

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

FERRAMENTA DE GERENCIAMENTO PARA O SGBD
MYSQL

BRUNO RAFAEL FLÔRES

BLUMENAU
2015

2015/1-06

BRUNO RAFAEL FLÔRES

FERRAMENTA DE GERENCIAMENTO PARA O SGBD

MYSQL

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Sistemas de Informação— Bacharelado.

Prof. Alexander Roberto Valdameri, Mestre - Orientador

BLUMENAU
2015

2015/1-06

FERRAMENTA DE GERENCIAMENTO PARA O SGBD

MYSQL

Por

BRUNO RAFAEL FLÔRES

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Alexander Roberto Valdameri, Mestre - Orientador, FURB

Membro: _____
Prof. Cláudio Ratke, Mestre - FURB

Membro: _____
Prof. Jacques Robert Heckmann, Mestre - FURB

Blumenau, 02 de julho de 2015.

Dedico este trabalho à minha família que sempre me incentivou a estudar e focar nos meus objetivos.

Um dia eu tive um sonho... Nesse sonho meu esforço era reconhecido, e minha determinação não era confundida com insanidade. Meus objetivos eram respeitados, e minha disciplina era admirada por todos.

Nina Veiga

AGRADECIMENTOS

À Deus, por todos os desafios e oportunidades.

Às minhas mães Rosane Zulma Flôres e Ivonésia Maria da Silva, por terem me apoiado em minhas decisões, e terem me ensinado que respeito, educação e honestidade é algo primordial, e que devemos tratar a todos, assim como gostaríamos de ser tratados, respeitando as diferenças de cada um.

Ao meu orientador, professor Alexander Roberto Valdameri, por ter aceitado me orientar neste trabalho, contribuindo com sugestões, incentivando e também me deixado de “pés no chão” quando eu começava a pensar alto demais.

Aos amigos do curso Maria Eduarda Demmer e Marcos Willian da Silva, que trilhamos a maior parte do percurso juntos, e vivemos muitos momentos, momentos felizes, outros nem tão felizes, mas que mesmo assim, ficarão na memória para sempre.

Dedico também aos amigos de longa data que se fazem presentes até os dias atuais, André Felipe Teixeira da Luz, Maicon Nunes Pereira e Valter José Haverroth Junior, pessoas que não posso chamar de conhecidos ou colegas, mas de verdadeiros amigos.

Aos professores do Departamento de Sistemas e Computação da Universidade Regional de Blumenau por suas contribuições durante os semestres letivos.

RESUMO

Este trabalho apresenta o desenvolvimento de um software de gerenciamento de banco de dados para o SGBD MySQL. Seu principal objetivo é permitir a manutenção de estruturas de uma base de dados, mas também, inserir, atualizar e excluir os dados armazenados. O usuário também pode alterar os textos da interface gráfica e criar o seu próprio dialeto, registrando vários dialetos. O software foi desenvolvido para plataforma *desktop* usando Java *Swing*.

Palavras chaves: MySQL. Java Swing. SGBD.

ABSTRACT

This work presents the development of a database management software for the MySQL DBMS. Your main objective is to allow the maintenance of structures in a database, but also insert, update and delete stored data. The user can also change the texts of the graphical interface and create your own dialect, registering many dialects. The software was developed for desktop platform using Java Swing.

Key-Words: MySQL. Java Swing. SGBD.

LISTA DE FIGURAS

Figura 1 - Tela principal do HeidiSQL.....	18
Figura 2 - Tela principal do MySQL Workbench	19
Figura 3 - Tela principal do FBConsole	20
Figura 4 - Tela de acesso ao Froog SQL	21
Figura 5 - Tela de desenvolvimento do Netbeans 8.0.2	23
Figura 6 - Tela de desenvolvimento do MER no MySQL WorkBench	24
Figura 7 - Casos de uso da utilização do sistema	26
Figura 8 - Casos de uso dos cadastros do sistema	26
Figura 9 - Modelo Entidade Relacionamento do Naveban.....	27
Figura 10 - Implementação da classe HistoricoComando	29
Figura 11 - Implementação da classe Get.....	30
Figura 12 - Método <code>delete</code> da classe <code>BasicConnection</code>	30
Figura 13 - Método que retorna o <i>bean</i> do banco.....	32
Figura 14 - Inserção de um registro de histórico na entidade <code>historico_comando</code>	33
Figura 15 - Implementação do método <code>insertBeans</code>	33
Figura 16 - Trecho de código responsável pela inserção registros no padrão <code>BBeans</code>	34
Figura 17 - Obtendo metadados do SGBD	35
Figura 18 - Tela de autenticação do Naveban	36
Figura 19 - Tela de gerenciamento de conexões	36
Figura 20 - Tela principal do Naveban.....	37
Figura 21 - Botões de controle de transação.....	37
Figura 22 - Botões de execução de sentença	37
Figura 23 - Visualização do consumo de memória	38
Figura 24 - Área de texto das sentenças SQL.....	38
Figura 25 - Tabela de resultados.....	39
Figura 26 - Árvore de <i>schemas</i>	39
Figura 27 - Árvore de histórico	40
Figura 28 - Barra de menus da tela principal usando o dialeto padrão da ferramenta	40
Figura 29 - Menu arquivo.....	41
Figura 30 - Menu Editar	42
Figura 31 - Menu Layout.....	42

Figura 32 - Menu Ferramentas	42
Figura 33 - Barra de menus da tela principal usando um novo dialeto cadastrado	43
Figura 34 - Menu Sobre.....	43
Figura 35 - Tela de pesquisa de texto	43
Figura 36 - Cadastro de Menu	44
Figura 37 - Cadastro de Label	45
Figura 38 - Cadastro de exemplo.....	45
Figura 39 - Cadastro de botões	46
Figura 40 - Cadastro de janelas	46
Figura 41 - Cadastro de textos.....	47
Figura 42 - Texto de exemplo	47
Figura 43 - Tela de cadastro de idioma	48
Figura 44 - Arquivo exportado para formato HTML	56
Figura 45 - Arquivo no formato de XML.....	57
Figura 46 - Arquivo exportado para arquivo XLS	57
Figura 47 - Estrutura do arquivo XLS	58
Figura 48 - Arquivo exportado para CSV	59
Figura 49 - Estrutura do arquivo CSV.....	59

LISTA DE QUADROS

Quadro 1 - Requisitos funcionais	24
Quadro 2 - Requisitos não funcionais	25
Quadro 3 - Comparativo com trabalhos correlatos.....	49
Quadro 4 - Descrição do caso de uso "Cadastro de idioma"	53
Quadro 5 - Descrição do caso de uso "Cadastro de conexão"	53
Quadro 6 - Descrição do caso de uso "Utilizar transação"	54
Quadro 7 - Descrição do caso e uso "Exportar consulta"	54
Quadro 8 - Descrição do caso de uso "Criar base de dados"	55
Quadro 9 - Descrição do caso de uso "Cadastro/alteração de textos"	55

LISTA DE SIGLAS

CSV - *Comma-separated values*
DBA - *Data Base Administrators*
DDL - *Data Definition Language*
DML - *Data Manipulation Language*
GPL - *General Public Licence*
HTML - *HyperText Markup Language*
IBM - *International Business Machines*
IDE - *Integrated Development Environment*
JDK - *Java Development Kit*
MER - *Modelo de Entidade e Relacionamento*
RF - *Requisitos Funcionais*
RNF - *Requisitos Não Funcionais*
SGBD – *Sistema Gerenciador de Banco de Dados.*
SQL - *Structured Query Language*
TCC - *Trabalho de Conclusão de Curso*
XML - *Extensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 PROBLEMA	14
1.2 JUSTIFICATIVA	14
1.3 OBJETIVOS.....	15
1.4 ESTRUTURA DO TRABALHO	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 SISTEMAS GERENCIADORES DE BANCOS DE DADOS.....	16
2.2 INTERFACES GRÁFICAS	17
2.3 FERRAMENTAS DE GERENCIAMENTO DO SGBD MYSQL.....	17
2.3.1 HeidiSQL	18
2.3.2 MySQL Workbench	19
2.4 TRABALHOS CORRELATOS	20
3 DESENVOLVIMENTO DA FERRAMENTA	22
3.1 LEVANTAMENTO DE INFORMAÇÕES	22
3.2 TÉCNICAS E FERRAMENTAS UTILIZADAS	23
3.2.1 Ferramentas utilizadas.....	23
3.2.2 Requisitos da ferramenta.....	24
3.2.3 Diagrama de casos de uso	25
3.2.4 Modelo Entidade Relacionamento	27
3.2.5 Persistência.....	28
3.2.6 Metadados	34
3.3 OPERACIONALIDADE DA FERRAMENTA.....	35
3.4 RESULTADOS E DISCUSSÃO	48
4 CONCLUSÕES	50
4.1 EXTENSÕES	50
REFERÊNCIAS	52
APÊNDICE A – Descrição dos Casos de Uso	53
APÊNDICE B – Exportação das informações	56

1 INTRODUÇÃO

Cada vez mais, novos dispositivos eletrônicos são disponibilizados no mercado, e utilizados para comunicação e compartilhamento de informações. *Smartphones*, computadores, *tablets*, relógios ou até mesmo os óculos (*google glass*), permitem que as pessoas se comuniquem, compartilhem dados, acessem redes sociais, contas bancárias, façam compras, entre muitas outras coisas. Grande parte dos serviços oferecidos na internet utiliza algum meio de persistência de dados para armazenar informações sobre os usuários que estão utilizando seu serviço. Para armazenar estas informações de forma organizada, de maneira de facilitar a manipulação ou consulta dessas informações posteriormente é necessário um sistema que gerencie a persistência e consulta a esses dados, garantindo que as informações sejam manipuladas de maneira íntegra.

Gerenciar dados é uma tarefa complexa e que ao longo dos tempos demandou a construção de um sistema capaz de responder as várias questões envolvidas no armazenamento e recuperação destes dados. Neste contexto surgem os Sistemas Gerenciadores de Bancos de Dados (SGBD) (ELMASRI; NAVATHE, 2005, p. 37).

Um SGBD é uma coleção de arquivos e programas inter-relacionados que permitem ao usuário o acesso para consultas e alterações desses dados. O maior benefício de um banco de dados é proporcionar ao usuário uma visão *abstrata* dos dados. Isto é, o sistema acaba por ocultar determinados detalhes sobre a forma de armazenamento e manutenção desses dados. (SILBERSCHATZ; KORTH; SUDARSHAN, 1999, p. 4, grifo do autor).

As informações disponibilizadas crescem rapidamente, e o valor dos dados é amplamente reconhecido, justificando a utilização de ferramentas para gerenciamento de dados no processo de armazenagem e extração de informações (RAMAKRISHNAN; GEHRKE, 2008, p. 2). Existem ferramentas que permitem visualizar as informações persistidas digitalmente em um banco de dados na forma de tabelas, algumas possuem interface gráfica, outra com a opção de utilizar apenas a linha de comando.

O trabalho aqui apresentado surgiu com a intenção criar uma ferramenta multi-idioma, que executará comandos *Structured Query Language* (SQL), possibilitando que o usuário tenha um melhor entendimento dos resultados que serão obtidos na execução de seus comandos, através de mensagens personalizadas.

1.1 PROBLEMA

No cenário atual, o usuário digita um comando SQL em uma ferramenta que faz o envio da sentença SQL ao SGBD, para que esse execute a sentença e retorne informações sobre a operação executada. O retorno, dependendo do tipo da requisição solicitada pode ser na forma de um conjunto de dados (registros), informações sobre o resultado do comando (por exemplo: função de agrupamento de dados ou quantidade de registros afetados) ou até mesmo mensagens de erros (exceções).

Apesar da existência das funções de restrição de integridade e consistência de dados nos SGBDs, existem sentenças SQL que podem causar grande perda de informações em uma base de dados. Em geral isto ocorre por descuido e, muitas vezes, por desconhecimento ou mau uso dos comandos da SQL.

Assim, este trabalho tem como objetivo, desenvolver uma ferramenta que, por meio de mensagens em uma interface gráfica, forneça informações e orientações ao usuário sobre as sentenças a serem executadas.

1.2 JUSTIFICATIVA

As ferramentas de gerenciamento de banco de dados disponíveis para utilização hoje, possuem muitas funcionalidades para facilitar a rotina dos usuários de SGBDs. Tarefas como criar e alterar a estrutura de uma tabela, inserir, remover e alterar dados, entre outras funcionalidades são completamente possíveis de serem realizadas por qualquer usuário, desde que este possua as devidas permissões. Mesmo que o usuário não tenha conhecimento da sintaxe dos comandos SQL, as aplicações oferecem meios para que o usuário realize essas tarefas por meio de uma interface gráfica, abstraindo do usuário a necessidade de conhecer a linguagem.

Baseado no cenário atual, este trabalho apresenta uma ferramenta que servirá como um suporte para os usuários interessados em aprender SQL. Esta ferramenta deverá emitir mensagens que forneçam ao usuário informações sobre o possível resultado de requisição ao SGBD, auxiliando o usuário em um melhor entendimento da prática de instruções SQL.

1.3 OBJETIVOS

O objetivo deste trabalho é apresentar o desenvolvimento de uma ferramenta de gerenciamento para o SGBD MySQL.

Os objetivos específicos deste trabalho são:

- a) permitir alteração das mensagens emitidas pela aplicação;
- b) permitir que o usuário digite mais de uma sentença SQL em uma mesma área de texto, separando as sentenças por ponto e vírgula;
- c) possibilitar a configuração de mais de um idioma.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está estruturado em 4 (quatro) capítulos que estão descritos a seguir.

No primeiro capítulo é apresentada uma introdução ao trabalho desenvolvido, seguido pela justificativa e objetivos do trabalho.

No segundo capítulo é apresentada as pesquisas realizadas sobre Sistemas Gerenciadores de Banco de Dados (SGBD), ferramentas de gerenciamento para o SGBD MySQL, e trabalhos correlatos.

No terceiro capítulo é apresentado o levantamento de informações acerca da ideia de desenvolvimento desta ferramenta, em seguida é apresentada técnicas e ferramentas utilizadas, operacionalidade da ferramenta, fechando o capítulo com resultados e discussões.

No quarto capítulo é apresentada as conclusões do desenvolvimento deste trabalho e sugestões de implementações futuras para extensões.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo busca explicar algumas ferramentas de conexão com o SGBD MySQL, descrever a função dos SGBDs, e detalhar alguns trabalhos e softwares correlatos.

2.1 SISTEMAS GERENCIADORES DE BANCOS DE DADOS

A história dos Sistemas Gerenciadores de Bancos de Dados (SGBD) começou na década de 1960 dentro da *International Business Machines* (IBM), quando a empresa sentiu a necessidade de otimizar o processo de armazenamento e organização de arquivos. Em 1970, Ted Codd fez a primeira publicação de um artigo falando sobre Banco de Dados Relacionais, onde relatava sua visão onde usuários não técnicos fossem capazes de acessar informações em um banco de dados através de comandos em inglês, onde as informações estariam armazenadas em estruturas equivalentes a tabelas (SANCHES, 2005).

Um **sistema de banco de dados** é basicamente apenas um *sistema computadorizado de manutenção de registros*. O **banco de dados**, por si só, pode ser considerado como o equivalente eletrônico de um armário de arquivamento; ou seja, ele é um repositório ou recipiente para uma coleção de arquivos de dados computadorizados. Os usuários de um sistema podem realizar (ou melhor, solicitar que o sistema realize) diversas operações envolvendo tais arquivos. (DATE, 2004, p. 3, grifos do autor).

É possível fazer uma analogia entre um banco de dados e um porta-arquivos, onde deposita-se as informações de maneira organizada, armazenadas em um local centralizado e de comum acesso. Um executivo pode inserir um arquivo (registro) em uma das partições (tabela) do porta-arquivo (banco de dados), e a secretária pode consultar esses arquivos para obter informações que sejam de seu interesse. Neste cenário, temos uma solução para o compartilhamento das informações, porém temos um problema, o acesso livre a essas informações.

De acordo com Gupta e Gietz (1990) “Para se ter acesso aos registros armazenados ou mesmo cadastrar novos registros, precisamos de um sistema que gerencie o banco de dados. Este SGBD é que torna possíveis as operações com o conteúdo do arquivo, como – "Traga-me este arquivo", "Atualize este registro””.

