

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Autarquia associada à Universidade de São Paulo

**CRIAÇÃO DE UM BANCO DE DADOS DINÂMICO E
ANÁLISE DE MEDIÇÕES LIDAR EM FORMATO WEB DO
LABORATÓRIO DE APLICAÇÕES AMBIENTAIS A LASER DO
INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES**

LUCILA MARIA VIOLA POZZETTI

**Dissertação apresentada como parte
dos requisitos para obtenção do Grau
de Mestre em Ciências na Área de
Tecnologia Nuclear - Materiais.**

**Orientador:
Dr. Eduardo Landulfo**

**SÃO PAULO
2006**

**Dedico esse trabalho em especial
ao meu marido Paulo Eduardo
pelo estímulo sem fronteiras.**

AGRADECIMENTOS

Chegou a hora de agradecer àquelas pessoas que, de alguma forma, contribuíram para a realização deste trabalho. Pessoas que me incentivaram de diferentes maneiras. A todos vocês o meu sincero “MUITO OBRIGADO”!

Em especial ao orientador Dr. Eduardo Landulfo, pela oportunidade, amizade, permanente incentivo e conhecimento transmitido, que foram essenciais não só para o desenvolvimento deste trabalho como também para o desenvolvimento da minha vida profissional;

Ao IPEN (Instituto de Pesquisas Energéticas e Nucleares) e USP (Universidade de São Paulo), em especial ao CLA (Centro de Lasers e Aplicações), pela oportunidade de fazer esse mestrado;

Aos professores (IPEN, IME, POLI e IAG), pela arte de ensinar e atenção;

Aos colegas do Laboratório de Aplicações Ambientais a Laser, pela amizade e incentivo. Cada um de vocês contribuiu de uma forma especial para esse trabalho;

A minha família, pelo carinho, incentivo e compreensão por tantas horas de dedicação a este trabalho; Sem vocês nada seria possível!

Aos colegas do IME, pela ajuda e incentivo;

Ao CEETEPS (Centro Estadual de Educação Tecnológica Paula Souza) pelo ensejo de seguir este percurso, e aos colegas pelo incentivo;

Aos colegas da PRODAM (Companhia de Processamento de Dados do Município de São Paulo), pela ajuda e incentivo;

A todas as pessoas que de alguma forma contribuíram para a obtenção deste importante objetivo na minha carreira;

E a DEUS, por mais uma vez, ter permitido vencer uma importante etapa da minha vida!

“O poder do conteúdo compensará a simplicidade da forma”.

I Ching

**CRIAÇÃO DE UM BANCO DE DADOS DINÂMICO E
ANÁLISE DE MEDIÇÕES LIDAR EM FORMATO WEB DO
LABORATÓRIO DE APLICAÇÕES AMBIENTAIS A LASER
DO INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES**

Lucila Maria Viola Pozzetti

RESUMO

O Laboratório de Aplicações Ambientais a Laser, situado no Centro de Lasers e Aplicações no IPEN (Instituto de Pesquisas Energéticas e Nucleares), efetua medidas das concentrações de aerossóis atmosféricos, enviando um feixe de laser à atmosfera e coletando a luz retroespalhada. Tal sistema fornece um grande número de parâmetros físicos que devem ser administrados de forma ágil para a obtenção de análises resultantes. Em consequência disso, a implementação de um banco de dados tornou-se imprescindível como instrumento de comunicação e visualização gráfica das medidas coletadas. Um critério de classificação destas valiosas informações foi adotado, estabelecendo níveis de armazenamento definidos a partir de características específicas aos tipos de dados determinados. A compilação e automação destas medidas promoverá a integração entre dados, análise e retorno otimizado de resultados das propriedades da atmosfera, propiciando futuras pesquisas e análise de dados.

**CREATION OF A DYNAMIC DATABASE AND
ANALYSIS OF LIDAR MEASUREMENTS IN WEB FORMAT AT THE
LABORATORY OF ENVIRONMENTAL LASER APPLICATIONS
AT THE NUCLEAR AND ENERGY RESEARCH INSTITUTE**

Lucila Maria Viola Pozzetti

ABSTRACT

The LIDAR system (Light Detection and Ranging) laser remote sensing at the Nuclear and Energy Research Institute – Laboratory of Environmental Laser Applications allows on line measurements of variations in the concentrations of atmospheric aerosols by sending a laser beam to the atmosphere and collecting the backscattered light. Such a system supplies a great number of physical parameters that must be managed in an agile form to the attainment of a real time analysis. Database implementation therefore becomes an important tool of communication and graphical visualization of measurements. A criterion for classification of this valuable information was adopted, establishing defined levels of storage from specific characteristics of the determined data types. The compilation and automation of these measurements will promote optimized integration between data, analysis and retrieval of the resulting properties and of the atmosphere, improving future research and data analysis.

SUMÁRIO

	Página
1. INTRODUÇÃO	14
1.1. Estrutura do texto	16
1.2. Sistema LIDAR	17
2. OBJETIVOS	20
3. CONSIDERAÇÕES TEÓRICAS	21
3.1. Análise de Requisitos	21
3.1.1. Arquitetura	22
3.1.2. Modelagem do Sistema	25
3.1.2.1. Contexto e modelo de uso	25
3.1.3. Aquisição de dados em tempo real	28
3.2. Banco de dados	29
3.2.1. Modelo de dados	31
3.2.2. Características	32
3.3. Sistema Gerenciador de Banco de dados	34
3.3.1. Definições	35
3.3.2. Características	37
3.4. Sistema de Análise de Dados	38
3.4.1. Definições	39
3.4.2. Características	39
3.5. Aplicação WEB	40
3.5.1. Definições	42
3.5.2. Características	45
4. METODOLOGIA	47
4.1. Equipamentos e Softwares	47

4.1.1. Princípios de operação	47
4.2. Especificação dos Dados	49
4.2.1. Tipos	50
4.2.2. Classificação	51
4.2.3. Organização	53
5. RESULTADOS E DISCUSSÕES	54
5.1. Banco de dados LIDAR	54
5.1.1. Características da elaboração	55
5.1.2. Estrutura	56
5.1.3. Consultas	57
5.2. Sistema de Análise de Dados LIDAR - SADL.....	58
5.2.1. Características da elaboração	59
5.2.2. Interface gráfica do Sistema de Análise de Dados LIDAR	60
5.2.2.1. Identificação dos dados	60
5.2.2.2. Níveis de dados	62
5.3. Web LIDAR	72
5.3.1. Características da elaboração	72
5.3.2. Pesquisa níveis de dados	73
6. CONCLUSÕES	78
6.1. Trabalhos futuros	79
ANEXOS	80
ANEXO A - Equação LIDAR e sua interpretação	81
ANEXO B - Especificação e Instrumentação do LIDAR	83
ANEXO C – Software de controle do Sistema LIDAR	85
REFERÊNCIAS BIBLIOGRÁFICAS	87

LISTA DE TABELAS

		Página
TABELA 1	Descrição do Caso de Uso Medir	26
TABELA 2	Comparação PostgreSQL vs MySQL	33
TABELA 3	Dados LIDAR	53
TABELA 4	Detalhamento da seleção de arquivos	63
TABELA 5	Detalhamento dos parâmetros de entrada e saída da análise de dados	68
TABELA 6	Especificação de análise LIDAR	69

LISTA DE FIGURAS

		Página
FIGURA 1	Diagrama ilustrativo dos componentes de um Sistema LIDAR genérico	18
FIGURA 2	Arquitetura em camada do Sistema LIDAR	23
FIGURA 3	Subsistema do Sistema LIDAR	23
FIGURA 4	Arquitetura do Sistema LIDAR	24
FIGURA 5	Caso de Uso do Sistema LIDAR	26
FIGURA 6	Seqüência de operação – Coleta de dados	27
FIGURA 7	Classes de objetos	27
FIGURA 8	Diagrama de estado	28
FIGURA 9	Ambiente do banco de dados	30
FIGURA 10	Ferramentas de desenvolvimento compatíveis com o PostgreSQL	34
FIGURA 11	Arquitetura aplicação WEB	42
FIGURA 12	Netbeans – módulos	45
FIGURA 13	Operação do Sistema LIDAR	48
FIGURA 14	Classificação por tipos de dados coletados pelo LIDAR	51
FIGURA 15	Diagrama do Banco de Dados LIDAR	57
FIGURA 16	Tela de identificação das medidas	61
FIGURA 17	Tela de seleção, visualização e transferência de arquivos ...	62
FIGURA 18	Leitura de dados analógico período I	64
FIGURA 19	Cálculo resultante de dados analógico período I	64
FIGURA 20	Gráfico de dados analógico período I (U.A. vs Canal)	64
FIGURA 21	Cálculo resultante de dados analógico período II	65
FIGURA 22	Gráfico de dados analógico período II (U.A. vs Canal)	65
FIGURA 23	Cálculo resultante de dados analógico período I e II	65
FIGURA 24	Gráfico de dados analógico período I e II (U.A. vs Canal)	65
FIGURA 25	Leitura de dados contagem de fótons período II	65

FIGURA 26	Cálculo resultante de dados contagem de fótons período II .	66
FIGURA 27	Gráfico de dados contagem de fótons período II (U.A. vs Canal).....	66
FIGURA 28	Interface gráfica da pré-análise de dados LIDAR	67
FIGURA 29	Análise de dois arquivos de dados referente à medida: Licel, comprimento de onda 532nm e contagem de fótons ...	68
FIGURA 30	Análise de períodos de dados referente à medida: Licel, comprimento de onda 532nm, analógico e contagem de fótons	69
FIGURA 31	Interface de Parâmetros de Análise de Dados LIDAR	70
FIGURA 32	Tela de seleção, visualização, geração e transferência de dados e gráficos de nível 2	70
FIGURA 33	Armazenamento e visualização de gráficos analisados	71
FIGURA 34	Gráfico nível 2 de categoria C	71
FIGURA 35	Estrutura de diretórios do Web LIDAR	73
FIGURA 36	Web Lidar – Níveis de dados	74
FIGURA 37	Web Lidar – Níveis 1	75
FIGURA 38	Web LIDAR – Nível 1 referente aos anos de 2001 a 2005	76
FIGURA 39	Web LIDAR – Gráfico de Nível 0 referente 20/08/2004	77
FIGURA 40	Sistema LIDAR	83
FIGURA 41	Publicação de dados utilizando o LabView	85

LISTA DE SIGLAS

AERONET	AErosol Robotic NETwork – Rede robótica de aerossol
API	Application Programming Interface – Interface de Programação de Aplicativos
AWT	Abstract Window Toolkit
BD	Banco de Dados
BDE	Borland Database Engine
CLA	Centro de Lasers e Aplicações
CPTEC	Centro de Previsão de Tempo
DDL	Data Definition Language – Linguagem de Definição de Dados
DML	Data Manipulation Language – Linguagem de Manipulação de Dados
EJB	Enterprise Java Beans
HTML	HyperText Markup Language
IDE	Integrated Development Environment – Ambiente Integrado de Desenvolvimento
INPE	Instituto Nacional de Pesquisas Espaciais
IPEN	Instituto de Pesquisas Energéticas e Nucleares
J2EE	Java 2 Enterprise Edition
JDBC	Java Database Connectivity
JSP	JavaServer Pages
JVM	Java Virtual Machine – Máquina Virtual Java
LAAL	Laboratório de Aplicações Ambientais a Laser
LASER	Light Amplification Stimulated Emission of Radiation
LICEL	Lidar Computing and Electronics
LIDAR	Light Detection and Ranging
NASA	National Aeronautics and Space Administration
ODBC	Open Database Connectivity
OHP	Observatory of Haute – Provence

OIDs	Identificador de Objetos
POO	Programação Orientada a Objetos
POSTQUEL	POSTgres QUERy Language – Linguagem de Consulta Postgres
RADAR	Radio Detection and Ranging
SGBD	Sistema Gerenciador de Banco de Dados
SONAR	Sound Navigation and Ranging
SQL	Structure Query Language – Linguagem de Consulta Estruturada
UML	Unified Modeling Language - Linguagem Unificada de Modelagem
WEB	World Wide Web – “WWW” ou “Web”
XML	eXtensible Markup Language

1. INTRODUÇÃO

Um dos problemas enfrentados pela sociedade moderna é o de encontrar um meio de se armazenar informações de forma segura e ágil para que se efetue sua busca rápida e ordenada. Banco de dados e sistemas de bancos de dados tornaram-se componentes essenciais para a resolução desses problemas.

Ao longo de um dia, quase todos nós praticamos diversas atividades que envolvem a interação com bancos de dados. Esta interação é exemplo do que podemos chamar de aplicações de banco de dados. Nos últimos anos, avanços tecnológicos têm levado a novas e interessantes aplicações de sistemas de banco de dados, que desempenham um papel crítico em quase todas as áreas da ciência.

A expressão banco de dados será amplamente utilizada neste trabalho, portanto o iniciaremos com uma definição do que é o banco de dados LIDAR.

O banco de dados LIDAR é uma coleção de dados relacionados. Definimos dados como medidas coletadas que podem ser armazenadas e que possuam um significado implícito, em outras palavras, este banco de dados possui uma fonte, da qual os dados são coletados, um grau de correlação com eventos do mundo real e um público que está ativamente interessado no conteúdo deste banco de dados.

Atualmente, um dos maiores problemas nas grandes cidades é a poluição do ar. São Paulo está entre as maiores áreas metropolitanas do mundo, nas quais a atividade humana invariavelmente causa enorme impacto na qualidade do ar, e, por conseqüência, na saúde da população. Neste contexto é necessário melhorar os meios de monitoração e controle da qualidade do ar, especialmente através do desenvolvimento de sistemas compactos com rápida recuperação de dados.

O Laboratório de Aplicações Ambientais a Laser, situado na área periférica na cidade de São Paulo (23° 33' S, 46°44'W) no Centro de Lasers e Aplicações (CLA) localizado no Instituto de Pesquisas Energéticas e Nucleares (IPEN),

desenvolveu um sistema de sensoriamento remoto LIDAR (*Light Detection and Ranging*).

Este sistema de retroespalhamento LIDAR, primeiro deste tipo no Brasil, está em funcionamento desde 2001, efetuando medidas das concentrações de aerossóis atmosféricos, através da emissão de um feixe de laser para a atmosfera, onde este por sua vez, interage com os constituintes atmosféricos durante a sua propagação, quando coleta-se a luz retroespalhada.

As primeiras medidas com o sistema LIDAR foram tomadas a partir de Agosto de 2001. Tal sistema fornece um grande número de parâmetros físicos que devem ser administrados de forma ágil para que se obtenham as análises resultantes. Até o primeiro semestre do ano de 2005, foram efetuadas medições em 234 dias gerando mais de 22.000 aquisições, ressaltando que o funcionamento deste sistema depende das condições meteorológicas, em virtude de não poder funcionar em dias de alta nebulosidade ou chuva.

Por conseqüência, a implementação de um banco de dados tornou-se um importante instrumento de análise e visualização gráfica das medidas coletadas. Foi adotado um critério de classificação destas valiosas informações, que estabelece níveis de armazenamento, definidos a partir de características específicas aos tipos determinados. A compilação e automação destas medidas, promoverá a integração entre dados e análise, obtendo-se assim um retorno otimizado dos resultados das propriedades da atmosfera, que propiciarão futuras pesquisas e análise dos dados.

Em pesquisas envolvendo coleta de dados atmosféricos, podemos destacar, dentre as que utilizam banco de dados, o Centro de previsão de tempo e estudos climáticos (CPTEC)¹, do Instituto Nacional de Pesquisas Espaciais (INPE) e a *AErosol RObotic NETwork (AERONET)*², da *NASA Goddard Space Flight Center*, onde os dados utilizados também são classificados em níveis.

Esta pesquisa apresenta-se diferenciada em relação as áreas envolvidas, tratando-se de uma das condicionantes desta implementação e proporcionando resultados válidos, do ponto de vista das áreas de conhecimento (física, meteorologia e informática). Destacamos assim a importância da característica multidisciplinar na pesquisa.

1.1. Estrutura do texto

O presente trabalho enfatiza inicialmente a teoria, onde é especificado o Sistema LIDAR. Seus elementos mais relevantes são demonstrados, além da apresentação da equação LIDAR de retroespalhamento simples, onde são considerados os coeficientes de retroespalhamento e extinção. Apresenta-se também a arquitetura envolvida, utilizada para facilitar a visualização do sistema LIDAR, a Linguagem Unificada de Modelagem (*UML*), o que propicia agilidade no processo de análise.

A seguir, apresenta-se a aquisição de dados em tempo real, o sistema de banco de dados, descrevendo noções de modelo relacional e orientado a objeto, o sistema gerenciador de banco de dados e o *Web* LIDAR, com definições das ferramentas utilizadas nas implementações, bem como as suas características e performances.

A parte experimental apresenta o LIDAR, sua especificação, seus princípios de operação, os tipos, a classificação e a organização dos dados. A especificação descreve os principais módulos do sistema, identificados como a fonte luminosa (LASER), o sistema de coleta e envio de feixe (Telescópio) e o sistema de detecção (detectores e módulos de aquisição).

Os princípios de operação demonstram as interfaces de funcionamento do sistema LIDAR, divididas em etapas relacionadas tais como: laser, aquisição de dados e divulgação dos dados. Os tipos de dados descrevem os diferentes modos de processamento e armazenamento, utilizando o osciloscópio e o registrados de transiente como equipamentos de aquisição.

Na classificação dos dados é abordado o critério de classificação adotado. São classificados como nível 0, 1 e 2 a partir de características específicas aos tipos determinados. Ao completar este processo, é apresentado o grande volume de dados coletados e já armazenados, tornando evidente a necessidade de organização destas informações em um banco de dados.

Finalmente, nos resultados, são mostradas as implementações identificadas como banco de dados, suas características de elaboração, a visualização da estrutura em níveis e as consultas ao servidor. O Sistema de Análise de Dados LIDAR exibe as interfaces gráficas de acesso ao servidor de banco de dados, assim como o *Web* LIDAR.

1.2. Sistema LIDAR

LIDAR é um acrônimo (*Light Detection and Ranging*). Sistemas LIDAR são sistemas baseados em laser que operam com princípios físicos similar aos do radar (*radio detection and ranging*) ou sonar (*sound navigation and ranging*).

No caso do LIDAR, um pulso de luz é emitido para atmosfera. O feixe de luz é espalhado em todas as direções pelas partículas e moléculas contidas na atmosfera. Uma parcela da luz é retroespalhada em direção ao sistema LIDAR, essa luz é coletada por um telescópio e direcionada para o fotodetector que mede a luz retroespalhada em função da distância das partículas e moléculas ao LIDAR³.

A idéia de se medir a densidade e carga de poeira em suspensão na alta atmosfera, por meio de técnicas de espalhamento de luz, já existe desde o começo da década de 30^{4,5}, quando se fazia a observação do espalhamento de um feixe de holofote com um detector localizado remotamente. A relação sinal-ruído, no entanto, era muito insatisfatória, já que a fonte de luz era policromática, não coerente e pouco colimada, o que obrigou a se descartar tal procedimento até a década de 60, quando surgiram os lasers.

A partir daquela época, o sensoriamento remoto com laser, desenvolveu-se o suficiente para tornar-se um poderoso instrumento na avaliação da presença de partículas e gases na atmosfera, na monitoração de emissão de gases de chaminés das indústrias, na prospecção de petróleo como indicador de vazamentos e emanações espontâneas do solo e até sistemas de vigilância militar de armas químicas e biológicas⁶.

Na aplicação de monitoração de poluentes na atmosfera, o LIDAR, com uma fonte laser de comprimento de onda de largura de banda estreita, pode informar sobre a presença e distribuição de partículas de 0,1 a 10 μm de diâmetro. Se o LIDAR for operado com dois sinais distintos em comprimento de onda pode-se obter informação da concentração específica do conteúdo de gases na atmosfera, atingindo-se um nível de sensibilidade de até 0,1 ppb, no caso das concentrações moleculares⁷.

Na sua forma mais genérica, o LIDAR emprega um laser como fonte pulsada de energia de magnitude útil e duração adequada. Em geral, os picos de energia atingem dezenas de megawatts e durações de 10 a 20 nanosegundos.

