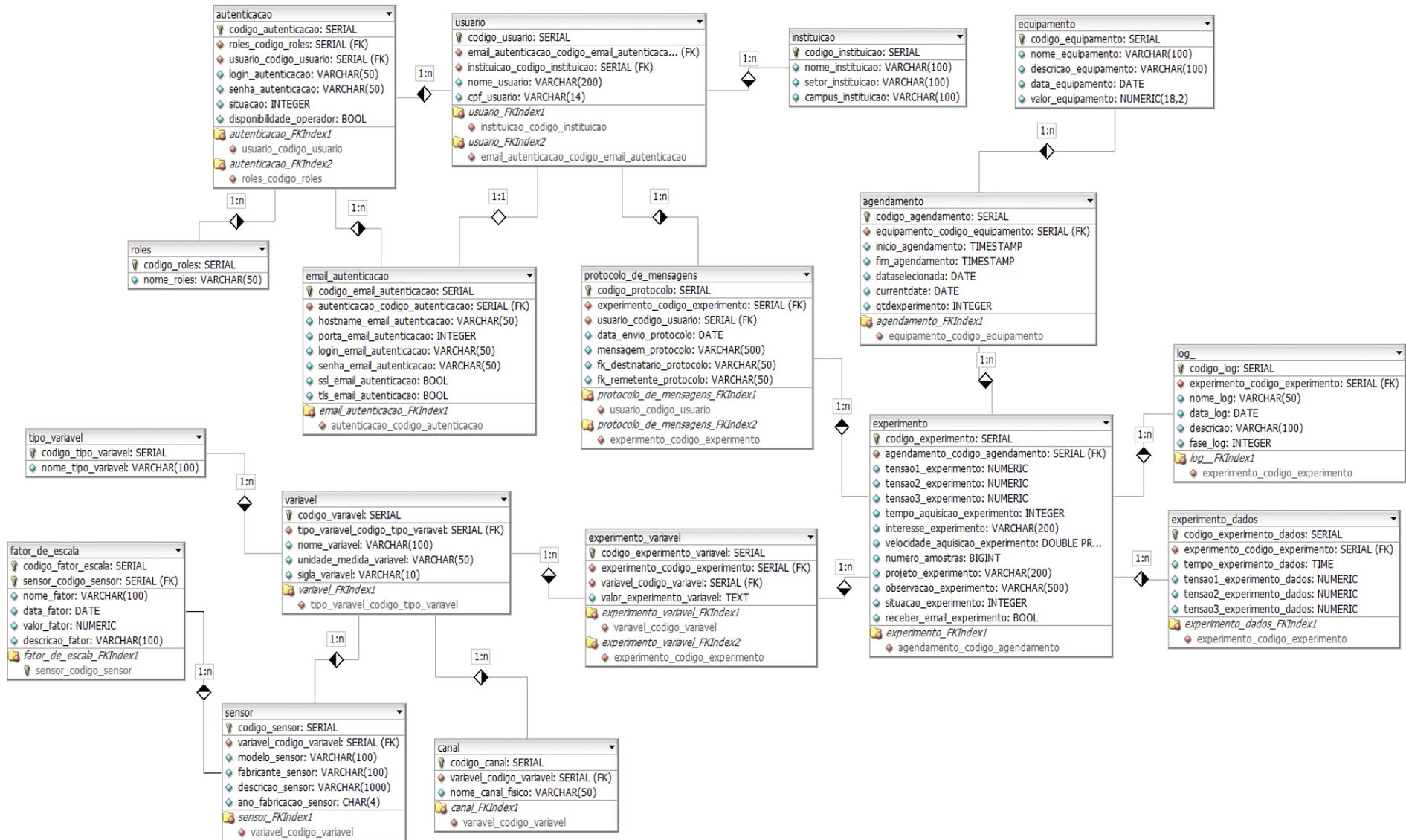


Figura 23 - Modelo Relacional

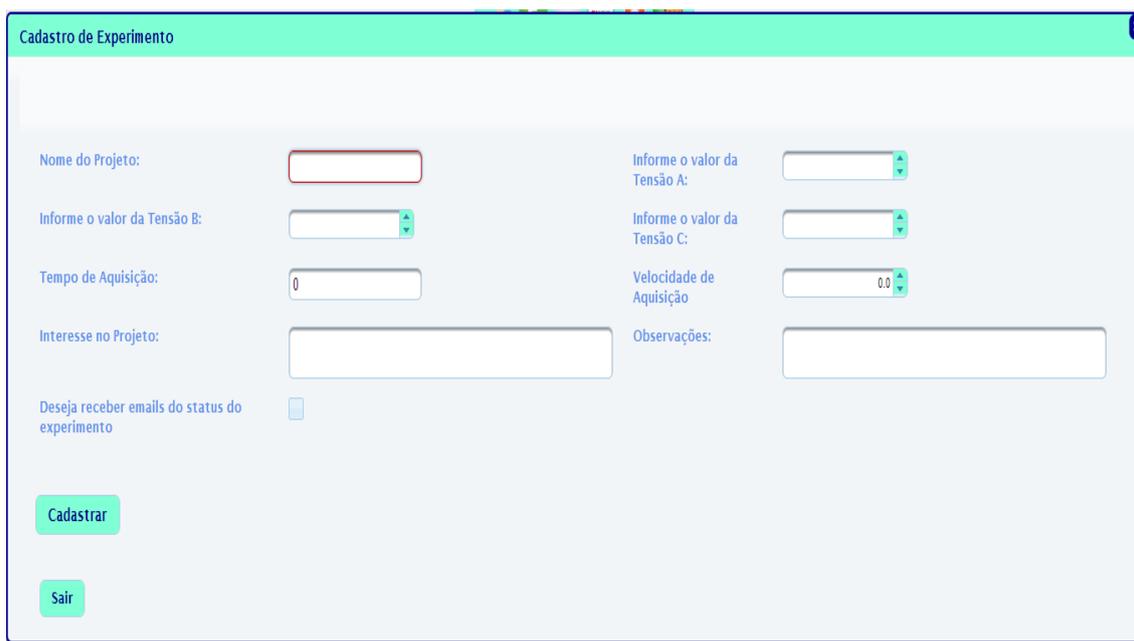
5.4 IMPLEMENTAÇÃO DO SUBSISTEMA DE EXPERIMENTAÇÃO

Nesta etapa foram implementados todos os casos de uso referentes ao subsistema de experimentação.

5.4.1 Manter Experimento

O processo de experimentação é dividido em cinco fases. Ao passar para a fase seguinte é realizado uma troca de mensagens entre o operador e o usuário.

Na Fase 0 do processo de experimentação o experimento é cadastrado pelo usuário conforme exibido na Figura 24. O formulário de cadastro possui a opção “Deseja receber emails do status do experimento”, ao selecionar esta opção é possível receber emails sobre a situação do experimento.



A imagem mostra uma interface web para o cadastro de um experimento. O formulário é dividido em duas colunas de campos de entrada. O lado esquerdo contém: 'Nome do Projeto:' (campo de texto), 'Informe o valor da Tensão B:' (campo de texto com seta para baixo), 'Tempo de Aquisição:' (campo de texto com o valor '0'), 'Interesse no Projeto:' (campo de texto), e 'Deseja receber emails do status do experimento' (checkbox). O lado direito contém: 'Informe o valor da Tensão A:' (campo de texto com seta para cima/baixo), 'Informe o valor da Tensão C:' (campo de texto com seta para cima/baixo), 'Velocidade de Aquisição' (campo de texto com o valor '0.0' e setas para cima/baixo), e 'Observações:' (campo de texto). Na base do formulário, há dois botões: 'Cadastrar' e 'Sair'.

Figura 24 – Fase 0 do Processo de Experimentação

Ao realizar o cadastro do experimento, a listagem de experimentos é atualizada conforme a Figura 25.



Figura 25 – Fase 0 do Processo de Experimentação

Nesta etapa é possível adicionar experimentos não lineares para o experimento ao clicar no botão “+” em verde. Ao clicar, é exibida a listagem de experimentos não lineares juntamente com o formulário de cadastro conforme a Figura 26.

O experimento não linear é utilizado quando é necessário alterar uma das 3 tensões em um determinado tempo na execução do experimento. Ele pode ser adicionado uma ou mais vezes.



Cadastro de Experimento

<< (1 of 1) >> 5

5

<< (1 of 1) >> 5

Tempo (em segundos): 10

Valor da Tensão A: 120

Valor da Tensão B: 127

Valor da Tensão C: 127

Adicionar

Sair

Figura 26 – Fase 0 Cadastro de Experimento Não Linear

Para prosseguir é necessário que o usuário adicione uma ou mais variáveis que deseja coletar ao final da execução do experimento ao experimento clicando no botão “caderneta com um +” conforme a figura 27.

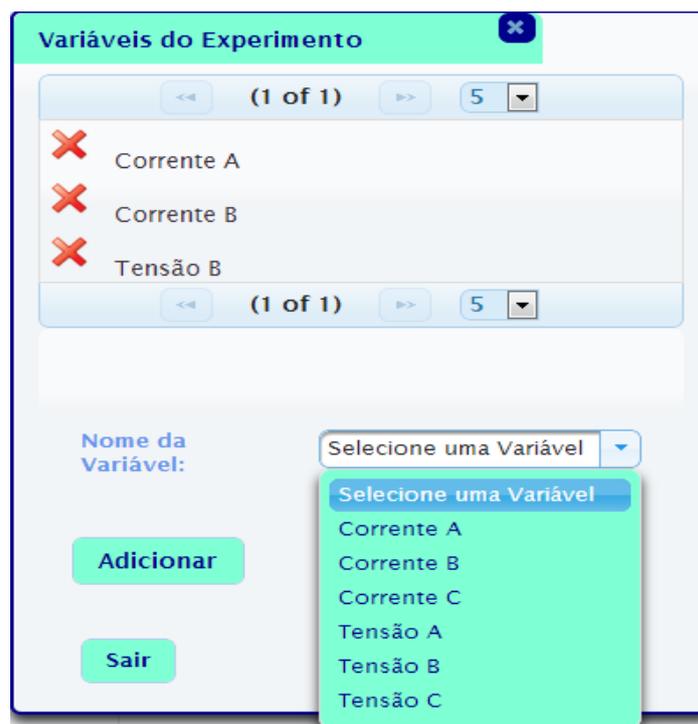


Figura 27 – Fase 0 Adicionar Variável

Na Fase 1 do processo de experimentação, é solicitado a liberação para o agendamento do experimento conforme a Figura 28. Após enviar a solicitação não é possível alterar ou excluir o experimento.

Figura 28 – Fase 1 do Processo de Experimentação

Na Fase 2 do processo de experimentação o operador verifica os dados do experimento e decide se permite ou nega a solicitação de liberação para a fase de Agendamento conforme a Figura 29.

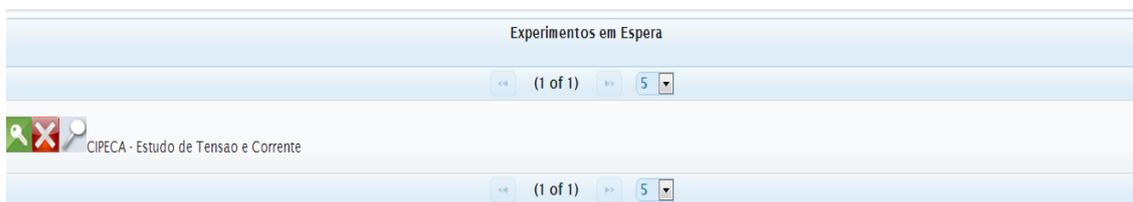


Figura 29 - Fase 2 do Processo de Experimentação

Caso o experimento seja negado, o operador irá escrever os motivos da negação da solicitação de liberação conforme a Figura 30.

O experimento irá retornar à Fase 1 para que o usuário possa corrigir as mudanças requeridas pelo operador.



The image shows a web form with a light blue header bar containing the text "Negar o experimento?" and a close button (X) in the top right corner. Below the header, the main content area has a light gray background. It features a bold heading "Descreva o motivo da negação do experimento" followed by a large, empty white text input field with a gray border and a scroll bar. At the bottom of the form, there are two buttons: "Enviar" and "Cancelar", both with a light blue background and dark text.

Figura 30 – Motivos da Negação de Liberação para o Agendamento

Na Fase 3 do processo de experimentação, o experimento já pode ser agendado se houver horários disponíveis para o agendamento conforme a Figura 31. Um experimento somente pode ser agendado com uma antecedência mínima de 24 horas.

Escolha o dia para verificar os horários disponíveis

Horário Disponível
 Experimento Agendado
 Dia Selecionado

Dia Atual
 Horário Não Disponível
 Final De Semana

<< < Outubro, 2013 > >>

Seg	Ter	Qua	Qui	Sex	Sáb	Dom
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Figura 31 - Fase 3 do Processo de Experimentação

Na Fase 4 do processo de experimentação, o experimento é executado pelo operador conforme a Figura 32.

Durante a execução do experimento, os dados são coletados através da aquisitora. Esses dados são gravados no banco de dados para serem utilizados na exportação do arquivo txt.



Figura 32 - Fase 4 do Processo de Experimentação

5.4.2 Protocolo de Mensagens

O protocolo de mensagens tem como intuito enviar e-mails informando a qual a fase do processo de experimentação em que o experimento se encontra. Os e-mails são enviados cada vez que o experimento avança uma fase.

As mensagens são trocadas entre os usuários e os operadores.

5.4.3 Log do Experimento

O log do experimento é armazenado para cada uma das fases do processo experimentação. Nele é exibido quando o experimento a situação do experimento em cada uma das fases.

O log é visualizado de maneiras diferentes para cada tipo de usuário. O operador irá visualizar apenas os logs de experimentos que requerem liberação para o agendamento e os que foram agendados conforme a Figura 33.

The image shows a user interface for viewing experiment logs. At the top, there is a legend with five phases, each represented by a colored circle:

- Fase 0 - Experimento cadastrado pelo usuário** (Black circle)
- Fase 1 - Usuário solicita liberação de experimento** (Red circle)
- Fase 2 - Operador libera experimento do usuário** (Orange circle)
- Fase 3 - Experimento agendado** (Blue circle)
- Fase 4 - Experimento executado** (Green circle)

Below the legend, the main area is titled "Logs dos Experimentos". It features a navigation bar with "<< (1 of 2) >>" and a dropdown menu set to "5". The logs are organized by project:

- Projeto: CIPECA - Estudo de Tensao e Corrente** (expanded):
 - Fase 3 (Blue circle):** Descrição: O experimento foi agendado com sucesso. Data: 15/10/2013
 - Fase 1 (Red circle):** Descrição: A solicitação de liberação foi enviada com sucesso. Data: 15/10/2013
- Projeto: Ensaio de Máquinas Elétricas** (collapsed)

Figura 33 - Log do Experimento - Visão do Operador

O usuário irá visualizar apenas os logs dos experimentos associados a ele conforme a Figura 34.

