

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

**FERRAMENTA PARA SINCRONIZAÇÃO ESTRUTURAL DE**  
**BASES DE DADOS MYSQL APLICADA À BLUEONE**  
**INFORMÁTICA LTDA**

**FERNANDO PEREIRA**

**BLUMENAU**  
**2011**

**2011/1-09**

**FERNANDO PEREIRA**

**FERRAMENTA PARA SINCRONIZAÇÃO ESTRUTURAL DE  
BASES DE DADOS MYSQL APLICADA À BLUEONE  
INFORMÁTICA LTDA**

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. Wilson Pedro Carli, Mestre – Orientador

**BLUMENAU  
2011**

**2011/1-09**

**FERRAMENTA PARA SINCRONIZAÇÃO ESTRUTURAL DE  
BASES DE DADOS MYSQL APLICADA À BLUEONE  
INFORMÁTICA LTDA**

Por

**FERNANDO PEREIRA**

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

Presidente: \_\_\_\_\_  
Prof. Wilson Pedro Carli, Mestre – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Cláudio Ratke, Mestre – FURB

Membro: \_\_\_\_\_  
Prof. Paulo Fernando da Silva, Mestre – FURB

Blumenau, 01 de julho de 2011.

Dedico este trabalho a minha família e a todos os amigos, especialmente aqueles que me ajudaram diretamente na realização deste.

## **AGRADECIMENTOS**

A Deus, pelo seu imenso amor e graça.

À minha família, sempre presente.

À minha namorada, Débora Ferreira, pelo imenso apoio.

Ao meu orientador, Wilson Pedro Carli, por ter acreditado na conclusão deste trabalho.

A força não provém da capacidade física e sim de uma vontade indomável.

Mahatma Gandhi

## RESUMO

Este trabalho apresenta o desenvolvimento de uma ferramenta cuja finalidade é sincronizar a estrutura de uma base de dados MySQL, considerando-se como parâmetro a estrutura de outra determinada base de dados. A ferramenta desenvolvida para o ambiente *Desktop* utiliza-se da linguagem de programação Object Pascal e permite que uma base de dados tenha sua estrutura alterada para ficar idêntica a outra base de dados. Também permite que sejam visualizadas graficamente as diferenças entre as duas bases de dados além de possibilitar que o usuário selecione quais itens devem ser realmente sincronizados.

Palavras-chave: Banco de dados. MySQL. Sincronização de estrutura.

## **ABSTRACT**

This work presents the development of a tool whose purpose is to synchronize the structure of a MySQL database, considering as a parameter the structure from another database. The developed tool for the desktop environment uses the Object Pascal programming language and allows a database to have its structure changed to be identical to another database. It also allows them to be viewed graphically the differences between the two databases beyond enable the user to select which items really should be synchronized.

Key-words: Database, MySQL, Structure synchronization.

## LISTA DE QUADROS

QUADRO 1 – REQUISITOS FUNCIONAIS.....	20
QUADRO 2 – REQUISITOS NÃO FUNCIONAIS.....	20
QUADRO 3 – DESCRIÇÃO DO CASO DE USO “UC01 – PRÉ-VISUALIZAR SINCRONIZAÇÃO”.....	35
QUADRO 4 – DESCRIÇÃO DO CASO DE USO “UC02 – SINCRONIZAR BASE DE DADOS”.....	36
QUADRO 5 – DESCRIÇÃO DO CASO DE USO “UC03 – GERENCIAR CONEXÕES” .....	37

## LISTA DE FIGURAS

<b>FIGURA 1 - EXEMPLO DE BASE DE DADOS.....</b>	<b>14</b>
<b>FIGURA 2 - EXEMPLO DE REGISTROS DE UMA TABELA.....</b>	<b>15</b>
<b>FIGURA 3- DIAGRAMA DE CASOS DE USO.....</b>	<b>21</b>
<b>FIGURA 4 - DIAGRAMA DE CLASSES.....</b>	<b>22</b>
<b>FIGURA 5 – DIAGRAMA DE ATIVIDADES PARA ATENDER O PROCESSO DE SINCRONIZAÇÃO.....</b>	<b>24</b>
<b>FIGURA 6 - FLUXO LÓGICO DO MÉTODO "COMPARAR" .....</b>	<b>26</b>
<b>FIGURA 7 - CÓDIGO FONTE - COMANDO PARA CRIAR TABELA.....</b>	<b>27</b>
<b>FIGURA 8 - CÓDIGO FONTE - MONTAGEM DO COMANDO COM COMENTÁRIOS.....</b>	<b>27</b>
<b>FIGURA 9 – TELA PARA GERENCIAR CONEXÕES.....</b>	<b>28</b>
<b>FIGURA 10 - TELA PARA SELECIONAR AS BASES DE DADOS.....</b>	<b>29</b>
<b>FIGURA 11 - OPÇÕES DE COMPARAÇÃO.....</b>	<b>29</b>
<b>FIGURA 12 - RESULTADO DA COMPARAÇÃO.....</b>	<b>30</b>
<b>FIGURA 13 - SCRIPT PARA SINCRONIZAÇÃO.....</b>	<b>31</b>

## **LISTA DE SIGLAS**

DDL - *Data Definition Language*

GPL – *General Public License*

SGBD – Sistema de Gerenciamento de Banco de Dados

SQL – *Structured Query Language*

WWW ou WEB - *World Wide Web*

XML – *Extensible Markup Language*

# SUMÁRIO

<b><u>1 INTRODUÇÃO.....</u></b>	<b><u>12</u></b>
1.1 OBJETIVOS DO TRABALHO.....	13
1.2 ESTRUTURA DO TRABALHO.....	13
<b><u>2 FUNDAMENTAÇÃO TEÓRICA.....</u></b>	<b><u>14</u></b>
2.1 BASE DE DADOS / BANCO DE DADOS.....	14
2.2 SISTEMA GERENCIADOR DE BANCO DE DADOS.....	15
2.2.1 MySQL.....	15
2.3 STRUCTURED QUERY LANGUAGE (SQL).....	16
2.3.1 Linguagem de Definição de Dados (Data Definition Language – DDL).....	16
2.4 TRABALHOS CORRELATOS.....	17
<b><u>3 DESENVOLVIMENTO.....</u></b>	<b><u>19</u></b>
3.1 LEVANTAMENTO DE INFORMAÇÕES.....	19
3.2 ESPECIFICAÇÃO.....	20
3.3 IMPLEMENTAÇÃO.....	25
3.3.1 Técnicas e ferramentas utilizadas.....	25
3.3.2 Operacionalidade da implementação.....	27
3.3.2.1 Gerenciamento de dados para conexão com as bases de dados.....	28
3.3.2.2 Seleção das bases de dados a serem comparadas.....	28
3.3.2.3 Opções para comparação.....	29
3.3.2.4 Resultado da comparação.....	30
3.3.2.5 Script com os comandos para sincronização.....	30
3.4 RESULTADOS E DISCUSSÃO.....	31
<b><u>4 CONCLUSÕES.....</u></b>	<b><u>32</u></b>
4.1 EXTENSÕES.....	32
<b><u>REFERÊNCIAS BIBLIOGRÁFICAS.....</u></b>	<b><u>34</u></b>
<b><u>APÊNDICE A – DETALHAMENTO DOS CASOS DE USO.....</u></b>	<b><u>35</u></b>

## 1 INTRODUÇÃO

A necessidade de integração de informações distintas, armazenadas em locais diferentes, seja fisicamente ou logicamente, é cada vez mais necessária. As empresas investem em dispositivos e redes de dados para expandirem o volume de seus negócios de maneira segura e rápida. Na mesma proporção, as empresas desenvolvedoras de software devem acompanhar a velocidade de seus clientes e a necessidade de aplicações mais estáveis e seguras, disponibilizando equipamentos e quadro funcional para atender a demanda crescente do mercado.