Utilizando a analogia anteriormente citada, o SGBD é como se fosse o único agente com acesso ao arquivo, e todas as pessoas que desejam realizar consultas, atualizações ou adição de informações conteúdo do porta-arquivo, tivessem a necessidade de solicitar esta ação ao agente responsável por este porta-arquivo, centralizando o gerenciamento das informações contidas no mesmo.

Para realizar operações neste porta-arquivos, é necessário passar instruções para o agente responsável pelo gerenciamento do porta arquivos, utilizando a linguagem SQL. A SQL não é uma linguagem especificamente criada para desenvolver sistemas, como são as linguagens de programação PASCAL, C, COBOL, entre outras. Ela é somente uma linguagem utilizada para facilitar o acesso de informações (por meio de consultas, atualizações e manipulações de dados) armazenadas em bancos de dados (MANZANO, 2002).

2.2 INTERFACES GRÁFICAS

Em meados da década de 80, nasceu uma nova forma de interagir com o mundo digital. Esta nova forma de interação mostrou-se bastante intuitiva e amigável, pois ela utilizava representações iconográficas que, quando clicadas, executavam instruções que antes era possível somente de serem feitas por meio de uma interface textual, escrito em uma tela preta, com o passar do tempo, esta nova forma de interação com o computador foi substituindo a forma original (STEFANELLI, 2015).

O objetivo de uma interface gráfica em um computador é facilitar o uso, isso significa que, a utilização de um computador por meio de uma interface gráfica deve trazer uma experiência fácil, agradável e eficaz do ponto de vista do usuário.

2.3 FERRAMENTAS DE GERENCIAMENTO DO SGBD MYSQL

Esta seção apresenta duas ferramentas de gerenciamento ao SGBD MySQL bem como, indica alguns pontos positivos que se destacam em cada ferramenta.

2.3.1 HeidiSQL

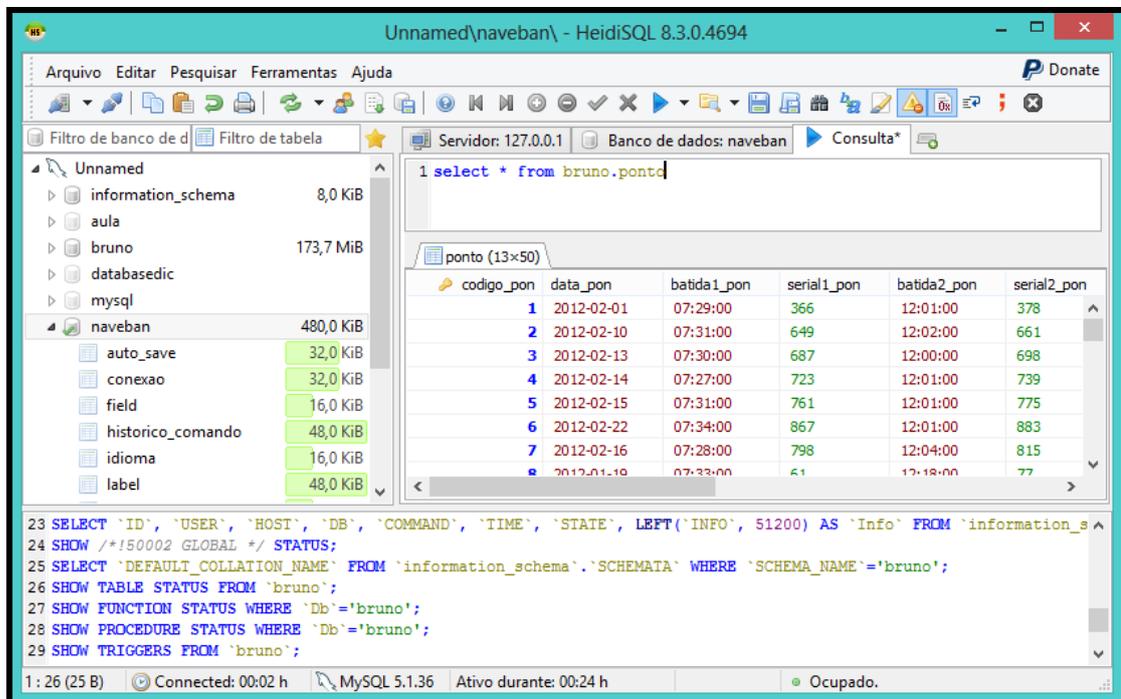
Esta ferramenta foi criada por Ansgar Becker e é distribuída atualmente sob licença *GNU General Public Licence* (GPL), permitindo que além de utilizar a ferramenta, quem tenha interesse em saber como ela funciona internamente, faça o *download* do código fonte e adapte às suas necessidades. Atualmente o projeto é mantido por vários colaboradores de diversas partes do mundo (HEIDISQL, 2014).

As funcionalidades que se destacam neste software são:

- visualização do tamanho e quantidade de registros de uma tabela em uma interface gráfica;
- o *backup* criado utilizando interface gráfica;
- alteração dos registros da tabela por meio de interface gráfica (neste caso, somente quando a mesma possui chave primária e o *select* conter apenas uma tabela);

Na Figura 1 é apresentada uma imagem da tela principal do HeidiSQL.

Figura 1 - Tela principal do HeidiSQL



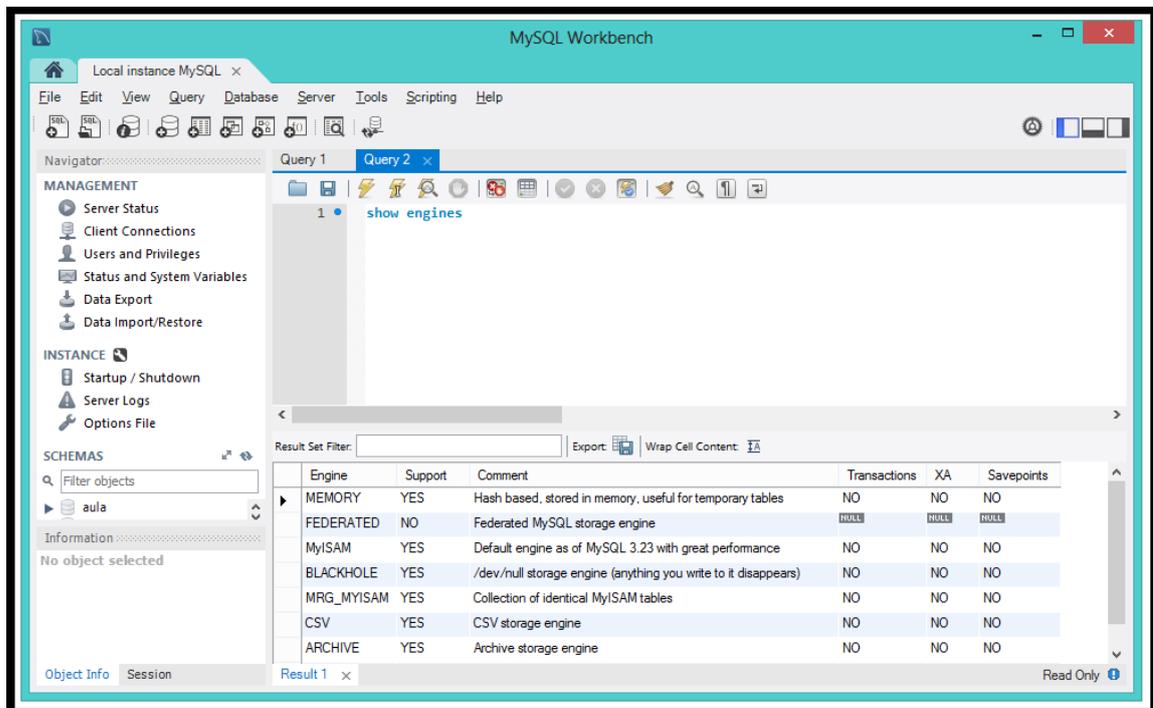
2.3.2 MySQL Workbench

Esta ferramenta é mantida pela ORACLE. Conforme documentação disponível, este software é sucessor do MySQL Query Browser (MYSQL, 2014). Suas funcionalidades foram incrementadas e suas ferramentas foram aperfeiçoadas para tornar o software mais robusto e intuitivo, tornando uma ferramenta de fácil utilização para novos usuários da ferramenta, e uma poderosa ferramenta de gerenciamento de SGBD por parte de *Data Base Administrators* (DBAs) experientes. Alguns recursos disponíveis nesta ferramenta são apresentados a seguir (MYSQL, 2014):

- a) monitoramento da utilização dos recursos do banco de dados;
- b) ferramenta para obter o Modelo de Entidade e Relacionamento (MER) de uma base de dados;
- c) controlar e cadastrar usuários por meio de interface gráfica, bem como ceder e revogar permissões;
- d) monitorar e remover usuários conectados ao SGBD.

Na Figura 2 é apresentada a uma imagem da tela principal do MySQL Workbench.

Figura 2 - Tela principal do MySQL Workbench

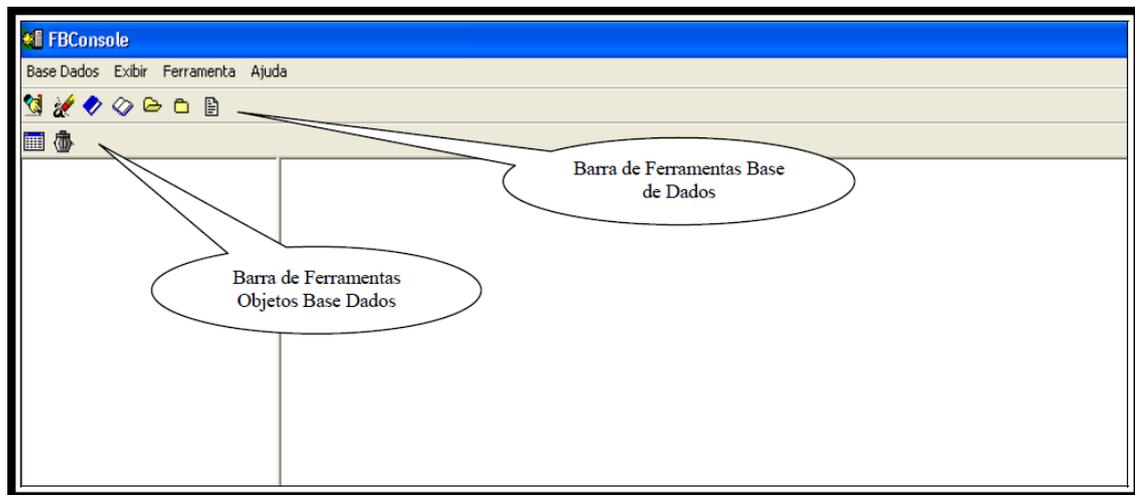


2.4 TRABALHOS CORRELATOS

É possível citar os Trabalhos de Conclusão de Cursos (TCC) de Carlos Eduardo Werner e de Paulo Fernando Odwazny como trabalhos correlatos. Ambos os alunos cursaram Bacharelado em Ciências da Computação na Universidade Regional de Blumenau.

Werner (2004) propôs em seu trabalho, a construção de um software para o gerenciamento do banco de dados Firebird batizado com o nome de FBConsole. Seu objetivo era desenvolver uma ferramenta que possibilitasse a administração de um banco de dados Firebird, pois na época havia carência de ferramentas *open-source* para realizar este trabalho. Na Figura 3 é apresentada a tela inicial do FBConsole.

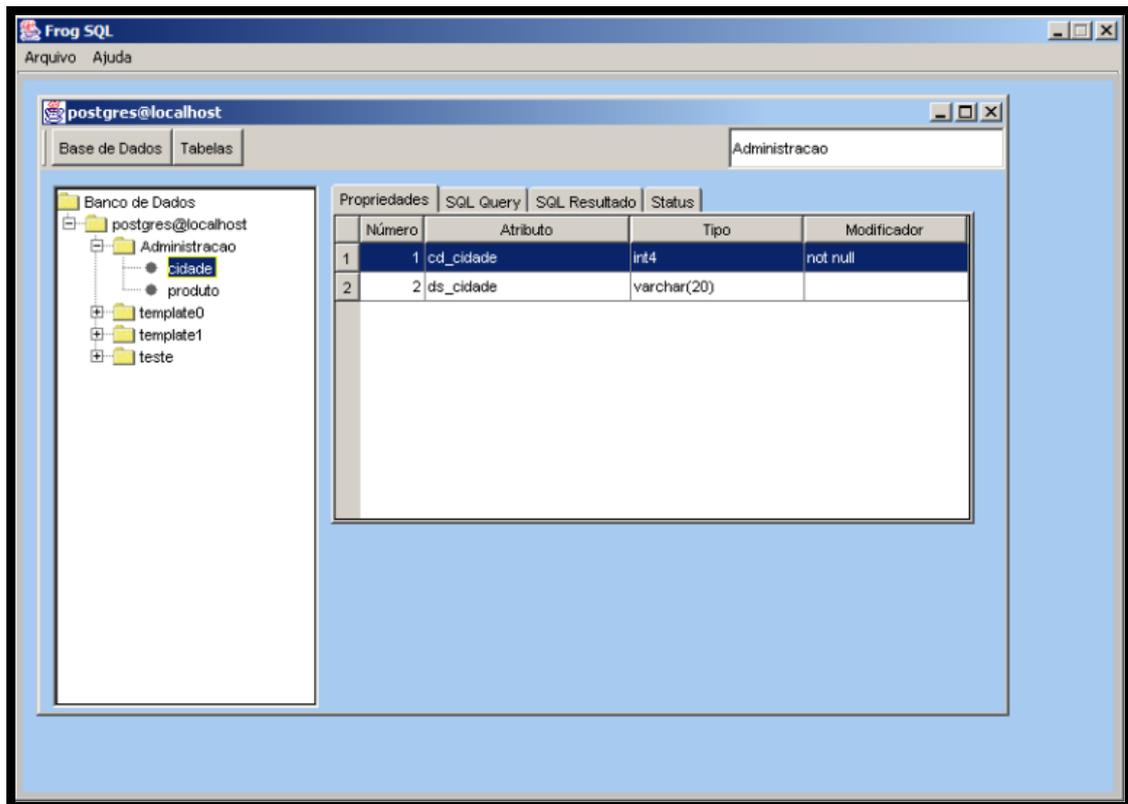
Figura 3 - Tela principal do FBConsole



Fonte: Werner (2004, p. 40).

Odwasny (2003) propôs desenvolver uma ferramenta com interface gráfica, capaz de manipular as estruturas e os dados em um SGBD PostgreSQL. O nome dado à ferramenta foi Frog SQL. Seus objetivos eram desenvolver uma interface gráfica para interação do usuário com o SGBD e disponibilizar as principais funcionalidades do SGBD na ferramenta que deveria ser capaz de executar sentenças de consulta, comandos *Data Definition Language* (DDL) e comandos *Data Manipulation Language* (DML). Na Figura 4 é apresentada a tela de acesso à ferramenta proposta pelo Odwasny.

Figura 4 - Tela de acesso ao Froog SQL



Fonte: Odwazny (2003, p.48).

3 DESENVOLVIMENTO DA FERRAMENTA

Neste capítulo será explanado o desenvolvimento da ferramenta Naveban, nele será descrito a especificação dos requisitos funcionais e não funcionais, diagrama de classes, diagrama de atividades, o modelo de entidade e relacionamento, e os diagramas de casos de uso. Também será apresentado implementações e ferramentas utilizadas para desenvolvimento e operação do sistema, a operacionalidade da implementação e os resultados obtidos.

3.1 LEVANTAMENTO DE INFORMAÇÕES

A ideia de desenvolver este software surgiu nas aulas de banco de dados. Durante as aulas, o professor deixou aberto para os alunos utilizarem a ferramenta que mais lhe agradasse para fazer os exercícios práticos. Para quem já tem algum envolvimento com banco de dados, e, já utiliza alguma ferramenta o cenário é perfeito, podendo utilizar uma ferramenta que já tem afinidade nas aulas. Porém, para as pessoas que nunca tiveram contato com ferramentas, fica difícil optar pela ferramenta A ou B. Normalmente elas vão optar por utilizar a ferramenta que alguém próximo utiliza, para assim ter suporte durante as aulas.

