

Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

**Software de Planejamento de Ensaios para o LabOceano
Utilizando Labview**

Autor:

Fábio Quintana Pinto

Orientador:

Prof. Antônio Cláudio Gómez de Sousa, Dr.

Examinador:

Prof. Aloysio de Castro Pinto Pedroza, Dr.

Examinador:

Prof. Heraldo Luis Silveira de Almeida, D.Sc.

DEL

Setembro de 2013

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Escola Politécnica – Departamento de Eletrônica e de Computação

Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária

Rio de Janeiro – RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

DEDICATÓRIA

Dedico esse trabalho a minha família, em especial a meus pais, minha avó e minha esposa.

Aos meus pais por sempre me darem o suporte necessário. Não tenho reclamações, apenas agradecimentos pelos ensinamentos e pelo amor.

A minha avó pelas orações, pelo apoio e pelo carinho.

A minha esposa pelas ajudas, pelos incentivos e pelos puxões de orelha que me colocaram no caminho certo.

AGRADECIMENTO

Agradeço ao meu orientador, pela paciência e pela confiança em meu trabalho. Agradeço ao LabOceano pelas oportunidades, e pelo trabalho que exerço. Agradeço ao engenheiro eletrônico Rodrigo Martins Vilaça, pela idéia inicial desse projeto e pelas dicas em LabView.

Agradeço aos meus amigos e engenheiros eletrônicos, Rodrigo de Britto Florêncio e Diego Heringer Cardoso, pela cooperação nesse projeto. Também agradeço a todas as pessoas do laboratório que contribuíram de alguma forma para o aprimoramento do programa.

RESUMO

Esse trabalho demonstra o desenvolvimento de um software, o porquê de sua criação, o contexto no qual o *software* reside e seus resultados. O *software* em questão chama-se Programa de Planejamento de Ensaios e foi elaborado para o Laboratório de Tecnologia Oceânica (LabOceano) da COPPE/UFRJ.

O Programa de Planejamento de Ensaios foi implementado com a linguagem de programação LabView e tem funções administrativas ao banco de dados MySQL. A idéia do programa é organizar a geração de projetos no LabOceano, mantendo um banco de dados com todas as informações pertinentes aos projetos e automatizando a geração de relatórios para os ensaios realizados no laboratório.

Esse trabalho engloba a arquitetura do *software*, sua implementação, seu ciclo de vida e seu banco de dados. É utilizada uma abordagem baseada no fluxo de dados, onde os dados determinam a execução, o que traz vantagens para aplicações científicas e de engenharia, principalmente em aplicações de aquisição e manipulação de dados.

Palavras-Chave: LabView, Banco de Dados, Engenharia de Software, LabOceano, Ensaios Hidrodinâmicos, Instrumentação, Modelos Reduzidos.

ABSTRACT

This work presents the development of a software, the reason for its inception, the context under which the software resides and its results. The software in question is called “Planejamento de Ensaios” and was designed for the Laboratório de Tecnologia Oceânica from COPPE/UFRJ.

The Software Planejamento de Ensaios has been implemented with LabView programming language, and has administrative functions to the MySQL database. The idea of the program is to organize the generation of projects in the LabOceano, keeping a database with all the projects information and automating the generation of reports for the tests performed in the laboratory.

This work also comprises the software architecture, its implementation, its lifecycle and its database, all shown based in a data flow approach, where the data determine the execution, which brings benefits for scientific and engineering applications, mainly in applications of acquisition and data manipulation.

Key-words: LabView, Database, Software Engineering, LabOceano, Hydrodynamic Testing, Instrumentation, Small Scale Models.

SIGLAS

LabOceano – Laboratório de Tecnologia Oceânica

LabView - Laboratory Virtual Instrument Engineering Workbench

MySQL – Sistema de Gerenciamento de Banco de Dados

ODBC - Open Database Connectivity

UFRJ – Universidade Federal do Rio de Janeiro

VI – Instrumento Virtual

Sumário

1	Introdução	1
	1.1 – Tema	1
	1.2 – Delimitação	1
	1.3 – Justificativa	1
	1.4 – Objetivos	2
	1.5 – Método	2
	1.6 – Descrição	3
2	Laboratório de Tecnologia Oceânica	5
	2.1 – Considerações Iniciais	5
	2.2 – O Laboratório e os Ensaios	5
	2.3 – Os Projetos	8
3	Especificações e Modelo de Análise	10
	3.1 – Considerações Iniciais	10
	3.2 – Ciclo de Vida	11
	3.1 – Considerações Iniciais	10
	3.3 – Diagrama de Fluxo de Dados	12
	3.4 – Diagrama de Transição de Estados	12
	3.5 – Diagrama Entidade Relacionamento	14

4	Arquitetura e Implementação	17
	4.1 – Considerações Iniciais	17
	4.2 – Blocos de Funções e suas relações	17
	4.3 – Principais Funções e suas Relações	20
	4.4 – Lógica de Funcionamento e Interface	27
5	Requisitos para Operação	29
	5.1 – Considerações Iniciais	29
	5.2 – Instalador do Programa	29
	5.3 – Comunicação com o Banco de Dados	31
	5.4 – Requisitos para geração do relatório	31
6	Resultados e Conclusão	32
	6.1 – Considerações Iniciais	32
	6.2 – Resultados e Melhorias	32
	6.3 – Conclusão	33
	Referências	34

Lista de Figuras

Figura 1- Laboratório de Tecnologia Oceânica.....	5
Figura 2- Tanque oceânico	6
Figura 3 – Gerador de Ondas.....	7
Figura 4 – Vento sendo gerado por oito ventiladores.....	7
Figura 5 – Modelo de uma plataforma de petróleo em escala reduzida	8
Figura 6 – Hierarquia de um Projeto	9
Figura 7 – Estrutura de Análise do Sistema	10
Figura 8 – Ciclo de Vida	11
Figura 9 – Visão macro do sistema	12
Figura 10 – Diagrama de Transição de Estados	13
Figura 11 – Diagrama Entidade Relacionamento.....	14
Figura 12 – Bloco de execução de uma função.....	17
Figura 13 – Diagrama hierárquico de relações - parte 1.....	18
Figura 14 – Diagrama hierárquico de relações - parte 2.....	18
Figura 15 – Diagrama hierárquico de relações - parte 3.....	19
Figura 16 – Função de acesso ao Programa de Planejamento de Ensaios.....	20
Figura 17 – Função principal do campo de Sensores	21
Figura 18 – Função de inserção de uma onda no bando de dados.....	22
Figura 19 – Função de remoção de um vento do banco de dados.....	22
Figura 20 – Função Escreve Correnteza.....	23
Figura 21 – Função Conversão de escala	23
Figura 22 – VI - Conversao_escala	24
Figura 23 – VI – inserir_ordem_teste.....	25
Figura 24 – VI – inserir_ordem -> Código interno	26
Figura 25 – VI – LaTeX_Geral	26
Figura 26 – VI – LaTeX_Geral – Código interno	27
Figura 27 – Parte da interface do usuário	28
Figura 28 – Estados no código do programa	28
Figura 29 – Instalador do Programa de Planejamento de Ensaios	30
Figura 30 – Atalho para o Programa de Planejamento de Ensaios.....	30

Lista de Tabelas

Tabela 1 – Dicionário de Dados	16
Tabela 2 – <i>Froude scaling table</i>	24

Capítulo 1

Introdução

1.1 – Tema

Desenvolvimento do *software* de planejamento de ensaios que será usado no Laboratório de Tecnologia Oceânica para estruturar, modelar e arquivar todos os ensaios realizados.

1.2 – Delimitação

Esse *software* limita-se ao planejamento de ensaios do LabOceano, ou seja, os dados reais dos ensaios não serão armazenados por ele. No entanto, existem projetos futuros onde ele irá se comunicar com o *software* de ensaios e com o *software* de calibração de sensores, formando assim uma rede de dados que englobará todos os arquivos pertinentes a qualquer projeto realizado pelo LabOceano.

1.3 – Justificativa

Esse *software* é a resposta a uma demanda crescente por organização, automatização de processos repetitivos, padronização, e arquivamento seguro de dados importantes.

Antes da implementação deste programa os arquivos de planejamento eram gerados por um editor de texto e o arquivo final era armazenado no servidor em formato PDF. Os dados permaneciam em planilhas Excel, e tabelas nos arquivos de texto, o que dificulta muito uma análise futura dos resultados do ensaio.

O Programa de Planejamento de Ensaio foi feito de forma iterativa, com a participação de engenheiros navais e tecnólogos navais que participam dos ensaios no LabOceano. Esse método de desenvolvimento foi escolhido, pois não havia algo definido de como seria o software; apenas se tinha em mente o que precisava ser melhorado. Ao longo de seu desenvolvimento, novas idéias e funcionalidades foram

surgindo e sendo agregadas ao produto, de acordo com a demanda dos serviços prestados.

O programa foi todo feito em LabView e utiliza-se de uma característica interessante dessa linguagem, que é a possibilidade de se comunicar com outros programas distintos como Banco de Dados MySQL, LaTeX e o Adobe Reader, que também fazem parte do projeto.

1.4 – Objetivos

O objetivo deste projeto é criar um programa que tem como função arquivar, organizar e padronizar todos os planejamentos de ensaios realizados no LabOceano. O programa também visa à automatização do processo de geração de relatórios, modificando um processo anteriormente lento, cansativo e passível de erros em algo rápido e automático. O programa tem a capacidade de carregar qualquer ensaio da base de dados, tornando fácil a pesquisa de algum dado em um ensaio antigo.

Sua base de dados será aberta para permitir que outros programas possam utilizar seus dados, por exemplo, para geração dos canais dos sensores no programa de aquisição de dados.

1.5 – Método

O método utilizado para desenvolvimento do *software* foi o estruturado, modelado através de diagramas de fluxo de dados. Foi requerida uma intensa comunicação entre as partes envolvidas (programador e usuário).

Foi utilizada uma linguagem gráfica que fornece uma visão particionada do sistema, e a partir de conexões e iterações entre os blocos de partições, o *software* foi crescendo e ganhando novas funcionalidades.

Como a linguagem escolhida foi LabView, a interface era criada em primeira instância, até porque facilitava a comunicação com o usuário. A partir da interface eram criadas as funções e os fluxos de dados entre as partições.

Cada atualização era validada pelo usuário, que sugeria novas modificações, e verificava maus funcionamentos do sistema recém criado. A ligação com entidades externas foi feita por último, quando já havia bastante consistência e recursos funcionais.

1.6 – Descrição

Conforme dito anteriormente, o Programa de Planejamento de Ensaios foi escrito na linguagem LabView, que é uma linguagem gráfica na qual a interface do usuário é feita em paralelo com o código do desenvolvedor, o que facilita e agiliza muito o processo de desenvolvimento.