Pulso com tais energias, cerca de 1 Joule, são direcionados por sistemas ópticos, telescópios, para a atmosfera. O esquema a seguir mostra um sistema e seus elementos mais relevantes.

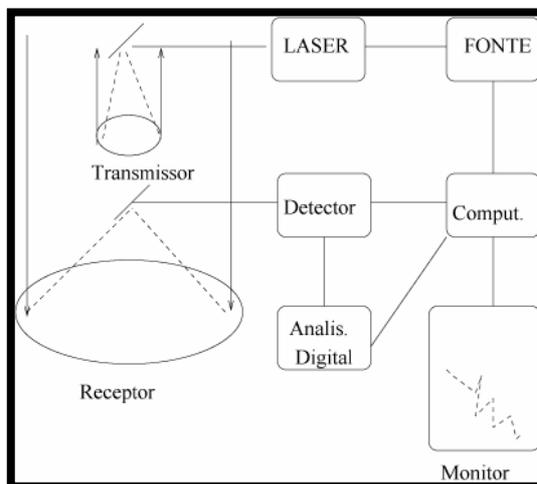


FIGURA 1 - Diagrama ilustrativo dos componentes de um sistema LIDAR genérico.

Conforme o feixe laser se propaga pela atmosfera, o mesmo é espalhado pelas moléculas de gás e pelas partículas ou gotículas encontradas em seu caminho. Uma pequena fração do feixe é retroespalhada na direção de coleta de dados, o receptor. A parte do feixe que se espalha para fora do caminho de propagação, ou de absorção pelos gases e partículas, corresponde a uma atenuação da intensidade. Essa atenuação ocorre *antes* e *depois* do feixe atingir a região de retroespalhamento.

Na sua chegada ao sistema de coleta, os fótons atingem o receptor de óptica refrativa ou refletiva, e em seguida são transferidos a um detector de resposta rápida, uma fotomultiplicadora ou detector de estado sólido, onde é gerado um sinal elétrico cuja intensidade em qualquer instante é proporcional à potência óptica recebida. Como a velocidade de propagação da luz é conhecida, a distância do volume espalhador que produz o sinal é determinada a partir do intervalo de tempo da transmissão do pulso. Estes fatores são especificados no ANEXO A através da interpretação da equação de LIDAR.

A magnitude do sinal recebido é determinada pelas propriedades de retroespalhamento da atmosfera em distâncias sucessivas e pela atenuação no

caminho de ida e volta. O retroespalhamento atmosférico, por sua vez, depende do comprimento de onda do laser, do número, tamanho, forma e propriedades de refração das partículas ou gotículas (ou moléculas) interceptadas pela luz incidente. Embora o fenômeno de espalhamento em um *ensemble* de elementos seja complicado, em primeira aproximação o retroespalhamento aumenta com a concentração de partículas.

Um fator de considerável importância na técnica LIDAR é a intensidade do mínimo sinal detectável, que é limitada pelo nível de ruído do sistema, tanto de fontes internas, que são controláveis, como de fontes externas, que não são controláveis. Destacamos o ruído térmico dos circuitos elétricos envolvido, ruídos intrínsecos dos detectores utilizados, e o ruído de origem solar, presente logicamente durante o dia⁸. Dado o alto grau de monocromaticidade do laser, a luz externa pode ser substancialmente eliminada pela utilização de filtros de banda passante estreita, centrada no comprimento de onda do laser.

2. OBJETIVOS

O presente trabalho tem por objetivo:

- A criação de um banco de dados das medidas LIDAR;
- A elaboração de uma metodologia de análise otimizada em função do banco de dados, tendo como produto final o desenvolvimento de um programa de análise que forneça as propriedades da atmosfera e sua estrutura;
- E a criação de um sistema de transferência automatizado das informações para um *site* da internet para mostra dos dados adquiridos em tempo real pelo equipamento LIDAR.

3. CONSIDERAÇÕES TEÓRICAS

3.1. Análise de Requisitos

O processo de desenvolvimento de um sistema computacional tem seu início quando é estabelecido aquilo que se quer implementar. O estabelecimento desse objetivo é realizado através da identificação dos requisitos do sistema.

Define-se requisito do sistema⁹ como sendo:

- Uma condição ou capacidade exigida para resolver um problema ou atingir um objetivo;
- Uma condição ou capacidade que deve ser atendida ou possuída por um sistema ou componente do sistema para satisfazer uma especificação.

Na análise de requisitos procurou-se sistematizar o processo de definição de requisitos, definindo métodos, técnicas e ferramentas que facilitaram esse trabalho de definição. Essa sistematização se fez necessária na medida em que o sistema tornou-se cada vez mais complexo, exigindo uma grande atenção na correta compreensão do problema antes da elaboração de uma solução¹⁰.

Apesar dos processos de desenvolvimento de sistemas variarem muito em tamanho e complexidade, bem como dependerem do tipo de sistema e das organizações envolvidas no seu desenvolvimento, existem algumas etapas principais que são comuns a todos eles.

Um modelo do processo de levantamento e análise de requisitos do sistema LIDAR envolveu as atividades a seguir:

- Compreensão do domínio: Desenvolver o entendimento do domínio do sistema, no caso a compreensão do funcionamento do sistema LIDAR;
- Coleta de requisitos: O processo de interação deste sistema para descobrir seus requisitos;

- Classificação: Atividade que considera o conjunto não estruturado dos requisitos e os organiza em grupos coerentes;
- Definição das prioridades: Estágio que envolve o conjunto de requisitos, definindo suas importâncias;
- Verificação de requisitos: Os requisitos foram verificados, quanto a sua conformidade com o que se deseja.

Nesta etapa, o levantamento e a análise de requisitos foi um processo iterativo, com *feedback* contínuo de cada atividade. O ciclo começa com a compreensão do domínio e termina com a verificação dos requisitos, onde a compreensão dos requisitos é melhorada a cada fase¹¹.

3.1.1. Arquitetura

É importante ressaltar o entendimento do que seja a arquitetura. Neste trabalho adotou-se a definição pela qual a arquitetura refere-se aos conceitos e regras que definem a estrutura do sistema de banco de dados LIDAR e o relacionamento entre suas interfaces. Essa definição inclui os elementos que compreendem as entidades do sistema, o relacionamento entre esses elementos e as restrições associadas a esses relacionamentos¹².

Essa definição de arquitetura, que compreende o *software* e o *hardware* do sistema, possibilitou variados graus de abstração e de detalhamento, podendo a arquitetura representar uma parte ou a totalidade do sistema. Assim, no desenvolvimento deste trabalho, os passos de análise de requisitos e o desenvolvimento da arquitetura do sistema de banco de dados se alternam, com um detalhamento maior a cada iteração do ciclo de desenvolvimento. A saída da análise de requisitos foi fornecida como entrada para a próxima fase de desenvolvimento, e a saída de desenvolvimento foi utilizada como entrada da próxima fase de análise de requisitos.

As ferramentas e métodos utilizados no desenvolvimento deste trabalho forneceram suporte a todos os aspectos, permitindo que os processos de análise de requisitos e desenvolvimento fossem integrados.

A necessidade de se construir uma arquitetura competente para o sistema torna-se cada vez mais iminente, já que envolve a agilidade no processo de análise.

A FIG. 2 ilustra a arquitetura do sistema LIDAR. Trata-se de uma arquitetura¹¹ em camadas, que reflete os diferentes estágios de processamento do sistema, como a aquisição, o armazenamento e a pré-análise e análise de dados. Nesta arquitetura, as atividades influenciam umas às outras, os objetos foram identificados e as interfaces, especificadas completa ou parcialmente. Entretanto, à medida que os objetos são produzidos, as definições individuais de objetos podem ser aprimoradas, o que pode significar mudanças na arquitetura do sistema.

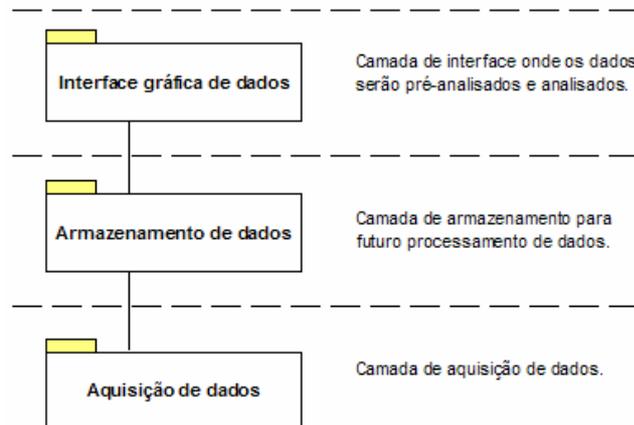


FIGURA 2: Arquitetura em camadas do sistema LIDAR.

Na FIG. 3, o modelo abstrato de arquitetura foi ampliado afim de mostrar os componentes dos subsistemas LIDAR. Novamente, os componentes são muito abstratos e foram derivados das informações contidas na descrição do sistema.

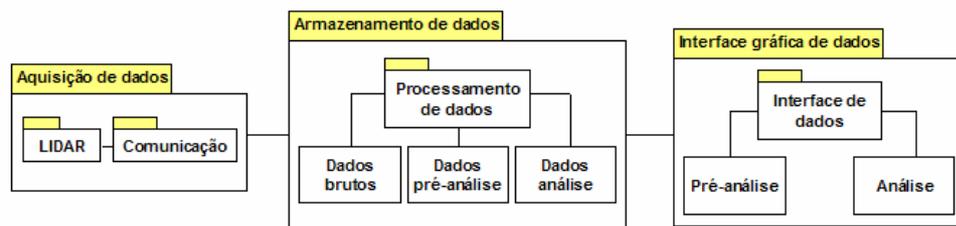


FIGURA 3: Subsistemas do Sistema LIDAR.

Os subsistemas¹³ são compostos de módulos e têm interfaces definidas, as quais são utilizadas para a comunicação entre eles. Cada módulo é um componente do sistema que fornece e/ou utiliza uma ou mais passagens de parâmetros para outros módulos, embora não sejam considerados sistemas independentes.

Uma arquitetura¹¹ em camadas é apropriada nesse caso porque cada estágio depende apenas do processamento do estágio precedente para sua operação. A FIG.3, apresenta também as diferentes camadas, inclusive o nome da camada em um símbolo de *package* UML, que foi denotada como um subsistema LIDAR.

Um *package* UML representa uma coleção de objetos e de outros *packages*. Essa opção é utilizada para mostrar que cada camada inclui uma série de outros componentes, observando-o como um modelo estático.

Na arquitetura do sistema LIDAR, representada como um modelo em camadas, conforme ilustrado na FIG. 4, há três *packages* UML dentro do *package* mais geral, denominado Sistema LIDAR.

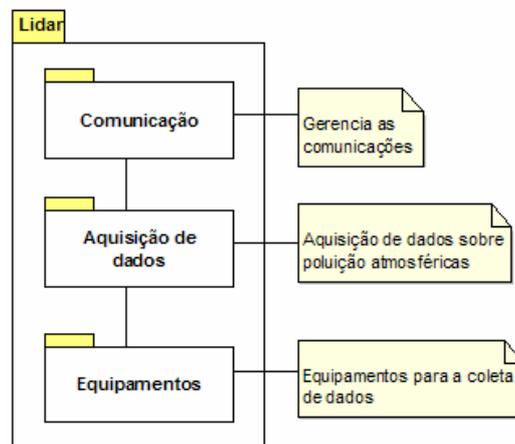


FIGURA 4: Arquitetura do Sistema LIDAR.

As três camadas do sistema são:

- A camada de comunicação, que é responsável pela interface de outras partes do sistema;
- A camada de aquisição de dados, que gera os dados a partir dos instrumentos, e;

- A camada dos instrumentos, que é um encapsulamento de todos os instrumentos (objetos) utilizados para coletar os dados brutos.

Esta visão geral da arquitetura integra a experiência extensiva do conhecimento englobando desde o funcionamento do LIDAR até a divulgação dos resultados em formato *Web*.

3.1.2. Modelagem do Sistema

Para facilitar a visualização do sistema LIDAR, foi utilizado a Linguagem Unificada de Modelagem (UML), que é uma linguagem gráfica para especificação, construção e documentação de artefatos de sistemas, definindo modelos em forma de diagramas, com nível de abstração e detalhamento desejado¹³.

O processo¹⁴ orientado a objetos utilizado neste trabalho incorpora atividades do sistema LIDAR como:

- Compreensão do contexto e dos modos de utilização do sistema.
- Especificação das interfaces dos objetos do sistema.

A estratégia orientada a objetos, utilizada no processo de desenvolvimento, envolve a análise do domínio do sistema, os requisitos identificados e a linguagem de programação. O sistema é constituído de objetos que interagem entre si, que mantêm seu próprio estado local e fornecem operações com base nessas informações de estado, envolvendo as classes de objetos e as relações entre essas classes. Quando o sistema está em execução, os objetos são criados dinamicamente, utilizando-se as definições das classes.

Os modelos¹³ utilizados para descrever este sistema são o estático e o dinâmico. No modelo estático, são mostradas as classes de objetos e seus relacionamentos, já no modelo dinâmico são apresentadas interações entre os objetos.

3.1.2.1. Contexto e modelo de uso

O contexto e o modelo de uso do sistema LIDAR representam dois modelos complementares das relações entre o sistema e seu ambiente:

- O contexto do sistema é um modelo estático, que descreve os outros sistemas no ambiente;
- O modelo de uso do sistema é um modelo dinâmico, que descreve como o sistema realmente interage com o seu ambiente.

Descrever essa compreensão ajuda a decidir como fornecer a funcionalidade necessária para o sistema e como estruturar o sistema de modo que ele possa se comunicar eficazmente com seu ambiente.

O modelo caso de uso do sistema LIDAR é descrito na FIG. 5, onde são mostradas as interações do sistema com o operador, para iniciar e desativar, medir e analisar os dados coletados e calibrar instrumentos.

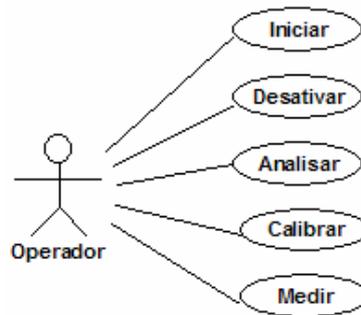


FIGURA 5: Caso de Uso do Sistema LIDAR.

A descrição do caso de uso “*Medir*”, conforme TAB.1, ajuda a compreender o funcionamento do sistema, identificando quais informações são trocadas, como a interação é iniciada, etc.

TABELA1: Descrição do Caso de Uso Medir.

Sistema	LIDAR
Caso de Uso	Medir.
Agente	Sistema de coleta de dados sobre poluição atmosférica.
Dados	O computador recebe os dados coletados pelo sistema LIDAR. Estes dados referem-se às propriedades de poluição atmosférica.
Estímulo	O <i>software Lab-View</i> requisita a transmissão dos dados.
Resposta	Os arquivos de dados coletados são enviados ao computador.
Comentário	Os dados coletados são recebidos a cada dois minutos, mas essa frequência pode diferir de uma medida para outra.

Descreve-se a seguir o modelo dinâmico de operação do sistema de coleta de dados que é representado pelo modelo de seqüência¹³, onde demonstra a seqüência de interações dos objetos, conforme mostrado na FIG. 6.

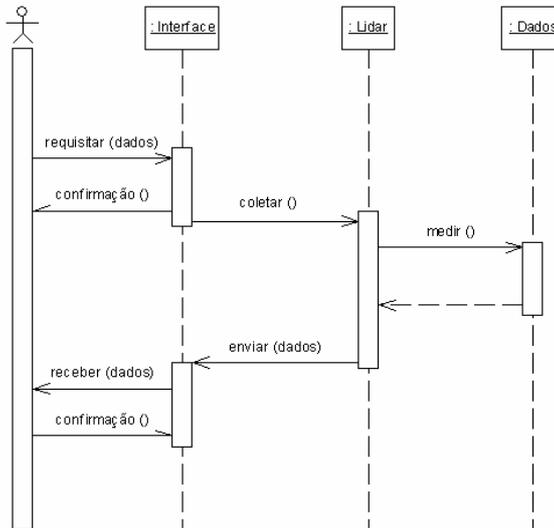


FIGURA 6: Seqüência de operações – coleta de dados.

A abordagem utilizada para o entendimento do comportamento geral do sistema e para a identificação de classes de objetos é apresentada na FIG. 7, onde este comportamento é atribuído a diferentes partes do sistema, sendo identificados os objetos Sistema LIDAR e Dados LIDAR.

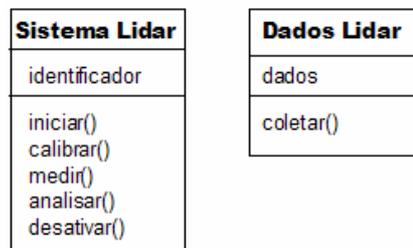


FIGURA 7: Classes de objetos.

Para modelar o comportamento do sistema, em resposta a eventos externos e internos, é utilizado o modelo de máquina de estado. Este modelo mostra os estados do sistema e os eventos que provocam as transições de um estado para o outro, ocultando o fluxo de dados dentro do sistema. Esse tipo de

modelo é útil para modelar sistemas em tempo real, que é o caso do sistema LIDAR, dirigido com frequência por estímulos do ambiente do sistema.

3.1.3. Aquisição de dados em tempo real

O modelo de máquina de estado¹⁴ é parte dos métodos em tempo real, como o que foi proposto por *Ward Mellor*¹⁵ e *Harel*¹⁶. Este método utiliza uma notação chamada *statecharts*¹, que foi a base para a notação de modelagem de máquina de estado na UML. Ele presume que, a qualquer momento, o sistema está em um estado dentro de possíveis estados. Quando um estímulo é recebido isso pode dar início a uma transição para um estado diferente, como por exemplo é o caso do sensor de umidade para ativação e desativação do laser para coleta de dados. A FIG. 8 é o *statechart* para o objeto sistema LIDAR, que mostra como ele responde a coleta de dados.

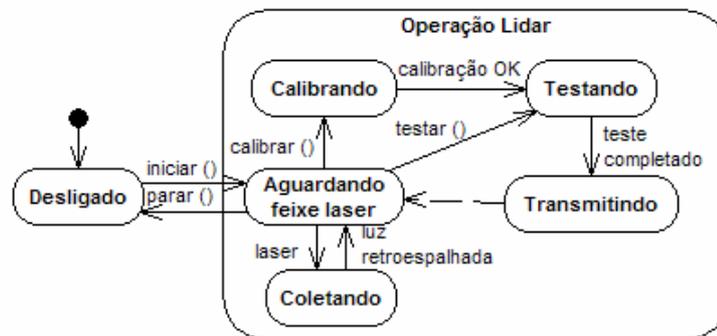


FIGURA 8: Diagrama de estado.

A aquisição de dados é realizada em tempo real¹⁷, o funcionamento depende dos resultados produzidos por ela e do tempo em que esses resultados são produzidos. Uma maneira de visualizar esta aquisição é equipará-la a um sistema de estímulo/resposta, considerando um determinado estímulo fornecido e a produção de uma resposta correspondente a este. Sua arquitetura é organizada de modo que o controle seja transferido ao manipulador apropriado para aquele estímulo, assim que ele for recebido.

¹ Inventado por *Harel*(1987), para descrever modelos de máquinas de estado.

Desta forma, o comportamento da aquisição de dados em tempo real, pode ser definido relacionando os estímulos que são recebidos pelo sistema, as resposta associadas e o tempo em que à resposta deve ser produzida. Identificou-se uma relação a um ambiente cliente servidor em tempo real, onde detalhes da performance são significativos¹⁸.

3.2. Banco de dados

Um sistema de banco de dados é dividido em módulos que tratam de cada uma das responsabilidades do sistema geral, onde seus componentes incluem:

- Gerenciador de arquivos, que gerencia a alocação do espaço na armazenagem do disco e as estruturas de dados usadas para representar a informação armazenada no disco;
- Gerenciador do banco de dados, que fornece a interface entre os dados armazenados no disco e os programas aplicativos e de consulta submetidos ao sistema;
- Processador de consultas, que traduz os comandos numa linguagem de consulta para instruções que o gerenciador do banco de dados pode interpretar.

