UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUIZ FERNANDO BÖHM

Elaboração de uma Estratégia de Deduplicação de Dados Utilizando Técnicas de Blocagem em um Cadastro Hospitalar de Pacientes

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Prof. Dr. Carlos Alberto Heuser
Orientador

Porto Alegre, julho de 2010.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Böhm, Luiz Fernando

Elaboração de uma Estratégia de Deduplicação de Dados Utilizando Técnicas de Blocagem em um Cadastro Hospitalar de Pacientes / Luiz Fernando Böhm. - Porto Alegre: Instituto de Informática da UFRGS, 2010.

32 f.: il.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Rio Grande do Sul. Curso de Ciência da Computação, Porto Alegre, BR-RS, 2010. Orientador: Heuser, Carlos A.

1. Deduplicação. 2. Blocagem. 3. Soundex. 4. BuscaBR. I. Heuser, Carlos A. II. Título.

AGRADECIMENTOS

A todos aqueles que direta ou indiretamente me incentivaram e me apoiaram durante todo o período da Graduação;

A minha família, especialmente a meus pais, por todo suporte e incentivo aos meus estudos;

A minha esposa, que se mostrou uma grande companheira, tendo muita paciência e me motivando em todos os momentos;

Ao meu orientador, pela confiança em mim depositada, e pelos ensinamentos ao longo do período de orientação;

Aos colegas do Hospital Nossa Senhora da Conceição S.A. e aos colegas do curso de Ciência da Computação;

Ao Instituto de Informática e à Universidade Federal do Rio Grande do Sul que proporcionaram um ensino público, gratuito e de qualidade.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
MSM	Micronetics Standard MUMPS
MUMPS	Massachusetts General Hospital Utility Multi-Programming System
PC	Pair Completeness
RAM	Random Access Memory
RR	Reduction Rate
SGBD	Sistema Gerenciador de Banco de Dados
SQL	Structured Query Language
UFRGS	Universidade Federal do Rio Grande do Sul

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

O presente trabalho consiste na elaboração de uma estratégia de deduplicação de dados utilizando técnicas de blocagem e algoritmos fonéticos em um cadastro hospitalar de pacientes. A chave de blocagem que apresenta a maior qualidade é a chave que utiliza os algoritmos fonéticos Soundex em conjunto com o BuscaBR, aplicados ao nome da mãe do paciente.

Todos os passos realizados na preparação da base de dados de testes, no pré-processamento dos dados, na deduplicação dos registros da base de dados completa e todas as métricas utilizadas na análise da qualidade dos resultados estão detalhados.

Como resultado deste trabalho é apresentada a estratégia de deduplicação que maximiza a quantidade de registros duplicados encontrados em uma base de dados de testes previamente avaliada, assim como o conjunto de registros possivelmente duplicados que foram encontrados na base de dados completa.

Também são analisadas propostas adicionais para melhorar o desempenho e a qualidade do processo de blocagem e deduplicação.

Palavras-Chave: Deduplicação, Blocagem, Soundex, BuscaBR

Elaborating a Record Linkage Strategy using Blocking techniques on a Hospital Patients Database

ABSTRACT

This work consists in elaborating a record linkage strategy using blocking techniques and phonetic algorithms on a hospital patient's database. The blocking key with the highest quality was the key using the phonetic algorithms Soundex combined with the BuscaBR, applied on the patient mother's name.

All the steps taken in preparing the test database, pre-processing of data, in the record linkage of the complete database and all the metrics used in analyzing the quality of the results are detailed.

As a result of this work is presented the record linkage strategy that maximizes the amount of duplicate records found in the test database previously evaluated, as well as possibly duplicate set of records that were found in the complete database.

Besides, additional proposals are analyzed to improve the performance and quality of the blocking and record linkage process.

Keywords: Record Linkage, Blocking, Soundex, BuscaBR

1 INTRODUÇÃO

Com a constante evolução dos meios de armazenamento e dos Sistemas de Gerenciamento de Banco de Dados está cada vez mais fácil armazenar grandes volumes de informação a um custo cada vez menor. Em contrapartida, extrair informações úteis e confiáveis dessa massa de dados pode ser uma tarefa árdua caso estes dados contenham erros de digitação, dados inválidos ou ausentes e até mesmo duplicados (JIN *et al*., 2003).

No caso de um cadastro de pacientes com milhões de registros, dados duplicados podem provocar uma série de problemas na gestão hospitalar visto que este cadastro representa uma tabela chave no sistema hospitalar. Além de desperdício de espaço em disco e do aumento dos custos no envio de malas direta, dados de pacientes duplicados tornam um prontuário médico segmentando, ocasionando dificuldades em elaborar um diagnóstico completo (quando um médico, por exemplo, acessa apenas os exames constantes em um dos muitos cadastros para este mesmo paciente), além de imprecisão em relatórios estatísticos de atendimentos que são utilizados na gestão de um hospital para provisionar recursos humanos, espaço físico e material no atendimento ambulatorial e de emergência.

Este trabalho tem como principal objetivo desenvolver uma estratégia de deduplicação de dados no cadastro de pacientes utilizando métodos de blocagem em conjunto com outras técnicas, como o uso de algoritmos fonéticos Soundex (RUSSEL *et al.*, 1918) e BuscaBR (LUCENA, 2006) e Distância de Levenshtein (LEVENSHTEIN, 1965) para identificar os registros duplicados tornando possível uma limpeza da base de dados permitindo minimizar os problemas citados anteriormente.

Como resultado, é identificado um conjunto de registros possivelmente duplicados, tornando possível a eliminação ou mesclagem destes registros após uma análise e eliminação de falsos positivos. Esta solução também pode ser utilizada na fase de inclusão de novos pacientes, onde é possível apresentar uma lista de registros similares para evitar novas inclusões duplicadas.

Este trabalho está organizado como segue. O Capítulo 2 apresenta o estado da arte na deduplicação de dados e no processo de blocagem assim como as métricas e algoritmos utilizados. O Capítulo 3 descreve a base de dados e seus atributos e também as ferramentas e técnicas que são utilizadas no trabalho. No Capítulo 4, são apresentadas a chave de blocagem e a estratégia escolhida. No Capítulo 5 estão descritos os resultados obtidos com a estratégia de deduplicação. Finalmente, o Capítulo 6 apresenta as conclusões e trabalhos futuros.

2 DEDUPLICAÇÃO DE DADOS

2.1 O Processo de Deduplicação

O processo de deduplicação de dados consiste em identificar de forma rápida e precisa os registros correspondentes a uma mesma entidade de uma ou mais base de dados, sendo uma entidade um conjunto de informações de um contexto específico, como pacientes, médicos ou medicamentos, armazenadas em meio digital, como uma base de dados relacional, por exemplo (LIFANG *et al*., 2003). Esta identificação, idealmente, é realizada através de uma comparação de cada um dos registros contra todos os demais – o que se torna inviável quando o número de registros é muito grande (CHRISTEN, 2007).