Desta forma a BlueOne Informática Ltda. (BLUEONE) procura atender esta necessidade de mercado. A mesma é uma empresa com sede própria no município de Blumenau, estado de Santa Catarina, que desenvolve softwares e sistemas internos de gerenciamento de processos baseados na *World Wide Web* (WWW ou WEB). Em sua maioria, os softwares desenvolvidos pela BLUEONE fazem uso de uma ou mais bases de dados cujo sistema gerenciador de banco de dados é o MySQL. Cada base de dados por sua vez possui uma estrutura pré-definida, contudo esta estrutura sofre alterações sob demanda de solicitações dos clientes da BLUEONE. Ou seja, de acordo com a necessidade de alterações nas funcionalidades, ou até mesmo a necessidade de novas funcionalidades dos softwares desenvolvidos, a estrutura é alterada.

Conforme as funcionalidades são adicionadas ou alteradas, os softwares são atualizados nos clientes por uma equipe responsável pelas atualizações. A atualização dos softwares inclui atualização de arquivos, criação de diretórios, inclusões de configurações na base de dados entre outros ajustes necessários. Um dos itens mais importantes durante a atualização dos softwares é a sincronização da estrutura das bases de dados.

A criação ou alteração de uma funcionalidade pode necessitar de uma alteração na estrutura da base de dados. Assim sendo, é necessário, ao atualizar os softwares sincronizar a estrutura das bases de dados, de modo que as estruturas de bases de dados existentes nos clientes sejam as mesmas existentes na BLUEONE.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo geral deste trabalho é desenvolver uma ferramenta que permita otimizar o processo de sincronização de estrutura de bases de dados dos clientes da BLUEONE.

Os objetivos específicos do trabalho são:

- a) identificar as informações que compõem a estrutura de uma base de dados;
- b) comparar e identificar quais informações estão divergentes entre duas estruturas de bases de dados;
- c) gerar uma seqüência de comandos (também conhecida como um *Script*) *Structured Query Language* (SQL) que permita sincronizar duas estruturas de bases de dados, onde a segunda estrutura deve ser alterada de modo a ficar idêntica à primeira estrutura.

## 1.2 ESTRUTURA DO TRABALHO

Este trabalho está estruturado em quatro capítulos.

No primeiro capítulo é contextualizado a introdução, a justificativa e os objetivos do trabalho.

No segundo capítulo é disponibilizada a fundamentação teórica necessária para um entendimento sobre banco de dados, sistema gerenciador de banco de dados e SQL, além de apresentar os trabalhos correlatos.

O terceiro capítulo tem como foco o desenvolvimento da ferramenta para sincronização estrutural de bases de dados MySQL, descrevendo os requisitos principais do problema assim como sua especificação e implementação, demonstrando a sua operacionalidade, apresentando os resultados e discussões.

O quarto capítulo apresenta as conclusões e sugestões para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda assuntos relevantes ao objetivo deste trabalho tais como a definição de base e banco de dados, sistema gerenciador de banco de dados, *Structured Query Language* (SQL), o processo atual e os trabalhos correlatos.

### 2.1 BASE DE DADOS / BANCO DE DADOS

Para Mecenas e Oliveira (2005), uma base de dados é um conjunto de dados, uma coleção de informações relacionadas entre si, referentes a um mesmo assunto. Organizadas com o propósito de servir de base para que o usuário as recupere, tire conclusões e tome decisões. Os autores ainda acrescentam que não se pode conceituar uma disposição desordenada de dados, ou seja, de arquivos contendo dados gerados aleatoriamente, como um banco de dados.

Cada base de dados é composta por uma ou mais tabelas. As tabelas possuem campos que determinam as informações que podem ser salvas na tabela. A figura 1 mostra um exemplo de base de dados.

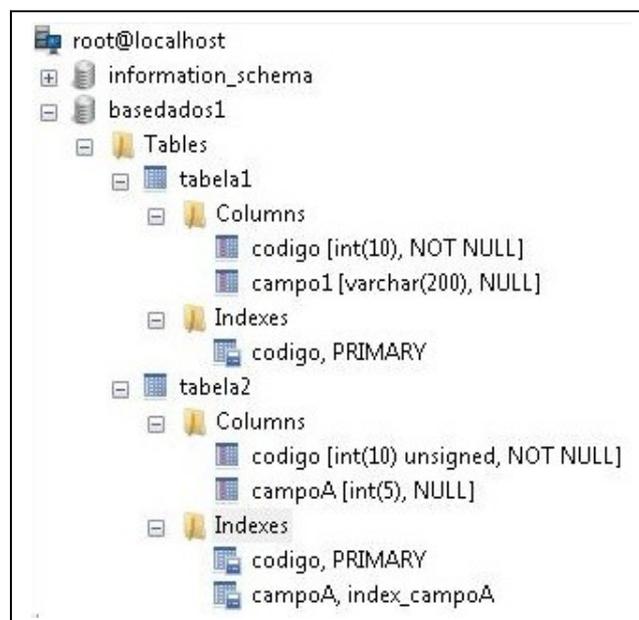


Figura 1- Exemplo de base de dados

Em cada tabela podemos ter vários registros (linhas) conforme mostra a figura 2.

	codigo	campol
<input type="checkbox"/>	1	teste1
<input type="checkbox"/>	2	teste2
<input type="checkbox"/>	3	teste3
<input type="checkbox"/>	4	teste4
<input type="checkbox"/>	5	teste5
<input type="checkbox"/>	6	teste6
<input type="checkbox"/>	7	teste7
<input type="checkbox"/>	8	teste8

Figura 2 - Exemplo de registros de uma tabela

Neste trabalho, considerar-se-á base de dados como sendo um conjunto de dados relacionados entre si e banco de dados um conjunto de bases.

## 2.2 SISTEMA GERENCIADOR DE BANCO DE DADOS

Sistema de Gerenciamento de Banco de Dados (SGBD) corresponde a um conjunto de programas para administrar o Banco de Dados e que desta forma os programas de aplicação não têm acesso aos dados diretamente. Sendo que todas as requisições de leitura ou escrita dos dados são realizadas através do SGBD.

“Um Sistema de Gerenciamento de Banco de Dados, ou SGBD, é um software projetado para auxiliar a manutenção e utilização de vastos conjuntos de dados.” (RAMAKRISHNAN; GEHRKE, 2008).

Abaixo estão listados os SGBDs apresentados por Suehring (2002):

- a) Oracle;
- b) Microsoft SQL Server;
- c) PostgreSQL;
- d) MySQL.

### 2.2.1 MySQL

De acordo com Mecnas e Oliveira (2005), MySQL é um sistema gerenciador de banco de dados cliente/servidor constituído de um servidor SQL. Para Suehring (2002),

MySQL oferece o melhor cenário entre outros SGBDs por estar disponível em várias plataformas, dentre elas, Linux, Windows, Solaris, FreeBSD, MacOS X entre outros. Por ser estável e possuir uma excelente documentação contendo material de referência. Além de sua licença estar sob a GNU General Public License (GPL).

## 2.3 STRUCTURED QUERY LANGUAGE (SQL)

A linguagem SQL foi criada na década de 70 e acabou se tornando a linguagem padrão para os bancos de dados relacionais. É constituída de um conjunto de instruções que possuem a capacidade de manipular dados, definirem estruturas de dados e especificar restrições de segurança e integridade (MECENAS; OLIVEIRA, 2005, pags. 28).

Segundo Furtado (1999), dentre as vantagens da padronização SQL, estão:

- a) custos reduzidos no treinamento, visto que os profissionais podem trocar de ambientes sem necessitar novo treinamento;
- b) portabilidade da aplicação, possibilitando que os softwares sejam executados sem alteração em ambientes diferentes de hardware e software;
- c) comunicação através do sistema, diferentes sistemas de gerenciamento de banco de dados poderiam ser capazes de se comunicar com mais.

Em contrapartida, as desvantagens são as seguintes:

- a) falta de expressões e/ou funções;
- b) restrições a campos longos;
- c) repressão da criatividade, por prescrever algumas soluções alternativas para determinados problemas, mas que poderiam ser mais bem solucionadas pelos desenvolvedores.