O sistema de banco de dados tornou-se um componente essencial para o gerenciamento e análise dos dados coletados pelo equipamento LIDAR. Como exemplo, podemos citar o *Observatory of Haute-Provence (OHP)*¹⁹, que opera há mais de vinte anos e utiliza a técnica baseada em retroespalhamento de feixes de laser por moléculas atmosféricas, produzindo um banco de dados confiável das informações pesquisadas.

O sistema de banco de dados LIDAR, conforme FIG.9, mostra o sistema de gerenciador de banco de dados, o banco de dados e o sistema de análise de dados LIDAR²⁰.

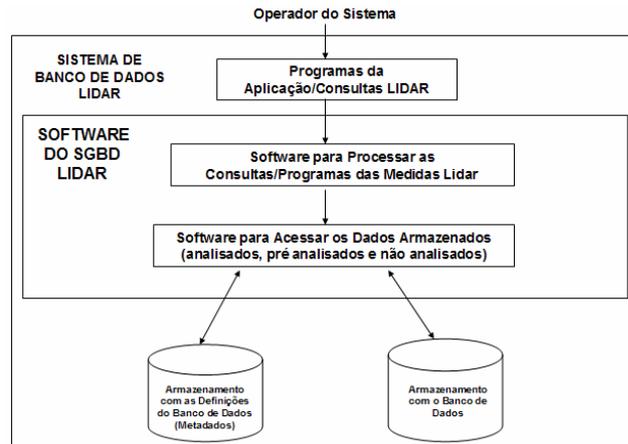


FIGURA 9: Ambiente do banco de dados.

No banco de dados as informações são relacionadas representando um domínio específico, já o sistema gerenciador de banco de dados permite a definição de estruturas para armazenamento de informações e fornecimento de mecanismos para manipular estas informações.

O esquema do banco de dados é especificado por um conjunto de definições chamado Linguagem de Definição de dados (*Data Definition Language, DDL*), onde o resultado da compilação de comandos de uma *DDL*, é um conjunto de tabelas que são armazenadas no diretório de dados. O diretório de dados é um arquivo que contém metadados.

O resultado da compilação destas definições é um conjunto de instruções que especificam a implementação do esquema do banco de dados. O acesso e manipulação aos dados são executados através da Linguagem de Manipulação de Dados (*Data Manipulation Language, DML*), utilizando uma consulta (*query*) na busca das informações.

No sistema de banco de dados agrega-se, além destas informações, o *software* para prover acesso aos dados. O principal objetivo deste sistema é possibilitar um ambiente que seja adequado e eficiente para uso na recuperação e armazenamento de informações e dados LIDAR.

Os recursos que um SGBD oferece ao usuário são: armazenamento persistente, interface de programação e gerenciamento de transações²¹. O sistema possui uma arquitetura Cliente-Servidor, onde o cliente “*front_end*” executa as tarefas do aplicativo, ou seja, fornece a interface do usuário (tela, e

processamento de entrada e saída), e o servidor “*back_end*” executa as consultas no banco de dados e retorna os resultados ao cliente. Para isso são necessárias soluções de *software* que possibilitem o tratamento de transações, as confirmações de transações (*commits*), e que se desfaçam transações (*rollbacks*), linguagens de consultas (*stored procedures*) e gatilhos (*triggers*). A principal vantagem desta arquitetura é a divisão do processamento entre dois sistemas, o que reduz o tráfego de dados na rede.

3.2.1. Modelo de dados

Um modelo de dados²⁰ é uma coletânea de conceitos que podem ser utilizados para descrever a estrutura de um banco de dados, fornecendo os meios necessários para alcançar um nível de abstração desejado e ocultação dos detalhes do armazenamento de dados que não são necessários para a maioria dos usuários.

O modelo é referenciado usando-se a terminologia de dados relacionais e dados orientados a objetos. Dados relacionais são modelados por tabelas, os itens dos dados são *tuplas* ou linhas da tabela e as *tuplas* têm um número fixo de componentes, cada um dos quais de um tipo fixo determinado pelo esquema da relação. No modelo de dados orientados a objetos, o item de dados elementar é um objeto. Os objetos são agrupados em classes e cada classe tem um esquema que é uma lista de propriedades:

- Algumas dessas propriedades são atributos que podem ser representados como atributos de uma *tupla* relacional;
- Outras propriedades são relacionamentos, que conectam um objeto a um ou mais objetos diferentes;
- Ainda outras propriedades são métodos, isto é, funções que podem ser aplicadas a objetos da classe.

O modelo relacional e o modelo de dados orientado a objetos compartilham muitas noções comuns, quase sempre com terminologia diferente. Existem analogias entre *tuplas* e objetos, e entre relações e classes. Dados relacionais são compostos por *tuplas*, que são compostas por atributos, já dados orientados a objetos, os objetos têm campos ou “variáveis de instâncias”, cujos valores

representam atributos do objeto ou o conjunto de objetos inter-relacionados, de acordo com algum relacionamento²¹.

Então é definida uma arquitetura de banco de dados para busca de dados e informações, com armazenamento seguro de arquivos, sendo que o desenvolvimento da aplicação de banco de dados é definido como o esquema dos dados e a implementação da estrutura da linguagem de consulta para a interface do banco de dados, enquanto o desenvolvimento da aplicação *Web* é definido e implementado a interface do usuário²².

3.2.2. Características

A abordagem deste trabalho está voltada para a proposição de um Banco de Dados Objeto-Relacional, com enfoque na Orientação a Objetos, dando uma breve introdução aos conceitos e requisitos básicos de um Banco de Dados e mostrando as suas principais características, como Integridade, restrições, segurança/privacidade, restauração, reorganização e eficiência.

Na criação do banco de dados pesquisou-se maneiras de se medir um *software*. Duas delas são a característica e a performance²³. Portanto, para definição da ferramenta a ser utilizada na implementação abordamos o *MySQL* (versão 4.1) e *PostgreSQL* (versão 8.0). O *MySQL*²⁴ é um sistema de gerenciamento de banco de dados relacional baseado em comandos *SQL* (*Structured Query Language*). Já o *PostgreSQL*²⁵ é um sofisticado SGBD Objeto-Relacional, o qual suporta praticamente todos os construtores da linguagem *SQL*, incluindo subconsultas (*subselects*), transações, funções e tipos definidos pelo usuário. Ele faz a integração de dados e objetos, ou seja, preserva as características relacionais e possibilita a utilização de características orientadas a objetos.

A TAB. 2 apresenta uma comparação dos dois, ilustrando a escolha do *PostgreSQL*. Não que ele possui uma plataforma sem falhas, mas tem um desempenho versátil razoável e todas as características essenciais a esta implementação. Nesta comparação, os itens assinalados demonstram características e performances melhores em relação aos itens não assinalados, de acordo com a versão citada anteriormente.

TABELA 2: Comparação PostgreSQL vs MySQL.

Características / Performances	PostgreSQL	MySQL
Licença (livre)	✓	✓
Plataformas (<i>Linux, Unix, Windows</i>)	✓	✓
SGBD Objeto-Relacional	✓	
Armazenamento de dados	✓	
Características (transações, subconsultas, gatilhos, visões, stored procedures, cursors, integridade referencial de chave estrangeira e travamento (<i>lock</i>) sofisticado)	✓	
Características (tipos definidos pelo usuário, herança, regras e controle de concorrência de múltiplas versões para reduzir travamentos (<i>locks</i>))	✓	
Performances (consultas simples feitas por poucos usuários)		✓
Performances (múltiplos usuários, consultas complexas e carga de consultas de leitura/escrita)	✓	
Índices	✓	
Junção usando índices	✓	

O *PostgreSQL* é um SGBD (Sistema Gerenciador de Banco de Dados) objeto-relacional de código aberto. Ele é extremamente robusto e confiável, além de ser flexível e rico em recursos, sendo considerado objeto-relacional por implementar, além das características de um SGBD relacional, algumas características de orientação a objetos, como herança e tipos personalizados.

O *postgreSQL* é um modelo relacional estendido, oferecendo objetos, *OIDs* (Identificador de Objetos), objetos compostos, herança múltipla, versões, dados históricos, e uma linguagem de consulta, *POSTQUEL*, extensão da linguagem do *INGRES*, *QUEL*. Possui também banco de dados cliente-servidor, onde existe um programa gerenciador de acesso que cuida da integridade dos dados, e outro que faz a interface com o usuário, portanto servidor e o cliente não precisam estar rodando na mesma máquina.

A FIG. 10 mostra que o *PostgreSQL* possui uma plataforma de desenvolvimento que possibilita a construção de aplicativos nas mais variadas ferramentas de desenvolvimento (ferramentas, linguagens e ambientes integrados, além de ferramentas para administração de bases *PostgreSQL*).

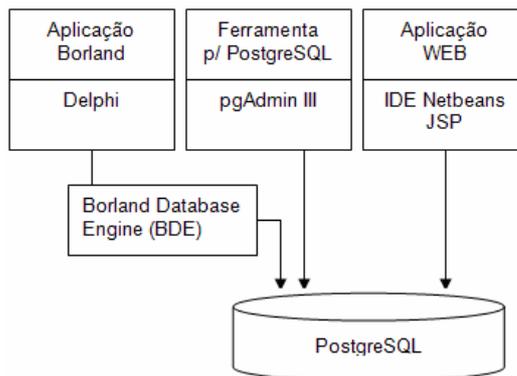


FIGURA 10: Ferramentas de desenvolvimento compatíveis com o *PostgreSQL*.

A segurança é uma característica de um banco de dados, ela lida com o controle de acesso como um todo, e com o controle de autorização para acessar partes específicas deste banco²⁰. O *PostgreSQL* implementa um controle de acesso, atribuindo-se senhas e autorização aos usuários, utilizando-se um sistema de concessão e revogação de privilégios a contas deste usuários.

Esse controle de acesso exige a classificação de usuários e valores de dados em classes de segurança que impõem regras que proíbem o fluxo de informações de níveis mais elevados para níveis mais baixos de segurança.

3.3. Sistema Gerenciador de Banco de dados

O Sistema Gerenciador de Banco de Dados (SGBD) consiste em uma coleção de dados inter-relacionados e em um conjunto de programas que compõe a camada responsável pelo armazenamento, acesso e recuperação dos dados no Sistema de Informação²⁶. O conjunto de dados, referenciado como banco de dados, contém informações sobre medidas coletadas.

O SGBD é considerado mais eficiente e seguro comparado a um sistema de processamento de arquivos, conforme apresenta-se posteriormente as características de cada um.

Seguem algumas desvantagens na utilização de um sistema de processamento de arquivos para armazenar dados.

- Redundância e inconsistência de dados;
- Dificuldade no acesso aos dados;

- Isolamento dos dados: como os dados estão espalhados, em arquivos separados e com formatos diferentes, é difícil escrever novos programas aplicativos para recuperar os dados adequados;
- Problemas de segurança: Se novos programas aplicativos forem adicionados, torna-se difícil assegurar restrições de segurança;
- Problemas de integridade: os valores dos dados armazenados necessitam satisfazer certas restrições;

Para evitar os problemas descritos acima, faz-se necessária a aplicação de um SGBD.

3.3.1. Definições

O objetivo principal deste SGBD é prover um ambiente que seja adequado e eficiente para recuperar e armazenar informações do banco. A linguagem padrão utilizada para consulta, manipulação e controle de acesso aos dados é o *SQL (Structured Query Language)*.

O sistema de banco de dado foi projetado para gerenciar grupos de informações. O gerenciamento de dados envolve a definição de estruturas para armazenamento de informação e o fornecimento de mecanismos para manipulá-las. Além disso, fornece segurança das informações armazenadas, incluindo a permissão de acesso.

Este sistema de banco de dados dispõe de uma visão abstrata de dados, isto é, o sistema omite certos detalhes sobre como os dados são armazenados e mantidos. Porém, essa estrutura complexa do banco de dados, tem direcionado diversos níveis de abstração. Para simplificar a interação do usuário com o sistema, são definidos três níveis de abstração (Nível Físico, Conceitual e de Visão).

O Nível Físico, o mais baixo nível de abstração, define de que forma os dados estão realmente armazenados. O nível seguinte, conceitual, descreve quais dados são armazenados no banco de dados e as relações que existem entre eles. O Nível de visões, o mais alto nível de abstração, é definido com o objetivo de simplificar a interação do usuário com o sistema, sendo apresentada

apenas parte do banco de dados, ou seja, as informações que interessam ao usuário.

Após a definição dos níveis de abstração, houve a necessidade de definir a independência de dados, ou seja, a habilidade de modificar a definição do esquema em um nível sem afetar a definição do esquema num nível mais alto. Existem dois níveis de independência dos dados (Física e Lógica). A Independência física de dados é a habilidade de modificar o esquema físico sem a necessidade de reescrever os programas aplicativos. As modificações no nível físico são ocasionalmente necessárias para melhorar o desempenho. A Independência lógica de dados é a habilidade de modificar o esquema conceitual sem a necessidade de reescrever os programas aplicativos. As modificações no nível conceitual são necessárias quando a estrutura lógica do banco de dados é alterada.

Os níveis de abstração discutidos (níveis físico, conceitual e de visão) não se aplicam somente à definição ou estrutura de dados, mas também à sua manipulação. A manipulação de dados significa a busca da informação, a inserção, a eliminação e a modificação de dados armazenados no banco de dados.

O poder dos bancos de dados vem do conhecimento e tecnologia que se desenvolveu ao longo de várias décadas e é incorporado em um tipo de *software* especializado chamado sistema de gerenciamento de banco de dados ou SGBD²¹.

Um SGBD é uma ferramenta poderosa para criar e gerenciar grandes quantidades de dados de forma eficiente e permitir que esses dados persistam durante longos espaços de tempo com segurança.

Os principais componentes de um SGBD são:

- O gerenciador de armazenamento: este componente é o responsável pelo armazenamento de dados, metadados (informações sobre o esquema ou a estrutura dos dados), índices (estrutura de dados para acelerar o acesso aos dados) e *logs* (registros de mudanças no banco de dados);
- O processador de consultas: este componente analisa consultas, otimiza essas consultas selecionando um plano de consultas e executa o plano sobre os dados armazenados;

- O gerenciador de transações: este componente é o responsável pelo registro de *log* de mudanças no banco de dados para dar suporte a recuperação após uma queda do sistema. Ele também admite a execução concorrente de transações de uma forma que assegura a atomicidade (uma transação é executada completamente, ou não é executada) e o isolamento (as transações são executadas como se não houvesse nenhuma outra transação sendo executada de modo concorrente).

3.3.2. Características

Os Sistemas de gerenciamento de bancos de dados²¹ são caracterizados por sua habilidade de admitir o acesso eficiente a grandes quantidades de dados, os quais persistem ao longo do tempo, por seu suporte para linguagens de consultas poderosas e transações duráveis que podem ser executadas de forma concorrente de um modo que parece atômico e independente de outras transações.

Os recursos que um SGBD oferece são:

- Armazenamento persistente. Do mesmo modo que um sistema de arquivos, um SGBD admite o armazenamento de quantidades muito grandes de dados, indo muito além do sistema de arquivos ao proporcionar flexibilidade, como estrutura de dados que permitem o acesso eficiente a quantidades muito grandes de dados;
- Interface de programação. Um SGBD permite acessar e modificar dados através de uma linguagem de consulta poderosa. Mais uma vez, sua vantagem sobre um sistema de arquivos é a flexibilidade para manipular dados armazenados de formas muito mais complexas que a leitura e gravação de arquivos;
- Gerenciamento de transações. Um SGBD admite o acesso concorrente aos dados, isto é, o acesso simultâneo por muitos processos distintos (chamados “transações”) ao mesmo tempo. Para evitar algumas das conseqüências indesejáveis do acesso simultâneo, o SGBD admite o isolamento, a aparência de que as

transações são executadas uma de cada vez, e a atomicidade, o requisito de que as transações sejam executadas completamente ou não sejam executadas de forma alguma.

As interações entre transações podem fazer o estado do banco de dados se tornar inconsistente, mesmo quando as transações preservam individualmente a correção do estado e não existe nenhuma falha do sistema. Desse modo, a ordem em que as etapas individuais de transações diferentes ocorrem precisa ser regulada de alguma maneira. É normal que diversas transações tenham acesso a um banco de dados ao mesmo tempo. A função de controlar essas etapas é dada ao componente escalonador (ou *scheduler*) do SGBD, e o processo geral de assegurar que as transações preservam a consistência quando forem executadas simultaneamente. A ela damos o nome de controle da concorrência.

As transações são desmembradas em ações, principalmente leitura e gravação no banco de dados. Uma seqüência dessas ações de uma ou mais transações é chamada de escalonamento.

3.4. Sistema de Análise de Dados

Na implementação do Sistema de Análise de Dados, utiliza-se a linguagem de Programação Orientada a Objetos (POO) *Delphi*²⁷ como interface, visualização e comunicação com o banco de dados, usando objetos como unidade de dados e de ações. Este ambiente de programação é baseado na linguagem *Object Pascal*, e implementa algumas definições, notadamente na parte de criação de classes e objetos.

A abordagem de orientação a objetos favorece a aplicação de diversos conceitos considerados fundamentais para o desenvolvimento. Nesta discussão, uma classe é um tipo de dado definido, que tem um estado (sua representação) e algumas operações (seu comportamento), possui alguns dados internos e alguns métodos, na forma de procedimento ou funções, e descreve as características genéricas e o comportamento de vários objetos semelhantes. Um objeto é uma instância de uma classe ou uma variável do tipo de dados definido pela classe.

3.4.1. Definições

A estrutura de uma aplicação *Delphi* estabelece três categorias de componentes: componentes visuais, componentes de acesso à base de dados e componentes de ligação.

Os componentes visuais são componentes responsáveis pela interface com o usuário denominada Apresentação.

Os componentes de acesso à base de dados são componentes com finalidade de criar toda a estrutura necessária para acessar e manipular o banco de dados, possuem propriedades e eventos destinados a implementação da lógica da aplicação.

E os componentes de ligação são componentes responsáveis pela interface entre os componentes visuais e os componentes de acesso à base de dados. Sua principal característica é tornar os componentes visuais independentes dos componentes de acesso, ou seja, a apresentação da lógica da aplicação.

No desenvolvimento de uma aplicação é utilizado o recurso de *Data Module*, que é um componente que fornece uma localização centralizada para componentes não-visíveis do *Delphi*. Sua finalidade é conter a lógica da aplicação e relacionar com o banco de dados. Assim, toda a lógica fica concentrada em um único ponto da aplicação facilitando a manutenção e futuras alterações.

O *Delphi* possui componentes que permitem o acesso e manipulação dos dados no servidor. Um exemplo, é o *TQuery* que por ser baseado em uma sentença *SQL*, permite trabalhar com mais de uma tabela do banco de dados ao mesmo tempo.

3.4.2. Características

O ambiente de uma aplicação permite que esta seja distribuída entre dois componentes físicos: a estação cliente e o servidor de banco de dados. Entretanto, logicamente podemos identificar três partes distintas dentro de uma aplicação, como apresentação, lógica do negócio e gerenciamento dos dados.

A apresentação é composta por componentes responsáveis pela interação da aplicação com o usuário final. Assim, esta recebe os dados e comandos do usuário e devolve-lhe informações através de elementos visuais como inserção, consultas, gráficos, etc.

A lógica da aplicação recebe os dados da interface e executa as operações e validações necessárias para enviá-las ao banco de dados. Da mesma forma, extrai os dados do banco de dados de acordo com as regras da aplicação e os envia para elementos da interface, para que sejam exibidos. Portanto, é responsável em interligar a interface visual com o banco de dados através da execução de transações, consistência dos dados e regras, ou seja, a parte funcional da aplicação.

E por último o gerenciamento dos dados é responsável pelo acesso e a manipulação dos dados no servidor, onde grande parte deste gerenciamento, é implementado pelo próprio servidor de banco de dados. O acesso aos serviços é realizado através da linguagem *SQL*. Porém, é também necessário um conjunto de comandos para enviar as sentenças *SQLs* e gerenciar a comunicação entre a aplicação e o servidor.