O LabView possui várias bibliotecas e ferramentas que foram utilizadas para comunicação com o banco de dados. Com elas é possível interagir com os dados que estão no banco, possibilitando acesso a todas as funções, como inserções, deleções e pesquisas.

O Banco de Dados escolhido foi o MySQL, por ser um programa gratuito, e suas características eram mais que suficientes para o desenvolvimento do banco de planejamento de ensaios.

Para geração dos relatórios foi utilizado o editor de textos MikTeX, que é uma distribuição TeX/LaTeX para Microsoft Windows, cuja licença é livre. A escolha do MikTeX foi feita pois o LaTeX possui alta qualidade tipográfica.

O capítulo 2 apresenta o Laboratório de Tecnologia Oceânica, um laboratório da Coppe/UFRJ que faz parte do programa de Engenharia Naval e Oceânica.

O capítulo 3 apresenta as especificações de requisitos técnicos do Programa de Planejamento de Ensaios. Serão demonstrados o ciclo de vida do programa, além dos elementos para análise do *software*, que são o diagrama de fluxo de dados, o diagrama de transição de estados, o diagrama entidade relacionamento e por fim, o dicionário de dados do sistema.

O capítulo 4 apresenta a arquitetura de construção das principais funções do programa, assim como a lógica de como foram implementadas e suas relações.

O capítulo 5 apresenta os requisitos para operação do Programa de Planejamento de Ensaios. Será descrito o que foi gerado e produzido para melhorar a compatibilidade do programa com as diversas versões do Windows e dessa forma facilitar sua instalação.

E por fim o capítulo 6 traz os resultados gerados pelo Programa de Planejamento de Ensaio e as conclusões do projeto.

Capítulo 2

Laboratório de Tecnologia Oceânica

2.1 – Considerações Iniciais

Esse capítulo apresenta o LabOceano, um laboratório da Coppe/UFRJ, do programa de Engenharia Naval e Oceânica, apresentado na Figura 1. As atividades realizadas no laboratório serão descritas tendo como foco principal os ensaios em seu tanque principal.



Figura 1- Laboratório de Tecnologia Oceânica
Fonte: LabOceano.

2.2 – O Laboratório e os Ensaios

O LabOceano realiza, desde sua inauguração em abril de 2003, testes e ensaios em seu tanque oceânico, cujas dimensões são 40 metros de comprimento, 30 metros de largura, 15 metros de profundidade, mais um poço central com mais 10 metros de profundidade e 5 metros de diâmetro. O tanque possui capacidade para 23 milhões de litros de água, conforme visto na Figura 2.

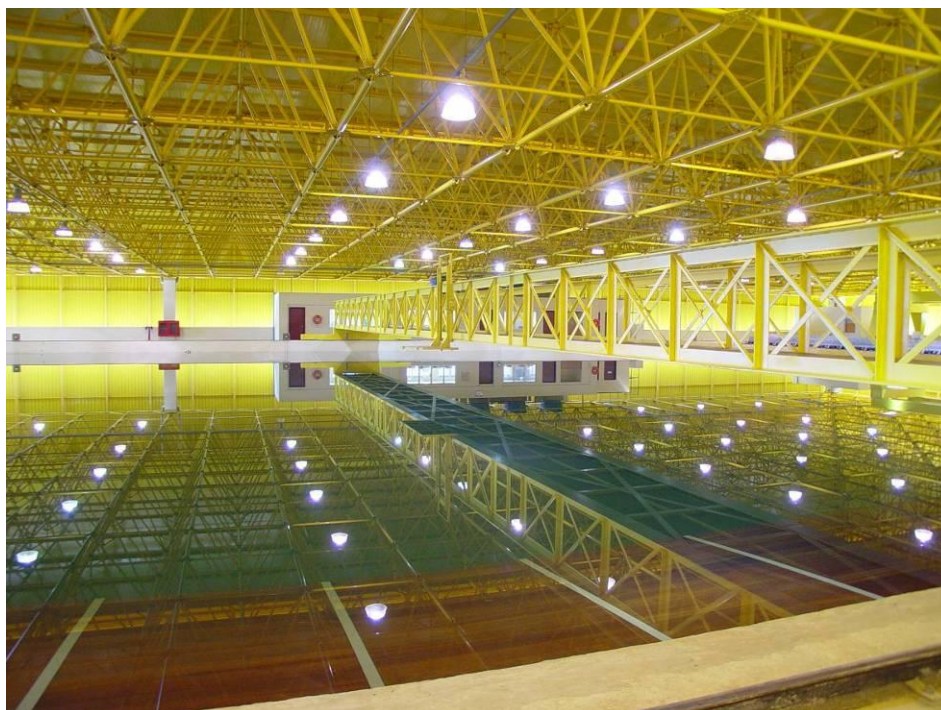


Figura 2- Tanque oceânico
Fonte: LabOceano.

O tanque oceânico também possui duas praias artificiais, com perfil parabólico (começa acima do nível da água e vai submergindo). Uma delas está posicionada ao longo do comprimento do tanque, e a outra fica oposta ao Gerador de Ondas. Elas têm a função de dissipar a energia das ondas, impedindo assim que elas reflitam e retornem ao modelo ensaiado atrapalhando os testes.

A grande maioria dos ensaios realizados no LabOceano utiliza seu Gerador de Ondas (Figura 3), composto por 75 segmentos independentes que podem gerar ondas com altura máxima de 0,5 metro, e com períodos que podem variar de 0,3 segundo até 5 segundos. Diversos tipos de ondas multidirecionais podem ser geradas, sendo as principais ondas regulares, com período e amplitude constantes, e ondas irregulares, com períodos e amplitudes variantes.



Figura 3 – Gerador de Ondas
Fonte: LabOceano.

Muitos ensaios também utilizam, além das ondas, ventos gerados por oito ventiladores, vistos na Figura 4, que podem gerar ventos com velocidades de até 12m/s. Os ventos também são simulados por motores de passo, ligados ao modelo por meio de cabo e controlados por controle PID, realimentado pelo sinal de uma célula de carga.

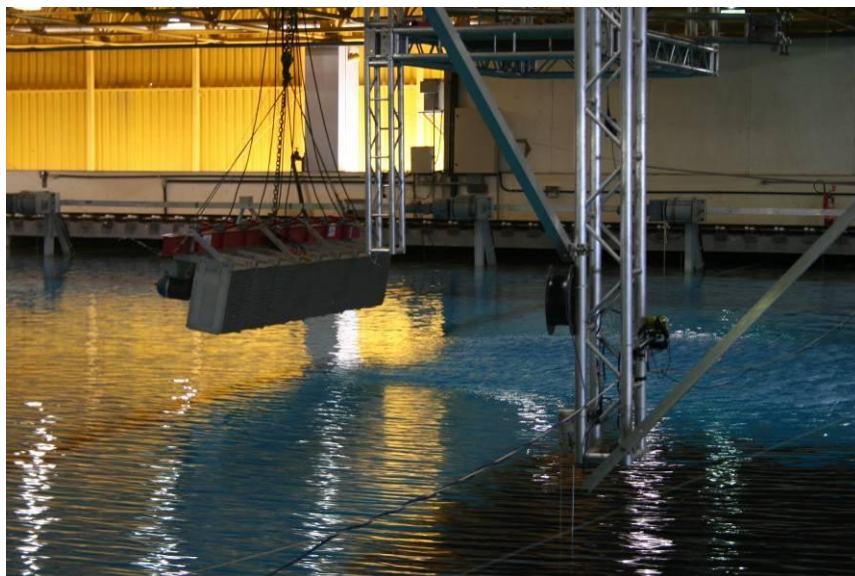


Figura 4 – Vento sendo gerado por oito ventiladores
Fonte: LabOceano.

Outra condição ambiental que também é simulada nos ensaios são as correntezas, que são geradas utilizando um motor de passo com cabo preso ao modelo e controle PID com realimentação por meio de uma célula de carga. O LabOceano também possui um projeto para correntezas reais, mas ele ainda não foi concluído.

Os modelos são instrumentados com transdutores. Os principais utilizados nos testes e ensaios são:

- Wave Probes -> Medem a altura da água
- Células de carga -> Medem a força aplicada
- Acelerômetros -> Medem a aceleração
- Girômetros -> Medem a velocidade angular

Em sua grande maioria os testes e ensaios realizados são para empresas da indústria offshore, tendo como principal cliente a Petrobras. No tanque principal os ensaios são feitos com modelos em escala reduzida, conforme apresentado na Figura 5.



Figura 5 – Modelo de uma plataforma de petróleo em escala reduzida
Fonte: LabOceano.

2.3 – Os Projetos

No LabOceano, quando um cliente contrata um ensaio, é criado um Projeto com código único. Esse projeto necessita de dados cadastrais que são:

- Nome do Projeto
- Número do Projeto
- Sigla do Projeto
- Nome do Gerente do Projeto
- Nome do Coordenador do Projeto
- Nome do Cliente

- Nome do Contato Principal com a empresa contratante
- Nome do Gerente da Naval responsável pelo projeto
- Nome do responsável pelo programa de ensaios

Após o cadastramento dos dados acima, são inseridas as condições ambientais do ensaio (eventos físicos simulados com o modelo em escala, que são ondas, ventos e correntezas).

Uma combinação de condições ambientais gera um teste com código único que, uma vez combinados, geram uma bateria de ensaios. A hierarquia de um Projeto é mostrada na Figura 6.

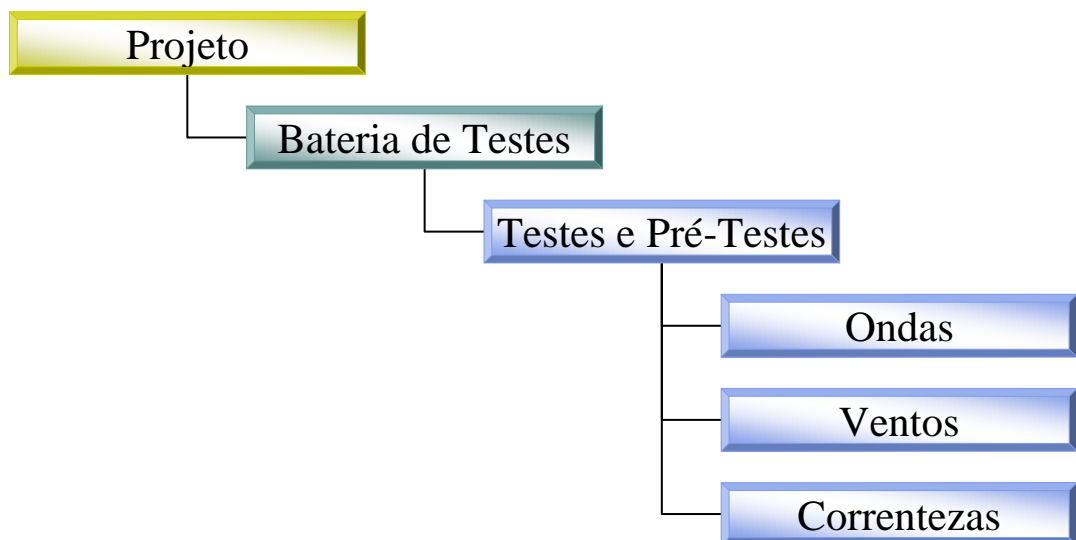


Figura 6 – Hierarquia de um Projeto

Capítulo 3

Especificações e Modelo de Análise

3.1 – Considerações Iniciais

Esse capítulo apresenta as especificações de requisitos técnicos do Programa de Planejamento de Ensaio. Serão demonstrados o ciclo de vida do programa, além dos elementos para análise do *software*, que são o diagrama de fluxo de dados, o diagrama de transição de estados, o diagrama entidade relacionamento e por fim, o dicionário de dados do sistema, conforme escrito por Roger S. Pressman [3] . A estrutura de análise utilizada está ilustrada na Figura 7.