Neste cenário, surgiu a ideia de desenvolver o Naveban, uma ferramenta que possibilitasse a personalização dos textos exibidos pela interface, permitindo o cadastro de diferentes dialetos, adequando os textos da interface, de acordo com o nível de conhecimento individual de cada usuário. Para auxiliar melhor os usuários, um dos requisitos da ferramenta era criar *logs* dos comandos executados em uma base centralizada, possibilitando o professor selecionar os dados, e verificar o desempenho e dificuldades de cada aluno, auxiliando o professor a identificar alunos que estejam com alguma dificuldade.

Os benefícios da utilização desta ferramenta vão além do auxílio do professor. Nem sempre a mesma máquina está disponível para ser utilizada pelos alunos, então, como os *logs* são centralizados em uma base localizada em um servidor remoto, os alunos poderão utilizar qualquer máquina para utilizar o mesmo, e as informações de comandos executados, e textos estarão disponíveis.

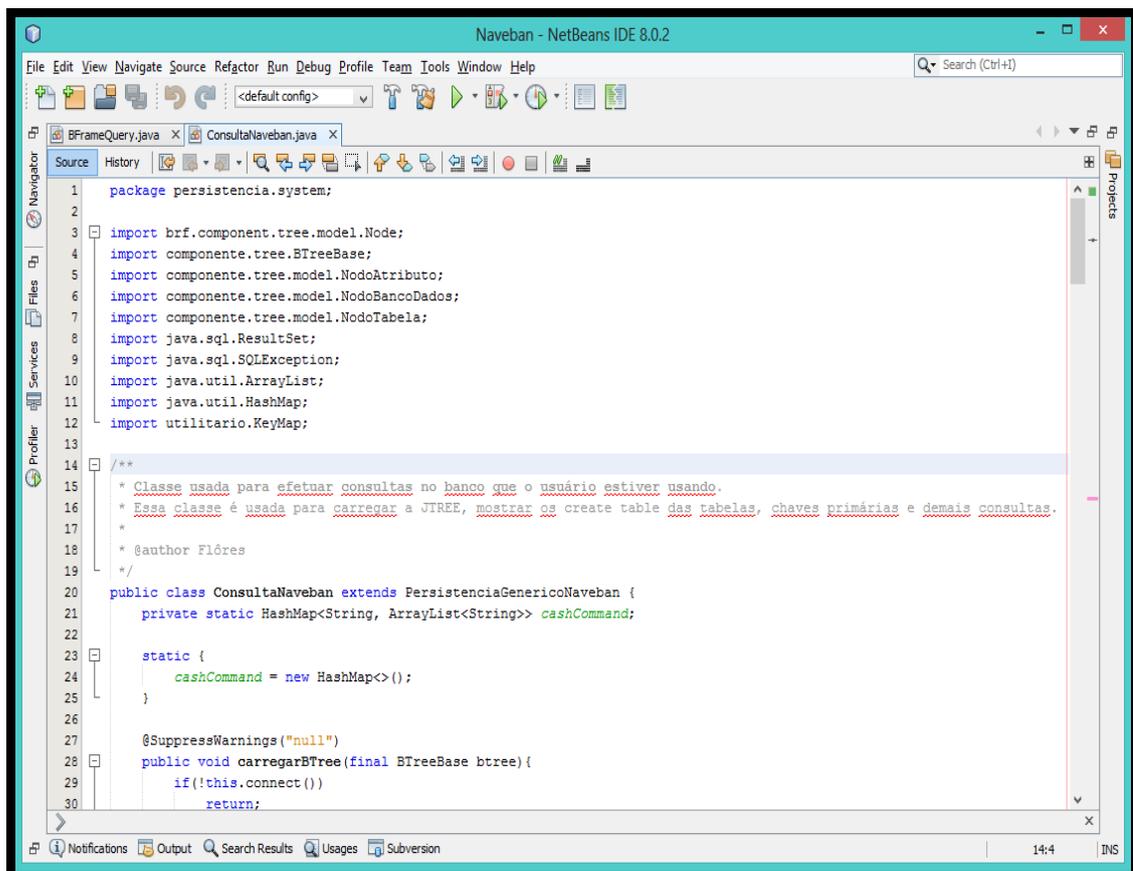
3.2 TÉCNICAS E FERRAMENTAS UTILIZADAS

Nesta seção são mostradas as técnicas e ferramentas utilizadas para desenvolvimento da aplicação.

3.2.1 Ferramentas utilizadas

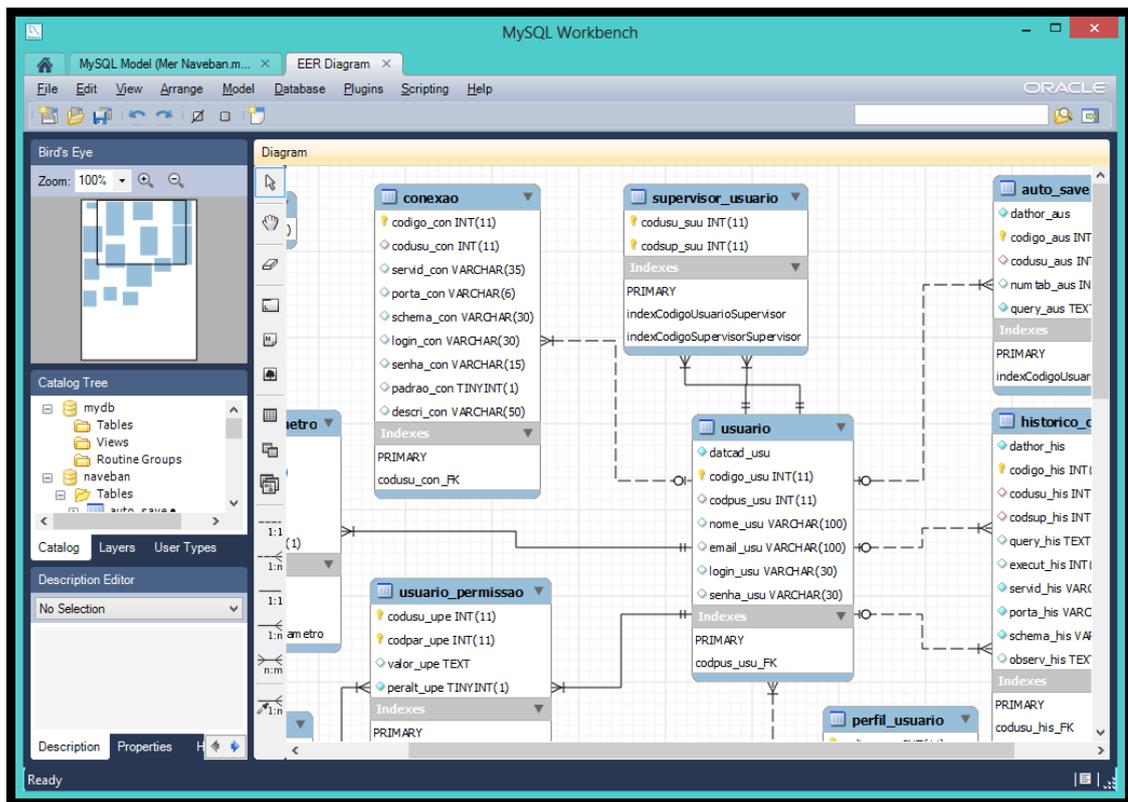
O software foi desenvolvido usando ferramentas livres de licenças pagas. Como linguagem de desenvolvimento foi escolhido Java. O Kit de Desenvolvimento Java (*Java Development Kit, JDK*) instalado era a versão 1.7.0 *build 67*, arquitetura 64 *bits*. Para desenvolvimento do software foi utilizado o Ambiente Integrado de Desenvolvimento (*Integrated Development Environment, IDE*) Netbeans 8.0.2. A Figura 5 apresenta a tela de desenvolvimento da IDE.

Figura 5 - Tela de desenvolvimento do Netbeans 8.0.2



Para desenhar o banco de dados e fazer o Modelo de Entidade e Relacionamento (MER) foi usado a ferramenta MySQL WorkBench 5.2.44. A Figura 6 apresenta a tela de desenvolvimento de MER da ferramenta.

Figura 6 - Tela de desenvolvimento do MER no MySQL WorkBench



3.2.2 Requisitos da ferramenta

Nesta subseção serão apresentadas funcionalidades da aplicação desenvolvidas. O Quadro 1 apresenta os Requisitos Funcionais (RF) da aplicação e os casos de uso ligados a ela.

Quadro 1 - Requisitos funcionais

Requisitos Funcionais	Caso(s) de uso
RF01: O sistema deverá permitir o cadastro de idiomas.	UC01, UC06
RF02: O sistema deverá armazenar o cadastro de mais de uma instância de conexão.	UC02
RF03: O sistema deverá permitir o usuário iniciar uma transação e salvar/desfazer.	UC03
RF04: O sistema deverá permitir o usuário exportar os dados da consulta realizada para arquivos nos formatos de HTML, XML, XLS e CSV.	UC04

RF05: O sistema deverá realizar a criação e manutenção da estrutura de dados necessária para seu funcionamento sem a necessidade do usuário executar script.	UC05
RF06: O sistema deverá criar logs dos comandos executados pelo usuário.	

O próximo quadro, o Quadro 2, apresenta a lista de Requisitos Não Funcionais (RNF) da aplicação desenvolvida.

Quadro 2 - Requisitos não funcionais

Requisitos Não Funcionais
RNF01: O sistema deverá ser implementado em Java 1.7
RNF02: O sistema deverá utilizar banco de dados MySQL para persistência de <i>logs</i> de usuários.
RNF03: O sistema deverá utilizar o banco de dados HSQLDB para persistência de suas configurações locais.

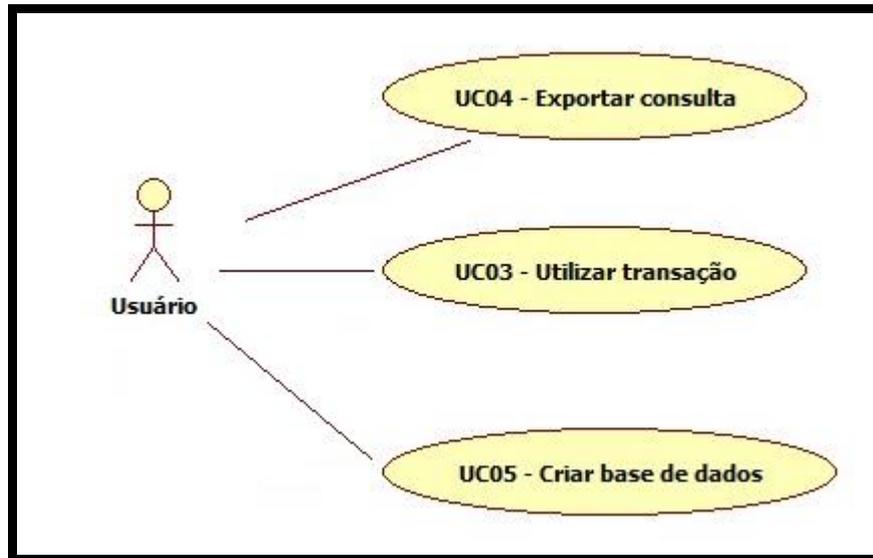
Para utilização em ambientes com mais de um microcomputador na mesma rede, é possível optar por salvar as informações da aplicação em um banco de dados remoto, atendendo ao requisito não funcional 02. Desta maneira, o usuário poderá utilizar a aplicação em qualquer equipamento da rede, desde que este esteja apontando para o mesmo servidor, suas informações serão carregadas. Caso nenhuma configuração seja realizada na aplicação, esta irá persistir tudo em um banco de dados local, atendendo ao requisito não funcional 03.

O banco de dados local é criado com a utilização da API java HSQLDB. Este software permite criar um banco de dados para aplicações, sem a necessidade de realizar qualquer instalação ou configuração de ambiente por parte do usuário, bastando o desenvolvedor informar um caminho do banco de dados para ser utilizado.

3.2.3 Diagrama de casos de uso

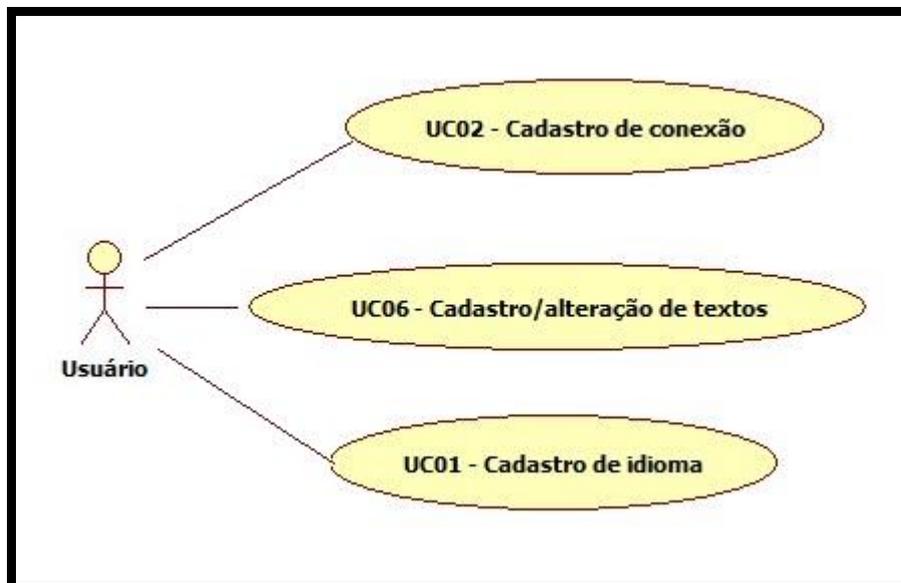
Nesta subseção serão apresentados os diagramas de casos de uso da aplicação. Um maior detalhamento dos casos de uso é feito no Apêndice A. Na Figura 7 são visualizados os casos de uso referentes à utilização da aplicação e dos atores envolvidos.

Figura 7 - Casos de uso da utilização do sistema



Na Figura 8 são visualizados os casos de uso que envolvem cadastros de informações no sistema.

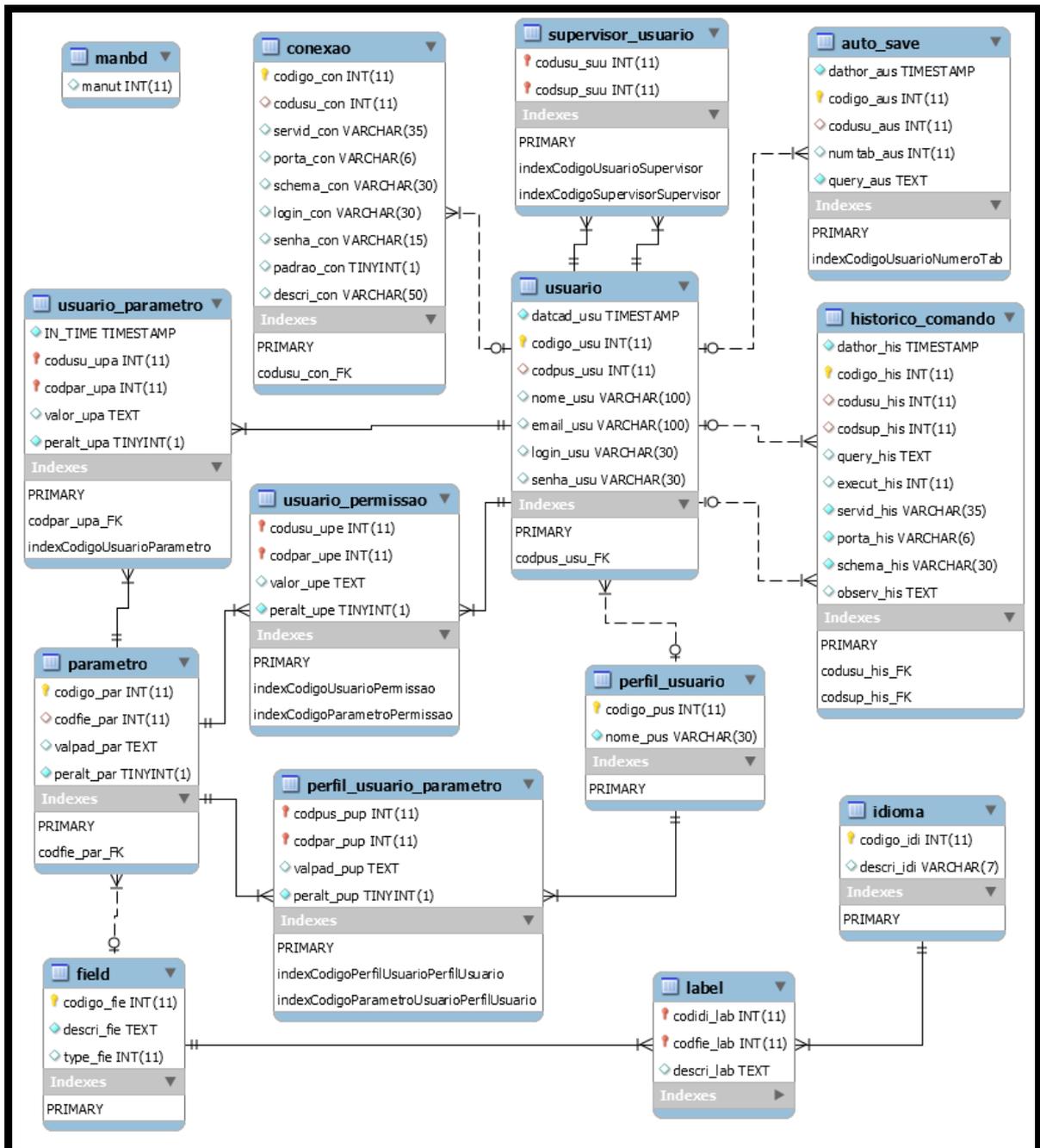
Figura 8 - Casos de uso dos cadastros do sistema



3.2.4 Modelo Entidade Relacionamento

A Figura 9 apresenta o Modelo Entidade Relacionamento com as tabelas e atributos que compõem o Naveban. O MER apresentado já contempla estruturas projetadas para implementações futuras, as quais serão listadas nas sugestões de trabalhos futuros.

