

RICARDO RAMOS DE OLIVEIRA

**DESENVOLVIMENTO DE UM SISTEMA DE INFORMAÇÃO WEB
UTILIZANDO PADRÕES DE PROJETO E JSF PARA UM
LABORATÓRIO DE ANÁLISES FÍSICO-QUÍMICAS**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

LAVRAS
MINAS GERAIS – BRASIL
2009

RICARDO RAMOS DE OLIVEIRA

**DESENVOLVIMENTO DE UM SISTEMA DE INFORMAÇÃO WEB
UTILIZANDO PADRÕES DE PROJETO E JSF PARA UM
LABORATÓRIO DE ANÁLISES FÍSICO-QUÍMICAS**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Área de Concentração:

Sistemas de Computação

Orientador:

André Vital Saúde

LAVRAS
MINAS GERAIS – BRASIL
2009

RICARDO RAMOS DE OLIVEIRA

**DESENVOLVIMENTO DE UM SISTEMA DE INFORMAÇÃO WEB
UTILIZANDO PADRÕES DE PROJETO E JSF PARA UM
LABORATÓRIO DE ANÁLISES FÍSICO-QUÍMICAS**

Monografia de graduação apresentada ao
Departamento de Ciência da Computação
da Universidade Federal de Lavras como
parte das exigências do curso de Ciência
da Computação para obtenção do título de
Bacharel em Ciência da Computação.

Aprovada em ____/____/____

Prof. Dr. Antônio Maria Pereira de Resende

Prof. Dra. Juliana Galvani Greggi

Prof. Dr. Paulo Henrique de Souza Bermejo
(Co-Orientador)

Prof. Dr. André Vital Saúde
(Orientador)

LAVRAS
MINAS GERAIS – BRASIL

"A sabedoria do humilde levantará a sua cabeça e o fará sentar-se no meio dos grandes."

Ec.11:1

Dedico à minha mãe Luzia, ao meu pai José, a minha irmã Cecília que contribuíram de um modo muito especial para o meu crescimento como ser humano e como profissional. Amo todos vocês!

AGRADECIMENTOS

Agradeço,
primeiramente a Deus que está sempre do meu lado.

Aos meus pais José e Luzia e a minha irmã Cecília, pessoas que me apoiaram e ajudaram a realizar os meus sonhos e me ensinaram a lutar pelos meus objetivos.

Aos meus tios Jair e Aurora, que estiveram presentes durante minha trajetória acadêmica, acreditando e ajudando a concretizar meus sonhos. Sempre serei grato a vocês.

Aos meus amigos pelo companheirismo e pelas horas de descontração. Sempre me lembrarei de todos vocês. Obrigado à todos.

Ao Laboratório de Análises Físico-Químicas de Aguardente da Universidade Federal de Lavras pelos dados fornecidos e a professora Maria das Graças Cardoso pela oportunidade, confiança e apoio durante a realização deste trabalho.

Por fim, agradeço ao meu orientador, o professor André Vital Saúde, por todo seu caráter, profissionalismo, paciência e dedicação para comigo durante as diversas atividades de pesquisa e andamento deste trabalho.

Desenvolvimento de um Sistema de Informação Web utilizando padrões de projeto e JSF para um laboratório de análises Físico-Químicas

RESUMO

Atualmente o Laboratório de Análises Físico-Químicas de Aguardente (LAFQA) da Universidade Federal de Lavras (UFLA) não possui um sistema de software que atenda às suas necessidades. O objetivo do presente trabalho foi desenvolver um sistema de informação Web personalizado para informatizar os processos envolvidos em cada etapa das análises de aguardente e cachaça. Para isso, foi desenvolvido um estudo de natureza tecnológica de caráter exploratório. O sistema de informação implementado baseou-se em padrões de projeto e tecnologia *JavaServer Faces* (JSF). Como resultado desse trabalho, obteve-se um sistema de informação Web que automatiza os procedimentos de análises Físico-Químicas e permite a um laboratório dessa natureza realizar suas atividades com mais eficiência e efetividade, devido a informatização.

Palavras Chaves: Desenvolvimento Sistema Web, Gestão da Informação e Padrões de Projeto.

Development of a Web Information System using Design Patterns and JSF in a Physical Chemical Analyses Laboratory

ABSTRACT

Nowadays the Physical Chemical Analyses Laboratory (LAFQA) at Universidade Federal de Lavras (UFLA) does not have a software system that considers its needs. The present work aimed to develop a personalized Web information system to computerize the involved processes on each stage of “aguardente” and “cachaça” analyses. For that, a study of technological nature with an exploratory character was developed. The implemented information system was based in design patterns and *JavaServer Faces* (JSF) technology. As a result of this work, a Web information system that automates the procedures for physical chemical analyses and enables a laboratory of this nature to fulfill its activities with more efficiency and effectiveness due to computerization was acquired.

Keywords: Web System Development, Information Management and Design Patterns

SUMÁRIO

LISTA DE FIGURAS.....	08
LISTA DE TABELAS.....	11
LISTA DE ABREVIATURAS E SIGLAS.....	12
1.INTRODUÇÃO	14
2.TECNOLOGIAS E PADRÕES DE PROJETO WEB DE DESENVOLVIMENTO.....	17
2.1.Servidores e Clientes Web.....	19
2.2.Tecnologias de Desenvolvimento Web.....	23
2.3. A arquitetura da aplicação Web.....	28
2.4. Padrões de Projeto.....	30
2.4.1. O padrão de Projeto Model View Controller MVC.....	31
2.4.2. O Padrão de Projeto Data Access Object (DAO).....	32
2.4.3. O Padrão de Projeto Factory.....	33
3.ANÁLISE FÍSICO-QUÍMICAS REALIZADAS EM LABORATÓRIOS.....	35
3.1.Características das bebidas	35
3.2.Cálculos Físico-Químicos do Laboratório.....	36
3.2.1 Análise do Exame Organoléptico.....	37
3.2.2 Análise do Grau Alcoólico Real a 20°C(% V/V).....	37
3.2.3 Análise do Extrato Seco a 100 °C (g/L).....	38
3.2.4 Análise de Álcool Metílico (mg/100mL de Álcool Anidro).....	39
3.2.5 Análise da Acidez Volátil em Ácido Acético (mg/100mL de ÁlcoolAnidro).....	39
3.2.6 Análise de Aldeídos em Aldeídos Acético (mg/100mL de Álcool nidro).....	41
3.2.7 Análise de Ésteres em Acetato de Etila (mg/100mL de Álcool Anidro).....	41
3.2.8 Análise Densidade Relativa (20/20°C).....	42
3.2.9 Soma dos Componentes Secundários (mg/100mL de Álcool Anidro).....	43
3.2.10 Curva Padrão.....	43
3.2.11 Curva Padrão do Álcool Superior.....	45

3.2.12 Curva Padrão do Cobre.....	46
3.2.13 Curva Padrão do Furfural.....	47
3.2.14 Análise do Álcool Superior (mg/100mL de Álcool Anidro).....	48
3.2.15 Análise de Cobre (mg/L).....	49
3.2.16 Análise de Furfural (mg/100mL de Álcool Anidro).....	50
<hr/>	
4.METODOLOGIA.....	52
4.1.Tipo da Pesquisa Realizada.....	52
4.2. Modelo de Processo de Desenvolvimento de software Utilizado.....	52
4.3.Levantamento de Requisitos.....	55
4.4.Documento de Requisitos do Sistema.....	59
4.5.Projeto do Sistema de Informação Web.....	63
4.5.1. Arquitetura do Software Web.....	63
4.5.2. Estrutura de Classes das Análises Laboratoriais.....	65
4.5.3. Diagramas de Seqüência do sistema Web.....	70
4.5.4. Modelo de Entidade Relacionamento MER.....	75
4.5.5. Modelo de Entidade Relacionamento Lógico.....	76
4.6.Implementação do Sistema de Informação Web.....	77
4.7.Testes do sistema de informação Web.....	78
5.RESULTADOS E DISCUSSÃO.....	79
5.1.Descrição do Software Web Desenvolvido.....	79
6.CONCLUSÕES.....	85
7.REFERÊNCIAS BIBLIOGRÁFICAS.....	87

LISTA DE FIGURAS

Figura 2-1 – Processo de solicitação e resposta.....	20
Figura 2-2 – Processo detalhado de uma solicitação	20
Figura 2-3 – Processo detalhado de uma resposta	21
Figura 2-4 – Processo de requisição e resposta HTTP	23
Figura 2-5 – Etapas da primeira execução de uma página JSP.....	26
Figura 2-6 – Relacionamento entre Servlets, Container e servidor	27
Figura 2-7 – Processamento de uma solicitação pelo Container	29
Figura 2-8 – Variação do Padrão de Projeto MVC usando Servlet e JSP	32
Figura 2-9 – Estrutura do Padrão de Projeto Data Access Object DAO.....	33
Figura 2-10 – Representação do Padrão de Projeto Factory.....	34
Figura 3-1 – Representação da Análise Grau Alcolico.....	38
Figura 3-2 – Representação da Análise Extrato Seco	38
Figura 3-3 – Representação da Análise Álcool Metílico	39
Figura 3-4 – Representação da Análise Acidez Volátil	40
Figura 3-5 – Representação da Análise Aldeídos	41
Figura 3-6 – Representação da Análise de Ésteres	42
Figura 3-7 – Representação da Análise de Densidade Relativa.....	43
Figura 3-8 – Representação da Análise Soma dos Componentes Secundários .	43
Figura 3-9 – Representação da Análise do Cálculo de R quadrado.....	45
Figura 3-10 – Representação da curva padrão do Álcool Superior	46

Figura 3-11 – Representação da curva padrão do Cobre	47
Figura 3-12 – Representação da curva padrão do Furfural	48
Figura 3 13 – Representação do cálculo da Análise do Álcool Superior.....	49
Figura 3 14 – Representação do cálculo da Análise de Cobre.....	50
Figura 3 15 – Representação do cálculo da Análise do Furfural.....	51
Figura 4-1 – Desenvolvimento evolucionário.....	53
Figura 4-2 – Cronograma de Atividades (1º semestre).....	54
Figura 4-3 – Cronograma de Atividades (2º semestre).....	54
Figura 4-4 – Diagrama de atividade do funcionamento do sistema do LAFQA.....	57
Figura 4-5 – Diagrama de Caso de Uso do Software do LAFQA.	58
Figura 4-6 – Arquitetura do Sistema de Software Web.....	64
Figura 4-7 – Estrutura de herança do Projeto.....	66
Figura 4-8 – Classe da Análise do Grau Alcoólico.....	67
Figura 4-9 – Classe da Análise com Curva.....	68
Figura 4-10 – Classe da Análise do Álcool Superior.....	69
Figura 4 11 – Estrutura das classes Curva Padrão e Regressão.....	70
Figura 4-12 – Operação de Create dos dados de um Produtor.....	71
Figura 4-13 – Operação de Update dos dados de um Produtor.....	72
Figura 4-14 – Operação de Read dos dados de um Produtor.....	73
Figura 4-15 – Operação de Delete dos dados de um Produtor.....	74

Figura 4-16 – Diagrama de Entidade Relacionamento do Software.....	75
Figura 4-17 – Diagrama da Base de Dados do Sistema de Software.....	76
Figura 4-18 – Camadas do Sistema de Software.....	77
Figura 5-1 – Tela Inicial do Sistema de Software.....	80
Figura 5-2 – Tela de Exibição de Listagem de Técnicos.....	81
Figura 5-3 – Tela de Cadastro da Solicitação de Serviço.....	82
Figura 5-4 – Tela de Análise do Extrato Seco.....	83
Figura 5-5 – Browser exibindo um laudo do Sistema.....	84

LISTA DE TABELAS

Tabela 4-1 Requisitos Funcionais do Sistema.....60

Tabela 4-2 Requisitos Não Funcionais do Sistema.....62

LISTA DE ABREVIATURAS E SIGLAS

<i>JSF</i>	<i>JavaServer Faces</i>
<i>MVC</i>	<i>Model-View-Controller</i>
LAFQA	Laboratório de Análises Físico-Químicas de Aguardente
<i>LIMS</i>	<i>Laboratory Information Management System</i>
MAPA	Ministério da Agricultura, Pecuária e Abastecimento
Ampaq	Associação dos Produtores de Aguardente de Qualidade
UFLA	Universidade Federal de Lavras
<i>JDBC</i>	<i>Java Database Connectivity</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
<i>IDE</i>	<i>Integrated Development Enviroment</i>
<i>JDK</i>	<i>Java Development Kit</i>
<i>JSP</i>	<i>Java Server Pages</i>
<i>URL</i>	<i>Uniform Resource Locators</i>
<i>HTML</i>	<i>Hypertext Markup Language</i>
<i>TCP</i>	<i>Transmission Control Protocol</i>
<i>IP</i>	<i>Internet Protocol</i>
<i>RFC</i>	<i>Request For Comments</i>
<i>HTTP</i>	<i>Hypertext Transfer Protocol</i>
<i>CPU</i>	<i>Central Processing Unit</i>

<i>J2EE</i>	<i>Java 2 Platform Enterprise Edition</i>
<i>DD</i>	<i>Deployment Descriptor</i>
<i>XML</i>	<i>eXtensible Markup Language</i>
<i>CSS</i>	<i>Cascading Style Sheets</i>
<i>RDBMS</i>	<i>Relational Database Management System</i>
<i>DAO</i>	<i>Data Access Object</i>
<i>UML</i>	<i>Unified Modeling Language</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>JVM</i>	<i>Java Virtual Machine</i>
<i>API</i>	<i>Application Programming Interface</i>
<i>JSTL</i>	<i>JavaServer Pages Standard Tag Library</i>

1. INTRODUÇÃO

O **Laboratório de Análises Físico-Químicas de Aguardente (LAFQA)** está situado no Campus da **Universidade Federal de Lavras (UFLA)**, e registrado pelo Conselho Regional de Química 2ª região/Belo Horizonte, Minas Gerais.

O laboratório atende à demanda de produtores de aguardente, especialmente na região do Sul de Minas auxiliando no controle de qualidade de bebida e também na emissão de boletim de registro da cachaça artesanal.

As análises químicas nesse laboratório são realizadas por profissionais do Departamento de Química da UFLA sob a coordenação da Prof.^a Dr.^a Maria das Graças Cardoso e da analista Cleusa de Fátima e Silva.

O laboratório LAFQA possui ligação com a **Associação dos Produtores de Aguardente de Qualidade (Ampaq)** que foi criada em 1988. Ela foi pioneira, estabelecendo normas de fabricação e criando um selo de qualidade, o primeiro para bebidas alcoólicas do País. Hoje 17 diferentes organizações de produtores em Minas Gerais tentam dar conta da demanda maior que a oferta. Só no estado de Minas Gerais, são produzidos cerca de 130 milhões de litros de aguardente por ano, e o consumo é de 180 milhões (MINAS FAZ CIÊNCIA, 2001).

O objetivo do laboratório é conscientizar os produtores pela qualidade do produto em nível de Estado, estabelecendo-se uma marca forte, para o atendimento do mercado interno e externo.

Atualmente, com o aumento das exigências do controle de qualidade de bebidas, o LAFQA realiza cada vez mais análises em maior quantidade de amostras. Consequentemente, torna-se necessário informatizar os processos do laboratório para garantir a qualidade, produtividade e tempos de resposta dos serviços prestados.

As ferramentas computacionais utilizadas pelos pesquisadores do laboratório não são integradas, e a maior parte dos cálculos físico-químicos são realizados à mão, ou seja, estão susceptíveis a erros causados por falta de atenção de algum técnico. Para os cálculos Físico-Químicos mais complexos é utilizado o software genérico Microsoft Excel. Os dados dos produtores não são armazenados digitalmente por computadores, essas informações estão guardadas em um caderno. Os produtos inflamáveis estão por toda parte e o risco de incêndio no laboratório é muito alto. Por isso o software foi criado para proteger também todas as informações do laboratório.

O objetivo deste trabalho foi desenvolver um sistema de informação Web personalizado que atenda as necessidades do LAFQA. Espera-se que o software desenvolvido, quando for implantado, permita otimizar o tempo gasto nos cálculos das análises Físico-Químicas e na emissão do laudo para o produtor, protegendo as informações assegurando que os dados obtidos não serão perdidos.

O presente trabalho encontra-se organizado da seguinte maneira: o Capítulo 2 apresenta as tecnologias e os padrões de projeto Web utilizados no desenvolvimento do software. A seguir o Capítulo 3 descreve os cálculos físico-químicos realizados pelo sistema de software construído. Os dois capítulos citados abordam o referencial teórico do presente trabalho. No Capítulo 4 é descrita a metodologia utilizada no desenvolvimento do software assim como os seus respectivos requisitos funcionais e não funcionais. Os resultados obtidos

são apresentados no Capítulo 5 assim como a discussão dos mesmos. Por fim, o Capítulo 6 conclui o trabalho realizado e apresenta também as perspectivas de trabalhos futuros. Logo em seguida as referências bibliográficas são listadas.

2. TECNOLOGIAS E PADRÕES DE PROJETO WEB DE DESENVOLVIMENTO

Os procedimentos utilizados para a pesquisa bibliográfica consistiram basicamente na realização de busca por artigos, periódicos, livros e sites na Internet relacionados ao tema.

