UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ, *CAMPUS* BANDEIRANTES.

SÉRGIO LUIS OTTÊNIO

PROTÓTIPO DE UMA FERRAMENTA PARA REALIZAÇÃO DA ETAPA DE PRÉ-PROCESSAMENTO NO PROCESSO DE DESCOBERTA DE CONHECIMENTO NÃO TRIVIAL EM BASE DE DADOS

BANDEIRANTES – PR
Maio 2009

UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ,
***CAMPUS* BANDEIRANTES.**

SÉRGIO LUIS OTTÊNIO

PROTÓTIPO DE UMA FERRAMENTA PARA REALIZAÇÃO DA ETAPA DE PRÉ-PROCESSAMENTO NO PROCESSO DE DESCOBERTA DE CONHECIMENTO NÃO TRIVIAL EM BASE DE DADOS

Monografia apresentada ao curso de Sistemas de Informação da Universidade Estadual do Norte do Paraná, campus Bandeirantes, para a obtenção do grau de Bacharel em Sistemas de Informação, orientado pelo Prof. Ms. Ricardo Gonçalves Coelho.

BANDEIRANTES – PR

2009

SÉRGIO LUIS OTTÊNIO

PROTÓTIPO DE UMA FERRAMENTA PARA REALIZAÇÃO DA ETAPA DE PRÉ-PROCESSAMENTO NO PROCESSO DE DESCOBERTA DE CONHECIMENTO NÃO TRIVIAL EM BASE DE DADOS

Monografia apresentada ao curso de Sistemas de Informação da Universidade Estadual do Norte do Paraná, campus Bandeirantes, para a obtenção do grau de Bacharel em Sistemas de Informação.

BANCA EXAMINADORA

_____

Prof. Ms. Ricardo Gonçalves Coelho
Orientador

_____

Prof. Ms. André Luís Andrade Menolli
Membro da banca examinadora

_____

Prof. Ms. Glauco Carlos Silva
Membro da banca examinadora

Bandeirantes, _____ de _____ 2009.

Aos meus pais que me ajudaram nesta grande conquista em minha vida, e todas as pessoas que contribuíram para a minha formação acadêmica.

AGRADECIMENTOS

Agradeço aos meus pais que não mediram esforços para que eu pudesse ter esta oportunidade de crescimento intelectual, e também de desenvolvimento como pessoa, durante estes anos que passei na Universidade. Ao meu orientador Prof. Ms. Ricardo Gonçalves Coelho, pela oportunidade de crescimento com o desenvolvimento deste trabalho, e pelas conversas durante este período. A todos os professores que contribuíram significativamente para minha formação, não somente com conhecimento em sala de aula, mas pelos conselhos de amigo, que contribuíram muito para minha formação. E por fim, agradeço a todos que me apoiaram e contribuíram de alguma forma para este momento.

RESUMO

O presente trabalho aborda o desenvolvimento de um protótipo de uma ferramenta para realização da etapa de pré-processamento no processo de descoberta de conhecimento não trivial em base de dados, preparando um novo conjunto de dados para a utilização na ferramenta. O processo de descoberta de conhecimento vem para garimpar informações com um grande potencial para a empresa, uma grande parte desses dados possui informações valiosas, como tendências e padrões que poderiam ser usados para melhorar as decisões de negócios, além de outras aplicações. A motivação para o crescimento desta área está ligada, principalmente, à existência de uma poderosa tecnologia para a realização da coleta, armazenamento e gerenciamento de grande quantidade de dados.

Palavras-chave: Pré-Processamento, KDD.

ABSTRACT

The present work deals the development of a prototype of a tool to perform the step of pre-precessing. In the non trivial knowlege in database, preparing a new set of data for the application Diffuse Multiobjective Genetic Algorithm for Knowledge Discovery. The process of Discovery of knowlege comes to gold mining information with great petencial for business, much of this data has valuable information, such as trends and patterns that could be used to improve business decisions, above of other applicatios. The motivation for the growth of this area is connected, mainly, the existence of a powerful technology to perform the collection, storing and managing large amounte of data.

Keyword: Pré-processing, KDD.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE SIGLAS

J2EE	Java 2 Enterprise Edition
J2ME	Java 2 Micro Edition
J2SE	Java 2 Standard Edition
KDD	Knowledge Discovery in Databases
UML	Unified Modeling Language
SQL	Structured Query Language

SUMÁRIO

1. INTRODUÇÃO

A grande quantidade de informação que se encontra nos bancos de dados de empresas informatizadas acaba ocultando informações valiosas, como tendências e padrões que poderiam ser usados para melhorar as decisões de negócios, além de outras aplicações.

O processo de descoberta de conhecimento vem para garimpar informações com um grande potencial para a empresa, que quase sempre não são notadas por esta. Encontrar padrões em um conjunto de dados não é uma tarefa fácil, para isso os dados devem seguir algumas etapas no processo de descoberta de conhecimento, fatores que aumentam a motivação para o crescimento desta área estão ligados, principalmente, à existência de uma poderosa tecnologia para a realização da coleta, armazenamento e gerenciamento de grande quantidade de dados.

O processo de descoberta de conhecimento inicia com a consolidação dos dados, no qual os dados são coletados de suas origens, e são colocados em uma base de dados independente, depois de obter a coleção de dados para a descoberta de conhecimento, é realizado o pré-processamento, onde é feito um tratamento nos dados, visando corrigir e excluir problemas que se encontram na base de dados. Após a coleta e a preparação dos dados, estes passam pela etapa de mineração de dados, onde são aplicados algoritmos que procuram por padrões que sejam relevantes dentro do conjunto de dados. Para finalizar o processo de descoberta de conhecimento é realizado o pós-processamento, no qual são analisados os padrões encontrados na etapa de mineração de dados, com a análise destes é gerado o conhecimento sobre o conjunto de informações trabalhado.

Nesse trabalho o foco principal será na etapa de Pré-processamento, no qual são utilizados alguns métodos e técnicas, que preparam e corrigem problemas nos dados, para que se possam obter melhores resultados na etapa de mineração de dados.

1.1 Objetivos

O objetivo do projeto é trabalhar na etapa de pré-processamento de dados, utilizando de métodos para que se possam ter dados com uma melhor qualidade a serem utilizados no processo de descoberta de conhecimento. Esta preparação dos dados é necessária, uma vez que se sabe que os dados que estão disponíveis, possuem alguns problemas como, dados com valores irregulares, ruídos, poluição, entre outros, que muitas vezes são gerados da união de dados de diferentes bases, ou de falhas no processo da criação, e outros fatores que contribuem para a existência destes.

Mesmo sabendo que os algoritmos utilizados na fase de mineração de dados, estão preparados para trabalharem com os dados nessa situação, pode-se obter resultados melhores uma vez que os problemas encontrados sejam removidos ou corrigidos antes do processo de mineração de dados.

Com um protótipo de uma ferramenta para o processo de pré-processamento espera-se que este processo seja feito de modo mais simplificado, para o usuário e facilite o trabalho na preparação dos dados para a etapa seguinte.

A criação deste novo conjunto de dados tem como objetivo a preparação para a aplicação do Algoritmo Genético Difuso Multiobjetivo para Descoberta de Conhecimento na etapa de mineração de dados, algoritmo este desenvolvido por Coelho (2004), neste conjunto de dados deve ser realizando a discretização dos atributos e realizando um tratamento nos dados, para que se tenha uma base preparada e voltada para o algoritmo definido anteriormente.

Para que se possa tornar esse processo mais simples, algumas tarefas são implementadas, com o objetivo de solucionar alguns problemas encontrados nas bases de dados a serem trabalhadas, e facilitar a seleção dos dados que são relevantes para o processo de descoberta de conhecimento.

1.2 Justificativas

O trabalho realizado com os dados na etapa de pré-processamento tem um foco em facilitar e melhorar a qualidade dos resultados obtidos na etapa seguinte de mineração de dados na descoberta de conhecimento.