3.5. Aplicação WEB

A aplicação *Web* utiliza a tecnologia *JavaServer Pages (JSP)* em *IDE NetBeans* (versão 4.0), o contêiner *Web Jakarta Tomcat* e o *drive JDBC* para conexão ao banco de dados.

A tecnologia *Java Server Pages (JSP)*²⁸ surgiu a partir da necessidade de se gerar conteúdo dinâmico para a *Web*, usando lógica, e dados no lado do servidor.

Antes do surgimento desta tecnologia, o processo era realizado por meio dos *servlets*, o que não era muito fácil, pois o *servlet* foi criado para trabalhar com código *Java* e não com *tags HTML*.

JSP faz parte da plataforma *J2EE (Java 2 Enterprise Edition)* e juntamente com o *Java Servlets* e *JavaBeans* é usado para desenvolver aplicações *web* eficientes, escaláveis e seguras²⁹.

Servlets são programas desenvolvidos na linguagem *Java* os quais rodam em um *Servlet Container*³⁰. Um Recipiente “*Container*” *Servlet* é como um servidor *Web* que trata requisições do usuário e gera respostas. Recipiente *Servlet* é diferente de Servidor *Web* porque ele é feito somente para *Servlets* e não para outros arquivos, como arquivos com extensão *html*, por exemplo. O Recipiente *Servlet* é responsável por manter o ciclo de vida do *Servlet*. Pode ser usado sozinho (*standalone*) ou conjugado com um servidor *Web*. Exemplo de Recipiente *Servlet* é o *Tomcat*, e de servidor *Web* o *Apache*.

Com *JSP* podemos criar aplicações *web* que se executam em vários servidores *web*, de múltiplas plataformas, já que *Java* é em essência uma linguagem multiplataforma.

As aplicações *JSP* seguem um padrão de estrutura de diretórios. Devem ter um diretório *WEB-INF*. Dentro desse, há outro diretório chamado *classes*. Nele constam todas as classes que pertencem a aplicação (arquivos *.class*), inclusive *servlets* e *beans*. Pacotes (arquivos *.jar*) são adicionados no diretório *WEB-INF/lib*. Os arquivos *JSP* são colocados no diretório raiz ou em qualquer outro diretório, menos no *WEB-INF* e sub-diretórios²⁹.

A página *JSP* é ótima para fazer o papel da visão (*view*), uma vez que possui facilidades para a inserção de componentes visuais e para a apresentação de informação, onde a participação de cada componente é descrita a seguir (FIG. 11).

Os *Servlets* atuam como controladores (*controller*), recebendo as requisições dos usuários. Após a realização das análises necessárias sobre a requisição, instancia-se o *JavaBean* armazenando-o no escopo adequado e encaminhando a requisição para a página *JSP*.

A *JSP* atua na apresentação utilizando os *JavaBeans* para obtenção dos dados a serem exibidos. A página não cria ou modifica o *bean*, simplesmente extrai e mostra os dados que o *servlet* criou.

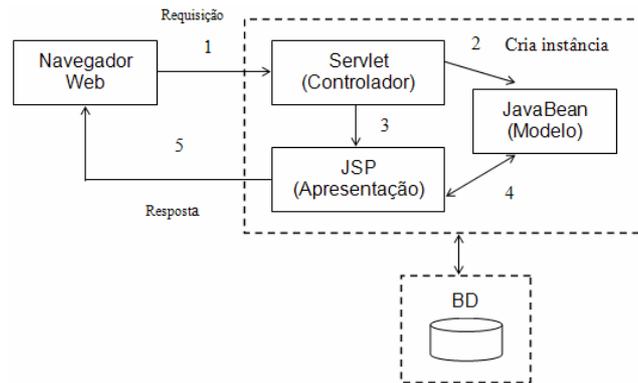


FIGURA 11: Arquitetura Aplicação Web.

Essa arquitetura possui vantagens como, a facilidade de manutenção onde a distribuição lógica das funções entre os módulos do sistema isola o impacto das modificações, e a escalabilidade, ou seja, controle de modificações necessárias para acompanhar o aumento da demanda de novas especificações.

3.5.1. Definições

As páginas *JSP* são compostas de código *HTML/XML* juntamente com a programação de *scripts* de servidor em sintaxe *Java*.

JSP utiliza a linguagem *Java*, onde esta é uma linguagem interpretada, ou seja, o código escrito em uma arquitetura pode ser portado para qualquer outra, portanto é compatível com qualquer plataforma que tenha uma *JVM (Java Virtual Machine)*³¹.

Esta tecnologia, *Java Server Pages*, permite que o código *Java* seja incorporado a páginas *HTML*, possibilitando que essas páginas sejam geradas dinamicamente.

Entretanto, *JSP* é uma combinação de:

- Conteúdo estático (expresso em *HTML*);
- Conteúdo dinâmico (expresso por meio de *tags JSP* + código *Java*).

Os arquivos *JSP* são publicados no servidor de aplicações da mesma forma que páginas estáticas, ou seja, o diretório de publicação é o mesmo. Um arquivo *JSP*, portanto, se parece muito com um arquivo *HTML*, diferenciando-se

apenas nos trechos onde o conteúdo deve ser gerado dinamicamente a partir de informações que estão no servidor.

Já o arquivo *web* com extensão *xml*, possui informações como, por exemplo, o tempo de validade da sessão, uma lista com nomes de arquivos que são abertos quando um diretório é acessado, e o ajuste das páginas de erro, para que não apareçam as causas do erro para o usuário. Há uma ordem necessária das configurações usadas neste arquivo *xml*. Trocando-se a ordem, a aplicação pode não funcionar.

A especificação de *JSP* é definida como uma extensão da *API* de *Servlets*, permitindo que páginas *JSP* sejam pré-compiladas em arquivos com extensão *class*, evitando demora no carregamento da página *JSP* na primeira vez que ela é acessada. Conseqüentemente, *servlets* e páginas *JSP* tem muito em comum. As páginas *JSP*, porem, estão sujeitas a uma fase de tradução e outra de processamento da requisição.

A fase de tradução é realizada apenas uma vez, a menos que a página *JSP* se atualize, e no caso é traduzida novamente. Supondo que não houve nenhum erro de sintaxe ou atualização na página, o resultado é uma página *JSP* que implementa a interface *Servlet*³². Por exemplo, quando uma solicitação para *JSP* é recebida, é verificado se existe compilação de *servlet* para a *JSP* com um registro de data/hora posterior ao *JSP* de origem. Caso esta verificação seja verdadeira, então é invocado o método de executar *servlet*, e em seguida enviada para o usuário. Caso contrário, o contêiner compila a *JSP* em arquivo com extensão *java* e *class*, instancia a classe, invocando assim o método de executar *servlet*.

Antes que esse *servlet* possa ser executado, um contêiner *Web* é requisitado, e neste caso utiliza-se o *Tomcat* que é um servidor de páginas *JSP* e *Servlets*. Desenvolvido pela fundação *Apache*, no projeto *Jakarta*, seu código é aberto e o programa é gratuito³³. Para sua instalação é necessário ter a máquina virtual *java* (*JDK* 1.3 ou mais atual) instalada no servidor³¹.

Considerando ainda outra definição de grande relevância, as convenções de *JavaBean* são o que nos permitem desenvolver *Beans*, porque elas permitem que o container *Bean* analise um arquivo de classe *Java* e interprete seus métodos como propriedades, designando a classe como um *Bean* de *Java*.

Um *Bean* é simplesmente uma classe de *Java* que segue um conjunto de determinações simples de *design* e nomeação delineado pela especificação de *JavaBeans*. Eles não precisam estender uma determinada classe ou implementar uma determinada interface, são objetos *Java*, mas como seguem um padrão de estrutura e definição fica mais fácil trabalhar com eles. É usual colocar *Bean* no nome de uma classe que é um *Bean*, para que seja melhor identificado.

O procedimento de conexão com a base de dados é feito através do *JDBC* (*Java DataBase Connectivity*) que é a *API Java* para comunicação com bancos de dados. Ele especifica a interface necessária para conectar a um banco de dados, executar comandos *SQL* e *queries*, e interpretar resultados. Esta interface é independente do banco de dados e da plataforma. As classes *JDBC* estão em um pacote, que precisa ser importado para a classe *Java* que faz uso de alguma classe desta *API*³⁰. Essa *API JDBC* não pode comunicar-se diretamente com o banco de dados. O acesso é fornecido por um *driver JDBC*, que é uma classe específica para o banco de dados e implementa a interface definida pela *API*.

Finalmente abordaremos alguns itens da plataforma *NetBeans*, conforme FIG.12, fornecendo uma base sólida de desenvolvimento de projetos e criação de módulos. Os principais recursos desta plataforma³⁴ são:

- Interface com o usuário (menus, barras de ferramentas e outros componentes);
- Editor rico em recursos para aplicações visuais *Swing* ou *AWT*, para aplicações *WEB* (*Servlets* e *JSP*);
- Gerenciamento da estrutura do seu projeto local ou remotamente.
- Escrito em 100% *Java*, portanto multi-plataforma;
- Ferramentas de gerenciamento de código, criação de *templates*.

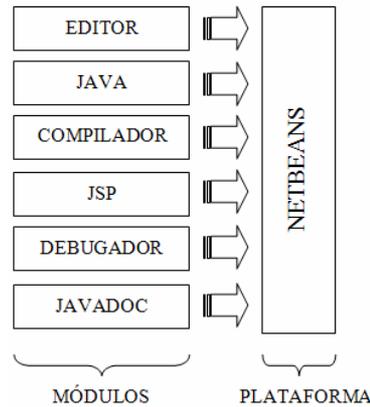


FIGURA 12: *Netbeans* – módulos.

3.5.2. Características

Na implementação para *Web* utiliza-se a tecnologia *JavaServer Pages* (*JSP*)²⁸, pois ela fornece uma maneira simplificada e rápida de desenvolvimento de aplicações *Web* dinâmicas e interativas com *Java*. O servidor utilizado foi o *Apache Tomcat*, que é responsável pela execução de páginas *JSP*.

JSP (*Java Server Pages*) é uma tecnologia para desenvolvimento de aplicações *Web* que possui portabilidade de plataforma, podendo ser executada em outros Sistemas Operacionais. Ela permite o acesso a banco de dados, o acesso a arquivos-texto, a captação de informações a partir de formulários, a captação de informações sobre o visitante e sobre o servidor, o uso de variáveis e *loops* entre outras coisas.

JSP pode parecer com *HTML* (ou *XML*), ambos contém texto encapsulado entre *tags*, que são definidas entre os símbolos < e >. Mas enquanto as *tags HTML* são processadas pelo navegador do cliente para mostrar a página, as *tags JSP* são usadas pelo servidor *web* para gerar conteúdo dinâmico.

O *JSP* não oferece nada que não se possa conseguir com os *servlets* puros. Entretanto, oferece a vantagem de ser facilmente codificada, o que facilita na elaboração e manutenção da aplicação. Além disso, essa tecnologia permite separar a programação lógica (parte dinâmica) da programação visual (parte estática), facilitando o desenvolvimento de aplicações robustas, onde programador e *designer* podem trabalhar no mesmo projeto, mas de forma independente. Promove-se assim o desenvolvimento rápido de aplicações e a

reutilização de código. Outra característica do *JSP* é produzir conteúdos dinâmicos que possam ser reutilizados.

Outra característica de impacto é que os *servlets* uma vez iniciados são mantidos na memória. Então toda requisição que chega, vai para o *servlet* na memória e este gera uma resposta. A característica de “manter na memória” faz com que usar *java servlets* seja um método extremamente rápido e eficiente para construir aplicações *Web*.

E por último aborda-se as características do ambiente integrado de desenvolvimento visual “*NetBeans IDE*” que é um conjunto de bibliotecas, módulos e *APIs*, que possibilita compilar, debugar, etc³⁴. Os seus recursos são: debugador e compilador de programas, suporta linguagens *Java*, *C*, *C++*, suporte à *XML* e *HTML*, *JSP*, *Servlets*, etc. Possui recursos para desenvolvimento *EJBs*, *Web Services*, e também total suporte ao *ANT* e *TOMCAT* integrado na *IDE*, indentação automática de código disponível, suporte a *Database*, *Data view*, *Connection wizard*, e sendo possível integrar módulos e extensões como o *J2ME Mobility*. É um produto *open source*, 100% *Java* e possui vários módulos de expansão (*modules*).

4. METODOLOGIA

4.1. Equipamentos e Softwares

No desenvolvimento deste trabalho foi feito um estudo inicial, para análise e compreensão dos *softwares* e equipamentos utilizados no LAAL. Este estudo envolve os princípios de operação do Sistema LIDAR, abrangendo o entendimento e operação do seu funcionamento. As especificações, bem como, a sua instrumentação são apresentadas no Anexo B.

Através da operação dos *softwares* responsáveis pelo controle do sistema foi possível entender o comportamento e funcionamento dele como um todo. O Anexo C mostra o *software* utilizado pelo sistema LIDAR.

Após o entendimento desses fatores iniciou-se a elaboração e desenvolvimento do sistema em questão, permitindo um ambiente visual de análises através de gráficos. Também utilizou-se para este fim uma especificação dos dados existentes, identificando seus tipos e classificações.

4.1.1. Princípios de operação

A operação do Sistema LIDAR, como apresentado na FIG.13, demonstra suas interfaces de funcionamento. Ela foi dividida em três etapas inter-relacionadas, visando um melhor entendimento: laser, aquisição de dados e divulgação dos dados.

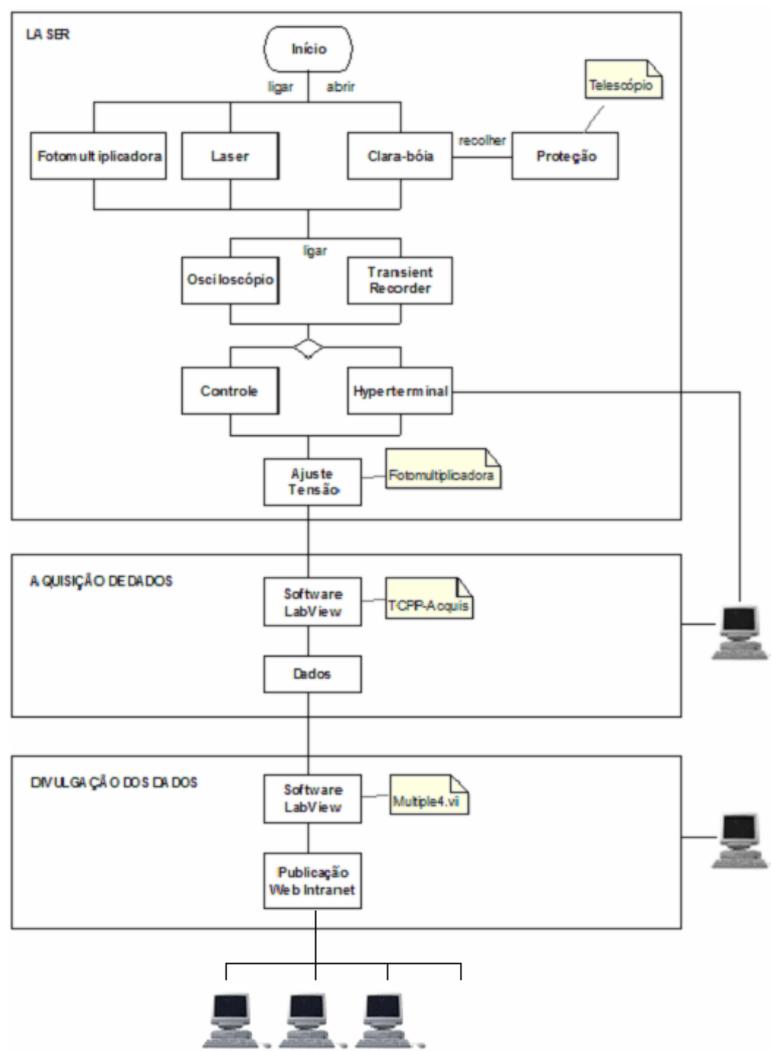


FIGURA 13: Operação do Sistema LIDAR.

A etapa laser é constituída pela fonte de luz (laser), telescópio, fotomultiplicadora, osciloscópio, registrador de transientes (*transcient recorder*) e computador.

A operação do laser em sensoriamento remoto tem como principais requisitos: coerência espacial e temporal, curta duração do pulso, alta taxa de repetição, baixa divergência e alta potência a fim de se obter boa resolução e longo alcance.

O feixe de laser é emitido para a atmosfera, ele interage com os constituintes durante sua propagação e a luz retroespalhada é coletada pelo telescópio. A energia retroespalhada é transmitida para a fotomultiplicadora, produzindo um sinal elétrico proporcional à intensidade óptica recebida. O sinal

elétrico contém informações sobre a presença e a distância dos espalhadores e absorvedores atmosféricos, cujos dados são digitalizados e armazenados em um sistema analógico ou digital, tais como osciloscópio e ou registrador de transientes.

O osciloscópio é utilizado para aquisições em modo analógico (baixas altitudes) e na verificação do alinhamento do sistema óptico. O registrador de transientes é utilizado para aquisições em modo analógico, contagem de *fótons* e na combinação dos dois tipos, alcançando melhor escala dinâmica, com resolução temporal elevada em taxas rápidas de repetição do sinal. O *HyperTerminal* é um programa utilizado para iniciar ou parar o feixe de laser remotamente através de um computador.

Na etapa de aquisição de dados o *software LabView* controla o registrador de transientes em taxas elevadas de transferência de dados. Ele é responsável pela configuração destes registros, pelo início e fim de aquisição através do módulo *TCP-PP-Acquis*. Os dados coletados são armazenados e visualizados graficamente com atualização em tempo real.

O funcionamento do sistema também conta com a divulgação gráfica dos dados em tempo real através do *software LabView* com publicação em formato *Web*.

4.2. Especificação dos Dados

O processo de levantamento de informações e especificação dos dados LIDAR define uma atividade, na qual ocorre uma intensa transferência de conhecimento sobre o domínio e a implementação do sistema. Podemos subdividi-la em quatro etapas para especificação e entendimento dos dados LIDAR:

- Entendimento do domínio do sistema: de forma geral, compreendem-se os requisitos de funcionamento do sistema LIDAR, com maior ênfase no contexto onde o sistema proposto está inserido;
- Compreensão do sistema proposto: onde há o entendimento dos detalhes que envolvem o sistema específico no qual ele será

aplicado. Nessa fase ocorre a especificação e extensão do conhecimento adquirido do domínio do sistema LIDAR;

- Percepção do negócio: o sistema desenvolvido contribui com agilidade, organização e segurança dos valiosos dados atmosféricos obtidos com o sistema LIDAR. Assim, as peculiaridades e relevância do sistema implementado e sua interação com as atividades do negócio precisam ser entendidas, para que se determine como elas podem contribuir para cumprir os objetivos do negócio;
- Estudo das necessidades e restrições: foi necessário entender as necessidades dos envolvidos com o sistema LIDAR. Em particular a influência no processo de trabalho, incluindo o papel de sistemas existentes.

A especificação dos Dados LIDAR se dividiu-se, então, em três partes: tipos, classificação e organização dos dados, conforme descritas nos itens subseqüentes.

4.2.1. Tipos

Os dados podem ser armazenados e processados em oito modos diferentes, utilizando o osciloscópio, o registrador de transientes (*transient recorder*) e a combinação dos dois como equipamentos de aquisição de dados.

O sinal do LIDAR se divide em três partes quando em contato com as partículas da atmosfera, sendo uma parte refletida, outra absorvida pelas partículas e a parte retroespalhada coletada pelo telescópio.

A FIG.14 apresenta os tipos de dados, classificados em modos de armazenamento de medidas coletadas pelo sistema LIDAR.