O software foi desenvolvido utilizando-se tecnologias e padrões de projeto para desenvolvimento Web visando facilitar a sua manutenção e a sua evolução. O paradigma de programação utilizado foi orientação a objetos. Foram usados a IDE (*Integrated Development Enviroment*) NetBeans 6.7.1 e o kit de desenvolvimento Java versão 6 (*JDK – Java Development Kit*). O servidor Web utilizado no projeto foi o Tomcat versão 6.0.20.

Todo o programa foi desenvolvido utilizando a linguagem de programação Java (Sun Microsystems) com a base de dados construída a partir do **Sistema de Gerenciamento de Banco de Dados - SGBD MySQL**. Este sistema de gerenciamento é um software que incorpora as funções de definição, recuperação e alteração de dados em um banco de dados (HEUSER, 1998). Para acessar um banco de dados é necessário possuir um *driver*, que nada mais é do que um tipo de software que tem por objetivo permitir a comunicação entre dois sistemas. O *Java Database Connectivity* (JDBC) é um tipo de *driver* que disponibiliza uma interface para o acesso a dados em um banco de dados relacional. É considerada a forma mais rápida e prática atualmente para conectar uma aplicação Java a um banco de dados. O acesso aos dados do sistema Web foi feito utilizando-se o *driver* JDBC.

Para criação e compilação dos laudos foi utilizado o iReport 3.7.1. Segundo JASPER (2009) o JasperReports é uma biblioteca popular *open source*, escrita em Java. E iReport é uma ferramenta visual para gerar relatórios que é compatível com o JasperReports.

Este Capítulo abordará as tecnologias voltadas para o desenvolvimento de aplicações Web que têm mudado constantemente. Como sabemos, inicialmente, os sites possuíam apenas conteúdo estático, ou seja, o conteúdo de uma página não podia ser modificado em tempo de execução. Depois, os sites passaram a oferecer páginas com conteúdos dinâmicos e personalizados.

Diversas tecnologias estão envolvidas no desenvolvimento das aplicações Web, por exemplo, o *Hypertext Transfer Protocol* (HTTP), *JavaServer Faces* (JSF), *Java Server Pages* (JSP) e *Servlets*.

O *software* foi desenvolvido utilizando-se a tecnologia JSF na sua versão 1.2, parte integrante da tecnologia do mundo Java *Enterprise Edition Java EE*.

A tecnologia JSF utiliza o protocolo HTTP para o processo de requisição e resposta entre cliente e servidor. A *Application Programming Interface API Servlet* fornece também outras opções, como segurança, registro de ciclo de vida, eventos, filtros, distribuição e assim por diante. Esses recursos todos formam a base do JSF.

É importante destacar que o JSF é implementado como um *servlet*, e como tal será executado por um container Web, como o Tomcat, por exemplo (KITO, 2005). Esse *servlet* nos permite escrever código orientado a componentes nas páginas, que conterão também todas as facilidades do JSP (*JavaServer Pages Standard Tag Library* JSTL, *JSP core tags*, *taglibs* e outros). Todas as tecnologias citadas serão detalhadas nas seções a seguir.

2.1. Servidores e Clientes Web

A Web consiste em milhões de clientes e servidores, conectados através de redes cabeadas e sem fio como as redes *wireless*. Quando um determinado usuário acessa uma determinada *Uniform Resource Locators* (URL) pelo navegador (o aplicativo cliente), uma solicitação é enviada para o servidor Web (o aplicativo servidor). O servidor normalmente responde para o cliente enviando a página Web solicitada (DEITEL, 2004).

Um servidor Web recebe uma solicitação e devolve algo para o cliente. O *browser* ou navegador permite ao usuário solicitar um recurso. O servidor recebe a solicitação, encontra o recurso e devolve a resposta para o cliente. Um recurso pode ser uma página Web, figura, imagem ou vídeo. O cliente solicita um recurso através da URL e o servidor o disponibiliza. Se o recurso solicitado não for encontrado pelo servidor uma página de erro “*404 Not Found*” é retornada como resposta para o cliente. O processo de solicitação e resposta pode ser visualizado pela Figura 2-1.

Segundo Basham (2008, pág. 5) “o cliente é o browser fazendo aquilo que o usuário solicitou”. O processo de uma resposta do servidor ao cliente é mostrada pela Figura 2-3.

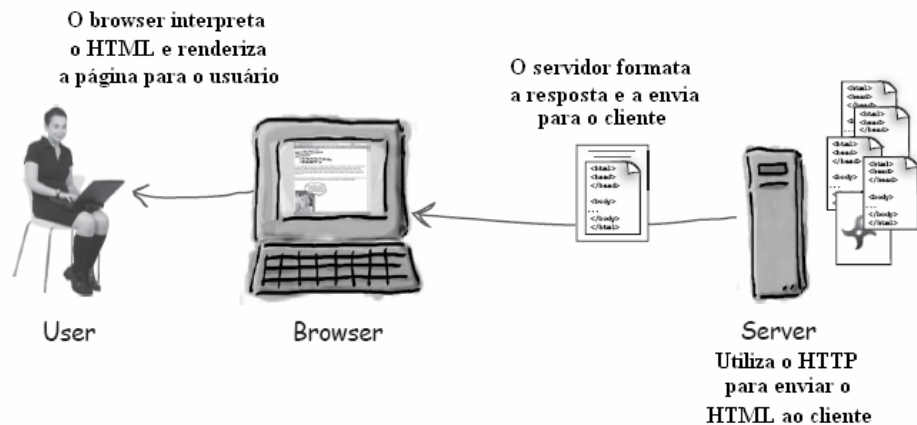


Figura 2-3 – Processo detalhado de uma resposta (Fonte: BASHAM et al, 2008)

Segundo Kurose (2003) a definição para a palavra protocolo é um conjunto de regras e convenções que permite a troca confiável de informação entre dois ou mais dispositivos de comunicação de dados.

Na Web os clientes e servidores se comunicam através do protocolo HTTP (*Hypertext Transfer Protocol*). O TCP (*Transmission Control Protocol* – Protocolo de Controle de Transmissão) e o IP (*Internet Protocol* – Protocolo da Internet) são conhecidos coletivamente por TCP/IP. A definição do protocolo TCP pode ser encontrada no *Request For Comments* ou RFC 793. Dentre os serviços que o TCP disponibiliza podemos citar o serviço da internet orientado a conexão, ou seja, a aplicação pode confiar que todos os dados serão entregues sem erro e na ordem certa. Já o protocolo IP especifica o formato da informação enviada e recebida entre os roteadores, que representam os sistemas intermediários de comunicação, e os hospedeiros como os computadores

pessoais tradicionais, servidores, Web TV's, computadores portáteis, *paggers* e até mesmo torradeiras (KUROSE, 2003).

O HTTP é executado no topo das camadas e possui características específicas para Web, mas ele depende do protocolo TCP/IP para obter a solicitação e a resposta completa de um lugar para o outro.

O Tomcat representa um exemplo de servidor Web que processa solicitação HTTP. O Mozilla é um exemplo de *browser* que oferece ao usuário o mecanismo para realizar solicitações HTTP e visualizar os dados retornados do servidor.

O HTML diz ao *browser* como ele deve exibir o conteúdo da página para o usuário final. Quando um servidor Web responde a uma solicitação ele envia código HTML através do protocolo HTTP, o browser conhece o protocolo HTTP e interpreta as instruções HTML que estão contidas dentro da resposta. O principal papel do protocolo HTTP é fornecer um ambiente de comunicação (requisição/resposta) entre servidores e clientes Web para a troca de informações codificadas em HTML. Resumindo o conteúdo do HTML pode ser considerado como um dado que foi inserido dentro de uma resposta HTTP.

O processo completo de requisição e resposta pode ser visualizado pela Figura 2-4.

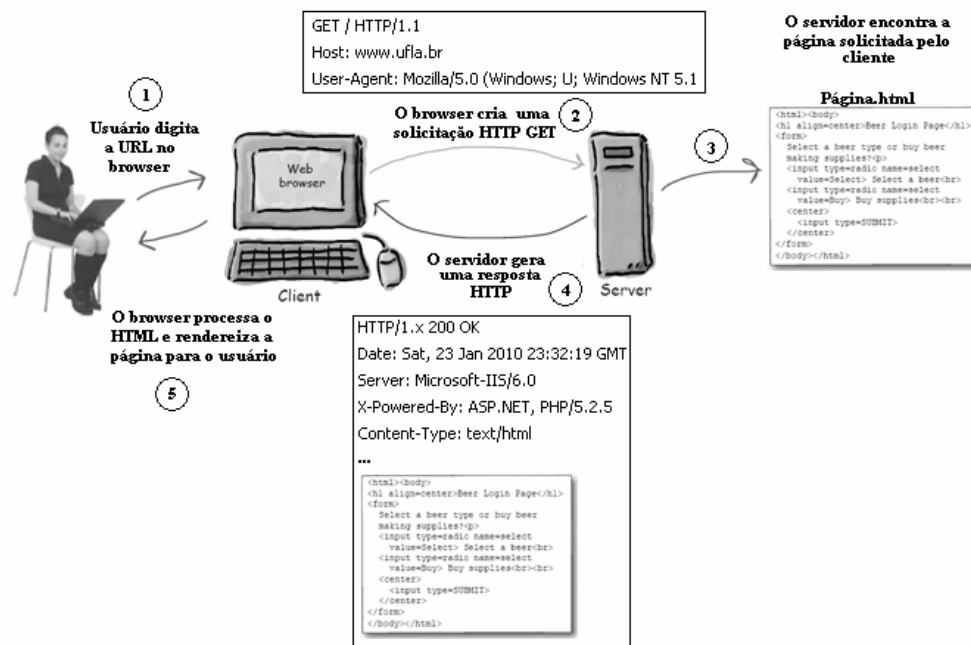


Figura 2-4 – Processo de requisição e resposta HTTP (Fonte: BASHAM et al, 2008)

Os cabeçalhos do HTTP da Figura 2-4 foram retirados do site da Universidade Federal de Lavras (UFLA) no dia 23 de Janeiro através do *plugin LiveHTTPHeaders* versão 0.15 do *browser Mozilla Firefox*.

2.2. Tecnologias de Desenvolvimento Web

O *JavaServer Faces JSF* é um *framework*¹ utilizado para desenvolvimento de aplicações Web, que executam do lado do servidor, desenvolvidas pela Sun Microsystems atualmente comprada pela empresa Oracle. Essa tecnologia faz parte da plataforma J2EE (*Java 2 Platform Enterprise Edition*) que oferece um conjunto de tecnologias e soluções robusta para a Web.

Segundo Kito (2005), o JSF é executado do lado do servidor em um container (detalhado na seção 2.3) Web padrão como, por exemplo, o Tomcat.

¹ Conjunto de classes que cooperam entre si permitindo construir projetos reutilizáveis para uma determinada categoria de software (GAMMA, 2000).

Quando o usuário clica em um botão em uma aplicação *swing*² ele vai disparar um evento que pode ser tratado diretamente no código do programa *desktop*. Em contraste, os navegadores Web não conhecem os componentes do JSF ou eventos, pois o *browser* só sabe interpretar HTML (*HyperText Markup Language*). Então quando um botão é clicado pelo usuário em um aplicativo *Faces* é gerada uma requisição que será enviada através do *browser* para o servidor pelo protocolo HTTP. O *Faces* é responsável por traduzir essa requisição em um evento que será processado pelo aplicativo no servidor. É responsável também por garantir que cada elemento da interface do usuário definidas no servidor seja exibida corretamente para o navegador.

O objetivo principal do JSF é tornar o desenvolvimento Web mais rápido e fácil. Permite aos desenvolvedores pensar em termos de componentes, eventos, *backing beans* ou *Managed Beans*. *Managed Beans* são *Java Beans*³ que podem ser usados para gerenciar melhor o comportamento dos componentes em uma página JSP. Na prática você pode criar *backing beans* diferentes para um ou mais componentes em uma página. O JSF permite pensar em estruturas de “alto nível” e suas interações, ao invés de requisição e resposta. A complexidade do desenvolvimento Web não é visualizada pelos programadores, permitindo que os desenvolvedores se concentrem mais na lógica de negócio da sua aplicação (KITO, 2005).

No JSF o controle é composto por um *servlet* denominado *FacesServlet*, por arquivos de configuração e por um conjunto de manipuladores de ações e observadores de eventos. O *FacesServlet* representa o estado da solicitação HTTP atual e é responsável também por receber requisições da Web, redirecioná-las para o modelo e então remeter uma resposta. Os arquivos de configuração são responsáveis por realizar associações e mapeamentos de ações e pela definição de regras de navegação. Os manipuladores de eventos são

² *Application Programming Interface* (API) do Java para interface gráfica.

³ Classe Java que possui métodos *Get's* e *Set's*.

responsáveis por receber os dados vindos da camada de visualização, acessar o modelo, e então devolver o resultado para o *FacesServlet* (KITO, 2005).

O *Java Server Pages* JSP nada mais é do que páginas HTML que incluem código Java e outras *tags*⁴ específicas. O JSP é uma extensão da tecnologia de *servlet* além de oferecer independência de plataforma, desempenho aperfeiçoado, separação de lógica de exibição, facilidade de administração, extensão empresarial e, mais importante, facilidade de uso (KURNIAWAN, 2002). A parte estática de uma página não precisa ser gerada, elas são codificadas na própria página JSP e o conteúdo dinâmico é gerado através do código JSP.

Quando uma página JSP é carregada pela primeira vez no container JSP, o código Java é compilado gerando um *servlet* que, ao ser executado, gera uma página HTML que será enviada para o navegador. As chamadas subseqüentes são encaminhadas diretamente ao *servlet* gerado na primeira requisição. A partir desse momento não haverá mais o processo de geração e compilação do *servlet*.

A Figura 2-5 mostra as etapas de execução de uma página JSP quando carregada pela primeira vez. Na primeira etapa a requisição é apenas enviada para o servidor Web e na segunda etapa o servidor apenas encaminha a solicitação para o container *Servlet/JSP/JSF*. Na terceira etapa o container verifica se já existe uma instância do *servlet*, caso contrário a página JSP é convertida em um *servlet* pelo container e na quarta etapa o container compila o *servlet* gerado através do JSP e logo em seguida o container instancia e executa o *bytecode* (código intermediário) gerado pela compilação do *servlet*. E na última etapa o container retorna a página de resposta para o cliente.

⁴ Estruturas utilizadas pelo navegador para exibir as páginas Web's.

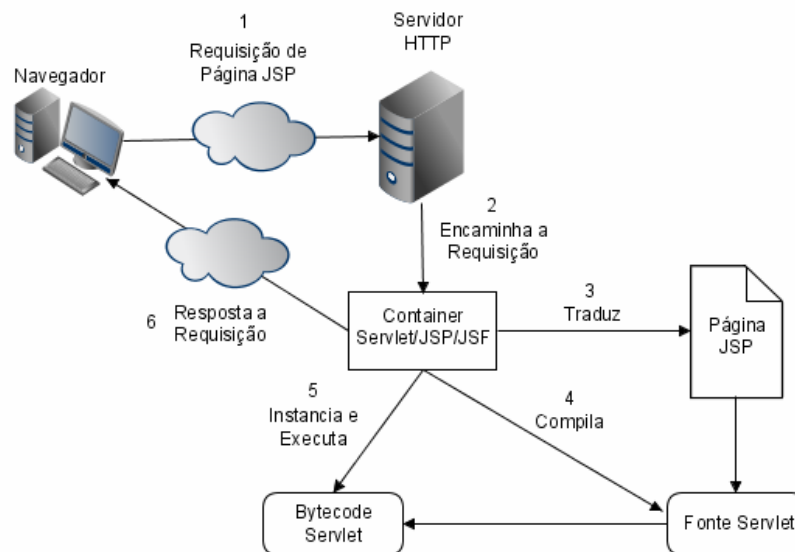


Figura 2-5 – Etapas da primeira execução de uma página JSP

Um *servlet* é uma classe Java que pode ser automaticamente carregada e executada por um *Container*. O *container* gerencia as instâncias dos *servlets*, além de prover os serviços de redes necessários para as requisições e respostas (KURNIAWAN, 2002). Para entender o funcionamento básico de um *servlet* é preciso conhecer o significado de processos e de *threads*, Silberchatz (2004) define o processo como um programa em execução, ou seja, é a unidade de trabalho em um sistema moderno de tempo compartilhado. O autor ainda afirma que uma *thread* é uma unidade básica de utilização de Unidade Central de Processamento (CPU) que possui um identificador id, um contador de programa, um conjunto de registradores e uma pilha. Um processo tradicional possui uma única *thread* de controle. Se o processo tiver múltiplas *threads* de controle, ele poderá executar mais de uma tarefa ao mesmo tempo.

Existe apenas uma instância de cada *servlet*, mas o container pode criar várias *threads* permitindo que apenas uma instância de um *servlet* possa atender

a mais de uma requisição simultaneamente. A Figura 2-6 fornece uma visão do relacionamento destes componentes.