2.2 Blocagem

A técnica de blocagem é utilizada para reduzir o número de comparações entre os registros. Esta comparação de todos os registros com os demais faz com que a complexidade seja quadrática ($O(n^2)$). Por exemplo, uma base de dados com 1.000 registros gera um total de 499.500 comparações. Quando estes mesmos 1.000 registros são divididos em 20 blocos com 50 registros cada, o número de comparações é reduzido para 24.500 (CHRISTEN, 2007).

Esta blocagem é realizada através da separação dos registros em blocos menores identificados por uma chave de blocagem onde, preferencialmente, todos os pares de registros duplicados estejam no mesmo bloco. Assim, apenas os registros que estão no mesmo bloco são comparados entre si, reduzindo o total de comparações sem que ocorra redução da precisão (BAXTER *et al.*, 2003).

2.3 Deduplicação dentro de um Bloco

A deduplicação dentro de um bloco consiste na execução de um algoritmo que compara cada registro com os demais do mesmo bloco e define se tal registro é um par duplicado do outro ou não.

2.4 Métricas de Qualidade

As métricas de qualidade utilizadas neste trabalho são métricas amplamente utilizadas em artigos científicos relacionados à blocagem, como o Pair Completeness (PC), Reduction Rate (RR) e F-Score (BAXTER *et al.*, 2003).

O PC (Figura 2.1) retorna a taxa de pares corretos após a blocagem, ou seja, indica qual a taxa dos pares duplicados que ficaram no mesmo bloco, possibilitando que sejam identificados no processo de deduplicação. Já o RR (Figura 2.2), indica qual a redução da quantidade de comparações que o processo de blocagem vai proporcionar. Ainda, o F-Score (Figura 2.3) serve como um referencial no processo de decisão sobre a chave de blocagem a ser utilizada já que retorna a média harmônica entre o PC e o RR. Cabe ressaltar que não são apenas essas métricas que pesam na decisão da melhor chave de blocagem, pois, quando o desempenho é um aspecto importante, deve-se também levar em conta o número de blocos gerados e a média de registros por bloco, visto que estes fatores interferem amplamente no tempo de processamento da deduplicação.

$$PC = \frac{\text{Pares Corretos nos Blocos}}{\text{Total de Pares Corretos}}$$

Figura 2.1: Pair Completeness – Taxa de Pares Corretos após Blocagem

$$RR = 1 - \frac{\text{Total de Pares Gerados nos Blocos}}{\text{Total de Pares Possíveis}}$$

Figura 2.2: Reduction Ratio – Taxa Redução no Número de Comparações

$$F\text{-}Score = \frac{2 \text{ x } RR \text{ x } PC}{RR + PC}$$

Figura 2.3: F-Score – Média Harmônica entre PC e RR

2.5 Algoritmos Utilizados

Para a criação dos blocos foram utilizados os algoritmos Soundex e BuscaBR. Para algumas verificações de similaridade de strings foi utilizada a Distância de Levenshtein.

A codificação Soundex consiste de uma letra seguida de três algarismos sendo que a letra é a primeira letra da string e os números são a codificação das demais consoantes segundo critérios do algoritmo como descrito abaixo:

1. Substituir as consoantes (sem mudar a primeira letra da string) como segue:

   b, f, p, v => 1

   c, g, j, k, q, s, x, z => 2

   d, t => 3

   l => 4

   m, n => 5

   r => 6
2. Substituir letras iguais adjacentes por um único dígito correspondente.
3. Remover todos os não-dígitos após a primeira letra.
4. Retornar a letra inicial e os primeiros três dígitos restantes. Se necessário, adicionar zeros à direita para formar uma letra e três dígitos.

Neste trabalho é utilizada a implementação do Soundex nativa do SQL Server 2000. Conforme a Tabela 2.1 pode-se observar que o algoritmo apresenta problemas quando as strings comparadas começam com letras diferentes, mesmo representando o mesmo fonema.

Tabela 2.1: Exemplos de Aplicação do Soundex

String	**Soundex**
GIULIETA ALVES	G430
JULIETA ALVEZ	J430
HELENA MARTINS	H450
ELENA MARTINS	E450
CAREN PEREIRA	C650
KAREN FERREIRA	K650

O BuscaBR é uma adaptação do algoritmo fonético Soundex para o português do Brasil, já que o Soundex tem como base a língua inglesa – o que apresenta inúmeras deficiências quando utilizado com fonemas de outras línguas. A codificação do BuscaBR executa substituições de conjuntos de letras por seu fonema correspondente, remove letras repetidas em seqüência, e também letras mudas. A implementação do algoritmo utilizada neste trabalho, é feita em forma de uma função do SQL Server 2000 com base nas regras descritas abaixo, conforme referência bibliográfica do algoritmo:

1. Converter todas as letras para Maiúsculo;
2. Eliminar todos os acentos;
3. Substituir Y por I;

4. Substituir BR por B;
5. Substituir PH por F;
6. Substituir GR, MG, NG, RG por G;
7. Substituir GE, GI, RJ, MJ, NJ por J;
8. Substituir Q, CA, CO, CU, C por K;
9. Substituir LH por L;
10. Substituir N, RM, GM, MD, SM e Terminação AO por M;
11. Substituir NH por N;
12. Substituir PR por P;
13. Substituir Ç, X, TS, C, Z, RS por S;
14. Substituir LT, TR, CT, RT, ST por T;
15. Substituir W por V;
16. Eliminar as terminações S, Z, R, R, M, N, AO e L;
17. Substituir R por L;
18. Eliminar todas as vogais e o H;
19. Eliminar todas as letras em duplicidade;

Como se observa na Tabela 2.2, o algoritmo BuscaBR elimina a distinção entre as letras iniciais do nome quando representam o mesmo fonema.