### 2.3.1 Linguagem de Definição de Dados (*Data Definition Language* – DDL)

A linguagem DDL é um conjunto de declarações SQL que fornecem meios para a definição e a modificação das estruturas de uma base de dados. Estas declarações, ou

comandos, são utilizadas para criar, alterar ou eliminar objetos da base de dados como tabelas, campos e índices (MECENAS; OLIVEIRA, 2005, p. 29).

Entre os principais comandos DDL pode-se citar:

- a) *create table*, responsável pela definição da estrutura, suas restrições de integridade e por criar uma tabela vazia;
- b) *alter table*, utilizado para modificar a definição de uma tabela;
- c) *drop table*, para remover uma tabela existente com todos os seus registros.

Com seus respectivos exemplos:

- a) 

```
CREATE TABLE tabela1 (
    codigo INT(10) NOT NULL AUTO_INCREMENT,
    PRIMARY KEY(codigo)
);
```
- b) 

```
ALTER TABLE tabela1 (
    ADD COLUMN campo1 VARCHAR(200) NULL AFTER codigo
);
```
- c) 

```
DROP TABLE tabela1;
```

## 2.4 TRABALHOS CORRELATOS

Durante o desenvolvimento deste trabalho, foram analisados três dos vários trabalhos encontrados. O primeiro trabalho pesquisado foi uma ferramenta de gerenciamento para o banco de dados Firebird. De acordo com Werner (2004) o objetivo do trabalho foi desenvolver uma ferramenta que permitisse gerenciar o SGBD Firebird. Um dos objetivos secundários é a manutenção da estrutura da base de dados.

Um aplicativo WEB para definição do modelo lógico no projeto de banco de dados relacional foi o segundo trabalho pesquisado. O mesmo tem por objetivo criar um aplicativo WEB para definição dos modelos lógicos nos projetos de banco de dados relacional. Como objetivo secundário, permitir que através do aplicativo seja possível definir a estrutura de um banco de dados relacional (BACHMANN, 2007).

O terceiro trabalho é uma ferramenta para replicação de dados no SGBD POSTGREESQL, cujo objetivo é elaborar uma ferramenta para replicação de dados, sendo que um de seus objetivos específicos é a atualização remota de dados (IMHOF, 2007).

Além dos três trabalhos, foi analisada também a ferramenta SQLyog Enterprise – MySQL GUI v8.02 RC que traz entre suas funcionalidades, a de sincronizar estruturas de bases de dados MySQL. Nesta ferramenta, após se estabelecer conexão com as duas bases de dados a serem comparadas, é possível compará-las e selecionar quais itens serão

sincronizados. Em seguida, a ferramenta cria o *script* para sincronização e permite ao usuário selecionar entre executar o *script* gerado ou salvá-lo em arquivo de texto (WEBYOG INC, 2011).

### 3 DESENVOLVIMENTO

Neste capítulo estão descritas as particularidades técnicas da ferramenta desenvolvida, tais como a descrição do mesmo e a apresentação dos requisitos funcionais, requisitos não funcionais, principais diagramas de caso de uso com suas respectivas descrições e principais softwares a serem utilizados.

#### 3.1 LEVANTAMENTO DE INFORMAÇÕES

Toda vez que o software de um cliente for atualizado, sua respectiva base de dados deve ter a estrutura sincronizada com da base de dados mantida no servidor da BLUEONE. Um dos itens mais importantes durante a atualização dos softwares desenvolvidos pela BLUEONE é a sincronização da estrutura das bases de dados.

A sincronização tem como objetivo tornar a estrutura das bases de dados existentes nos servidores dos clientes idêntica à estrutura existente no servidor da BLUEONE, que por sua vez sofre modificações sob demanda para atender as necessidades de seus clientes.

O processo de sincronização da estrutura da base de dados é o primeiro item a ser executado. E compreende as seguintes etapas:

- a) efetuar cópia de segurança da base de dados atual no cliente;
- b) obter a listagem completa das tabelas (da base do cliente) com seus campos, sendo que cada campo possui ainda um tipo e um valor padrão;
- c) obter a listagem completa das tabelas (da base existente na BLUEONE) com seus campos, sendo que cada campo possui ainda um tipo e um valor padrão;
- d) comparar as duas listagens, identificando as diferenças como ausência de campos em determinadas tabelas, definição incorreta de valores padrão, ausência de índices entre outros aspectos;
- e) elaborar uma seqüência de comandos SQL que, sendo executada no SGBD do cliente, efetue as alterações necessárias para que a estrutura seja sincronizada;
- f) obter novamente as duas listagens e compará-las para verificar se a sincronização foi efetuada com sucesso. Caso contrário, restaura-se a cópia de segurança e se repete o procedimento.

Como este processo é manual e as diferenças são identificadas visualmente por um técnico, existem duas possibilidades causadoras de falhas ou erros:

- a) identificação incompleta das diferenças das estruturas, que levará à falta de alguns itens a serem sincronizados, que por sua vez causarão erros impedindo o bom funcionamento do software em questão;
- b) sincronização incompleta da estrutura, mesmo que todas as diferenças sejam identificadas, poder-se-á verificar a falha na criação dos comandos, onde, alguns itens podem, eventualmente, não serem corretamente sincronizados.

Este processo demanda recursos como tempo, mão de obra e conhecimento técnico por parte do responsável a criar e executar os comandos SQL para a sincronização. O Quadro 1 e Quadro 2 apresentam, respectivamente, os requisitos funcionais (e sua rastreabilidade) e os requisitos não funcionais.

<b>Requisitos Funcionais</b>	<b>Caso de Uso</b>
RF01. A ferramenta deverá permitir ao usuário pré-visualizar as alterações que serão realizadas pela sincronização.	UC01
RF02. A ferramenta deverá permitir a sincronização da estrutura de duas bases do mesmo banco de dados ou de banco de dados distintos, de caminho único, origem (base 1) para destino (base 2).	UC02
RF03. A ferramenta deverá permitir gerar um arquivo de texto com os comandos DDL necessários para efetuar a sincronização.	UC03
RF04. A ferramenta deverá permitir ao usuário selecionar quais itens deverão ser sincronizados ou ignorados.	UC04
RF05. A ferramenta deverá permitir ao usuário gerenciar os dados de conexão de bancos de dados.	UC05

Quadro 1 – Requisitos funcionais

<b>Requisitos Não funcionais</b>
RNF01. A ferramenta deverá ser desenvolvida para a plataforma Windows.
RNF02. A ferramenta deverá poder sincronizar banco de dados MySQL, compatível com a versão 5.5.1.
RNF03. A ferramenta não terá controle de acesso ou perfis de usuários.

Quadro 2 – Requisitos não funcionais

### 3.2 ESPECIFICAÇÃO

Nesta seção estão descritos os diagramas elaborados para o desenvolvimento da ferramenta, tais como diagrama de casos de uso, diagrama de atividades e diagrama de

classes. A figura 2 apresenta o diagrama de casos de uso, sendo que o detalhamento dos mesmos está descrito no Apêndice A.