Produzindo um conjunto de dados livres de problemas relacionados à qualidade e entre outros, é possível produzir resultados mais precisos na etapa de mineração de dados, uma vez que a qualidade dos resultados obtidos está ligada diretamente com a qualidade dos dados trabalhados, e isso faz com que se tenha um ganho nos resultados finas do processo de descoberta de conhecimento.

A preparação de um novo conjunto de dados para a aplicação do Algoritmo Genético Difuso Multiobjetivo para Descoberta de Conhecimento, torna mais rápido este processo, uma vez que estes dados serão moldados para atender as especificações do algoritmo definido.

Com uma precisão e qualidade maior obtido com o a descoberta de conhecimento, proporciona uma vantagem nos trabalhos realizados que se utilizar desta técnica, pois terá uma resposta mais rápida e objetiva, para atender as necessidades específicas de cada caso que se trabalhe com a descoberta de conhecimento.

1.3 Organização do trabalho

A estrutura do trabalho se apresenta distribuída da seguinte maneira: no capítulo 2 são apresentados os conceitos relacionados ao Processo de Descoberta de Conhecimento e suas etapas. No capítulo 3 são apresentados a descrição do sistema desenvolvido neste trabalho. Já no capítulo 4 são apresentadas as ferramentas e metodologias empregadas no desenvolvimento deste do trabalho. E por fim no capítulo 5 são apresentados os resultados e as conclusões do trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão mostrados alguns conceitos sobre o Processo de Descoberta de Conhecimento e suas etapas, o pré-processamento que é a principal etapa que será trabalhada no projeto, terá mais detalhes em um tópico onde serão apresentados métodos e técnicas muito utilizadas para o tratamento dos dados antes da mineração de dados.

2.1 Descoberta de Conhecimento

"KDD (Knowledge Discovery in Databases) é o processo não trivial de identificação, a partir de dados, de padrões que sejam válidos, novos, potencialmente úteis e compreensíveis" (Fayyad, 1996 apud ROMÃO, 2002. p. 41).

De acordo com Romão (2002), um sistema de KDD pode gerar um conhecimento eficiente, mais esse conhecimento pode não ser interessante para o usuário, por isso é recomendável utilizar algum método para medir o grau de interesse sobre o conhecimento descoberto.

Existe uma grande quantidade de métodos utilizados para a descoberta de conhecimento, e estes produzem um grande número de regras, a maioria dessas regras geradas é irrelevante para o usuário, e isso acaba dificultando a descoberta de regras que sejam relevantes. (Romão, 2002)

Métodos objetivos ajudam a descoberta de conhecimento relevante, que em geral trabalham de forma autônoma, e métodos subjetivos os quais levam em conta o conhecimento prévio do usuário sobre o domínio da aplicação. (Romão, 2002)

De acordo com Cruzes et al (2005), para cada uma das atividades do processo de classificação existem aplicações específicas para o uso do conhecimento prévio do minerador. Com isso é possível considerar como relevante alguns tipos de conhecimento que facilitam ou melhoram a execução das atividades do processo de descoberta de conhecimento:

- Conhecimento sobre Domínio dos Dados – conhecimento do minerador sobre o domínio dos dados que estão sendo explorados e sobre os quais se deseja criar um modelo de classificação.

- Conhecimento sobre Exploração de Dados - conhecimento do minerador sobre mecanismos, ferramentas e técnicas de exploração de dados.
- Conhecimento sobre Tarefa de Classificação: conhecimento do minerador sobre algoritmos de classificação e mais especificamente sobre o algoritmo usado no processo de classificação
- Conhecimento sobre Processo de Coleta dos Dados: conhecimento do minerador sobre mecanismos utilizados para a coleta dos dados em questão, objetivos da coleta, usos prévios dos dados, informações sobre qualidade dos dados, etc.

2.2. Principais etapas da Descoberta de Conhecimento

Segundo Fayyad et al. (1997 *apud* Coelho, 2004), o KDD é um processo não trivial de identificação de padrões de dados, e também um processo interativo e iterativo composto por numerosas etapas que podem ser resumidas como:

- Desenvolver um conhecimento sobre o domínio da aplicação: inclui conhecimento anterior pertinente e identificar os objetivos para a aplicação;
- Identificar o conjunto de dados a ser trabalhado: definir qual será o foco para realizar a descoberta de conhecimento;
- Filtrar os dados e pré-processamento: Manter somente os dados que serão úteis para o processo, e definir padrões para os dados a serem utilizados.
- Redução de dados e projeção: descoberta de características úteis para representar os dados, encontrar representações constantes para os dados;
- Determinar a função de mineração de dados: decisão do propósito do modelo derivado através de um algoritmo de mineração de dados;
- Escolha do algoritmo de mineração de dados: Definir os algoritmos a serem utilizados e realizar a escolha dos modelos utilizados para a procura de padrões de dados;
- Mineração de dados: procurar padrões interessantes em uma forma de representação;
- Interpretação: visualização dos padrões extraídos, remoção de padrões redundantes ou irrelevantes, e tradução dos úteis em termos entendíveis pelos usuários;

- Uso do conhecimento descoberto;

Pode se resumir em quatro partes as etapas nas quais o analista se envolve na descoberta de conhecimento, que são: Consolidação de Dados, Pré-processamento, Mineração de dados, Pós-processamento.

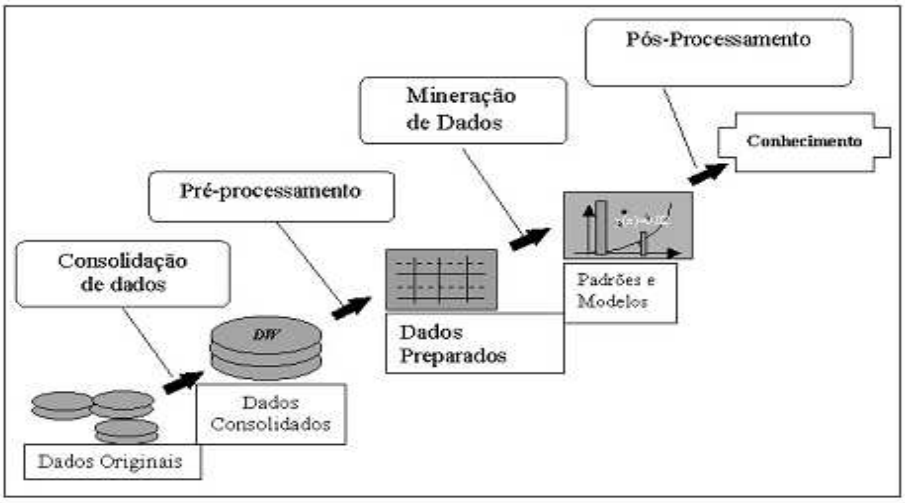

Figura 1 - Etapas do processo de descoberta de conhecimento.

Fonte: (Coelho, 2004)

A figura acima apresenta as etapas que estão envolvidas no processo de descoberta de conhecimento, no qual se tem os dados em suas fontes e estados originais, estes são coletados na criação dos Data Warehouse, que são uma coleção de dados que darão suporte para a realização de descoberta de conhecimento. Após a coleta e destes dados é realizado uma preparação para que se possam ter dados com qualidade para a etapa de mineração de dados, que procura por padrões que sejam relevantes, e estes padrões são analisados para que se possa ser extraído o conhecimento.

2.2.1 Consolidação de dados

Pelo fato de que os dados analisados não serem necessariamente de uma única fonte, é preciso fazer um refinamento deste, para que se possam eliminar dados que tenham valores nulos, valores já existentes, para que se possa ter uma base dados pronta para ser estudada.

Segundo KIMBALL et al. (1998 *apud* Coelho, 2004), para auxiliar nesse processo, utiliza-se o Data Warehouse (DW), que consiste em criar bases de dados independentes para que se possa trabalhar com a base de dados para facilitar o uso da informação.