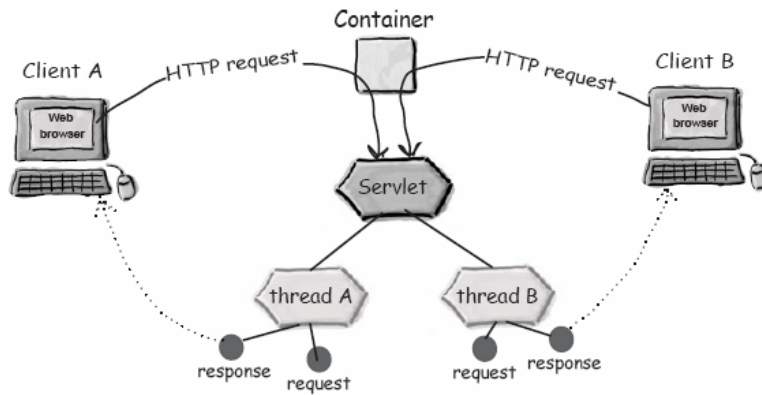


Figura 2-6 – Relacionamento entre *Servlets*, *Container* e servidor Web
(Fonte: BASHAM et al, 2008)

A criação de *threads* não sobrecarrega o servidor como a criação de processos, além de permitir uma melhora no tempo de resposta. Os *servlets* podem trabalhar com qualquer tipo de servidor e não apenas com servidores Web, uma vez que a API do *servlets* não possui ligação com o ambiente do servidor sendo independente de protocolos e plataformas. Os *servlet* são utilizados para a construção de aplicações dinâmicas na Web, pois a maior parte das páginas Web é construída no momento do atendimento de uma requisição HTTP, possibilitando criar páginas com conteúdo variável, de acordo com o usuário, tempo ou informações armazenadas no banco de dados.

2.3. A arquitetura da aplicação Web

Os *Servlets* não possuem um método *main()*. São gerenciados por uma outra aplicação Java chamada Container. O Tomcat é um exemplo de Container desenvolvido sob a plataforma Java que utiliza uma Máquina Virtual Java JVM para gerenciar as instâncias dos servlets. Este servidor implementa a API Servlet fornecida pelo projeto jakarta (SUN, 2002). Esta implementação serve como referência pela Sun Microsystem e foi utilizada para o desenvolvimento do software. Quando uma aplicação Web recebe uma solicitação para um *servlet*, ao contrário de uma página estática, o servidor entrega a solicitação para o container, que se encarrega de entregar ao *servlet*, a requisição e resposta HTTP e chamar os métodos do *servlet*.

Através da Figura 2-7 é possível visualizar como um container processa uma solicitação HTTP. Um usuário clica em um link que contém uma URL para um *servlet*. Na primeira etapa a solicitação é enviada para o servidor, e na segunda etapa o container percebe que é uma requisição para um *servlet* e cria dois objetos um *HttpServletResponse* e um *HttpServletRequest*. Na terceira etapa o container encontra o servlet correto a partir da URL, cria uma *thread* para a requisição passando os dois objetos criados. Na quarta etapa o container aciona o método *service()* do *servlet*, a partir desse momento o método *service()* pode chamar o método *doGet()* ou o método *doPost()* depende da requisição feita, considerando que a requisição feita no exemplo foi um HTTP GET, o método *doGet()* será acionado na quinta etapa, ele será responsável por gerar dinamicamente a página de resposta e inserir-la no objeto resposta. E na última etapa o *thread* termina, o container converte o objeto resposta em uma resposta HTTP enviando de volta a resposta para o cliente e logo em seguida apaga os objetos requisição e resposta.

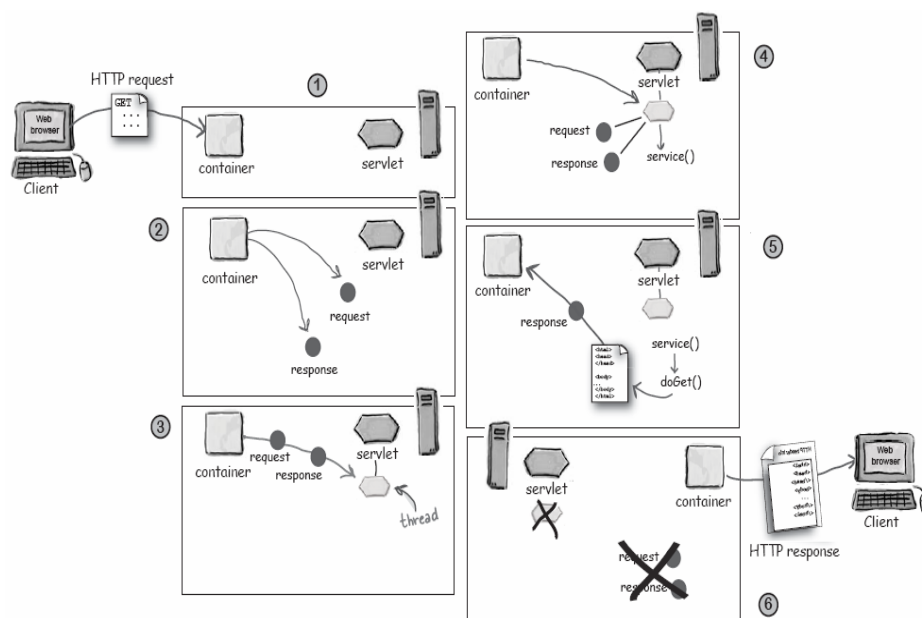


Figura 2-7 – Processamento de uma solicitação pelo container (Fonte: BASHAM et al, 2008)

O container oferece uma maneira simples para os *servlets* se comunicarem com o servidor, controla o ciclo de vida de um *servlet* e gerencia os recursos do *servlet*, sendo responsável por criar automaticamente uma nova *thread* em Java para cada requisição do *servlet* recebida. Quando o *servlet* conclui a execução do método de serviço HTTP para a solicitação daquele cliente, a *thread* é finalizada. Através de um *Deployment Descriptor (DD)*⁵ XML (*eXtensible Markup Language*) pode ser usado para configurar a segurança, sem precisar alterar código Java com a vantagem de não precisar recompilar os arquivos fontes.

Além de gerenciar *threads*, fazer a conexão via *sockets*⁶ e criar um *listener*⁷, para o *socket* o container também faz o gerenciamento da memória ele transforma os códigos JSP em código Java.

⁵ Arquivo de configuração do container.

⁶ Estruturas que permitem que as funções de software se interconectem.

⁷ Objeto que reage a um evento.

2.4. Padrões de Projeto

Segundo Gama (2000), um padrão de projeto são as descrições de objetos e classes que se comunicam e precisam ser personalizadas para resolver um problema geral de projeto num contexto particular. A essência de um padrão é uma solução reutilizável para um problema recorrente.

Gamma (2000) afirma que um padrão de projeto deve nomear, abstrair e identificar os aspectos chave de uma estrutura de projeto comum, que será útil para o desenvolvimento de um projeto orientado a objetos reutilizável. O padrão de projeto identifica as classes e instâncias participantes, seus papéis, colaborações e a distribuição de responsabilidades. Cada padrão tem um foco em um determinado problema ou um tópico particular de projeto orientado a objetos. Ele descreve em que situação pode ser aplicada, se ele pode ser aplicado em função de outras restrições de projeto e as consequências, custos e benefícios de sua utilização. Quando o projeto for implementado, um padrão de projeto deve fornecer exemplos de código para ilustrar a implementação do padrão.

A escolha da linguagem de programação é importante porque influencia o ponto de vista do projetista (usuário do padrão). Essa escolha determina o que pode ou não ser implementado facilmente. Alguns padrões podem ser expressos mais facilmente em uma linguagem que em outras.

Os padrões de projeto utilizados no presente trabalho são o *Model-View-Controller* (MVC), *Data Access Object* (DAO) e o padrão de projeto *Factory* todos os padrões são descritos a seguir:

2.4.1. O padrão de Projeto *Model View Controller* MVC

A visão é a camada responsável pela apresentação. Ela recebe o estado do modelo através do controlador. Os objetos dessa camada recebem os dados de entrada do usuário que serão repassados para o controlador (FOX, 2006).

A camada do controlador interliga a camada da visão à camada de negócio ou modelo, ou seja, é a fronteira entre ambas as camadas. No *framework* JSF quem faz o papel do controlador é o *servlet FacesContext*, que retira da solicitação do usuário os dados de entrada e interpreta o que eles significam para o modelo. Obriga o modelo a se atualizar e disponibiliza o novo estado do modelo para a visão (BASHAM et al, 2008).

O modelo é a camada que abriga a verdadeira lógica de negócio. Possui as regras para obtenção e atualização do estado e fornece ao controlador o acesso aos dados, pois é a única parte do sistema que se comunica com o banco de dados (BASHAM et al, 2008).

A maioria dos *frameworks* Web usa alguma variação do padrão de projeto MVC. A Figura 2-8 detalha um exemplo de uma variação do padrão de projeto MVC.

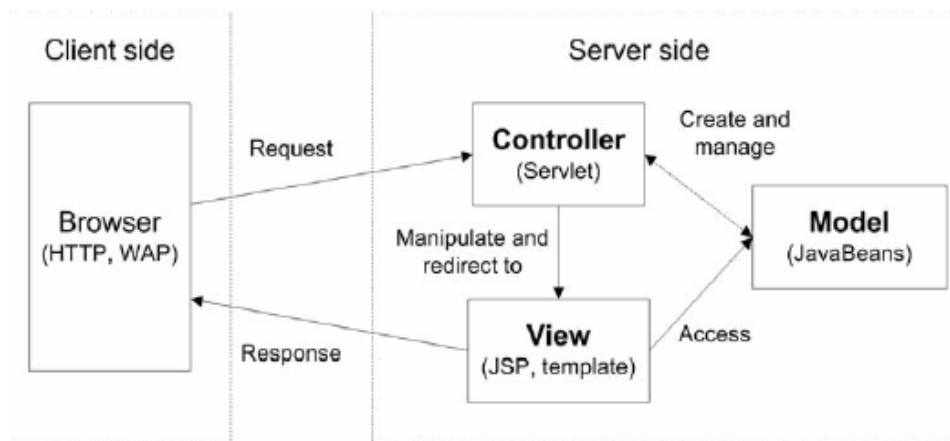


Figura 2-8 – Variação do Padrão de Projeto MVC usando *Servlet* e *JSP*
(Fonte: KITO, 2005)

2.4.2. O Padrão de Projeto *Data Access Object* (DAO)

O DAO implementa o mecanismo de acesso necessário para trabalhar com a fonte de dados. A fonte de dados poderia ser um armazenamento persistente como um *Relational Database Management System* (RDBMS), um serviço externo ou um repositório com um outro banco de dados (SUN, 2002).

O padrão de projeto DAO é usado para abstrair e encapsular todo o acesso à fonte de dados além de gerenciar a conexão com a fonte de dados para obter e armazenar os dados. É responsável por implementar o mecanismo de acesso necessário para trabalhar com a fonte de dados (SUN, 2002).

O componente de negócio que depende do DAO usa a interface mais simples expostas pelo DAO para seus clientes. O DAO oculta completamente os detalhes de implementação da persistência⁸ dos dados de seus clientes. Porque a interface exposta pelo DAO para clientes não se altera quando a implementação do código é modificada, este padrão permite que o DAO se adapte a diferentes regimes de armazenamento, sem afetar seus clientes ou os componentes de

⁸ O processo de armazenagem e captura de dados em um banco de dados é chamado de persistência.

negócio. Essencialmente, o DAO funciona como um adaptador entre o componente e a fonte de dados.

A Figura 2-9 apresenta a estrutura do padrão DAO. A classe *DataAccessObject* encapsula o acesso aos dados, é mantida pela classe *DataSource*, uma base de dados, por exemplo. A classe *BusinessObject* utiliza o objeto *DataAccessObject*. Ao utilizar esse objeto *DataAccessObject*, o objeto cliente recebe ou envia um objeto do tipo *TransferObject*. Esse objeto contém os dados a serem enviados ou trazidos da origem dos dados.

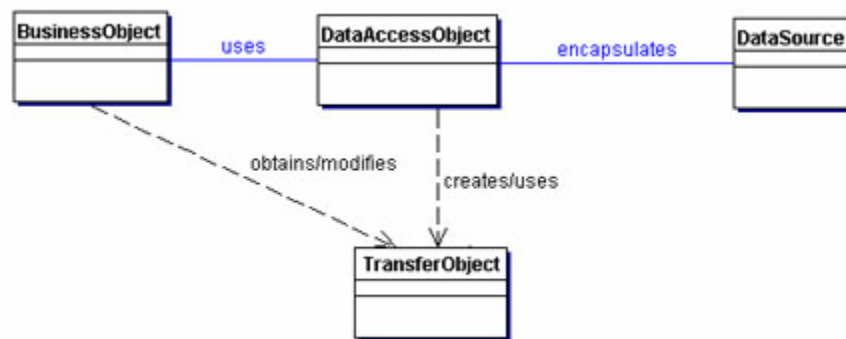


Figura 2-9 – Estrutura do Padrão de Projeto *Data Access Object* DAO
(Fonte: SUN, 2002)

2.4.3. O Padrão de Projeto *Factory*

Um dos padrões de projeto mais conhecidos e mais utilizados é o padrão *Factory*. Este padrão é responsável por criar e gerenciar o ciclo de vida de um outro objeto. A obtenção de uma conexão de dados usando uma fábrica gerenciadora de conexões ou um objeto de conexão é um exemplo do padrão de projeto *Factory* que foi utilizado no presente trabalho. A Figura 2-10 mostra um exemplo desse padrão de projeto (QUINSTREET, 2008).

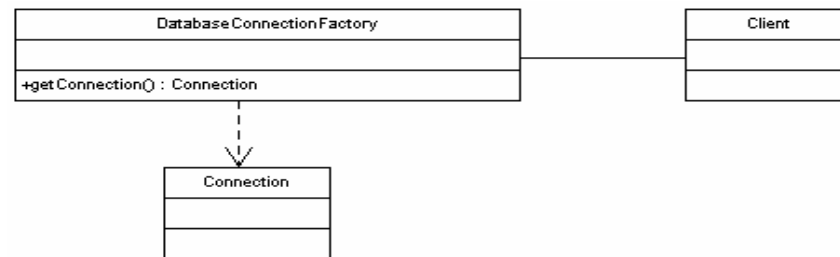


Figura 2-10 – Representação do Padrão de Projeto Factory (Fonte: QUINSTREET, 2008)

3. ANÁLISE FÍSICO-QUÍMICAS REALIZADAS EM LABORATÓRIOS

Neste capítulo apresenta-se os cálculos de cada análise Físico-Química, facilitando o entendimento da implementação do software explicada no Capítulo 4 em Metodologias.

No **Laboratório de Análises Físico-Químicas de Aguardente (LAFQA)** realiza-se análises como: exame organoléptico, grau alcoólico, densidade relativa, cobre, acidez volátil, aldeídos, ésteres, álcoois superiores, metanol e extrato seco. Atualmente, o *software* desenvolvido está em fase de integração com o sistema laboratorial e fará parte dos processos das análises constituindo-se de um sistema de software sócio-técnico (SOMMERVILLE, 2007).

As análises físico-químicas devem ser realizadas segundo metodologia estabelecida pelo **Ministério da Agricultura, Pecuária e Abastecimento MAPA** (BRASIL, 2005).

3.1. Características das Bebidas

Um produto de qualidade sempre é requisitado pelos seus consumidores. A aguardente de cana-de-açúcar é conhecida popularmente como cachaça. Os aspectos do controle de qualidade da bebida associado a sua importância exigem a realização de análises Físico-Químicas para designação da aguardente de boa qualidade.

A legislação define e registra os tipos de aguardente produzidos e comercializados no território nacional. Entre os tipos de bebida podemos citar a aguardente de cana-de-açúcar (cachaça), a aguardente de frutas, rum, tequila dentre outras mais.

O decreto Federal 2.314, de 4 de setembro de 1997, artigo 91, do MAPA, (BRASIL, 1997), estabelece os valores dos padrões de qualidade das análises de aguardente. O teor alcoólico deve ser de 38% - 54% v/v à temperatura de 20°C como padrão de leitura do grau alcoólico; a soma dos componentes voláteis (aldeídos, ácidos, ésteres, furfural e álcoois superiores) não pode ser inferior a 200 mg/100mL de álcool anidro. Posteriormente, pelo decreto nº 4.851, de 02/10/2003 (BRASIL, 2003), definiu-se que:

- **Aguardente de cana** – é a bebida com graduação alcoólica de 38% a 54% em volume, 20°C, obtida do destilado alcoólico simples de cana-de-açúcar ou pela destilação do mosto fermentado de cana-de-açúcar, podendo ser adicionada de açúcares até 6 gramas/litro.

- **Cachaça** – é a denominação típica e exclusiva da aguardente de cana produzida no Brasil, com graduação alcoólica de 38% a 48% em volume, a 20°C, obtida pela destilação do mosto fermentado de cana-de-açúcar com características sensoriais peculiares, podendo ser adicionada de açúcares até 6 gramas/litro, expressos em sacarose.

3.2. Cálculos Físico-Químicos do Laboratório

As análises descritas por Cardoso (2006) são realizadas pelo LAFQA e foram implementadas no presente trabalho. É importante destacar que as análises de Álcool Superior, Furfural e Cobre utilizam uma curva padrão que será detalhada na subseção 3.2.10 deste Capítulo. Para cada análise descrita a seguir, representamos em forma de uma caixa preta os componentes de software que as implementam. A caixa preta mostra quais são os parâmetros de entrada fornecidos do laboratório e qual a saída gerada como resposta pelo sistema. As entradas entre colchetes são fornecidas pelo próprio sistema. Os resultados serão

armazenados na base de dados e posteriormente serão usados para emissão do laudo.