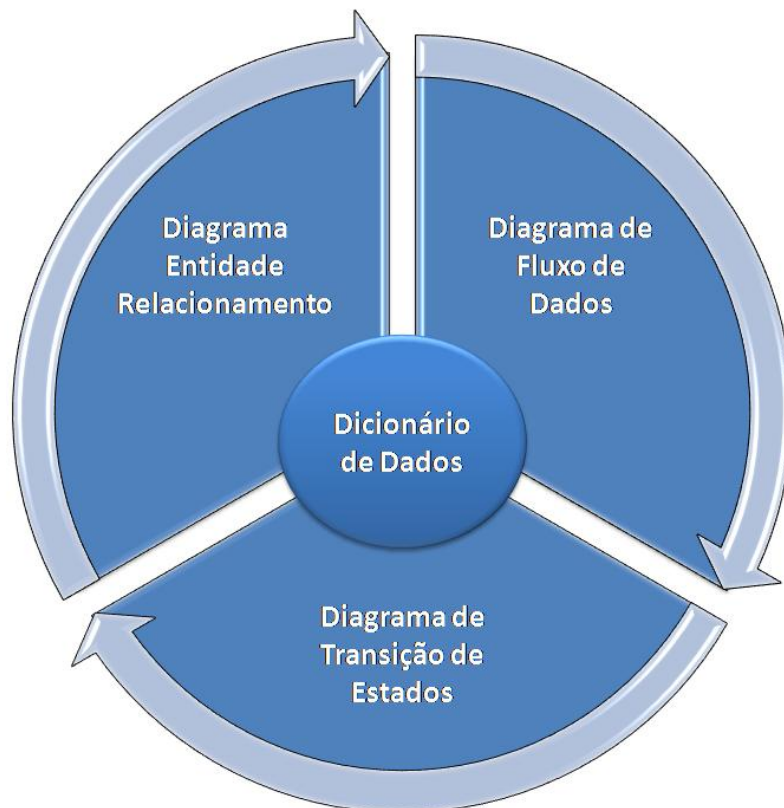


Figura 7 – Estrutura de Análise do Sistema

3.2 – Ciclo de Vida

O ciclo de vida do programa é bem dinâmico. A partir de uma definição inicial os requisitos foram sendo projetados, implementados e testados. Após a avaliação do usuário, uma nova versão era gerada e novos requisitos eram definidos. O ciclo de vida do tipo incremental, utilizado na construção deste programa, está ilustrado na Figura 8.

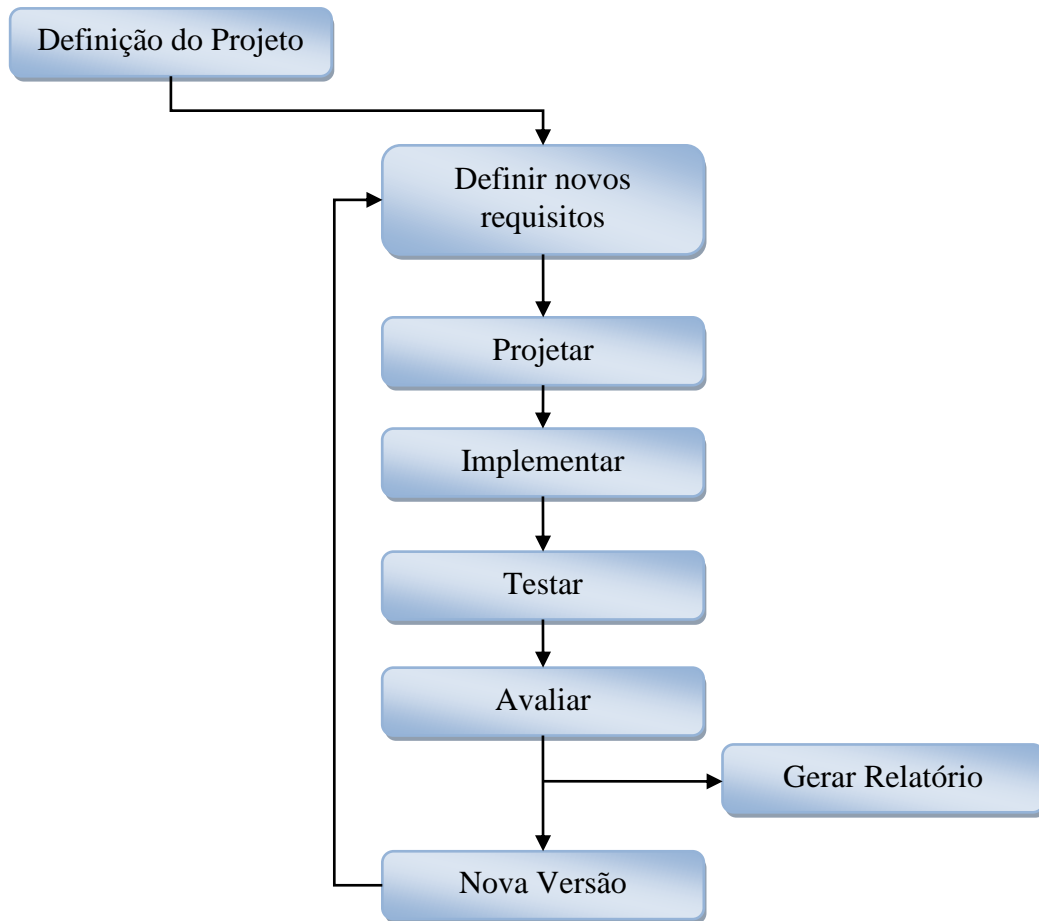


Figura 8 – Ciclo de Vida

3.3 – Diagrama de Fluxo de Dados

O diagrama de fluxo de dados do Programa de Planejamento de Ensaios é ilustrado na Figura 9. Com o diagrama é possível visualizar as interações no programa, onde os dados são armazenados, quais as estradas e saídas, como os dados são transformados e seus percursos.

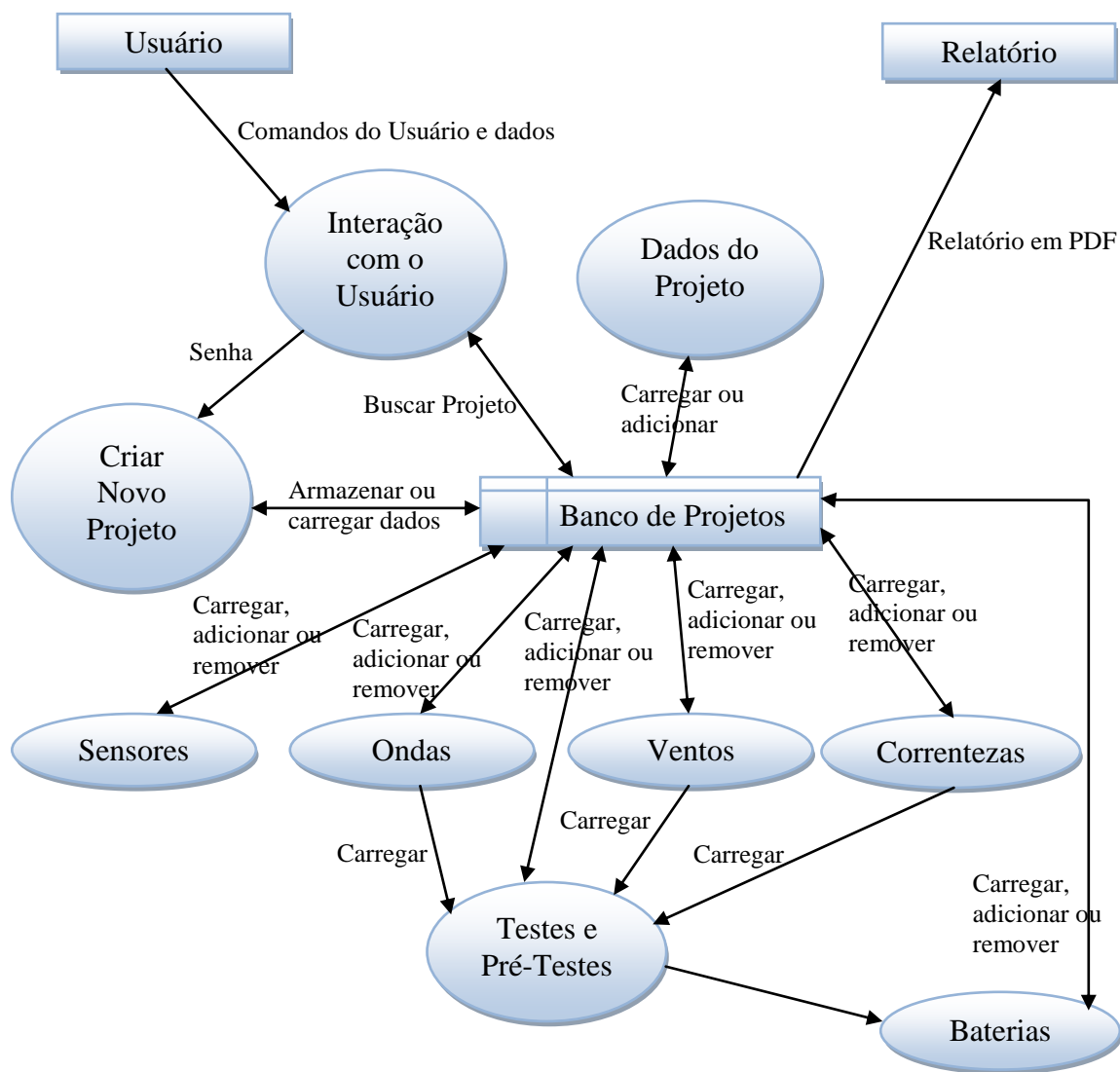


Figura 9 – Visão macro do sistema

3.4 – Diagrama de Transição de Estados

A representação utilizando um diagrama de transição de estados é mostrada na Figura 10. Essa representação mostra o comportamento do sistema e seus possíveis eventos de transição, cada *loop* está representado por uma cor distinta.