Com o uso do DW é possível obter um ganho considerável em relação a custo/beneficio, tempo e produtividade, deixando claro que o é necessário que os dados já tenham sido filtrados para se ter uma base de qualidade, já que a qualidade do conhecimento obtido depende diretamente da qualidade dos dados utilizados.

2.2.2 Pré-processamento

No pré-processamento se compreende as funções que se relacionam a captação, à organização e ao tratamento de dados a serem trabalhados, esta etapa é realizada visando preparar os dados para os algoritmos da etapa de mineração de dados.

Segundo Batista (2003), é na fase de pré-processamento que se aprimora a qualidade dos dados coletados. Freqüentemente, os dados coletados apresentam diversos problemas, tais como grande quantidade de valores desconhecidos, ruído, atributos de baixo valor preditivo, grande desproporção entre o número de exemplos de cada classe.

Como Batista (2003), mostra que o pré-processamento é uma atividade semi-automática, já que está ligada com a interferência direta do analista dos dados para identificar problemas existentes com os dados.

As bases de dados costumam conter diversos atributos irrelevantes que, se não passarem por um processo de filtragem para serem removidos, podem tornar o processo de mineração de dados difícil. Por isso Romão (2002) afirma que a seleção de atributos relevantes é um dos maiores desafios para realizar uma tarefa de MD específica.

2.2.3 Mineração de dados

É na etapa de mineração de dados que é determinado qual algoritmo será utilizado na descoberta de conhecimento.

Durante a etapa de mineração de fatos, é realizada a busca efetiva por conhecimentos úteis no contexto da aplicação de KDD. Pode-se considerar a principal etapa do processo de KDD. Na mineração de dados, são definidas as técnicas e os algoritmos a serem utilizados no problema em questão, através de Redes Neurais, Algoritmos Genéricos, modelos Estatísticos e probabilísticos (Goldschmidt e Passos, 2005 *apud* Boente, 2007).

O uso de ferramentas para a descoberta de conhecimento em bases de dados vem integrar a tecnologia e a aprendizagem organizacional em busca da gestão estratégica do conhecimento. O processo de KDD não se resume apenas a garimpagem de dados, mas este é considerado o processo elementar para geração do conhecimento gerado a partir de Bases de Dados. Na verdade, além da mineração de dados existem ainda o pré-processamento e o pós-processamento (Boente, 2006 *apud* Boente, 2007).

2.2.4 Pós-processamento

Esta etapa tem como objetivo tratar os resultados obtidos dos algoritmos na fase de mineração, para que se possa ter uma melhor compreensão dos dados obtidos, assim acelerar a interpretação destes.

Segundo Fayyad et al (1996 apud Rezende, 2005), o grande objetivo do processo de Extração de Conhecimento de Base de Dados é encontrar conhecimento a partir de um conjunto de dados selecionado e tratado, e utilizar o resultado deste processo para ajudar na tomadas de decisão. Portanto, um requisito importante para o usuário final desse processo, é que se tenha o conhecimento descoberto em uma linguagem que seja de fácil compreensão, útil e interessante, de forma que este forneça um suporte ao usuário final do processo de tomada de decisão.

Segundo Romão (2002), O pós-processamento é tem como principal finalidade avaliar o processo de descoberta, simplificar os resultados para melhorar a compreensão e/ou selecionar conhecimento descoberto que seja mais relevante. Quando são geradas muitas regras, é importante remover algumas regras e/ou condições para que se possa ter uma melhor compreensão do conhecimento extraído.

Segundo Boente (2007), fica por responsabilidade de especialista em KDD e o especialista de domínio da aplicação avaliar os resultados obtidos e definirem novas alternativas de investigação dos dados.

Complementando a etapa de pós-processamento, são realizadas as seguintes operações: Simplificação do Modelo de Conhecimento, Transformações do Modelo de Conhecimento e Organização e Apresentação dos Resultados. (Boente, 2007)

2.3 Pré-processamento

O pré-processamento de dados é uma das etapas que compõem o KDD, e esta etapa é vista como um ponto onde é preciso ter uma grande quantidade de conhecimentos sobre o domínio trabalhado, para que se possam identificar dados de qualidade que serão utilizados no processo de mineração de dados.

Segundo Rezende (2005), esta etapa é responsável pela a aplicação de métodos para tratamento, limpeza e redução do volume de dados, para se ter mais qualidade nos dados, antes de iniciar a etapa de extração de padrões.

Uma visão compartilhada por muitos pesquisadores mostra que os dados coletados diretamente de banco de dados, são geralmente de má qualidade, segundo Batista (2003), mesmo com a utilização de algoritmos preparados para trabalhar com dados em tais situações, os resultados obtidos seriam melhores caso essas irregularidades encontradas com os dados fossem removidas ou corrigidas.

De acordo com Rabelo (2007), nesta etapa do processo de descoberta de conhecimento, são utilizados métodos para realizar o tratamento de distorções, ausência de dados ou, simplesmente, é realizada uma reorganização das informações.

Segundo Carvalho at al (2003), no pré-processamento são realizadas atividades que visam gerar uma representação conveniente para os algoritmos que serão utilizados no processo de mineração de dados, a partir da base de dados. Nesta etapa também é realizada a seleção de atributos relevantes, que podem ser automática e/ou manual, amostragem, transformações de representação, etc;

Segundo Zhang at al (2003, apud Soares, 2007), a preparação realizada nesta etapa da descoberta de conhecimento, é de grande importância,

sabendo que quando esta não é realizada da melhor maneira, tem interferência direta na qualidade dos resultados obtidos no processo no processo de descoberta de padrões ocultos. Este processo é responsável por 80% do total de esforços de tratamento de dados.

Soares (2007), diz que grande parte dos esforços para a melhora na descoberta de conhecimento nas ultimas décadas, está na tarefa de mineração de dados, e alerta para que se tenha uma equivalência em estudos com foco nas tarefas de pré-processamento de dados, uma vez que esta é extremamente importante para a obtenção de padrões de qualidade oculto nos dados.

De acordo com Batista (2003) o pré-processamento de dados é um processo semi-automático, e nesta fase depende da capacidade do analista de dados em identificar os problemas presentes nos dados, além da natureza desses problemas, e utilizar os métodos definidos por ele que sejam mais apropriados para solucionar cada um dos problemas encontrados.

De acordo com Rezende (2005), é importante salientar que a execução das transformações deve ser guiada pelos objetivos do processo de extração a fim de que o conjunto de dados gerado apresente as características necessárias para que os objetivos sejam cumpridos.

2.3.1 Pré-Processamento de Dados Fortemente Dependente de Conhecimento de Domínio

> As tarefas de pré-processamento de dados fortemente dependentes de domínio vêm sendo estudadas há alguns anos. Essas tarefas são muito semelhantes às tarefas que são encontradas no processo de carga de um Data Warehouse. (Batista, 2003. p. 41)

Batista (2003), nos mostra que como os dados do Data Warehouse normalmente tem sua origem de sistemas transacionais, freqüentemente apresentam diversos problemas, visando o auxilio das soluções para esses problemas foi criada uma classe de ferramentas chamada ETL (Extraction, Transformation and Load), algumas das principais tarefas são:

2.3.1.1 Identificação de Inconsistência

Segundo Batista (2003), muitos programadores diferentes e programas implementados em linguagens diferentes podem gerar arquivos de formatos diferentes, informações podem estar duplicadas em diversos lugares, podem estar com mesmo rótulo e dados diferentes ou mesmos dados e rótulos diferentes, isso gera uma inconsistência nos dados.

> Inconsistências podem facilmente acontecer no processo de integração de dados, onde tabelas com atributos com nomes diferentes devem se transformar em uma única. Dados inconsistentes também podem ser gerados por transações, se as restrições de integridade das tabelas de um banco de dados não tiverem sido impostas da forma correta. (Soares, 2007. p. 26)

De acordo com Rezende (2005), o processo de unificação dos dados é necessário para que se forme uma única fonte de dados no formato atributo-valor, que será utilizado como entrada para o algoritmo de Extração de Padrões, e com os dados em uma única fonte é possível realizar essa verificação de inconsistência.