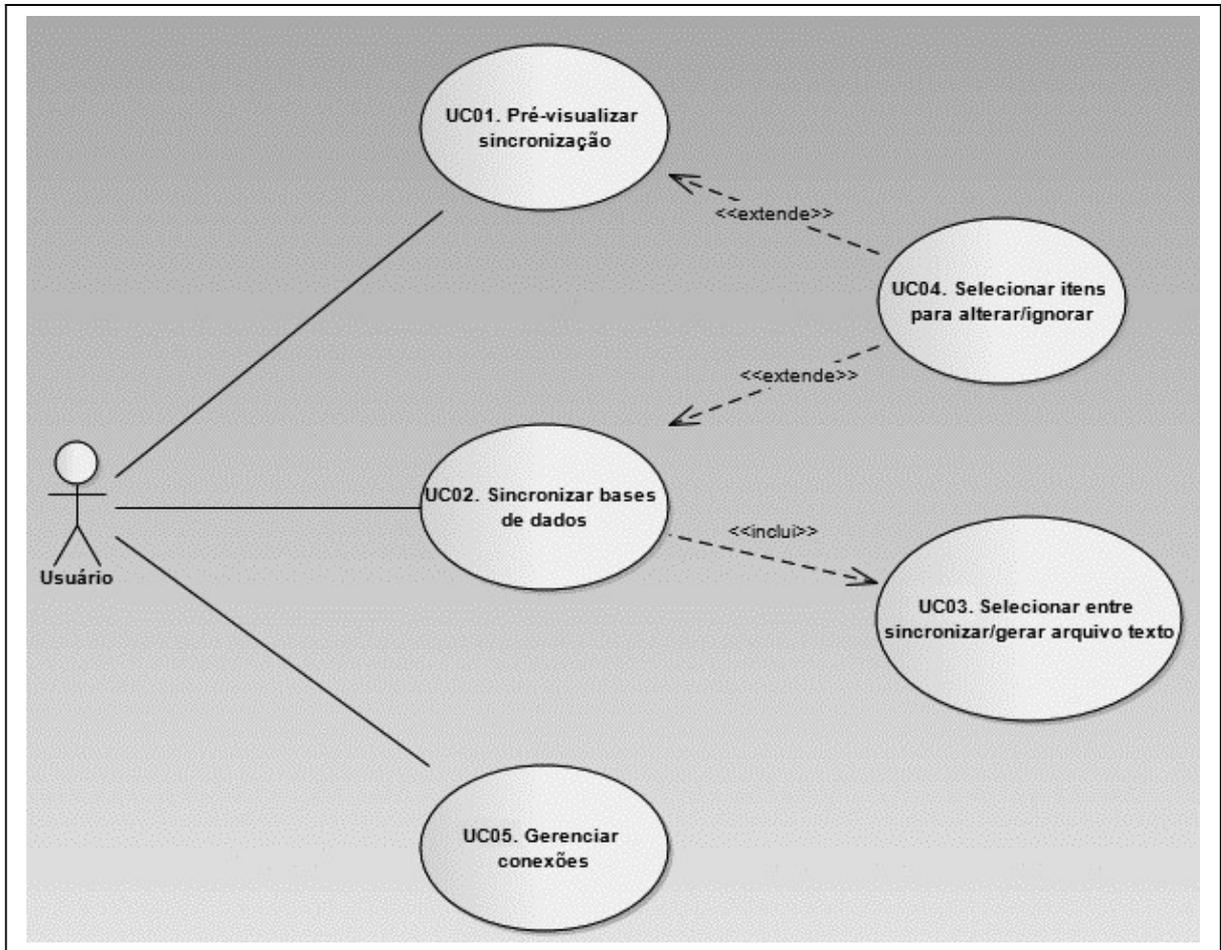


Figura 3- Diagrama de casos de uso

A figura 4 apresenta o diagrama de classes elaborado para o desenvolvimento da ferramenta.

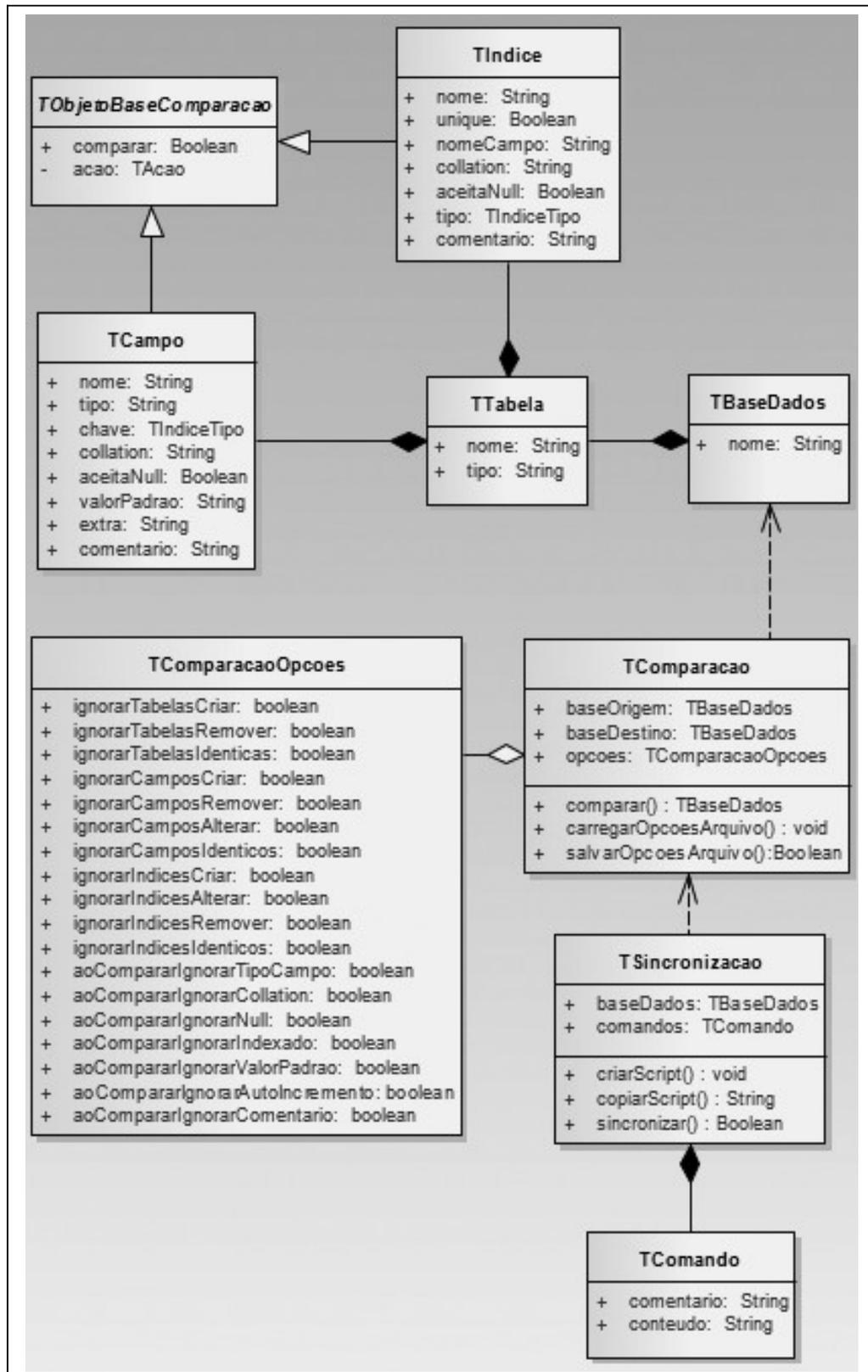


Figura 4 - Diagrama de classes

As principais classes são:

- a) TBaseDados – responsável pela representação da estrutura completa de uma base de dados;
- b) TComparacao – responsável pela comparação entre duas bases de dados. Possui dois atributos principais baseOrigem e baseDestino, que são instâncias de TBaseDados, e um método cujo retorno é uma instância de TBaseDados com a estrutura da base de dados de destino sincronizada. Os atributos baseOrigem e baseDestino representam, respectivamente, a base de dados com a estrutura desejada, e a base de dados que será alterada para ficar com a mesma estrutura da baseOrigem;
- c) TSinconizacao – responsável pela criação dos comandos com base no atributo baseDados, cuja instância é obtida através do método comparar da classe TComparacao. Possui o método que retorna em SQL os comandos para sincronizar a estrutura destino.

O diagrama de atividades descreve o fluxo do processo com o auxílio da ferramenta. A figura 5 mostra o diagrama de atividades para atender o processo de sincronização.