The screenshot displays the 'Log dos Experimentos' interface. At the top, there is a legend with five phases, each represented by a colored circle:

- Fase 0 - Experimento cadastrado pelo usuário** (Black circle)
- Fase 1 - Usuário solicita liberação de experimento** (Red circle)
- Fase 2 - Operador libera experimento do usuário** (Orange circle)
- Fase 3 - Experimento agendado** (Blue circle)
- Fase 4 - Experimento executado** (Green circle)

Below the legend, the main content area is titled 'Ultimos Experimentos agendados'. It features a navigation bar with '<<' and '>>' buttons, '(1 of 1)', and a dropdown menu showing '5'. A blue header bar indicates the project: 'Projeto: CIPECA - Estudo de Tensao e Corrente'. The log entries are as follows:

Ícone	Descrição	Data
●	O experimento foi cadastrado com sucesso	15/10/2013
●	A solicitação de liberação foi enviada com sucesso	15/10/2013
●	O Experimento foi liberado pelo operador	15/10/2013
●	O experimento foi agendado com sucesso	15/10/2013

Figura 34 - Log dos Experimentos - Visão do Usuário

O administrador irá visualizar os logs de todos os experimentos cadastrados conforme a Figura 35.

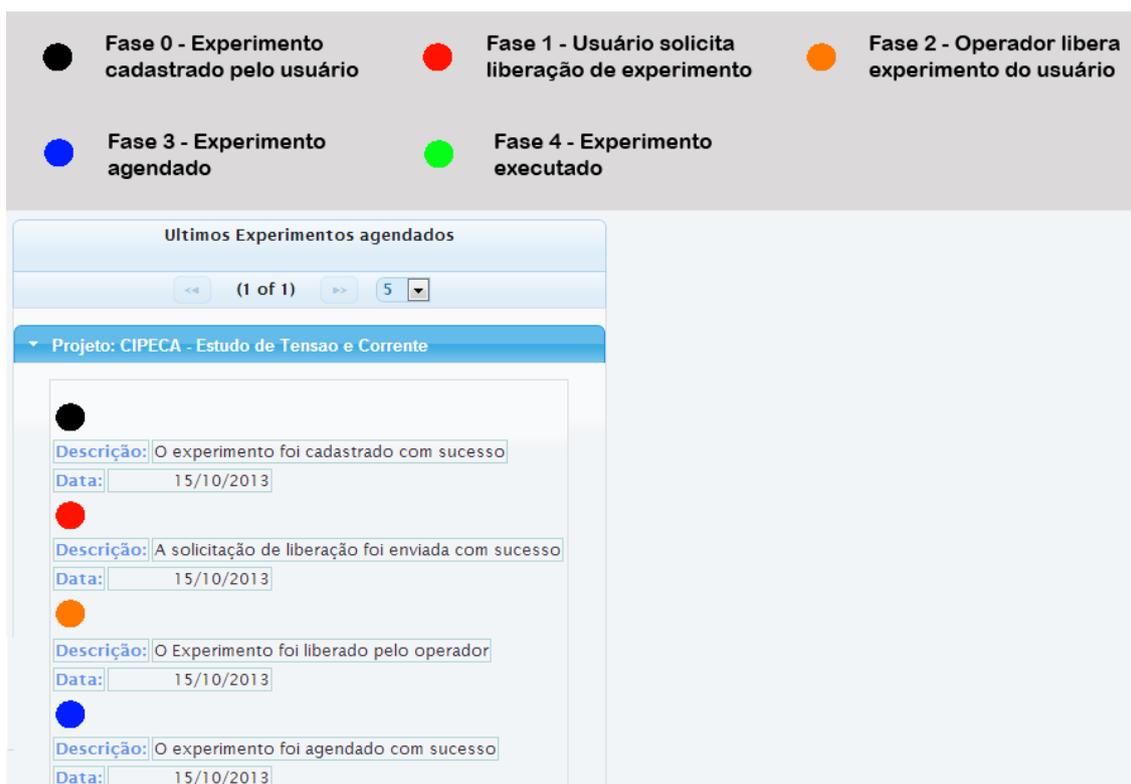


Figura 35 - Log dos Experimentos - Visão Administrador

5.5 IMPLEMENTAÇÃO DO SUBSISTEMA DE BACKUP

Nesta etapa foram implementados todos os casos de uso referentes ao subsistema de backup.

O Backup tem como função criar uma cópia de segurança de todos os dados cadastrados no sistema.

O sistema irá realizar o backup, em um horário com poucos usuários e que seja bem provável que nenhum experimento seja realizado durante este período. O backup será realizado diariamente.

No pacote backup foi necessário desenvolver duas classes uma com o nome de Temporizador e a outra de Backup.

Na classe Temporizador é chamado o método realizar backup a cada 24 horas, conforme a figura 36:

```

public class Temporizador implements ServletContextListener {
    public static final long TEMPO = (1000 * 60 *60 *24);
    public static void ExecutarBackup () {
        Timer timer = null;
        final Backup backup=new Backup();
        if (timer == null) {
            timer = new Timer();
            TimerTask tarefa = new TimerTask() {
                @Override
                public void run() {
                    try {
                        backup.realizaBackup();
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            };
            timer.scheduleAtFixedRate(tarefa, TEMPO, TEMPO);
        }
    }
}

```

Figura 36 – Classe Temporizador

O método realizar backup executa o arquivo back.bat através do prompt de comando, conforme a figura 37.

```

public class Backup {
    public void realizaBackup() throws IOException {
        Runtime r = Runtime.getRuntime();
        Process p = r.exec("cmd.exe /c back.bat");
        if (p != null) {
            OutputStream outputStream = p.getOutputStream();
            outputStream.flush();
            outputStream.close();
            InputStreamReader streamReader = new InputStreamReader(p.getErrorStream());
            BufferedReader reader = new BufferedReader(streamReader);
            String linha;
            while ((linha = reader.readLine()) != null) {
                System.out.println(linha);
            }
        }
    }
}

```

Figura 37 - Classe Backup

No arquivo back.bat é inserido o usuário e senha de um usuário existente no postgresSQL. Além de, configurar o nome, data e o horário do arquivo backup a ser gerado. Na figura 38 é exibido o arquivo back.bat.

```

set PGUSER=marcus
set PGPASSWORD=123456

for /f "tokens=1-4 delims=/ " %%a in ('date /T') do set year=%%c
for /f "tokens=1-4 delims=/ " %%a in ('date /T') do set month=%%a
for /f "tokens=1-4 delims=/ " %%a in ('date /T') do set day=%%b
set TODAY=%day%-%month%-%year%

for /f "tokens=1 delims=: " %%h in ('time /T') do set hour=%%h
for /f "tokens=2 delims=: " %%m in ('time /T') do set minutes=%%m
for /f "tokens=3 delims=: " %%a in ('time /T') do set ampm=%%a
set NOW=%hour%-%minutes%

C:\Users\Administrador\Documents\post\bin\pg_dump.exe -i -h localhost -p 5432 -U postgres -F c -b -o -v -f "C:\tcc\CIPECA%Today%\%NOW%.backup" tcc
pause

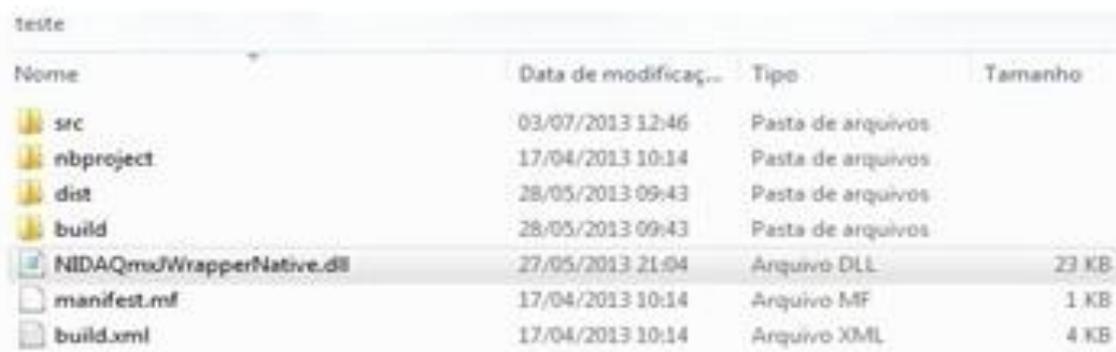
```

Figura 38 – Arquivo back.bat

5.6 IMPLEMENTAÇÃO DO SUBSISTEMA DE EXPORTAÇÃO DE DADOS

Nesta etapa foram implementados todos os casos de uso referentes ao subsistema de exportação de dados.

Ao executar um experimento é executado a função *ExecutarExperimento()*, essa função inicialmente verifica a listagem de variáveis requeridas e cria um vetor da classe *Tarefa* conforme a Figura 40. A classe *Tarefa* possui as variáveis e funções criadas pelo desenvolvedor do driver. É necessário inserir o arquivo *NIDAQmxJWrapperNative.dll* na pasta raiz do projeto conforme a Figura 39 para que ocorra a comunicação entre o sistema e a placa de aquisição de dados.



Nome	Data de modificaç...	Tipo	Tamanho
src	03/07/2013 12:46	Pasta de arquivos	
nibproject	17/04/2013 10:14	Pasta de arquivos	
dist	28/05/2013 09:43	Pasta de arquivos	
build	28/05/2013 09:43	Pasta de arquivos	
NIDAQmxJWrapperNative.dll	27/05/2013 21:04	Arquivo DLL	23 KB
manifest.mf	17/04/2013 10:14	Arquivo MF	1 KB
build.xml	17/04/2013 10:14	Arquivo XML	4 KB

Figura 39 - Adicionando arquivo dll na raiz do projeto

Também é necessário adicionar o arquivo *NIDAQmxJWrapperNative.jar* na pasta *Bibliotecas* do projeto.

Após realizar essas tarefas, já é possível utilizar o driver criado no projeto.

Além disso, a classe *Tarefa* é estendida da classe *Thread* para poder executar *Threads*.

```
Tarefa listaVariavel=new Tarefa().getTarefaSession();
Tabela tabela=new TabelaDAO().getTabelaSession();
Tarefa[] tarefas = new Tarefa[listaVariavel.getListaVariavel().size()];
ExperimentoVariavel[] vetorExperimentoVariavel=
    new ExperimentoVariavel
        [listaVariavel.getListaVariavel().size()];
int contador=0;
for (int i=0;i<tarefas.length;i++) {
    tarefas[i]=new Tarefa();
}
```

Figura 40 - Criando Vetor de Tarefa

Após isso é iniciado um laço de repetição que verifica o tipo da variável para que se possam realizar as configurações necessárias para o tipo da variável, conforme a Figura 41.

```
for (String nome: listaVariavel.getListaVariavel()) {
    ExperimentoVariavel experimentoVariavel=new ExperimentoVariavel();
    Task executar=new Task("Executar Experimento "+nome);
    List<Variavel> variavel=new VariavelDAO().buscaVariavel(nome);
    if(variavel.get(0).getFkCodigoTipoVariavel().
        getNomeTipoVariavel().equalsIgnoreCase("tensão")){
        executar=tarefas[contador].createCanalVoltagem
            (executar,variavel.get(0).getCanal());
    }else{
        if(variavel.get(0).getFkCodigoTipoVariavel().
            getNomeTipoVariavel().equalsIgnoreCase("corrente")){
            executar=tarefas[contador].createCurrentCanal
                (executar,variavel.get(0).getCanal());
        }else{
            if(variavel.get(0).getFkCodigoTipoVariavel().
                getNomeTipoVariavel().
                equalsIgnoreCase("velocidade")){
            executar=tarefas[contador].createVelocityCanal
                (executar,variavel.get(0).getCanal());
            }else{
                if(variavel.get(0).getFkCodigoTipoVariavel()
                    .getNomeTipoVariavel().
                    equalsIgnoreCase("torque")){
                    executar=tarefas[contador].
                        createTorqueCanal (executar,variavel.get(0).
                            getCanal());
                }
            }
        }
    }
}
```

Figura 41 - Verifica o tipo da variável

Em seguida, a Tarefa é iniciada e cria-se um leitor para realizar a coleta de dados da variável, conforme a Figura 42.

```

experimentoVariavel.setFkCodigoExperimento(tabela.getExperimento());
experimentoVariavel.setFkCodigoVariavel(variavel.get(0));
vetorExperimentoVariavel[contador]=experimentoVariavel;
executar=tarefas[contador].configuraRelogioDeAmostras(executar);
tarefas[contador].iniciarTarefa(executar);
AnalogChannelReader leitor=tarefas[contador].createLeitor(executar);
double[] dados=tarefas[contador].leituraDeDados(leitor);
tarefas[contador].setValor(dados);
++contador;

```

Figura 42 - Iniciando a Tarefa

Por fim, é iniciador o vetor de Tarefas e através da função join, cada Tarefa é executada sequencialmente conforme a Figura 43.

```

for (int i = 0; i < tarefas.length; i++) {
    tarefas[i].start();
    try {
        tarefas[i].join();
    } catch (InterruptedException e) {
    }
}

```

Figura 43 – Executando a Tarefa

Esse subsistema armazena os dados gerados pela execução do experimento no banco de dados. Além disso, os dados são inseridos em um arquivo txt e disponibilizados para download tanto para o usuário, operador e administrador conforme a figura 44. É possível uma filtragem de pesquisa para encontrar um determinado experimento através do nome do projeto, nome do usuário, pelo nome da Instituição, pelo nome do equipamento, e por um intervalo de data de cadastro do experimento.



Figura 44 - Listagem de Experimentos para Download

Após fazer o download do experimento no formato.txt, será disponibilizado os dados conforme a figura 45. Cada coluna terá os valores de uma variável, facilitando a leitura de importação desses dados para o software matLab, onde é realizado o estudo dos valores coletados.