2.3.1.2 Identificação de Poluição e Ruídos

Batista (2003) define poluição de dados como a presença de dados distorcidos, os quais não representam valores verdadeiros. A tentativa do usuário em utilizar o sistema além das suas funcionalidades ou em outras vezes, a insistência em entrar com dados incorretos podem gerar poluição em uma base de dados.

Santos (2007) definem ruídos em base de dados, como sendo erros randônicos que acontecem no conteúdo dos atributos.

Alguns métodos utilizados para tratar esse problema são: Binning, Regressão, Agrupamento.

De acordo com Santos (2007), no método *binning* os valores dos atributos são ordenados, depois eles são divididos em blocos com o mesmo tamanho e ajustar os valores dos blocos.

Os ajustes realizados no método de regressão são feitos a partir do resultado de uma função de regressão linear. Regressão linear é definido por Anderson et al. (2005 *apud* Santos, 2007) como um processo estatístico usado para

desenvolver uma equação que mostre a relação existente entre duas ou mais variáveis.

Batista et al (2000, apud Rezend, 2005), nos mostra que o resultado do processo de extração possivelmente será utilizado em um processo de tomada de decisão, de acordo com isso, a qualidade dos dados é um fator extremamente importante, já que qualidade nos dados reflete em qualidade nos resultados. Por isso, técnicas de limpeza devem ser aplicadas aos dados a fim de garantir sua qualidade.

2.3.1.3 Verificação de Integridade

Segundo Batista (2003) para verificar a integridade dos dados é preciso realizar uma análise das relações permitidas entre os atributos, para que se possa ter a faixa de valores validos a serem analisados. Mais existe um caso especial na verificação de integridade, que são os casos de identificação de casos extremos.

> Casos extremos são casos em que a combinação dos valores é válida, pois os atributos estão dentro de faixas de valores aceitáveis, entretanto a combinação dos valores dos atributos é muito improvável. (Batista, 2003. p. 43)

2.3.1.4 Identificação de Atributos Duplicados e Redundantes

Esse tipo de problema ocorre quando existe dados em uma tabela que contenha as mesmas informações, um exemplo pode ser visto da seguinte forma; Em uma tabela de vendas se tem os seguintes campos: *preço_unitário*, *quantidade* e *preço_total*, isso faz com que se repita uma informação que pode ser extraída da tabela sem a necessidade de um determinado campo, que no caso vem a ser o *preço_total.*

Segundo Romão (2002), a diversas razões que contribuem para a redundância de dados, tais como: objetivando o aumentar o desempenho do sistema ou devido à integração de dados de fontes diferentes.

Para Batista (2003), o principal prejuízo que se tem com o problema da redundância de dados para a maioria dos algoritmos utilizados na fase de mineração de dados é um aumento no tempo de processamento.

Segundo Batista (2003), existe métodos de seleção de atributos, que podem ser utilizados para tentar identificar e remover os atributos redundantes quando estes não forem solucionados durante a fase de coleta de dados.

2.3.1.5 Defaults

Este tipo de problema ocorre devido permissão que muitos gerenciadores de banco de dados fornecem para a criação de valores defaults para alguns atributos.

Segundo Batista (2003), isso pode causar confusão caso o analista de dados não estiver informado a respeito desses dados definidos como default, valores default podem ser perigosos quando a intenção do está voltada em uma análise predativa.

“Um valor default pode estar ligado condicionalmente a outros atributos, o que pode criar padrões significantes à primeira vista. Na realidade, tais valores default condicionais simplesmente representam falta de informação.” (Batista, 2003. p. 43)

2.3.2 Pré-Processamento de Dados Fracamente Dependente de Conhecimento de Domínio

As tarefas realizadas nesta etapa têm a característica de serem tipicamente solucionadas através de métodos que extraem as informações necessárias para tratar os problemas do próprio conjunto de dados de acordo com Batista (2003).

2.3.2.1 Tratamento de Valores Desconhecidos

O tratamento de valores desconhecido é um problema muito comum na fase de pré-processamento, as técnicas utilizadas para resolver este tipo de problema geralmente são simples, um exemplo pode ser a substituição dos valores desconhecidos pela media ou moda do atributo, mais técnicas mais elaboradas vem sendo implementadas e avaliadas experimentalmente, segundo Batisata (2003).

A distribuição dos valores desconhecidos é outra opção para se trabalhar com este problema.

> Valores desconhecidos dispostos aleatoriamente nos dados podem ser considerados um problema menos sério do que quando esses valores não estão aleatoriamente distribuídos. Por outro lado, valores desconhecidos distribuídos não aleatoriamente são um problema sério independente de quão poucos existam, uma vez que esses valores podem afetar a generabilidade dos resultados. Neste caso, torna-se necessário utilizar algum método para estimar e substituir os valores desconhecidos. (Gustavo et al, 2001. p. 3)

2.3.2.2 Identificação e Descrição de Valores Extremos

Este tipo de problema ocorre quando se têm dados que fogem do padrão dos demais dados que se encontram na base a ser trabalhada.

Segundo Santos (2007), o método de agrupamento, pode ser utilizado para detectar os elementos que estão fora dos limites, os valores similares são organizados em grupos, os valores que ficam distantes da média de seu grupo são considerados discrepantes.

Batista (2002) diz que é preciso ficar atento com valores extremos, pois estes podem parecer a princípio que são dados que não possuem relevância dentro da base analisada, no entanto, algumas vezes esses dados com valores extremos podem representar as informações mais interessantes, a qual o analista de dados esteja procurando.

2.3.2.3 Construção de Atributos

Segundo Soares (2007), para se obter uma melhora considerável no processo de descoberta de conhecimento, pode se realizar a consolidação dos dados contidos em tabelas, uma das tarefas importantes da etapa de pré-processamento é a construção de atributos, onde se cria novas colunas na tabela, refletindo alguma transformação dos dados existentes nas tabelas de um conjunto de dados.

Atributos fracamente, indiretamente ou condicionalmente relevantes podem ser individualmente inadequados, entretanto, esses atributos podem ser convenientemente combinados gerando novos atributos que podem mostrar-se altamente representativos para a descrição de um conceito. (Michalski, 1978; Bloedorn & Michalski, 1998 apud Batista, 2003)

“Construção de atributos é o processo de composição de atributos ditos primitivos, produzindo-se novos atributos possivelmente relevantes para a descrição de um conceito.” (Batista, 2003. p. 46)

Outro método que pode ser utilizado na construção de atributos é a utilização da indução construtiva, na qual o novo atributo é criado tomando como base o valor de outros atributos. Caso os atributos originais utilizados na construção do novo atributo não estejam presentes em um novo modelo, eles podem ser descartados, reduzindo assim o número de atributos. Um fator importante relacionado à utilização de indução construtiva, esta ligado ao resultado final, podendo aumentar consideravelmente a qualidade do conhecimento extraído (Lee, 2000 apud Rezende, 2005)

2.3.2.4 Seleção de Atributos Relevantes

Para que se possa ter uma eficiente aplicação das técnicas de mineração de dados, antes é necessário realizar uma preparação destes dados, e um dos maiores desafios é realizar a seleção de atributos relevantes.

Wrappers e Processo por filtro

Romão (2002), mostra duas das principais abordagens utilizadas para este fim, que são: Processo Envoltório (Wrappers) e Processo por Filtro.

> A abordagem wrappers consiste em selecionar um subconjunto de atributos e medir a precisão do classificador induzido sobre esse subconjunto de atributos. É realizada uma busca pelo subconjunto que gera o classificador com menor erro. Essa busca avalia cada subconjunto candidato, até que o critério de parada, relacionado com a precisão do classificador induzido, seja satisfeito. (Batista, 2003. p. 45)

Para Romão (2002), as técnicas do tipo envoltório geralmente são mais efetivas, comparadas com as técnicas do tipo filtro, seus resultados tem uma taxa de erro de classificação menor.