Figura 9 - Modelo Entidade Relacionamento do Naveban



A seguir é apresentada uma breve descrição de cada entidade criada e seu propósito no banco de dados da aplicação:

- a) `usuario`: entidade que armazena os usuários da aplicação;
- b) `supervisor_usuario`: entidade que armazena o supervisor do usuário;
- c) `perfil_usuario`: entidade que armazena os perfis cadastrados no software;
- d) `usuario_parametro`: entidade que armazena os parâmetros específicos para cada usuário;
- e) `perfil_usuario_parametro`: entidade que armazena os valores dos parâmetros cadastrados para os perfis;
- f) `usuario_permissao`: entidade que armazena os usuários que tem permissão de alterar parâmetros e quais parâmetros podem ser alterados por eles;
- g) `parametro`: entidade que armazena os parâmetros do sistema;
- h) `field`: entidade que armazena os códigos e textos padrões usados no software;
- i) `idioma`: entidade que armazena os idiomas cadastrados no sistema;
- j) `label`: entidade que armazena os textos cadastrados em cada idioma;
- k) `conexao`: entidade que armazena as informações da conexão feita pelo usuário com o SGBD;
- l) `auto_save`: entidade que armazena os textos digitados e as tabs criadas na aplicação;
- m) `historico_comando`: entidade que armazena as sentenças executadas pelos usuários e os dados da conexão;
- n) `manbd`: entidade que armazena o número da manutenção da base de dados do Naveban.

3.2.5 Persistência

Para cada tabela criada no banco, foi criada uma classe no projeto para manipulação dos dados. A Figura 10 apresenta a implementação da classe `HistoricoComando.java`, representante da entidade `historico_comando` do banco de dados.

Figura 10 - Implementação da classe HistoricoComando

```

16  * @author Flôres
17  */
18  @Table(desc = "historico_comando")
19  public class HistoricoComando implements BBeans {
20      private int codigo_his;
21      private Usuario codusu_his;
22      private Usuario codsup_his;
23      private String query_his;
24      private Date dathor_his;
25      private int execut_his;
26      private String servid_his;
27      private String porta;
28      private String schema_his;
29      private String observ_his;
30
31      public HistoricoComando() {
32      }
33
34      public HistoricoComando(int codigo_his, Usuario codusu_his, Usuario codsu
35
36
37
38      public HistoricoComando(int codigo_his, Usuario codusu_his, Usuario codsu
39
40
41
42
43
44
45
46
47
48
49
50
51      @PrimaryKey
52      @Get(desc = "codigo_his", type = BBEANS_INT)
53      public int getCodigo_his() {
54          return codigo_his;
55      }
56
57      @Set(desc = "codigo_his", type = Integer.class)
58      public void setCodigo_his(int codigo_his) {
59          this.codigo_his = codigo_his;
60      }
61

```

Na Figura 10 é possível ver anotações na classe que servem para que o projeto responsável por realizar a persistência e manipulação dos dados localize as tabelas, atributos e tipos de atributos para montar a sentença SQL de acordo com a finalidade da operação. A Figura 11 apresenta como é criada a classe de anotação Get, e seus parâmetros desc e type.

Figura 11 - Implementação da classe Get

```

1 package brf.reflect;
2
3 import ...4 lines
4
5
6
7
8 /**
9  * @author Flôres
10 */
11 @Target(value = {ElementType.METHOD})
12 @Retention(RetentionPolicy.RUNTIME)
13 public @interface Get {
14     public String desc();
15     public int type();
16 }

```

A Figura 12 apresenta a classe `BasicConnection.java`. Esta classe, é a base de todas as classes de comunicação com banco de dados, ela é uma classe abstrata contém os métodos básicos de comunicação com o banco pré-implementados que tem um funcionamento padrão independente do SGBD utilizado. Ela também define alguns métodos abstratos a serem implementados pelas classes filhas.

Figura 12 - Método delete da classe BasicConnection

```

584 protected final boolean delete(BBeans beans) throws Exception {
585     ArrayList<Method> primaryKeys = BClassUtil.getMethods(beans.getClass(), PrimaryKey.class);
586
587     if(primaryKeys.isEmpty())
588         throw new EmptyPrimaryKeyException("Chave primária não cadastrada na classe " + beans.getClass().getCanonicalName());
589
590     StringBuilder sql = new StringBuilder(150);
591     sql.append("DELETE FROM ").append(beans.getClass().getAnnotation(Table.class).desc()).append(' ').append(" WHERE ");
592
593     for(Method method : primaryKeys){
594         String s = method.getAnnotation(Get.class).desc();
595         sql.append(' ').append(s).append('=').append(':').append(s).append(" AND");
596     }
597
598     sql.delete(sql.length()-4, sql.length());
599
600     PreparedStatement ps = this.getPreparedStatement(sql.toString());
601     for (Method metodo : primaryKeys) {
602         try {
603             ps.setObject(primaryKeys.indexOf(metodo) + 1, metodo.invoke(beans));
604         } catch (Exception x) {
605             System.err.println("SQL = [" + sql.toString() + "]");
606             System.err.println("Parâmetro = " + metodo.getAnnotation(Get.class).desc());
607             System.err.println("Índice = " + primaryKeys.indexOf(metodo) + 1);
608             if(!metodo.isAnnotationPresent(Encrypt.class))
609                 System.err.println("Valor = " + metodo.invoke(beans));
610             else
611                 System.err.println("Valor = *****");
612             throw x;
613         }
614     }
615
616     return ps.executeUpdate() > 0;
617 }

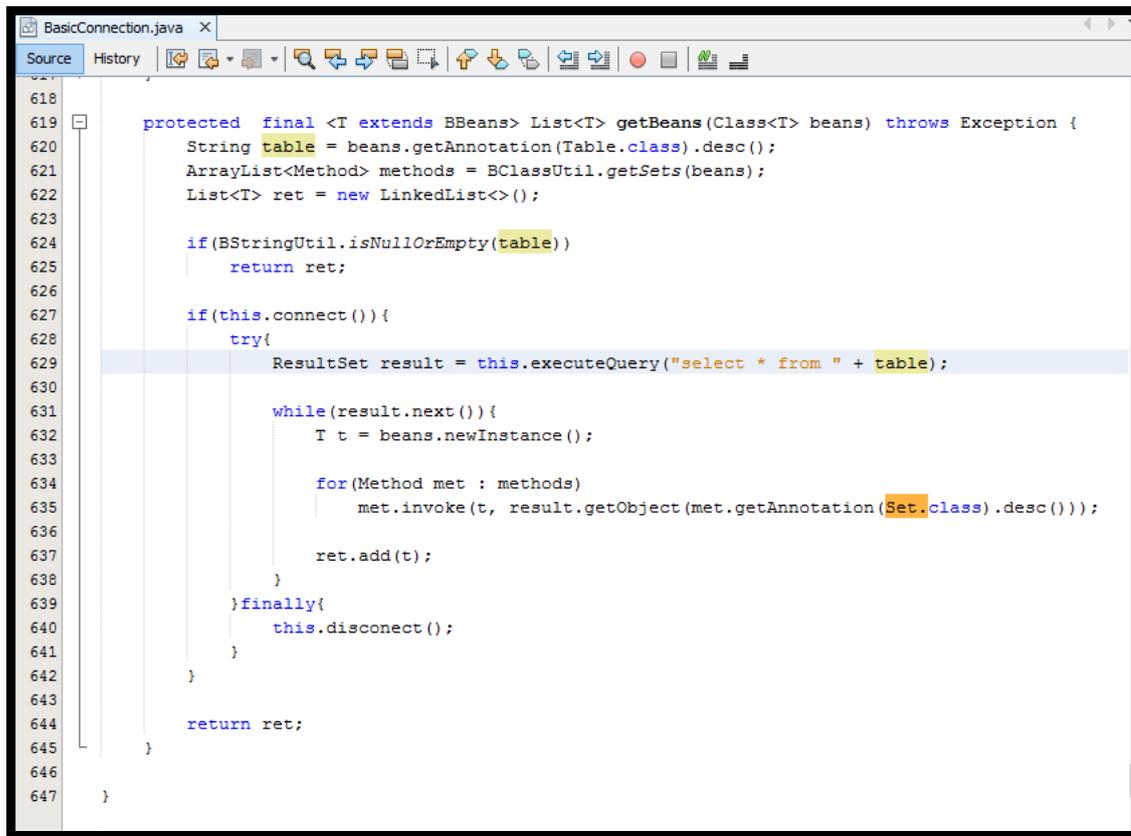
```

Na imagem é possível observar a implementação do método `delete`, onde através de uma classe utilitária, obtém-se os métodos que contém anotações de `PrimaryKey`, e, a partir deles, acessamos os valores correspondentes às chaves primárias do objeto e a sentença SQL que fará a remoção do registro correspondente ao objeto no banco de dados é montada.

A anotação `PrimaryKey` não contém nenhum parâmetro a ser passado, portanto, ela é utilizada em conjunto com a classe `Get`, que através do atributo `desc`, possibilita a obtenção do nome da coluna no banco, correspondente ao atributo da classe. A extração do valor da variável no objeto é feita através do uso de reflexão.

A Figura 13 apresenta a implementação do método que obtém os registros do banco e retorna uma lista com objetos correspondentes, o tipo de objeto e lista são carregados usando uma funcionalidade da linguagem java conhecida por *generics*¹. Para retorno desses objetos, o método recebe a referência da classe a ser retornada por parâmetro, e através deste parâmetro, é obtido as instâncias de classe. Com o auxílio de um método utilitário, obtém-se os métodos responsáveis por setar o valor nos atributos da classe, e carrega-se os registros do banco de dados fazendo uma iteração da lista de métodos.

¹ Com *generics* é possível declarar métodos, classes e interfaces que aceitam determinados valores relacionados com o valor definido na declaração do mesmo.

Figura 13 - Método que retorna o *bean* do banco


```

618
619     protected final <T extends BBeans> List<T> getBeans(Class<T> beans) throws Exception {
620         String table = beans.getAnnotation(Table.class).desc();
621         ArrayList<Method> methods = BClassUtil.getSets(beans);
622         List<T> ret = new LinkedList<>();
623
624         if(BStringUtil.isNullOrEmpty(table))
625             return ret;
626
627         if(this.connect()){
628             try{
629                 ResultSet result = this.executeQuery("select * from " + table);
630
631                 while(result.next()){
632                     T t = beans.newInstance();
633
634                     for(Method met : methods)
635                         met.invoke(t, result.getObject(met.getAnnotation(Set.class).desc()));
636
637                     ret.add(t);
638                 }
639             }finally{
640                 this.disconnect();
641             }
642         }
643
644         return ret;
645     }
646
647 }

```

Na Figura 14 pode-se visualizar como são persistidos os históricos de comandos executados pelos usuários na aplicação. No início do método é declarado um objeto do tipo `HistoricoComando` que será utilizado para persistir a sentença digitada pelo usuário, e alguns dados para posterior consulta. Caso ocorra alguma exceção na tentativa de executar o comando, é criado um histórico com a mensagem retornada do banco de dados para que fique registrado o comando que o usuário tentou executar e também a exceção lançada, caso contrário, ao finalizar a execução do método, caso nenhuma exceção tiver sido lançada (objeto `his` for nulo), um histórico com o comando é criado. Na linha 153 tem-se que este histórico é adicionado ao método `setHistoricoComando` da classe `BFrameQuery`, isso é feito para que a interface gráfica da aplicação atualize e carregue o histórico na árvore de históricos. Logo em seguida é chamado o método `persistHistorico`, e este é o responsável por persistir o objeto passado no banco de dados por meio do método `insertBeans`.

Figura 14 - Inserção de um registro de histórico na entidade historico_comando

```

134     }
135     }
136
137     rows = new ArrayList<>();
138     mapUpdates = new MapUpdate();
139
140     return (sqlStartsWith(sqlLower, "describe", "show")) ? processarResultSetNormalmente(numeroDeColu
141 } catch (Exception xx) {
142     tableLabels = new String[0];
143     this.fireTableChanged();
144     int excep = BExceptionUtil.getException(xx);
145     his = new HistoricoComando(0, ParametrosUsuario.getCurrentUser(), null, command.toString(),
146         new Date(), excep, ParametrosUsuario.getConnection(), xx.getLocalizedMessage());
147     throw xx;
148 } finally {
149     if(his == null) {
150         his = new HistoricoComando(0, ParametrosUsuario.getCurrentUser(), null, command.toString(),
151             new Date(), 0, ParametrosUsuario.getConnection(), null);
152     }
153     BFrameQuery.setHistoricoComando(his);
154     persistHistorico(his);
155     this.lastCommand = new StringBuilder(command);
156     this.fireTableChanged();
157 }
158
159
160 private void persistHistorico(HistoricoComando his) {
161     try {
162         ParametrosSistema.getPersistenceSystem().insertBeans(his);
163     } catch (Throwable ex) {
164         Catch.captureException(CatchObject.get(ex, "Erro ao salvar log"));
165     }
166 }

```

Na Figura 15 é possível visualizar a implementação do método insertBeans, que sobrescreve o método de sua interface PersistenciaBasic. Como o Naveban persiste seus dados tanto em banco de dados remoto, quanto local, essa interface é utilizada para manter as classes responsáveis pela persistência nas diferentes plataformas compatíveis.

Figura 15 - Implementação do método insertBeans

```

437
438     @Override
439     public boolean insertBeans(BBeans beans) throws Exception {
440         return super.insert(beans);
441     }
442
443     @Override
444     public boolean insertBeans(Collection<? extends BBeans> beans) throws Exception {
445         return super.insert(beans);
446     }
447

```

Como se pode observar, a sobrescrita deste método na classe `HSQLComunication` faz uma chamada para o método `insert` da super classe que é a `BasicConnection`. Na Figura 16 é mostrado um trecho de código que monta a sentença SQL responsável por inserir registros no banco de dados.

Figura 16 - Trecho de código responsável pela inserção registros no padrão BBeans

```

420 for (Method metodo : metodos) {
421     if (metodo.isAnnotationPresent(PrimaryKey.class)) {
422         String column = metodo.getAnnotation(Get.class).desc();
423         primaryKeys.add(column);
424     } else {
425         String column = metodo.getAnnotation(Get.class).desc();
426         columns.add(column);
427     }
428 }
429
430 if(primaryKeys.isEmpty())
431     throw new EmptyPrimaryKeyException("Chave primária não cadastrada na classe " + beans.getClass().getCanonicalName());
432
433 sql.append("INSERT INTO ").append(beans.getClass().getAnnotation(Table.class).desc()).append(' ').append('(');
434 for(String s : primaryKeys)
435     sql.append(' ').append(s).append(',');
436
437 for(String s : columns)
438     sql.append(' ').append(s).append(',');
439
440 sql.deleteCharAt(sql.length() -1).append(" VALUES (");
441
442 for(String s : primaryKeys){
443     sql.append(" ?,");
444     all.add(s);
445 }
446
447 for(String s : columns){
448     sql.append(" ?,");
449     all.add(s);
450 }
451
452 sql.deleteCharAt(sql.length() -1).append(")");

```

O método monta a sentença usando os métodos mapeados com as anotações `Get`, que permitem termos acesso aos nomes das colunas da tabela em que o registro será inserido.