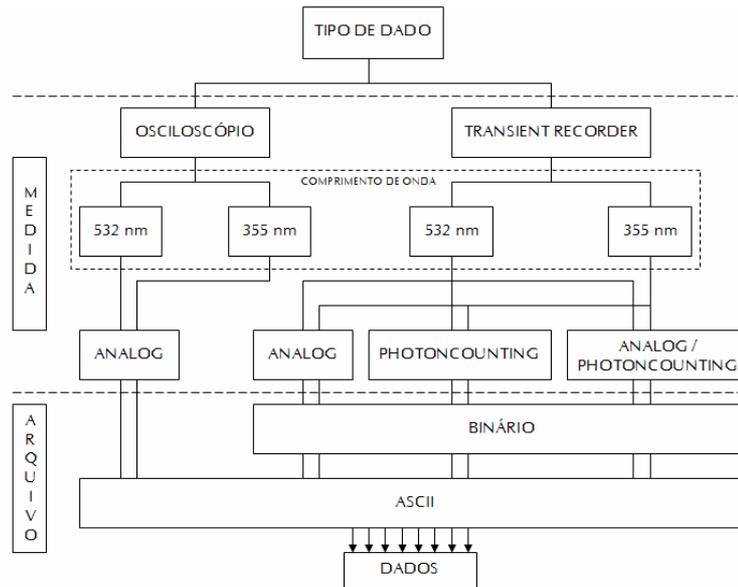


FIGURA 14: Classificação por tipos de dados coletados pelo LIDAR.

A medida pode ser coletada utilizando os comprimentos de ondas 532 nm, 355 nm e os dois em conjunto. Na aquisição de dados com o osciloscópio, o tipo de dado gerado é o modo analógico, observando-se distâncias próximas, de até 2000 m. Seu armazenamento ocorre em formato de arquivo *ascii*. Já o registrador de transientes, além do modo analógico, gera também o modo contagem de *fótons* (*photoncounting*) que corresponde a sinais retroespalhados em distâncias maiores, também combinando os dos dois modos. O armazenamento deste último tipo é em formato de arquivo binário, que é transformado em *ascii* para análise. O sistema de aquisição adquire os dados da medida em tempo real gerando um arquivo a cada 2 minutos.

4.2.2. Classificação

O critério de classificação adotado para o armazenamento das medidas coletadas, a partir de características específicas aos tipos determinados, permite a organização das observações destas dados, classificando-os como nível ^{2, 35} 0, 1 e 2.

O nível 0 caracteriza o dado cru no formato nativo (binário/ascii) ou convertido (ascii), classificando como não analisado. Estes dados são

visualizados graficamente em tempo real no sistema de aquisição de dados e podem ser armazenados no banco de dados juntamente com suas características, definidas na medição.

No nível 1 os dados são caracterizados como pré-analisados, podendo ser também armazenados no banco, os parâmetros de entrada e saída do perfil observado e o gráfico gerado.

O nível 2 caracteriza o dado analisado, armazenando no banco de dados tanto os parâmetros de entrada e saída como os gráficos. Classifica-se este dado em categorias³⁶ A, B e C, de acordo com algumas características comuns dos perfis verticais do coeficiente de retroespalhamento. Essa categorização segue um critério com relação ao número de camadas em diferentes alturas, a fontes potenciais de poluição e a meteorologia³⁷.

A categoria A apresenta somente uma camada de aerossol abaixo de 1,5 km, com a fonte potencial destes aerossóis originada de regiões vizinhas do local onde o LIDAR está situado, possuindo, quanto à meteorologia, dias provavelmente ensolarados, precedidos de chuva com “lavagem” dos poluentes, pouco vento, alta amplitude de temperatura e boas condições de visibilidade.

A categoria B apresenta mais camadas de aerossóis entre 1,5 e 2,5 km, com fontes potenciais de aerossóis originada de fontes locais de poluição ou acumulada de dias anteriores. Apresenta, quanto à meteorologia, dias provavelmente sem chuva, vento moderado na superfície e em altas altitudes, com visibilidade prejudicada.

Finalmente, a categoria C apresenta mais camadas acima de 2,5 km, além das presentes abaixo, com fontes potenciais de aerossóis originadas do transporte de poluição de áreas mais distantes, possuindo, quanto à meteorologia, dias provavelmente muito secos, com condições ruins para dispersão de poluentes, ocorrência de inversões térmicas, sem chuva e com ausência de vento.

O acesso aos resultados da análise de nível 0 e 2, estão restritos aos integrantes do grupo LAAL, já o acesso aos resultados da análise de nível 1, estão restritos a intranet do IPEN.

4.2.3. Organização

O perfil vertical de aerossóis tem um papel importante na caracterização da poluição atmosférica. O LIDAR permite coletar dados espaciais e temporais deste perfil. Estes dados são armazenados em um sistema de arquivos, seguindo uma estrutura de diretórios, inicialmente em ordem de datas, com ano, mês e dia. A vantagem deste método de armazenamento é a sua simplicidade, entretanto há uma dificuldade em controlar a redundância e integridade destes dados. Não há também nenhuma indicação de relacionamento entre os dados, e o usuário precisa saber exatamente como os dados estão armazenados para poder acessá-los de forma consistente.

Então, a organização das informações se faz necessário devido ao grande volume de dados coletados e já armazenados. Desde 2001, ano em que o sistema LIDAR foi criado, tem-se coletado valiosas informações de propriedades atmosféricas. A TAB. 3 mostra a quantidade de dados LIDAR armazenados em arquivo e organizados por data. Tornou-se imprescindível a criação do banco de dados para o armazenamento destas importantes informações.

TABELA 3: Dados LIDAR.

Ano	Dias	Aquisições
2005	36	8.079
2004	64	9.037
2003	50	1.426
2002	54	3.281
2001	30	518
Total =	234	22.341

Com a implementação do novo modelo, estes dados foram organizados por tipos que identificam características definidas no sistema de aquisição, sendo hierarquizados através de diretórios, seguindo uma estrutura estabelecida. Esta estrutura é composta por níveis de dados, tipo de equipamento, tipo de dado e datas, tendo além desta nova hierarquia de diretórios, o armazenamento destas informações em um banco de dados.

5. RESULTADOS E DISCUSSÕES

5.1. Banco de dados LIDAR

O banco de dados LIDAR consiste de um servidor que fornece dados armazenados ao sistema gerenciador de banco de dados, cujo aplicativo demonstra de forma visual as informações coletadas da atmosfera através do sistema LIDAR. O servidor do banco de dados e o sistema gerenciador podem estar em uma única máquina, ou o banco de dados em uma máquina e o sistema gerenciador distribuído por várias máquinas. A princípio, o sistema está configurado para uma única máquina.

Os dados do servidor residem em um espaço de endereços no banco de dados, ou seja, os endereços físicos e lógicos são *strings* de *bytes* que nos permitem determinar a localização dentro do sistema de armazenamento no qual o registro pode ser encontrado.

A transação de dados no servidor é realizada seguindo estas etapas:

- Leitura – A transação lê, a partir do banco de dados, todos os dados em seu conjunto de leitura. Ela também calcula em seu espaço de endereços local todos os resultados que irá gravar.
- Validação – A validação da transação é efetuada comparando seus conjuntos de leitura e gravação com o de outras transações. Se a validação falhar, a transação será revertida; caso contrário, ela prosseguirá até a gravação.
- Gravação – A transação grava no banco de dados os valores para os dados de seu conjunto de gravação.

Neste trabalho, conforme já mencionado anteriormente, definiu-se para implementação do banco de dados LIDAR, o SGBD *PostgreSQL* (versão 8.0). Considera-se necessários, na descrição a seguir, alguns detalhes referentes ao sistema de catálogo do *PostgreSQL*, que é responsável por armazenar toda a estrutura funcional do banco de dados LIDAR, criado e mantido por ele, o

chamado *metadados*. Este catálogo é composto por uma estrutura organizacional do banco de dados LIDAR em tabelas, onde são guardadas informações como: nome do banco de dados, de tabelas, de colunas das tabelas, de tipos de colunas, chaves, usuários, privilégios de usuários, índices, restrições, etc.

Portanto, o catálogo do PostgreSQL é composto por tabelas regulares, totalmente gerenciadas e manipuladas pelo próprio PostgreSQL, baseado em todas as definições criadas e manipuladas via SQL.

5.1.1. Características da elaboração

O banco de dados objeto relacional LIDAR é um sistema de armazenamento de dados e acesso controlado destes dados. Seus principais objetos utilizados nesta implementação, são tabelas e índices. As tabelas armazenam os dados fisicamente, e os índices são ferramentas usadas pelo gerenciador para facilitar a busca de registros dentro de uma tabela, melhorando o desempenho do banco. Para isto utilizamos o *pgAdmin III* que é uma plataforma de desenvolvimento e administração para o banco de dados *PostgreSQL*.

Os índices²⁶ *B-tree*, utilizados neste banco, tratam consultas de igualdade e de faixa, em dados que podem ser classificados em alguma ordem, sempre que uma coluna indexada está envolvida em uma comparação. Estes índices também são utilizados para obrigar a unicidade do valor de uma coluna, ou a unicidade dos valores combinados de mais de uma coluna, como é o caso do campo “nome do arquivo” pertencente à identificação de dados, isto faz com que o índice seja declarado como único, não podendo existir na tabela mais de um registro com valores indexados iguais.

Na definição das chaves primárias, chave esta que permite a classificação única de cada registro de uma tabela, ou da restrição de unicidade, o índice único criado a partir da chave primária não permite a inclusão de linhas duplicadas. Essa restrição abrange as colunas que compõem a chave primária ou as colunas únicas, sendo este o mecanismo que impõe a restrição.

5.1.2. Estrutura

A estrutura do banco de dados criado, bem como as várias classes, as chaves de identificação (primária e estrangeira), mostrando a integridade dos dados, e por fim os relacionamentos entre elas, pode ser visualizada na FIG.15. As classes são descrições de conjuntos de objetos que compartilham os mesmos atributos, operações e relacionamentos.

A classe “*identifica*” representa a tabela onde serão armazenadas as identificações das medidas, já a classe “*dados*” lida com os dados coletados referentes ao nível 0. Estes são visualizados graficamente e armazenados na classe “*gráfico nível 0*”.

O nível 1 é representado pelas classes “*resultado da pré-análise*”, “*entrada de dados da pré-análise*” e “*gráfico nível 1*”. Na classe “*resultado da pré-análise*” os atributos definidos são resultados de cálculos geradores dos eixos x e y do gráfico referenciado na classe “*gráfico nível 1*”, sendo a classe “*entrada de dados da pré-análise*” responsável por valores de entrada utilizados na classe “*resultado*”.

No nível 2, a classe “*parâmetros*” fixa os parâmetros de entrada e saída da análise, sendo esta visualizada graficamente pela classe “*gráficos analisados de nível 2*”, onde seus resultados são atributos da classe denominada “*sp*”. A classe “*gráficos nível 2*” determina os atributos dos dados e gráficos pertencentes ao nível 2.

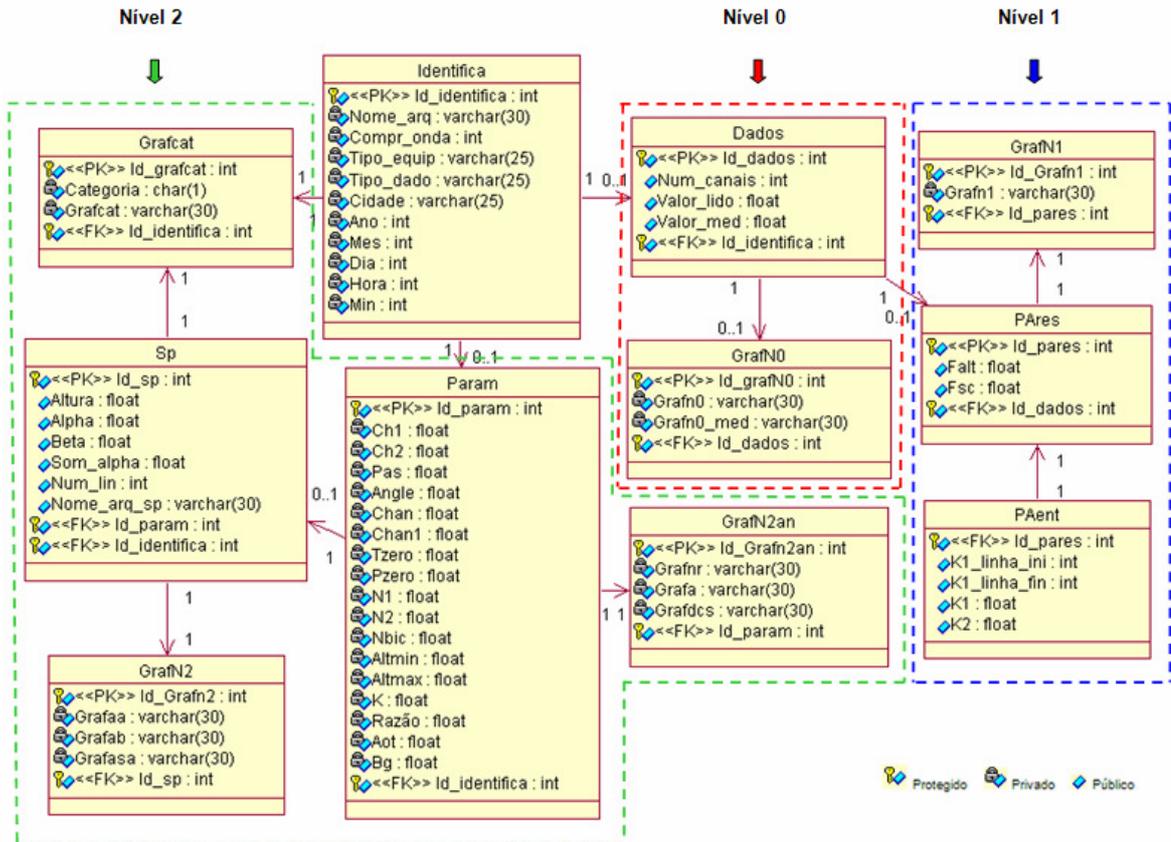


FIGURA 15: Diagrama do Banco de Dados LIDAR.

5.1.3 Consultas

O servidor de banco de dados *PostgreSQL*, utilizado neste trabalho, consiste de um compilador de consultas na Linguagem de Consulta Estruturada (SQL). Quando ele é ativado, se comunica com o Sistema de Análise de Dados LIDAR (SADL) e com o *Web LIDAR*. Para executar uma consulta, tanto o SADL como o *Web LIDAR* se conectam ao *PostgreSQL*, através dos métodos de acesso *ODBC* e *JDBC*, respectivamente.

A interface do servidor de banco de dados possibilita que sejam elaboradas consultas, utilizando funções na linguagem SQL, substituindo assim métodos de acesso a banco de dados. Essa função é “*execute sql query*” – que executa um comando “*select*”. Para este método de consulta podem ser fornecidos diversos parâmetros, como o nome do arquivo, o tipo de dado, o gráfico, a data, o valor lido, etc. Ao receber a consulta, o servidor de banco de dados fará um

processamento, compilando-a. Assim podemos fornecer uma consulta como é exemplificado a seguir:

```
select * from tb_identifica
where ano = 2005 and
      tipo_equip = 'Osciloscópio' or tipo_equip = 'Transient Recorder' and
      compr_onda = 532 or compr_onda = 355
```



id_identifica (int4)	nome_arq (varchar)	compr_onda (int4)	tipo_equip (varchar)	tipo_dado (varchar)	cidade (varchar)	ano ...	mes ...	dia...	hora...	min...
26	D11M4A2005-H10M51S34.bt	532	Osciloscópio	Analog	São Paulo	2005	4	11	10	51
27	SP0504111051	355	Transient Recorder	Analog e Photoncounting	São Paulo	2005	4	11	10	51

Uma outra função é a execução de um comando de atualização do banco de dados, onde são permitidos também os comandos “*insert*”, “*update*” e “*delete*”. Nesta função testamos a inserção de dados ao banco, onde a validação e a integridade destes são garantidas devido ao estabelecimento de regras através da criação de índices e chaves primárias e estrangeiras, conforme exemplo abaixo:

```
CREATE UNIQUE INDEX inumcalident
ON tb_dados
USING btree
(num_canais, id_identifica);
```

5.2. Sistema de Análise de Dados LIDAR - SADL

O Sistema de Análise de Dados LIDAR provê a interface entre os dados armazenados no banco de dados e as solicitações submetidas ao sistema, sendo responsável pelas tarefas descritas a seguir.

Na interação com o gerenciamento de arquivos, os dados coletados através do equipamento LIDAR, em conjunto com o sistema de aquisição de dados, que utiliza o gerenciador de arquivos *LabView*, são armazenados em um sistema de arquivos, seguindo uma estrutura de diretórios. O SADL lê as informações dos diversos arquivos armazenados e envia para o banco de dados. Portanto, o SADL é responsável pelo armazenamento, busca e atualização dos dados ao banco.

Para não haver duplicidade de dados, um cumprimento de integridade é referenciado com as informações de dados LIDAR armazenados no banco,

satisfazendo algumas restrições de consistência. Por exemplo, o arquivo de dados só poderá ser cadastrado nos níveis, se o mesmo já estiver cadastrado em identificação de dados. O sistema gerenciador de banco de dados pode então determinar se as atualizações no banco de dados resultam numa violação da restrição; em caso positivo, ações apropriadas são tomadas.

Para impor requisitos de segurança, há controle de acesso, através de permissão ao usuário, pois nem todo usuário do banco de dados necessita ter acesso a todo o banco. Em princípio, está configurado um usuário com permissões definidas e estabelecidas, já que o acesso à análise é efetuado somente ao grupo LAAL.

Com relação à segurança do sistema, a seguir são mencionados métodos exemplificados para tal. Um computador, como qualquer outro dispositivo mecânico ou elétrico, está sujeito à falha. As causas das falhas incluem quebras de disco, falhas na energia e erros de *software*. Em cada um dos casos, as informações que se referem ao banco de dados podem ser perdidas. É de responsabilidade do sistema gerenciador do banco de dados (*PostgreSQL*) a recuperação e o restabelecimento do banco de dados em seu estado anterior ao da ocorrência da falha. Isto é realizado através de Cópias de reserva (*backup*) e recuperação (*restore*).

Outra situação introduzida neste estudo, é a de haver a possibilidade de instalação e configuração deste SGBD em várias máquinas. Com diversos integrantes do grupo utilizando o banco de dados concorrentemente, deve haver um controle entre usuários simultâneos, que é outra atribuição de grande relevância do gerenciador de banco de dados (*PostgreSQL*).

5.2.1. Características da elaboração

O Sistema de Análise de Dados LIDAR, utiliza a linguagem de programação *Delphi* (versão 7.0) como interface gráfica de acesso ao servidor de banco de dados *PostgreSQL*. As telas desenvolvidas neste *software*, possuem interfaces gráficas definidas como, identificação de medidas e níveis de dados, estabelecendo relacionamentos entre elas.

O *Delphi* é uma ferramenta dotada de um *IDE (Integrated Development Environment – Ambiente Integrado de Desenvolvimento)*, fundamentada no *Pascal* Orientado a Objetos, com objetos prontos para acessar banco de dados *SQL* em conjunto com o administrador *Borland Database Engine (BDE)*, tendo este a conectividade entre o software e o banco de dados, e sendo também responsável por todo o processamento realizado nos dados.

Neste aplicativo, utilizou-se acesso exclusivo *SQL*, portanto com *queries* e objetos ligados à conexão com o *PostgreSQL*.

É importante ressaltar que no decorrer do texto, quando é citado *SADL*, este se refere ao Sistema de Análise de Dados *LIDAR (Delphi)*, o qual é o responsável por analisar dados e manipular as informações contidas no banco de dados (*PostgreSQL*).

As principais telas do *SADL*, são mostradas em seguida, com enfoque a suas funcionalidades em figuras, o que garante melhor compreensão dos resultados.

5.2.2. Interface gráfica do Sistema de Análise de Dados LIDAR

O Sistema de Análise de dados *LIDAR* foi implementado seguindo alguns critérios de desenvolvimento e validação de dados. Na elaboração das interfaces gráficas, estes critérios foram estabelecidos gerando identificação e níveis de dados.

Foi determinado que os dados não podem ser cadastrados e validados como níveis, caso não estejam gravadas as suas respectivas identificações de dados, isto é, cada dado pertence a uma determinada medida, cuja medida possui parâmetros estabelecidos no processo de aquisição de dados do sistema *LIDAR*.