Segundo Bala et al. (1995 *apud* Romão, 2002), técnicas utilizadas para realizar a seleção de atributos do tipo envoltório, demandam de inúmeras execuções do algoritmo utilizado na mineração de dados, este grande número de execuções geralmente consomem mais tempo de processamento.

De acordo com Soares (2007), no processo por filtro, de acordo com alguns critérios os atributos são selecionados e utilizados na etapa de mineração de dados, sem levar em consideração o algoritmo de classificação que será aplicado aos atributos selecionados.

Romão (2002) defende que as técnicas do tipo filtro, normalmente são mais eficientes, já que consomem um menor tempo de processamento.

Concordando com o ponto de vista de Romão, Soares (2007) diz que:

> A abordagem wrapper oferece a clara vantagem de gerar um subconjunto de atributos que podem aumentar significativamente a precisão dos algoritmos de mineração de dados a serem executados sobre o conjunto de dados. Todavia, esta abordagem apresenta duas significativas desvantagens: a de ser muito mais lenta do que o método filter, já que a busca pelo melhor subconjunto de atributos é um processo que demanda muito tempo de processamento. Além disso, a melhor configuração de atributos para um dado algoritmo de classificação pode não ser tão boa para um outro classificador, o que faz com que a seleção de atributos feita pelo modelo wrapper seja dependente do algoritmo utilizado. (Soares, 2007. p. 32)

A Figura 2 esquematiza graficamente as duas abordagens nesta é mostrado as etapas que compõem as abordagens Wrapper e processo de Filtro, como se pode ver na abordagem Wrapper, é iniciado com a entrada do conjunto completo de atributos, com base nesse conjunto completo são gerados sub-conjuntos de atributos candidatos, em cima desses conjuntos são aplicados algoritmos que realizam a classificação destes, após essa classificação e feita uma avaliação do desempenho obtido pelo algoritmo. Caso o resultado apresentado sejam os finais o processo é interrompido e fornecido um subconjunto de atributos, senão um ciclo é iniciado, e o processo volta para a etapa de geração de sub-conjunto de atributos, e assim é feito até chegar ao resulto esperado.

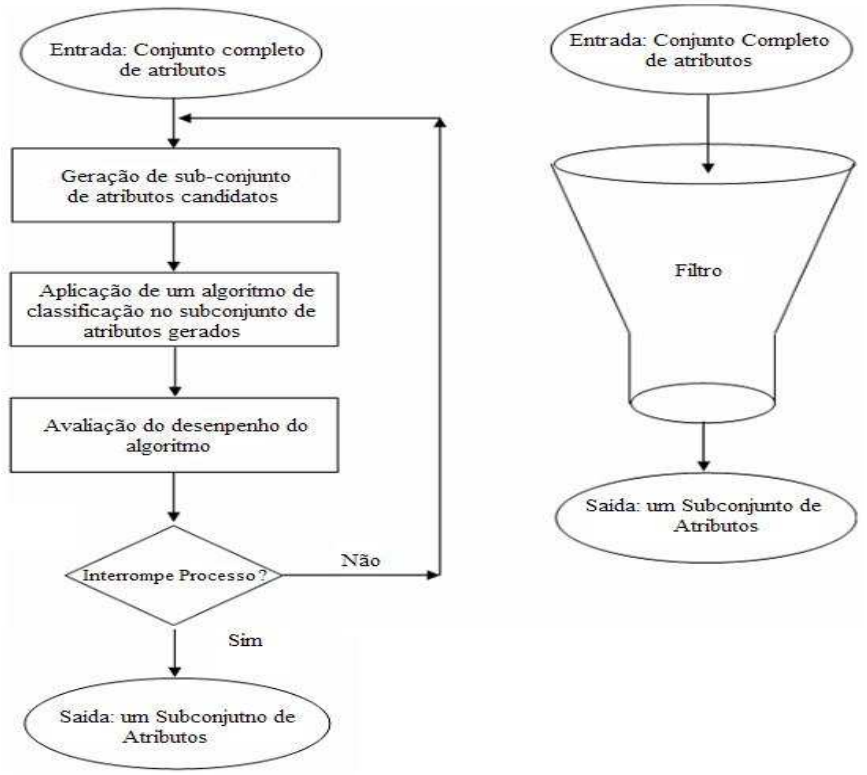

Figura 2 - As abordagens filter e wrapper para a seleção de atributos.
Fonte: (Soares, 2007)

Na abordagem de processo por filtro, é iniciado com a entrada de um conjunto completo de atributos, que passam por um filtro que ao seu final fornece um subconjunto de atributos.

Análise de Componentes Principais

É uma técnica bastante utiliza no processo de seleção de atributos segundo Soares (2007), já que esta é defina por ele como sendo um método que oferece uma boa estabilidade e praticidade no seu uso.

Segundo Smith (2002 apud Soares, 2007), a Análise de Componentes Principais (PCA – Principal Component Analysis), é uma técnica estatística que é muito utilizada em várias áreas do conhecimento, tais como reconhecimento e compressão de imagens, e que reflete a aplicação direta dos conceitos de sistemas de base em Álgebra Linear.

2.3.3 Atributos Categóricos e Atributos Contínuos

Dentro das regras de produção encontram-se atributos categóricos e também atributos contínuos.

Segundo Coelho (2004), com os atributos contínuos pode ser realizado um processo para fuzzificar estes, esse processo faz com que se tenham atributos com valores mais adequados a linguagem natural tais como: alto, médio e baixo.

> O processo de “fuzzificação” consiste na transformação de valores discretos para valores difusos. Nesse processo calcula-se o grau de pertinência que um valor discreto possui em relação a um conjunto difuso. Um valor discreto pode pertencer a vários conjuntos difusos, com graus de pertinência distintos em cada um deles. O cálculo do grau de pertinência é feito com o uso de funções de pertinência trapezoidal. (Coelho, 2004)

Essa transformação não é possível de ser realizada com atributos categóricos, pois estes já assumem um valor categórico especifico de seu domínio original, como pode ser visto com o atributo SEXO, que somente pode assumir dois valores, Masculino ou Feminino.

2.3.4 Seleção e Redução de Dados

Segundo Rezende (2005), em virtude das restrições de espaço em memória ou tempo de processamento, o número de exemplos e de atributos disponíveis para análise pode acabar inviabilizar a utilização de algoritmos de Extração de Padrões, uma vez que sempre procura-se obter resultados com maior qualidade e com custo e tempo menor.

Weiss & Indurkhya (1998, apud Rezende, 2005) propõem uma solução para este problema, segundo o autor, pode ser necessária a aplicação de métodos para a redução dos dados antes de iniciar a busca pelos padrões. Esta redução pode ser feita de três:

- Redução do número de exemplos;
- Redução do número de atributos; atributo!redução
- Redução do número de valores de um atributo.

A redução do número de exemplos deve ser feita a fim de manter as características do conjunto de dados original, isto é, por meio da geração de

amostras representativas dos dados (Glymour, Madigan, & Smyth 1997, apud Rezende, 2005).

A abordagem mais utilizada para redução do número de exemplos segundo Weiss & Indurkhya (1998, apud Rezende, 2005) é a abordagem de amostragem aleatória, pois este método tende a produzir amostras representativas.

Caso as amostragens trabalhadas não forem representativas, ou se a quantidade desses exemplos for insuficiente para caracterizar os padrões embutidos nos dados, isso pode fazer com que os modelos encontrados não possam representar a realidade, não tendo, portanto, valor. Outro problema que pode surgir é referente à quantidade relativamente pequena de exemplos, pode ocorrer *overfitting*, isto é, o modelo gerado pode "decorar" os dados do conjunto de treinamento, não se adequando para utilização com novos exemplos. (Fayyad, Piatetsky-Shapiro, & Smyth, 1996 apud Rezende, 2005)

Rezende (2005), diz que a redução do número de atributos pode ser utilizada para reduzir o espaço de busca pela solução, assim acelerando o processo. O objetivo é selecionar um subconjunto dos atributos, procurando unir qualidade e tempo de execução. Esta redução deve ser realizada de preferência acompanhada de um especialista do domínio, pois a retirada de dados de potencial útil, pode interferir diretamente na qualidade do conhecimento extraído. Além disso, pelo fato que no inicio do processo não se tem conhecimento sobre quais atributos serão importantes para atingir os objetivos, deve-se remover somente aqueles atributos que, com certeza, não têm nenhuma importância para o modelo final, para evitar perda de qualidade nas informações.