Tabela 2.2: Exemplos de Aplicação do BuscaBR

String	**BuscaBR**
GIULIETA ALVES	JLT LVS
JULIETA ALVEZ	JLT LVS
CARINA DA SILVA	KLM D SLV
KARINA SILVA	KLM SLV
CAREN PEREIRA	KLM PL
KAREN FERREIRA	KLM FL

A Distância de Levenshtein retorna a quantidade mínima de operações (remoções, inserções e substituições de caracteres) para transformar uma string em outra. Na versão normalizada, retorna o resultado do número de operações dividido pelo tamanho da maior string comparada – o que resulta em um número entre 0 e 1. Abaixo o pseudocódigo do algoritmo da Distância de Levenshtein:

```
Função Levenshtein (Caracter : str1[1..tamStr1], Caracter : str2[1..tamStr2]) : Inteiro
Início
/*  variável matriz com tamStr1+1 linhas e tamStr2+1 colunas */
Inteiro:  matriz[0..tamStr1, 0..tamStr2]
/*  variáveis X e Y são usadas como índice para iterar str1 e str2 */
Inteiro:  X, Y, custo

Para X de 0 até tamStr1
        matriz[X, 0] := X
Para Y de 0 até tamStr2
        matriz[0, Y] := Y

Para X de 1 até tamStr1
        Para Y de 1 até tamStr2
                Se str1[X] = str2[Y] Então custo := 0
                        Senão custo := 1
                matriz[X, Y] := menor(
                matriz[X-1, Y  ] + 1,          /* Deletar */
                matriz[X  , Y-1] + 1,          /* Inserir */
                matriz[X-1, Y-1] + custo       /* Substituir */
                                )

Levenshtein := matriz[tamStr1, tamStr2]
Fim
```

No trabalho são utilizadas duas implementações: uma implementação em uma função no SQL Server 2000 e outra em linguagem de programação MUMPS. A Tabela 2.3 apresenta um exemplo da aplicação do algoritmo da Distância de Levenshtein retornando o número de operações e o resultado normalizado.

Tabela 2.3: Exemplo de Aplicação da Distância de Levenshtein

String 1	**String 2**	**Distância de Levenshtein (nº de operações)**	**Distância de Levenshtein (normalizada)**
GIULIETA ALVES	JULIETA ALVEZ	3	0,21
CARINA DA SILVA	KARINA SILVA	4	0,27
JOAO SEM NOME	JOAO NINGUEM	7	0,54

3 DESENVOLVIMENTO E EXPERIMENTOS

3.1 Base de Dados

A base de dados utilizada neste trabalho é a tabela do cadastro de pacientes de um hospital, armazenada em uma base de dados relacional no SQL Server 2000. A base de dados completa, amostrada em agosto/2009, apresenta 2.841.881 registros e os atributos relevantes para o trabalho são o REGISTRO DO PACIENTE, NOME, DATA DE NASCIMENTO e NOME DA MÃE.

Para efetuar a escolha da chave de blocagem e elaborar a estratégia de deduplicação, foi gerada uma base de dados de teste com 10.000 registros extraídos da base de dados completa, conforme os seguintes critérios:

- Inserção de 450 registros que possuem duplicatas, selecionados manualmente conforme ordenação da tabela completa pelos atributos acima descritos;
- Inserção das duplicatas destes registros;
- Inserção de 348 registros incompletos: DATA DE NASCIMENTO incorreta, sem NOME (Ignorado) ou sem NOME DA MÃE (NI, por exemplo), conforme dados estatísticos extraídos da base de dados completa;
- Inserção dos registros restantes para completar 10.000 registros, escolhidos randomicamente.

Com isso, a base de dados de teste apresenta um total de 513 registros que possuem pelo menos uma duplicata, perfazendo um total de 1090 registros (somando-se todos os duplicados). Com estes 513 registros duplicados, pode-se obter 658 pares de registros. Como a bibliografia na área trabalha com medidas de qualidade sobre pares de registros, o número que se deve ter como base para medição de todos os índices é o de **658 pares de registros duplicados**.

Cabe observar, que a porcentagem de registros duplicados e de registros incompletos que a base de dados de testes deveria ter, foi obtida através de heurísticas que os Analistas de Sistemas responsáveis pela manutenção do cadastro de pacientes consideraram adequadas.

3.2 Características dos Atributos da Base de Dados

Atributo NOME: Limitado em 250 caracteres, pode apresentar 2 tipos de dados que devem ser tratados especialmente:

1- Recém-Nascidos (RN): Quando é registrado um recém-nascido no sistema que ainda não tem um documento de identificação oficial (certidão de nascimento, geralmente) ele é registrado como "RN Nome_da_Mãe X", onde X é um número que representa o cardinal do filho que esta paciente está registrando. Por ex. "RN Maria da Silva" ou "RN Maria da Silva II", caso já exista o primeiro RN desta Mãe no cadastro, ou ainda "RN Maria da Silva G I" caso este RN seja o primeiro filho cadastrado e tenha irmão gêmeo. Podem também aparecer nomes como "RN II de Maria da Silva".

2- Pacientes Não Identificados: Outro tipo de inserção especial para o atributo NOME, é quando não é possível identificar um paciente que é atendido (que chega desacordado e sem um documento, por exemplo). Assim, ele é cadastrado como "Ignorado Y", onde Y é a numeração por extenso do primeiro número disponível para um "ignorado". Por ex. "Ignorado Trinta e Cinco" representa o trigésimo quinto paciente deste tipo cadastrado no sistema.

Atributo NOME DA MÃE: Também com 250 caracteres no máximo. Assim como no caso do atributo NOME, pode haver casos de Mães não identificadas (por desconhecimento do paciente ou em casos de abandonos de recém-nascidos, por exemplo). Nesses casos, o atributo NOME DA MÃE recebe também um identificador de desconhecido. Os mais utilizados são: "NI" (Não Informado) e "Sem Informação".

Atributo DATA DE NASCIMENTO: Alfanumérico com no máximo 10 caracteres (para o formato dd/mm/aaaa). Este atributo pode conter uma data inválida (geralmente com o ano maior do que o ano atual) ou, na maioria dos casos, com a string "0//".

3.3 Ferramentas Utilizadas

- SQL Server 2000 Enterprise Edition: SGBD que armazena o cadastro de pacientes a fim de agilizar e flexibilizar as consultas.
- SQL Query Analyzer: Ferramenta do SQL Server 2000 utilizada para elaborar e analisar as consultas bem como criar índices e verificar o seu custo de execução.
- MSM Workstation 2.0: API de desenvolvimento de janelas baseado na linguagem MUMPS.

- MUMPS: SGBD e linguagem de programação desenvolvida pelo Hospital Geral de Massachusetts.

3.4 Técnicas Preliminares

Antes de iniciar o processo de blocagem e deduplicação foi necessário efetuar um pré-processamento da base de dados. Os passos realizados estão descritos abaixo:

- Alteração dos atributos NOME e NOME DA MÃE para Case Insensitive e Accent Insensitive, a fim de padronizar os dados e tornar a comparação das strings mais fácil e rápida;
- Criação do atributo BUSCABR_NOME – que contém a string resultante da aplicação do algoritmo BuscaBR sobre o NOME do Paciente;
- Criação do atributo BUSCABR_MAE – que contém a string resultante da aplicação do algoritmo BuscaBR sobre o NOME DA MÃE;
- Criação do atributo BLOCO – que contém a string que identifica o bloco ao qual o registro está vinculado;
- Criação do atributo FLAG – que vai receber o REGISTRO DO PACIENTE que corresponde a sua duplicata, ou 0 (zero) caso não seja localizada uma duplicata;
- Criação de índices, conforme a necessidade em cada etapa do processo.