5.2.2.1. Identificação dos dados

A interface gráfica de identificação dos dados é utilizada tanto para efetuar o cadastro de parâmetros, definidos na configuração do sistema de aquisição de dados, como também para consulta das identificações pertencentes a estes

dados. Estes parâmetros de identificação da medida são encontrados na estrutura do cabeçalho do arquivo.

A FIG. 16 mostra a interface gráfica de identificação das medidas. As letras indicam os dados de entrada, e os números apontam as interfaces com outras telas, conforme detalhes a seguir:

- A: Nome do arquivo de dados;
- B: Tipo de equipamento (Osciloscópio ou registrador de transientes);
- C: Tipo de dado (analógico, contagem de fótons ou a combinação dos dois);
- D: Cidade onde foi realizada a coleta do dado;
- E: Data (ano / mês / dia / hora / min);
- F: Comprimento de onda;
- G: Inserção ou visualização de dados.

As interfaces definidas são mostradas por números estabelecendo os seguintes níveis de dados:

- 1 - Tela de dados referentes ao nível 0;
- 2 - Tela de dados referentes ao nível 1;
- 3 - Tela de dados referentes ao nível 2

nome_arq	compr_onda	tipo equip	tipo_dado	cidade	ano	mes	dia	hora	min
D23M8A2002-H7M3553	532	Osciloscópio	Analog	São Paulo	2002	8	23	7	3
D6M3A2003-H9M55556	532	Osciloscópio	Analog	São Paulo	2003	3	6	9	55
SP0408060914a	532	Transient Recorder	Analog	São Paulo	2004	10	7	18	5
SP0408061615a	532	Transient Recorder	Analog	São Paulo	2004	10	14	18	3
SP0504111051	365	Transient Recorder	Analog e Photoncounting	São Paulo	2005	4	11	10	51

FIGURA16: Tela de identificação das medidas.

5.2.2.2. Níveis de dados

Os perfis característicos de análise de dados LIDAR foram identificados, e então foram estabelecidos três níveis, sendo determinados como nível 0, nível 1 e nível 2.

Os dados de nível 0, contêm o dado cru do sinal do LIDAR, e vários parâmetros de diagnósticos desde sinal. Os arquivos de dados deste nível são coletados em tempo real e armazenados pelo computador no sistema de aquisição de dados. O sistema LIDAR mede o retroespalhamento do sinal, onde a intensidade deste sinal é controlada pela fotomultiplicadora para obtenção de um retorno não saturado.

A FIG. 27 apresenta a interface gráfica do software no nível 0, mostrando o processo de seleção e visualização de medidas coletadas com transferência para o banco de dados e a geração gráfica do sinal comparado com o canal.

O detalhamento da seleção de arquivos coletados pelo equipamento LIDAR estão contidos na TAB. 4.

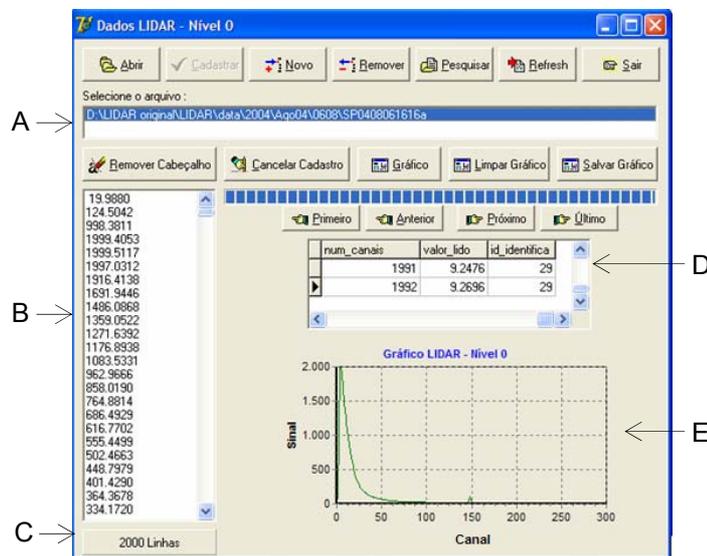


FIGURA 17: Tela de seleção, visualização, transferência de arquivos.

TABELA 4: Detalhamento da seleção de arquivos coletados.

Legenda	Comentário
A	Seleção de arquivos referente a medidas realizadas pelo equipamento LIDAR
B	Dados do arquivo selecionado
C	Número de canais de dados
D	Transferência dos dados do arquivo selecionado para o Banco de Dados
E	Visualização gráfica de dados nível 0

Considerando que as medidas coletadas geram um arquivo em média a cada dois minutos, temos um grande volume de arquivos de dados LIDAR. Porém, devido a essa quantidade elevada de dados, há a necessidade de inserção e manipulação de dados por diretório de dados, para se obter uma análise por um período de tempo determinado, lembrando que a possibilidade de análise através de determinado dado ou arquivo continua válida, caso seja necessário este nível de detalhamento específico.

Uma consistência foi estabelecida, então, após a abertura da tela de identificação das medidas. Esta consistência é feita através da aparição de uma caixa de diálogo onde o usuário define se a manipulação dos dados será por diretório ou arquivo. Diante disso, se ficar estabelecido que será por arquivo, nesse caso, a tela de identificação das medidas será ativada com todas as opções habilitadas, caso contrário, estarão habilitadas somente algumas opções, e no momento da confirmação, a tela de manipulação por diretórios será ativada.

As figuras a seguir mostram as telas de dados por diretório, contemplando a utilização das medidas através da análise por períodos determinados, bem como seus tipos de dados coletados.

Os períodos determinados pertencem ao intervalo da medida, ou seja, o período I foi determinado tendo como início, o horário matutino até o seu término. Já o período II foi determinado tendo como início o horário vespertino até o término da coleta, independentemente do horário. E o período III foi determinado tendo como início o horário noturno até o encerramento da medida.

Há casos em que podemos ter período I com arquivos coletados no horário matutino até vespertino, e período II no horário vespertino até noturno. Os horários são controlados pelo nome do arquivo coletado, sendo composto pelo ano, mês, dia, hora, minuto e segundo.

Inicialmente, é acionado o evento do botão de leitura de dados, onde são lidos os valores obtidos pelo sistema LIDAR, como também o seu tipo. Após a leitura destes dados, é gerado um novo arquivo com cálculo resultante do período determinado lido. Finalmente é plotado o gráfico dos dados (unidade arbitrária - u.a.) versus canal.

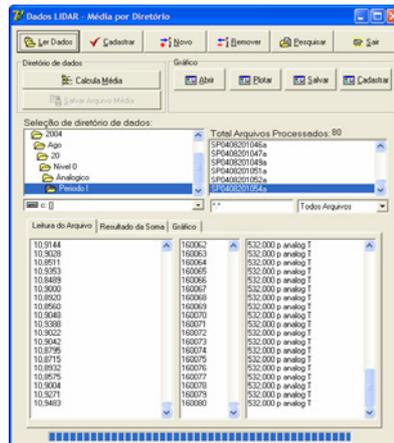


FIGURA18: Leitura de dados analógicos período I

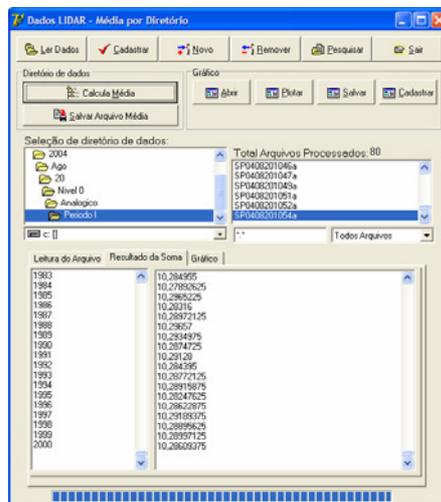


FIGURA 19: Cálculo resultante de dados analógico período I.

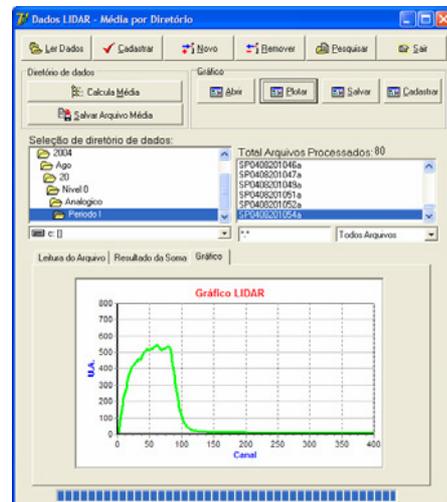


FIGURA 20: Gráfico de dados analógico período I (U.A. vs Canal).

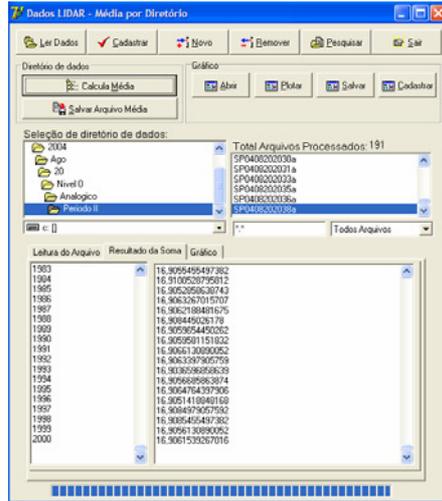


FIGURA 21: Cálculo resultante de dados analógico período II.

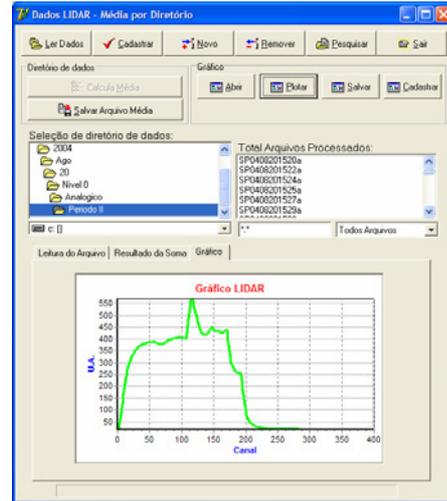


FIGURA 22: Gráfico de dados analógico período II (U.A. vs Canal).

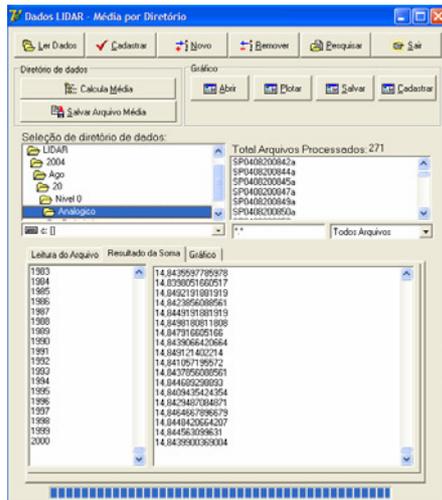


FIGURA 23: Cálculo resultante de dados analógico período I e II.

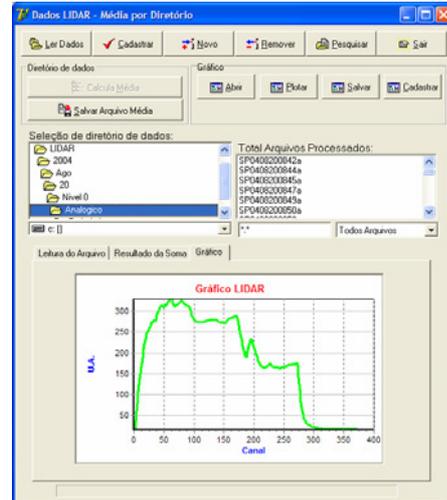


FIGURA 24: Gráfico de dados analógico período I e II (U.A. vs Canal).

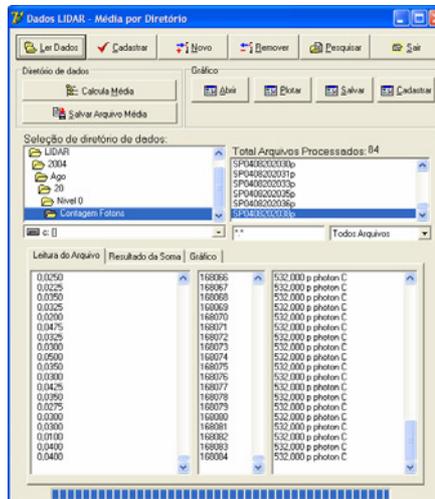


FIGURA 25: Leitura de dados contagem de fótons período II.

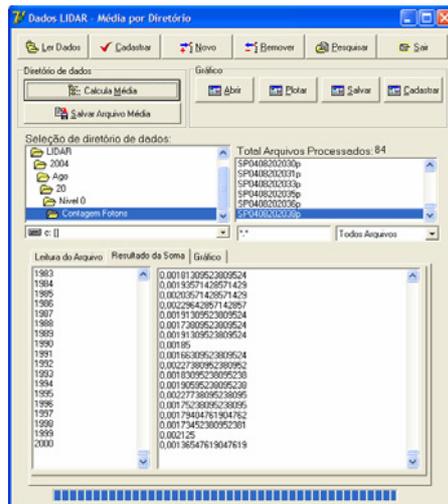


FIGURA 26: Cálculo resultante de dados contagem de fótons período II.

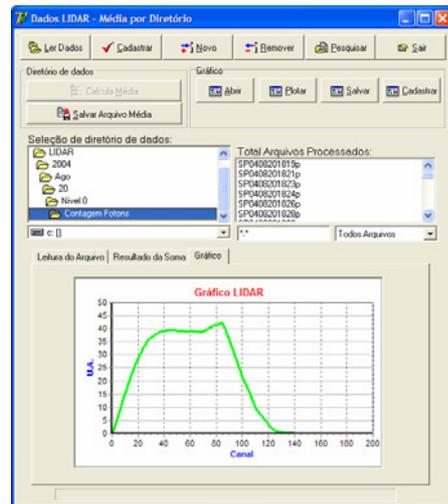


FIGURA 27: Gráfico de dados contagem de fótons período II (U.A. vs Canal).

No caso, leitura de dados contagem de fótons período II, exemplificado anteriormente, temos arquivos coletados no horário vespertino até o noturno.

Os dados de nível 1, são os dados de nível 0 que foram processados e pré-analisados em função da pesquisa de parâmetros atmosféricos relacionados a medida. O resultado desta pré-análise nos dados de nível 0, resulta em um gráfico da altura (km) pelo alcance do sinal corrigido, incluindo a radiação de fundo e a resolução como parâmetros de entradas para a obtenção do cálculo, para geração de dados de nível 1.

A seguir é apresentada a interface gráfica de dados de nível 1, assim como os dados resultantes da pré-análise que podem ser obtidos com resoluções diferentes (FIG.28). Esta pré-análise é requerida, pois permite a visualização gráfica da estrutura vertical de aerossóis.

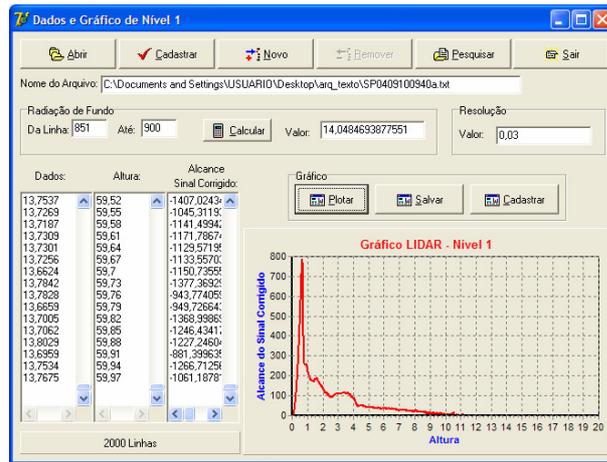


FIGURA 28: Interface gráfica da pré-análise de dados LIDAR.

Os dados de nível 2 são identificados como análise de dados e incluem o sinal corrigido com a distância, o sinal molecular normalizado³⁸, a derivada do logaritmo do sinal, o perfil do coeficiente de extinção pela altura, o perfil vertical do coeficiente de extinção α , o perfil do coeficiente de retroespalhamento β e a espessura ótica de aerossóis^{38,39}.

Os dados de nível 2 são obtidos através do programa de análise que utiliza o método de inversão de *Klett*⁴⁰ para a solução da equação de LIDAR. Para tal são utilizados parâmetros para obtenção de resultados. Os parâmetros de entrada e saída, conforme descritos na TAB. 5, são armazenados indicando os valores referentes à análise. Os valores resultantes, pertencentes a cálculos gráficos, também são arquivados para posteriores análises, como ocorre com seus respectivos gráficos. A seguir é apresentada a interface gráfica do programa de análise, mostrando alguns dados analisados.

TABELA 5: Detalhamento dos parâmetros de entrada e saída da análise de dados.

LEGENDA	FUNÇÃO
canal inicial, canal final	Região de fundo, entrada em canais.
passo, ângulo	Passo ou resolução, de 0,015 a 0,060 km.
canal inicial, canal final	Região de cálculo, entrada em canais.
TZero, PZero	Temperatura e pressão na superfície.
N1, N2 NBic	Dados de filtragem, utilização de filtros polinomiais e binomiais.
Alt. Min., Alt. Max., <i>Klett</i>	Alturas em que se encontram o sinal molecular e a altura de inversão de <i>Klett</i>
Razão	Razão entre o retroespalhamento e extinção (RL).
BG	Valor de BG variável utilizado, quando o fundo de escala máximo for de 7,5 km ou menos.
Gráfico <i>Norm Rayleigh</i> (linha superior)	Sinal corrigido com a distância.
Gráfico <i>Norm Rayleigh</i> (linha inferior)	Sinal molecular normalizado.
Gráfico DCS	Derivada do logaritmo do sinal.
Gráfico Alpha	Perfil do coeficiente de extinção pela altura.
AOT	Espessura Óptica de Aerossóis (EOA)

As figuras abaixo mostram alguns dados analisados e seus respectivos gráficos.

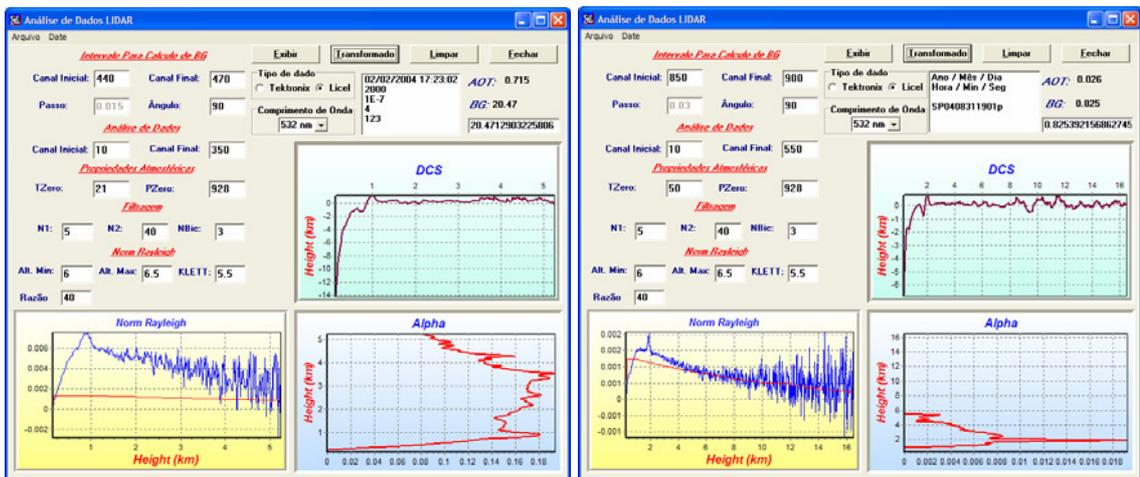


FIGURA 29: Análise de dois arquivos de dados referente à medida: *Licel*, comprimento de onda 532nm e contagem de *fótons*.