Ensaio de Máquinas Elétricas - Bloco de notas		
Arquivo	Editar	Ajuda
2.1772360023432407E-6	-0.004457791390056906	-0.005856249814969945
5.339919446764153E-4	-0.005109234676674237	-0.0059830279364237095
0.0010690573203883343	-0.0058022878540141464	-0.005960272886858436
0.0012582481146888467	-0.006180021690696654	-0.005934267117069281
0.001288154563449629	-0.006475837530455875	-0.00589070745539077
0.0012237906846696968	-0.006920536828026312	-0.005723620426136108
0.001167228488369907	-0.007400344420483474	-0.005421303550386931
0.0010567046568853839	-0.007648050160396421	-0.004930445287695028
7.836457791832034E-4	-0.007575233760825743	-0.004370672275004196
3.5455324674584903E-4	-0.007394493105421919	-0.0040039919342873004
-2.12369058543407E-4	-0.0071740936251773555	-0.00368737270268281
-6.694177312343704E-4	-0.0074504056746156405	-0.003264780487215854
-0.0010250445689536482	-0.007510869274584143	-0.0027635213595426397
-0.0014222804756791867	-0.007510869274584143	-0.0024794096171490833
-0.0020457655168967474	-0.007445204505100403	-0.002408544228430938
-0.0024417014280997467	-0.0073288283526013066	-0.002459255240084987
-0.0025054152655145435	-0.007052516374948314	-0.0024930625824647226
-0.0024514535458353484	-0.0062859952581956705	-0.0025600271290846825
-0.0024690073579736837	-0.006000581832442224	-0.0028031799841647604
-0.002373436606309126	-0.005483717344309054	-0.0031802620684074894
-0.0022011492168388315	-0.0051105349633229595	-0.0037068769638631104
-0.001842271411291935	-0.004782862810929751	-0.004182130942514554
-0.0013039548768360005	-0.004303057446723858	-0.004511103092606754
-7.630379533108414E-4	-0.004016994780194515	-0.004805617816213076
-4.080612972863619E-4	-0.004310209014801413	-0.005316630431338403
-1.797711233491549E-5	-0.004531257517795734	-0.005737273450032894
5.0473563553401E-4	-0.00511248539330106	-0.005871203130364737
0.0010508533952269534	-0.005806838862570839	-0.005854299382556433
0.0012549974137403682	-0.006190424003141567	-0.0058243927530602715
0.0013018075074711765	-0.006458933763683904	-0.0057671800749440885
0.0012224904042930518	-0.006901682609967947	-0.0056000930817412
0.0011756803107953679	-0.00741984880456022	-0.005299726699515936
0.001054754236332517	-0.007642848988605884	-0.004819920962789365
7.719432557988728E-4	-0.007587586542063681	-0.004256897327181273
3.376496010359549E-4	-0.007430901289238466	-0.00390061931707071
-2.3057298719301534E-4	-0.0071695426045545	-0.0035703471452281487
-6.915225055422741E-4	-0.007473810938167354	-0.003156206827551211
-0.0010354468174323637	-0.007499816787968235	-0.0026529973317446012
-0.0014411345554150186	-0.007525822639257822	-0.002376687312015767
-0.0020646196049772312	-0.007451055820809206	-0.0023090726352059756
-0.002459255240084987	-0.007297621344624623	-0.002368885618335327
-0.0025067155479488856	-0.0069836009420124155	-0.002392940840687672
-0.0024625059460384132	-0.006598714928695068	-0.002456004534141022
-0.002464456369615014	-0.005980427359283204	-0.0026874548215084487
-0.0023480811021132737	-0.005531827981584467	-0.0030710382770667588
-0.0021634410368091515	-0.005087779947356023	-0.003600903816983579
-0.0018110646502409883	-0.004762058235349259	-0.00407290702010516
-0.001292902486168007	-0.004508502521656892	-0.004400578835689685
-7.519855657784947E-4	-0.00408721015190281	-0.004691842797518979
-4.0090975344430436E-4	-0.004329063149161371	-0.005210006899922903
8.028498403145284E-6	-0.004484447240827343	-0.005628049267908902
5.209891406261696E-4	-0.005159295714586198	-0.005745075178119881

Figura 45 - Arquivo .txt com os dados de um Experimento executado.

5.7 FASE DE TRANSIÇÃO

A fase de transição tem como objetivo entregar o sistema desenvolvido. Nesta fase é liberada uma versão beta para a realização de testes com o usuário. Com os resultados dos testes, verificam-se as pequenas falhas que passaram despercebidas na Fase de Construção.

Nesta fase também são disponibilizados materiais de treinamento que possibilitem aos usuários utilizar o sistema.

5.7.1 DIAGRAMA DE IMPLANTAÇÃO

O diagrama de implantação foi desenvolvido baseando-se na estrutura onde foi implantado o sistema. Possui os seguintes componentes:

- Banco de dados;*
- Servidor de aplicação;*
- Servidor;*
- Browser;*
- Controladora.*

A Figura 46 apresenta o diagrama de implantação.

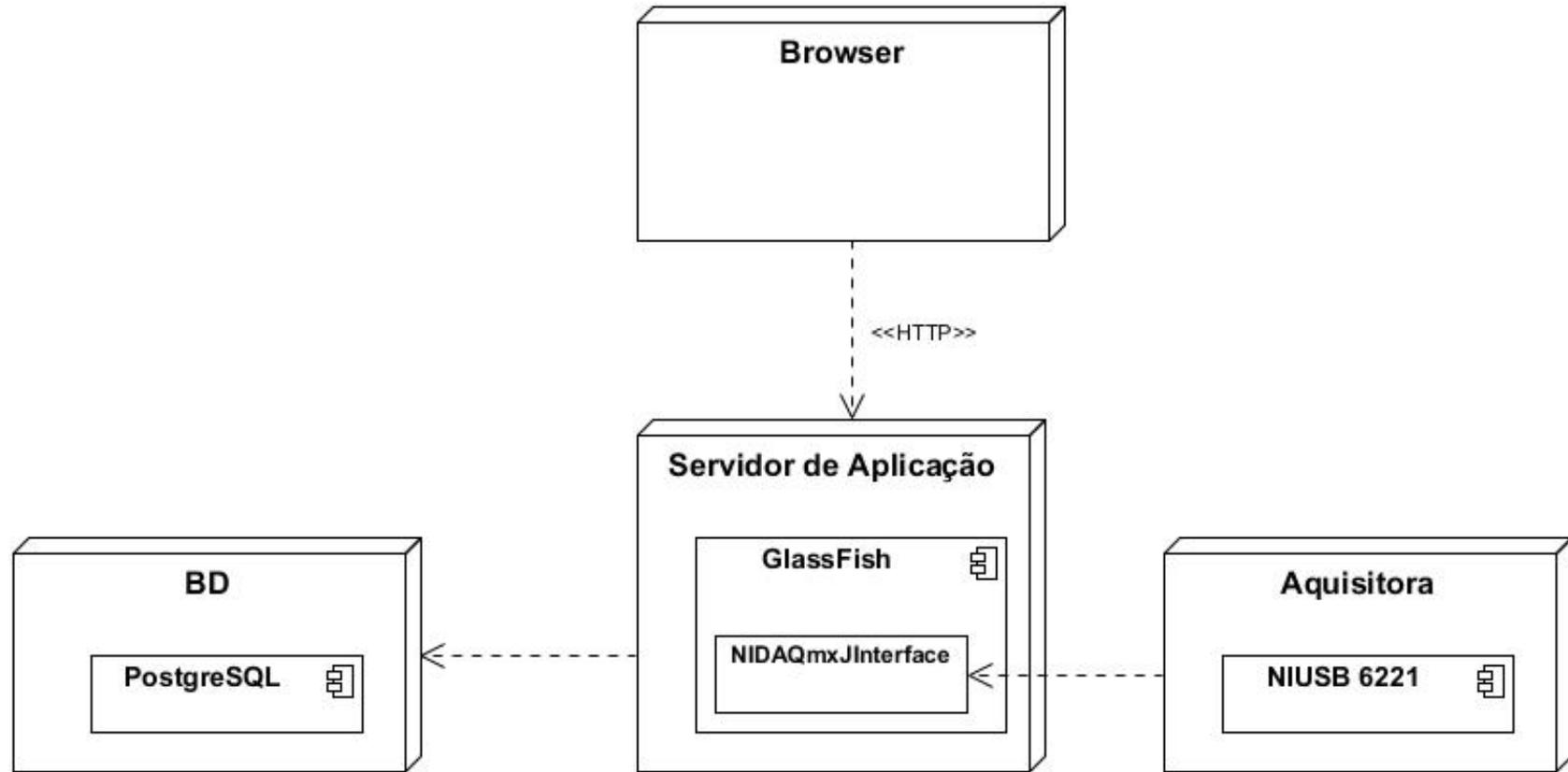


Figura 46 - Diagrama de Implantação

6. CONCLUSÃO

Durante o desenvolvimento do trabalho de diplomação foi possível estudar e aplicar na prática o processo de desenvolvimento de software. Além disso, também foi necessário o uso dos conceitos estudados durante o curso de graduação.

Na etapa de levantamento dos requisitos foi possível vivenciar o cenário da coleta de requisitos com o cliente que foi uma experiência útil para o desenvolvimento profissional.

Este trabalho necessitou de muito tempo para o entendimento das tecnologias e frameworks. Por meio de muita pesquisa e empenho, foi possível adquirir o conhecimento necessário para utilizá-las no sistema.

O sistema obtido ao final deste trabalho foi implementado no Laboratório de Controle e Automação do CIPECA – Campus Cornélio Procópio que buscava uma solução viável para a permissão de acesso do equipamento do laboratório para os usuários externos a instituição. Além disso, o desenvolvimento de uma base de dados para a consulta e armazenamento de experimentos trouxe melhorias aos utilizadores do equipamento.

7. TRABALHOS FUTUROS

Com o término do trabalho é possível buscar novas melhorias para o sistema que foi desenvolvido.

Após a implementação do sistema, foram surgindo idéias para melhorar o processo de experimentação. As idéias são:

- *Deixar em automático as ações do tipo de usuário operador:
Futuramente, as ações do operador serão executadas automaticamente.*
- *Programar uma web cam para o sistema:
A idéia é que o sistema possa automaticamente ligar uma web cam que estará instalada no laboratório. A web cam irá ser ligada durante a execução do experimento que o usuário agendou. Desse modo, o usuário poderá ter uma interação maior com o experimento do laboratório.*

8. REFERÊNCIAS

ALBRECHT, Allan J.; GAFFNEY, J. **Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation.** *IEEE Transactions on Software Engineering*. SE-9, 6, 1983.

BOEKEL, Rafael Van. **Por dentro do PrimeFaces 2.2.** Disponível em: <http://www.devmedia.com.br/por-dentro-do-primefaces-2-2-artigo-revista-java-magazine-93/21651> > Acessado em 10/08/2012.

CRUZ T. **Planeje seu modelo de Dados.** Disponível em: <http://www.linhadecodigo.com.br/artigo/332/planeje-o-seu-modelo-de-dados.aspx> > Acesso em: 14 de Junho de 2013.

Doutorado em Engenharia Elétrica (Conceito CAPES 6).
Universidade de São Paulo. **Estimador Neural de Velocidade Para Motores de Indução Trifásicos** (Qualificado em 16/05/2007), Ano de Obtenção: 2007.

GOEDEL, A. ; Graciola, Clayton Luiz ; SILVA, S. A. O. ; NASCIMENTO, Claudionor Francisco Do ; SUETAKE, Marcelo . **A Comparative Study for Single and Multilayer Neural Networks Applied to Speed Estimation in Induction Motors.** In: *XIX IEEE International Congress on Electrical Machines (ICEM 2010)*, 2010, Roma. *IEEE Proceedings of XIX International Congress on Electrical Machines (paper code PID1372065)*, 2010.

GUEDES, G. T. A. **UML 2 Guia de Consulta Rápida.** 2. Ed. São Paulo: Novatec, 2005.

RIBEIRO, Alberto. **Hibernate.** Disponível em: <http://javawora.blogspot.com.br/2007/08/hibernate.html> > Acessado em 10/09/2012.

JACOBI, Jonas. **Pro JSF e Ajax: Construindo Componentes Ricos Para a Internet.** São Paulo: Ciência Moderna, 2007.

JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James. **The Unified Software Development Process.** Addison-wesley, 1999.

MANZANO, José Augusto N.G. **PostgreSQL 8.3.0 Interativo: Guia de Orientação e Desenvolvimento.** 1. ed. São Paulo: Érica, 2008.

NetBeans. **NetBeans IDE - The Smarter Way to Code.** Disponível em: <http://netbeans.org/features/index.html> > Acessado em 18/05/2012.

PRESSMAN, Roger S. **Engenharia de software.** 6. ed. São Paulo: McGraw-Hill, 2006.

SUETAKE, Marcelo ; SILVA, Ivan Nunes da ; GOEDEL, A. . **Embedded DSP-Based Compact Fuzzy System and Its Application for Induction Motor V/f Speed Control (Qualis Engenharia IV).** *IEEE Transactions on Industrial Electronics (1982. Print)*, v. 58, p. 750-760, 2011.

SUETAKE, Marcelo ; SILVA, Ivan Nunes da ; GOEDTEL, A. . Sistema fuzzy compacto embarcado em DSP e sua aplicação para controle V/f de motores de indução (Qualis Engenharia IV). Controle & Automação (Online), v. 21, p. 245-259, 2010.

APÊNDICE A – IMPLANTAÇÃO DO SISTEMA

Para a implantação do sistema, foi necessária uma máquina como servidor, que foi disponibilizado pelo professor Gabriel Canhadas Genvigir, além disso, contendo o seguinte Hardware:

- *Processador Intel Pentium CORE I5 3,2 GHz;*
- *Memória RAM 4 MB;*
- *Disco Rígido 520GB.*

Foi definido e liberado pela instituição um número de ip, uma máscara de rede e um DNS fixo para que fosse possível o acesso interno e externo do servidor. As configurações de ip, máscara e DNS são apresentadas na figura 47 a seguir:

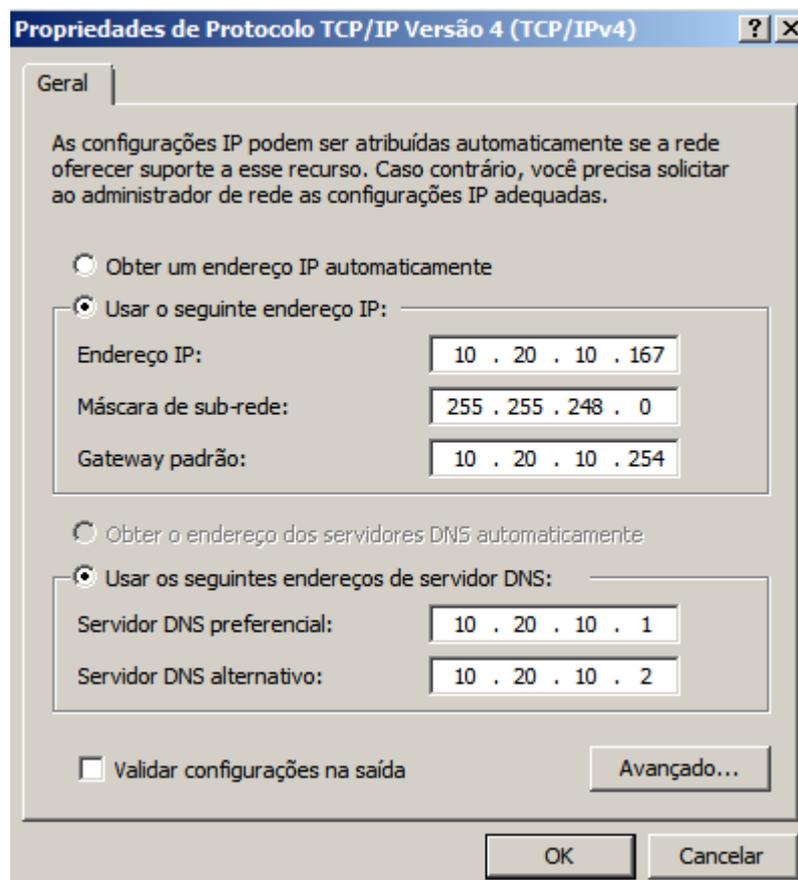


Figura 47 - Configuração do TCP/IPV4

Além dos recursos de hardware, foram usados recursos de software para a execução das tarefas, sendo listados a seguir:

- **NetBeans IDE 7.2:** onde está localizada a aplicação e seu código fonte, e para que o servidor possa ser iniciado;
- **GlassFish 3.1.1.2:** servidor da aplicação;

- **PostgreSQL 9.1:** *como sistema de gerenciamento de banco de dados relacional;*
- **NIDAQ 9.6.1:** *driver necessário para reconhecer a placa aquisitora de dados via USB.*
- **Java JDK 7 (32 bits):** *conjunto de utilitários que permitem criar sistemas de software para a plataforma Java. É composto por compilador e bibliotecas.*
- **Microsoft Windows Server 2008 R2 Standard (64 bits):** *sistema operacional utilizado para o servidor do sistema;*

No desenvolvimento do projeto foram necessárias algumas bibliotecas, que devem ser adicionadas na aplicação. As bibliotecas são:

- *commons-email-1.2;*
- *NIDAQmxJInterface;*
- *iText-2.1.7;*
- *cupertino-1.0.8;*
- *primefaces-3.4.1;*
- *richfaces;*

A figura 48 apresenta as bibliotecas que foram necessárias para o desenvolvimento do sistema.