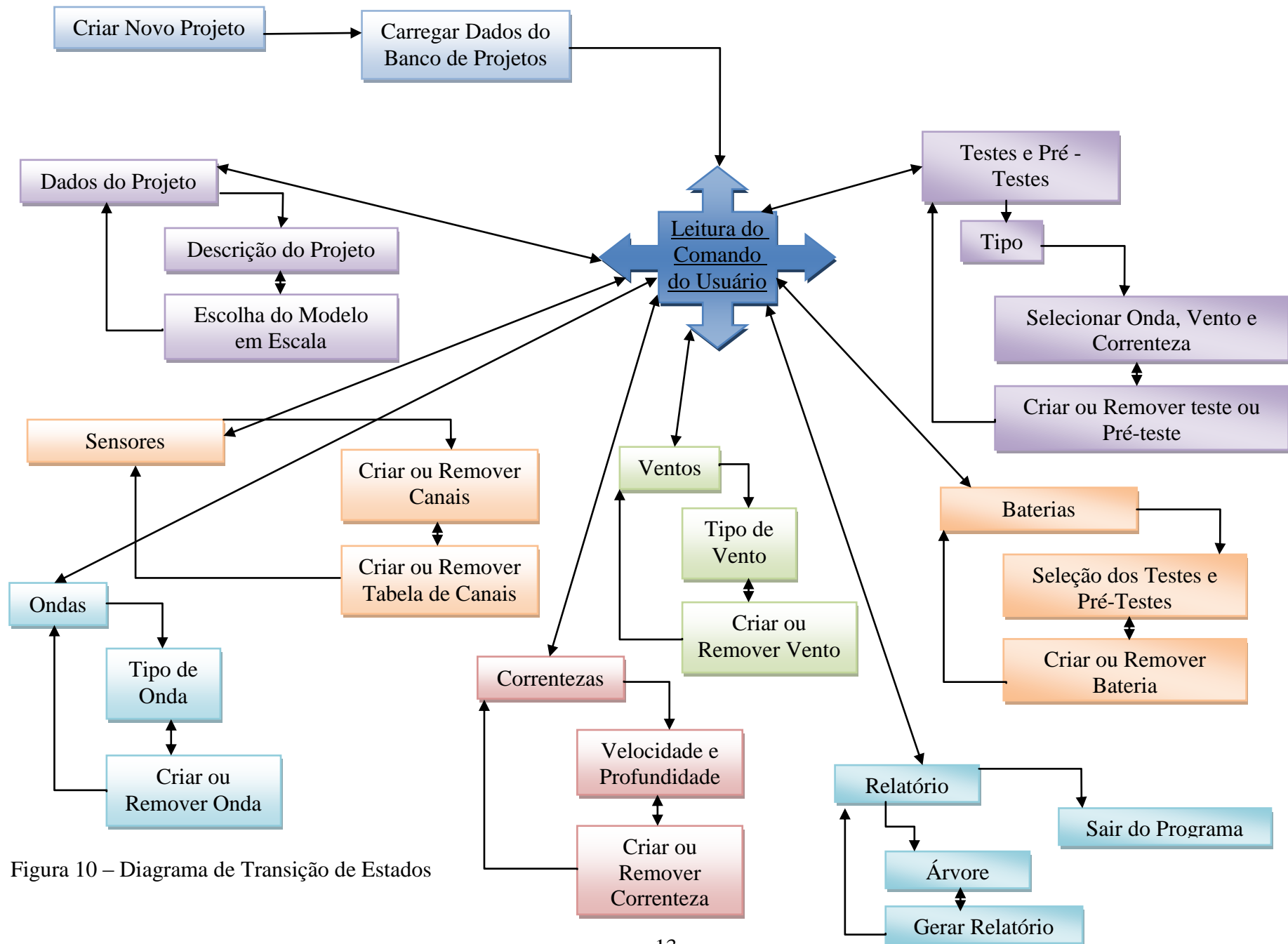


Figura 10 – Diagrama de Transição de Estados

3.5 – Diagrama Entidade Relacionamento

Segue na Figura 11 o diagrama Entidade Relacionamento do banco de dados do Programa de Planejamento de Ensaios. O banco de dados é acessível de todos os computadores do laboratório, bastando o usuário ter permissão para acesso.

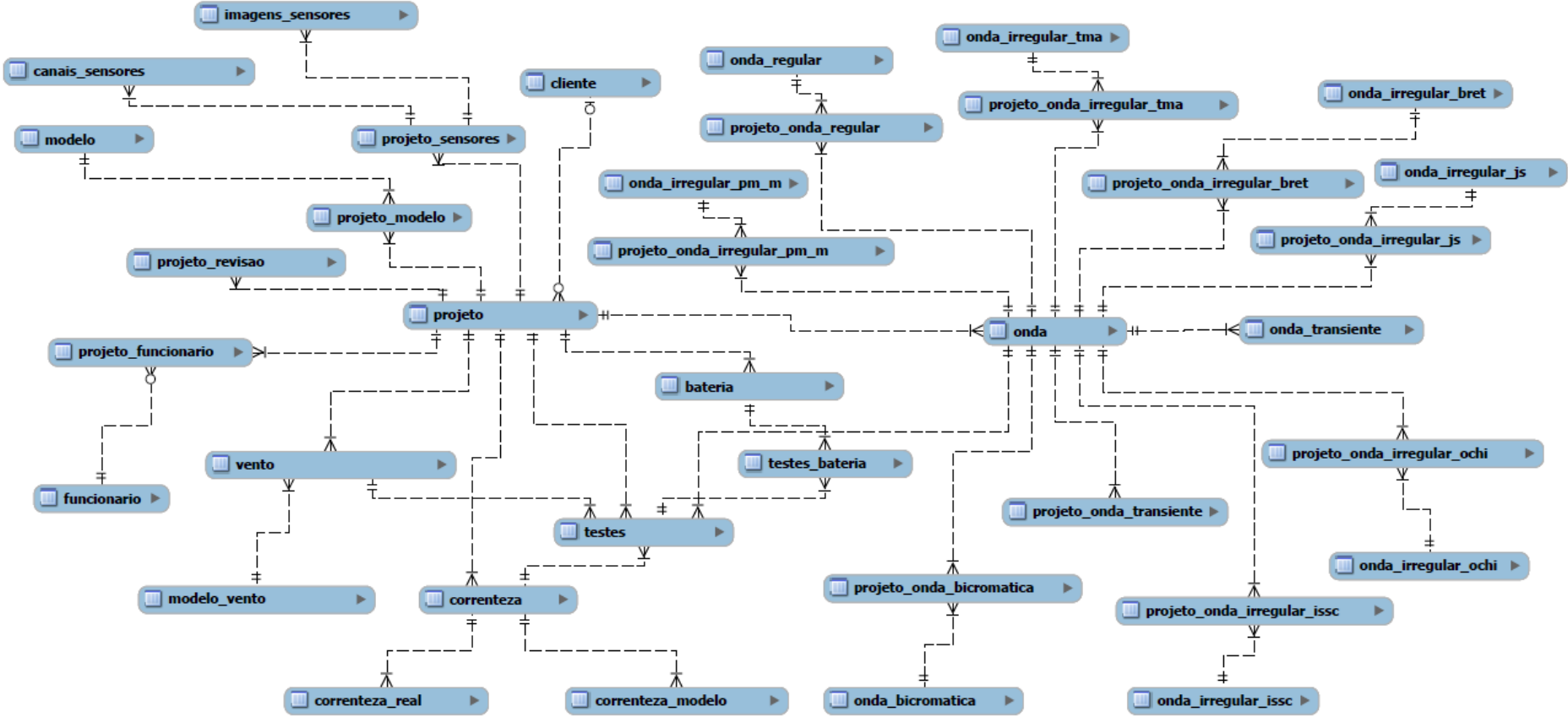


Figura 11 – Diagrama Entidade Relacionamento

As tabelas do banco de dados foram feitas utilizando o sistema de gerenciamento de banco de dados MySQL. O banco foi implementado de forma a facilitar novas adições de tabelas, pois são previstos novos tipos de ondas, ventos e correntezas. A decisão foi de criar um banco bem abrangente, com diversas entidades, tornando mais fácil uma atualização ou remoção de certa tabela, pois a mesma é independente do restante.

3.6 – Dicionário de Dados

O dicionário de dados organiza numa lista todos os elementos utilizados pelo sistema, ele é a referência de todos os termos envolvidos e com ele é possível fazer as associações do sistema. Devido ao grande número de metadados no sistema, apenas os principais foram escolhidos para o dicionário de dados.

Na Tabela 1 segue o dicionário de dados dos principais itens do sistema.

Nome do Campo	Utilização	Descrição
Descrição do Projeto	Entrada no Relatório e Banco de Projetos	Parágrafo onde entram os comentários sobre o projeto no relatório
Modelos do Projeto	Entrada de escalas no Banco de Projetos	Com as escalas selecionadas são feitos os cálculos
Nome do Canal	Entrada no relatório	Identificação da entrada do sensor
Sensor	Entrada no relatório	Nome do sensor
Posição	Entrada no relatório	Local de instalação no modelo
Escalas	VI de conversão	Escala utilizada para os cálculos
Tipo de Onda	Seleção dos metadados de uma onda	Todos os tipos de ondas cadastrados no sistema
Sufixo da Onda	Entrada no Banco de Projetos	Código único de cada onda gerada
Tipo de Vento	Seleção dos metadados de um vento	Todos os tipos de vento cadastrados no sistema
Sufixo do Vento	Entrada no Banco de Projetos	Código único de cada vento gerado
ID	Entrada de identificação no relatório	Descrição resumida da condição ambiental
Selecionar tipo de Escala	Entrada de seleção na interface	Escala real ou do modelo para os cálculos
Sufixo da	Entrada no Banco de	Código único de cada correnteza

Correnteza	Projetos	gerada
Tolerância %	Entrada na interface	Erro percentual aceitável na simulação da condição ambiental
Pré-Teste	Seleção na interface	Teste sem condições ambientais
Teste	Seleção na interface	Teste com condições ambientais
Tempo (min) (seg)	Entrada na interface	Duração dos eventos
Nome do teste ou pré-teste	Entrada no Banco de Projetos	Código único para cada teste gerado
Nome da Bateria	Entrada no Banco de Projetos	Código único de cada bateria gerada
Tempo adicional (horas)	Entrada na interface	Tempo extra para preparação do ensaio
Características dos testes	Entrada no relatório	Tabela de testes com suas condições ambientais
Dias (Jornada de 8 horas)	Indicador na interface	Cálculo do tempo total dos ensaios em dias de 8 horas
Criar árvore	Controle indicativo na interface	Árvore de planejamento com todos as baterias, testes e condições ambientais
Ordena Baterias	Controle indicativo na interface	Altera a ordem das baterias criadas
Ordena Testes	Controle indicativo na interface	Altera a ordem dos testes criados

Tabela 1 – Dicionário de Dados

Capítulo 4

Arquitetura e Implementação

4.1 – Considerações Iniciais

Nesse capítulo será apresentado a arquitetura de construção das principais funções do programa, assim como a lógica de como foram implementadas e suas relações.