A terceira forma proposta para realizar a redução dos dados, consiste na redução do número de valores de um atributo. Isso é feito, geralmente, por discretização ou suavização dos valores de um atributo contínuo.

> Discretização de um atributo consiste na substituição de um atributo contínuo (inteiro ou real) por um atributo discreto, por meio do agrupamento de seus valores. Essencialmente, um algoritmo de discretização aceita como entrada os valores de um atributo contínuo e gera como saída uma pequena lista de intervalos ordenados [$\mathbf{V}_{inferior::}\mathbf{V}_{superior}$], do modo que $\mathbf{V}_{inferior}$ e $\mathbf{V}_{superior}$ são, respectivamente, os limites inferior e superior do intervalo. Os métodos de discretização podem ser classificados em supervisionados ou não-supervisionados, locais ou globais, e parametrizados ou não-parametrizados. (Félix, Rezende, Monard, & Caulkins, 2000 apud Rezende, 2005)

Segundo Weiss & Indurkhya (1998, apud Rezende, 2005), o objetivo da suavização dos valores de um atributo é diminuir o número de valores do mesmo sem discretizá-lo. Nesse outro método, são agrupados os valores de um determinado atributo, mas, diferente do processo da discretização, cada grupo de valores é substituído por um valor numérico que o represente. Esse novo valor pode ser a média, a mediana ou mesmo os valores de borda de cada grupo.

3. METODOLOGIA

O projeto desenvolvido possui caráter teórico-prático, no qual foi realizado um estudo sobre o Processo de Descoberta de Conhecimento, apresentado uma descrição conceitual sobre as etapas existente neste processo, dando uma ênfase maior na etapa de pré-processamento que vem a ser o principal objetivo do trabalho, e da implementação do protótipo da ferramenta proposta.

No desenvolvimento a metodologia abordada foi a Orientação a Objetos, em conjunto com a UML como linguagem para o projeto de modelagem da ferramenta. Outras ferramentas utilizadas no desenvolvimento são apresentadas no item 3.2.

Para auxiliar no desenvolvimento utilizado o método iterativo e incremental, desenvolvendo e testando cada etapa do projeto, e sempre que necessário realizando modificações ou adições de novas funcionalidades, estas eram então incrementadas, dessa forma, o projeto evoluiu em versões, através da construção incremental e iterativa de novas funcionalidades até que o sistema completo apresentado estivesse construído.

4. SISTEMA PROPOSTO

O trabalho desenvolvido consiste em um sistema para trabalhar na etapa de pré-processamento no processo de descoberta de conhecimento, utilizando de uma interface simplificada, assim proporcionando mais agilidade e precisão na definição da nova base de dados preparando esta para a aplicação de um Algoritmo Genético Difuso Multiobjetivo para Descoberta de Conhecimento na etapa de mineração de dados, algoritmo este desenvolvido por Coelho (2004).

A motivação para desenvolver este sistema, esta ligada ao fato da importância que os dados estão ganhando dentro de grandes empresas, uma vez que estes são muito importantes para auxiliar as tomadas decisões e controle destas.

O processo realizado para gerar esta nova tabela, reúne as informações sobre os campos selecionados pelo usuário, para internamente construir um *Select*. Ao final deste processo será criada uma tabela no Banco de Dados com a estrutura e os dados retornados por esta consulta. O procedimento realizado para criar a Nova Tabela, gera uma cláusula SQL com a estrutura apresentada abaixo.

CREATE TABLE Nome da Nova Tabela
[**AS**]
SELECT (...)

Onde:

CREATE TABLE: é a cláusula da linguagem SQL, utilizada para criar uma nova tabela no Banco de Dados. Ela não sofre nenhuma alteração nos quatro SGBD's utilizados.

Nome da Nova Tabela: Nome definido pelo usuário na criação da Nova Tabela. Para bancos que utilizam schemas, o nome da tabela será antecedido pelo nome do schema e um ponto, como pode ser visto abaixo:

nomeSchema.nomeTabela

[AS]: Quando o SGBD escolhido for o PostgreSQL, será necessário utilizar a cláusula AS, antes da consulta.

SELECT (...): Consulta montada a partir dos dados definidos pelo usuário na interface. Ela tem a seguinte estrutura:

SELECT
Alias_tabela1.Campo1 AS Alias_campo1
Alias_tabela2.Campo2 AS Alias_campo2
.
FROM Nome_Tabela_base Alias_Tabela_Base
JOIN Tabela1 Alias_Tabela1 **ON** (...)
JOIN Tabela2 Alias_Tabela2 **ON** (...)
.
.
GROUP BY Alias_tabela1.Campo1, ...
Onde:

SELECT: Cláusula SQL que indica uma consulta no Banco de Dados.

Alias_tabela1.Campo1 AS Alias_campo1: Campo retornado pela consulta, sendo que:

Alias_tabela1 é o alias definido para a tabela na cláusula JOIN,

Campo1, referente ao campo da tabela.

Alias_campo1, nome dado ao campo na consulta.

FROM Nome_Tabela_base Alias_Tabela_Base: Tabela que dará origem à operação de junção, definido no sistema como a tabela base.

JOIN Tabela1 Alias_Tabela1 ON (...): Junção entre as tabelas. Neste ponto é definido o alias para cada tabela presente na consulta.

GROUP BY: Agrupa os dados. Neste ponto são definidos conjuntos que irão compor o contexto da pesquisa.

Para reunir esses campos que serão selecionados pelo usuário para a criação do *Select*, o sistema constrói uma estrutura quem vem a ser uma representação do banco de dados, mas somente com tabelas e campos que estão no contexto que vai sendo construído pelo usuário.

Essa estrutura é diferenciada para Banco de Dados que utilizam ou não *schemas,* esta diferença é refletida no primeiro nível da estrutura, na qual será completada pelos *schemas* como pode ser visto na Figura 3.

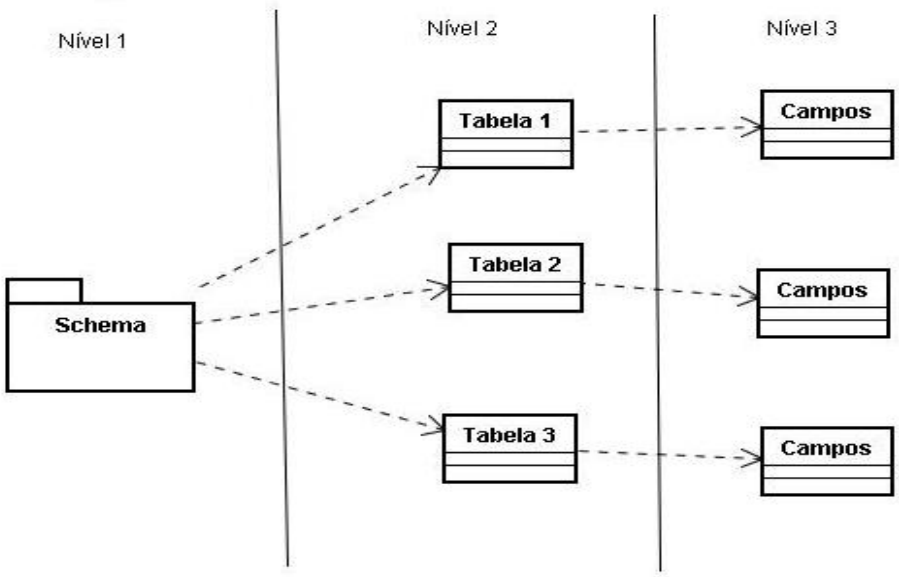

Figura 3 – Estrutura de nível montada pelo sistema.