4 SOLUÇÃO PROPOSTA

A solução proposta de deduplicação consiste em utilizar técnicas de blocagem com a finalidade de reduzir o número de comparações entre os registros, porém maximizando o índice de registros duplicados localizados. A figura 4.1 mostra o número de comparações na base de dados completa do cadastro de pacientes com 2.841.881 registros caso não fosse utilizada blocagem:

$$\frac{2.841.881 \text{ x } (2.841.881 - 1)}{2} = 4.038.142.388.140$$

Figura 4.1: Número total de comparações sem utilizar blocagem.

4.1 Escolha da Chave de Blocagem

A escolha da chave de blocagem foi feita analisando-se a base de dados de teste. Foram testadas diversas chaves e a análise da chave a ser utilizada foi feita com base nos índices PC e RR, além do número de blocos e quantidade de registros por bloco. Na Tabela 4.1 é apresentada a análise das diferentes chaves de blocagem que foram testadas.

Tabela 4.1: Análise das Chaves de Blocagem

Chave de Blocagem	**Nº de Pares Duplicados nos Blocos**	**Nº de Pares Possíveis**	**Nº de Blocos**	**Média de Registros por Bloco**
Soundex BuscaBR Mãe	635	2277215	408	24,51
Ano Nascimento	634	638849	110	90,91
Soundex Mãe	632	1191941	968	10,33
Data Nascimento	621	2816	8034	1,24
Soundex BuscaBR Nome	602	939456	425	23,53
Soundex Nome	595	373344	989	10,11
3 Primeiras Letras Nome	583	662962	765	13,07
BuscaBR Mãe	556	51875	8187	1,22
BuscaBR Nome	395	501	9556	1,05

A Tabela 4.2 apresenta um comparativo entre as métricas de qualidade de todas as chaves de blocagem testadas.

Tabela 4.2: Métricas de Qualidade das Chaves de Blocagem

Chave de Blocagem	PC	RR	F-SCORE
Soundex BuscaBR Mãe	96,50%	95,445%	95,97%
Ano Nascimento	96,35%	98,722%	97,52%
Soundex Mãe	96,05%	97,616%	96,83%
Data Nascimento	94,38%	99,994%	97,10%
Soundex BuscaBR Nome	91,49%	98,121%	94,69%
Soundex Nome	90,43%	99,253%	94,63%
3 Primeiras Letras Nome	88,60%	98,674%	93,37%
BuscaBR Mãe	84,50%	99,896%	91,55%
BuscaBR Nome	60,03%	99,999%	75,02%

4.2 Técnica Atual

Com os resultados da análise realizada sobre a base de dados de teste, definiu-se que a chave de blocagem que a ser utilizada na base de dados completa é a chave gerada com a aplicação do algoritmo Soundex sobre a string resultante da aplicação do algoritmo BuscaBR no atributo NOME DA MÃE. Esta chave de blocagem é a que apresenta o maior número de pares duplicados dentro dos mesmos blocos (635 pares de um total de 658 pares possíveis). Esta chave de blocagem não é a de maior F-Score, mas como a precisão é mais importante do que o desempenho para este trabalho, ela foi escolhida porque seu PC foi o maior entre as demais chaves (96,50%). Mesmo não sendo a chave com maior RR, ela apresenta 95,445% de redução do número de comparações, o que significa que este número passou de 49.995.000 para 2.277.215 comparações.

Após a escolha da chave de blocagem, foi elaborada uma interface em MSM Workstation (ver Apêndice) para ajudar na visualização do processo de deduplicação e no desenvolvimento de cada etapa (Pré-processamento dos registros com Nome da Mãe inválido, Sub-blocagem, Deduplicação e Exibição dos Resultados).

Foram elaboradas três estratégias conforme descrição abaixo:

Estratégia 1: Os registros analisados são considerados como representantes da mesma entidade quando uma dessas regras é válida:

1- A distância de edição do NOME DA MÃE e da DATA DE NASCIMENTO são iguais a zero e a distância de edição do NOME é menor do que 0,25;

2- A distância de edição do NOME é igual a zero, a distância de edição do NOME DA MÃE é menor do que 0,2 e a distância de edição da DATA DE NASCIMENTO é menor do que 0,2;

3- A distância de edição do NOME e a distância de edição da DATA DE NASCIMENTO são iguais a zero;

4- A distância de edição do NOME e do NOME DA MÃE são iguais a zero;

5- Caso uma das DATAS DE NASCIMENTO seja inválida e a distância de edição do NOME seja menor do que 0,25 e a distância de edição do NOME DA MÃE seja menor do que 0,2;

6- A distância de edição da DATA DE NASCIMENTO seja igual a zero e a distância de edição do NOME DA MÃE seja menor do que 0,25 e a distância de edição do NOME seja menor do que 0,2;

7- A distância de edição da DATA DE NASCIMENTO seja menor ou igual a 0,2, a distância de edição do NOME DA MÃE seja menor do que 0,25 e pelo menos 2 partes do NOME sejam iguais (apenas para partes com mais de 3 caracteres).

Nesta estratégia, foi utilizada uma implementação do algoritmo de Distância de Edição no SQL Server 2000.

Os registros com o atributo NOME DA MÃE inválido ficaram de fora da blocagem, fato que fez com que o número de 635 pares possíveis fosse reduzido para 627 pares. Também não foi feito nenhum tratamento especial com os recém nascidos e com alterações de nomes devido à mudança de estado civil.

Com isso, 427 registros que têm NOME DA MÃE inválido ficaram de fora dos blocos, sendo que nestes registros há 36 pares de registros duplicados. Conforme é apresentado na Tabela 4.3, a porcentagem de pares encontrados é de 84,95% sobre o total de pares da base de dados de teste.

Tabela 4.3: Estatísticas da Estratégia 1

Nº de Blocos	**Média de Registros por Bloco**	**Pares Possíveis**	**Pares Encontrados**	**% Sobre Possíveis**	**% Sobre Total**	**Tempo de Execução**
408	23,46	627	559	89,15%	84,95%	3 h 23 min

Estratégia 2: Com relação à Estratégia 1, foram feitas as seguintes modificações:

1- Adição dos registros com NOME DA MÃE inválido ao primeiro Bloco do REGISTRO encontrado (caso exista) com o atributo BUSCABR_NOME igual, eliminando espaços em branco.

2- Sub-Blocagem de blocos com mais do que 170 registros – Reduziu o tempo de deduplicação, sem reduzir a porcentagem de pares duplicados localizados.