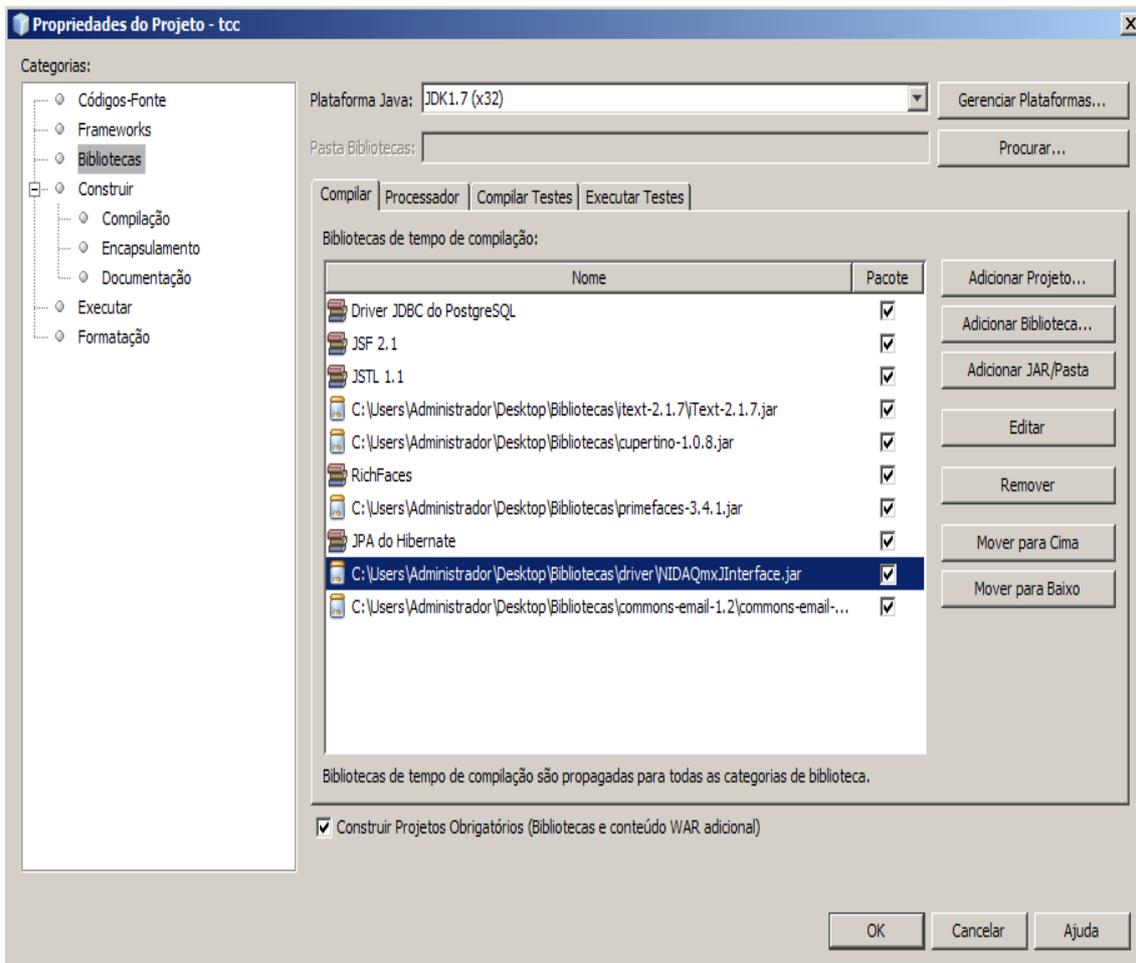


Figura 48 - Bibliotecas do Projeto

A interface utilizada para o PostgreSQL é o pgAdmin 3. Abaixo segue o script do banco de dados do sistema:

```
create table tipo_variavel(
```

- *codigo_tipo_variavel serial primary key,*
- *nome_tipo_variavel varchar(50) not null*

```
);
```

```
create table variavel(
```

- *codigo_variavel serial primary key,*
- *nome_variavel varchar(100) not null,*
- *unidade_medida_variavel varchar(50) not null,*
- *sigla_variavel varchar(10) not null,*
- *fk_codigo_tipo_variavel int not null,*

```

    • foreign key (fk_codigo_tipo_variavel) references tipo_variavel
      (codigo_tipo_variavel)
);
create table canal(
    • codigo_canal serial primary key,
    • nome_canal_fisico varchar(50) not null,
    • fk_codigo_variavel int unique not null,
    • foreign key (fk_codigo_variavel) references variavel(codigo_variavel)
);
create table sensor(
    • codigo_sensor serial primary key,
    • modelo_sensor varchar(100) not null,
    • fabricante_sensor varchar(100) not null,
    • descricao_sensor varchar(1000) not null,
    • ano_fabricacao_sensor char(4) not null,
    • fk_codigo_variavel int unique not null,
    • foreign key (fk_codigo_variavel) references variavel (codigo_variavel)
);
create table fator_de_escala(
    • codigo_fator_escala serial primary key,
    • fk_codigo_sensor int unique not null,
    • nome_fator varchar(100) not null,
    • data_fator date not null,
    • valor_fator decimal (18,2) not null,
    • descricao_fator varchar(1000) not null,
    • foreign key (fk_codigo_sensor) references sensor (codigo_sensor) on
      delete cascade
);
create table instituicao(
    • codigo_instituicao serial primary key,
    • nome_instituicao varchar(100) not null,
    • setor_instituicao varchar(100) not null,
    • campus_instituicao varchar(100) not null

```

);

create table roles(

- *codigo_rols serial primary key,*
- *nome_rols varchar(50) unique not null*

);

create table usuario(

- *codigo_usuario serial primary key,*
- *nome_usuario varchar(200) not null,*
- *cpf_usuario char(14) unique not null,*
- *fk_codigo_instituicao int not null,*
- *foreign key (fk_codigo_instituicao) references instituicao (codigo_instituicao)*

);

create table autenticacao(

- *codigo_autenticacao serial primary key,*
- *login_autenticacao varchar(50) unique not null,*
- *senha_autenticacao varchar(50) not null,*
- *situacao integer default 0,*
- *disponibilidade_operador boolean null,*
- *fk_codigo_rols int not null,*
- *fk_codigo_usuario int unique,*
- *foreign key (fk_codigo_rols) references roles (codigo_rols),*
- *foreign key (fk_codigo_usuario) references usuario(codigo_usuario) on delete cascade*

);

create table email_autenticacao(

- *codigo_email_autenticacao serial primary key,*
- *hostname_email_autenticacao varchar(50) null,*
- *porta_email_autenticacao int null,*
- *login_email_autenticacao varchar(50) unique null,*
- *senha_email_autenticacao varchar(50) null,*

- *ssl_email_autenticacao* boolean null,
- *tls_email_autenticacao* boolean null,
- *fk_codigo_autenticacao* int unique not null,
- foreign key (*fk_codigo_autenticacao*) references
autenticacao(codigo_autenticacao) on delete cascade

);

create table equipamento(

- *codigo Equipamento* serial primary key,
- *nome Equipamento* varchar(100) not null
- *descrição Equipamento* varchar(100) not null,
- *data Equipamento* numeric (18,2) not null,
- *valor Equipamento* numeric (18,2) not null

);

create table agendamento(

- *codigo agendamento* serial primary key,
- *inicio agendamento* timestamp,
- *fim agendamento* timestamp,
- *currentdate* date,
- *dataselecionada* date,
- *qtdexperimento* int,
- *fk_codigo Equipamento* int not null,
- foreign key (*fk_codigo Equipamento*) references
equipamento(codigo Equipamento)

);

create table experimento(

- *codigo experimento* serial primary key,
- *tensao1 experimento* decimal null,
- *tensao2 experimento* decimal null,
- *tensao3 experimento* decimal null,
- *tempo aquisicao experimento* int not null,
- *interesse experimento* varchar(200) not null,
- *velocidade aquisicao experimento* double precision not null,
- *numero amostras* bigint null,

- *projeto_experimento* varchar(200) not null,
- *observacao_experimento* varchar(500) null,
- *situacao_experimento* integer,
- *receber_email_experimento* boolean not null,
- *fk_codigo_agendamento* int null,
- foreign key (fk_codigo_agendamento) references
agendamento(codigo_agendamento)

);

create table *experimento_dados*(

- *codigo_experimento_dados* serial primary key,
- *fk_codigo_experimento* integer not null,
- *tempo_experimento_dados* time not null,
- *tensao1_experimento_dados* decimal null,
- *tensao2_experimento_dados* decimal null,
- *tensao3_experimento_dados* decimal null,
- foreign key (fk_codigo_experimento) references
experimento(codigo_experimento) on delete cascade

);

create table *experimento_variavel*(

- *codigo_experimento_variavel* serial primary key,
- *fk_codigo_variavel* int not null,
- *fk_codigo_experimento* int not null,
- *valor_experimento_variavel* text null, --ALTERADO
- foreign key (fk_codigo_variavel) references *variavel*(codigo_variavel),
- foreign key (fk_codigo_experimento) references
experimento(codigo_experimento) on delete cascade

);

create table *log*(

- *codigo_log* serial primary key,
- *nome_log* varchar(50) not null,
- *data_log* date not null,
- *descricao* varchar(100) not null,
- *fase_log* int not null,

- *fk_codigo_experimento int not null,*
- *foreign key (fk_codigo_experimento) references
experimento(codigo_experimento) on delete cascade*

);

create table protocolo_de_mensagens(

- *codigo_protocolo serial primary key,*
- *data_envio_protocolo date not null,*
- *mensagem_protocolo varchar(500) not null,*
- *fk_destinatario_protocolo varchar(50) not null,*
- *fk_remetente_protocolo varchar(50) not null,*
- *fk_codigo_experimento int not null,*
- *foreign key (fk_remetente_protocolo) references
autenticacao(login_autenticacao),*
- *foreign key (fk_destinatario_protocolo) references
autenticacao(login_autenticacao),*
- *foreign key (fk_codigo_experimento) references
experimento(codigo_experimento) on delete cascade*

);

insert into roles (nome_rol) values ('Administrador');

insert into roles (nome_rol) values ('Usuario');

insert into roles (nome_rol) values ('Operador');

*O sistema está implantado nas dependências da UTFPR, o servidor está localizado no CIPECA, a seguir a URL para o acesso:
<http://tesla.cp.utfpr.edu.br:8080/tcc/faces/index.xhtml>.*

APÊNDICE B - CASOS DE TESTES

CASO DE TESTE CADASTRO DE USUÁRIO

Passos:

- 1. Clicar no botão “Login” no menu acima.*
- 2. Clicar em “clique aqui” ao lado de [Não possui cadastro].*
- 3. Preencher o formulário de cadastro com valores válidos e clicar no botão “Cadastrar”.*

Resultado esperado:

- 1. A tela para inserir o login é exibida.*
- 2. A tela com o formulário de cadastro é exibida.*
- 3. Verificação de dados inválidos. Se os dados estiverem válidos deverá aparecer uma mensagem de cadastro realizado.*

CASO DE TESTE LOGIN.

Passos:

- 1. O usuário irá clicar no botão “Login” no menu acima.*
- 2. Preencher com login e senha válida e clicar no botão “Login”.*
- 3. O administrador irá realizar os passos de 1-2.*
- 4. Clicar no botão “Usuário” do menu acima.*
- 5. Clicar no botão que possui uma chave branca.*
- 6. Usuário realiza os passos de 1-3*

Resultado esperado:

- 1. A tela para inserir o login é exibida.*
- 2. Verificação dos dados de login. Se estiverem corretos, deverá aparecer uma mensagem avisando que o usuário ainda não tem permissão para utilizar o sistema.*
- 3. Exibição da página do administrador.*
- 4. Exibição da página com os usuários em espera.*
- 5. Mensagem avisando que o usuário possui acesso permitido.*
- 6. Exibição da página do usuário.*

CASO DE TESTE CADASTRAR EXPERIMENTO - USUÁRIO

Passos:

- 1. O usuário realizará o login.*
- 2. Clicar no botão “Experimentos” do menu acima.*
- 3. Clicar no botão “+”.*
- 4. Preencher o formulário de cadastro e clicar em “Cadastrar”.*
- 5. Clicar no botão da prancheta que possui um “+” verde.*
- 6. Escolher algumas variáveis e clicar em “Adicionar”.*

Resultado esperado:

- 1. Exibição da página do usuário.*
- 2. Exibição da página com os experimentos listados;*
- 3. Exibir o formulário de cadastro de um experimento.*
- 4. Verificação de dados inválidos. Se os dados estiverem válidos deverá aparecer uma mensagem de experimento cadastrado e o experimento estará listado. Após isso o sistema irá enviar um email ao operador.*
- 5. Exibir a tela com o formulário para a inclusão de variáveis para o experimento.*
- 6. Variáveis adicionadas.*

CASO DE TESTE SOLICITAR LIBERAÇÃO PARA O AGENDAMENTO-USUÁRIO

Passos:

- 1. O usuário realizará o login.*
- 2. Clicar no botão “Experimentos” do menu acima.*
- 3. Clicar no botão com a lupa.*
- 4. Clicar no botão “Solicitar Liberação”.*

Resultado esperado:

- 1. Exibição da página do usuário.*
- 2. Exibição da página com os experimentos listados;*
- 3. Exibir o formulário com os dados do experimento.*

4. *Verificação de dados inválidos. Se os dados estiverem válidos deverá aparecer uma mensagem que a solicitação foi enviada com sucesso e o experimento não estará listado. Após isso o sistema irá enviar um email ao operador.*

CASO DE TESTE LIBERAÇÃO EXPERIMENTO - OPERADOR

Passos:

1. *O operador irá verificar se o email foi enviado com sucesso.*
2. *O operador realizará o login.*
3. *Clicar no botão “Experimentos” do menu acima.*
4. *Clicar no botão com a “lupa”.*
5. *Clicar no botão “vermelho com x”.*
6. *Clicar no botão com a “chave verde”.*
7. *Clicar no “Ok”.*

Resultado esperado:

1. *Foi enviado um email alertando o operador sobre um novo experimento aguardando a liberação para o agendamento.*
2. *Exibição da página do operador.*
3. *Exibição da página com os experimentos aguardando a liberação para o agendamento.*
4. *Exibir os dados do experimento.*
5. *Exibi os dados do experimento não linear, se estiver vazio é porque não foi cadastrado.*
6. *Exibe uma janela de confirmação de liberação de agendamento.*
7. *Atualiza a lista de experimentos a serem liberados e envia um email para o usuário, avisando que o seu experimento já pode ser agendado.*

CASO DE TESTE AGENDAMENTO - ADMINISTRADOR

Passos:

1. *O administrador realizará o login.*
2. *Clicar no botão “Agendamento” no menu acima.*
3. *Selecionar um dia e clicar no botão com a lupa.*

4. *Preencher o formulário e clicar no botão “Cadastrar”.*
5. *Selecionar o dia em que cadastrou o horário e clicar no botão com a lupa*

Resultado esperado:

1. *Exibição da página do administrador.*
2. *Exibição da página com a agenda. A agenda do administrador deverá destacar em vermelho os dias que possuem horários para realizar o agendamento e deixar desabilitados os dias cuja data é inferior ao dia atual ou finais de semana.*
3. *Exibir a tela com os horários criados para aquele dia e possibilitar o cadastro de um novo horário*
4. *Verificação de horário inválido. Caso o horário seja válido, o dia deverá ficar destacado em vermelho e uma mensagem avisando que o horário foi criado é exibida.*
5. *Listagem do novo horário criado.*

CASO DE TESTE AGENDAMENTO - USUÁRIO

Passos:

1. *O usuário realizará o login.*
2. *Clicar no botão “Agendamento” no menu acima.*
3. *Selecionar um dia e clicar no botão com a lupa.*
4. *Clicar no botão com a lupa.*
5. *Clicar no botão com a agenda.*
6. *Selecionar o dia com o experimento agendado e clicar no botão com a lupa.*

Resultado esperado:

1. *Exibição da página do usuário.*
2. *Exibição da página com a agenda. A agenda do usuário deverá destacar em preto os dias que possuem horários para realizar o agendamento, destacar em vermelho os dias em que o usuário agendou um experimento e deixar desabilitados os dias cuja data é inferior ao dia atual ou finais de semana.*
3. *Exibir a tela com os horários criados para aquele dia e os experimentos agendados.*
4. *Exibir a tela para realizar o agendamento.*

5. *Validação do horário. Se estiver válido. O dia deverá ficar destacado em vermelho e uma mensagem avisando que o experimento foi agendado.*

6. *Exibir a tela listando o experimento agendado.*

CASO DE TESTE EXECUÇÃO DO EXPERIMENTO - OPERADOR

Passos:

1. *O operador realizará o login.*

2. *Clicar no botão “Agendamento” no menu acima.*

3. *Selecionar um dia e clicar no botão com a lupa.*

4. *Selecionar um horário e clicar no botão com a lupa.*

5. *Escolher um experimento e configurar o equipamento conforme solicitado no experimento.*

6. *Clicar no botão “Start”.*

7. *Clicar no botão “Ok”.*

Resultado esperado:

1. *Exibição da página do operador.*

2. *Exibição da página com a agenda. A agenda do operador deverá destacar em vermelho os dias que possuem experimentos agendados e deixar desabilitados os dias cuja data é inferior ao dia atual ou finais de semana.*

3. *Exibir a tela com os horários criados para aquele dia que possuem experimentos agendados.*

4. *Exibir a tela com os experimentos agendados para o horário.*

5. *Começar a coleta de dados do equipamento e ao final aparecer uma mensagem para gravar os dados no banco.*

6. *Dados do experimento gravados no banco de dados.*

CASO DE TESTE DOWNLOAD DOS DADOS DO EXPERIMENTO

Passos:

1. *Clicar no botão “Download” no menu acima.*

2. *Clicar em um experimento listado.*

3. *Clicar em download*

4. *Abrir o arquivo.*

Resultado esperado:

- 1. Exibição da página com os experimentos executados.*
- 2. Exibição da janela para o download do arquivo com os dados do experimento.*
- 3. Arquivo baixado pelo usuário.*
- 4. Arquivo com os dados do experimento.*

APÊNDICE C – PROPOSTA DO TRABALHO DE DIPLOMAÇÃO

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE INFORMÁTICA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

HALUAN HAYAMA DA SILVA

SISTEMA GERENCIADOR DE EXPERIMENTOS PARA O MOTOR DE
INDUÇÃO TRIFÁSICA DO LABORATÓRIO DE CONTROLE E
AUTOMAÇÃO - MÓDULO EXPERIMENTAÇÃO

PROPOSTA DE TRABALHO DE DIPLOMAÇÃO

CORNÉLIO PROCÓPIO

2012

HALUAN HAYAMA DA SILVA

**SISTEMA GERENCIADOR DE EXPERIMENTOS PARA O MOTOR DE
INDUÇÃO TRIFÁSICA DO LABORATÓRIO DE CONTROLE E
AUTOMAÇÃO - MÓDULO EXPERIMENTAÇÃO**

Proposta de trabalho de conclusão de curso de graduação do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Coordenação do curso de Tecnologia em Análise e Desenvolvimento de Sistemas – COADS – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Elias Canhadas Genvigir

CORNÉLIO PROCÓPIO

2012

RESUMO

DA SILVA, Haluan Hayama. Sistema gerenciador de experimentos para o motor de indução trifásica do Laboratório de Controle e Automação – Módulo Experimentação. 2012. 20 f. Trabalho de Diplomação (Tecnologia em Análise e Desenvolvimento de Sistemas), Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2012.

Esta proposta descreve um projeto cujo objetivo é o desenvolvimento de um sistema web para o gerenciamento dos experimentos de um motor de indução trifásica. O sistema irá permitir que usuários externos ao laboratório, devidamente cadastrados no sistema, possam realizar experimentos utilizando o equipamento do Laboratório de Controle de Automação. Além de fornecer uma base de dados para o armazenamento dos experimentos realizados. Sendo assim, o desenvolvimento do projeto irá trazer ao processo de experimentação do laboratório critérios como: armazenamento, confiabilidade, extensibilidade e usabilidade.

Palavras-chave: *Gerenciamento de experimentos. Motor de indução trifásica. Usuários externos. Armazenamento. Laboratório de Controle de Automação.*

ABSTRACT

DA SILVA, Haluan Hayama. System manager experiments for three-phase induction motor Laboratory Automation and Control – Experimentation Module. 2012. 20 f. Trabalho de Diplomação (Tecnologia em Análise e Desenvolvimento de Sistemas), Federal Technology University - Paraná. Cornélio Procópio, 2012.

This proposal describes a project whose goal is the development of a web system for the managing experiments for a three-phase induction motor. The system will allow users outside the laboratory, duly registered in the system, can perform experiments using the equipment of the Laboratory and Automation Control. In addition to providing a database for the storage of experiments performed. Thus, the development of the project will bring to the process of experimentation laboratory criteria as: storage, reliability, extensibility and usability.

Keywords: *Management experiments. Three-phase induction motor. External users. Storage. Laboratory Automation Control.*

LISTA DE ILUSTRAÇÕES

<i>Figura 1 - Diagrama de Casos de uso.....</i>	<i>20</i>
<i>Figura 2 - Diagrama de Pacotes.....</i>	<i>23</i>
<i>Figura 3 - Diagrama de Subsistemas.....</i>	<i>25</i>
<i>Figura 4 - Módulo Experimentação.....</i>	<i>28</i>
<i>Figura 5 - Módulo Exportar Dados.....</i>	<i>29</i>
<i>Figura 6 – UP.....</i>	<i>30</i>

LISTA DE QUADROS

Quadro 1 - Cronograma do Trabalho de Diplomação 34

LISTA DE TABELAS

<i>Tabela 1 - Requisitos funcionais (contínua).....</i>	<i>16</i>
<i>Tabela 2 - Respectivas funções dos atores no sistema</i>	<i>19</i>
<i>Tabela 3- Relação de Requisitos com Casos de uso - Módulo Experimentação.....</i>	<i>27</i>
<i>Tabela 4 - Relação de Requisitos com Casos de uso - Módulo Exportar dados.....</i>	<i>28</i>

LISTA DE SIGLAS

<i>APF</i>	<i>Análise de Pontos por Função</i>
<i>CPF</i>	<i>Cadastro de Pessoas Físicas</i>
<i>IDE</i>	<i>Integrated Development Environment</i>
<i>JDBC</i>	<i>Java Database Connectivity</i>
<i>JSF</i>	<i>Java Server Faces</i>
<i>RG</i>	<i>Registro Geral</i>
<i>SGBDOR</i>	<i>Sistema de Gerenciamento de Banco de Dados Relacional</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>USB</i>	<i>Universal Serial Bus</i>
<i>VT</i>	<i>Virtual Terminal</i>

LISTA DE ACRÔNIMOS

<i>AJAX</i>	<i>Asynchronous Javascript And XML</i>
<i>CIPECA</i>	<i>Centro Integrado de Pesquisa em Controle e Automação</i>
<i>MATLAB</i>	<i>MATrix LABoratory</i>
<i>MIT</i>	<i>Motor de Indução Trifásica</i>
<i>UI</i>	<i>User Interface</i>

SUMÁRIO

1. INTRODUÇÃO	11
1.1 OBJETIVOS GERAIS.....	11
2. COMPREENDER O DOMÍNIO DA APLICAÇÃO	13
2.1 Descrição do Domínio do Negócio	13
2.2 Descrição das Características do Negócio.....	13
2.3 COMPREENDER O PROBLEMA A SER RESOLVIDO	14
2.4 Descrição das Complexidades (Restrições Do Sistema).....	15
3. COMPREENDER AS NECESSIDADES DOS USUÁRIOS	16
3.1 REQUISITOS FUNCIONAIS.....	16
3.2 Requisitos não Funcionais	18
3.3 Tarefa - Identificar Atores e Funções.....	19
3.4 Definir Casos de Uso	19
4. Tecnologias	21
5. Arquitetura Inicial.....	23
5.1 Diagrama de Pacotes	23
5.2 Diagrama de Subsistemas	24
6. METODOLOGIA.....	30
6.1 Processo Unificado	30
7. ANÁLISE DE PONTOS POR FUNÇÃO	33
7.1 Cálculo dos pontos por função	33
8. CRONOGRAMA.....	34
9. Referências	35

1. INTRODUÇÃO

Esta proposta descreve um projeto cujo objetivo é o desenvolvimento de um sistema Web para o gerenciamento de um MIT (Motor de Indução Trifásica) que está situado no Laboratório de Controle e Automação.

O sistema proposto é composto por vários módulos que serão desenvolvidos por dois alunos do curso, ou seja, o sistema terá seu desenvolvimento dividido em partes.

É responsabilidade desta proposta o desenvolvimento dos módulos de: Experimentação, Exportação de Dados e Backup. Os módulos de Administração e Agendamento serão desenvolvidos pelo aluno Marcus Vinicius Sanches Sant' Anna.

Este trabalho está dividido nos seguintes capítulos que são apresentados da seguinte maneira:

- *Capítulo 2 - Compreender o Domínio da Aplicação;*
- *Capítulo 3 - Compreender as Necessidades dos Usuários;*
- *Capítulo 4 - Tecnologias;*
- *Capítulo 5 - Arquitetura Inicial;*
- *Capítulo 6 - Metodologia;*
- *Capítulo 7 - Pontos por Função;*
- *Capítulo 8 - Cronograma;*
- *Capítulo 9 - Referências.*

1.1 OBJETIVOS GERAIS

O sistema a ser desenvolvido possui como objetivos:

- 1) *Permitir o acesso remoto para que usuários externos realizem experimentações.*
- 2) *Possuir uma base de dados que armazene usuários, instituições, variáveis, sensores e experimentos.*
- 3) *Possuir um protocolo de mensagens entre operador e usuário.*
- 4) *Disponibilizar uma agenda para gerenciar os experimentos dos usuários.*

- 5) *Possuir uma base de dados que permita o acesso rápido para a consulta dos dados armazenados.*
- 6) *O sistema deve contemplar itens de confiabilidade e usabilidade.*

2. COMPREENDER O DOMÍNIO DA APLICAÇÃO

Este capítulo apresenta o resultado das atividades de levantamento de requisitos do sistema proposto. Tais atividades iniciam-se pela descrição do domínio do negócio na subseção 2.1. Uma descrição de como ocorre atualmente o processo de experimentação utilizando o MIT e o algoritmo no MATLAB é relatada na subseção 2.2. Os problemas a serem resolvidos do domínio do negócio são apresentados na subseção 2.3. E por fim, a subseção 2.4 mostra quais são as restrições do sistema.

2.1 DESCRIÇÃO DO DOMÍNIO DO NEGÓCIO

O Laboratório de Controle e Automação, situado na sala nove no CIPECA (Centro Integrado de Pesquisa em Controle e Automação), é supervisionado pelo Prof. Dr. Alessandro Goedel e tem como objetivo realizar experimentos com motores e máquinas elétricas.

2.2 DESCRIÇÃO DAS CARACTERÍSTICAS DO NEGÓCIO

O Laboratório possui um MIT, onde os estagiários e alunos que cursam o mestrado de Engenharia Elétrica realizam os experimentos.

Em um único experimento é possível coletar todas as variáveis disponíveis pelo equipamento de experimentação ou selecionar as variáveis que desejar.

O procedimento de experimentação ocorre quando o operador regula as variáveis de tensão para cada uma das três fases do MIT através de um painel de controle.

Durante o processo de experimentação é possível alterar a tensão em cada um das fases. Ao alterar a tensão em uma das fases, ocorrerá uma variação de tensão nas fases restantes.

O operador inicia o software MATLAB e abre um arquivo contendo um algoritmo. Neste algoritmo é modificado a taxa de amostragem e o tempo de aquisição.

A taxa de amostragem é medida em amostras/segundo com um sinal de 60 Hertz e o tempo de aquisição é medido em segundos.

Após a regulagem da taxa de amostragem e do tempo de aquisição, o operador executa o algoritmo e liga o motor através do painel de controle. É necessário aguardar alguns segundos entre essas duas operações para evitar a perda de dados.

Os sensores utilizados no equipamento são os seguintes:

- *Três do modelo LV25-P;*
- *Três do modelo LAH25-NP;*
- *Um Como Torque Type 4700;*
- *Um Varimot V.V -120.*

Os sensores servem para captar respectivamente os valores das variáveis de tensão, corrente, torque e velocidade. Esses valores são enviados para a placa aquisitora que está conectada a um computador.

Ao terminar a execução do experimento, são coletadas as variáveis que o operador designou no início da experimentação. Os dados são transferidos para uma matriz no formato txt e enviados ao MATLAB.

A matriz possui oito colunas na qual as variáveis são divididas na seguinte maneira:

- *Três de tensão;*
- *Três de corrente;*
- *Uma para o torque;*
- *Uma para a velocidade.*

A quantidade de linhas existentes em uma matriz é definida pelo valor da taxa de amostragem (valores coletados ao longo do tempo).

2.3 COMPREENDER O PROBLEMA A SER RESOLVIDO

Atualmente o processo de experimentação possibilita o acesso ao equipamento apenas para os usuários internos do laboratório. A idéia é permitir que usuários externos a instituição e remotamente localizados possam, através do uso da Internet acessar o equipamento do Laboratório de Controle e Automação e realizar os seus experimentos.