3.2.6 Metadados

O dicionário de dados pode ser considerado um banco de dados isolado (um banco de dados do sistema, não do usuário), ele contém dados sobre os dados, ou seja, definições de outros objetos do sistema, em vez de somente “dados crus” (DATE, 2004, p. 39).

O padrão SQL inclui especificações de um catálogo padrão chamado *Information Schema* (esquema de informações). O *Information Schema* consiste em um conjunto de tabelas SQL cujo conteúdo reproduz efetivamente, de uma forma definida com precisão, todas as definições de todos os outros esquemas no catálogo em questão (DATE, 2004, p. 74).

A Figura 17 mostra como é feito o carregamento dos metadados do SGBD pelo Naveban.

Figura 17 - Obtendo metadados do SGBD

```

35 String sql = "SELECT"
36           + " TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME,"
37           + " if(IS_NULLABLE like 'NO', 0, 1) as IS_NULLABLE"
38           + ", concat(COLUMN_TYPE, ' ', EXTRA), COLUMN_KEY"
39           + " FROM information_schema.`COLUMNS`"
40           + " WHERE table_schema NOT LIKE 'naveban'"
41           + " AND table_schema NOT LIKE 'mysql'"
42           + " AND table_schema NOT LIKE 'information_schema'"
43           + " ORDER BY TABLE_SCHEMA, TABLE_NAME, ORDINAL_POSITION";
44
45 try {
46     ResultSet r = this.executeQuery(sql);
47     String bd = "";
48     String tabela = "";
49     Node nodo = null;
50     Node root = btree.getRoot();
51     while (r.next()){
52         if(!bd.equals(r.getString(1))){
53             bd = r.getString(1);
54             root.add(new NodoBancoDados(bd, ""));
55             tabela = r.getString(2);
56             nodo = root.getChildByName(bd);
57             nodo.add(new NodoTabela(tabela, ""));
58             nodo = nodo.getChildByName(tabela);
59         }else if(!tabela.equals(r.getString(2))){
60             tabela = r.getString(2);
61             root.getChildByName(r.getString(1)).add(nodo = new NodoTabela(tabela, ""));
62         }
63
64         KeyMap.put(tabela, KeyMap.TABLE);
65         nodo.add(new NodoAtributo(r.getString(3), r.getString(5), r.getString(6), r.getBoo
66     }
67
68     this.carregarSchemasRestantes(btree);
69 } catch (SQLException ex) {
70     Catch.captureException(CatchObject.get(ex, "Erro ao resgatar informações dos bancos de
71

```

3.3 OPERACIONALIDADE DA FERRAMENTA

Nesta subseção serão apresentadas as principais telas do sistema e suas funcionalidades.

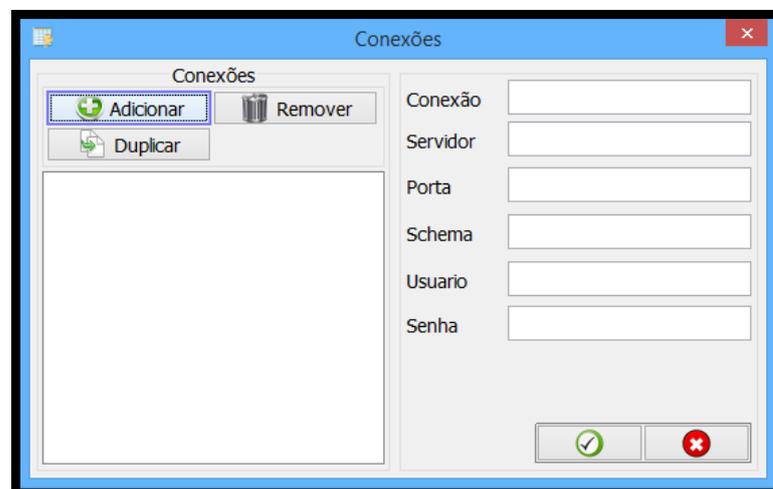
Na Figura 18 é apresentado a tela de autenticação do Naveban. Por *default*, caso o usuário não valores para determinados campos, alguns já contém valores pré-configurados para tentar estabelecer uma conexão com o banco de dados.

Figura 18 - Tela de autenticação do Naveban



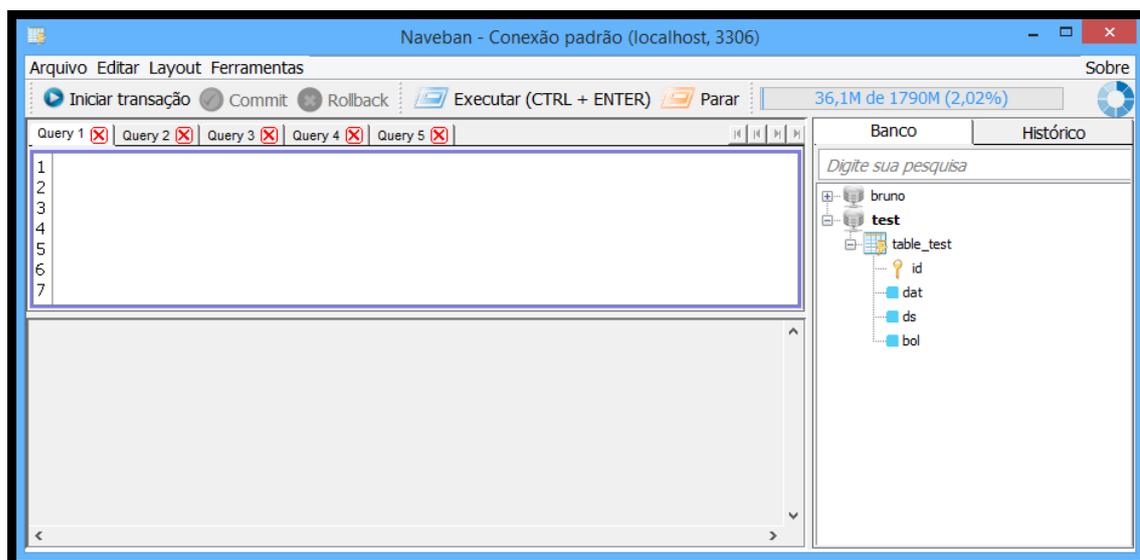
Na Figura 19 é apresentada a tela de cadastro de conexão, por meio desta, o usuário pode cadastrar, remover e editar as informações das conexões que ele possui cadastradas na aplicação.

Figura 19 - Tela de gerenciamento de conexões



Na Figura 20 é apresentada a tela principal da aplicação.

Figura 20 - Tela principal do Naveban



A Figura 21 apresenta os botões da tela inicial responsáveis pelo controle de transação, onde é possível o usuário iniciar uma transação, executar as sentenças SQL e então, persistir as alterações (realizar o *commit*) ou então desfazer as alterações (*rollback*) conforme necessidade.

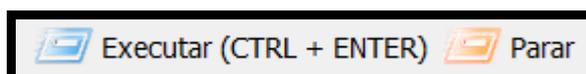
Figura 21 - Botões de controle de transação



Quando o usuário inicia uma transação, a habilitação dos botões é invertida, então, os botões de “Commit” e “Rollback” ficam habilitados para utilização e o botão de “Iniciar transação” é desabilitado.

A Figura 22 apresenta o botão que executa as sentenças SQL e o botão que finaliza a execução da mesma caso o usuário necessite finalizar a execução da sentença.

Figura 22 - Botões de execução de sentença



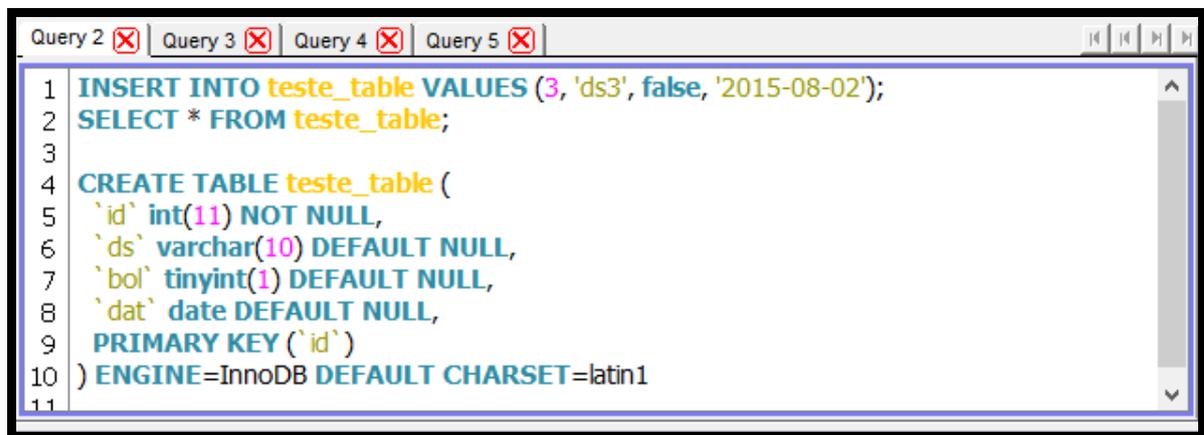
A Figura 23 apresenta o componente que mostra o quanto está sendo consumido de memória pela aplicação e quanto ela tem reservada para utilização. Entre parênteses é disponibilizado para o usuário o percentual de memória utilizada.

Figura 23 - Visualização do consumo de memória

36,1M de 1790M (2,02%)

Na Figura 24 é apresentada a área de texto onde o usuário poderá digitar suas sentenças para serem executadas. Para digitar mais de um comando por área de texto, o usuário deverá por um ponto-e-vírgula (;) no fim da sentença e uma quebra de linha (*Enter*) entre as sentenças digitadas, atendendo ao objetivo B proposto neste trabalho.

Figura 24 - Área de texto das sentenças SQL



```

1  INSERT INTO teste_table VALUES (3, 'ds3', false, '2015-08-02');
2  SELECT * FROM teste_table;
3
4  CREATE TABLE teste_table (
5    `id` int(11) NOT NULL,
6    `ds` varchar(10) DEFAULT NULL,
7    `bol` tinyint(1) DEFAULT NULL,
8    `dat` date DEFAULT NULL,
9    PRIMARY KEY (`id`)
10 ) ENGINE=InnoDB DEFAULT CHARSET=latin1
11

```

A aplicação permite criar várias áreas de texto através de um menu disponível na tela principal. Mais à frente será explanado os menus do sistema, porém, é importante destacar neste momento, que o usuário pode criar quantas abas julgar necessário, podendo navegar entre elas através dos botões localizados no canto superior direito, conforme se observa na Figura 24. Para fechar uma *tab* o usuário pode clicar no “X” da aba.

A Figura 25 é apresenta a tabela que mostra o resultado das consultas executadas pelo usuário e também o componente responsável por apresentar as estatísticas e resultados obtidos das sentenças executadas pelos usuários da aplicação.

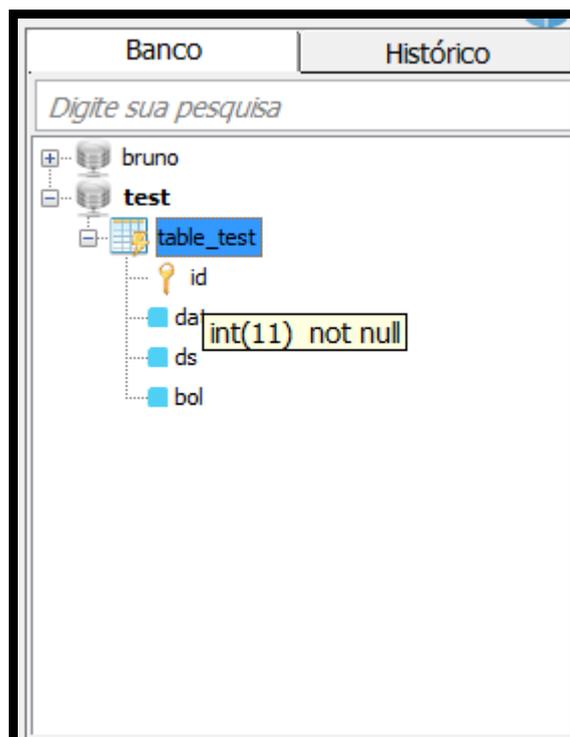
Figura 25 - Tabela de resultados

id	dat	ds	bol
1	2015-06-02	teste 1	1
2	2015-06-02	teste 2	1
3	2015-06-02	teste 3	0

< 3 de registros retornados (00.0073ms) >

Este componente mostra a quantidade de registros retornada em sentenças de consultas e quantidade de registros afetadas em casos de comandos de manipulação do banco de dados. Entre parênteses é apresentado o tempo que a sentença precisou para ser executada, o cálculo é realizado subtraindo o tempo em nano segundos antes de iniciar a execução da sentença, do tempo em que a sentença é retornada do banco.

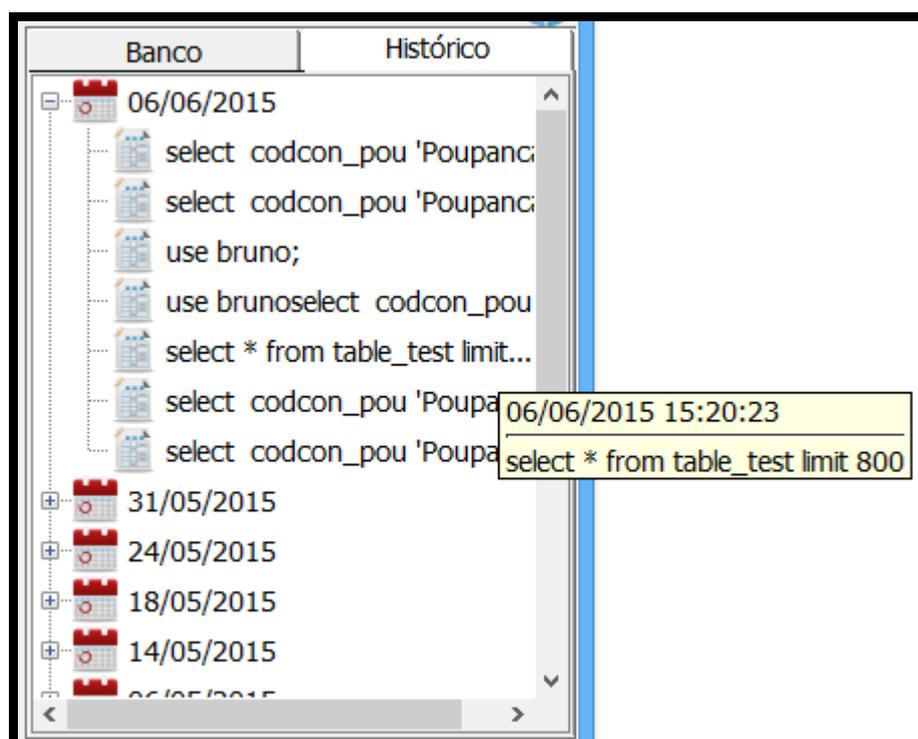
A Figura 26 apresenta a árvore que mostra para o usuário os *schemas*, tabelas e atributos do banco de dados em que a conexão foi estabelecida. Ao passar o *mouse* por cima dos atributos da tabela, é apresentado o tipo do atributo e também informa ao usuário se o atributo permite valores nulos ou não.

Figura 26 - Árvore de *schemas*

Na caixa de texto onde está escrito “Digite sua pesquisa” é possível digitar o nome de uma tabela ou atributo, e a seleção do componente irá mudar para a posição em que a sentença for equivalente, para usar esta funcionalidade. Esta pesquisa é realizada somente nos nós da árvore que estiverem abertos.

Na Figura 27 é apresentado a árvore de histórico de comandos executados pelo usuário.

Figura 27 - Árvore de histórico



Ao posicionar o *mouse* sobre um comando, uma área contendo o texto da sentença SQL, junto com a data e hora de execução da mesma é apresentada.