Com a adição dos registros com NOME DA MÃE inválido aos blocos, restaram apenas 388 registros que não puderam ser aproveitados, sendo que há somente 2 pares de registros duplicados neste conjunto. Assim, o total de pares possíveis passou de 627 para 650. A sub-blocagem foi realizada utilizando o mês de nascimento como critério. Os blocos foram subdivididos, conforme seu tamanho, em 2, 3, 4, 6 ou 12 sub-blocos de modo que ficassem com no máximo 170 registros cada.

Como mostra a Tabela 4.4, a Estratégia 2 retorna 87,69% do total de pares da base de dados de teste e reduz significativamente o tempo de execução com o auxílio da sub-blocagem.

Tabela 4.4: Estatísticas da Estratégia 2

Nº de Blocos	**Média de Registros por Bloco**	**Pares Possíveis**	**Pares Encontrados**	**% Sobre Possíveis**	**% Sobre Total**	**Tempo de Execução**
432	22,25	650	577	88,77%	87,69%	1 h 14 min

Estratégia 3: Com relação à Estratégia 2, foram feitas as seguintes modificações:

1- Blocos não foram subdivididos;

2- Implementação do algoritmo de Distância de Edição em MUMPS para acelerar o processamento, evitando chavear o contexto da aplicação para o SQL Server 2000;

3- Tratamento dos recém nascidos – comparando o NOME com o NOME DA MÃE;

4- Tratamento de irmãos gêmeos – Caso NOME DA MÃE e DATA DE NASCIMENTO sejam iguais e NOMES similares, verifica a diferença entre os números de REGISTRO. Caso seja menor do que 306, considera que é o irmão gêmeo (cadastrado no sistema no mesmo dia);

5- Para comparar os pedaços do NOME e NOME DA MÃE, foram utilizados os atributos BUSCABR_NOME e BUSCABR_MÃE ao invés dos atributos NOME e NOME DA MÃE, respectivamente;

6- Alteração da validação da DATA DE NASCIMENTO – Não considera apenas 1 caractere diferente, mas também 1 dia, 1 mês ou 1 ano de diferença entre as DATAS DE NASCIMENTO como válidas;

7- Alteração da ordem das regras para eliminar testes desnecessários – as regras mais restritivas foram colocadas em primeiro lugar.

Como se observa na Tabela 4.5, a Estratégia 3 retorna 96,05% do total de pares da base de testes e ainda reduz o tempo de execução para 11 minutos. O aumento do total de pares encontrados foi possível devido ao tratamento dos casos especiais (RNs e alterações de nomes por mudança de estado civil), uso de fonética com os atributos BUSCABR_NOME e BUSCABR_MAE ao invés dos atributos originais NOME e NOME DA MÃE e também com o refinamento da regra para a comparação das DATAS DE NASCIMENTO (regra 6).

A redução no tempo de execução foi possível devido, principalmente, à implementação do algoritmo de Distância de Edição na própria aplicação. Com isso, evitou-se o chaveamento de contexto entre a aplicação e o SGDB. Outro fator relevante é a troca na ordem de validação das regras, fato este que fez com que os testes que mais descartam (ou confirmam) a duplicidade dos registros fossem executados primeiramente.

Tabela 4.5: Estatísticas da Estratégia 3

Nº de Blocos	Média de Registros por Bloco	Pares Possíveis	Pares Encontrados	% Sobre Possíveis	% Sobre Total	Tempo de Execução
408	23,56	650	632	97,23%	96,05%	11 min

Vale citar, que as três estratégias foram testadas no mesmo computador, com processador Intel Core 2 Duo de 1,83 GHZ e 2 GB de memória RAM. Cada umas das estratégias foi executada 7 vezes e o tempo de execução exibido nos resultados é a média dos tempos de execução, excluindo-se o maior e o menor tempo.

Os valores de 0,25 como parâmetro para distância de edição entre os NOMES DA MÃE e de 0,2 (que representa 1 caractere para uma data no formato dd/mm/aaaa) para a distância de edição entre as DATAS DE NASCIMENTO foram obtidos com testes sobre a base de dados de teste avaliada e estes foram os valores que apresentaram melhor qualidade para serem utilizados como limiares.

Para o processo de deduplicação optou-se pela Estratégia 3, visto que ela apresenta os melhores resultados tanto em porcentagem de pares duplicados encontrados, quanto em tempo de execução, conforme é visto nos gráficos da Figura 4.2 e da Figura 4.3.

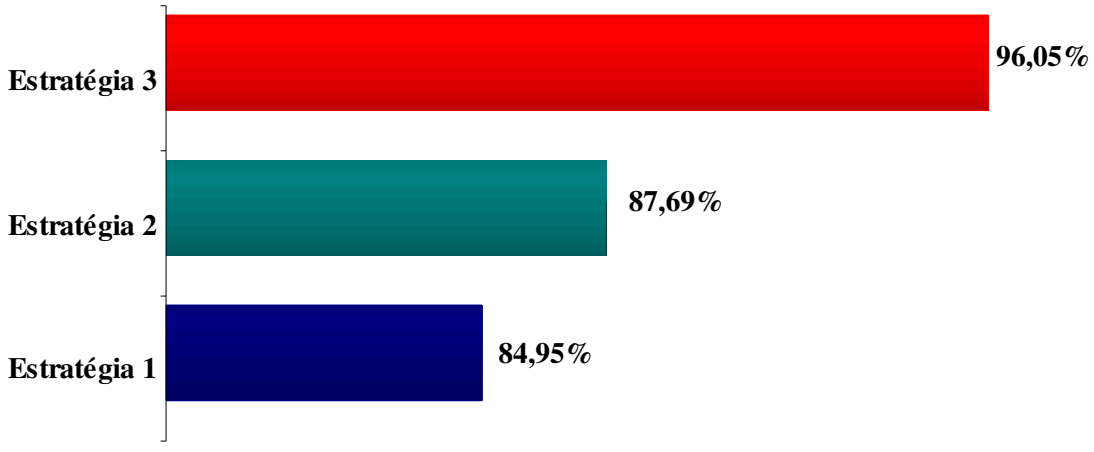


Figura 4.2: Porcentagem de pares de registros encontrados sobre o total de pares da base.