4.2 – Blocos de Funções e suas relações

O Programa de Planejamento de Ensaios é constituído de 104 blocos de funções, cada qual com uma única funcionalidade. Esses blocos de funções podem ser vistos como caixas com entradas e saídas, que realizam uma operação com os dados. Na linguagem LabView, estes blocos de funções são chamados de *virtual instruments* (VIs).

Na Figura 12 está ilustrado um bloco (VI) genérico. Cada bloco pode possuir entradas e saídas e pode estar ligados a outros blocos ou a funções externas, como a comunicação com o banco de dados, ou geração de um PDF.



Figura 12 – Bloco de execução de uma função

Para visualização das 104 VIs do programa e suas relações, elas foram divididas em três partes (Figura 13, Figura 14, e Figura 15). As linhas azuis representam as ligações entre os blocos.

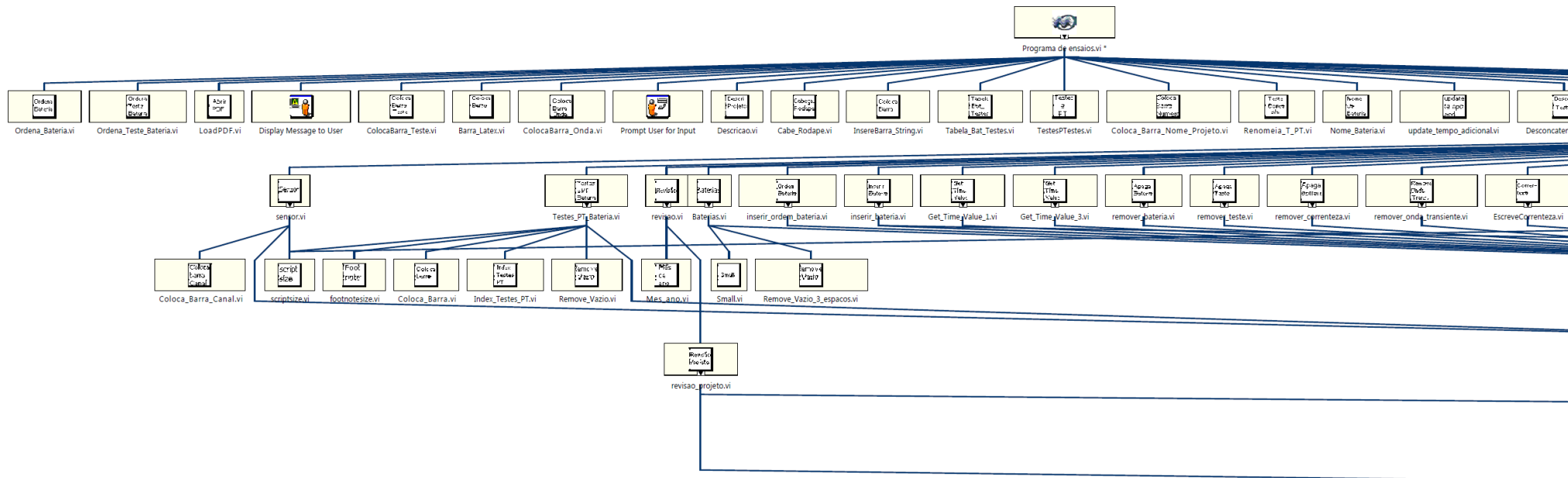


Figura 13 – Diagrama hierárquico de relações - parte 1

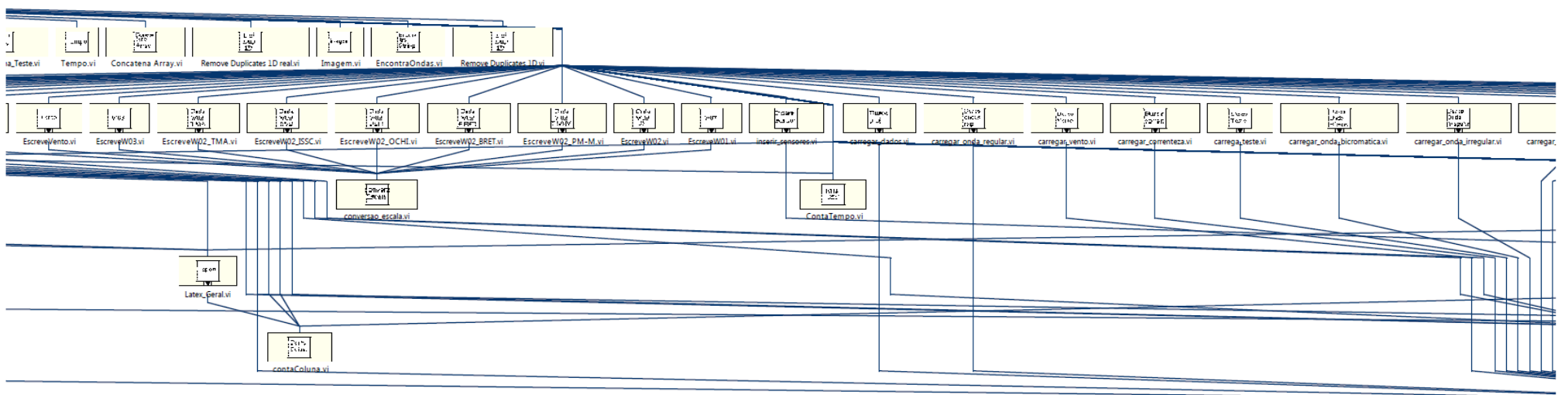


Figura 14 – Diagrama hierárquico de relações - parte 2

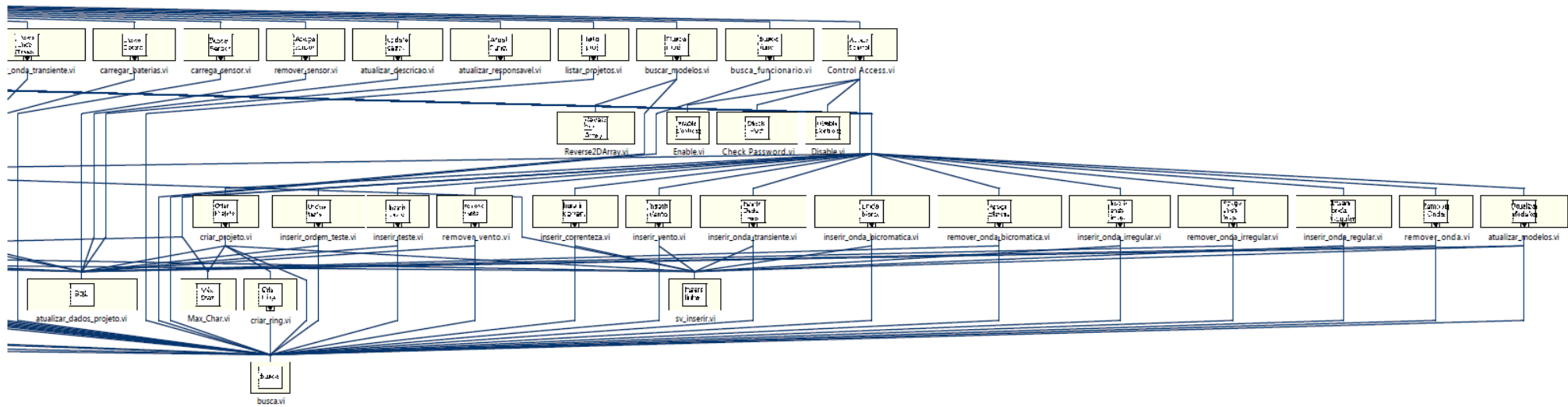


Figura 15 – Diagrama hierárquico de relações - parte 3

4.3 – Principais Funções e suas Relações

Algumas funções foram escolhidas para um melhor detalhamento de sua arquitetura. Seguindo a ordem do diagrama de transição de estados, a primeira função que o usuário encontra é a de controle de acesso. Nessa função temos a identificação do usuário, a possibilidade de criação de um novo projeto, ou simplesmente o carregamento dos dados de um projeto que foi salvo no banco de projetos. Sua arquitetura é mostrada na Figura 16.