Nesta estrutura o sistema constrói uma ramificação, quando uma tabela tem referência para outra tabela, esta tabela que é referenciada é criada dentro da tabela que a referenciou assim esta nova tabela criada dentro da primeira estrutura apresentada na Figura 3, vem a ser uma ramificação da tabela que referência outras tabelas, como pode ser visto na Figura 4.

Esta estrutura é criada dando continuidade para o Nível 2 apresentado na Figura 3, sendo criada quando algum campo da tabela que se encontra no Nível 2, é uma referência para alguma outra tabela, referência esta que se da com uma chave estrangeira.

Com essas estrutura montadas, o sistema percorre estas ligações e vai adicionado as informações necessárias para a criação do *Select* que retorne uma estrutura contendo todos os campos selecionados pelo usuário, finalizando assim construindo uma tabela com dados que representem um contexto definido pelo usuário, e pronta para receber algoritmos que realizem o tratamento necessários para prepara a base a aplicação do Algoritmo Genético Difuso Multiobjetivo para Descoberta de Conhecimento na etapa de mineração de dados.

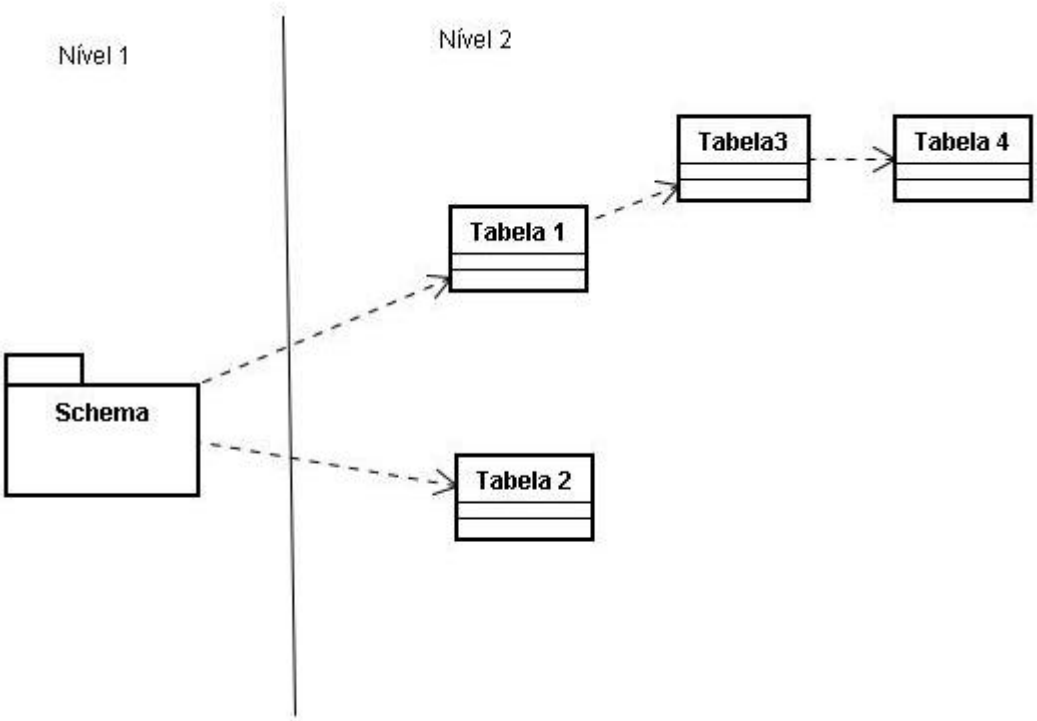

Figura 4 – Ramificação de Tabelas ligadas por chave estrangeira.

O protótipo desenvolvido neste projeto conclui a parte que cabe a seleção e organização dos dados, restando agora realizara o tratamento específico determinado nos objetivos do trabalho para prepara uma base que possa utilizada pelo Algoritmo Genético Difuso Multiobjetivo para Descoberta de Conhecimento.

4.1 Abordagem Orientada a Objetos

A escolha do uso da abordagem orientada a objetos, justifica pelo fato do ganho com reutilização de código, estrutura que possibilita uma manutenção de maneira mais simplificada e organizada, e facilidade na leitura do código desenvolvido.

> A programação orientada a objetos é uma forma de programar onde se utiliza o máximo possível de semelhança com o mundo real. Para poder programar orientado a objetos, que, diferente das linguagens imperativas, não se trata de uma seqüência de comandos e ações entregues para que o computador execute, é necessário do programador um grau de interpretação mais avançada.(Correia, 2007)

A reutilização de código oferecida pela orientação objeto permite que esta linguagem forneça um ganho de desempenho ao programador, este tipo de recurso

de destaca na linguagem como um diferencial, tornado esta uma metodologia de com grande potencial para desenvolvimento de software.

> Um dos grandes diferenciais da programação orientada a objetos em relação a outros paradigmas de programação que também permitem a definição de estruturas e operações sobre essas estruturas está no conceito de herança, mecanismo através do qual definições existentes podem ser facilmente estendidas. Juntamente com a herança deve ser enfatizada a importância do polimorfismo, que permite selecionar funcionalidades que um programa irá utilizar de forma dinâmica, durante sua execução.(Ricarte, 2001)

4.2 Ferramentas

O Quadro 1 apresenta as ferramentas adotadas e seus respectivos propósitos, para a consecução dos objetivos estabelecidos no trabalho

Quadro 1: Ferramentas utilizadas no projeto

Ferramenta	**Propósito**
J2SE, J2EE e J2ME	Plataformas da linguagem Java voltadas para o desenvolvimento de aplicações voltadas para os ambientes *desktop*, *web* e móvel.
JUDE Community versão 5	Ferramenta de modelagem UML
NetBeans IDE versão 6.5	Ambiente de desenvolvimento de aplicações Java.
Windows XP Service Pack 3	Plataforma operacional na qual foi desenvolvida a aplicação.
PostgreSQL versão 8.2	Sistema Gerenciador de Banco de Dados Objeto-Relacional.
MySQL Server versão 5.0	Sistema Gerenciador de Banco de

	Dados Relacional.
Oracle Database versão 11g	Sistema Gerenciador de Banco de Dados Relacional.
FireBird	Sistema Gerenciador de Banco de Dados Relacional.

4.3.1 Diagrama de Casos de Uso

- **Casos de Uso do Usuário**

1 – Abrir tela para realizar conexão com o Banco de Dados:

Este caso de uso se inicia quando o usuário inicia o sistema.
1- O usuário solicita o início da aplicação.
2- O sistema retorna a Tela de Login.
Include: Abre a Tela Login.

2 – Realizar conexão com Banco de Dados

Este caso de uso se inicia quando o usuário decide fazer o login na ferramenta.
1- O usuário insere os parâmetros necessários.
2- O sistema faz a validação dos dados.
Fluxo alternativo:
1.1 Se usuário pode fornecer dados incorreto.
2.1- Caso não consiga fazer a validação, o sistema emite uma mensagem referente ao campo preenchido com dados incorretos.

Include: Acessar Tela Principal.
Esse caso de uso é executado no momento em que o sistema faz a validação dos dados e realiza a conexão com o Banco de Dados.
1- O sistema abre a Tela Principal.

3 - Definir Tabela Base:

Este caso de uso se inicia quando o usuário define a tabela base.
1- O usuário solicita a opção criar nova tabela.
2- O sistema retorna a tela de definição da Nova Tabela.
3- O usuário insere os parâmetros da Nova Tabela.
4- O usuário define tabela base.
5- O sistema faz a validação dos dados.
Fluxo alternativo:
3.1 Se usuário pode fornecer dados incorreto.
4.1 O usuário não define tabela base.
4.2 O usuário tenta definir uma tabela com chave primaria composta como tabela base, o sistema não permite e emite uma mensagem.
5.1 Se não for possível fazer a validação, o sistema emite uma mensagem referente ao campo preenchido com dados incorretos.