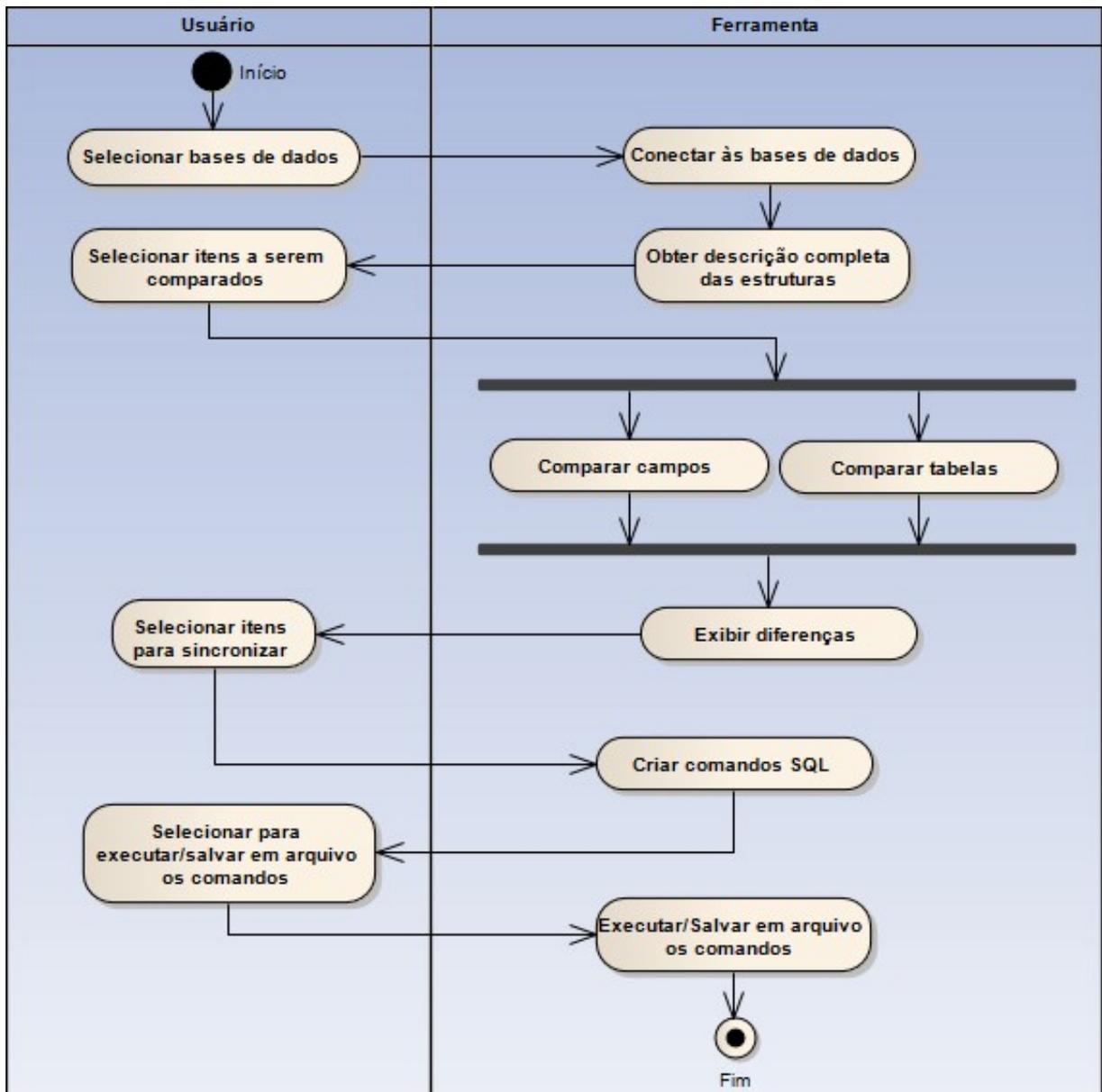


Figura 5 – Diagrama de atividades para atender o processo de sincronização

Ao iniciar a ferramenta o usuário deverá selecionar as bases de dados a terem suas estruturas comparadas. Após selecionar as bases, a ferramenta irá se conectar às mesmas e obterá a descrição completa das estruturas. Neste momento serão apresentadas ao usuário as duas estruturas obtidas permitindo-o selecionar quais itens serão comparados. Em seguida a ferramenta irá criar os comandos SQL e o usuário poderá optar por executá-los ou salvá-los em arquivo SQL. Por fim, a ferramenta irá executar ou salvar em arquivo os comandos DDL.

### 3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas na implementação e na operacionalidade da ferramenta.

#### 3.3.1 Técnicas e ferramentas utilizadas

A ferramenta foi desenvolvida em Object Pascal e foi adotado o ambiente de programação Borland Delphi 7. O desenvolvimento foi baseado em orientação a objetos e assim sendo, para a implementação da ferramenta.

Para efetuar a comparação entre duas bases de dados, primeiro, as informações estruturais são obtidas e mapeadas para os atributos das classes responsáveis pela representação estrutural (TObjetoBaseComparacao, TIndice, TCampo, TTabela e TBaseDados). Isto é feito tanto para a base de dados denominada origem como para a denominada destino (que terá sua estrutura sincronizada).

Depois, tendo-se as duas estruturas mapeadas será possível compará-las. A comparação é feita com auxílio dos atributos e métodos das classes TComparacao, TComparacaoOpcoes, TSincronizacao e TComando. A figura 6 mostra o fluxo lógico do método “comparar” da classe TComparacao que é executado para cada tabela existente na base de dados origem, e através do qual, obtém-se um objeto TBaseDados que representa a estrutura da base de dados destino alterada, sincronizada.