3.2.1. Análise do Exame Organoléptico

O exame organoléptico é realizado observando a amostra de cachaça contra um transluminador de luz branca. Os seguintes parâmetros são observados: aspectos, coloração, limpidez, presença de corpos estranhos e vazamento. O resultado dessa análise pode ser normal ou anormal de acordo com os parâmetros analisados pelos técnico do laboratório. No software o valor de entrada dessa análise é normal ou anormal, caso o resultado for anormal as características da anormalidade serão persistidas na base de dados e posteriormente farão parte do resultado da análise do exame organoléptico.

3.2.2. Análise do Grau Alcoólico Real a 20°C(%V/V)

Valores permitidos pelo MAPA: 38 – 54 °GL (aguardente); 38-48°GL (cachaça).

$$GR = \frac{\sum(leituras)}{n} \quad (3.1)$$

A variável *leituras* descritas pela Equação (3.1) são valores de entrada obtidos através do aparelho eletrônico conhecido por alcoômetro graduado em unidades GL (Gay Lussac), onde n é o número de repetições das leituras. O sistema pode fazer no máximo até cinco repetições. O valor de *GR* é o resultado da análise do grau alcoólico. A representação da Equação (3.1) é descrita pela Figura 3-1.



Figura 3-1 – Representação da Análise Grau Alcoólico

3.2.3. Análise do Extrato Seco a 100 °C (g/L)

A análise de Extrato Seco analisa a amostra descarbonizada. Consiste na determinação gravimétrica do extrato obtido depois da evaporação lenta em banho-maria a 100°C, durante 3 horas.

$$ExtSec = (Peso_capsula_anterior(g) - Peso_capsula_posterior(g)) * 40(g/L) \quad (3.2)$$

O peso da cápsula anterior é a medida obtida através da balança analítica antes da evaporação, e o peso da cápsula posterior é a medida obtida após a evaporação, esta análise procura analisar resíduos da cachaça que não são visíveis ao olho humano. O valor do *ExtSec* é o resultado da análise do extrato seco. Os detalhes da Equação (3.2) são representados pela Figura 3-2.



Figura 3-2 – Representação da Análise Extrato Seco

3.2.4. Análise de Álcool Metílico (mg/100mL de Álcool Anidro)

O cálculo referente a análise do Metanol é dado a seguir pelas Equações (3.3) e (3.4) respectivamente apresentadas abaixo:

Onde o valor de F é dado por: $F = \frac{GR}{5}$ (3.3),

$$C_m = \left[\frac{(A * 2,5 * F)}{(A_p * GR)} \right] * 0,799 * 1000 \quad (3.4)$$

Onde, C_m é a concentração de metanol em mg/100mL de álcool anidro, A é a absorvância¹ da amostra que é dada pela média das leituras de absorvâncias obtidas do espectrofotômetro, F é o fator de diluição do grau alcoólico, A_p é a absorvância do Padrão. Lembrando que GR é o grau alcoólico real que é o resultado de uma outra análise, portanto, o técnico não fornece este valor como parâmetro de entrada, esse valor é reutilizado pelo sistema. Uma representação da Equação (3.4) é descrita pela Figura 3-3.

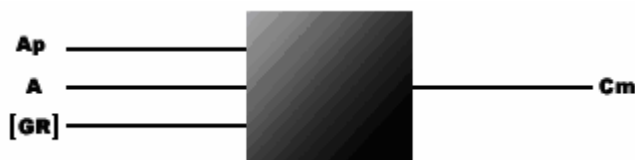


Figura 3-3 – Representação da Análise Álcool Metílico

3.2.5. Análise da Acidez Volátil em Ácido Acético (mg/100mL de Álcool Anidro)

Os cálculos referentes à análise da Acidez Volátil são:

¹Logaritmo decimal do quociente entre as intensidades das radiações, incidente e transmitida.

$$N = 0,1 * F_c \quad (3.5)$$

$$A_v = \frac{(E_q * n * N)}{10 * V} \quad (3.6)$$

$$A_{v.a.a} = \frac{(A_v * 100 * 1000)}{GR} \quad (3.7)$$

Onde, N é a normalidade da solução de hidróxido de sódio, F_c é o fator de correção, A_v é o valor do resultado da Acidez Volátil, n é o volume de hidróxido de sódio gastos na titulação em mL, E_q é uma constante e V é o volume da amostra em mL.

Na Equação (3.7) $A_{v.a.a}$ é o resultado expresso em mg/100mL de Álcool Anidro e GR é o resultado da análise do grau alcoólico calculado anteriormente. A representação dos parâmetros de entrada e saída das Equações (3.5), (3.6) e (3.7) é representada pela Figura 3-4.

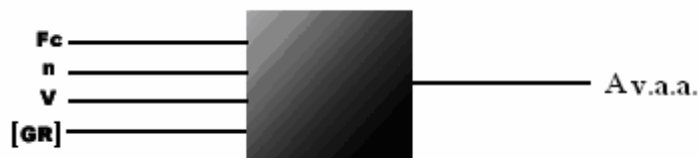


Figura 3-4 – Representação da Análise Acidez Volátil

3.2.6. Análise de Aldeídos em Aldeídos Acético (mg/100mL de Álcool Anidro)

Os cálculos referente a análise de Aldeídos são:

$$N = 0,05 * Fc \quad (3.8)$$

$$Ca = \frac{(4,4 * n * N)}{GR} * 1000 \quad (3.9)$$

Onde N é a normalidade da solução de iodo, F_c é o fator de correção, Ca é a concentração de aldeídos totais em mg por 100mL de álcool anidro, n é o volume da solução de iodo gastos na titulação, em mL e GR é o valor do grau alcoólico. A representação dos valores de entrada e saída descrita pelas Equações (3.8) e (3.9) é representada na Figura 3-5.

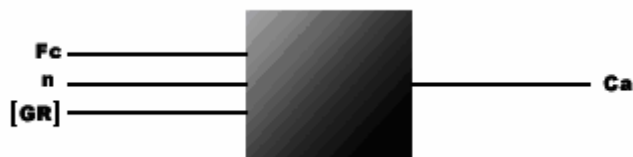


Figura 3-5 – Representação da Análise de Aldeídos

3.2.7. Análise de Ésteres em Acetato de Etila (mg/100mL de Álcool Anidro)

Os cálculos referente a análise de Ésteres são:

$$E_{aa} = \frac{(n * N * 8,8)}{V * GR} * 100 * 1000, \quad (3.10)$$

onde N é descrito pela Equação (3.5).

Onde E_{aa} é a concentração de ésteres em mg/100 mL de álcool anidro, GR é o grau alcoólico, n é o volume em mL da solução de hidróxido de sódio gastos na titulação, N é a normalidade da solução de hidróxido de sódio e V é o volume fixo de 50 mL da amostra titulada. A representação da Equação (3.10) é descrita pela Figura 3-6.



Figura 3-6 – Representação da Análise de Ésteres

3.2.8. Análise Densidade Relativa (20/20°C)

A densidade relativa é calculada através da média do número de repetições da amostra escolhida pelo usuário do sistema de software e os valores das leituras de densidade são obtidos através do densímetro eletrônico.

A Equação da Densidade Relativa é expressa da seguinte forma:

$$Dens = \frac{\left(\sum leituras\right)}{n} \quad (3.11)$$

A representação da Equação (3.11) é descrita pela Figura 3-7.



Figura 3-7 – Representação da Análise de Densidade Relativa

3.2.9. Soma dos Componentes Secundários (mg/100mL de Álcool Anidro)

A análise de soma dos Componentes Secundários é calculada através da soma dos resultados das análises de Acidez Volátil $A_{v.a.a}$, Aldeídos Ca , Ésteres Totais E_{aa} e das análises Alcoois Superiores As e Furfural F que serão detalhadas melhor nas próximas seções. A Figura 3-8 representa como entrada do software os resultados das outras análises e a saída é representada pelo resultado de SCS.

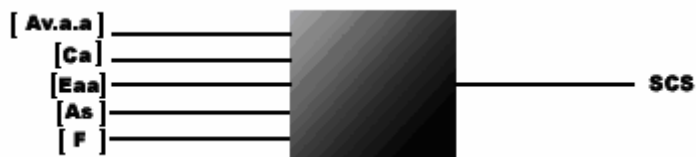


Figura 3-8 – Representação da Análise Soma dos Componentes Secundários

3.2.10. Curva Padrão

A quantificação das análises de Álcool Superior, Cobre e Furfural são realizada pela comparação dos valores de absorbância referentes às amostras com uma curva de calibração construída previamente. Esta curva é conhecida como curva padrão ela serve como uma referência para efetuar os cálculos do Álcool Superior, Cobre e Furfural.

Os cálculos estatísticos de regressão são usados para encontrar a

qualidade do ajuste da curva padrão das análises de Álcool Superior, Furfural e Cobre. O valor do R^2 é dado pelo seguinte cálculo (DRAPER; SMITH, 1998):

$$SQX = \sum X^2 - \frac{1}{n} * (\sum X)^2 \quad (3.12)$$

$$SQY = \sum Y^2 - \frac{1}{n} * (\sum y)^2 \quad (3.13)$$

$$SPXY = \sum (X * Y) - \frac{\sum X * \sum Y}{n} \quad (3.14)$$

$$\hat{b} = \frac{SPXY}{SQX}$$

$$\hat{a} = \frac{\sum Y}{n} - \hat{b} * \frac{\sum X}{n} \quad (3.15)$$

$$\hat{y} = \hat{a} + \hat{b}x \quad (3.16)$$

$$SQRL = \frac{(SPXY)^2}{SQX}$$

$$R^2 = \frac{SQRL}{SQY} \quad (3.17)$$

Onde X são os valores das concentrações do padrão. Esses valores são constantes. O Y são os valores das absorbâncias do padrão obtidos pelo aparelho espectrofotômetro.

3.2.11. Curva Padrão do Álcool Superior

Para o cálculo da curva padrão do Álcool Superior utiliza-se na Equação (3.12) os seguintes valores de X : 0.00, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30 e 0.35 para as concentrações. Através da Equação (3.13) os valores de Y ou as absorbâncias, são as entradas do software como mostrado na Figura 3-9. Lembrando que o valor de n nas Equações (3.12), (3.13), (3.14) e (3.15) é fixo e igual a 8.

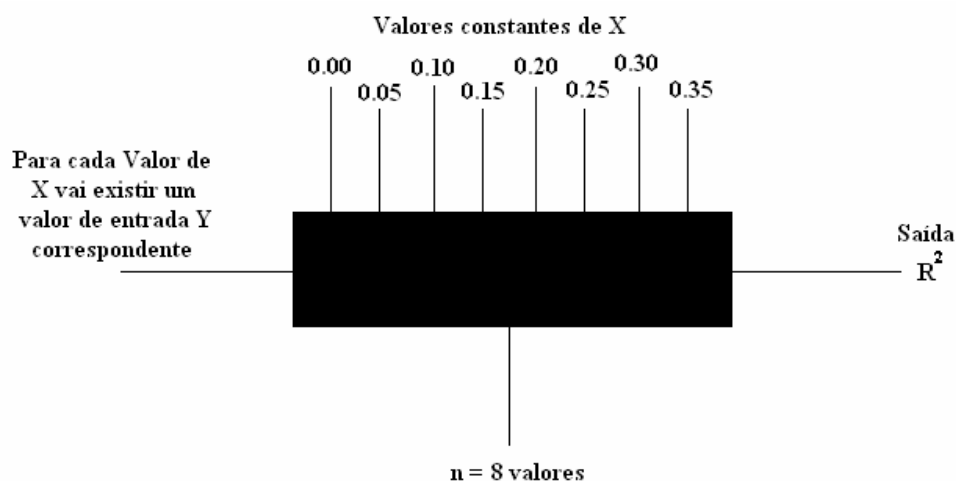


Figura 3-9 – Representação do cálculo de R quadrado

Se o valor encontrado do R^2 for maior que o que 0,99, calcula-se o valor de K_m através da seguinte Equação (3.18):

$$K_m = \frac{\sum Absorbâncias(y)}{\sum concentrações(x)} \quad (3.18)$$

Caso contrário o valor do R^2 será recalculado e então um novo valor será

usado para o cálculo do K_m . O resultado da Equação (3.18) será usado mais tarde para calcular o valor do resultado do Álcool Superior. Através da Figura 3-10 podemos visualizar uma representação real do cálculo de uma curva padrão da análise de álcool superior fornecida pelo LAFQA.

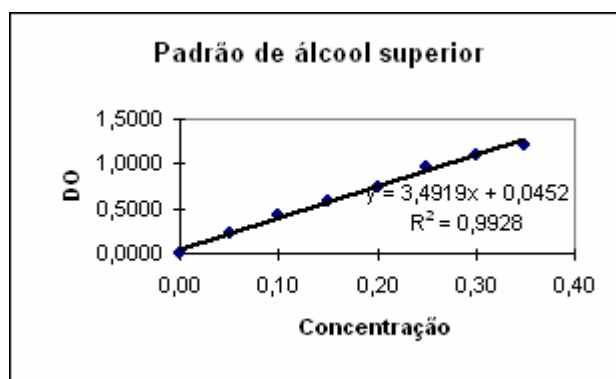


Figura 3-10 – Representação da curva padrão do Álcool Superior

3.2.12. Curva Padrão do Cobre

Para o cálculo da curva padrão do Cobre utiliza-se na Equação (3.12) os seguintes valores de X : 0.000, 0.039, 0.079, 0.118, 0.157, 0.196 e 0.236 para as concentrações. Através da Equação (3.13) os valores de Y ou as absorbâncias, são as entradas do software como mostrado na Figura 3-9. A única diferença dos cálculos dessa análise em relação a análise do Álcool Superior são os valores de X e o valor de n , que é igual a 7.

Se o valor encontrado do R^2 for maior que o que 0,99, calcula-se o valor de K_m através da Equação (3.18).

Caso contrário o valor do R^2 será recalculado e então um novo valor será

usado para o cálculo do K_m . O resultado da Equação (3.18) será utilizado para calcular o valor do resultado do Cobre. A Figura 3-11 mostra um gráfico da curva padrão do cobre.

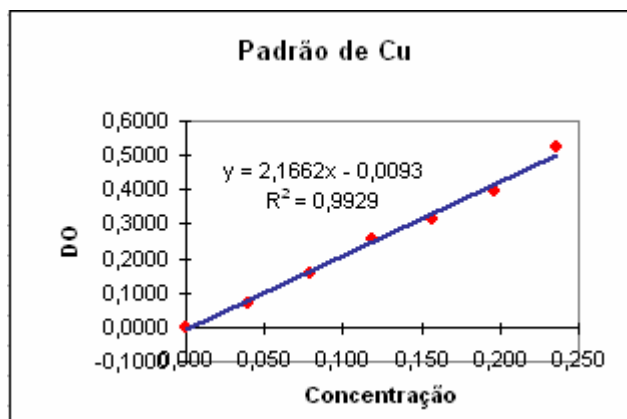


Figura 3-11 - Representação da curva padrão do Cobre

3.2.13. Curva Padrão do Furfural

Para o cálculo da curva padrão do Furfural utiliza-se na Equação (3.12) os seguintes valores de X : 0.0, 0.5, 1.0, 2.0, 3.0, 4.0 e 5.0 para as concentrações. Através da Equação (3.13) os valores de Y ou as absorbâncias, são as entradas do software como mostrado na Figura 3-9. A única diferença dos cálculos dessa análise em relação a análise do Cobre são os valores de X .

A Curva Padrão do Furfural não usa o cálculo do K_m , mas se o valor do R^2 for menor do que 0,99 os cálculos devem ser refeitos. A Figura 3-12 mostra um gráfico da curva padrão do furfural.

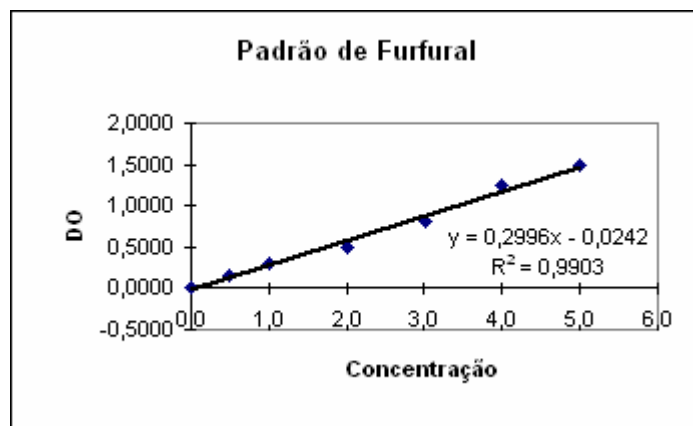


Figura 3-12 - Representação da curva padrão do Furfural

3.2.14. Análise do Álcool Superior (mg/100mL de Álcool Anidro)

Os álcoois superiores encontrados nas bebidas alcoólicas são aqueles resultantes da fermentação que dão à bebida seu sabor e buquê característico. Juntamente com ésteres, são os principais componentes flavorizantes inerentes à bebida recentemente destilada ou envelhecida em madeira.