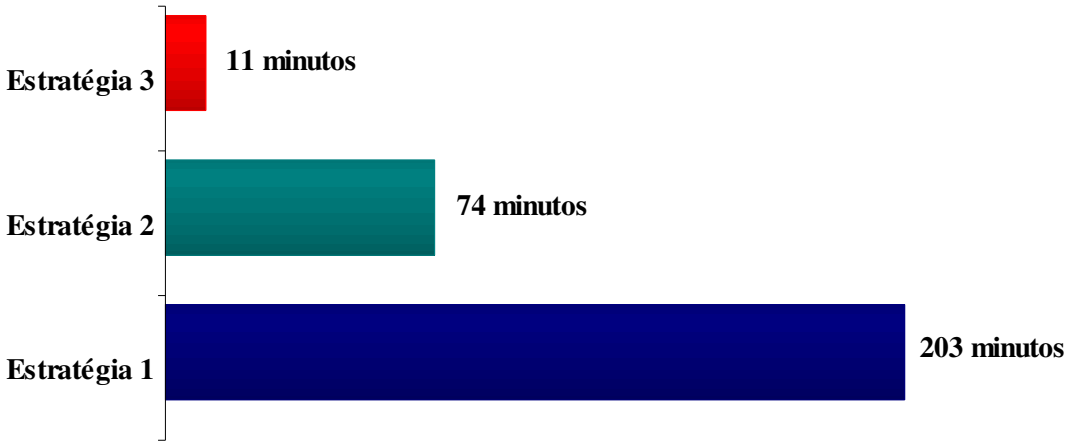


Figura 4.3: Comparativo do Tempo de Execução das Estratégias.

5 RESULTADOS

Após a definição da estratégia que seria utilizada na deduplicação da base de dados completa, foram executados os passos descritos abaixo:

A. Pré-processamento dos dados:

   Assim como na preparação da base de dados de testes, foi necessário efetuar a alteração dos atributos NOME e NOME DA MÃE para Case Insensitive e Accent Insensitive e a criação dos atributos BUSCABR_NOME, BUSCABR_MAE, BLOCO e FLAG. Estes passos foram executados em tempos irrisórios.

   Já o processo de popular os atributos BUSCABR_NOME, BUSCABR_MAE e BLOCO, demorou 47 minutos e 48 segundos.

B. Localização de registros com o atributo NOME DA MÃE inválido:

   Este processamento foi executado em 12 minutos e encontrou 108.296 registros, o que representa 3,81% da base de dados.

C. Adição dos registros com NOME DA MÃE inválido aos Blocos:

   Esta etapa demorou 98 minutos e conseguiu adicionar 56.312 registros a blocos válidos, restando então 51.984 registros (1,83% da base de dados) com NOME DA MÃE inválido que ficaram de fora do processo de deduplicação.

D. Sub-blocagem:

   Optou-se por dividir os blocos a fim de agilizar o processamento da deduplicação. A base de dados sem a sub-blocagem gerou blocos com mais de 100.000 registros, o que tornaria o tempo de execução do algoritmo de deduplicação inviável para conclusão deste trabalho. Assim, blocos com mais de 5.000 registros foram subdivididos utilizando o critério do mês e do dia de nascimento, conforme a necessidade, gerando assim, 4.343 blocos com uma média de 642,39 registros por bloco. A sub-blocagem foi executada em 35 minutos e 43 segundos.

E. Deduplicação:

A deduplicação, executada no mesmo PC onde foram feitas as análises da base de dados de testes, foi executada em 8 dias, 9 horas e 20 minutos e encontrou um total de 741.957 registros, o que representa 26,11% da base de dados.

F. Redução do Número de Comparações:

Utilizando os critérios de blocagem acima descritos, o número total de comparações foi reduzido de 4.038.142.388.140 para 1.803.802.961, ou seja, uma redução do número de comparações (RR) de 99,96%. Utilizando como base o tempo de execução da deduplicação utilizando a blocagem, o tempo estimado de deduplicação da base de dados completa, sem blocagem, seria de aproximadamente de 18.780 dias (o que representa mais do que 51 anos).

6 CONCLUSÃO

Este trabalho tem como objetivo, principalmente, a elaboração de uma estratégia de deduplicação de dados em um cadastro hospitalar de pacientes utilizando técnicas de blocagem associadas ao uso de algoritmos fonéticos, tornando possível uma limpeza da base de dados permitindo minimizar problemas como a segmentação do prontuário médico. Como resultado secundário, esta estratégia poderá ser utilizada como algoritmo para busca de registros similares durante o processo de cadastramento de novos pacientes.

Com esta deduplicação foram identificados os registros possivelmente duplicados, que serão analisados pelos responsáveis pelo cadastramento de pacientes. Estes registros serão excluídos, caso não exista um histórico de atendimentos de emergência, exames, consultas programadas, cirurgias entre outros, ou mesclados, caso exista um histórico relevante para o prontuário deste paciente.

Para realizar esta identificação foi necessário fazer a blocagem dos registros a fim de reduzir o número de comparações e, assim, reduzir também o tempo total do processamento da deduplicação. A escolha da chave de blocagem foi feita utilizando-se uma base de dados de teste com 10.000 registros previamente avaliados. Esta base de dados de teste foi gerada tentando ao máximo manter as características da base de dados completa, como o número de registros duplicados, tipos de dados incorretos e incompletos, utilizando conhecimento prévio dos Analistas responsáveis pelo cadastro de pacientes.

Após vários testes, chegou-se à chave de blocagem utilizando em conjunto os algoritmos fonéticos Soundex e BuscaBR sobre o nome da mãe do paciente, blocagem esta, que permitiu que fosse possível a identificação de 96,5% dos registros duplicados. A deduplicação aplicada na base de dados completa, com uso desta chave de blocagem, tornou possível a identificação de um total de 741.957 registros possivelmente duplicados. Este processo reduziu em 99,96% o número de comparações entre os registros e foi executado em 8 dias, 9 horas e 20 minutos.

Durante o trabalho foram identificadas algumas limitações, como a limitação de tempo; pois o ideal seria não precisar fazer uma sub-blocagem da base de dados completa para evitar que alguns registros duplicados ficassem em blocos diferentes. Outra dificuldade encontrada foi a de gerar uma base de dados de teste avaliada mais fiel à base de dados completa, já que a base de dados de teste foi planejada com aproximadamente 10% de registros duplicados e na base de dados completa foram encontrados 26,11% de registros possivelmente duplicados.

Como trabalhos futuros para melhorar o resultado e o desempenho da solução, destaca-se o uso de paralelismo (blocos diferentes podem ser analisados em paralelo) e uma adaptação do algoritmo para dar pesos diferentes para nomes comuns e para nomes

menos comuns (Maria da Silva comparado com Maria da Sillva não deveria ter o mesmo peso do que comparar Cipriana Schnetzer com Cipriana Schnetizer – ambos com apenas 1 caractere a mais).

Outra medida que deve ser adotada é uma melhoria na validação dos registros incluídos, como, por exemplo, impossibilitar o cadastramento de um paciente sem data de nascimento e também pedir uma confirmação extra do usuário para incluir um paciente com dados muito semelhantes a outro paciente existente.