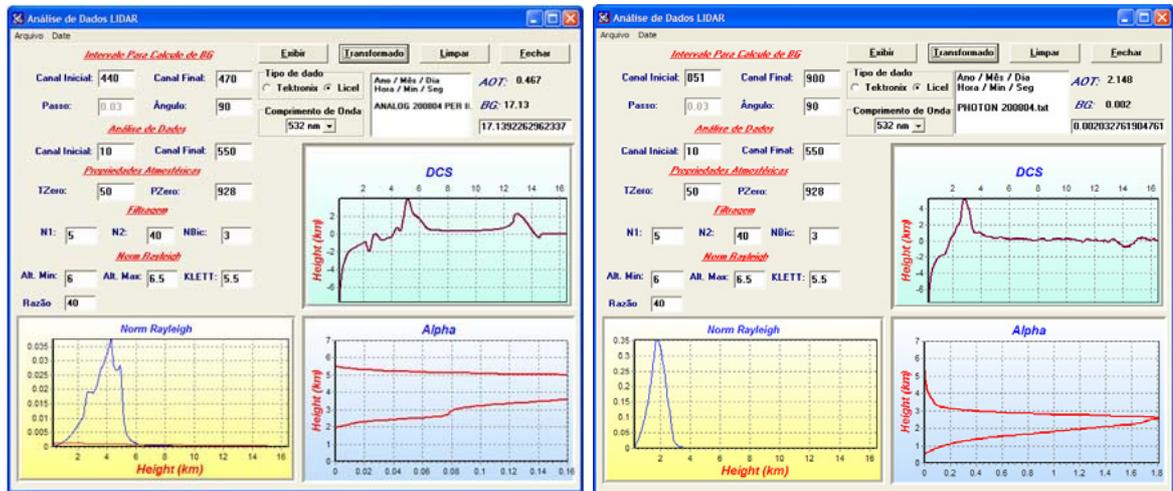


FIGURA 30: Análise de períodos de dados referente à medida: *Licel*, comprimento de onda 532nm, analógico e contagem de *fótons*.

As especificações de níveis 2, referentes as FIG. 31, 32 e 33, apresentadas posteriormente são detalhadas a seguir (TAB. 6).

TABELA 6: Especificação de análise LIDAR.

Legenda	Comentário
PA	Parâmetros de análise de dados LIDAR
G1	Perfil vertical do coeficiente de extinção α
G2	Perfil do coeficiente de retroespalhamento β
G3	Espessura ótica de aerossóis
G4	Sinal molecular normalizado
G5	Sinal corrigido com a distância
G6	Derivada do logaritmo do sinal

A FIG. 31 apresenta a tela de armazenamento e visualização de parâmetros da análise, detalhados anteriormente na TAB.5. Os parâmetros armazenados, identificados como nível 2, são obtidos através do programa de análise. Essa tela possui uma interface com os dados e gráficos obtidos no programa de análise e os gráficos gerados a partir destes dados analisados.

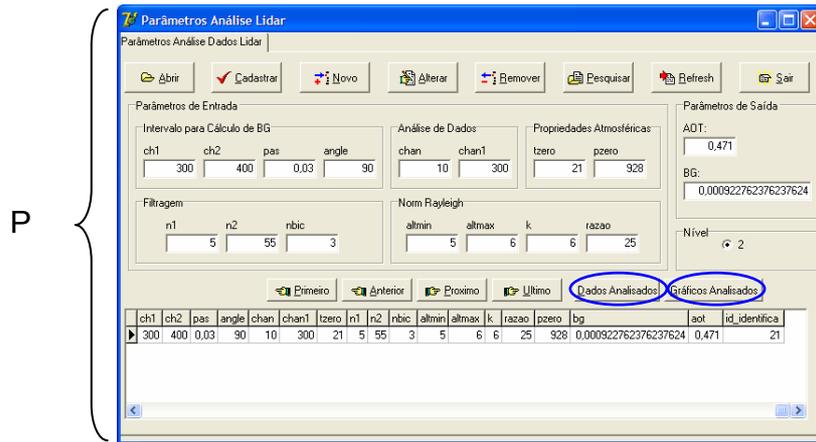


FIGURA 31: Interface de Parâmetros de Análise de Dados LIDAR.

Através dos dados analisados são gerados os gráficos referentes ao perfil vertical do coeficiente de extinção α , perfil do coeficiente de retroespalhamento β e espessura óptica de aerossóis. A FIG. 32 contempla o resultado desta análise no formato gráfico tendo como parametrização a altura (km).

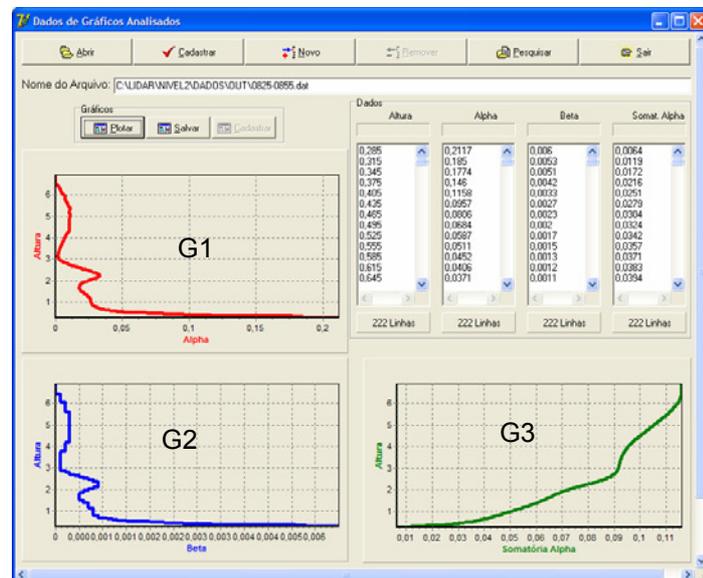


FIGURA 32: Tela de seleção, visualização, geração e transferência de dados e gráficos de nível 2.

Para controle de dados analisados e não analisados, os gráficos obtidos são armazenados e relacionados com seus respectivos dados. Terão acesso a estas informações, somente os integrantes do grupo LAAL, cuja funcionalidade

seja análise de dados. Na FIG. 33 são visualizados os gráficos de sinal corrigido com a distância e sinal molecular normalizado, e a derivada do logaritmo do sinal.

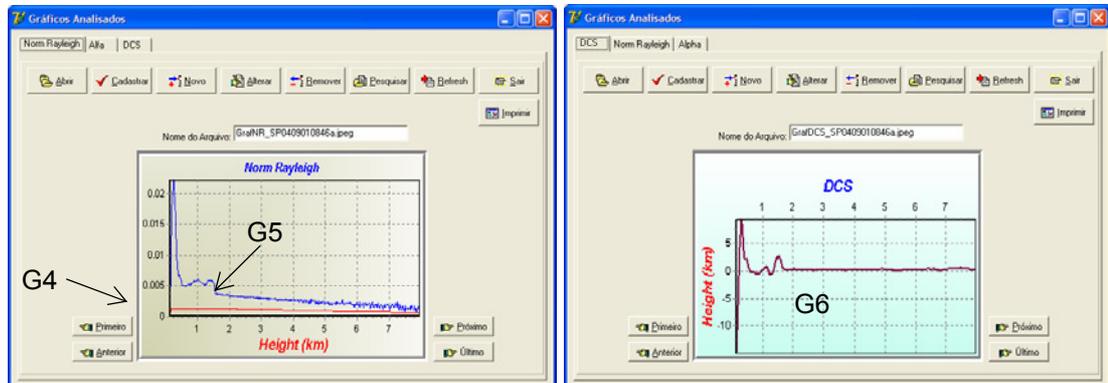


FIGURA 33: Armazenamento e visualização de gráficos analisados.

Ainda como nível 2 temos a categorização dos dados, sendo estes após análise, também armazenados no banco. A FIG. 34 exibe um exemplo de categoria C, com altura (km) pelo coeficiente de retroespalhamento ($\text{km}^{-1}\text{sr}^{-1}$).

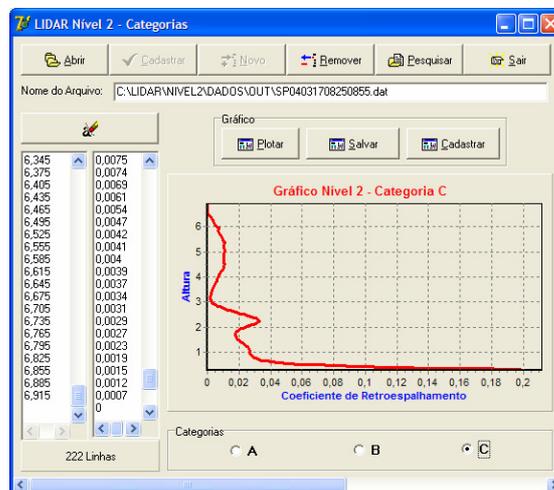


FIGURA 34: Gráfico Nível 2 de categoria C.

5.3. Web LIDAR

A implementação para *Web* utiliza a tecnologia *JavaServer Pages (JSP)*, que permite servir conteúdo dinâmico em um *site Web* e o contêiner *Web Jakarta Tomcat*, por ser livre e facilmente acessível.

Na definição desta aplicação, estabeleceu-se como critério ideal, referente ao escopo, a abordagem de um padrão de tecnologia poderosa e escalonável. Como evidência disto, salienta-se a recompilação automática, que permite que alterações feitas no código da página sejam automaticamente visíveis em sua apresentação. Assim, não é necessário interromper o funcionamento da aplicação para incorporar uma modificação de *layout*, por exemplo. Isso torna este critério válido para a implementação desenvolvida, destacando a sua simplicidade, facilidade de acesso, manutenção e a possibilidade de ampliação da aplicação beneficiando futuras pesquisas.

O arcabouço de desenvolvimento possui, portanto, uma estrutura para futuras regras e especificações dentro do trabalho proposto.

Este *Web LIDAR* permite consultas a dados LIDAR, referentes aos níveis 0, 1 e 2 de análise, tendo, como visualização do resultado da busca, gráficos em formato *jpeg*, pertinentes a estes dados.

5.3.1. Características da elaboração

No desenvolvimento desta aplicação *Web*, a tecnologia *JavaServer Pages (JSP)* é implantada em um contêiner que processa *JSP*. Este segue uma estrutura de diretórios para o seu funcionamento.

Para esta implementação foi criado um diretório *WEBINF*. Dentro desse, há outros subdiretórios e arquivos pertencentes à aplicação. Pacotes (arquivos com extensão *jar*) estão adicionados no diretório *WEB-INF/lib/*. Os arquivos *JSP* estão colocados no diretório raiz. A FIG.35 apresenta a estrutura de diretórios da aplicação *Web LIDAR*.

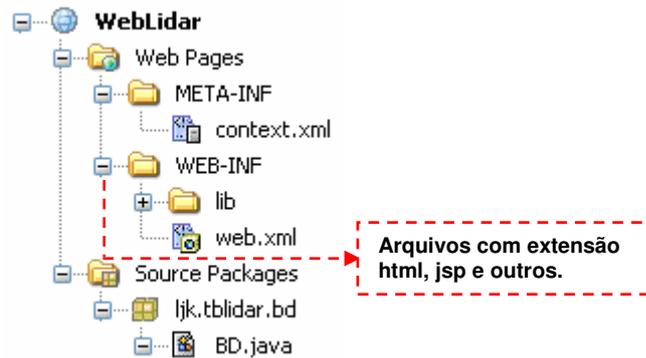


FIGURA 35: Estrutura de diretórios do *Web LIDAR*.

Então, na forma pelo qual esta aplicação foi construída, utilizou-se o seguinte:

- O uso de controlador para instanciar *bean* e encaminhar solicitações para *JSP* apropriada.
- As *JSPs* não contêm nenhuma lógica do negócio, é composta de *HTML* com funcionalidades *Java* incorporadas nelas.

5.3.2. Pesquisa níveis de dados

O acesso a níveis de dados através do *Web LIDAR* funciona da seguinte maneira. Este acesso é possível conforme um ciclo de vida definido em três etapas:

1. Uma etapa de inicialização;
2. Uma etapa de atendimento de requisições;
3. E uma etapa de finalização.

Em geral, o ciclo de vida da *JSP* pode ser observado como descrito a seguir.

Quando a *JSP* é solicitada pela primeira vez, ela é traduzida para um *servlet Java*. O *servlet* é utilizado para enviar a resposta de volta ao usuário. Portanto, o conteúdo real da *JSP* no servidor não é o que o usuário vê em seu navegador *Web*. Isto é, o usuário vê o resultado do servidor processando a *JSP*.

O funcionamento por trás da página *JSP* é o seguinte: existe um *Servlet* especial, chamado Compilador de Página - *Page Compiler*, que intercepta requisições direcionadas a recursos com extensão "*jsp*".

No instante em que é recebida uma requisição para uma página *JSP*, o compilador de página transforma essa página em um *Servlet* e o compila, sendo que o resultado dessa compilação é carregado em memória para evitar que esse processo tenha que ser repetido para todas as requisições recebidas.

O compilador de página também verifica a data de alteração do arquivo que contém a página *JSP*, caso essa data se modifique, o processo de compilação é executado novamente para garantir que modificações feitas na página sejam visíveis para os usuários da aplicação.

Devido a todo esse processo de compilação e recompilação, pode-se observar que o primeiro acesso após a criação ou modificação de uma página *JSP* é sempre mais lento que os acessos seguintes (até que haja uma modificação no conteúdo da página). Isso é por causa do processo que precisa ocorrer para criar os arquivos com extensão *java* e *class* e então instanciar o *servlet* resultante.

Nesta aplicação, o usuário visita a *home page* e tem a opção de pesquisar os níveis de dados LIDAR. Ele pode visualizar o perfil gráfico dos dados coletados da atmosfera através do sistema LIDAR, referente ao nível, ano, mês e dia pesquisados.

Os níveis de dados na *Web* são apresentados conforme exemplificado nas figuras a seguir. A FIG. 36 apresenta o *Web* LIDAR, demonstrando o acesso aos níveis.

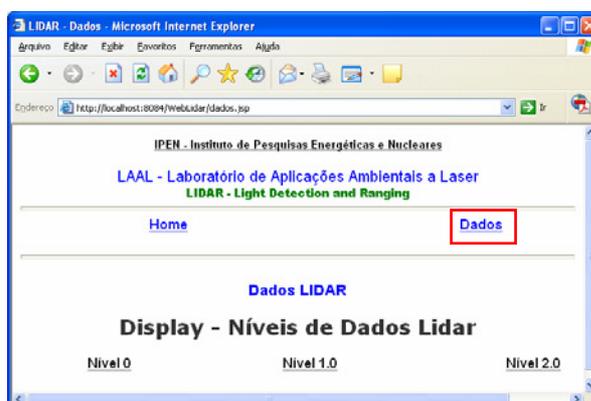


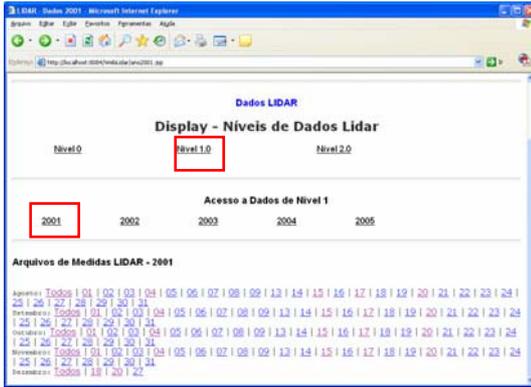
FIGURA 36: *Web* LIDAR – Níveis de dados.

Após a pesquisa referente aos níveis de dados, é acessado o ano referente ao nível escolhido. Na FIG. 37 apresenta o acesso ao nível 1.

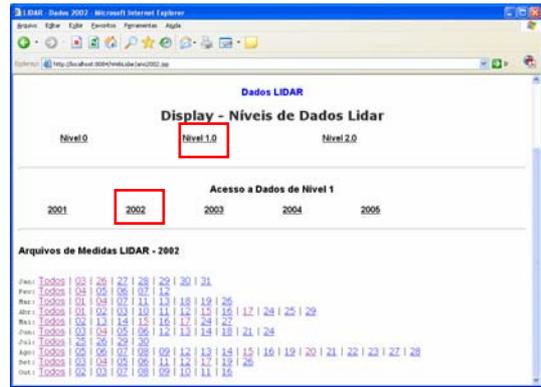


FIGURA 37: Web LIDAR – Nível 1.

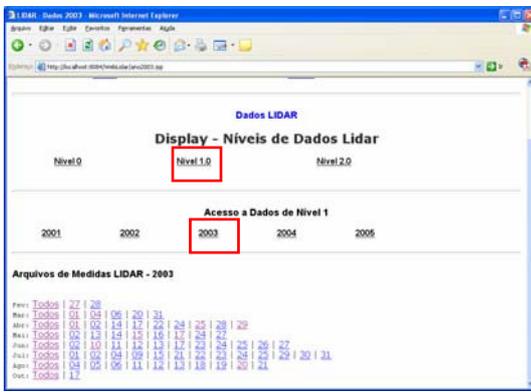
Efetuada a escolha do nível e ano desejado, são visualizados os meses e seus respectivos dias, em que houveram coletas de dados LIDAR. A seguir, há uma exibição dos *links* de acesso para os gráficos referentes aos anos 2001, 2002, 2003, 2004 e 2005.



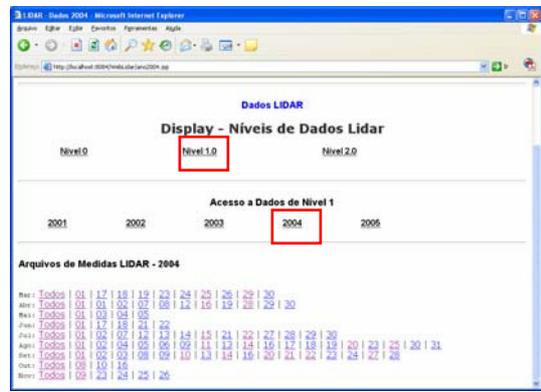
Dados - Nível 1 - 2001



Dados - Nível 1 - 2002



Dados - Nível 1 - 2003



Dados - Nível 1 - 2004



Dados - Nível 1 - 2005

FIGURA 38: Web LIDAR – Nível 1 referente aos anos de 2001 a 2005.

A figura abaixo mostra o resultado obtido da pesquisa, referente ao nível e data especificada e o acesso ao gráfico de nível 0, pertencente ao ano de 2004, ao mês de agosto e ao dia 20.

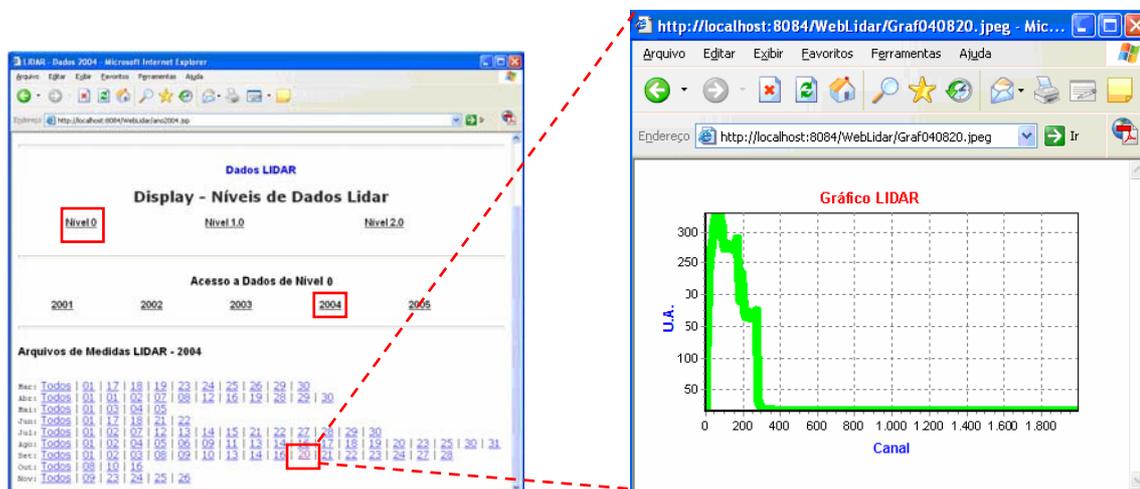


FIGURA 39: Web LIDAR – Gráfico de Nível 0 referente 20/08/2004.