Na Figura 28 é apresentada a barra de menus da tela principal. O conteúdo de cada menu será apresentado separadamente, para que assim, seja possível explicar melhor as funcionalidades de cada um.

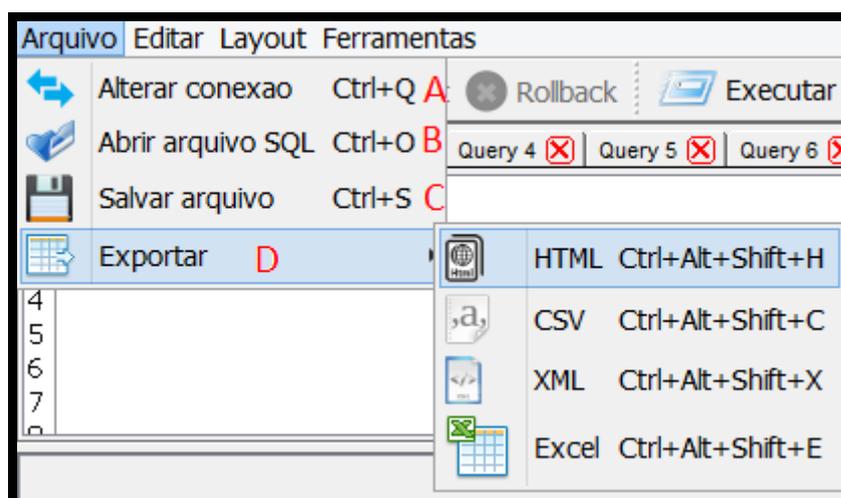
Figura 28 - Barra de menus da tela principal usando o dialeto padrão da ferramenta



Na Figura 29 é apresentado o menu “Arquivo”. Utilizando o menu “Alterar conexão” (A), o usuário pode alterar os dados da conexão, e conectar em outro servidor sem a necessidade de reiniciar a aplicação. Com a opção de “Abrir arquivo SQL” (B) o usuário pode

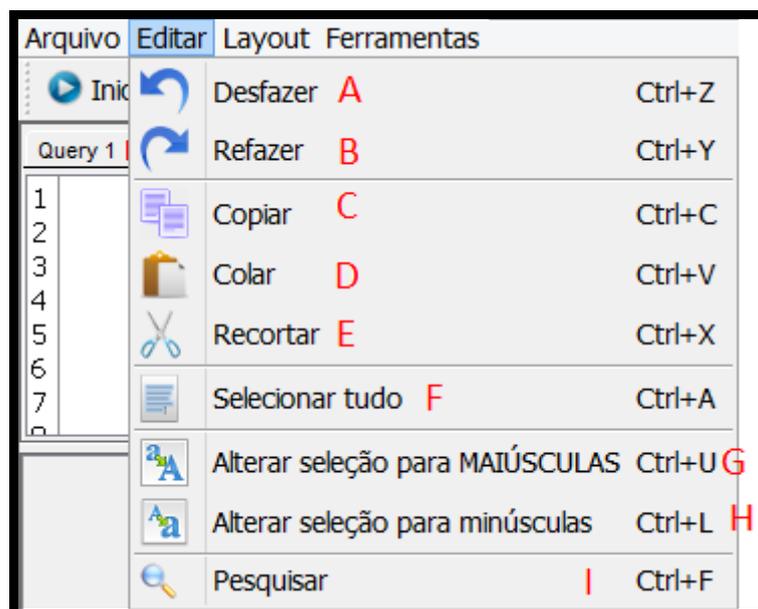
abrir um arquivo de texto em formato SQL, e este será importado para a uma nova área de texto. A opção de “Salvar arquivo” permite o usuário salvar o texto da aba selecionada em um arquivo no formato SQL. O menu “Exportar” (D), permite que o usuário exporte o resultado do último comando executado para os formatos HTML, CSV, XML e XLS, atendendo o requisito funcional 4. No apêndice B é apresentado o resultado obtido com a exportação de arquivos.

Figura 29 - Menu arquivo



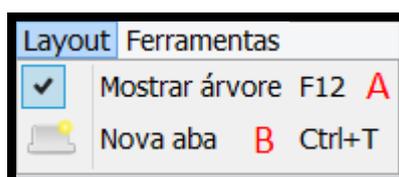
A Figura 30 apresenta o menu “Editar”. Este menu apresenta ações de edição de texto que são executadas na área de texto principal da aplicação. Com opções agrupadas por funcionalidades, no primeiro grupo é possível ver que estão dispostos os menus “Desfazer” (A) e “Refazer” (B) edições de texto. No segundo grupo tem-se os menus de manipulação da área de transferência do Sistema Operacional (SO), com os menus “Copiar” (C), “Colar” (D) e “Recortar” (E). No terceiro grupo existe apenas uma opção, que seleciona todo o texto do campo de texto principal, acessível através da opção “Selecionar tudo” (F). O quarto grupo permite que o usuário selecione um texto e alterne entre letras maiúsculas e minúsculas usando os menus “Alternar seleção para MAIÚSCULAS” (G) e “Alterar seleção para minúsculas” (H). No último grupo está disponível o menu “Pesquisar” (I), este menu abre uma janela para que o usuário possa digitar uma sentença a ser pesquisada.

Figura 30 - Menu Editar



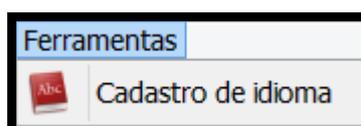
A Figura 31 apresenta o menu “Layout”. Este menu contém a opção “Mostrar árvore” (A) que permite o usuário mostrar/ocultar às árvores de *schemas* e históricos localizadas à direita da tela principal, e também, o menu “Nova aba” (B) que permite que o usuário crie uma nova aba de texto. A Figura 24 mostra a aplicação contendo 4 abas que foram adicionadas utilizando este recurso.

Figura 31 - Menu Layout



A Figura 32 apresenta o menu “Ferramentas”. Neste o dicionário da aplicação torna-se acessível através do menu “Cadastro de idioma”.

Figura 32 - Menu Ferramentas



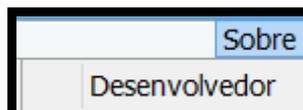
Atendendo ao Requisito Funcional 01, a Figura 33 apresenta a barra de menu principal da ferramenta usando um novo dialeto cadastrado. Na Figura 28 este mesmo menu é apresentado usando o dialeto padrão do sistema.

Figura 33 - Barra de menus da tela principal usando um novo dialeto cadastrado



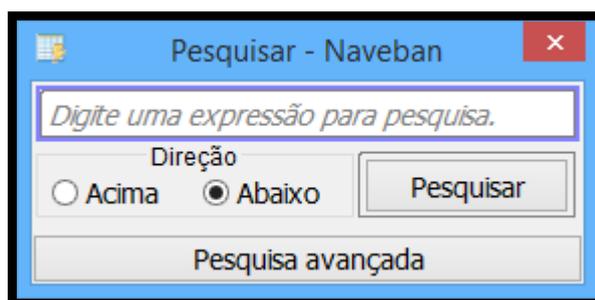
A Figura 34 apresenta o menu “Sobre”. Este, contém o menu “Desenvolvedor”, que mostra informações sobre o desenvolvedor da ferramenta.

Figura 34 - Menu Sobre



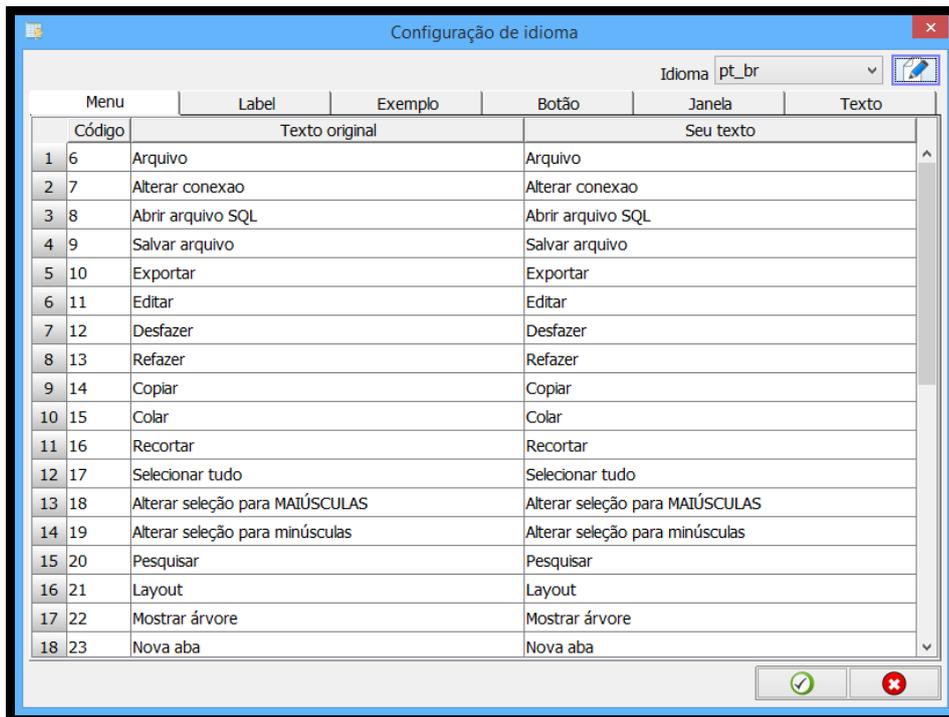
A Figura 35 apresenta a tela de pesquisa de texto que é exibida ao selecionar o menu de “Pesquisar” (Item I, da Figura 31).

Figura 35 - Tela de pesquisa de texto



A Figura 36 apresenta a tela de configuração de idiomas, nesta tela o usuário pode selecionar o idioma que pretende utilizar, e alterar os textos que serão exibidos pela aplicação. Esta tela é acessível usando o menu “Ferramentas” → “Cadastro de idioma” apresentado na Figura 35. Após alterar os textos ou então selecionar outro idioma, é necessário que o usuário reinicie a aplicação para que os textos da interface gráfica sejam carregados.

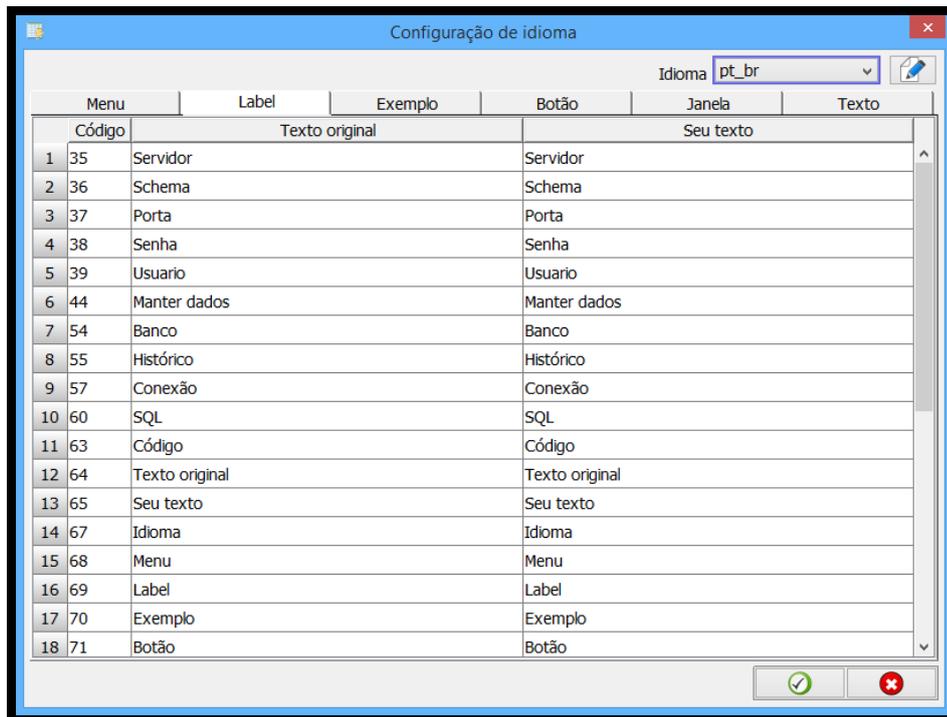
Figura 36 - Cadastro de Menu



Configuração de idioma						
						Idioma
						pt_br
Menu	Label	Exemplo	Botão	Janela	Texto	
	Código	Texto original		Seu texto		
1	6	Arquivo		Arquivo		
2	7	Alterar conexao		Alterar conexao		
3	8	Abrir arquivo SQL		Abrir arquivo SQL		
4	9	Salvar arquivo		Salvar arquivo		
5	10	Exportar		Exportar		
6	11	Editar		Editar		
7	12	Desfazer		Desfazer		
8	13	Refazer		Refazer		
9	14	Copiar		Copiar		
10	15	Colar		Colar		
11	16	Recortar		Recortar		
12	17	Selecionar tudo		Selecionar tudo		
13	18	Alterar seleção para MAIÚSCULAS		Alterar seleção para MAIÚSCULAS		
14	19	Alterar seleção para minúsculas		Alterar seleção para minúsculas		
15	20	Pesquisar		Pesquisar		
16	21	Layout		Layout		
17	22	Mostrar árvore		Mostrar árvore		
18	23	Nova aba		Nova aba		

Esta tela contém todas as expressões possíveis de serem alteradas pelo usuário agrupadas por componentes. A primeira aba apresentada na Figura 36 mostra os textos dos diversos menus cadastrados na aplicação. Na Figura 37 visualiza-se os textos dos *labels* (etiqueta que mostra um texto informativo para o usuário) da aplicação.

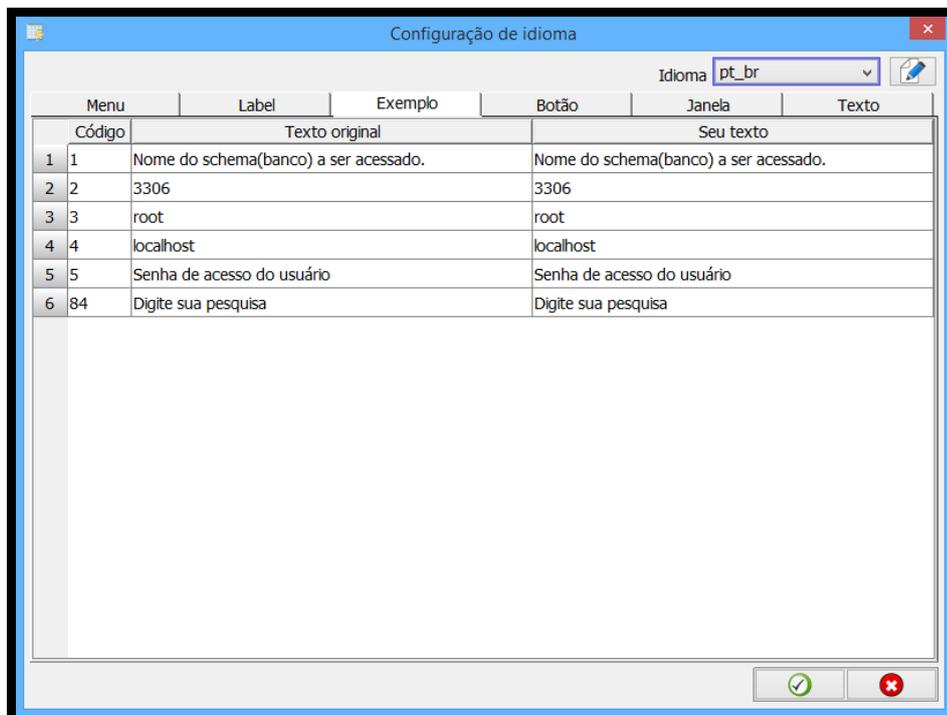
Figura 37 - Cadastro de Label



Menu	Label	Exemplo	Botão	Janela	Texto
Código	Texto original				Seu texto
1	35	Servidor	Servidor		
2	36	Schema	Schema		
3	37	Porta	Porta		
4	38	Senha	Senha		
5	39	Usuario	Usuario		
6	44	Manter dados	Manter dados		
7	54	Banco	Banco		
8	55	Histórico	Histórico		
9	57	Conexão	Conexão		
10	60	SQL	SQL		
11	63	Código	Código		
12	64	Texto original	Texto original		
13	65	Seu texto	Seu texto		
14	67	Idioma	Idioma		
15	68	Menu	Menu		
16	69	Label	Label		
17	70	Exemplo	Exemplo		
18	71	Botão	Botão		

Na Figura 38 estão os textos que são apresentados dentro dos componentes de texto como forma de exemplo para o usuário.