Cabe ainda salientar que esta solução proposta pode ser utilizada para a deduplicação de qualquer base de dados de língua portuguesa (português brasileiro) com as devidas adaptações para atender as especificidades de cada base de dados.

REFERÊNCIAS

BAXTER, R.; CHRISTEN, P.; CHURCHES, T. **A comparison of fast blocking methods for record linkage.** ACM SIGKDD workshop on Data Cleaning, Record Linkage and Object Consolidation, Washington DC, 2003. p. 25-27.

CHRISTEN, P. **Towards Parameter-free Blocking for Scalable Record Linkage**. ANU Joint Computer Science Technical Report Series, Agosto/2007.

DE LUCENA, F. J. T. **Busca Fonética em Português do Brasil**, 2006.

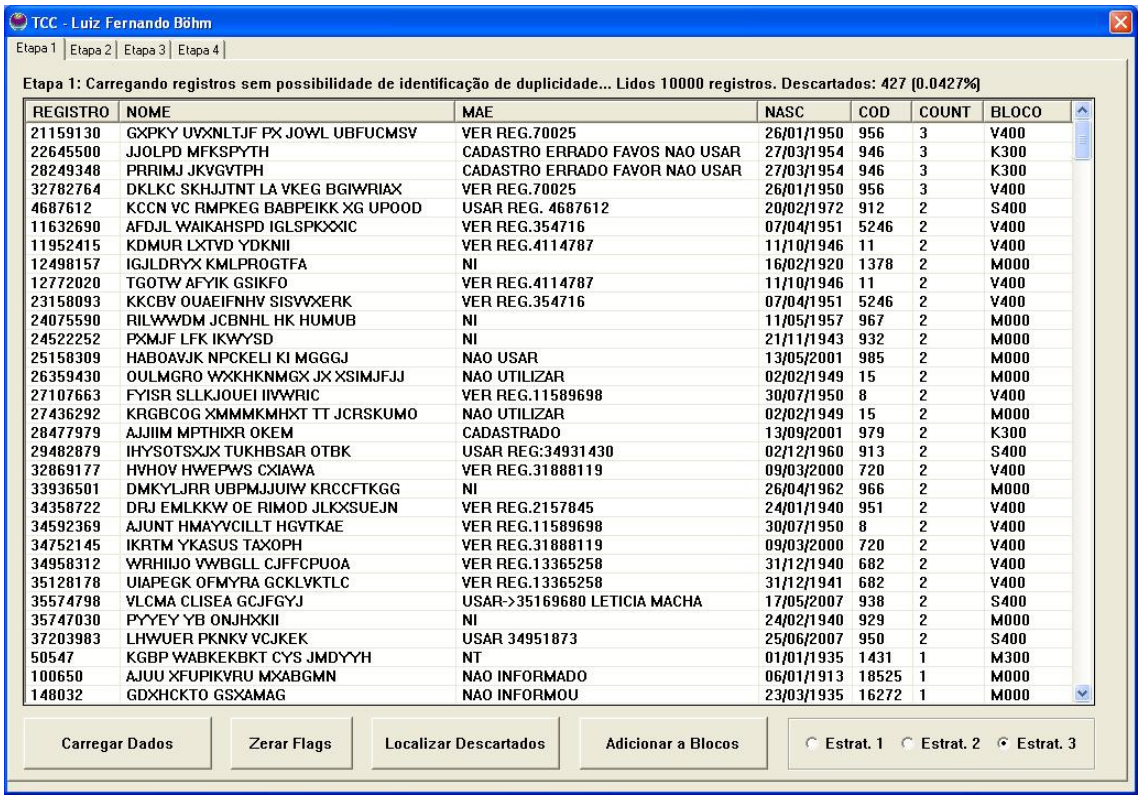
JIN, L.; LI, C.; MEHROTRA, S. **Efficient Record Linkage in Large Data Sets**, 2003.

LEVENSHTEIN, V. I. **Binary codes capable of correcting spurious insertions and deletions of ones. Problems of Information transmission**, Problems of Information Transmission v.1, n.1, p. 8–17, 1965.

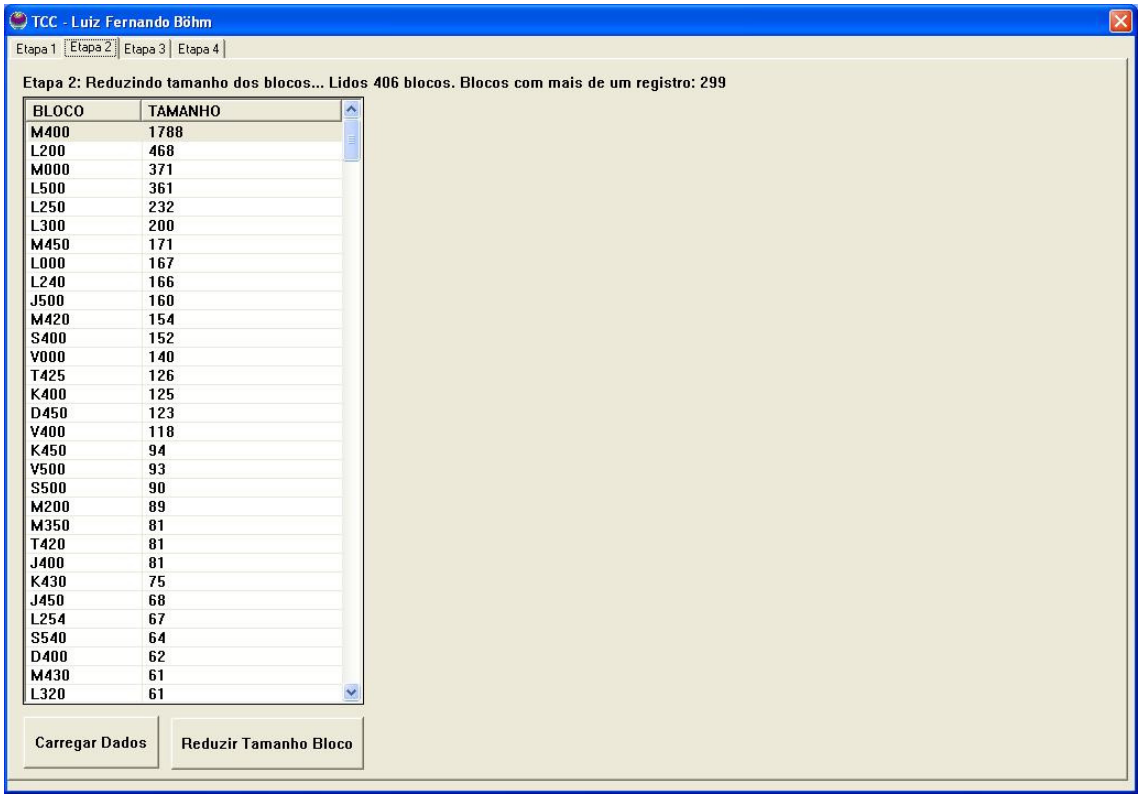
LIFANG, G.; BAXTER, R.; DEANE, V.; CHRIS, R. **Record Linkage: Current Practice and Future Directions**. Technical Report 03/83, CSIRO Mathematical and Information Sciences, April 2003.