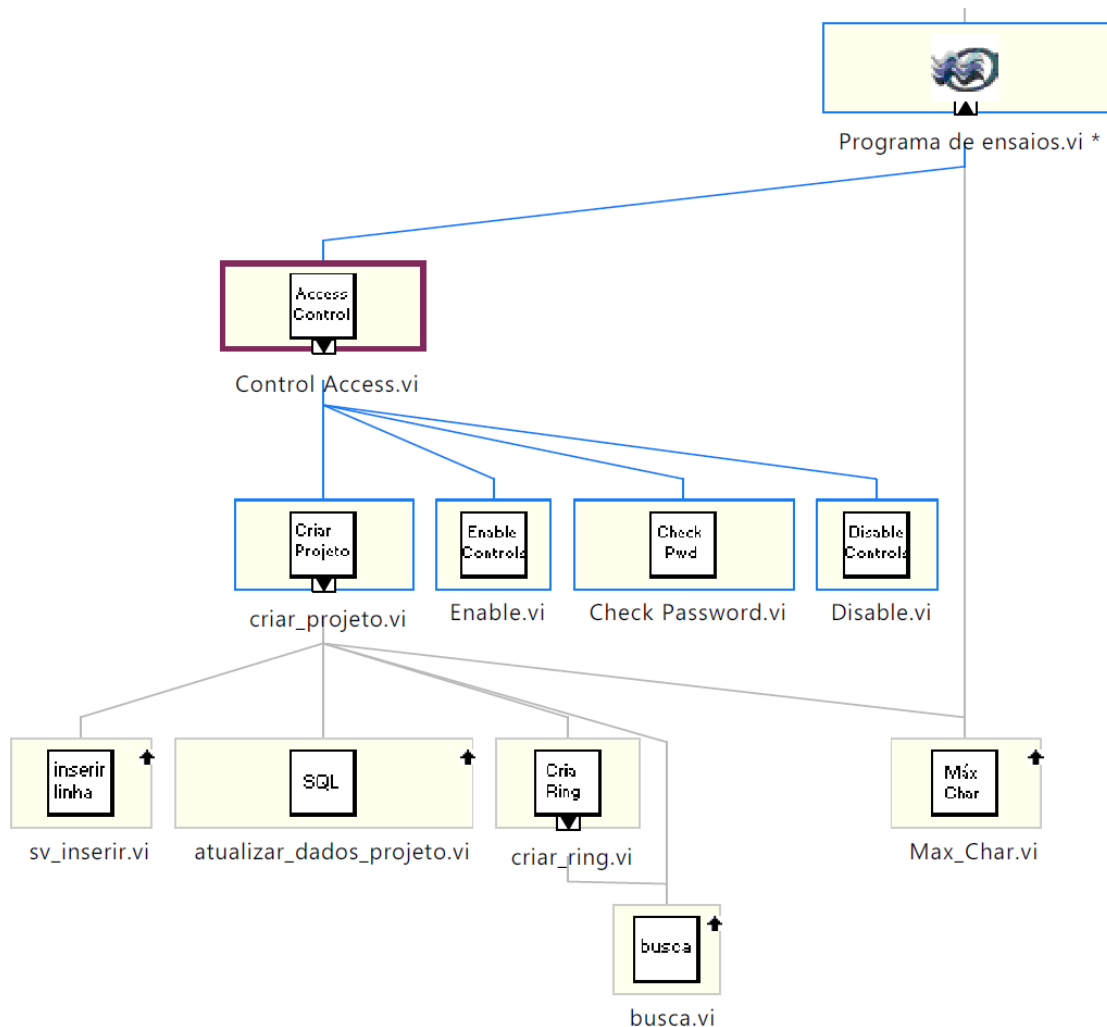


Figura 16 – Função de acesso ao Programa de Planejamento de Ensaios

A função nomeada “Control Access” possui quatro VIs internas e diretas e mais cinco outras VIs indiretas utilizadas pela função “Criar Projeto” que, por sua vez, utiliza uma função global “Máx Char”.

Para a criação de um novo projeto é requerida uma senha ao usuário. A função “Check Password” recebe e verifica a senha e, caso seja validada, o usuário poderá entrar com os dados de um novo projeto e salvá-lo no banco de dados.

Na parte de criação dos sensores a VI principal foi nomeada “Sensor”, a qual possui três VIs principais mais uma VI indireta. Sua arquitetura de relações é mostrada na Figura 17. Esse trecho de código organiza as *strings* inseridas pelo usuário e as coloca no padrão utilizado no relatório PDF (VI “LaTeX_Geral”).

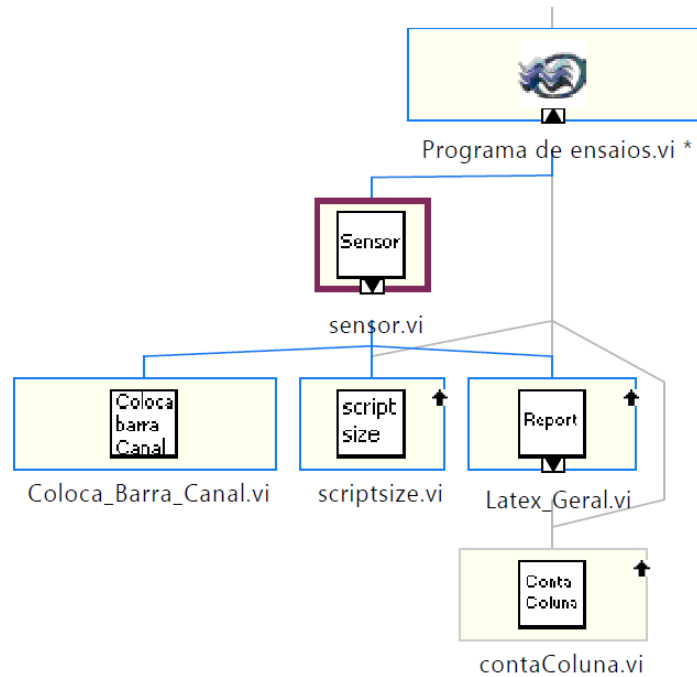


Figura 17 – Função principal do campo de Sensores

Na parte de criação de ondas, foi escolhido o trecho da arquitetura de inserção de uma onda do tipo irregular no banco de dados, como visto na Figura 18. Antes da inserção de qualquer item no banco de dados é feita uma consulta para ver se o item já existe no banco e, caso não exista, ele é inserido. Para tais tarefas foram utilizadas as VIs “Busca” e “Inserir linha”. Arquiteturas similares são utilizadas para inserção de outros itens no banco de dados como, por exemplo, a inserção de testes e baterias.



Figura 18 – Função de inserção de uma onda no bando de dados

Da parte de criação de um novo vento, será ilustrada a remoção de um vento do banco de dados (Figura 19). Para remoção de um vento do banco de dados é feita uma busca para encontrar o vento que será removido e, após encontrá-lo, o vento é apagado e o banco é atualizado no programa. Arquiteturas similares são utilizadas para remoção de outros itens do banco de dados como, por exemplo, remoção de ondas e correntezas.

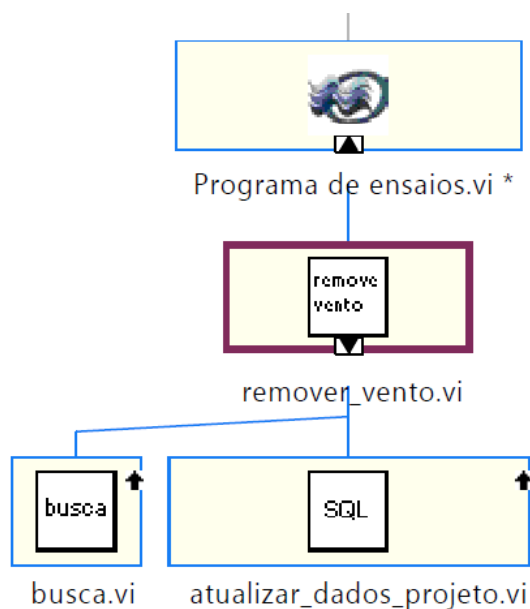


Figura 19 – Função de remoção de um vento do banco de dados

A arquitetura escolhida na criação de correntezas foi a “EscreveCorrenteza”. Essa função pega todas as entradas numéricas inseridas pelo usuário e aplica a

conversão para a escala selecionada. Essa arquitetura também é utilizada nas ondas e ventos, e pode ser vista na Figura 20.

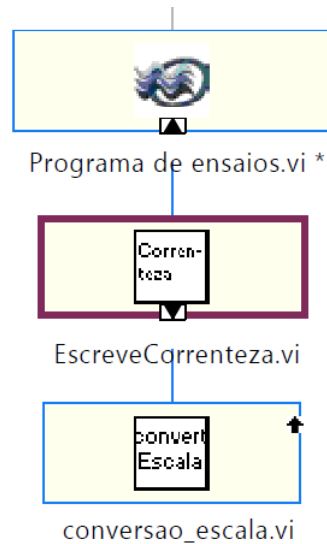


Figura 20 – Função Escreve Correnteza

Outra arquitetura importante, utilizada por todos os blocos de escrita numéricos que exigem conversões de escala, é a “Conversao_escala”, mostrada na Figura 21.

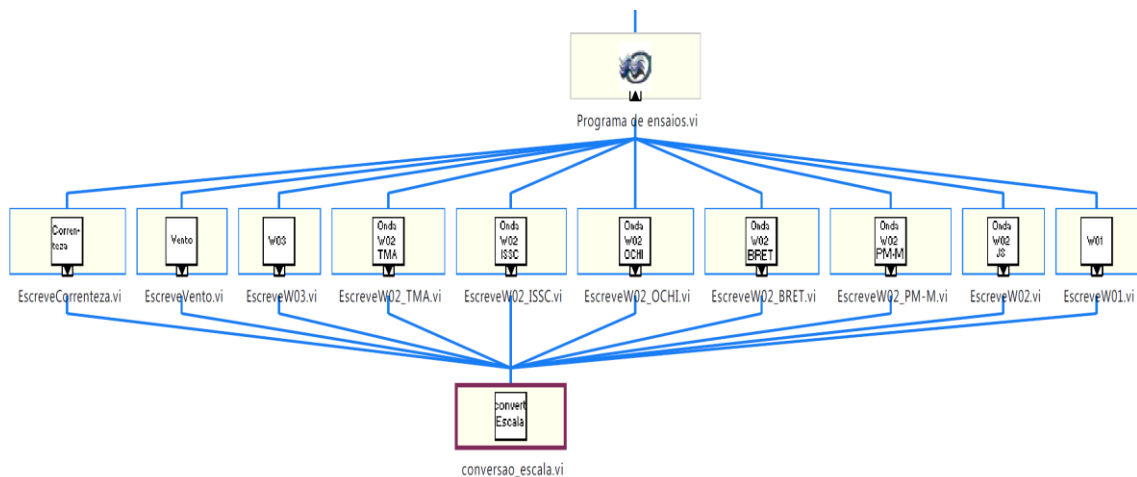


Figura 21 – Função Conversão de escala

Esse bloco obedece às leis de conversão da tabela de Froude [1], como mostrado na Tabela 2.

Physical Parameter	Unit	Multiplication factor
Length	[m]	λ
Structural mass:	[kg]	$\lambda^3 \cdot \rho_F / \rho_M$
Force:	[N]	$\lambda^3 \cdot \rho_F / \rho_M$
Moment:	[Nm]	$\lambda^4 \cdot \rho_F / \rho_M$
Acceleration:	[m/s ²]	$a_F = a_M$
Time:	[s]	$\sqrt{\lambda}$
Pressure:	[Pa=N/m ²]	$\lambda \cdot \rho_F / \rho_M$

Tabela 2 – Froude scaling table

Fonte: Experimental Methods in Marine Hydrodynamics [4].

Essas conversões são muito importantes para testes em escala reduzida, pois o tempo, as forças e acelerações, por exemplo, medidos no modelo em escala, devem ser amplificados para um modelo real. Um exemplo ilustrativo de conversão para o tempo numa escala de 1:50, supondo um teste de duração de um minuto no modelo reduzido, equivaleria a aproximadamente sete minutos na escala real.

Na Figura 28 temos a VI “conversao_escala” mostrando todas as suas entradas de dados (Modelo, Valor, Unidade, Conversão, Real e Escala), e sua saída (Valor em escala). Os dados com linhas azuis são números inteiros, e aqueles com linha laranja são números reais.