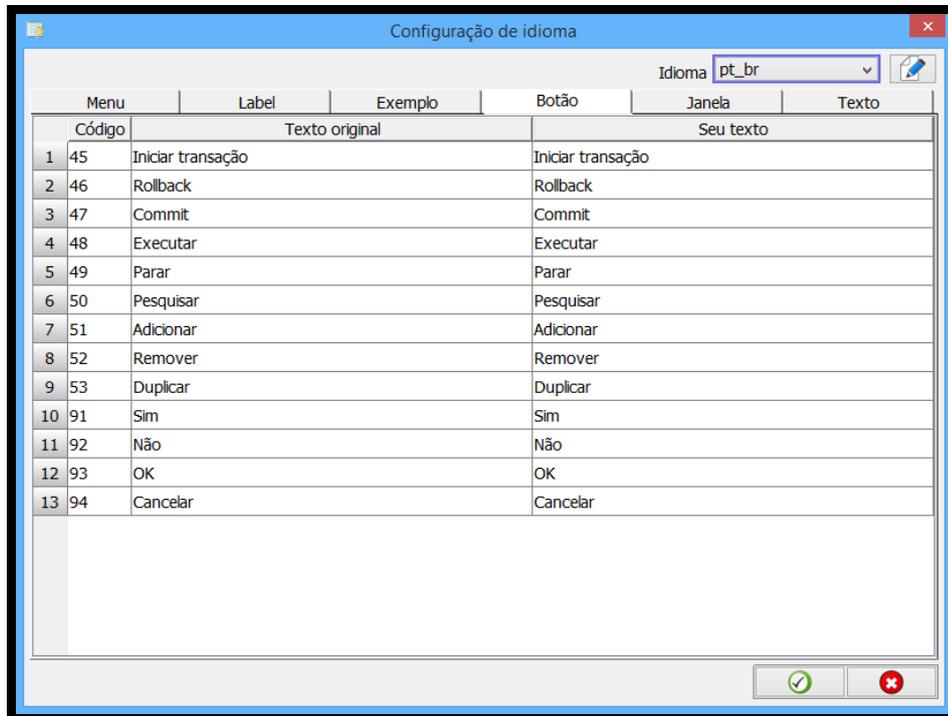
Figura 38 - Cadastro de exemplo



Menu	Label	Exemplo	Botão	Janela	Texto
Código	Texto original				Seu texto
1	1	Nome do schema(banco) a ser acessado.	Nome do schema(banco) a ser acessado.		
2	2	3306	3306		
3	3	root	root		
4	4	localhost	localhost		
5	5	Senha de acesso do usuário	Senha de acesso do usuário		
6	84	Digite sua pesquisa	Digite sua pesquisa		

A Figura 39 apresenta a aba que contém os textos de botões da aplicação.

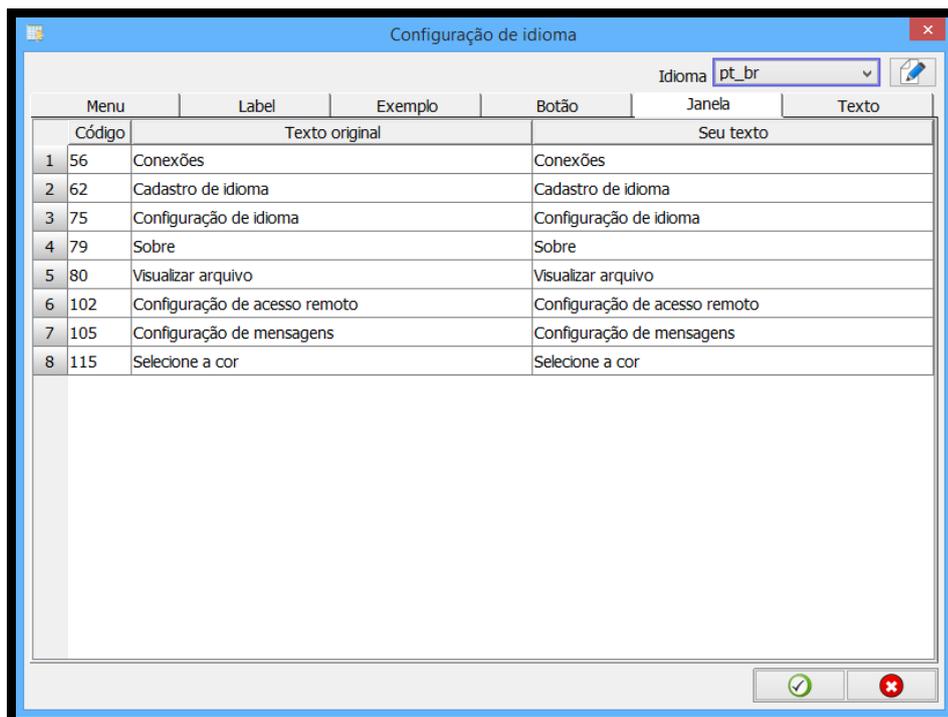
Figura 39 - Cadastro de botões



Menu	Label	Exemplo	Botão	Janela	Texto
Código	Texto original		Seu texto		
1 45	Iniciar transação		Iniciar transação		
2 46	Rollback		Rollback		
3 47	Commit		Commit		
4 48	Executar		Executar		
5 49	Parar		Parar		
6 50	Pesquisar		Pesquisar		
7 51	Adicionar		Adicionar		
8 52	Remover		Remover		
9 53	Duplicar		Duplicar		
10 91	Sim		Sim		
11 92	Não		Não		
12 93	OK		OK		
13 94	Cancelar		Cancelar		

A Figura 40 apresenta a aba que contém os textos de cadastro de janelas. Nesta aba é possível alterar os textos dos títulos das janelas da aplicação.

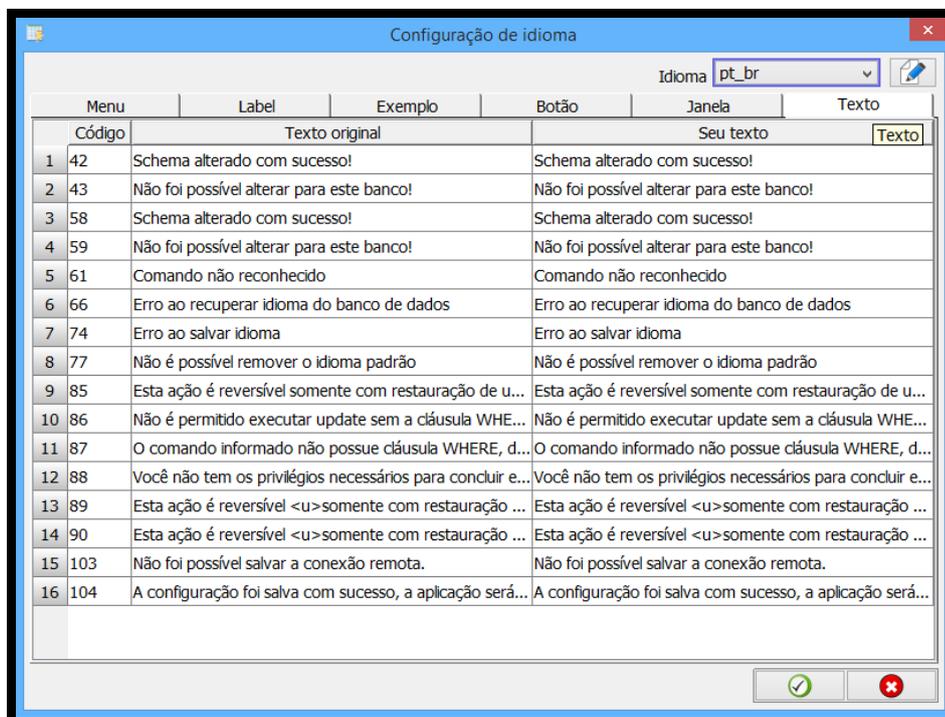
Figura 40 - Cadastro de janelas



Menu	Label	Exemplo	Botão	Janela	Texto
Código	Texto original		Seu texto		
1 56	Conexões		Conexões		
2 62	Cadastro de idioma		Cadastro de idioma		
3 75	Configuração de idioma		Configuração de idioma		
4 79	Sobre		Sobre		
5 80	Visualizar arquivo		Visualizar arquivo		
6 102	Configuração de acesso remoto		Configuração de acesso remoto		
7 105	Configuração de mensagens		Configuração de mensagens		
8 115	Selecione a cor		Selecione a cor		

A Figura 41 apresenta a aba que contém os textos de mensagens apresentadas pela aplicação. Esses textos podem ser cadastrados com *tags* HTML caso o usuário queira utilizar formatação na mensagem apresentada.

Figura 41 - Cadastro de textos



Menu	Label	Exemplo	Botão	Janela	Texto
1	42	Schema alterado com sucesso!	Schema alterado com sucesso!		
2	43	Não foi possível alterar para este banco!	Não foi possível alterar para este banco!		
3	58	Schema alterado com sucesso!	Schema alterado com sucesso!		
4	59	Não foi possível alterar para este banco!	Não foi possível alterar para este banco!		
5	61	Comando não reconhecido	Comando não reconhecido		
6	66	Erro ao recuperar idioma do banco de dados	Erro ao recuperar idioma do banco de dados		
7	74	Erro ao salvar idioma	Erro ao salvar idioma		
8	77	Não é possível remover o idioma padrão	Não é possível remover o idioma padrão		
9	85	Esta ação é reversível somente com restauração de u...	Esta ação é reversível somente com restauração de u...		
10	86	Não é permitido executar update sem a cláusula WHE...	Não é permitido executar update sem a cláusula WHE...		
11	87	O comando informado não possui cláusula WHERE, d...	O comando informado não possui cláusula WHERE, d...		
12	88	Você não tem os privilégios necessários para concluir e...	Você não tem os privilégios necessários para concluir e...		
13	89	Esta ação é reversível <u>somente com restauração ...	Esta ação é reversível <u>somente com restauração ...		
14	90	Esta ação é reversível <u>somente com restauração ...	Esta ação é reversível <u>somente com restauração ...		
15	103	Não foi possível salvar a conexão remota.	Não foi possível salvar a conexão remota.		
16	104	A configuração foi salva com sucesso, a aplicação será...	A configuração foi salva com sucesso, a aplicação será...		

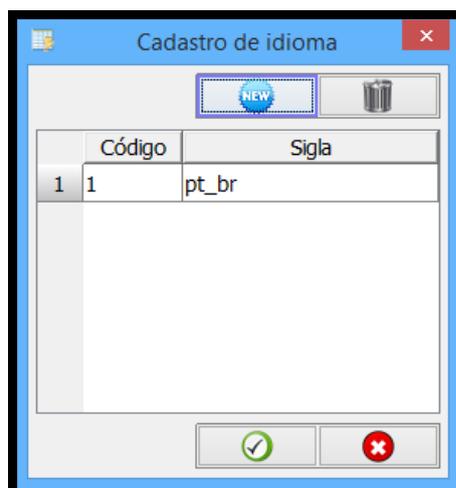
A Figura 42 apresenta um componente do sistema com um texto de exemplo. Esses textos são desenhados nos campos de texto do sistema como forma de exemplo de informação a ser inserida ou descrição do campo.

Figura 42 - Texto de exemplo



Caso o usuário queira cadastrar um novo dialeto deve usar o botão ao lado do *combo* de seleção de idioma. Na Figura 43 é apresentada a tela de cadastro de idiomas, onde o usuário pode criar um novo registro de idioma, ou excluir um existente.

Figura 43 - Tela de cadastro de idioma



3.4 RESULTADOS E DISCUSSÃO

Este trabalho desenvolveu uma ferramenta que possibilitasse a alteração dos textos da interface, para um melhor aproveitamento da aplicação por parte do usuário, que poderia alterar os textos usando um dialeto que lhe fosse mais compreensível. A ferramenta não está limitada ao cadastro de somente um dialeto, permitindo que diferentes usuários usem diferentes dialetos.

Comparando o software desenvolvido neste trabalho com os softwares desenvolvidos por Werner (2004) e Odwazny (2003), as seguintes características podem ser citadas afim de destacar as similaridades entre os mesmos:

- a) ambos tinham o objetivo de permitir a execução de sentenças SQL dos usuários;
- b) ambos continham interface gráfica para interação entre o usuário e o SGBD;
- c) ambos são aplicações desktop.

O software desenvolvido neste trabalho, se difere do software desenvolvido por Werner (2004), pelo fato deste possibilitar o cadastro e edição dos textos exibidos pela interface de interação com o usuário.

O foco do trabalho realizado por Odwazny (2003), foi desenvolver um software para o SGBD PostgreSQL. Assim como no trabalho de Werner, este trabalho se diverge do trabalho apresentado e desenvolvido por Odwazny (2003), por permitir que os textos da interface sejam alterados. O Quadro 3 apresenta uma comparação de algumas características da

ferramenta desenvolvida neste trabalho, e os trabalhos realizados por Werner (2004) e Odwazny (2003).

Quadro 3 - Comparativo com trabalhos correlatos

Característica	Software deste trabalho	Software de Werner (2004)	Software de Odwazny (2003)
Linguagem de desenvolvimento	Java	Delphi	Java
SGBD de utilização	MySQL	Firebird	PostgreSQL
Persistência de informações	Banco de dados (Hsqldb e MySQL)	Arquivos de texto (extensão .ini)	Não
Permite exportar os dados da consulta?	Sim	Não	Sim

4 CONCLUSÕES

O sistema desenvolvido neste trabalho atendeu a todos os objetivos e requisitos mencionados neste trabalho. O software permite criar, remover estruturas de dados, adicionar e revogar permissões, assim como inserir, remover e selecionar registros, atendendo funções básicas para que seja possível criar um banco de dados e inserir informações no mesmo.

Como o sistema persiste suas informações em um arquivo dentro de seu próprio diretório, os usuários, podem carregá-lo em um *pen drive* e utilizar a partir de qualquer equipamento que possua Java 7 ou superior instalado, que suas configurações, preferências e dialeto serão carregados. Caso esteja usando em um ambiente distribuído, o usuário poderá optar por utilizar a aplicação em qualquer microcomputador, que ao realizar o *login*, suas informações serão recuperadas.

Pode-se destacar neste trabalho, a exportação das informações da consulta para arquivos de vários formatos, permitindo que o usuário visualize as informações em outros ambientes, conforme for mais conveniente.

Outro item de destaque é a possibilidade do próprio usuário alterar os textos exibidos pela interface, permitindo sua tradução para outros idiomas ou até mesmo dialetos tornem mais assertivas as opções para usuários leigos, por exemplo.

O software desenvolvido neste trabalho, foi utilizado em sala de aula, em uma turma de Banco de Dados I, ministrada pelo orientador deste trabalho, professor Alexander Roberto Valdameri. Na finalização da aplicação, quando foi disponibilizado para utilização em sala, tinha-se a ideia de realizar uma pesquisa qualitativa da mesma, porém, como a turma não apresentava número significativo, e não era obrigatório o uso do Banco de Dados MySQL em sala, ficou inviável a realização de uma pesquisa qualitativa da ferramenta.

4.1 EXTENSÕES

Quando a ideia de desenvolvimento deste trabalho foi concebida, foram reunidas muitas informações sobre quais funcionalidades uma ferramenta com a finalidade de gerenciar um banco de dados poderia conter, com base nestas ideias a estrutura do banco de dados criado para esta aplicação, já contempla algumas sugestões que serão pontuadas nos próprios parágrafos.

Para trabalhos futuros, é sugerível que seja implantado um controle de segurança para que sejam feitos níveis de acesso para execução de sentenças SQL, ligando o usuário a uma segunda pessoa, que ficará responsável por validar o comando a ser executado, e assim, permitir ou não a execução do mesmo. Criar uma tela de parâmetros para os usuários e cadastrar parâmetros que permitam personalizar o comportamento da aplicação e as permissões dos usuários. Criar telas para criação e edição de estruturas de tabelas por meio de interface gráfica. Incluir na tela principal, uma área de texto para execução de *scripts* contendo uma lista de comandos a serem executados.

REFERÊNCIAS

DATE, C. J. **Introdução a sistemas de bancos de dados**. Tradução Daniel Vieira. Rio de Janeiro: Elsevier, Campus, 2004.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. Tradução Marília Guimarães Pinheiro et all. 4. ed. São Paulo: Pearson Education do Brasil, 2005.