Os usuários devem previamente estar cadastrados no sistema e possuírem direitos de acesso ao sistema de controle de experimentação.

Outro ponto a ser analisado é que os dados dos experimentos são armazenados em uma matriz no formato txt, o que pode dificultar as operações de consulta e controle de experimentos.

O sistema a ser desenvolvido visa estabelecer a comunicação entre a placa de Aquisição National Instruments USB6009 e o computador.

Durante a execução do experimento as seguintes variáveis são coletadas:

- *Tensão;*
- *Corrente;*
- *Torque;*
- *Velocidade.*

Será necessária a criação de uma base de dados para o armazenamento e o gerenciamento das variáveis e dos experimentos executados. A base de dados também proporcionará um controle de usuários, o agendamento de experimentação, além de facilitar as consultas e geração de relatórios.

Outro ponto a ser abordado é que o sistema deve ter a premissa de elementos de: confiabilidade, operação (tanto local quanto remota), interface amigável e um feedback rápido em suas operações.

2.4 DESCRIÇÃO DAS COMPLEXIDADES (RESTRICÇÕES DO SISTEMA)

Podem ser observadas as seguintes complexidades referentes ao desenvolvimento do sistema:

- *A aplicação deverá ser desenvolvida usando tecnologia Web.*
- *A busca de Drivers compatíveis com a placa aquisitora para realizar a comunicação com o sistema.*
- *A criação de uma interface que atenda aos requisitos de confiabilidade, extensibilidade e usabilidade.*

3. COMPREENDER AS NECESSIDADES DOS USUÁRIOS

Este capítulo trata quais são as necessidades dos usuários. Os requisitos funcionais e requisitos não funcionais que foram levantados em entrevistas e reuniões com o Prof. Dr. Alessandro Goedel e os estagiários do laboratório são apresentados, respectivamente na subseção 3.1 e 3.2. Os atores e suas respectivas funções são descritas na subseção 3.3. Finalmente, a subseção 3.4 apresenta o diagrama de casos de uso do sistema proposto.

3.1 REQUISITOS FUNCIONAIS

Os requisitos funcionais apresentam as principais funcionalidades do sistemas, ou seja, eles descrevem o que o sistema deve fazer. Todavia, não apresentam como tais itens serão feitos. A tabela 1 apresenta os requisitos do sistema que foram levantados durante entrevistas com o Prof. Alessandro Goedel e os estagiários do laboratório.

Tabela 10 - Requisitos funcionais (contínua)

Código	Descrição
RF01	<i>O sistema deverá permitir a inclusão, exclusão e edição de usuários. O usuário pode ser interno ou externo. Um usuário possui os seguintes dados: nome, CPF, RG, login, senha e instituição.</i>
RF02	<i>Para o manuseio do painel de controle do equipamento o usuário externo precisará de um operador disponível. O usuário interno poderá operar este painel.</i>
RF03	<i>Após o usuário realizar o seu cadastro, será enviado uma solicitação de confirmação de cadastro ao email do administrador. Será necessária a confirmação do administrador para que o usuário tenha acesso ao sistema.</i>
RF04	<i>O administrador poderá alterar e consultar os seus dados. Um administrador possui os seguintes dados: nome, CPF, RG, login e senha.</i>
RF05	<i>O sistema deverá permitir a inclusão, edição e exclusão de instituições. Uma instituição possui os seguintes dados: nome, setor, campus.</i>
RF06	<i>O sistema deverá permitir o relatório de usuários por instituições.</i>
RF07	<i>O sistema deverá possuir um cadastro das variáveis já utilizadas nos experimentos, que são as seguintes: tensão, corrente, torque e velocidade. Uma variável possui os seguintes dados: nome, unidade de medida e sigla.</i>
RF08	<i>Além de gerenciar as variáveis atuais já utilizadas, o sistema deverá permitir a inclusão de novas variáveis.</i>
RF09	<i>Apenas o administrador terá a permissão para cadastrar, editar e excluir as variáveis.</i>
RF10	<i>Uma variável possui apenas um sensor associado a ela.</i>

Tabela 1 – Requisitos funcionais (contínua)

Código	Descrição
RF11	<i>O sistema deverá emitir relatórios de variáveis por sensores.</i>
RF12	<i>O sistema deverá permitir a inclusão, edição e exclusão de sensores. Um sensor possui os seguintes dados: modelo, fabricante, ano de aquisição e descrição.</i>
RF13	<i>O sistema deverá gerenciar os sensores que atualmente o equipamento possui e permitir a inclusão de novos sensores.</i>
RF14	<i>Apenas o administrador terá a permissão para cadastrar, editar e excluir sensores.</i>
RF15	<i>O sistema deverá realizar a inclusão, exclusão e edição dos fatores de escala. Um fator de escala possui os seguintes dados: nome, data, descrição e valor.</i>
RF16	<i>Apenas o administrador poderá cadastrar o fator de escala.</i>
RF17	<i>Um fator de escala está associado apenas a um sensor.</i>
RF18	<i>O sistema deverá realizar a inclusão, exclusão e edição de operadores. Um operador possui os seguintes dados: nome, CPF, RG, login e senha.</i>
RF19	<i>Apenas o administrador terá a permissão para cadastrar um operador.</i>
RF20	<i>Um operador está associado a um ou vários experimentos.</i>
RF21	<i>O sistema deverá permitir o cadastro de experimentos. Um experimento possui os seguintes dados: tempo de aquisição, taxa de amostragem, variáveis, horário, interesse e projeto.</i>
RF22	<i>Durante o cadastro de experimentos, o sistema deverá permitir que o usuário altere o valor de uma ou mais tensões e determinar o tempo em que ocorrerá a alteração.</i>
RF23	<i>O sistema deverá permitir que usuários externos realizem experimentos.</i>
RF24	<i>O sistema deverá permitir que o usuário escolha as variáveis que serão coletadas nos experimentos.</i>
RF25	<i>Um usuário poderá consultar todos os experimentos cadastrados na base de dados.</i>
RF26	<i>Os experimentos a serem realizados pelos usuários externos deverão possuir a confirmação de um operador.</i>
RF27	<i>Um operador poderá consultar todos os experimentos cadastrados na base de dados.</i>
RF28	<i>O administrador poderá consultar todos os experimentos cadastrados.</i>
RF29	<i>O sistema deverá permitir que apenas o administrador possa excluir os experimentos já realizados.</i>
RF30	<i>O processo para a realização de um experimento é dividido em cinco fases:</i> <ol style="list-style-type: none"> <i>5. Usuário envia solicitação de experimento ao operador.</i> <i>6. Operador envia uma resposta ao usuário. Caso a resposta seja negativa, o operador irá escrever uma justificativa explicando por que o experimento não pode ser realizado.</i> <i>7. Após a aprovação da fase anterior, o operador irá agendar o experimento</i> <i>8. O experimento será executado. Caso haja motivos para o cancelamento, será enviado um e-mail ao usuário explicando os motivos.</i> <i>9. Após a realização do experimento, os dados coletados serão armazenados na base de dados e disponibilizados no sistema.</i>
RF31	<i>O sistema deverá fornecer um arquivo de saída no formato txt com os dados referentes ao experimento.</i>
RF32	<i>O sistema deverá possuir um protocolo de mensagens. Um protocolo de mensagens possui: código, mensagem, remetente, destinatário e data de envio.</i>

Tabela 1 – Requisitos funcionais (conclusão)

Código	Descrição
RF33	<i>O sistema deverá emitir um relatório de quantos experimentos um usuário realizou.</i>
RF34	<i>O sistema deverá emitir um relatório de quais instituições realizaram experimentos.</i>
RF35	<i>Durante as fases do processo serão enviadas mensagens automaticamente para o e-mail do usuário, informando a situação do pedido de experimentação.</i>
RF36	<i>Para cada fase do processo, o sistema deverá armazenar um log com as seguintes informações: nome, data e descrição.</i>
RF37	<i>O sistema deverá emitir um relatório com os logs dos experimentos.</i>
RF38	<i>O usuário poderá consultar apenas o seu protocolo de mensagens dos seus experimentos.</i>
RF39	<i>O operador poderá consultar apenas o protocolo de mensagens dos experimentos que estejam associados a ele.</i>
RF40	<i>O administrador poderá consultar todos os protocolos de mensagens dos experimentos.</i>
RF41	<i>O administrador poderá emitir relatórios de todos os experimentos cadastrados.</i>
RF42	<i>O operador poderá emitir relatórios apenas de experimentos associados a ele.</i>
RF43	<i>O usuário poderá emitir relatórios apenas de experimentos associados a ele.</i>
RF44	<i>O sistema deverá proporcionar o agendamento dos experimentos.</i>
RF45	<i>A agenda de experimentação deverá possuir os seguintes dados: anos, meses, dias disponíveis, horários e quantidade de experimentos.</i>
RF46	<i>A agenda de experimentação é gerenciada apenas pelo administrador.</i>
RF47	<i>Caso o experimento esteja fora das cargas nominais, conjugado ou com desbalanceamento de tensão, o sistema deverá proporcionar um intervalo de pelo menos uma hora até o próximo experimento.</i>
RF48	<i>O agendamento de experimentação será realizado por um operador.</i>
RF49	<i>O sistema deverá emitir relatórios dos experimentos agendados por dia, mês e ano.</i>

3.2 REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais descrevem restrições sobre o processo de desenvolvimento. Foram levantados os seguintes requisitos não funcionais do sistema:

- 4) O sistema deverá possuir três níveis de acesso (usuário, operador e administrador).*
- 5) Deverá ser implementado um controle de visão ao acesso. Este controle irá restringir o que cada usuário poderá visualizar no sistema.*
- 6) O sistema deverá ser confiável e que possibilite a segurança dos dados armazenados.*

3.3 TAREFA - IDENTIFICAR ATORES E FUNÇÕES

Os atores participam no sistema realizando as suas respectivas funções. A função reflete o papel que o ator desempenha no sistema.

A tabela 2 representa os atores e as suas respectivas funções no sistema.

Tabela 11 - Respektivas funções dos atores no sistema

Ator	Função
<i>Operador</i>	<ol style="list-style-type: none"> 1. Consultar todos os experimentos cadastrados na base de dados. 2. Consultar apenas os protocolos de mensagens dos experimentos que estejam associados a ele. 3. Confirmar a solicitação de experimentos dos usuários. 4. Realizar o agendamento dos experimentos. 5. Regular as variáveis de tensão no painel de controle, conforme informado no pedido de experimento. 6. Executar os experimentos agendados.
<i>Usuário</i>	<ol style="list-style-type: none"> 1. Gerenciar os seus dados. 2. Enviar solicitações de experimentos. 3. Consultar todos os experimentos cadastrados na base de dados. 4. Consultar apenas os protocolos de mensagens dos experimentos que estejam associados a ele.
<i>Administrador</i>	<ol style="list-style-type: none"> 1. Realizar o cadastro de operadores. 2. Realizar o cadastro de sensores. 3. Realizar o cadastro do fator de escala. 4. Realizar o cadastro de variáveis. 5. Confirmar a solicitação de cadastro do usuário. 6. Criar a agenda de experimentos. 7. Consultar todos os experimentos cadastrados na base de dados. 8. Consultar os protocolos de mensagens de todos os experimentos.

3.4 DEFINIR CASOS DE USO

O diagrama de casos de uso descreve um cenário que mostra as funcionalidades do sistema.

Um diagrama de casos de uso possui os seguintes componentes: ator, casos de uso e seus respectivos relacionamentos.

O ator é representado por um boneco. Um ator é um usuário do sistema, que pode ser um usuário humano ou um sistema computacional.

O caso de uso é representado por uma elipse. Um caso de uso define uma funcionalidade do sistema.

Em um relacionamento o ator pode se comunicar com um ou mais casos de uso, da mesma forma um caso de uso pode se relacionar com um ou mais atores.

Na Figura 1 é apresentado o diagrama de caso de uso do sistema.

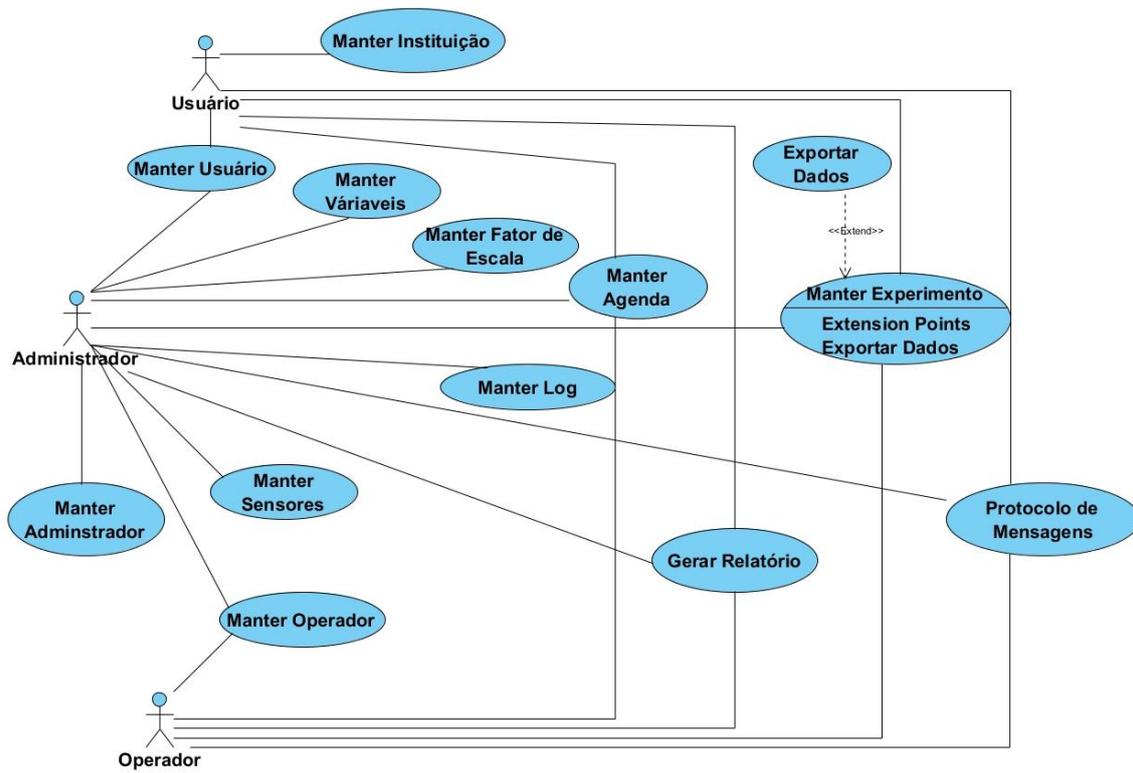


Figura 49 - Diagrama de Casos de uso

4. TECNOLOGIAS

Neste capítulo são apresentadas as tecnologias a serem utilizadas no desenvolvimento do sistema. Este sistema será desenvolvido utilizando a linguagem de programação JAVA e outras tecnologias descritas a seguir:

- *JSF 2.1 - Conforme Jacobi (2007), o JSF é um framework de componentes UIs que permite a migração de tecnologias ultrapassadas como VTs baseados em caracteres, para tecnologias e plataformas atualizadas e baseadas em padrões, como JSF e Java.*

O JSF permite o aproveitamento das últimas tecnologias causando o mínimo de impacto nas aplicações JSF existente. A tecnologia também adiciona o reuso de funcionalidades e aparência visual a indústria de software. Dessa maneira, é fornecido um caminho fácil para criar UIs a partir de um conjunto de componentes UI reutilizáveis. (Jacobi, 2007).