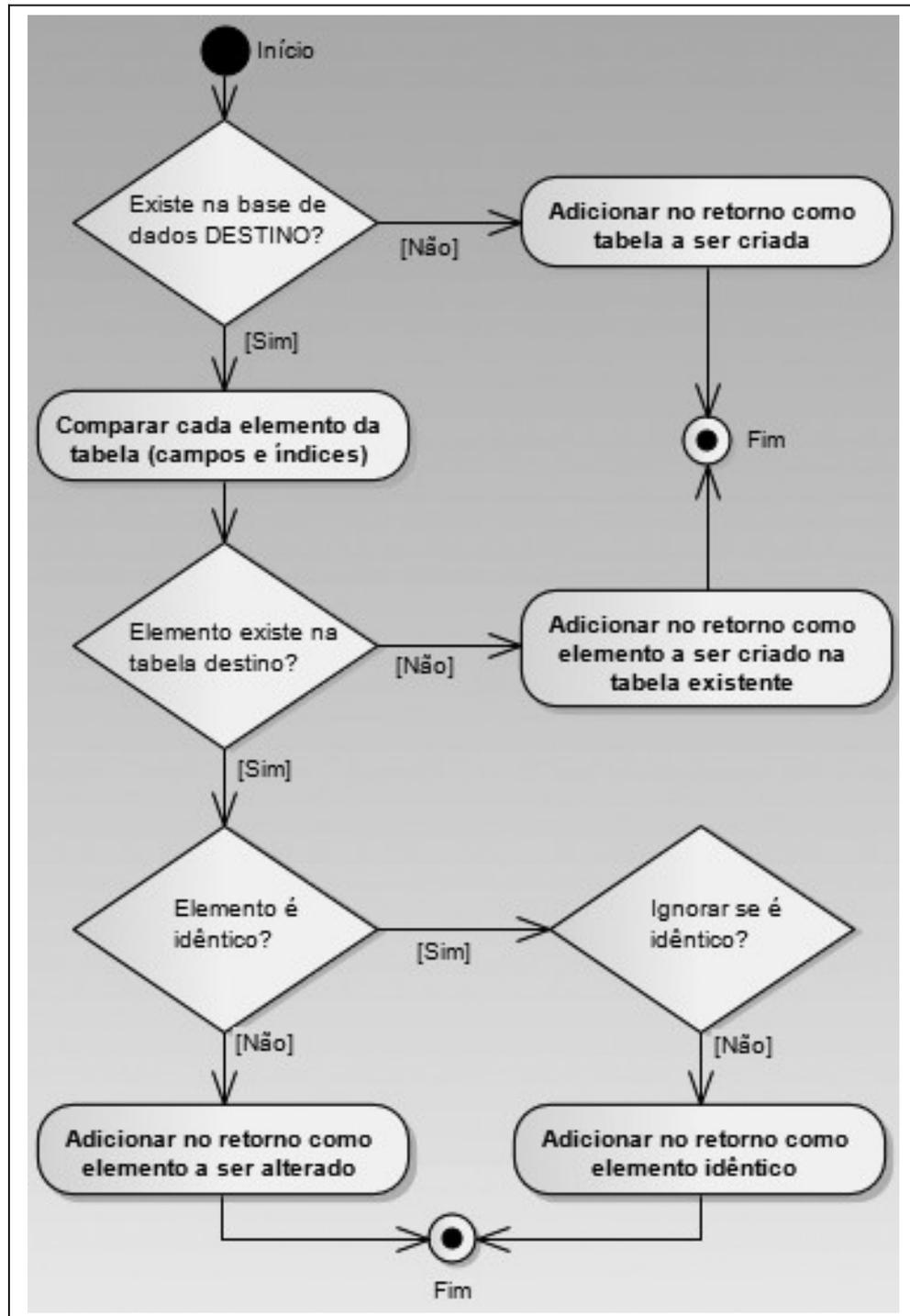


Figura 6 - Fluxo lógico do método "comparar"

A partir deste objeto instanciado e devidamente carregado com as informações é possível criar o *script* SQL. A figura 7 e figura 8 demonstram trechos do código responsáveis pela criação do mesmo.

```

var
  xComandoCampos, xComandoIndices: String;
begin
  case pTabela.acao of

    aCriar      : begin
      xComandoCampos := criarComandoCampos(False, pTabela.campos);
      xComandoIndices := criarComandoIndices(False, pTabela.indices);
      if ((xComandoCampos <> '') and (xComandoIndices <> '')) then
        xComandoCampos := xComandoCampos + ',';

      Result := '';
      if ((xComandoCampos <> '') or (xComandoIndices <> '')) then
      begin
        Result := concatenar(['CREATE TABLE ', pTabela.nome, ' (', '']);

        if (xComandoCampos <> '') then
          Result := concatenar([Result, xComandoCampos], pQuebraLinha);

        if (xComandoIndices <> '') then
          Result := concatenar([Result, xComandoIndices], pQuebraLinha);

        Result := concatenar([Result, ');', pQuebraLinha+pQuebraLinha], '');
      end;
    end;

    aAlterar    : begin
      xComandoCampos := criarComandoCampos(True, pTabela.campos);
      xComandoIndices := criarComandoIndices(True, pTabela.indices);
  
```

Figura 7 - Código fonte - comando para criar tabela

```

var
  I: Integer;
  xTabela: TTabela;
  xComando: String;
begin
  for I := 0 to (baseDados.tabelas.Count-1) do
  begin
    xTabela := baseDados.tabelas[I];
    if (xTabela.comparar) then
    begin
      xComando := criarComandoTabela(xTabela);
      if (xComando <> '') then
      case xTabela.acao of
        aNenhuma : comandos.Add(TComando.Create('', '/* SEM DIFERENÇAS NA TABELA ' + xTabela.nome + ' */'));
        aCriar    : comandos.Add(TComando.Create(xComando, '/* CRIAR TABELA ' + xTabela.nome + ' */'));
        aAlterar  : comandos.Add(TComando.Create(xComando, '/* ALTERAR TABELA ' + xTabela.nome + ' */'));
        aRemover  : comandos.Add(TComando.Create(xComando, '/* REMOVER TABELA ' + xTabela.nome + ' */'));
      end;
    end;
  end;
end;
end;
end;

```

Figura 8 - Código fonte - Montagem do comando com comentários

### 3.3.2 Operacionalidade da implementação

Nesta subseção é apresentada a seqüência de telas e operações para conseguir utilizar corretamente a ferramenta para sincronização estrutural de bases de dados MySQL e obter a seqüência de comandos necessários para sincronizar determinada estrutura de base de dados.

### 3.3.2.1 Gerenciamento de dados para conexão com as bases de dados

Para que a ferramenta se conecte nas determinadas bases de dados e possa ler as informações de suas estruturas é obrigatório ter o endereço do servidor, usuário e senha de acesso ao MySQL (com permissões de leitura da estrutura das bases de dados) e opcionalmente a porta para conexão. Na figura 9 é demonstrada a tela para gerenciar as conexões, onde são mantidas estas informações.



Figura 9 – Tela para gerenciar conexões

### 3.3.2.2 Seleção das bases de dados a serem comparadas

Tendo-se os dados para conexão às bases de dados, podemos então selecionar duas bases para realizar uma comparação. Para isto é necessário selecionar a conexão de origem, em seguida a base de dados de origem, a conexão de destino e a base de dados de destino. Depois de selecionadas as bases, é disponibilizada a opção para compará-las além de ser possível selecionar quais itens, eventualmente, devem ser ignorados. A figura 10 demonstra a seleção das bases de dados.

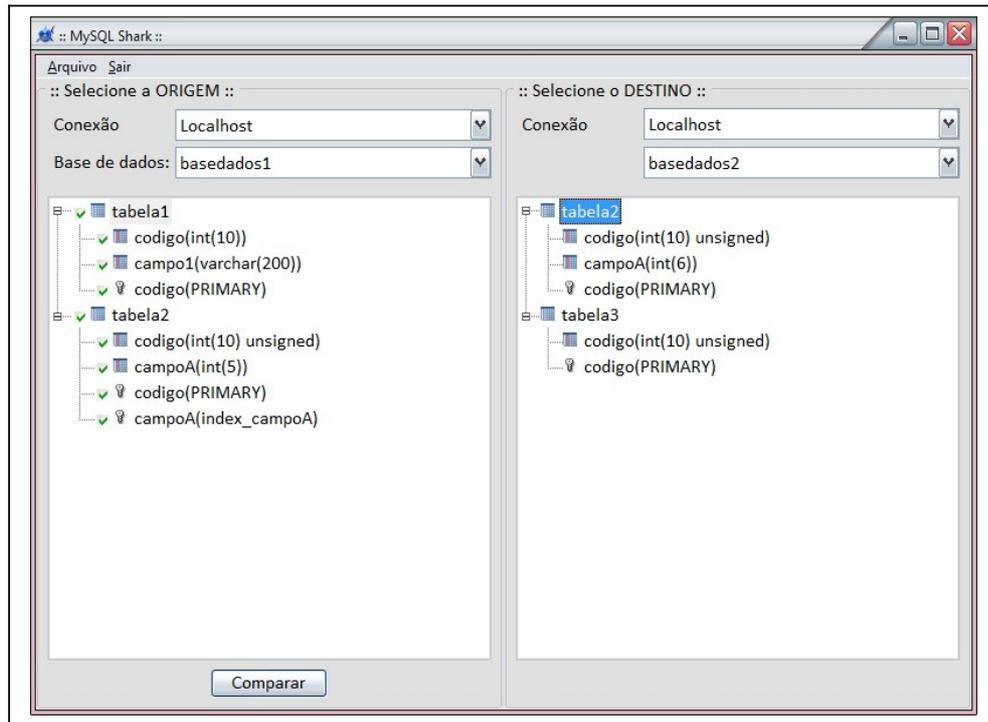


Figura 10 - Tela para selecionar as bases de dados

### 3.3.2.3 Opções para comparação

Ao clicar na opção para “Comparar” duas bases de dados, é apresentada tela para selecionar algumas opções de comparação, que é demonstrada pela figura 11.