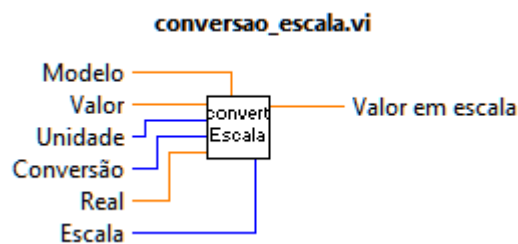


Figura 22 – VI - Conversao_escala

Um exemplo de arquitetura que se comunica diretamente com o banco de dados pode ser ilustrada pela VI “inserir_ordem_teste”, ela está apresentada na Figura 23.

Essa VI executa duas funções para se comunicar com o banco, “busca” e “atualizar_dados_projeto”.

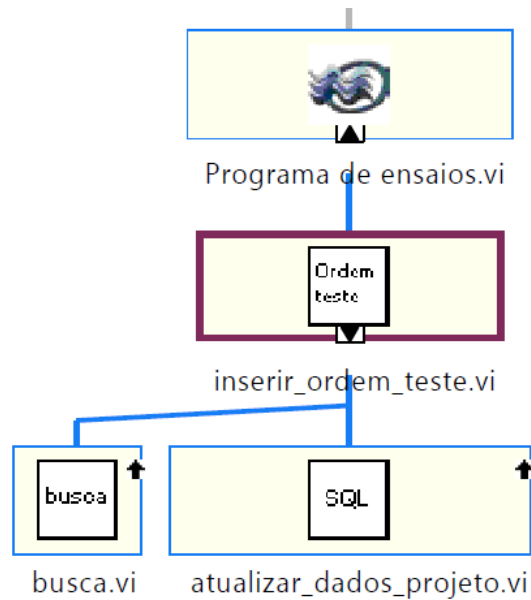


Figura 23 – VI – inserir_ordem_teste

Para melhor entendimento de como os dados são modificados no banco de dados, basta visualizar a Figura 24, que mostra parte do código da VI “inserir_ordem_teste”.

É possível ver algumas funções utilizadas na busca do teste no banco, sendo elas:

- SELECT
- FROM
- WHERE
- AND
- UPDATE
- SET

Com essas funções é feita uma busca pelo teste selecionado e seu valor de ordenação é alterado no banco.

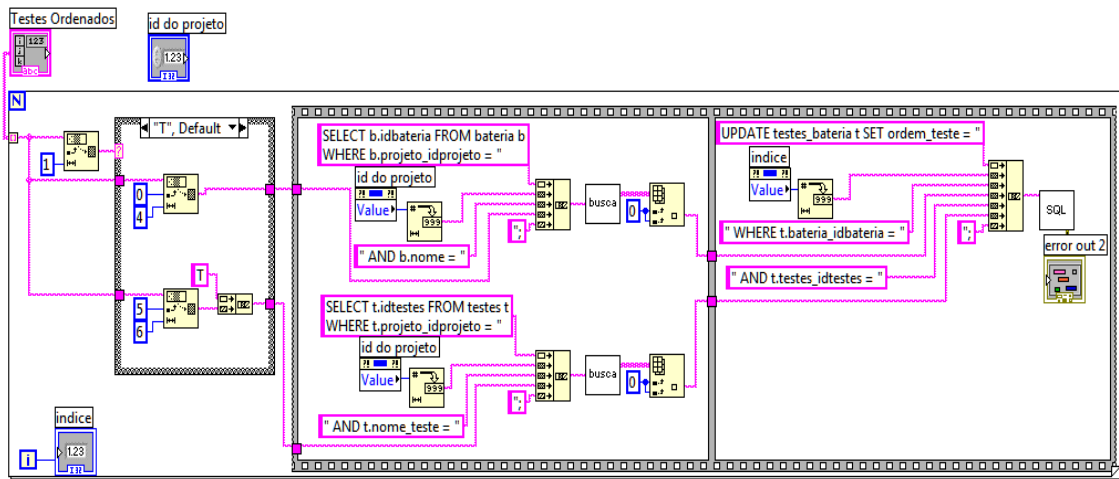


Figura 24 – VI – inserir_ordem -> Código interno

Para geração do relatório foi utilizada a arquitetura mostrada na Figura 25, onde é possível visualizar várias VIs em paralelo com a VI “LaTeX_Geral”. Essas VIs são de tratamento de *strings* para o relatório e são chamadas pelas funções que escrevem no relatório, sendo elas “Sensor”, “Testes_PT_Bateria” e “Baterias”.

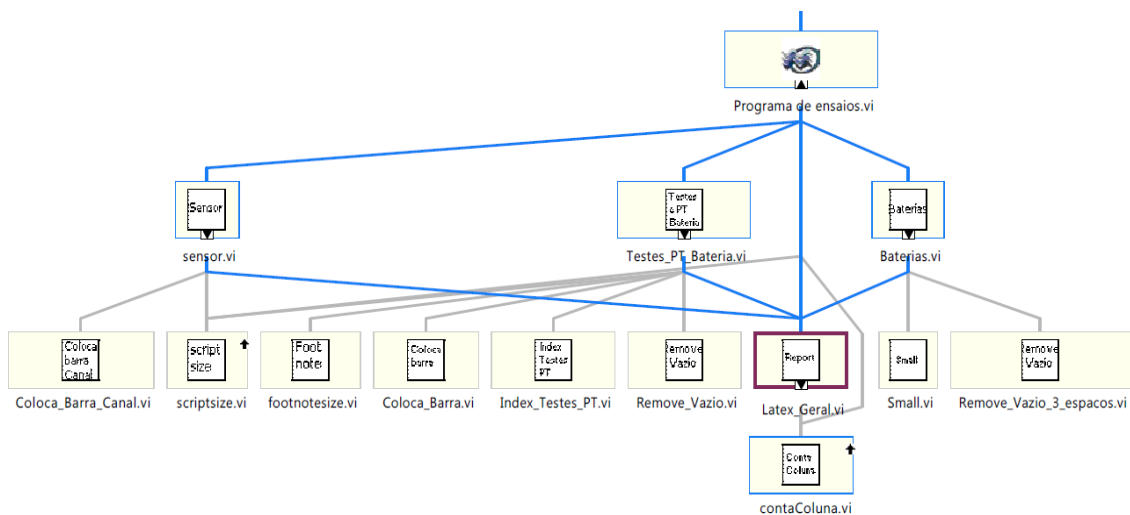


Figura 25 – VI – LaTeX_Geral

Na Figura 26 é mostrado parte do código dessa VI, que tem como função montar o texto no padrão utilizado pelo editor de textos LaTeX. No código fica visível onde a VI “contaColula” é chamada. Essa VI conta o número de colunas e devolve um inteiro que será o número de iterações do “for loop” da VI “LaTeX_geral”.

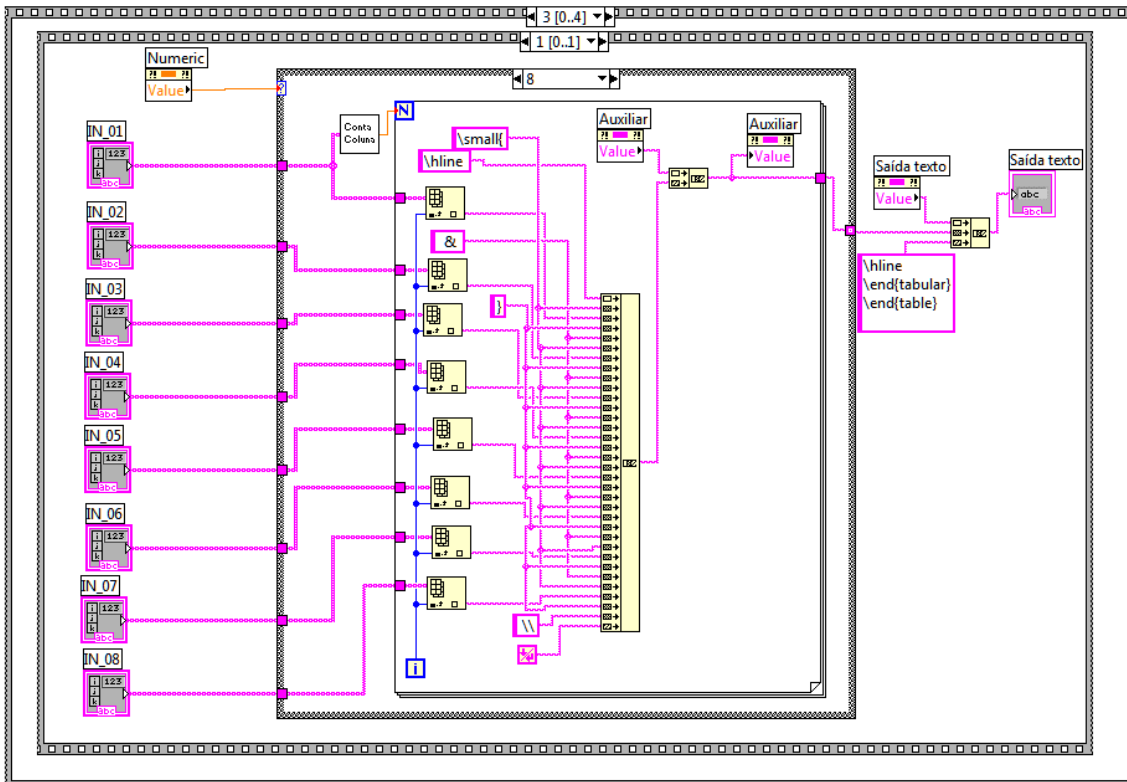


Figura 26 – VI – LaTeX_Geral – Código interno

4.4 – Lógica de Funcionamento e Interface

Apesar de o programa ser bastante complexo, ele obedece a uma lógica simples de funcionamento. As iterações do código foram particionadas se baseando na interface do usuário e o código do programa foi dividido em oito partes, sendo elas:

- Dados do Projeto
- Posicionamento dos Sensores
- Programa de Ondas
- Programa de Ventos
- Programa de Correntezas
- Criação de Testes
- Criação das Baterias
- Programa de Ensaios

Essas divisões foram feitas da mesma forma como na interface do usuário. Na Figura 27 temos a aba principal superior da interface do usuário com essas divisões. Quando o usuário seleciona uma aba, o programa fica rodando apenas dentro dela, até

que ele selecione outra aba. Dessa forma poupa-se processamento com controles desnecessários, que não serão utilizados numa determinada “aba” do programa.



Figura 27 – Parte da interface do usuário

Na Figura 28 temos parte do controle principal do programa, onde podemos ver os estados divididos como na interface.