6. CONCLUSÕES

O propósito deste trabalho foi a criação de um banco de dados das medidas LIDAR. Para a elaboração do banco de dados procurou-se, inicialmente, um entendimento do sistema LIDAR, bem como a interpretação de equações, os instrumentos, o sistema de aquisição de dados em tempo real e de sua operação.

No processo de análise foi utilizado a Linguagem Unificada de Modelagem (UML), para facilitar a visualização do sistema, demarcando a compreensão dos modelos estático e dinâmico envolvidos no sistema LIDAR. Noções de modelo de banco de dados relacional e orientado a objeto foram abordados, definindo a implementação como um banco de dados objeto-relacional.

Adotou-se uma metodologia de identificação e organização aos tipos de dados. Para obtenção de um resultado que permitisse a integração de informações das medidas LIDAR e proporcionasse gerenciamento, segurança e facilidade na busca de resultados, propiciando embasamento para futuras pesquisas e análise de dados foi utilizado o *PostgreSQL*.

O desenvolvimento de uma metodologia de análise otimizada em função do banco de dados, tendo como produto final um programa de análise que forneça as propriedades da atmosfera e sua estrutura, foi abordado primeiramente em relação à compreensão de análise de dados e à determinação de propriedades importantes de visualização e armazenamento destes dados.

Foram adotados, então, um método de identificação da medida e outro de classificação por níveis de dados, como nível 0, 1 e 2, determinando-se também a sua categorização como A, B e C.

Um Sistema de Análise de Dados LIDAR foi desenvolvido, utilizando-se a linguagem de programação *Delphi* para análise e pré-análise dos dados do LIDAR, resultado disso foi a maior agilidade na visualização dos resultados obtidos, o que incluiu uma boa interface gráfica.

A criação de um sistema de transferência automatizado das informações para um *site* da *internet*, para mostra dos dados adquiridos em tempo real pelo

equipamento LIDAR, foi implementado sobre a estrutura de banco de dados definida.

A metodologia adotada resulta em um controle de acesso das informações, tornando-o um elemento essencial para a segurança destes dados. Esta implementação para *Web* utiliza o contêiner *Web Jakarta Tomcat* e a tecnologia *JavaServer Pages (JSP)*, tornando válido o conteúdo dinâmico em um *site Web*, obtendo como resultado o gráfico referente ao nível e data especificada.

Portanto, fica evidente que a criação deste banco de dados se faz necessária para a organização, segurança e agilidade nos resultados das importantes informações coletadas pelo equipamento LIDAR já que há um grande volume de dados coletados.

A implementação deste banco de dados propicia enorme crescimento da pesquisa, contribuindo para estudo como fonte de dados de informações de propriedades atmosféricas, tornando-a um instrumento essencial da comunicação e visualização gráfica destes dados.

6.1. Trabalhos Futuros

É importante salientar as sugestões referentes às possibilidades de trabalhos futuros. Considerando nesta pesquisa perspectivas de continuidade, e destacando as de maior contribuição para o aprimoramento e possibilidade de novas análises, a seguir descreve-se algumas sugestões de maior relevância.

O desenvolvimento de um serviço para ativar a inserção automática de dados ao banco.

A implementação de um *Web Service*, para manipulação de dados através da *Web*.

A criação de gráficos em tempo real referentes a resultados de análises de dados LIDAR na *Web* (Java).

E o desenvolvimento de novas metodologias de análise das informações coletadas da atmosfera pelo equipamento LIDAR.

ANEXOS

ANEXO A - Equação LIDAR e sua interpretação

A atmosfera determina o sinal recebido pelo LIDAR, via retroespalhamento da luz através de dois fatores: o coeficiente de retroespalhamento β e o coeficiente de atenuação (ou extinção) α . Estes fatores são considerados na apresentação da equação de LIDAR de retroespalhamento simples^{41,42} :

$$P_r(R) = P_o \left(\frac{c\tau}{2} \right) \beta(R) A_r R^{-2} \exp \left[-2 \int_0^R \alpha(r) dr \right]$$

Onde P_r é a potência recebida num instante t , P_o é a potência transmitida no instante t_o , c é a velocidade da luz, τ é a duração do pulso, R é a distância, A_r é a área efetiva do receptor. Assim, as medidas com LIDAR fornecem informações sobre as características de espalhamento e atenuação da atmosfera, observando-se que os efeitos relacionados possam ser de alguma maneira separados.

Por outro lado, em algumas aplicações há interesse em se saber também o número de partículas ou concentração em massa a partir das propriedades ópticas da atmosfera. Nesse sentido é necessário ter disponível um modelo atmosférico, e a partir daí extrair uma relação entre α e β antes da solução da equação ser encontrada.

A obtenção dos parâmetros físicos da atmosfera a partir das medidas com um sistema LIDAR requerem, primeiro, a separação das características ópticas das interações com gases e material particulado, e posteriormente, o conhecimento apropriado das razões entre esses parâmetros, por exemplo: razão de retroespalhamento, número de partículas, extinção-massa, etc.

Frequentemente, por razões de simplificação, uma interação é descartada conforme a região da atmosfera a ser estudada, isto é, na atmosfera mais elevada (estratosfera), o retroespalhamento por particulado é mínimo comparado com o retroespalhamento gasoso, enquanto que nas atmosferas sobre centros urbanos o contrário é verdadeiro⁴³.

Para a obtenção dos perfis verticais dos coeficientes de extinção e retroespalhamento na baixa troposfera utilizam-se algoritmos de inversão^{44,45} dos sinais de LIDAR. No entanto, os parâmetros microfísicos dos aerossóis, tais como: a concentração, a distribuição de tamanho, o índice de refração e o tamanho das partículas devem ser obtidos indiretamente em conjunto com outras técnicas, e em se tratando de um sistema fixo (monoestático) de frequência única à obtenção desses parâmetros é ambígua.

Esse problema pode ser solucionado experimentalmente, ao se adicionar mais frequências ao sistema e um grau de liberdade nos ângulos de varredura do sistema de entrega/coleta do sinal retroespalhado, além disso, pode-se através de técnicas mais refinadas de análise em métodos de inversão.

ANEXO B - Especificação e Instrumentação do LIDAR

O sistema LIDAR é composto por 3 módulos principais: A Fonte Luminosa (LASER), o Sistema de Coleta e Envio de Feixe (Telescópio) e o Sistema de Detecção (Detectores + Módulos de Aquisição), como descrito a seguir e mostrado na FIG.40.

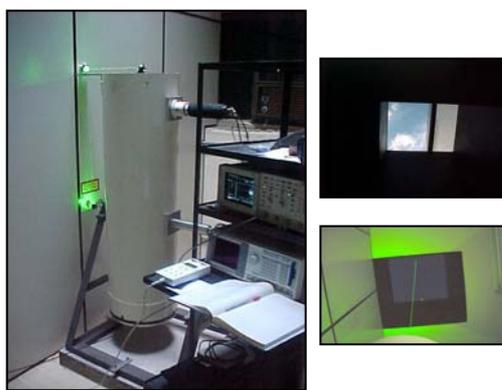


FIGURA 40: Sistema LIDAR.

A Fonte Luminosa consiste de um laser de estado sólido de Nd:YAG refrigerado a ar. As principais características desse laser são: 20Hz de taxa de repetição, 160 mJ @532 nm (máxima) de energia por pulso, 4 ns @532 nm de duração temporal do pulso, 0,5 mrad de divergência e 6 mm de diâmetro do feixe de saída.

Um Telescópio Newtoniano com emissão coaxial, intitulado MSP-LIDAR 1, com as seguintes características: 300 m de distância mínima detectável, 30 cm de diâmetro do espelho primário e material Duran 50 (baixo coeficiente de dilatação térmica), 130 cm de distância focal do espelho e deposição de filme de SiO e alumínio para otimizar a recepção na luz visível e absorver no Infravermelho.

O sistema de detecção consiste de uma fotomultiplicadora RCA, cujo catodo (S-20) possui resposta espectral no intervalo de 185 a 900 nm. Dispomos também de um fotodiodo de avalanche. O fotodiodo de avalanche (FDA), é um tipo de detector fotovoltaico, o seu uso é muito indicado em aplicações de intensidade muito baixas e resposta rápida, em regime de “*photoncounting*”. O

detector possui uma resposta espectral entre 350 e 1050 nm e uma eficiência quântica bem elevada: 80%, otimizada na região de 550nm.

Após a luz atingir o detector, há necessidade de um sistema eletrônico ou um osciloscópio digital para armazenar os dados.

Em geral, temos dois modos de armazenar e processar esses dados, conhecidos como: “analógico” e contagem de *fótons* (*photoncounting*). No modo analógico, utilizado para distâncias próximas, até 6000 m, o detector produz uma corrente que é amostrada por um *Transient recorder*, com taxas de amostragem de MHz a GHz, a luz retroespalhada de muitos tiros de laser, são armazenadas em uma memória de resposta rápida ou processadas em um módulo específico para tal, ou mesmo em um computador.

No sistema *photoncounting*, os sinais são amplificados antes de serem processados e discriminados, de maneira que somente sinais dentro de uma faixa de intensidade, inferior e superior, são considerados, incrementando assim a sensibilidade do sistema, e níveis mais baixos de intensidade são detectados. Isso corresponde a sinais retroespalhados em distâncias maiores.

O osciloscópio digital é utilizado nas aquisições em modo analógico (baixas altitudes) e também na verificação do alinhamento do sistema óptico e apresenta as seguintes características: 1 GHz de largura de banda, 4 número de canais, 4 número de amostradores, 4 GS/s de taxa de amostragem, 1mV/Divisão de sensibilidade, 8 bits (256 níveis) e 11 (8192 níveis – c/ média) de resolução vertical, 50 K – com possibilidade de extensão de 8 M/canal de tamanho máximo de registro e compatibilidade com uma placa GPIB, saída em disquete para *ZIP-DRIVE* de saídas (Periféricos).

ANEXO C - Software de Controle do Sistema LIDAR

O funcionamento do *software* de controle do sistema LIDAR (*LabView*), atua após a luz atingir o detector. Há ainda a necessidade de um sistema eletrônico (*Transient recorder*) ou um osciloscópio digital para a observação do sinal e de seu espectro assim como seu armazenamento de dados.

O *LabView*⁴⁶ é uma linguagem de programação gráfica pertencente à *National Instruments*. O acrônimo vem de “*Laboratory Virtual Instrument Engineering Workbench*”.

O principal campo de aplicação do *LabView* em relação ao sistema LIDAR é na técnica de medição. No sistema de aquisição de dados e sua manipulação é utilizado uma programação desenvolvida através do modelo *data flow*, específica do *software*, tendo cada módulo do programa uma função determinada para cada situação, seja de instrumentação e armazenamento, ou de lógica, análise e publicação de dados.

A FIG.41 mostra a publicação de dados coletados utilizando o *software LabView*.

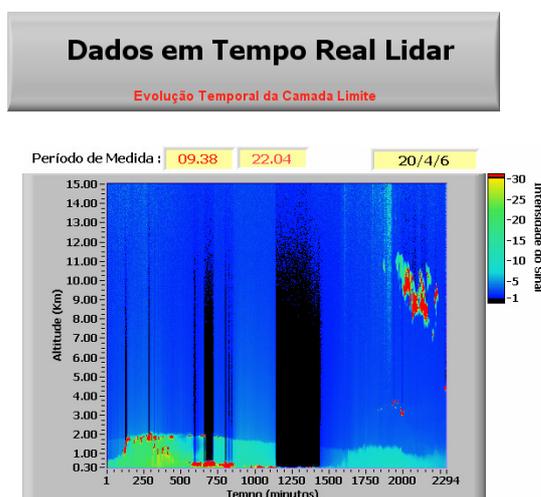


FIGURA 41: Publicação de dados utilizando o *LabView*.

O *software* usa uma linguagem de programação chamada G. Essa linguagem possui diretiva como *PASCAL* e *C*, mas que ao invés de utilizar comandos na forma de texto para gerar as linhas de código, usa uma linguagem

de programação gráfica, ou seja, o programa é feito na forma de um diagrama de blocos.

Estes módulos de programação são chamados Instrumentos Virtuais ou simplesmente *IVs*. Um *IV*, assim como um programa usual, é composto por um conjunto de instruções que fazem a manipulação e fluxo dos dados, e por uma interface com o usuário, na qual se encontram as entradas e saídas necessárias para geração do sinal através da composição de seus harmônicos, onde são compostos por dois elementos:

- Painel frontal - constitui a interface com o usuário, apresentando de forma visual, todos os controles, gráficos e indicadores formando uma tela que simula o painel físico do sistema.
- Diagrama de blocos - é a estrutura do programa propriamente dita que contém o código fonte construído de forma gráfica.

REFERÊNCIAS BIBLIOGRÁFICAS

-
- 1 CPTEC, *Centro de previsão de tempo e estudos climáticos*. Disponível em: <http://www.cptec.inpe.br>. Acesso em: 02 ago. 2005.
 - 2 AERONET, *AERosol ROBotic NETwork, NASA Goddard Space Flight Center*. Disponível em: <http://aeronet.gsfc.nasa.gov>. Acesso em: 01 ago. 2005.
 - 3 Kovalev, Vladimir A., Eichinger, William E. *Elastic Lidar: Theory Practice, and Analysis Methods*. John Wiley & Sons, Inc., 2004.
 - 4 Hulbert, E.O.: *Journal of the Optical Society of America*. 27, 344, 1937.
 - 5 Johnson, E.A.; Meyer, R.C.; Hopkins, R.E.; Mock, W.H.: *Journal of the Optical Society of America*. 29, 512, 1939.
 - 6 Troy, C.T. et al: *Photonics Spectra – Environment April Issue*, 101, 1998.
 - 7 Kobayashi, T; *Techniques for Laser Remote Sensing of the Environment*, *Rem. Sens. Rev.* 3, 1, 1987.
 - 8 R.B. Stull, *Introduction to Boundary Layer Meteorology*, Kluwer Academic, Dordrecht, The Netherlands, 1988.
 - 9 IEEE, Std 610.12 IEEE Standard Glossary of Software Engineering Terminology. *IEEE Computer Society – Software Engineering Standards Committee*, New York, NY, 1990.
 - 10 NUSEIBEH, B. A., EASTERBROOK, S.M., Requirements Engineering: A Roadmap. In: *“The Future of Software Engineering”* Anthony Finkelstein (Ed.), ACM Press, Limerick, 2000.
 - 11 Sommerville, I. *Software Engineering*, 6th Edition. Addison-Wesley, 2001.
 - 12 ISO-10746-2-Recommendation S.902/ISO/IEC 10746-2: Basic Reference Model of Open Distributed Processing – Part 2: Foundations, International Organization for Standardization, 1995.
 - 13 Booch, G., Rumbaugh, J., Jacobson, I. *The unified modeling language user guide*. Addison Wesley, 1999.
 - 14 Rumbaugh, J., Jacobson, I. et al. *The unified software development process*. Addison Wesley, 1999. (7)

-
- 15 Ward, P. e Mellor, S. *Structured development for real-time systems*. Prentice Hall, 1985.
- 16 Harel, D. *Statecharts: a visual formalism for complex systems*. Sci. Comput. Programming, 8(3), p.231-274, 1987.
- 17 Cholawsky WJ. *Real-time manufacturing database architecture*. Control Engineering 51 (8): 31-+ Aug, 2004.
- 18 Kanitkar V, Delis A. *Real-time processing in client-server databases*. IEEE Transactions On Computers 51 (3): 269-288 Mar, 2002.
- 19 Finger Fg, Gelman Me, Wild Jd, Chanin MI, Hauchecorne A, Miller Aj. *Evaluation Of Nmc Upper-Stratospheric Temperature Analyses Using Rocketsonde And Lidar Data*. Bulletin Of The American Meteorological Society 74 (5): 789-799 May, 1993.
- 20 R. Elmasri and S.B.Navathe. *Database Systems: Fundamentals of Database Systems*, Addison Wesley, 2000.
- 21 Ullman Jeffrey D., *Principles of Database and Knowledge: Base Systems*, Volume 1, Computer Science Press, 1988.
- 22 Lamar JE, Cronin CK, Scott LE. *A review of steps taken to create an international virtual laboratory at NASA Langley for aerodynamic prediction and comparison*. Progress In Aerospace Sciences 40 (3): 163-172 Apr, 2004.
- 23 PostgreSQL, *Documentation*. Disponível em: <http://www.postgresql.org>. Acesso em: 08 ago. 2005.
- 24 MYSQL, *Mysql reference manual*. Disponível em : [http:// dev.mysql.com](http://dev.mysql.com). Acesso em: 08 ago. 2005.
- 25 PostgreSQL ORG. *PostgreSQL 7.3 Administrator's Guide*, 2003.
- 26 Raghu Ramakrishnan, *Database Management Systems*, WCSB/MacGraw-Hill, 1998.
- 27 Cantù Marco. *Mastering Delphi 7*, Sybex, 2003.
- 28 Todd, Nick. *JavaServer pages: o guia do desenvolvedor*. Elsevier, 2003.
- 29 JSP, *Java Server Pages*. Disponível em: <http://java.sun.com/products/jsp>. Acesso em: 10 de nov. 2005.
- 30 Stardeveloper. *Articles Categorin*. Disponível em: <http://stardeveloper.com>. Acesso em: 12 de nov. 2005.

-
- 31 Java, *Sun Developer Network (SDN)*. Disponível em: <http://java.sun.com>. Acesso em: 10 de jul. 2005.
- 32 Servlets, *Servlets Article and Documentation*. Disponível em: <http://www.servlet.com>. Acesso em: 06 de out. 2005.
- 33 Apache Tomcat, *The Apache Jakarta Project*. Disponível em: <http://jakarta.apache.org>. Acesso em: 02 de jan. 2005.
- 34 Netbeans, *Tutorials, Guides and Articles*. Disponível em: <http://www.netbeans.org>. Acesso em: 14 de fev. 2005.
- 35 B. N. Holben, T. F. Eck, I. Slutsker, D. Tanré, J. P. Buis, A. Setzer, E. Vermote, J. AA. Reagan, Y. J. Kaufman, T. Nakajima, F. Lavenu, I. Jankowiak, A. Smirnov; *AERONET – A Federates Instrument Network and Data Archive for aerosol Characterization*, Remote Sens. Environment, 66, 1-16, 1998.
- 36 E. Landulfo et al, *Synergetic measurements of aerosols over São Paulo, Brazil using LIDAR, sunphotometer and satellite data during the dry season*; Atmos. Chemistry and Physics, 3, 1523-1539, 2003.
- 37 Landulfo, E. ; Papayannis, Alexandros ; Gonçalves, Alexandre ; Freitas, Anderson Zanardi De ; Castanho, Andrea D A ; Pozzetti, Lucila Maria Viola ; Lima, Everton de ; Souza, Renata Fernandes de . *Análise e categorização de perfis lidar para caracterização de material particulado no município de São Paulo*. In: XIII Congresso Brasileiro de Meteorologia, 2004, Fortaleza, 2004.
- 38 Klett J.D., *Extinction boundary value algorithms for LIDAR inversion*. Appl.Opt. 25, 2462, 1986.
- 39 Grabowski J., Papayannis A., Blaszczak Z., *Remote detection of aerosol in the lower troposphere by a scattering lidar*. SPIE – The International Society for Optical Engineering; 27 Sep-01 Oct, 1999.
- 40 J. Klett; *LIDAR inversion with variable backscatter/extinction ratios*, Appl. Opt., 24, 1638-1643, 1985.
- 41 Measures, R; *Laser Remote Sensing - Fundamentals and Applications*, Krieger Publishing Company, Florida, USA, 1984.
- 42 R.T.H. Collis e P.B. Russell. *Lidar Measurement of Particles and Gases by Elastic Backscattering and Differential Absorption in Laser Monitoring of the Atmosphere*. Ed. E.D. Hinkley, Springer, New York, 1976.
- 43 G.S. Kent, W. Keenlside. *J. Atmos. Sci.*, 31, 1409, 1974.

44 Heintzenber, J.H., *et al.*. *Appl. Optics.*, 20, 1308, 1981.

45 Klett, J.D.; *Stable Analytical Inversion for Processing LIDAR Return*, *Appl. Optics.*, 20, 211-220, 1981.

46 National Instruments Corporation. *LabVIEW Tutorial Manual*. 1996.