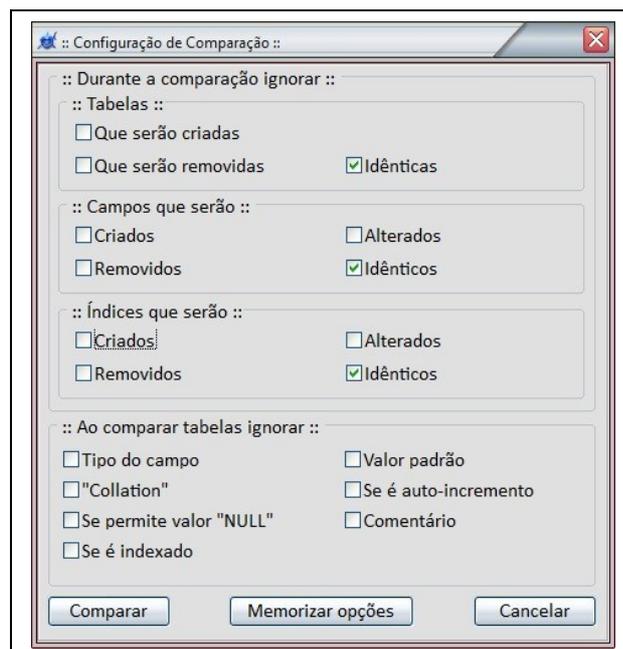


Figura 11 - Opções de comparação

### 3.3.2.4 Resultado da comparação

Após o processamento das diferenças, de acordo com as opções de comparação selecionadas, o resultado (quais alterações serão realizadas) é mostrado à direita. Onde de cores verde são os itens a serem criados, de cor vermelha os que serão alterados, azul os idênticos e de cor cinza os itens que serão removidos conforme ilustra a figura 12.

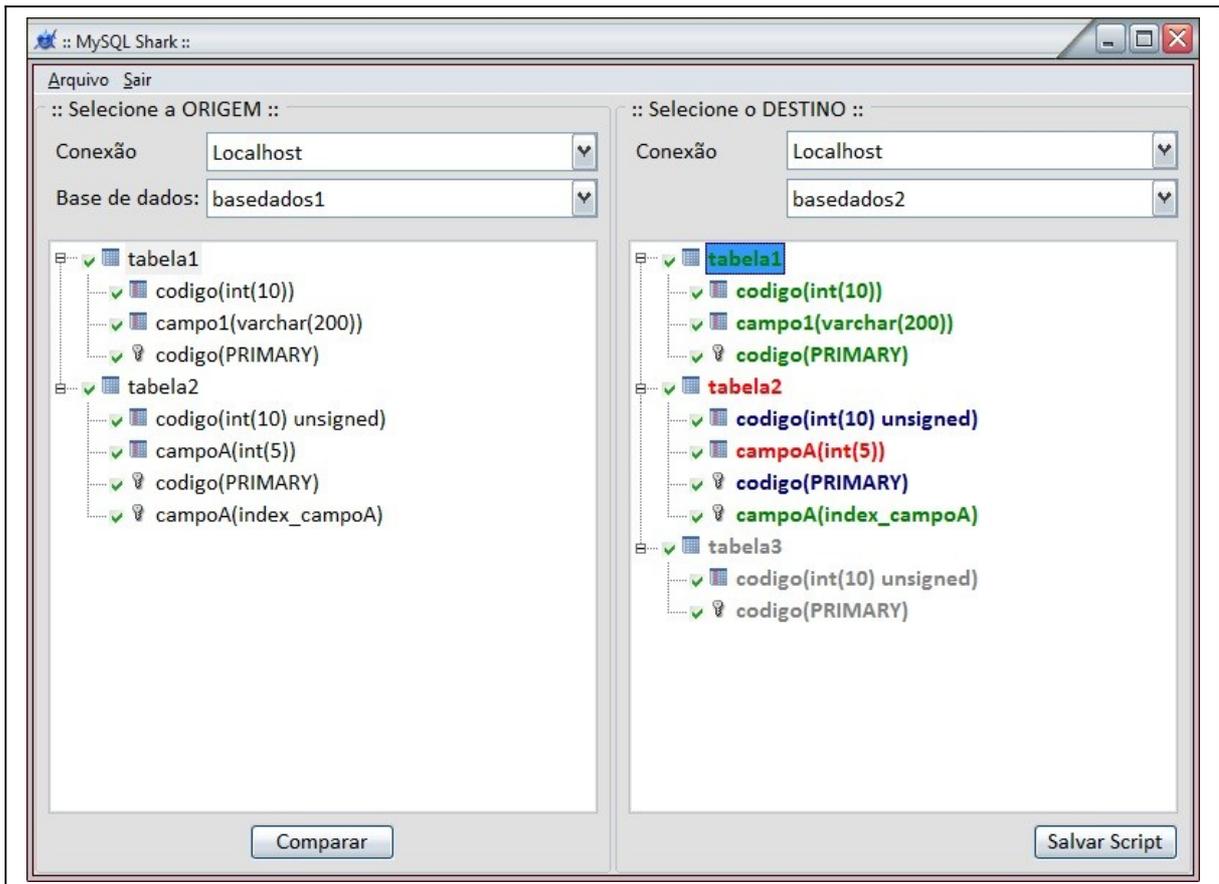


Figura 12 - Resultado da comparação

### 3.3.2.5 Script com os comandos para sincronização

Por fim, pode-se copiar o *script* para a área de transferência ou salvar em arquivo. A figura 13 demonstra o *script* gerado neste exemplo.

```

-----1-----2-----3-----4-----5-----
1  /** MySQL SHARK 1.0 **/
2  /** SINCRONIZAR basedados2 **/
3
4  USE basedados2;
5
6  /** INÍCIO DO SCRIPT **/
7
8  /** CRIAR TABELA tabela1 **/
9  CREATE TABLE tabela1 (
10     codigo int(10) NOT NULL,
11     campo1 varchar(200) COLLATE latin1_swedish_ci NULL,
12     PRIMARY KEY(codigo));
13
14 /** ALTERAR TABELA tabela2 **/
15 ALTER TABLE tabela2
16     CHANGE campoA int(5) NULL DEFAULT "1" COMMENT "",
17     KEY index_campoA (campoA) COMMENT="";
18
19 /** REMOVER TABELA tabela3 **/
20 DROP TABLE tabela3;
21
22 /** FIM DO SCRIPT **/

```

Figura 13 - *Script* para sincronização

### 3.4 RESULTADOS E DISCUSSÃO

O desenvolvimento deste trabalho permitiu a otimização do processo de sincronização de estrutura de bases de dados dos clientes da BLUEONE através do desenvolvimento da ferramenta, ou seja, o objetivo principal deste trabalho foi alcançado.

Os objetivos específicos também foram alcançados, conforme demonstrado anteriormente nas figuras 10 onde é possível selecionar duas bases de dados para serem comparadas, na figura 12 que mostra a visualização das diferenças (resultado da comparação) e na figura 13 apresenta o *script* SQL gerado.

Comparando-se com os trabalhos correlatos, nenhum dos mesmos possui funcionalidades semelhantes à de sincronizar as estruturas entre duas bases de dados.

A ferramenta desenvolvida, já está em e em conversa informal com os usuários, verificou-se a satisfação dos mesmos. A mesma mostrou-se também mais adequada, ao conseguir comparar a estrutura de uma base de dados MySQL versão 5.1.3 com outra MySQL versão 4.1. A ferramenta SQLyog Enterprise – MySQL GUI v8.02 RC, desenvolvida pela empresa Webyog, ao ser testada, apresentou erros ao comparar as mesmas estruturas.

## 4 CONCLUSÕES

Neste trabalho se propôs o desenvolvimento de uma ferramenta que permitisse a otimização do processo de sincronização da estrutura de bases de dados dos clientes da BLUEONE. A ferramenta desenvolvida conseguiu atender seus objetivos.

Como principal benefício da ferramenta pode-se citar a redução da possibilidade de falhas resultantes das operações manuais e a otimização do processo, que se tornou mais ágil e íntegro, permitindo à BLUEONE realocar recursos antes voltados à sincronização para outras necessidades da organização. Além de seus objetivos, a ferramenta disponibiliza ao usuário uma interface gráfica através da qual é possível que o mesmo visualize as diferenças antes de efetuar a sincronização. Após comparar e visualizar a sincronização, o usuário pode ainda selecionar quais itens deverão mesmo ser sincronizados ou não.