Na análise do álcool superior o valor da concentração C é dado por:

$$C = \frac{\bar{m}}{K_m} \quad (3.19)$$

Onde \bar{m} é a média das leituras de absorbâncias obtidas através do espectrofotômetro:

$$\bar{m} = \frac{\sum \text{Absorbâncias}}{n} \quad (3.20)$$

É importante destacar a diferença das Absorbâncias usada no cálculo do K_m da Equação (3.18) e das Absorbâncias usadas na Equação (3.20), a primeira são valores padrões que não estão relacionados a nenhuma amostra e a segunda são valores obtidos da amostra que será analisada.

Pela regra de três representada pela Equação (3.21) podemos perceber que o valor de x será constante e igual a 0,1 mL destilado, onde destilado é a própria amostra destilada.

$$\begin{array}{l} 5mL \text{ destilado} \cdots \cdots \cdots 100mL \\ x \cdots \cdots \cdots 2mL \end{array} \quad (3.21)$$

$$x = 0,1$$

O valor de x é utilizado pela Equação (3.22) para calcular o valor de y .

$$\begin{array}{l} x = 0,1 \cdots \cdots \cdots C \\ 100mL \cdots \cdots \cdots y \end{array} \quad (3.22)$$

O resultado da análise do Álcool Superior é dado por As e GR é o grau alcoólico.

$$As = \frac{(y * 100)}{GR} \quad (3.23)$$

A representação dos cálculos físico-químicos pode ser visualizada através da Figura 3-13.

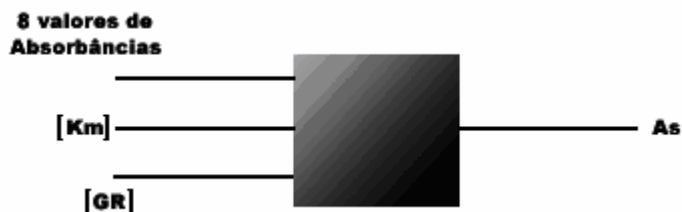


Figura 3-13 – Representação do cálculo da Análise do Álcool Superior

3.2.15. Análise de Cobre (mg/L)

Presente principalmente em cachaças produzidas em alambique de cobre. O valor máximo permitido deste elemento químico é de 5mg/L.

Os cálculos referentes à análise de Cobre são apresentados abaixo. De acordo com a Equação (3.19), temos que:

$$\begin{array}{l} C \cdots \cdots \cdots 10mL \\ x \cdots \cdots \cdots 1000mL \end{array} \quad (3.24)$$

$$\begin{array}{l} 249,5(g) \text{ CuSO}_4 \cdots \cdots \cdots 63,5(g) \text{ Cu} \\ x \cdots \cdots \cdots y \end{array} \quad (3.25)$$

CuSO_4 - Sulfato de Cobre

Cu - Cobre

O valor do K_m é dado pela Equação (3.18). O resultado da análise do cobre é dado pelo valor de y em mg/L. A representação dos cálculos físico-químicos pode ser visualizada através da Figura 3-14.



Figura 3-14 – Representação do cálculo da Análise de Cobre

3.2.16. Análise de Furfural (mg/100mL de Álcool Anidro)

A quantidade de furfural é expressa em miligrama de furfural para 100 mL de álcool anidro. Os cálculos referentes a análise do furfural são descritas a seguir:

Isolando o x da Equação (3.16) temos que:
$$x = \frac{(y - \hat{a})}{\hat{b}}$$

O valor do Furfural é dado por F , V_f é o volume de cada amostra e GR é o resultado do grau alcoólico:

$$F = \frac{(x * V_f)}{GR} \quad (3.26)$$

A representação da Equação (3.26) é descrita pela Figura 3-15 abaixo:

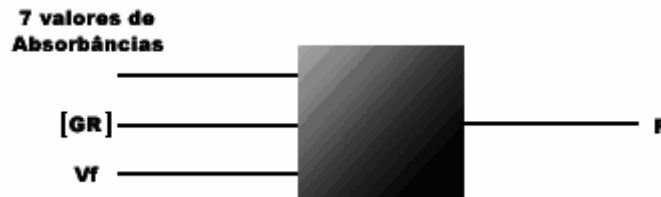


Figura 3-15 – Representação do cálculo da Análise do Furfural

No total, o software computará os cálculos de 12 análises específicas do LAFQA.

4. METODOLOGIA

A implementação do software personalizado para o Laboratório de Análises Físico-Químicas de Aguardente (LAFQA) foi realizada através do levantamento de requisitos com os próprios técnicos do laboratório, e das reuniões realizadas semanalmente. No decorrer do Projeto foram desenvolvidas versões intermediárias utilizando o modelo evolucionário (SOMMERVILLE, 2007).

Após o processo da coleta de informações do laboratório, a próxima etapa foi realizar um estudo sobre a modelagem do sistema e da sua respectiva base de dados.

4.1. Tipo da Pesquisa Realizada

Conforme Jung (2004), Marconi & Lakatos (2003) através do método científico a pesquisa realizada é de natureza tecnológica, com objetivos de caráter exploratório, utilizando procedimentos experimentais fundamentados em um trabalho de campo.

4.2. Modelo de Processo de Desenvolvimento de Software Utilizado

Um processo de software é um conjunto coerente de atividades para o desenvolvimento de software (SOMMERVILLE, 2007). Não existe um processo ideal, e várias organizações desenvolveram abordagens diferentes para o desenvolvimento de software.

Um modelo de processo são abstrações do processo que podem ser usadas para explicar diferentes abordagens para o desenvolvimento de software.

O modelo de processo utilizado no presente trabalho foi o desenvolvimento evolucionário. Segundo Sommerville (2007) esta abordagem intercala as atividades de especificações, desenvolvimento e validação. Uma versão inicial do sistema é desenvolvida baseada em especificações abstratas. O sistema é refinado com as entradas do cliente para produzir um sistema que satisfaça as necessidades do cliente.

Através da Figura 4-1 podemos visualizar melhor o modelo de processo evolucionário com uma implementação inicial, e de acordo com os comentários do usuário o refinamento do resultado é obtido por meio de várias versões intermediárias até o desenvolvimento de uma versão final do software. O *feedback* com o cliente é rápido e as atividades de especificação, desenvolvimento e validação são intercaladas (SOMMERVILLE, 2007).



Figura 4-1 – Desenvolvimento evolucionário (Fonte: SOMMERVILLE, 2007).

O sistema de informação Web foi desenvolvido utilizando-se o modelo evolucionário exploratório, onde o principal objetivo é trabalhar com o cliente para explorar os requisitos e entregar um sistema final. O desenvolvimento

começa com partes compreendidas e evolui por meio da adição de novas características propostas pelo cliente (SOMMERVILLE, 2007).

A seguir, é representado o cronograma do processo de software evolucionário utilizado no projeto. A Figura 4-2 e a Figura 4-3 apresentam as atividades desenvolvidas:

Atividades	1º Mês	2º Mês	3º Mês	4º Mês	5º Mês	6º Mês
1						
2						
3						

Figura 4-2 – Cronograma de Atividades (1º semestre)

Atividades	7º Mês	8º Mês	9º Mês	10º Mês	11º Mês	12º Mês
4						
5						
6						
7						
8						
9						
10						

Figura 4-3 – Cronograma de Atividades (2º semestre)

1. **Análise de Requisitos:** Realizada através da compreensão e definição de quais serviços são necessários e identificação das restrições do sistema.
2. **Modelagem do Sistema:** Estudo das entidades e relacionamentos do sistema.
3. **Desenvolvimento e *debugging*:** Implementação dos requisitos do sistema.
4. **Desenvolvimento e *debugging*:** Implementação dos requisitos do sistema.
5. **Fase de Teste:** Fase onde foram realizados vários teste de *debugging* de cada módulo do sistema de software.

6. **1ª versão:** Testes da primeira versão do sistema com os técnicos do laboratório.
7. **2ª versão:** Testes da segunda versão do sistema com os técnicos do laboratório.
8. **3ª versão:** Testes da terceira versão do sistema com os técnicos do laboratório.
9. **4ª versão:** Testes da quarta versão do sistema com os técnicos do laboratório.

Versão Final: Testes da versão final com o apoio dos técnicos do laboratório.

4.3. Levantamento de Requisitos

O levantamento de requisitos possibilitou conhecer e estudar os processos envolvidos em cada etapa das análises e identificar as necessidades do laboratório. Um estudo detalhado foi realizado juntamente com os técnicos, sobre as funções e os tipos de operações que o software deveria realizar e executar em cada tarefa. O software processa e executa os cálculos Físico-Químicos além de realizar diversas outras operações sobre os dados dos personagens envolvidos com o laboratório e armazena-os na base de dados.

A primeira etapa após o processo do levantamento de requisitos foi implementar os cálculos referentes as análises. Os testes foram realizados juntamente com os técnicos do laboratório com valores de entrada reais para avaliar os resultados obtidos.

A modelagem do sistema de software foi realizada através de um estudo das entidades¹ do sistema e os seus respectivos relacionamentos². Basicamente o processo de modelagem consistiu em avaliar quais tipos de dados o programa deveria armazenar e processar para retornar a saída correta para o usuário final.

¹ Conjunto da realidade modelada sobre os quais deseja-se manter informações no banco de dados (HEUSER, 1998).

² Conjunto de associações entre ocorrências de entidades (HEUSER, 1998).

A Figura 4-4 ilustra o funcionamento geral do software, assim como a interação do software com o técnico do laboratório. A rotina de funcionamento do LAFQA consiste em receber as amostras dos produtores de cachaça para a análise e a partir desse ponto o técnico registra o pedido da solicitação. Um técnico pode realizar várias solicitações de n amostras e cada amostra pode conter n análises.

Após concluir a etapa de solicitação, o técnico obtém o número do protocolo, que é o identificador da solicitação, ou seja, é o número do pedido do produtor. Na próxima etapa o laboratório inicia as análises com a ajuda de equipamentos como o espectrofotômetro. Em posse dos dados obtidos através do aparelho o técnico insere manualmente os valores no sistema de software para calcular o resultado final da análise. Este resultado é posteriormente armazenado e será utilizado pelo sistema na emissão do laudo.

O software oferece *feedback* ao produtor. Esta comunicação é feita através do Status da Análise, que pode assumir os seguintes valores: NOVA, PROCESSANDO e CONCLUÍDA. Esse status foi implementado com o intuito de fornecer informações do andamento da amostra para o seu respectivo produtor. O objetivo final do sistema computacional é gerar o laudo para o produtor com as informações armazenadas no banco de dados, mas para que isso ocorra, a regra de negócio estabelecida foi: todas as análises de uma amostra devem obrigatoriamente possuir o *status* de CONCLUÍDA, caso contrário o laudo não poderá ser gerado pelo sistema computacional.

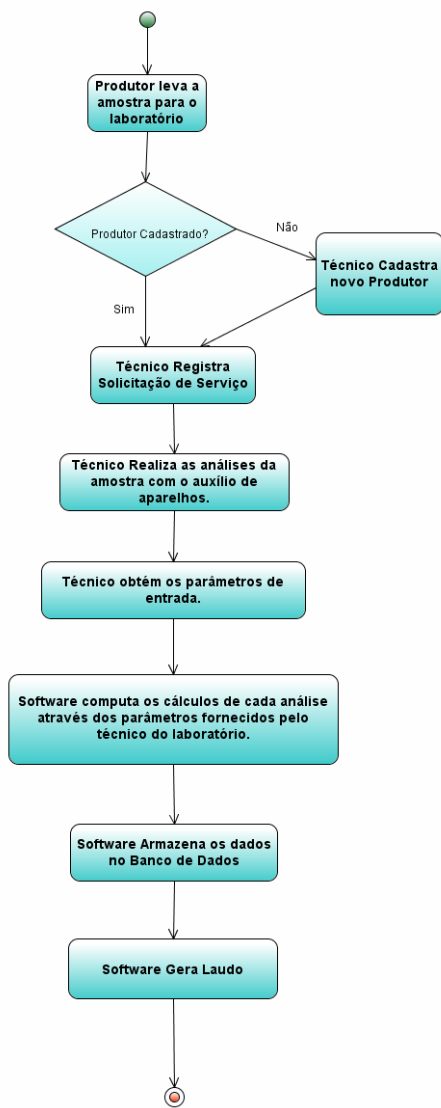


Figura 4-4 - Diagrama de atividade do funcionamento do sistema do LAFQA

A Figura 4-5 apresenta o Diagrama de Caso de Uso do software para ilustrar a interação de seus usuários. Para a representação do Diagrama de Caso

de Uso foram utilizados os elementos de modelagem sugeridos pela *Unified Modeling Language* (UML) (BOOCH *et al*, 1999).

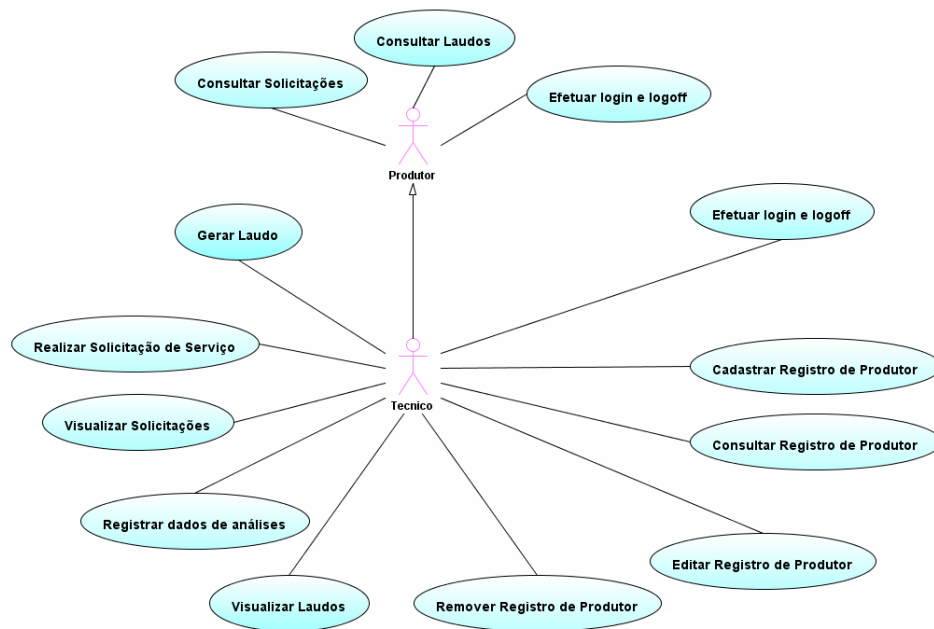


Figura 4-5 - Diagrama de Caso de Uso do Software do LAFQA

O Diagrama de caso de uso apresentado pela Figura 4-5 ilustra a interação dos tipos de usuários com o sistema de software, o técnico e o produtor. Ambos poderão acessar o sistema. Nenhum outro tipo de usuário será permitido pelo sistema, pessoas não cadastradas não conseguirão se autenticar no sistema. O técnico poderá consultar e gerar os laudos de qualquer produtor, registrar os dados da solicitação de serviço, assim como operações básicas de cadastro.

Os usuários do sistema de informação Web são classificados em dois tipos:

1) Técnico é um administrador do sistema e tem permissões para realizar as seguintes operações: i) efetuar *login* e *logoff*; ii) cadastrar, consultar, editar e remover (CRUD – *Create, Read, Update, Delete*) os dados de um técnico ou de um produtor; iii) registrar dados de amostras e análises; iv) registrar dados de uma solicitação de serviço; vi) gerar laudo; vii) visualizar os dados das solicitações e viii) visualizar laudos de todos os produtores.

2) Produtor é um usuário comum do sistema e pode apenas i) efetuar *login* e *logoff*; ii) ver o status das análises da sua amostra assim como o status dos seus laudos; iv) ver os dados de todas as solicitações efetuadas por ele.

4.4. Documento de Requisitos do Sistema

Identificação dos atores

Ator é qualquer entidade externa que de alguma forma interaja com a aplicação, seja ele uma pessoa, equipamento ou outro sistema computacional. Atores definem papéis que os usuários podem fazer (BOOCH *et al*, 1999). Com base em uma análise detalhada do Laboratório de Análises Físico-Químicas de Aguardente, chegou-se a definição dos atores descritos a seguir:

- **[AT01] Técnico:** O técnico é um administrador do sistema, responsável pela execução de análises físico-químicas e por criar e manter todos os cadastros do sistema (técnicos, produtores, solicitações, amostras, análises e laudos), dentre outras funcionalidades de acesso restrito.
- **[AT02] Produtor:** O produtor é um usuário comum do sistema com acesso restrito.

Requisitos Funcionais

Os requisitos Funcionais especificam ações que um sistema deve ser capaz de executar, sem levar em consideração restrições físicas. Os requisitos funcionais especificam, portanto, o comportamento de entrada e saída de um sistema (SOMMERVILLE, 2007).

Na Tabela 4-1 estão detalhados os requisitos funcionais (RF) ilustrados pela Figura 4-5 do diagrama de caso de uso.