RUSSELL, R.; ODELL, M. **Soundex Patent 01 261 167**, 1918.

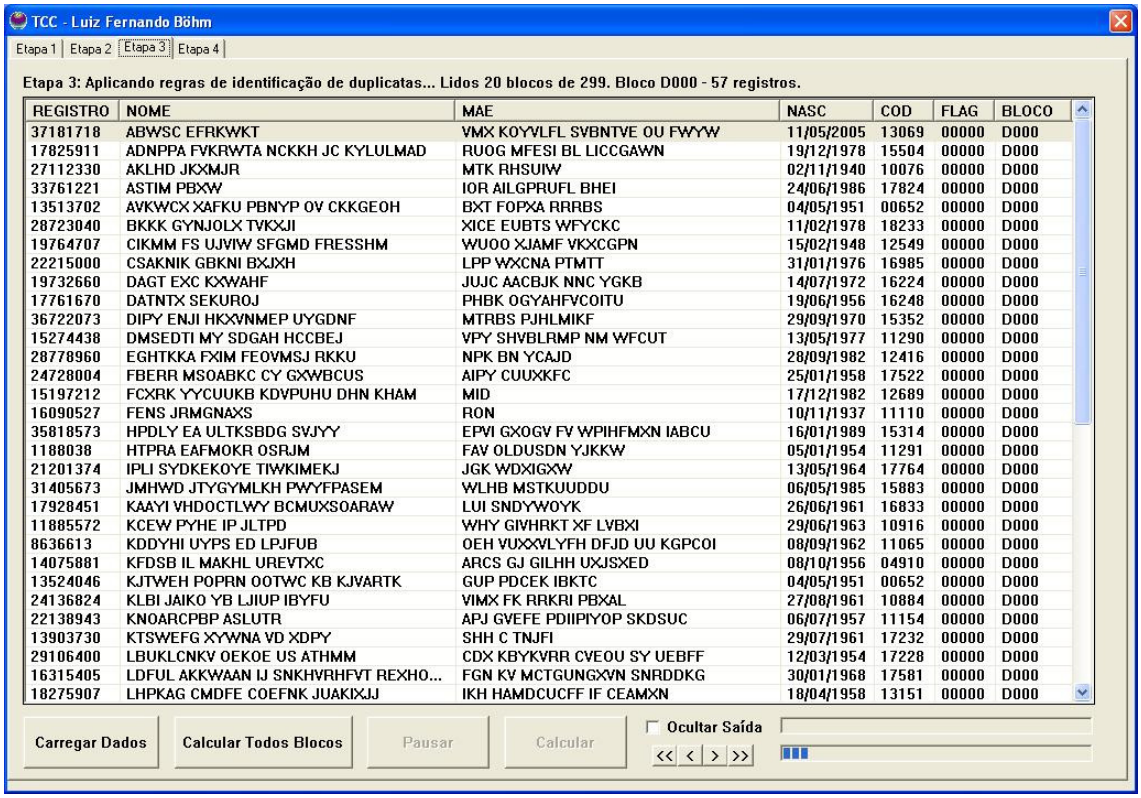
APÊNDICE – TELAS DO SISTEMA



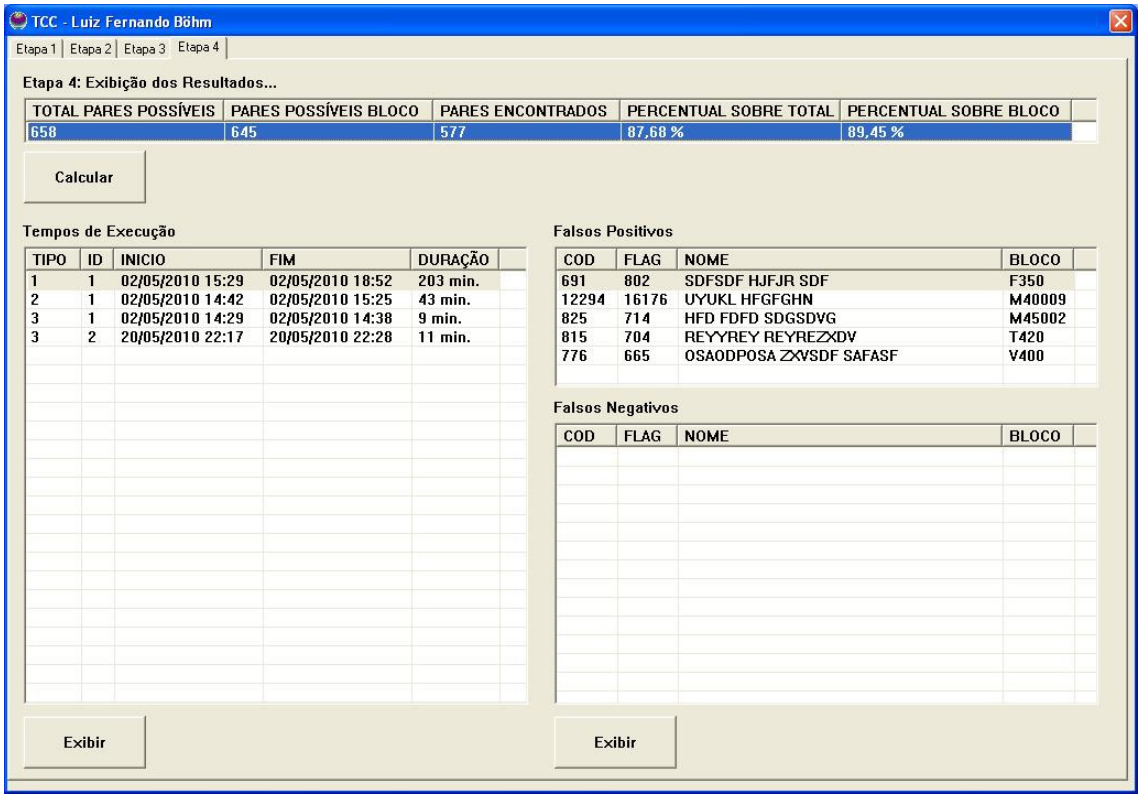
Etapa 1: Pré-Processamento dos Registros com Nome da Mãe inválido.



Etapa 2: Sub-Blocagem.



Etapa 3: Deduplicação dos Registros do Bloco.



Etapa 4: Exibição dos Resultados.