GUPTA, Umang; GIETZ, William. **Guia do programador em SQL**. Tradução Follow-up Traduções e Assessoria de Informática. Rio de Janeiro: Campus, 1990.

HEIDISQL. **Technical help document**. Local: Desconhecido. 2014. Disponível em: <<http://www.heidisql.com/help.php#license>>. Acesso em: 17 Set. 2014.

MANZANO, José A. N. G. **Estudo dirigido de SQL**. São Paulo: Érica, 2002.

MYSQL. **MySQL :: MySQL Product Support EOL Announcements**. Local: Desconhecido. 2009. Disponível em: <<http://www.mysql.com/support/eol-notice.html>>. Acesso em: 08 Set. 2014.

_____. **Manual MySQL Workbench**. Local: Desconhecido. 2014. Disponível em: <<http://downloads.mysql.com/docs/workbench-en.pdf>>. Acesso em: 17 Set. 2014.

ODWAZNY, Paulo F. **Protótipo para administração de SGBD PostgreSQL**. 2003. 56f. Trabalho de Conclusão de Curso (Bacharel em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de gerenciamento de banco de dados**. Tradução Célia Taniwake, João Eduardo Nóbrega Tortello. 3. ed. São Paulo: McGraw-Hill, 2008.

SANCHES, Andre R. **Banco de Dados**. Local: Desconhecido 2005. Disponível em: <<http://www.ime.usp.br/~andrrs/aulas/bd2005-1/aula2.html>>. Acesso em: 04 de Set. 2014.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados**. Tradução: Marília Guimarães Pinheiro, Cláudio César Canhette. 3. ed. São Paulo: Pearson Education, 1999.

STEFANELLI, Eduardo José; **A Importância do Profissional de Comunicação Gráfica na Produção de Material em EAD**. Local: São Paulo. 2015. Disponível em: <http://www.abed.org.br/site/pt/midioteca/textos_ead/627/a_importancia_do_profissional_de_comunicacao_grafica_na_producao_de_material_em_ead_>. Acesso em: 28 Abr. 2015.

WERNER, Carlos E. **Ferramenta de gerenciamento para o banco de dados Firebird**. 2004. 49f. Trabalho de Conclusão de Curso (Bacharel em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

APÊNDICE A – Descrição dos Casos de Uso

Nos quadros a seguir serão detalhados os casos de uso do sistema proposto. O Quadro 4 apresenta o caso de uso “Cadastro de idioma”.

Quadro 4 - Descrição do caso de uso "Cadastro de idioma"

UC01	Cadastro de idioma
Descrição	O usuário acessa a tela de Cadastro de idioma e pode alterar, inserir e excluir um idioma.
Ator	Administrador
Pré-condição	O usuário deverá estar logado no sistema.
Fluxo principal	1. O sistema carrega os idiomas cadastrados. 2. O usuário opta por editar, remover ou adicionar um novo idioma.
Fluxo alternativo (a)	1. Usuário não informar um nome para o idioma. 2. Alerta com mensagem de campo obrigatório.
Fluxo alternativo (b)	1. Usuário tenta excluir o idioma padrão. 2. Emissão de uma caixa de informação com uma mensagem indicando que a exclusão do idioma padrão não é permitida.
Cenário – Edição	1. Sistema mostra os idiomas cadastrados. 2. Usuário seleciona um dos idiomas. 3. Usuário informa o nome desejado. 4. Usuário pressiona o botão de salvar. 5. O sistema fecha a tela de Cadastro de idioma.
Cenário – Inclusão	1. Usuário seleciona a opção de adicionar um novo registro. 2. Usuário informa o nome do novo idioma. 3. Usuário pressiona o botão de salvar. 4. O sistema fecha a tela de cadastro de idioma.
Cenário – Exclusão	1. O usuário seleciona o idioma que deseja excluir. 2. O usuário pressiona o botão de exclusão. 3. O sistema faz a remoção do idioma.
Pós-condição	Usuário inseriu, removeu ou editou um idioma da aplicação.

O Quadro 5 apresenta a descrição do caso de uso “Cadastro de conexão”.

Quadro 5 - Descrição do caso de uso "Cadastro de conexão"

UC02	Cadastro de conexão
Descrição	O usuário acessa a tela de conexões, e edita, cria ou remove conexões.
Ator	Usuário
Pré-condição	O sistema deve acessar a tela de conexões.
Fluxo principal	1. O sistema carrega as conexões cadastradas. 2. O usuário seleciona uma conexão.
Cenário – Edição	1. O usuário seleciona uma conexão cadastrada. 2. O sistema carrega as informações da conexão. 3. O usuário altera as informações. 4. O usuário salva as alterações pressionando o botão de salvar da

	tela. 5. O sistema fecha a tela de conexões.
Cenário – Exclusão	1. O usuário seleciona uma conexão. 2. O usuário pressiona o botão de excluir. 3. O sistema exclui a conexão cadastrada. 4. O usuário salva as alterações pressionando o botão de salvar da tela. 5. O sistema fecha a tela de conexões.
Cenário – Inclusão (a)	1. O usuário pressiona o botão de novo. 2. O sistema limpa os campos para o usuário preencher com os dados do novo registro. 3. O usuário salva as alterações pressionando o botão de salvar da tela. 4. O sistema fecha a tela de conexões.
Cenário – Inclusão (b)	1. O usuário seleciona uma conexão existente. 2. O usuário pressiona o botão de duplicar. 3. O sistema preenche os campos com as informações da conexão selecionada. 4. O usuário altera as informações para a nova conexão. 5. O usuário salva as alterações pressionando o botão de salvar da tela. 6. O sistema fecha a tela de conexões.
Pós-condição	Usuário inseriu, removeu ou editou uma configuração de conexão com o SGBD.

O Quadro 6 apresenta a descrição do caso de uso “Utilizar transação”.

Quadro 6 - Descrição do caso de uso "Utilizar transação"

UC03	Utilizar transação
Descrição	O usuário acessa inicia uma transação no banco de dados.
Ator	Usuário
Pré-condição	O usuário deverá estar logado no sistema.
Fluxo principal	1. O usuário pressiona o botão de iniciar transação.
Pós-condição	O não salva as alterações do usuário automaticamente.

O Quadro 7 apresenta a descrição do caso de uso “Exportar consulta”.

Quadro 7 - Descrição do caso e uso "Exportar consulta"

UC04	Exportar consulta.
Descrição	As informações da consulta do usuário são exportadas.
Ator	Usuário
Pré-condição	O usuário deverá estar logado no sistema.
Fluxo principal	1. O usuário acessa a opção de exportar dados da consulta. 2. O usuário seleciona um diretório e o nome do arquivo para salvar. 3. O usuário seleciona o formato do arquivo que ele deseja

	<p>exportar.</p> <ol style="list-style-type: none"> 4. O usuário confirma a operação. 5. O sistema cria um arquivo no caminho especificado. 6. O sistema mostra uma mensagem para o usuário indicando o sucesso ou fracasso da criação do arquivo.
--	---

O Quadro 8 apresenta a descrição do caso de uso “Criar base de dados”.

Quadro 8 - Descrição do caso de uso "Criar base de dados"

UC05	Criar base de dados
Descrição	A aplicação é configurada para utilizar uma base centralizada para salvar suas informações.
Ator	Usuário
Pré-condição	O usuário deverá estar logado no sistema.
Fluxo principal	<ol style="list-style-type: none"> 1. O usuário executa o sistema pela primeira vez. 2. O sistema cria uma pasta “.isonvb” dentro da pasta de execução da aplicação. 3. O sistema inicia o processo de criação da base de dados local do usuário.
Fluxo alternativo	<ol style="list-style-type: none"> 1. O usuário acessa a tela de informações da conexão através do atalho CTRL + SHIFT + ALT + U. 2. O usuário preenche as informações solicitadas. 3. O usuário pressiona o botão de salvar. 4. O sistema inicia o processo de manutenção e cria a base de dados. 5. O sistema fecha a tela de informações da conexão.
Pós-condição	O sistema passa a utilizar base remota para persistir suas informações.

O Quadro 9 apresenta a descrição do caso de uso “Cadastro/alteração de textos”.

Quadro 9 - Descrição do caso de uso "Cadastro/alteração de textos"

UC06	Cadastro/alteração de textos
Descrição	O usuário acessa a tela de Configuração de idioma, e edita o texto a ser exibido na interface gráfica.
Ator	Usuário
Pré-condição	O usuário deverá estar logado no sistema.
Fluxo principal	<ol style="list-style-type: none"> 1. O sistema carrega os textos do idioma selecionado 2. O usuário seleciona um texto e altera o mesmo. 3. O usuário salva as alterações pressionando o botão de salvar da tela. 4. O sistema fecha a tela de Configuração de idioma.
Pós-condição	Usuário alterou o texto exibido na interface da aplicação.

APÊNDICE B – Exportação das informações

Este Apêndice apresenta exportações realizadas pela aplicação.

A Figura 44 apresenta uma exportação realizada para arquivo HTML.

Figura 44 - Arquivo exportado para formato HTML

The screenshot displays a web application interface with a SQL query and its results. The query is `select * from table_test limit 800`. The results are shown in a table with four columns: `id`, `dat`, `ds`, and `bol`. The data rows are:

id	dat	ds	bol
1	2015-06-02	teste 1	true
2	2015-06-02	teste 2	true
3	2015-06-02	teste 3	false

Below the table, the HTML export is shown in a browser's developer tools. The HTML structure is as follows:

```
<html>
  <head>...</head>
  <body>
    <title>
      Naveban
    </title>
    <hr>
    <pre> select * from table_test limit 800</pre>
    <br>
    <hr>
    <table border="1" width="435">
      <thead>
        <tr>
          <th>id</th>
          <th>dat</th>
          <th>ds</th>
          <th>bol</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>1</td>
          <td>2015-06-02</td>
          <td>teste 1</td>
          <td>true</td>
        </tr>
        <tr>
          <td>2</td>
          <td>2015-06-02</td>
          <td>teste 2</td>
          <td>true</td>
        </tr>
        <tr>
          <td>3</td>
          <td>2015-06-02</td>
          <td>teste 3</td>
          <td>>false</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

A Figura 45 é apresenta uma exportação realizada para arquivo XML.

Figura 45 - Arquivo no formato de XML

```

<NAVEBAN>
  <row>
    <field column="id">1</field>
    <field column="dat">2015-06-02</field>
    <field column="ds">teste 1</field>
    <field column="bol">>true</field>
  </row>
  <row>
    <field column="id">2</field>
    <field column="dat">2015-06-02</field>
    <field column="ds">teste 2</field>
    <field column="bol">>true</field>
  </row>
  <row>
    <field column="id">3</field>
    <field column="dat">2015-06-02</field>
    <field column="ds">teste 3</field>
    <field column="bol">>false</field>
  </row>
</NAVEBAN>

```

A Figura 46 apresenta uma exportação realizada para arquivo XLS.

Figura 46 - Arquivo exportado para arquivo XLS

The screenshot shows the Microsoft Excel interface with the following data in the spreadsheet:

	A	B	C	D
1	id	dat	ds	bol
2		1	2015-06-02	teste 1
3		2	2015-06-02	teste 2
4		3	2015-06-02	teste 3
5				false
6				

The Excel window title is "teste - Excel" and the active sheet is named "NAVEBAN". The status bar at the bottom indicates "PRONTO" and a zoom level of 100%.

A Figura 47 apresenta a estrutura do arquivo XLS.

Figura 47 - Estrutura do arquivo XLS

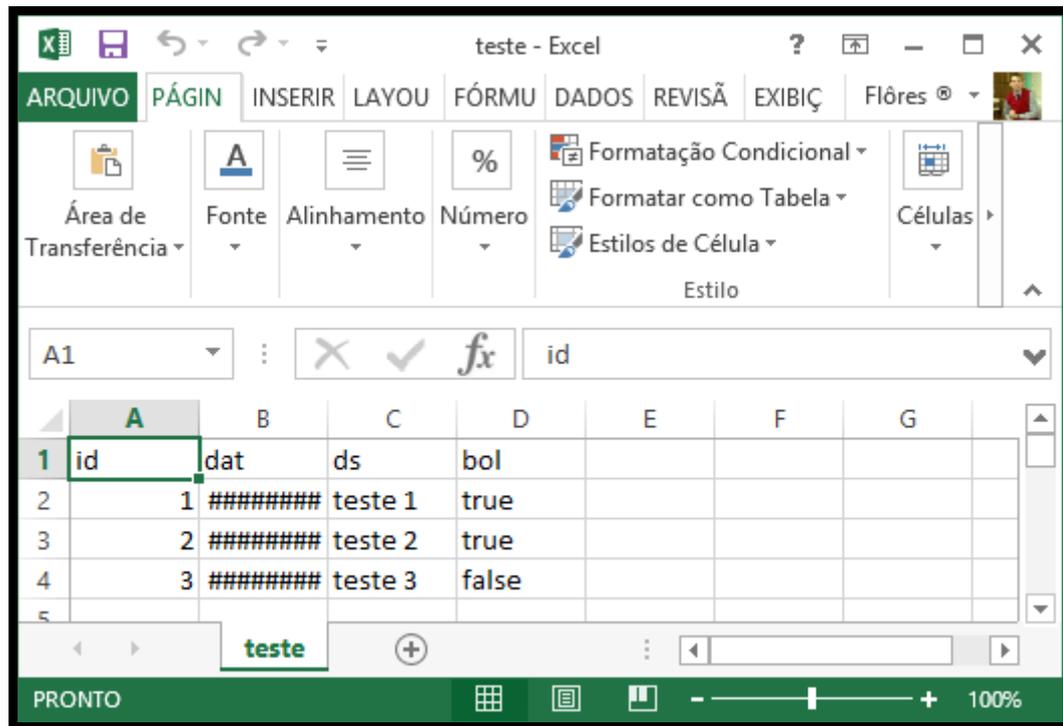
```

<?xml version="1.0" encoding="UTF-8"?>
<Workbook xmlns="urn:schemas-microsoft-com:office:spreadsheet"
  xmlns:x="urn:schemas-microsoft-com:office:excel"
  xmlns:ss="urn:schemas-microsoft-com:office:spreadsheet"
  xmlns:html="http://www.w3.org/TR/REC-html40">
  <Worksheet ss:Name="NAVEBAN">
    <Table>
      <Column ss:Index="1" ss:AutoFitWidth="0" ss:Width="75"/>
      <Column ss:Index="2" ss:AutoFitWidth="0" ss:Width="100"/>
      <Column ss:Index="3" ss:AutoFitWidth="0" ss:Width="200"/>
      <Column ss:Index="4" ss:AutoFitWidth="0" ss:Width="60"/>
      <Row>
        <Cell><Data ss:Type="String">id</Data></Cell>
        <Cell><Data ss:Type="String">dat</Data></Cell>
        <Cell><Data ss:Type="String">ds</Data></Cell>
        <Cell><Data ss:Type="String">bol</Data></Cell>
      </Row>
      <Row>
        <Cell><Data ss:Type="Number">1</Data></Cell>
        <Cell><Data ss:Type="String">2015-06-02</Data></Cell>
        <Cell><Data ss:Type="String">teste 1</Data></Cell>
        <Cell><Data ss:Type="String">>true</Data></Cell>
      </Row>
      <Row>
        <Cell><Data ss:Type="Number">2</Data></Cell>
        <Cell><Data ss:Type="String">2015-06-02</Data></Cell>
        <Cell><Data ss:Type="String">teste 2</Data></Cell>
        <Cell><Data ss:Type="String">>true</Data></Cell>
      </Row>
      <Row>
        <Cell><Data ss:Type="Number">3</Data></Cell>
        <Cell><Data ss:Type="String">2015-06-02</Data></Cell>
        <Cell><Data ss:Type="String">teste 3</Data></Cell>
        <Cell><Data ss:Type="String">>false</Data></Cell>
      </Row>
    </Table>
  </Worksheet>
</Workbook>

```

A Figura 48 apresenta uma exportação realizada para arquivo CSV.

Figura 48 - Arquivo exportado para CSV



A Figura 49 apresenta a estrutura do arquivo CSV.

Figura 49 - Estrutura do arquivo CSV

```
id;dat;ds;bol;
1;2015-06-02;teste 1>true;
2;2015-06-02;teste 2>true;
3;2015-06-02;teste 3>false;
```