Legenda:

- **Id:** Identificador numérico único;
- **Nome:** Nome único para descrever a função do caso de uso;
- **Descrição:** Descrição sucinta das atividades;
- **Ator:** ator(es) que interage(m) com o software;

Tabela 4-1: Requisitos Funcionais do Sistema

ID	Nome	Descrição	Ator
[RF001]	Efetuar <i>Login</i> e <i>Logoff</i>	A autenticação e autorização de usuários foram utilizadas com o intuito de proteger e dar maior segurança para as informações confidenciais da organização de cada produtor e das suas respectivas análises, viabilizando dessa forma a formação de novos processos.	[AT01],[AT02]

[RF002]	Consultar Laudos	Consultar <i>status</i> do laudo e o andamento das análises.	[AT01],[AT02]
[RF003]	Consultar solicitações	Consultar as informações básicas como data, hora e o número do protocolo da solicitação.	[AT01],[AT02]
[RF004]	Gerar Laudo	Após concluir todas as análises o Técnico poderá emitir o laudo das análises que será entregue para o produtor.	[AT01]
[RF005]	Realizar Solicitação de Serviço	Apenas os Técnicos podem registrar os dados da Solicitação de Serviço no sistema Web.	[AT01]
[RF006]	Visualizar Solicitação de Serviço	Visualizar dados da solicitação como o número de amostras da solicitação. Assim como informações do técnico responsável pela solicitação de serviço.	[AT01]
[RF007]	Registrar Dados de Análises	Registrar os dados obtidos como resultado de cada análise Físico-Química.	[AT01]
[RF008]	Visualizar Laudos	Visualizar todas as informações sobre os laudos dos produtores.	[AT01]
[RF009]	Remover Registro de	Excluir dados dos produtores que não possuam solicitações de	[AT01]

	Produtor	serviço cadastradas no sistema.	
[RF010]	Editar registro de Produtor	Editar ou alterar os dados dos produtores.	[AT01]
[RF011]	Consultar Registro de Produtor	Consultar os dados de qualquer produtor cadastrado no sistema.	[AT01]
[RF012]	Cadastrar Registro de Produtor	Cadastrar os dados dos Produtores como nome, endereço, CNPJ entre outros.	[AT01]

Requisitos Não Funcionais

Os requisitos não funcionais descrevem apenas atributos do sistema ou atributos do ambiente do sistema, como segurança, desempenho, usabilidade e confiabilidade (SOMMERVILLE, 2007). Na Tabela 4-2 estão detalhados os requisitos não funcionais do sistema (RNF):

Tabela 4-2: Requisitos Não Funcionais do Sistema

ID	Descrição
[RNF001]	[Segurança] Descreve os requisitos associados à integridade dos dados, privacidade, como o sistema trata de informação confidencial, liberação de acesso aos usuários do sistema.
[RNF002]	[Desempenho]

	Descreve o tempo de resposta do sistema durante o uso dos recursos disponibilizados.
[RNF003]	[Usabilidade] Descreve os requisitos não-funcionais associados à facilidade de uso do sistema.
[RNF004]	[Confiabilidade] Descreve os requisitos não funcionais associados à frequência de falha, e a robustez do sistema na recuperação destas falhas.

4.5. Projeto do Sistema de Informação Web

A essência do projeto de software segundo Sommerville (2007) é tomar decisões sobre a organização lógica do software baseando-se em experiências anteriores.

O Projeto de arquitetura é um processo criativo em que se tenta estabelecer uma organização do sistema que satisfaça os requisitos funcionais e não funcionais do sistema de software.

4.5.1. Arquitetura do Software Web

Após todo o processo de modelagem, foi estruturada a arquitetura do software Web que pode ser visualizada através da Figura 4-6 abaixo:

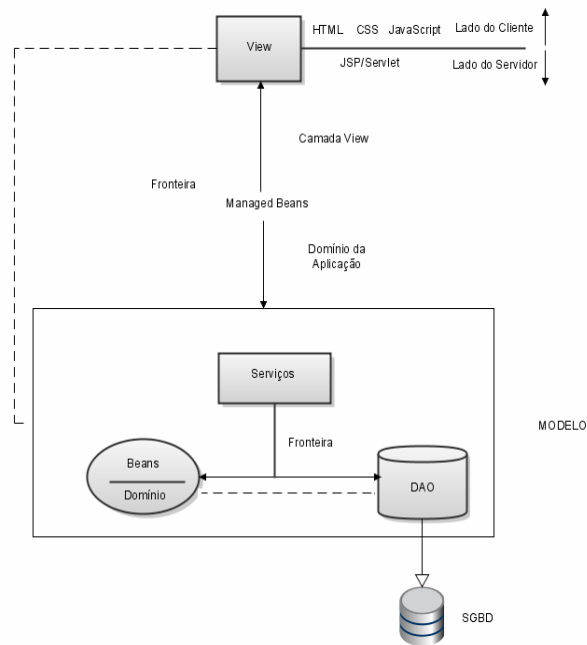


Figura 4-6 – Arquitetura do Sistema de Software Web

Através da Figura 4-6 podemos observar que a camada de visão é composta pelas telas do sistema, no lado do cliente as tecnologias utilizadas foram o HTML, CSS e JavaScript e parte dessa camada é composta também pelas páginas JSP e pelos *Servlets*. Os Managed Beans ou beans gerenciados pelo JSF fazem a fronteira entre a camada de visão e o modelo de Negócio do sistema. A camada de serviços pertence ao domínio da aplicação ela é responsável por validar os dados dos Managed Beans além de ser a fronteira entre as classes Java Beans e a camada DAO responsável pela persistência dos dados e pela comunicação direta com o **Sistema de Gerenciamento de Banco de Dados - SGBD MySQL**.

4.5.2. Estrutura de Classes das Análises Laboratoriais

Para Deitel (2004), herança é o recurso utilizado para “*criar uma nova classe a partir de uma classe existente, a nova classe herda os atributos e comportamento dessa classe existente*”. A Generalização e a Especialização foram utilizadas na estrutura das classes envolvidas. Segundo Guedes (2005) este relacionamento é uma forma de associação entre Casos de Uso na qual existem dois ou mais Casos de Uso com características semelhantes entre si. A estrutura de um Caso de Uso generalizado é herdada pelos Casos de Uso especializados, incluindo quaisquer possíveis associações.

Todas as classes da Figura 4-7 representam apenas uma entidade (Análise) no modelo de Entidade Relacionamento da Figura 4-16. Na classe Análise todos os atributos possuem os métodos Get's e Set's com exceção dos atributos resultado e curva que possuem apenas o método *get()*, pois o atributo resultado é definido de acordo com a análise a ser realizada. Um método *calcula()* foi implementado em cada classe das análises e cada método possui parâmetros específicos de entrada. O atributo curva é setado somente através das especializações de AnaliseComCurva pois é a única classe que possui uma referência à classe CurvaPadrao. Essa classe possui o método *calculaCurvaPadrao()* onde o atributo curva é setado. Através da Figura 4-7 podemos visualizar melhor a estrutura de herança utilizada no presente trabalho.

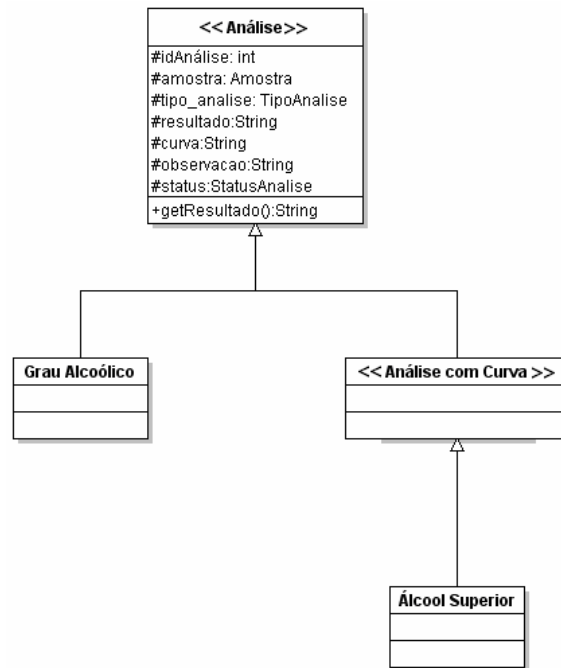


Figura 4-7 – Estrutura de herança do Projeto

O conceito de herança foi utilizado, pois todas as análises possuem o atributo **resultado** que é uma informação comum entre todas elas. Ao invés de definir o atributo **resultado** para todas as classes (gerando redundância), foi criada uma classe abstrata *Análise* onde o atributo **resultado** foi inserido dentro dela, então herdamos as propriedades para nossas classes filhas que são responsáveis por implementar o seu método *calcula()*. Todas as classes filhas herdam o método *getResultado()* da superclasse ou classe mãe *Análise*.

A classe *Grau Alcoólico* é filha da classe *Análise* e possui apenas dois atributos, são eles: o número de repetições e uma *flag* para indicar se a análise já foi concluída anteriormente e um método *calcula()* com o seguinte parâmetro: somatório das leituras descritas pela Figura 3-1 do Capítulo 3. Este método é responsável também por validar o valor do somatório recebido. Caso o valor

passado por parâmetro seja inválido, uma exceção específica será lançada. Os detalhes da classe do Grau Alcoólico são mostrados pela Figura 4-8.

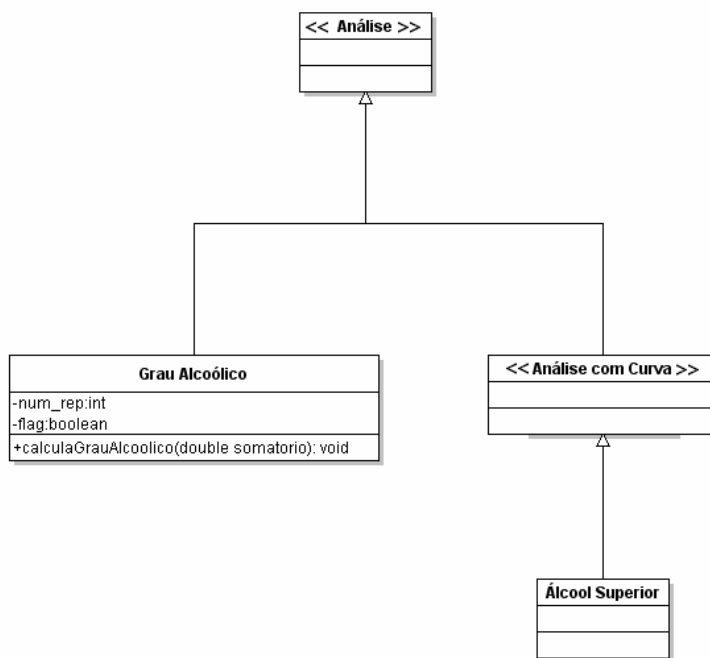


Figura 4-8 – Classe da Análise do Grau Alcoólico

Todas as análises possuem o atributo curva, que é herdado da superclasse Análise, mas apenas as análises do Álcool Superior, Cobre e Furfural utilizam esse atributo, pois para todas as outras análises que não são descendentes diretamente da classe AnaliseComCurva, o valor do atributo curva será persistido como nulo no banco de dados. A estrutura da classe Abstrata Análise Com Curva pode ser visualizada pela Figura 4-9.

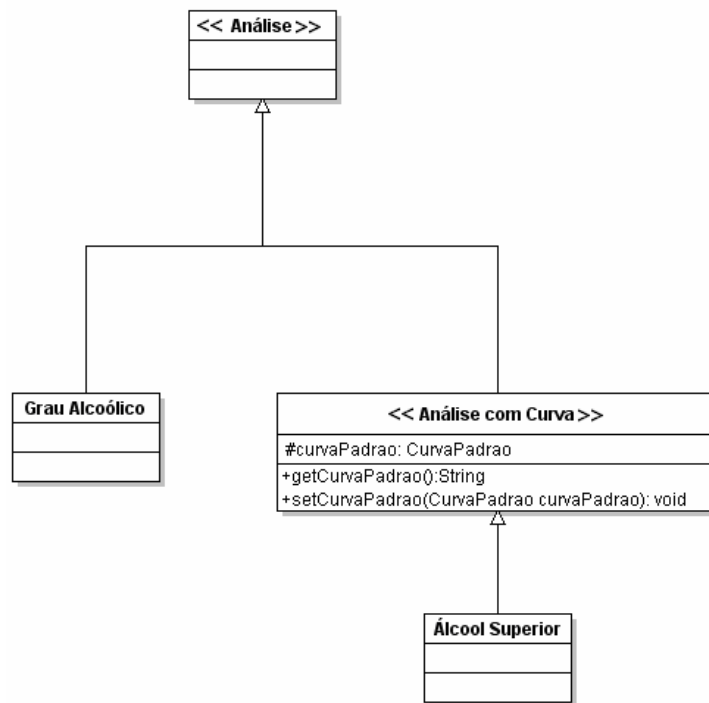


Figura 4-9 – Classe da Análise com Curva

A classe do Álcool Superior é uma especialização da classe AnaliseComCurva. Como mostrado na Figura 4-10 essa análise possui apenas um atributo constante que é utilizada nos cálculos e apenas um único método *calculaAlcoolSuperior()* com dois parâmetros: a média das absorbâncias e o valor do resultado da análise do grau alcoólico. O resultado é armazenado na variável resultado da superclasse, o método *getResultado()* da classe mãe é usado para obter o valor do resultado. Os detalhes da estrutura da classe do Álcool Superior são mostrado a seguir pela Figura 4-10.

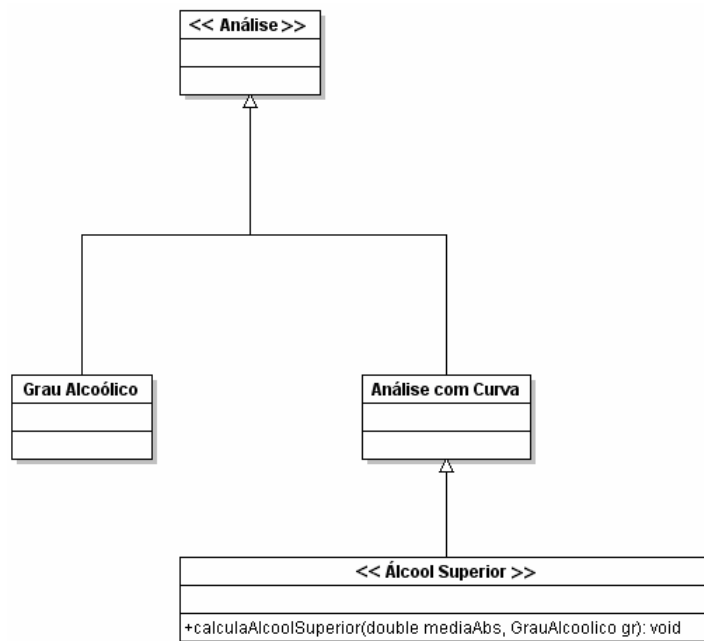


Figura 4-10 – Classe da Análise do Álcool Superior

A classe *AnáliseComCurva* possui uma agregação indicando que ela possui uma referência a classe *CurvaPadrao* que por sua vez possui uma referência a classe *Regressão*. A classe *CurvaPadrao* possui um construtor com o parâmetro: Nome da Análise. De acordo com o nome passado para o construtor, um vetor com as concentrações específicas será carregado na memória para efeito de cálculo. O método *calculaCurvaPadrao()* da classe *CurvaPadrao* retorna uma *string* com uma função do 1º grau que será armazenada no banco de dados. Essa classe possui também um método *getCurva()* que sobrescreve o método da superclasse *Análise*. A classe *Regressão* apenas realiza os cálculos descritos pelas Equações (3.12) à (3.17), a estrutura é representada a seguir pela Figura 4-11.

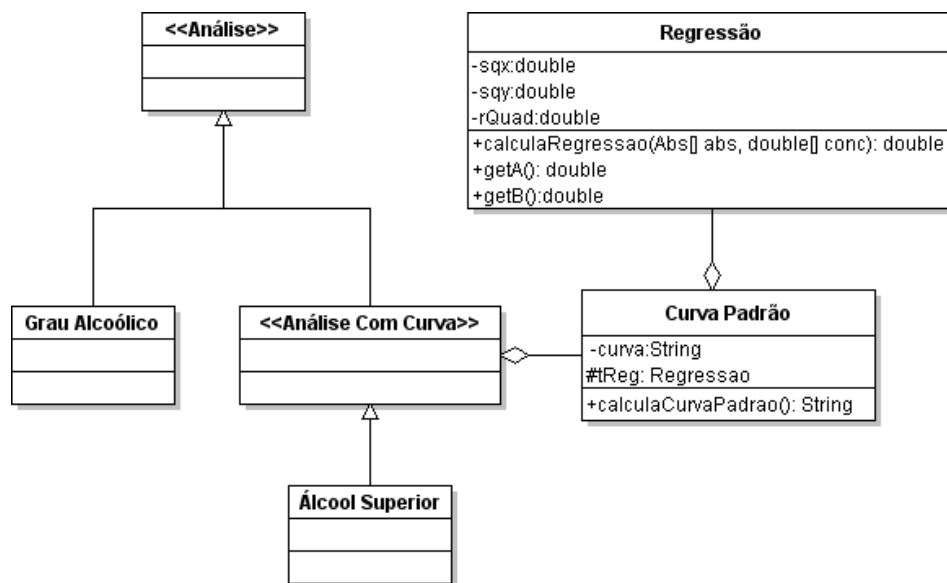


Figura 4-11 – Estrutura das classes Curva Padrão e Regressão

4.5.3. Diagramas de Seqüência do sistema Web

Um Diagrama de Seqüência descreve a maneira como os grupos de objetos colaboram em algum comportamento ao longo do tempo (SOMMERVILLE, 2007). Ele registra o comportamento de um único caso de uso, exibindo os objetos e as mensagens trocadas entre eles. A seguir, são apresentados os seguintes Diagramas de Seqüência:

- Cadastrar Produtor (*Create*) Figura 4-12;
- Editar Produtor (*Update*) Figura 4-13;
- Buscar Produtor (*Read*) Figura 4-14;
- Remover Produtor (*Delete*) Figura 4-15;

O processo de *Create* de um produtor é uma operação válida apenas para o usuário técnico. Através da Figura 4-12 é possível visualizar como um técnico cadastra os dados de um produtor no sistema.