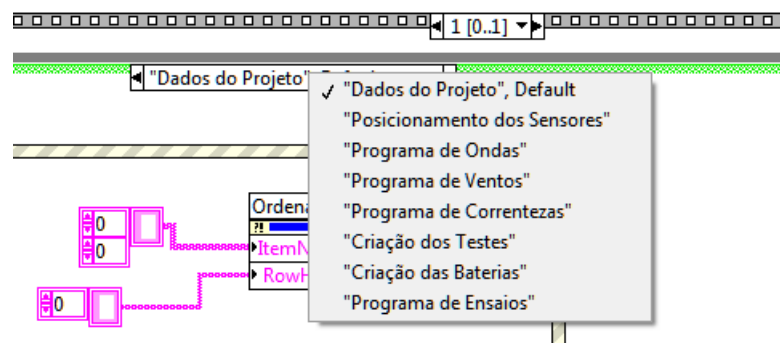


Figura 28 – Estados no código do programa

Essas divisões tiveram o intuito de facilitar o entendimento do programa, pois todas as funções de uma aba da interface estarão numa mesma parte do código. Por exemplo, todas as funções relacionadas com a criação de ondas estarão apenas na parte do “Programa de Ondas”; é como se fossem oito programas independentes dentro do Programa de Planejamento de Ensaios.

Capítulo 5

Requisitos para Operação

5.1 – Considerações Iniciais

Nesse capítulo serão apresentados os requisitos para operação do Programa de Planejamento de Ensaios. Será descrito o que foi gerado e produzido para melhorar a compatibilidade do programa com as diversas versões do Windows e dessa forma facilitar sua instalação.

5.2– Instalador do Programa

Após inúmeras versões o Programa de Planejamento de Ensaios tornou-se funcional, e foi necessário criar seu executável para instalação em outras máquinas. Não era interessante instalar todo o programa LabView e MySQL nos computadores que iriam rodar o Programa de Planejamento de Ensaios, por se tratarem de programas que exigem muita memória RAM do computador, mesmo sem estarem em uso efetivo, pois eles rodam em segundo plano. São programas apenas para máquinas de desenvolvedores, não para usuários finais.

Para solucionar o problema da instalação do LabView foi gerado um instalador para o Programa de Planejamento de Ensaios mostrado na Figura 29. Esse instalador combina todas as bibliotecas necessárias para um perfeito funcionamento do programa, e dessa forma qualquer máquina com sistema operacional Windows é capaz de rodar o Programa de Planejamento de Ensaios, sem a necessidade da instalação do programa LabView. O instalador também cria uma pasta no menu “iniciar” do Windows, e um atalho com ícone para o programa mostrado na Figura 30.

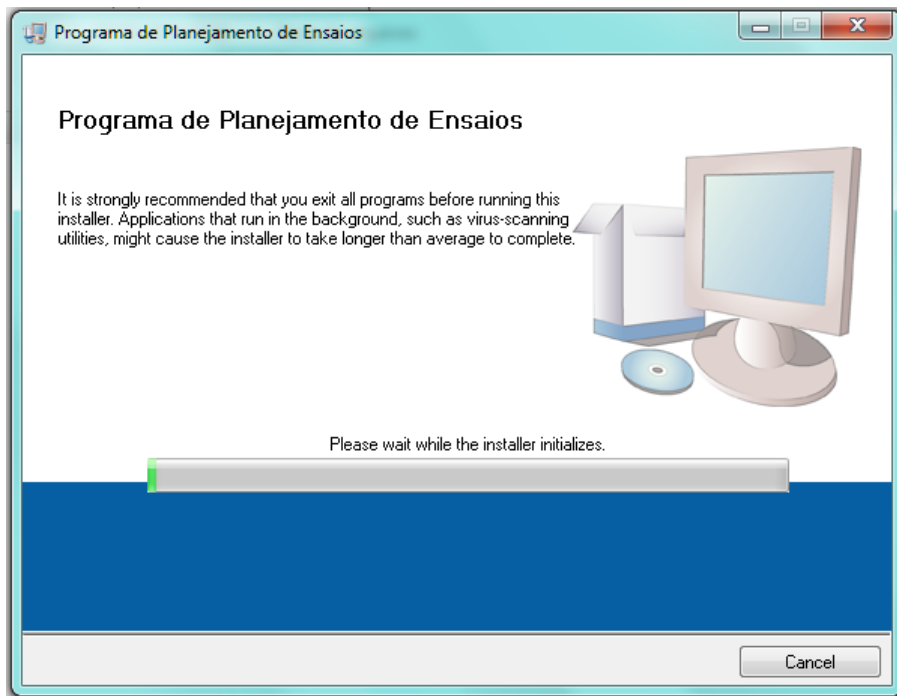


Figura 29 – Instalador do Programa de Planejamento de Ensaios

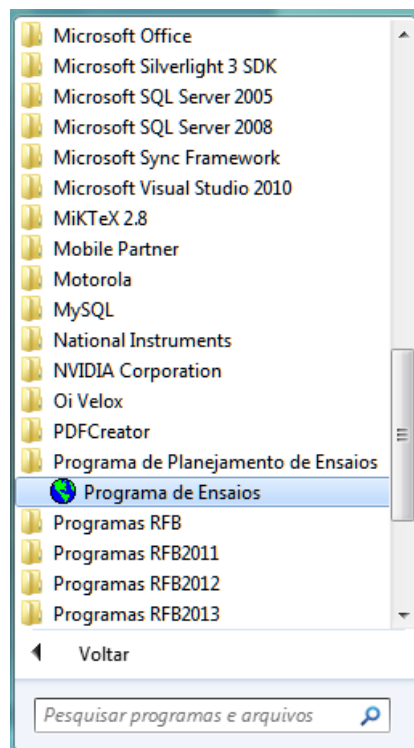


Figura 30 – Atalho para o Programa de Planejamento de Ensaios

5.3– Comunicação com o Banco de Dados

O servidor do banco de dados foi instalado numa máquina que roda o tempo todo, e foi dado um IP ao banco de dados que está visível para todas as máquinas do laboratório.

Para comunicação com o banco de dados foi utilizado um ODBC (Open Database Connectivity). Com essa aplicação é possível um computador ter acesso ao banco de dados e suas ferramentas administrativas, sem a necessidade de instalação do MySQL.

Foi adicionado ao instalador comentado na seção 5.2 um arquivo de configuração para acesso ao banco de dados, o qual é instalado numa pasta oculta do Windows. Foi escolhido um local que existe em todas as versões do sistema operacional Windows, o que permitiu automatizar o processo de instalação e tornar o arquivo invisível para o usuário.

O conteúdo do arquivo de configuração pode ser visto abaixo. Ele contém apenas seis linhas necessárias para comunicação com o banco de dados, sendo elas:

```
[ODBC]
DRIVER=MySQL ODBC 5.1 Driver
server=IP
database=Nome do Banco
user=Nome do usuário
password=Senha para acesso ao Banco
```

5.4 – Requisitos para geração do relatório

Para gerar um relatório é necessário ter instalado no computador o editor de textos MikTex, que é uma distribuição TeX/LaTeX para Microsoft Windows, de licença livre, e também o leitor de arquivos PDF o Adobe Acrobat Reader, que é um freeware.

Capítulo 6

Resultados e Conclusão

6.1 – Considerações Iniciais

Nesse capítulo serão apresentados os resultados gerados pelo Programa de Planejamento de Ensaios e as conclusões do projeto.

6.2 – Resultados e Melhorias

Três projetos do LabOceano foram feitos utilizando o Programa de Planejamento de Ensaios, mas nenhum relatório final foi gerado até o momento. O programa ainda está em fase de testes e algumas modificações precisam ser feitas antes da versão final ser lançada.

Seguem abaixo algumas atualizações futuras já planejadas:

- Melhorar a segurança para acesso ao banco de projetos;
- Melhorar a segurança para acesso ao Programa de Planejamento de Ensaios;
- Adicionar outros tipos de ondas;
- Adicionar capacidade de carregar e adicionar imagens no banco de dados e no relatório;
- Melhorar o *layout* do relatório gerado pelo programa;
- Conectar o Programa de Planejamento de Ensaios com o Programa de calibração de sensores e também com o Programa de aquisição de dados, ambos desenvolvidos e em uso no LabOceano.

A segurança do Programa de Planejamento de Ensaios ainda está fraca; não existe criptografia para as senhas nem para os arquivos de configuração. Sem dúvida é um ponto que necessita de atualização urgente.

Quando o Programa de Planejamento de Ensaio foi proposto, uma idéia era ele se comunicar com outros programas utilizados no LabOceano. Dessa forma os dados nele armazenados poderiam ser alimentados por outro programa, tais como o de calibração dos sensores, e também ele poderia alimentar outros programas, como o de aquisição de dados. Com essas melhorias haveria um ganho em velocidade no preenchimento de campos e minimizaria erros do usuário na entrada desses dados.

6.3– Conclusão

Ao projetar e implementar o Programa de Planejamento de Ensaio, foi possível aprender a criar algo em conjunto com outras pessoas. Foi um grande ganho de experiência na área de criação de um *software*.

Uma nova linguagem de programação foi aprendida, a linguagem LabView, que é muito útil no setor de engenharia eletrônica. Também foi possível aprimorar bastante os conhecimentos com banco de dados e colocar em prática o que aprendi na faculdade. Outro ganho foi conhecer um editor de textos promissor chamado LaTeX, muito utilizado na área acadêmica. Apesar de não ter o considerado intuitivo, ele possui excelente qualidade tipográfica.

O projeto acabou sendo muito útil em minha formação como engenheiro eletrônico, e também na minha carreira profissional, pois englobou vários campos de conhecimento e, graças aos conhecimentos adquiridos na faculdade, foi possível encontrar soluções para os problemas que surgiram no âmbito profissional.

Referências

- [1] YUNUS A. C.; CIMBALA M. J. Fluid Mechanics: Fundamentals and Applications. 2nd Edition, Reno: McGraw-Hill, 2010.
- [2] LOPES, T. E. C.; DÓRIA, E. S. “Análise Estruturada de Sistemas para Aplicações Convencionais”; disponível em:
<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&ved=0CDQQFjAB&url=http%3A%2F%2Ffipp.unoeste.br%2F~emerson%2Fengsof%2Fanaliseestruturada.ppt&ei=v3sTUoSfKYn68QTV1oHICQ&usg=AFQjCNFjdIN0fBgU0x_N5PD5wal5R1iSqw&sig2=yZq_rsr0SSsXbcSqd24YUg&bvm=bv.50952593,d.eWU>, (Acesso em 11/07/2013).
- [3] PRESSMAN, R. S. Software Engineering a practitioner’s approach. Fifth Edition, New York: McGraw-Hill, 2001.
- [4] STEEN, S. Experimental Methods in Marine Hydrodynamics, General Modelling and Scaling Laws; disponível em:
<http://www.ivt.ntnu.no/imt/courses/tmr7/lecture/Scaling_Laws.pdf>, (Acesso em 25/06/2013).