Como maior dificuldade, destacou-se a questão do desenvolvimento da rotina de comparação das estruturas, devido ao fato de algumas das bases de clientes da BLUEONE não serem mantidas em bancos de dados MySQL, cuja versão é anterior à 5.5.1. Entretanto, esta dificuldade foi superada e a ferramenta consegue comparar versões anteriores como a 4.0.

A ferramenta possui algumas limitações identificadas, tais como:

- a) ausência do controle das informações mantidas na base de dados. Se um campo for removido ou subdividido em mais campos, a informação será perdida;
- b) consistência na alteração de índices ou chaves estrangeiras.

O desenvolvimento deste trabalho permitiu a ampliação dos conhecimentos e possibilitou crescimento pessoal e profissional através de conhecimentos obtidos em buscas e leituras, e também, da prática que juntos, permitiram a conclusão deste trabalho.

### 4.1 EXTENSÕES

A ferramenta desenvolvida neste trabalho oferece os recursos necessários à otimização da sincronização, contudo, outros recursos poderiam ser incluídos. Dentre eles, destacam-se:

- a) portabilidade – visto que a ferramenta foi desenvolvida somente para o ambiente Microsoft Windows;
- b) controle da seqüência dos comandos no *script* – para a BLUEONE não verificou-se

esta necessidade mas para outras organizações, pode ser necessário controlar esta seqüência, onde um comando deve vir antes de outro pois, sendo executados de forma inversa podem ocasionar erros ou até mesmo a perda de dados.

## REFERÊNCIAS BIBLIOGRÁFICAS

BACHMANN, J. **Aplicativo web para definição do modelo lógico no projeto de banco de dados relacional**. 2007. 96f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

FURTADO, M. R. F.. **Protótipo de software para criação e gerenciamento de replicação em banco de dados**. 1999. 65f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

IMHOF, M. O. Q.. **Ferramenta para replicação de dados no SGBD POSTGREESQL**. 2007. 57f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

MECENAS, I.; OLIVEIRA, V. **Banco de Dados: do modelo conceitual à implementação física**. Rio de Janeiro: Alta Books Ltda, 2005.

RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de Gerenciamento de Banco de Dados**. São Paulo: McGraw-Hill, 2008.

SUEHRING, Steve. **MySQL: a Bíblia**. Rio de Janeiro: Campus, 2002.

WEBYOG INC. **SQLyog Enterprise – MySQL GUI v8.02 RC**. Santa Clara, USA, 2011. Disponível em: <[www.webyog.com](http://www.webyog.com)>. Acessado em: 16 jun. 2011.

WERNER, C. E.. **Ferramenta de gerenciamento para o banco de dados Firebird**. 2004. 50f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

## APÊNDICE A – DETALHAMENTO DOS CASOS DE USO

No Quadro 3 apresenta-se o detalhamento do caso de uso “UC01 – Pré-visualizar sincronização”.

Nome do caso de uso	UC01 – Pré-visualizar sincronização.
Descrição	Usuário pré-visualiza sincronização.
Ator	Usuário
Pré-condição	As conexões dos bancos de dados devem estar previamente cadastradas. Os usuários dos bancos de dados devem ter no mínimo permissão de leitura da estrutura das bases de dados.
Fluxo principal	<ol style="list-style-type: none"> <li>1. Usuário acessa a ferramenta;</li> <li>2. Ferramenta requisita que sejam selecionadas duas bases de dados;</li> <li>3. Usuário selecione duas bases de dados, podendo estas, serem de um mesmo banco de dados ou de banco de dados distintos;</li> <li>4. Ferramenta requisita que sejam informados quais itens estruturais deverão ser comparados;</li> <li>5. Usuário informa quais itens da estrutura deverão ser comparados;</li> <li>6. Ferramenta identifica as estruturas a serem comparadas;</li> <li>7. Ferramenta compara os itens selecionados e apresenta visualmente as diferenças encontradas;</li> </ol>
Fluxo alternativo	<ul style="list-style-type: none"> <li>• no item 3, caso um ou os dois bancos de dados não estiverem acessíveis, a ferramenta exibirá um alerta com mensagem notificando a falha de conexão;</li> <li>• no item 6, alerta com mensagem notificando falha na leitura das estruturas;</li> <li>• no item 7, alerta com mensagem notificando falha na comparação das estruturas.</li> </ul>
Pós-condição	Usuário visualiza as alterações que serão feitas para efetuar a sincronização dos itens previamente selecionados.

Quadro 3 – Descrição do caso de uso “UC01 – Pré-visualizar sincronização”

No Quadro 4 apresenta-se o detalhamento do caso de uso “UC02. Sincronizar base de dados”.

Nome do caso de uso	UC02 – Sincronizar base de dados.
Descrição	Usuário sincroniza base de dados.
Ator	Usuário
Pré-condição	As conexões dos bancos de dados devem estar previamente cadastradas. Os usuários dos bancos de dados devem ter no mínimo permissão de leitura da estrutura das bases de dados.
Fluxo principal	<ol style="list-style-type: none"> <li>1. Usuário acessa a ferramenta;</li> <li>2. Ferramenta requisita que sejam selecionadas duas bases de dados;</li> <li>3. Usuário selecione duas bases de dados, podendo estas, serem de um mesmo banco de dados ou de banco de dados distintos;</li> </ol>

	<ol style="list-style-type: none"> <li>4. Ferramenta requisita que sejam informados quais itens estruturais deverão ser comparados;</li> <li>5. Usuário informa quais itens da estrutura deverão ser comparados;</li> <li>6. Ferramenta compara os itens selecionados e apresenta visualmente as diferenças encontradas;</li> <li>7. Ferramenta requisita que seja selecionada opção entre efetuar a sincronização ou gerar arquivo de texto com os comandos necessários para sincronizar a base de dados;</li> <li>8. Usuário seleciona a opção desejada;</li> <li>9. Ferramenta identifica a estrutura a ser comparada;</li> <li>10. Ferramenta sincroniza a base de dados ou gera o arquivo de texto.</li> </ol>
Fluxo alternativo	<ul style="list-style-type: none"> <li>• no item 3, caso um ou os dois bancos de dados não estiverem acessíveis, a ferramenta exibirá um alerta com mensagem notificando a falha de conexão;</li> <li>• no item 9, pode-se exibir um alerta com mensagem notificando falha na leitura das estruturas;</li> <li>• ou alerta com mensagem notificando falha na comparação das estruturas;</li> <li>• no item 10, caso houver falha na execução dos comandos a estrutura ou na geração do arquivo de texto, a ferramenta exibirá alerta.</li> </ul>
Pós-condição	Base de dados sincronizada ou arquivo de texto gerado.

Quadro 4 – Descrição do caso de uso “UC02 – Sincronizar base de dados”

No Quadro 5 apresenta-se o detalhamento do caso de uso “UC03 – Gerenciar conexões”.

Nome do caso de uso	UC03 – Gerenciar conexões.
Descrição	Usuário inclui, altera o remove dados de conexão a banco de dados.
Ator	Usuário
Fluxo principal	<ol style="list-style-type: none"> <li>1. O usuário acessa a opção para gerenciar as conexões.</li> <li>2. Ferramenta apresenta as conexões já salvas;</li> <li>3. O usuário seleciona se deseja incluir uma nova conexão, ou seleciona a conexão que deseja alterar/remover;</li> <li>4. Ferramenta requisita dados necessários para conexão;</li> <li>5. Usuário informa os dados requisitados;</li> <li>6. Ferramenta salva as informações em arquivo em formato <i>Extensible Markup Language (XML)</i>;</li> </ol>
Fluxo alternativo	<ul style="list-style-type: none"> <li>• no item 5, caso o usuário não informar os dados corretamente, a ferramenta exibirá um alerta;</li> <li>• no item 6, caso houver falha na escrita do arquivo XML exibirá um alerta;</li> </ul>
Pós-condição	Dados de conexão a banco de dados incluídos, alterados ou removidos.

Quadro 5 – Descrição do caso de uso “UC03 – Gerenciar conexões”