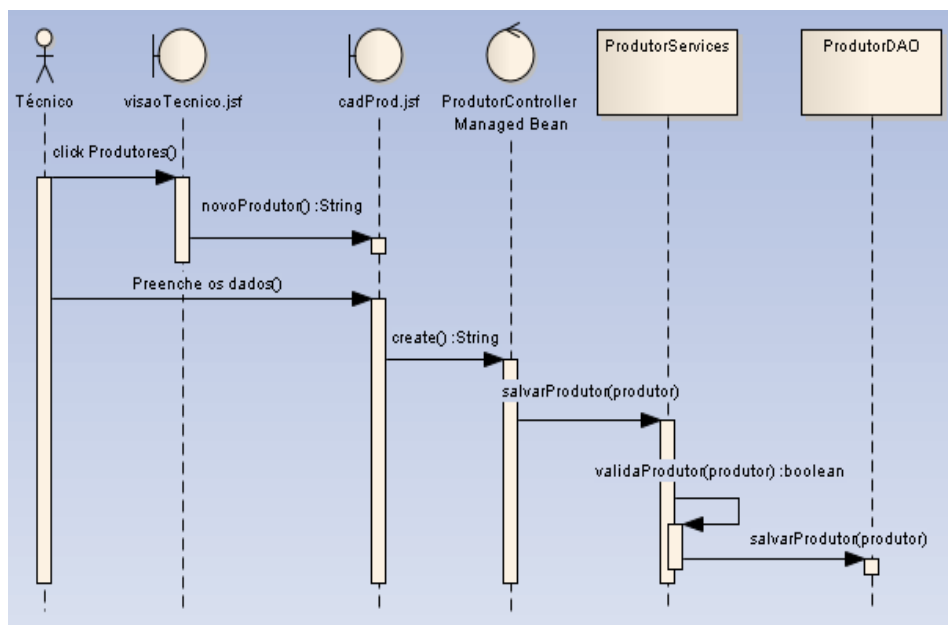


Figura 4-12 – Operação de *Create* dos dados de um Produtor

A Figura 4-13 ilustra o um processo de Update, ou seja, a atualização dos dados de um produtor feita por um técnico do laboratório.

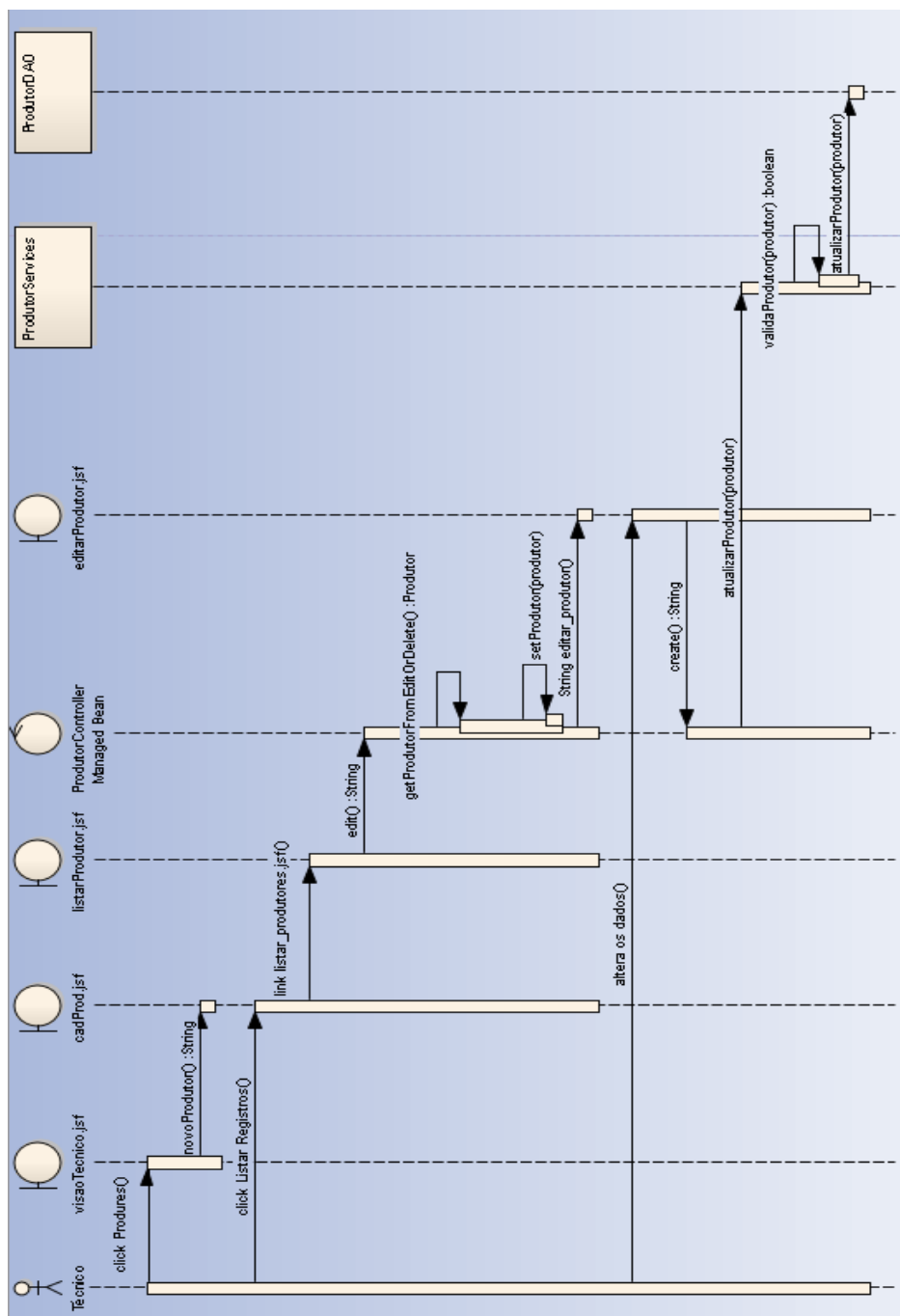


Figura 4-13 – Operação de *Update* dos dados de um Produtor

Através da Figura 4-14 podemos visualizar o diagrama de sequência que detalha o caso de uso específico de uma operação de Busca de um produtor, sendo executada por um técnico do laboratório. É Importante destacar que o método *buscarProdutores()* retorna uma lista com o nome de produtores semelhantes além de ordenar os nomes em ordem alfabética na camada de serviços.

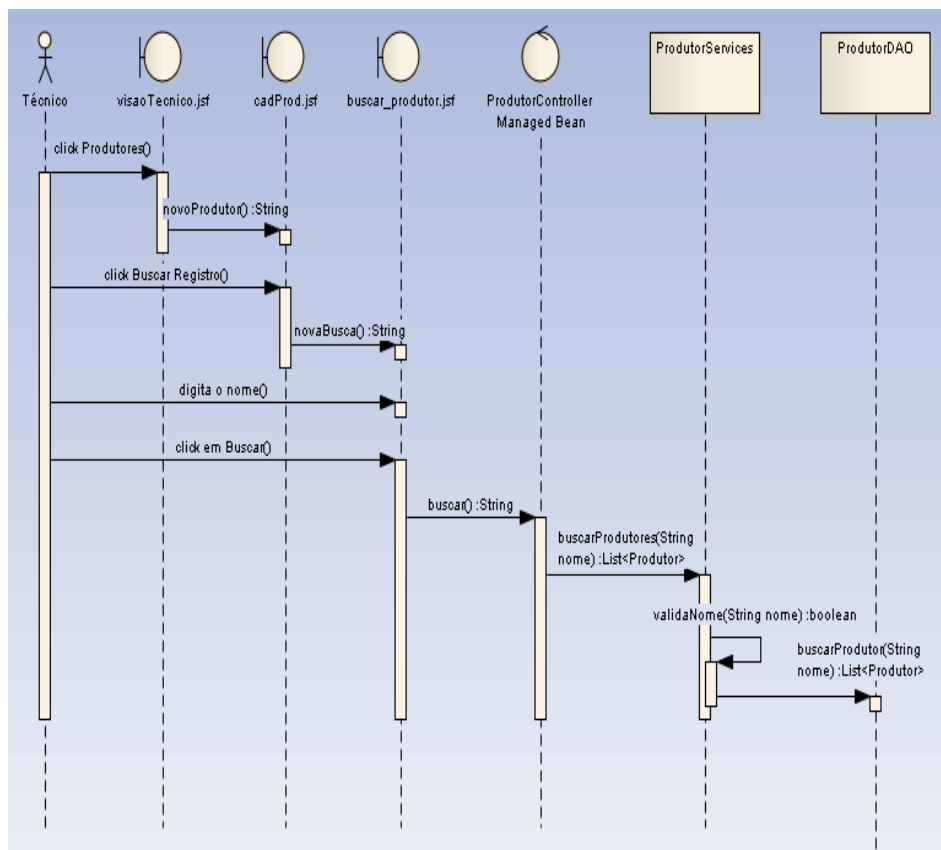


Figura 4-14 – Operação de *Read* dos dados de um Produtor

A Figura 4-15 ilustra o diagrama de seqüência para o caso de uso específico da operação de Excluir os dados de um Produtor, feita através de um técnico do LAFQA.

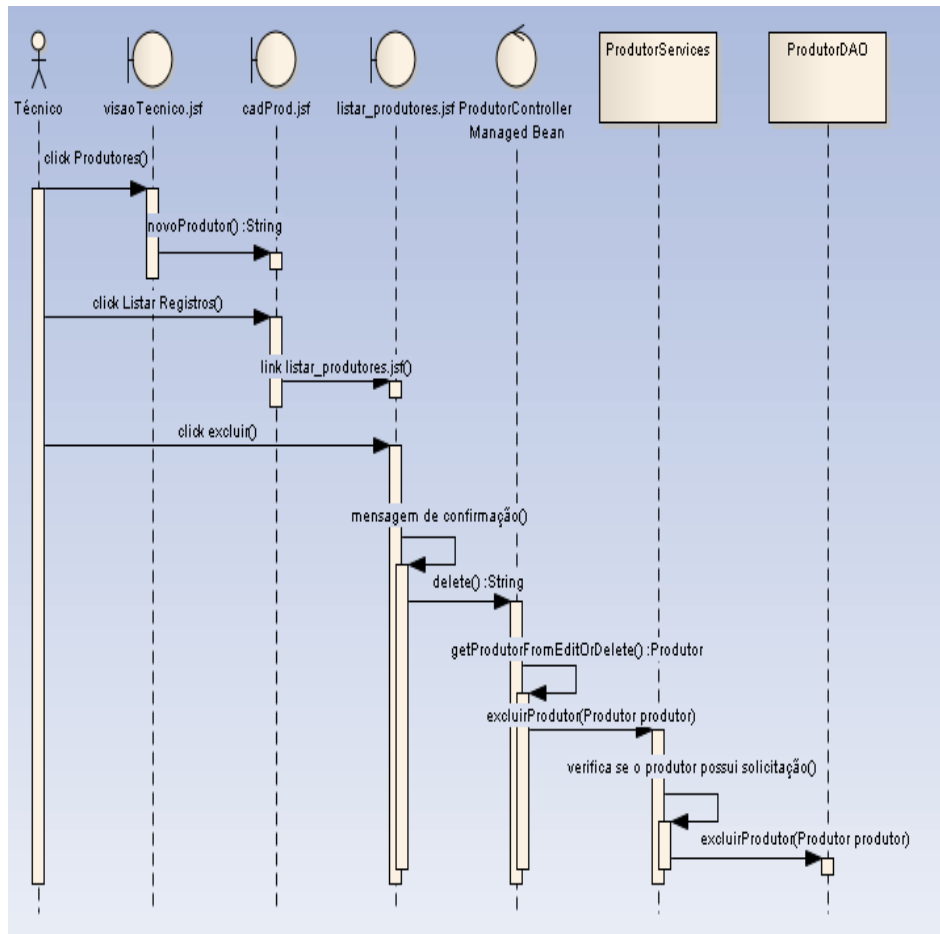


Figura 4-15 – Operação de *Delete* dos dados de um Produtor

4.5.4. Modelo de Entidade Relacionamento MER

Na Figura 4-16 está representado o diagrama de entidade relacionamento do sistema de software Web, onde podemos ver que todos os técnicos e produtores são usuários do sistema. Os técnicos registram as solicitações de serviços no sistema. Uma solicitação pode estar relacionada a várias amostras e uma amostra pode conter todo o conjunto de análises realizadas pelo LAFQA ou um subconjunto dessas análises, mas apenas um laudo de cada amostra será gerado pelo programa. Os atributos identificadores estão representados em negrito.

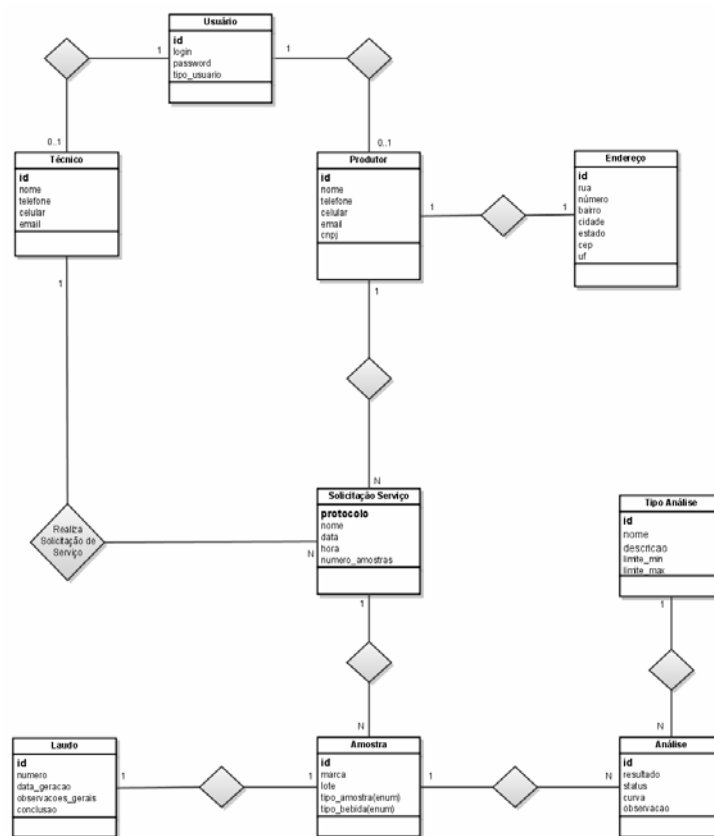


Figura 4-16 - Diagrama de Entidade Relacionamento do Software

4.5.5. Modelo de Entidade Relacionamento Lógico

O modelo Lógico do sistema Web foi obtido através do Modelo de Entidade Relacionamento (MER) da Figura 4-16. A Figura 4-17 abaixo detalha a base de dados utilizada no sistema.

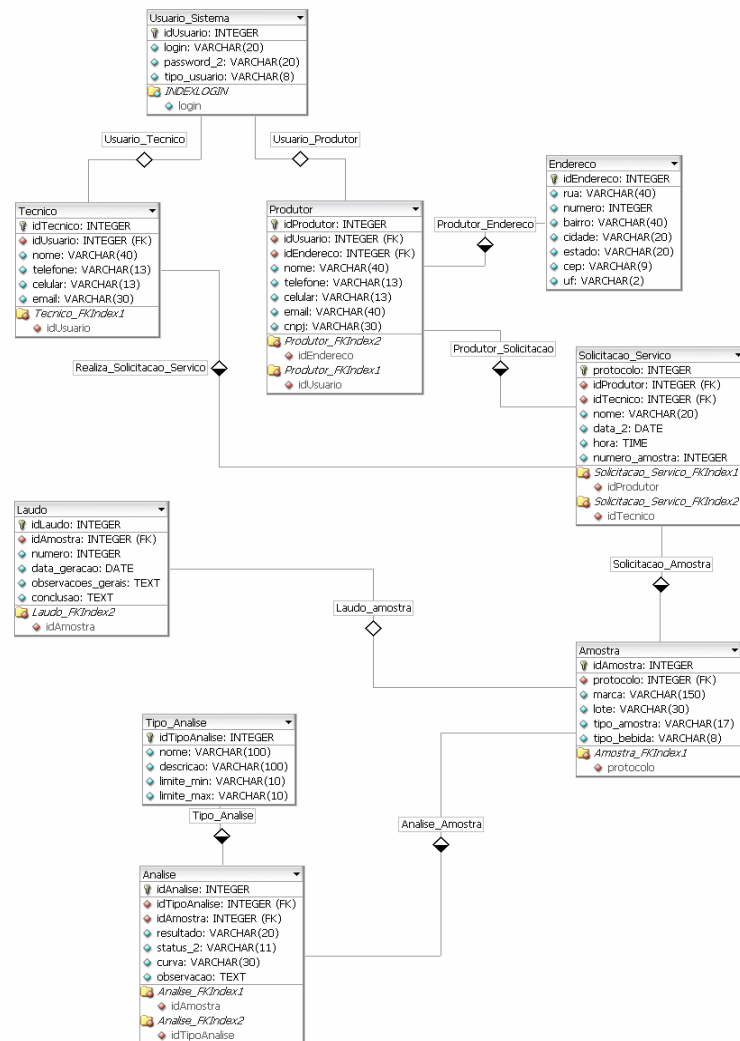


Figura 4-17 - Diagrama da Base de Dados do Sistema de Software

4.6. Implementação do Sistema de Informação Web

Após concluir todo o projeto do sistema a próxima etapa foi desenvolver ou implementar os padrões de projeto discutidos nas seções 2.4.1, 2.4.2 e 2.4.3. É importante destacar, no entanto, que a camada representada pelo modelo foi subdividida em duas outras camadas: serviços e DAO, respectivamente. A Figura 4-18 mostra as camadas do sistema computacional.