O modelo de navegação do JSF fornece uma opção de ajuste das regras de navegação. As regras de navegação definem como será realizada a navegação de uma view para outra em uma aplicação Web. (Jacobi, 2007).

Normalmente, as regras de navegação no JSF são definidas dentro do arquivo de configuração do JSF (faces-config.xml) e são direcionadas a cada página. (Jacobi, 2007).

- *PrimeFaces 2.2 - O PrimeFaces é um framework da Prime Teknoloji (empresa da Turquia) que oferece um conjunto de componentes ricos para o JavaServer Faces. Seus componentes foram construídos para trabalhar com AJAX por “default”, isto é, não é necessário nenhum esforço extra por parte do desenvolvedor para realização de chamadas assíncronas ao servidor. Além disso, o PrimeFaces dá suporte à criação de funcionalidades que fazem uso de Ajax Push e permite a aplicação de temas (skins) com o objetivo de mudar a aparência dos componentes de forma simples. (BOEKEL,*
- *Hibernate 3.2.5 - O Hibernate é um framework de persistência que permite a utilização de banco de dados relacional.*

Uma das vantagens de utilizá-lo é que o desenvolvimento de aplicação se torna mais rápido, por não ter que escrever códigos e mais códigos SQL.

A idéia do hibernate é encapsular todo o código SQL Jdbc por trás de sua implementação, ficando transparente para o desenvolvedor.

- *Banco de Dados Relacional - PostgreSQL 9.1 - O modelo relacional de banco de dados foi desenvolvido pelo cientista inglês DR. Edgar Frank “Ted” Codd em junho de 1970, com a publicação do artigo “A Relational Model of Data for Large Shared Data Banks” na revista Communications of the ACM, v.13, n.6, PP. 377-387, publicada por Association for Computing Machinery, Inc.*

Baseado em (Manzano, 2008), além de conceitos básicos como: campo, tabela, banco de dados, e base de dados. O modelo relacional insere os seguintes conceitos:

- *Chave primaria;*
- *Chave estrangeira;*
- *Integridade referencial.*

O programa PostgreSQL é um SGBDOR baseado no gerenciador banco de dados POSTGRES versão 4.2. (Manzano, 2008).

- *NETBEANS IDE 7.1.1 - O IDE NetBeans é uma ferramenta open-source para desenvolvedores de software.*

Inclui ferramentas necessárias para criar um ambiente de desenvolvimento web e aplicações móveis com a plataforma Java. A versão 7.1.1 introduz o suporte para o JavaFX 2.0, bem como ferramentas para depuração visual do Swing e interfaces de usuário JavaFX. Outro destaque é a inclusão de novos recursos de depuração de PHP e JEE.

5. ARQUITETURA INICIAL

Neste capítulo é apresentada a arquitetura a ser utilizada no desenvolvimento do sistema. A subseção 5.1 descreve os pacotes a serem utilizados no sistema, a subseção 5.2 apresenta a divisão em módulos e por fim, a subseção 5.2.1 trata dos módulos a serem desenvolvidos neste trabalho conforme a Figura 3.

5.1 DIAGRAMA DE PACOTES

O Diagrama de pacotes descreve os pacotes ou partes do sistema, divididos em agrupamentos lógicos.

Na Figura 2 são apresentados os devidos pacotes do sistema, juntamente com as suas dependências.

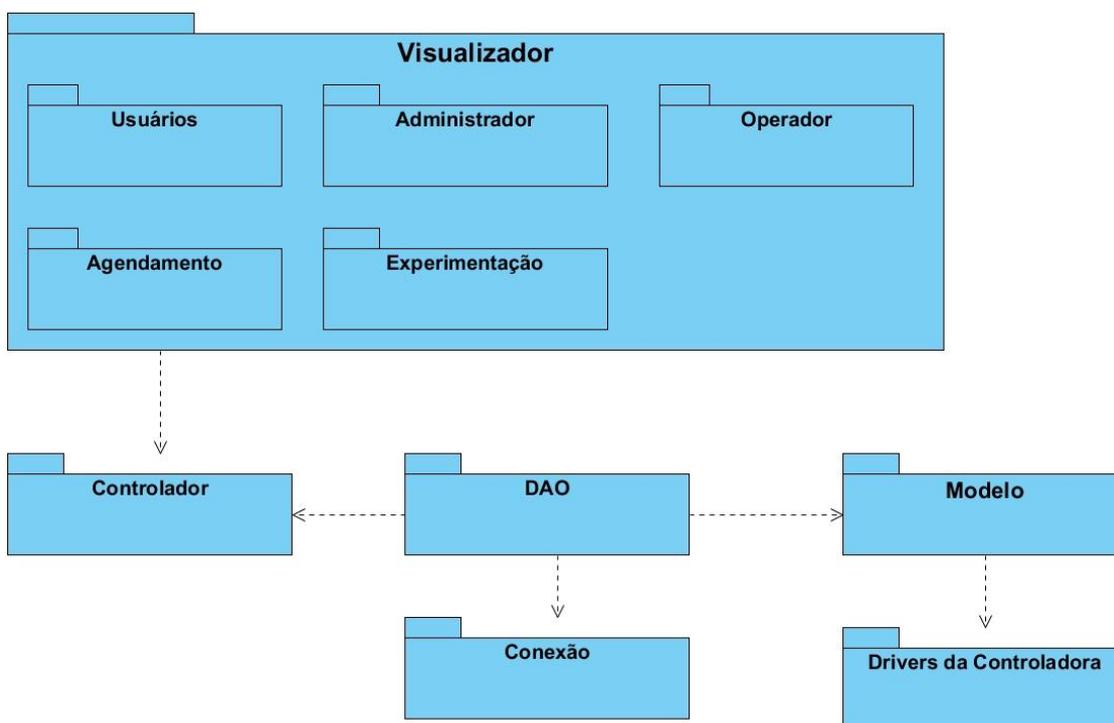


Figura 50 - Diagrama de Pacotes

O diagrama apresentado na Figura 2 exibe os pacotes que são descritos a seguir:

- *DAO*: Possui as classes com as regras de negócio do sistema.
- *Modelo*: Possui os beans das entidades (GET e SET).
- *Conexão*: Possui a classe de conexão com o banco de dados.

- *Controlador: Possui classes com a função de gerenciar o redirecionamento das páginas do sistema.*
- *Pacote Visualizador: Possui as páginas xhtml do sistema. É composto pelos seguintes pacotes:*
 - *Operador: O pacote de operador possui páginas xhtml restritas, que são visualizadas apenas pelos operadores.*
 - *Usuário: O pacote de usuário possui páginas xhtml restritas, que são visualizadas apenas pelos usuários.*
 - *Administrador: O pacote administrador possui páginas xhtml restritas, que são visualizadas apenas pelo administrador.*
 - *Agendamento: O pacote de agendamento possui as classes e páginas xhtml restritas aos demais usuários (operadores, usuários e administrador) que estejam logados.*
 - *Experimentação: O pacote de experimentação possui as classes e páginas xhtml restritas aos demais usuários (operadores, usuários e administrador) que estejam logados no sistema.*
- *Drives da Controladora: O pacote Drivers da Controladora possui as classes responsáveis pela conexão com a Placa Aquisitora.*

5.2 DIAGRAMA DE SUBSISTEMAS

O diagrama de subsistemas representa os subsistemas envolvidos no sistema. Além de representar as dependências entre os módulos.

A Figura 3 demonstra o comportamento dos subsistemas do sistema em questão, sendo que a parte destacada em vermelho representa os subsistemas a serem desenvolvidos nesta proposta.

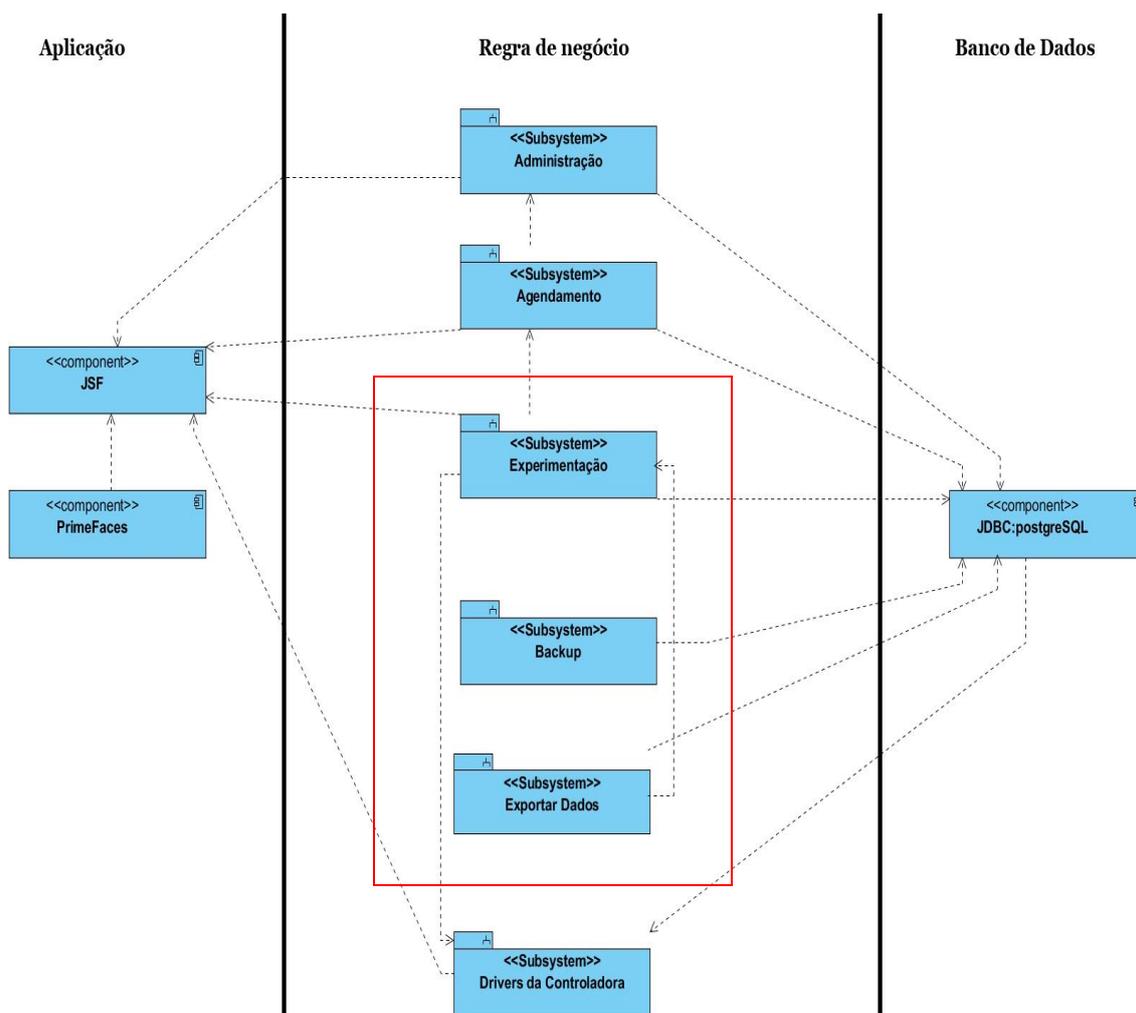


Figura 51 - Diagrama de Subsistemas

O diagrama apresentado na Figura 3 exibe os subsistemas que são descritos a seguir:

- *Administração:* Neste módulo serão implementados os cadastros do sistema exceto o cadastro de experimentação. O módulo é constituído pelos seguintes diagramas de pacotes:
 - *Administrador;*
 - *Usuário;*
 - *Operador;*
 - *Bean;*
 - *DAO;*
 - *Conexão.*

- *Agendamento: O módulo de agendamento é responsável por gerenciar os dias e horários disponíveis para a realização de experimentos, inclusive a quantidade de experimentos a ser realizado por dia.*

Este módulo é constituído pelos seguintes pacotes:

- *Agendamento;*
 - *Bean;*
 - *DAO;*
 - *Conexão;*
 - *Usuário;*
 - *Operador;*
 - *Administração.*
-
- *Experimentação: O módulo de experimentação tem a função de cadastrar e gerenciar os experimentos. Este módulo é constituído pelos seguintes pacotes:*
 - *Experimentação;*
 - *Operador;*
 - *Usuário;*
 - *DAO;*
 - *BEAN;*
 - *Conexão.*
-
- *Drivers da Controladora: O módulo de Drivers da controladora possui os drivers para a comunicação com a placa aquisitora. Este módulo é constituído pelos seguintes pacotes:*
 - *Controladora;*
 - *Conexão;*
 - *DAO.*
-
- *Backup: O módulo de Backup tem como função criar uma cópia de segurança dos dados do sistema.*
-
- *Exportar Dados: O módulo de exportar dados tem como função criar arquivos de saída dos dados dos experimentos e disponibilizá-los no sistema.*

5.2.1 Subsistemas a serem desenvolvidos nesta proposta

Esta proposta de trabalho compõe os seguintes módulos a serem desenvolvidos:

- *Experimentação: O módulo de experimentação irá cadastrar e gerenciar os experimentos.*

A realização do experimento é dividida em fases. Inicialmente, o usuário envia uma solicitação de cadastro de experimento ao operador, após a confirmação de solicitação de cadastro será iniciada a fase seguinte que é o agendamento do experimento. E finalmente, na última fase o experimento será executado e os dados coletados serão armazenados na base de dados e disponibilizados no sistema.

A cada fase do processo de experimentação serão armazenados logs com informações como data e horário em que a fase ocorreu. Além de, enviar mensagens automáticas para o e-mail do usuário, informando a situação do pedido do seu experimento.

A relação dos casos de uso da Figura 1 com os requisitos da Tabela 2 que pertencem a este módulo são apresentados na Tabela 3.

Tabela 12- Relação de Requisitos com Casos de uso - Módulo Experimentação

Casos de Uso	Código dos Requisitos									
<i>Manter Experimento</i>	<i>RF21</i>	<i>RF22</i>	<i>RF23</i>	<i>RF24</i>	<i>RF25</i>	<i>RF26</i>	<i>RF27</i>	<i>RF28</i>	<i>RF29</i>	<i>RF30</i>
<i>Manter Log</i>	<i>RF36</i>									
<i>Protocolo de Mensagens</i>	<i>RF32</i>	<i>RF35</i>	<i>RF38</i>	<i>RF39</i>	<i>RF40</i>					
<i>Gerar Relatório</i>	<i>RF33</i>	<i>RF34</i>	<i>RF37</i>	<i>RF41</i>	<i>RF42</i>	<i>RF43</i>				

Conforme o diagrama de classes da figura 4 é possível observar uma dependência deste módulo com os módulos de Administração e Agendamento que serão desenvolvidos por Marcus Vinicius Sanches.