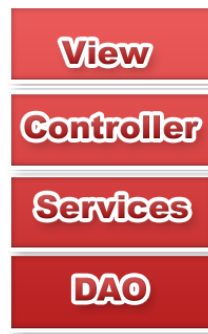


Figura 4-18 - Camadas do Sistema de Software

A visão é a camada de visualização. É nessa camada onde estão todas as páginas JSF do sistema, ela é responsável por tratar todos os eventos dos usuários, e se comunica apenas com a camada do controlador.

O controlador é a camada responsável por receber os dados de entrada da visão, além de fornecer uma resposta ao evento ocorrido. Esta camada faz a comunicação direta com a camada de serviço, responsável pela validação dos objetos. A camada de serviço interliga as camadas do controlador e do DAO além de facilitar a manutenção do sistema, pois encapsula a camada de persistência aos dados. Uma fábrica de conexões foi utilizada na implementação do software. Ela é responsável por criar e produzir conexões que serão utilizadas pela camada de serviços, essa classe é a *ConnectionFactory*. O código fonte do presente trabalho encontra-se no formato digital.

4.7. Testes do Sistema de Informação Web

O estágio de desenvolvimento de software é o processo de conversão de uma especificação de sistema em um sistema executável. É importante destacar que nessa atividade foram realizados vários testes no próprio código-fonte do sistema, pois os testes permitem encontrar os defeitos e garantir a confiabilidade do software. O teste de defeito e o *debugging* são processos diferentes. O teste permite detectar a presença de defeitos já o *debugging* está relacionado a localização e correção desses defeitos para atender aos requisitos. O processo de depuração faz parte do desenvolvimento quanto do teste de software (SOMMERVILLE, 2007).

Geralmente o desenvolvimento e o teste de componentes são intercalados. Os programadores elaboram seus próprios dados de teste e testam os códigos de forma incremental à medida em que são desenvolvidos. No presente trabalho durante o desenvolvimento do código fonte do sistema de informação Web foram realizados apenas os testes de *debugging*.

5. RESULTADOS E DISCUSSÃO

Os resultados estão no desenvolvimento da ferramenta de software Web. Espera-se que a informatização dos processos otimize as etapas anteriormente realizadas manualmente pelos técnicos. Com a utilização do programa espera-se que novos processos sejam criados em decorrência da informatização do laboratório, mas em contrapartida agilize os processos envolvido em cada etapa das análises e na emissão do laudo para o produtor.

5.1. Descrição do Software Web Desenvolvido

O software do Laboratório de Análises Físico-Químicas de Aguardente (LAFQA) possibilitará o gerenciamento das informações confidenciais do laboratório de forma segura. É importante destacar a proteção da informação, onde um produtor jamais poderá ver informações do laudo de outro produtor, pois os produtores são concorrentes de mercado.

O sistema de segurança do software foi feito através da autenticação e da autorização dos usuários do sistema. Através da tela inicial de acesso do sistema representada pela Figura 5-1, o software busca a informação na base de dados do sistema e verifica a identidade daquele usuário através de uma consulta em *Structured Query Language* (SQL), se aquele usuário está cadastrado no sistema. Logo em seguida após o processo de autenticação vem a autorização que normalmente significa *quem acessa o que?*



Figura 5-1 – Tela Inicial do Sistema de Software

Se o usuário não estiver cadastrado o sistema retorna uma mensagem dizendo que o usuário não existe. Após a autenticação do sistema, se o usuário for um técnico o sistema o direcionará para a página principal do sistema.

Através da Figura 5-2 podemos visualizar no cabeçalho da página que existem oito *links* principais, a saber. Principal, Técnicos, Produtores, Solicitação de Serviço, Curva Padrão, Laudos, Sobre e Sair. A seguir cada link será detalhado melhor. No *link* intitulado principal possui apenas um breve histórico do laboratório.

Quando o usuário clicar no *link* técnicos será direcionado diretamente para a página de cadastro de um novo técnico. No lado esquerdo dessa página, sob o título Técnico, há três *links*, sendo eles:

- Novo Registro: através desse *link* o usuário técnico cadastra um novo técnico no sistema.
- Listar Registros: através desse *link* o usuário é direcionado para a própria página da Figura 5-2. Essa página permite que o técnico visualize os dados de

todos os técnicos cadastrados no sistema, além de editar e remover informações de um outro técnico do sistema.

- **Buscar Registro:** este link permite ao técnico buscar informações de outros técnicos cadastrados no sistema.

Laboratório de Análises Físico-Químicas de Aguardente (LAFQA)						
Principal Técnicos Produtores Solicitações de serviço Curva Padrão Laudos Sobre Sair						
Técnico  Novo Registro  Listar Registros  Buscar Registro	Nome	Telefone	Celular	E-mail	Editar	Remover
	Anderson Junior Silva	(32)3564-1203	(32)8865-1020	anderson@yahoo.com.br		
	Antônio Carlos Moraes Souza	(11)2653-9874	(11)9152-2367	antonio2345@gmail.com		
	Antônio Carlos Moraes Souza	(11)2653-9874	(11)9152-2367	antonio2345@gmail.com		
	Bryan Adams	(35)5468-2154	(35)9128-8652	hudson@gmail.com		
	Célio Freitas da Silva	(11)5069-2514	(11)9125-2365	celiofreitas@gmail.com		
	Hudson Zoff	(35)3821-2514	(35)8845-2365	hudson@gmail.com		
<div>Voltar Avançar</div>						
Laboratório de Análises Físico-Químicas de Aguardente - LAFQA						

Figura 5-2 - Tela de Exibição de Listagem de Técnicos

No link produtores o técnico será direcionado diretamente para a página de cadastro de um novo produtor. No lado esquerdo dessa página, sob o título Produtor, há três *links*, sendo eles:

- **Novo Registro:** através desse *link* o usuário técnico cadastra um novo produtor no sistema.
- **Listar Registros:** através desse *link* o usuário técnico é direcionado para a página de listagem dos produtores. Essa página permite que o técnico visualize os dados de todos os produtores cadastrados no sistema, além de editar e remover informações de qualquer produtor do sistema.
- **Buscar Registro:** este link permite ao técnico buscar informações de qualquer produtor no sistema.

No link solicitação de serviço o técnico será direcionado diretamente para a página de cadastro de uma nova solicitação representada pela Figura 5-3. Este link é muito importante, pois interliga duas partes do sistema, ou seja, os dados dos produtores e técnicos aos dados das amostras e das análises.

No lado esquerdo dessa página, sob o título Solicitação, há quatro *links*, sendo eles:

- Nova Solicitação de Serviço: através desse *link* o usuário técnico cadastra uma nova solicitação de serviço no sistema.
- Listar todas solicitações: através desse *link* o usuário técnico é direcionado para a página de listagem de todas as solicitações já cadastradas no sistema. Essa página permite que o técnico visualize os dados de todas as solicitações cadastradas no sistema, em ordem da última solicitação registrada.
- Listar Solicitações do Produtor: faz a busca de uma solicitação específica de um determinado produtor.
- Buscar Solicitação de Serviço: realiza a busca de uma determinada solicitação de serviço através do número de protocolo.

A interface web do Laboratório de Análises Físico-Químicas de Aguardente (LAFQA) apresenta uma barra de navegação superior com o nome do laboratório e links para: Principal, Técnicos, Produtores, Solicitações de serviço, Curva Padrão, Laudos, Sobre e Sair. O conteúdo principal é dividido em duas seções. À esquerda, sob o título "Solicitação", há um menu vertical com quatro opções: "Nova Solicitação de Serviço", "Listar Todas Solicitações", "Listar Solicitações do Produtor" e "Buscar Solicitação de Serviço". À direita, a seção "Nova Solicitação" contém um formulário com os seguintes campos: "Produtor" (menu suspenso com "Luiz Paulo Júnior" selecionado), "Número de Amostras" (campo de texto) e "Nome da Solicitação" (campo de texto). Abaixo dos campos, há um botão "Avançar". Uma barra de rodapé azul na base da página repete o nome do laboratório: "Laboratório de Análises Físico-Químicas de Aguardente - LAFQA".

Figura 5-3 – Tela de Cadastro da Solicitação de Serviço

Após cadastrar uma solicitação serviço no sistema os técnicos poderão realizar as análises da amostra dos produtores. A Figura 5-4 mostra a página dos cálculos Físico-Químicos do Extrato Seco.

Laboratório de Análises Físico-Químicas de Aguardente (LAFQA)

Principal | Técnicos | Produtores | Solicitações de serviço | Curva Padrão | Laudos | Sobre | Sair

Análises
Exame Organoléptico
Extrato Seco

← PRINCIPAL ANÁLISES CONCLUÍDAS

Extrato Seco a 100C(g/L)

Peso da capsula antes: (g)

Peso da capsula Depois: (g)

Resultado: 400.000

Laboratório de Análises Físico-Químicas de Aguardente - LAFQA

Figura 5-4 – Tela de Análise do Extrato Seco

O *link* curva padrão possui três *links* no menu esquerdo da página, são eles: Álcool Superior, Cobre e Furfural. Cada *link* desse é direcionado para uma página específica, onde será realizado o cálculo da curva padrão escolhida pelo técnico. O link de Laudos permite gerar e buscar o laudo de um produtor. Os laudos do sistema são gerados no formato de arquivo pdf e poderá ser impresso diretamente a partir do *browser*.

Existe também no sistema um *link* denominado sobre, com os créditos do programa e equipe participante do projeto e por último o *link* sair, para encerrar a sessão do usuário. É importante destacar que um laudo só possui valor quando assinado pela Prof.^a Dr.^a Maria das Graças Cardoso, e só será entregue ao produtor, após a taxa de cobrança de cada análise tiver sido paga anteriormente.

A Figura 5-5 ilustra um laudo gerado pelo sistema de informação Web, sendo exibido pelo *browser*.

1.	Exame Organoléptico	-X-
2.	Densidade Relativa (20/20°C)	-X-
3.	Cobre (mg/L)	5,0
4.	Extrato Seco a 100°C(g/L)	-X-
5.	Grau Alcoólico Real a 20°C (%V/V)	2.36
6.	Acidez Volátil em Ácido Acético (mg/100mL de Álcool Anidro)	95.27
7.	Álcool Superior (mg/100mL de Álcool Anidro)	52.84
8.	Furfural (mg/100mL de Álcool Anidro)	52.84
9.	Aldeídos em Aldeídos Acéticos (mg/100mL de Álcool Anidro)	77.52
10.	Ésteres em Acetato de Etila (mg/100mL de Álcool Anidro)	7.84
11.	Soma dos Componentes Secundários (mg/100mL de Álcool Anidro)	200,0
12.	Álcool Metílico (mg/100mL de Álcool Anidro)	-X-
13.	Açúcares Totais (g/L em Sacarose) - Aguardente Simples	<= 6,0

Figura 5-5 – Browser exibindo um laudo do Sistema

6. CONCLUSÕES

Com o software desenvolvido por este trabalho, espera-se que o laboratório consiga otimizar os processos envolvidos em cada análise e agilizar a emissão do boletim de registro ao produtor. A instalação do software no servidor poderá ser feita fora do local de trabalho dos técnicos, protegendo os dados através de *backup's* sistemáticos. O programa obtido através do presente trabalho possibilita integrar todos os dados do Laboratório de Análises Físico-Químicas em um único sistema computacional. As tecnologias baseadas na Web permitem oferecer aos usuários a oportunidade de consultar os resultados das análises em tempo real a partir de qualquer navegador Web e de qualquer sistema operacional.

O software pode ser evoluído a partir de características genéricas previstas no modelo, como trabalho futuro, implementar as análises de Açúcares Totais (g/L em Sacarose) – Aguardente Simples e Açúcares Totais (g/L em Sacarose) – Aguardente Adoçada, que atualmente não são realizadas pelo (LAFQA), mas futuramente essas análises serão realizadas pelo laboratório.

Antes de implantar o sistema será necessário fazer os testes de consistência e os testes de segurança do software, pois o sistema de autenticação do software não utiliza o filtro oferecido pelo *JavaServer Faces JSF*. Esses testes servem para evitar que um usuário não logado consiga acessar uma página do sistema.

Outra parte da evolução do software diz respeito ao controle de pagamento dos laudos gerado pelo sistema e por último transformar o software em um ***Laboratory Information Management System (LIMS)***, onde os

equipamentos do laboratório comunicariam diretamente com o sistema de informação Web, conseqüentemente os técnicos não precisariam fornecer os parâmetros de entrada, pois seriam recebidos diretamente dos equipamentos laboratoriais (PASZKO, 2002).

7. REFERÊNCIAS BIBLIOGRÁFICAS

BASHAM, B.; SIERRA K.; BATES B. **Head First Servlets & JSP**, 2 Edition. O'REILLY. 2008.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The Unified Modeling Language User Guide**. Addison-Wesley, 1999.

BRASIL. Ministério da Agricultura, Pecuária e Abastecimento. Decreto 2314, 4 set. 1997. Brasília DF: **Diário Oficial**, 1990.

BRASIL. Ministério da Agricultura, Pecuária e Abastecimento. Decreto 4.851, 2 out. 2003. Brasília, DF: **Diário Oficial**, 2003.

BRASIL. Leis, decretos, etc. Instrução Normativa nº 13, de 29 de junho de 2005. **Diário Oficial da União**, Brasília, 30 jun. 2005.

CARDOSO, M. das G. **Produção de Aguardente de Cana**. 2. ed. Ver. e ampl. Lavras: Editora UFLA, 2006.

DEITEL, H.M. e DEITEL, P.J. **Java How to Program**, 6th Edition. Prentice Hall, 2004.

DRAPER, N.R.; SMITH, H. **Applied Regression analysis**. third edition, 1998.

FOX, C. **Introduction to software engineering design: processes, principles, and patterns with UML 2**. 1 st ed. Boston: Pearson, 2006.

GAMMA, E. **Padrões de Projeto: soluções de software orientado a objetos**, Porto Alegre: Bookman, 2000.

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

GONÇALVES, E. **Desenvolvendo Aplicações Web com Netbeans IDE 6**. Editora Ciência Moderna, 2008.

HEUSER, C. A. **Projeto de Banco de Dados**. Série Livros Didáticos. 5. ed., n. 4. Rio Grande do Sul: Sagra Luzzatto, 1998.

JASPER. **JasperReports – Home**. 2009. Disponível em
<<http://jasperreports.sourceforge.net/>>. Acesso em 8 set. 2009.

JUNG, C. F. **Metodologia para Pesquisa e Desenvolvimento: aplicada a novas tecnologias, produtos e processos**. Axcel Books do Brasil Editora, Rio de Janeiro, RJ, 2004.

KITO, D. **JAVASERVER FACES IN ACTION**. Mann Foreword by Ed Burns: MANNING. 2005.

KURNIAWAN, B. **Java para a Web com Servlet, JSP e EJB**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2002.

KUROSE, J. F. **Redes de Computadores e a internet: uma nova abordagem** / James F. KUROSE, Keith W. Ross; Tradução Arlete Simille Marques; Revisão Técnica Wagner Luiz Zucchi. - 1. ed - São Paulo: ADDISON WESLEY, 2003.

MARCONI, M. A.; LAKATOS, E. M. **Fundamentos de Metodologia Científica**. São Paulo: Atlas, 2003.

MINAS FAZ CIÊNCIA. **Cachaça de Minas**. Disponível em:
<http://revista.fapemig.br/materia.php?id=140>. Acesso em: 10 ago.2009.

PASZKO, C.; TURNER, E. **Laboratory Information Management Systems** Second Edition, Revised and Expanded. 2002.

QUINSTREET Copyright 2010 Inc. Disponível em:
<http://www.developer.com/design/article.php/3309461/Using-Design-Patterns-in-UML.htm>. Acesso em: 7 ago. 2009.

SILBERSCHATZ, A. **Sistemas Operacionais: Conceitos e Aplicações**. ABRAHAM SILBERSCHATZ, PETER BAER GALVIN, GREG GAGME; Tradução de DAniel Vieira. Rio de Janeiro: ELSEVIER, 2004.

SOMMERVILLE, I. **Engenharia de software**/ Ian Sommerville; tradução André Maurício de Andrade Ribeiro; revisão técnica Kechi Hirama. -- São Paulo: Addison Wesley, 2007.

SUN MICROSYSTEMS, 2001-2002. Disponível em:
<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>.
Acesso em: 5 de set. 2009.