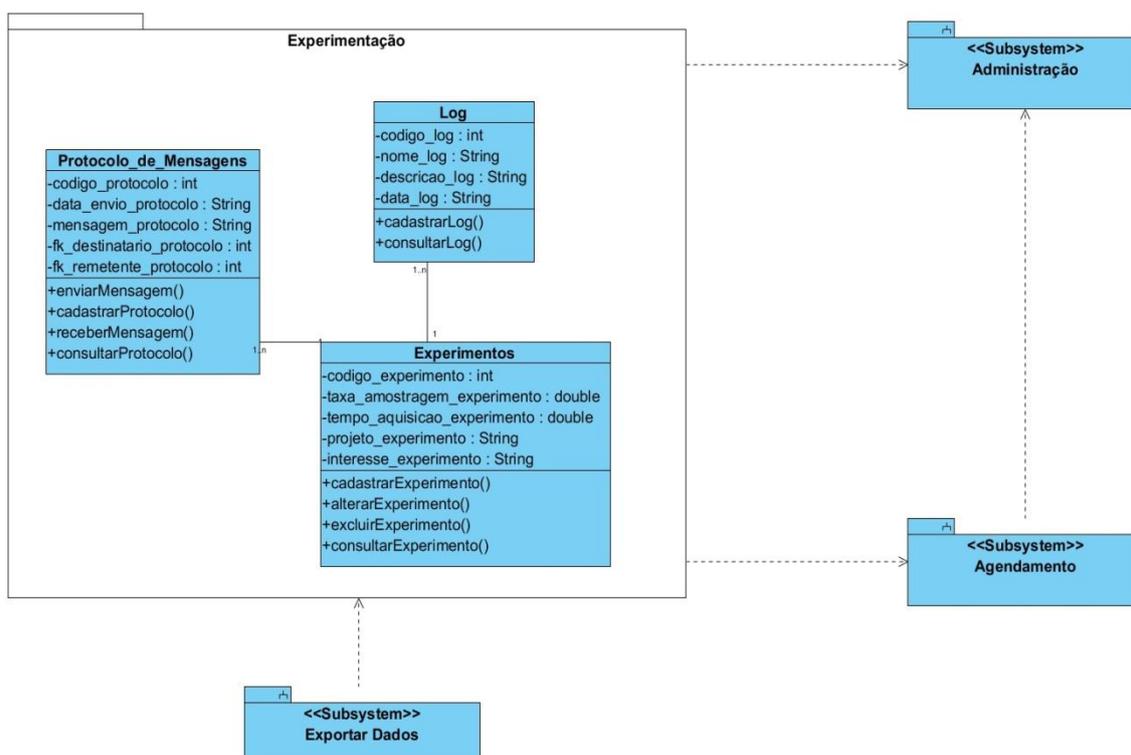


Figura 52 - Módulo Experimentação

- *Exportar dados: O módulo de Exportação de dados irá transformar os dados de um experimento em um arquivo de saída.*

Após a realização do experimento, o sistema irá automaticamente gerar um arquivo de saída no formato txt com os dados coletados na experimentação. Este arquivo será disponibilizado posteriormente no sistema.

A relação dos casos de uso da Figura 1 com os requisitos da Tabela 2 que pertencem a este módulo são apresentados na Tabela 3.

Tabela 13 - Relação de Requisitos com Casos de uso - Módulo Exportar dados

Casos de Uso	Código dos Requisitos
<i>Exportar Dados</i>	<i>RF31</i>

Conforme o diagrama de classes da figura 5 este subsistema depende do módulo de experimentação.

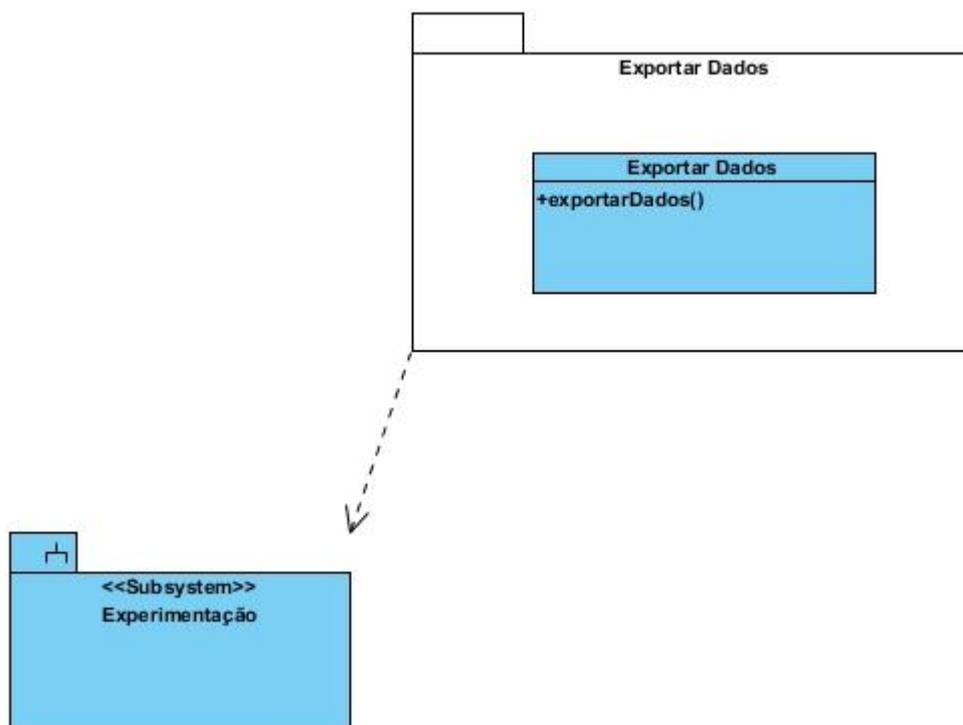


Figura 53 - Módulo Exportar Dados

- *Backup: O módulo de Backup irá criar uma cópia de segurança dos dados de cadastros e experimentos do sistema. Esta cópia será utilizada caso o sistema esteja corrompido por algum erro.*

O sistema irá realizar o backup offline, em um horário com poucos usuários. Este backup será realizado semanalmente.

Conforme o diagrama de classes da figura xxx este subsistema depende do módulo de experimentação.

6. METODOLOGIA

Este capítulo apresenta a metodologia escolhida para ser utilizada no desenvolvimento do projeto.

O Processo Unificado é um ciclo de vida que permite a divisão de atividades em fases com as iterações necessárias ao desenvolvimento do projeto. A subseção 6.1 apresenta uma descrição sobre a metodologia do UP (Processo Unificado).

6.1 PROCESSO UNIFICADO

Modelos prescritivos de processo de software têm sido aplicados durante muitos anos em um esforço de trazer ordem e estrutura para o desenvolvimento de softwares. Cada um desses modelos convencionais sugere um fluxo de processo um tanto diferente, mas todos realizam o mesmo conjunto de atividades genéricas de arcabouço: comunicação, planejamento, modelagem, construção e implantação. (Pressman 2006).

O Processo Unificado é um processo de software “orientado por casos de uso, centrado na arquitetura, iterativo e incremental”, projetado como um arcabouço para métodos e ferramentas UML. (Pressman 2006).

O Processo Unificado é um modelo incremental no qual quatro fases são definidas, conforme a Figura 4.

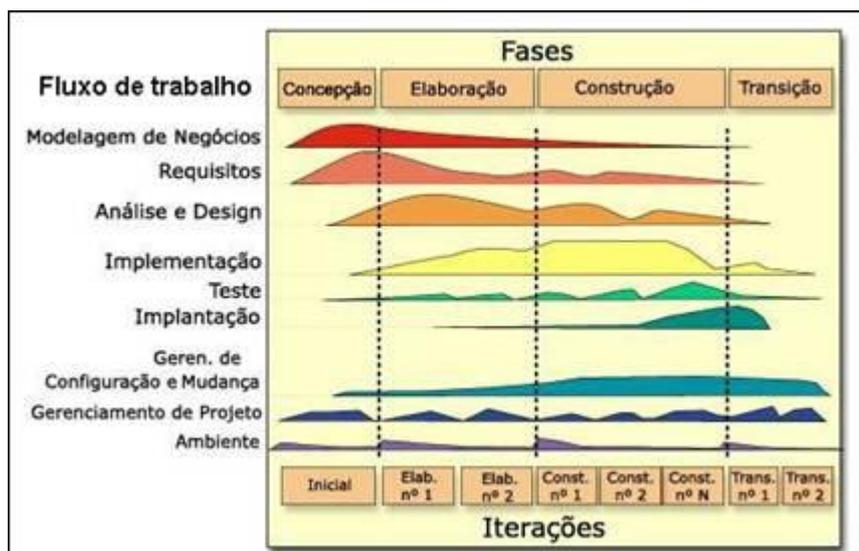


Figura 54 – UP

- *Fase de Concepção*

A fase de concepção engloba tanto a comunicação com o cliente quanto atividades de planejamento. Enfatiza o desenvolvimento e refinamento de casos de uso como modelo principal.

Os artefatos gerados na fase de concepção são: documento de visão, modelo inicial de casos de uso, caso de negócios inicial, avaliação inicial de risco, plano de projeto, fases e interações, modelo de negócio, se necessário um ou mais protótipos.

- *Fase de Elaboração*

A fase de elaboração possui atividades de comunicação com o cliente e modelagem com foco na criação de modelos de análise e projeto com ênfase nas definições de classes e representações arquiteturais.

Os artefatos gerados na fase de elaboração são: modelo de caso de uso, requisitos suplementares, modelo de análise, descrição da arquitetura do software, protótipo arquitetural executável, modelo de projeto preliminar, lista de riscos revisada, plano de projeto incluindo planos de iteração, fluxos de trabalho adaptados marcos, produtos técnicos de trabalho, manual preliminar do usuário.

- *Fase de Construção*

A fase de construção refina e então traduz o modelo de projeto para componentes de software implementados.

Os artefatos gerados na fase de construção são: modelo de projeto, componentes de software, incremento integrado de software, plano e procedimento de teste, caso de teste, documentação de apoio, manuais do usuário, manuais de instalação, descrição do incremento atual.

- *Fase de Transição*

Na fase de transição é transferido o software do desenvolvedor para o usuário final para testes beta e aceitação.

Os artefatos gerados na fase de transição são: incremento de software entregue, relatório de teste beta, realimentação geral do usuário.

O sistema a ser desenvolvido apresentará 3 iterações, sendo que a cada nova iteração serão desenvolvidas novas funcionalidades, além de refinar e corrigir possíveis erros das iterações anteriores.

A primeira iteração será composta pelo módulo de experimentação, a segunda por exportar dados e finalmente, a terceira iteração será implementado o módulo de backup.

7. ANÁLISE DE PONTOS POR FUNÇÃO

Este capítulo apresenta a técnica de APF (Análise de Pontos por Função). A APF é um método criado por Allan Albrecht. Este método tem como objetivo medir o tamanho funcional de um software, não se importando em como ele desenvolvido, ou seja, qual foi a linguagem de programação utilizada ou outras tecnologias. A subseção 8.1 exibe o cálculo dos PF.

7.1 CÁLCULO DOS PONTOS POR FUNÇÃO

A seguir será apresentado o cálculo dos pontos por função dos módulos de Experimentação, Exportar Dados e Backup.

ALI	QUANTIDADE	TOTAL
Baixa (ALlb)	13	91
Média(ALLm)	0	0
Alta (ALLa)	0	0
AIE		
Baixa (AIEb)	0	0
Média (AIE m)	0	0
Alta (AIEa)	0	0
IE		
Baixa (IEb)	5	15
Média (IE m)	2	8
Alta (IEa)	3	18
OE		
Baixa (OEB)	0	0
Média (OEm)	1	5
Alta (OEa)	4	28
	TOTAL	165

8. CRONOGRAMA

Este capítulo apresenta o cronograma proposto para o desenvolvimento dos módulos de Experimentação, Exportar Dados e Backup. No cronograma é exibido quando as tarefas são executadas durante cada iteração conforme o quadro 1.

Tarefa	Nov/ 2012				Dez/ 2012				Jan/ 2013				Fev/ 2013				Mar/ 2013				Abril/ 2013							
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
Modelagem de negócio	■	■														■									■			
Requisitos	■	■	■																						■			
Análise e Design	■	■	■																									
Implementação			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Teste											■	■	■	■	■	■			■	■					■	■	■	■
Implantação																												
Elaboração do TCC	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Apresentação do TCC																												■
	<i>Iteração 1</i>												<i>Iteração 2</i>				<i>Iteração 3</i>											

Quadro 1 - Cronograma do Trabalho de Diplomação

9. REFERÊNCIAS

SUETAKE, Marcelo ; SILVA, Ivan Nunes da ; GOEDEL, A. . **Embedded DSP-Based Compact Fuzzy System and Its Application for Induction Motor V/f Speed Control (Qualis Engenharia IV)**. *IEEE Transactions on Industrial Electronics (1982. Print)*, v. 58, p. 750-760, 2011.

SUETAKE, Marcelo ; SILVA, Ivan Nunes da ; GOEDEL, A. . **Sistema fuzzy compacto embarcado em DSP e sua aplicação para controle V/f de motores de indução (Qualis Engenharia IV)**. *Controle & Automação (Online)*, v. 21, p. 245-259, 2010.

GOEDEL, A. ; Graciola, Clayton Luiz ; SILVA, S. A. O. ; NASCIMENTO, Claudionor Francisco Do ; SUETAKE, Marcelo . **A Comparative Study for Single and Multilayer Neural Networks Applied to Speed Estimation in Induction Motors**. In: *XIX IEEE International Congress on Electrical Machines (ICEM 2010)*, 2010, Roma. *IEEE Proceedings of XIX International Congress on Electrical Machines (paper code PID1372065)*, 2010.

Doutorado em Engenharia Elétrica (Conceito CAPES 6) .
Universidade de São Paulo. **Estimador Neural de Velocidade Para Motores de Indução Trifásicos (Qualificado em 16/05/2007), Ano de Obtenção: 2007.**

JACOBI, Jonas. *Pro JSF e Ajax: Construindo Componentes Ricos Para a Internet*. São Paulo: Ciência Moderna, 2007.

MANZANO, José Augusto N.G. *PostgreSQL 8.3.0 Interativo: Guia de Orientação e Desenvolvimento*. 1. ed. São Paulo: Érica, 2008.

NetBeans IDE - The Smarter Way to Code Disponível em: <
<http://netbeans.org/features/index.html> > Acessado em 18/05/2012.

Hibernate Disponível em: < <http://javawora.blogspot.com.br/2007/08/hibernate.html> >
Acessado em 10/09/2012.

PRESSMAN, Roger S. **Engenharia de software**. 6. ed. São Paulo: McGraw-Hill, 2006.

Por dentro do PrimeFaces 2.2. disponível em: < <http://www.devmedia.com.br/por-dentro-do-primefaces-2-2-artigo-revista-java-magazine-93/21651> > Acessado em 10/08/2012.96949849