Extend: Abrir Tela para definir tabela base.
Esse caso de uso é executado no momento em que o usuário decide definir a tabela base.
1- O sistema retorna as tabelas existentes no sistema.

4 – Modificar Tabela Base.

Este caso de uso se inicia quando o usuário decide modificar a tabela base.
1- O usuário solicita a opção para modificar a tabela base.
2- O sistema retorna a tela de definição de tabela base.
3- O usuário define tabela base.
5- O sistema faz a validação dos dados.
Fluxo alternativo:
3.1 O usuário tenta definir uma tabela com chave primaria composta como tabela base, o sistema não permite e emite uma mensagem.

5 – Inserir Campo na Nova Tabela.

Este caso de uso se inicia quando o usuário decide inserir campos na Nova Tabela.
1- O usuário seleciona o campo que deseja inserir na Nova Tabela.
2- O usuário define informações sobre o campo inserido.
3- O sistema faz inclusão do campo em Nova Tabela.
Fluxo alternativo:
2.1 O usuário pode definir operação incompatível com o tipo de dado selecionado, o sistema emite mensagem informado o erro.
2.2 O usuário tenta inserir um campo já existente no contexto definido.
2.3 O usuário pode tentar inserir campo que seja chave estrangeira, o sistema oferece mais de uma opção para campos deste tipo.

Extend: Abrir Tela para inserir campo chave estrangeira.
Esse caso de uso é executado no momento em que o usuário tenta inserir um campo, que seja chave estrangeira e escolhe a opção de encontrar tabela que foi referencia pelo campo chave estrangeira.
1- O sistema abre a Tela Inserir Campo Chave Estrangeira.

6 – Remover campo da Nova Tabela.

Este caso de uso se inicia quando o usuário decide remover campos na Nova Tabela.
1- O usuário seleciona o campo que deseja remover da Nova Tabela.
3- O sistema faz exclusão do campo da estrutura de Nova Tabela.

7 – Exibir informações do campo de Nova Tabela.

Este caso de uso se inicia quando o usuário decide obter informações sobre um campo inserido em Nova Tabela.
1– O usuário solicita do sistema informações sobre um determinado campo existente em Nova Tabela.
2– O sistema retorna informações sobre o campo selecionado.

8 – Visualizar Nova Tabela.

Este caso de uso se inicia quando o usuário decide ter uma visualização da Nova Tabela.
1– O usuário solicita do sistema uma visualização de Nova Tabela.
2– O sistema retorna todos os dados já inseridos em Nova Tabela.
Fluxo alternativo:
1.1 O usuário solicita uma visualização de Nova Tabela antes de inserir dados nesta, o sistema retorna uma mensagem.

9 – Executar SQL.

Este caso de uso se inicia quando o usuário decide realizar uma consulta SQL.
1– O usuário realiza a consulta utilizando comandos SQL referente ao banco de dados definido no início da aplicação.

10 – Criar Nova Tabela.

Este caso de uso se inicia quando o usuário decide finalizar a construção da Nova Tabela.
1– O usuário solicita do sistema a Finalização da Nova Tabela.
Fluxo alternativo:
1 – O usuário pode ter colocado campo em Nova Tabela, e não ter feito a ligação entre as tabelas, o sistema emite uma mensagem de erro e não permite gerar a Nova Tabela.

11 – Realizar Discretização

Este caso de uso se inicia quando o usuário decide discretizar os dados da Nova Tabela.
1 – Usuário seleciona o atributo a ser discretizado.
Fluxo alternativo
1 – Atributo não pode ser discretizado, sistema emite uma mensagem correspondente.

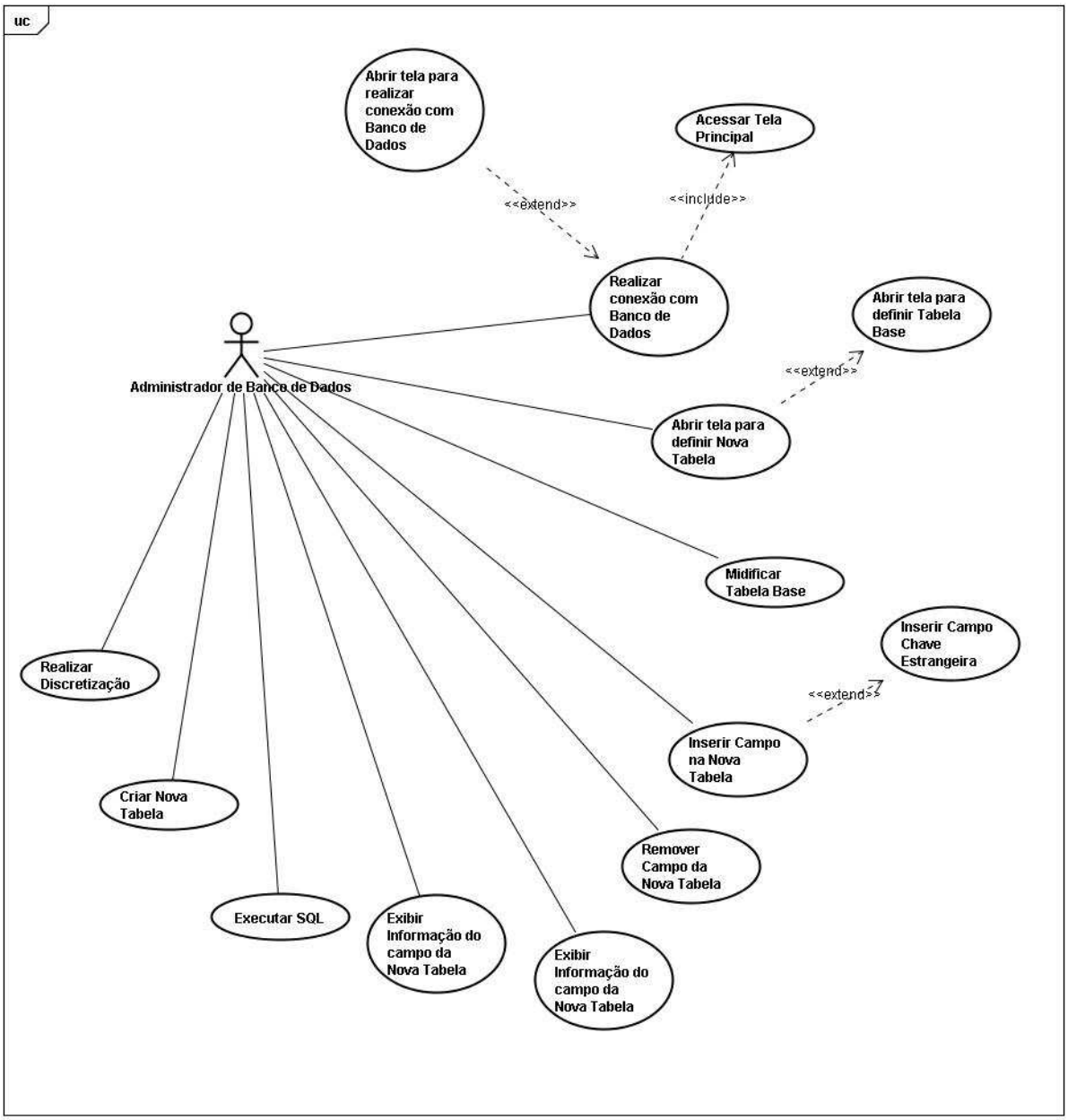

Figura 5: Diagrama de Caso de Uso.

4.3.2 Diagrama de Classes

Os Diagramas de classe do sistema formam divididos em:

- Acesso.
- Metadados.
- Componentes.
- Dialogos.
- Telas.

- Estrutura.

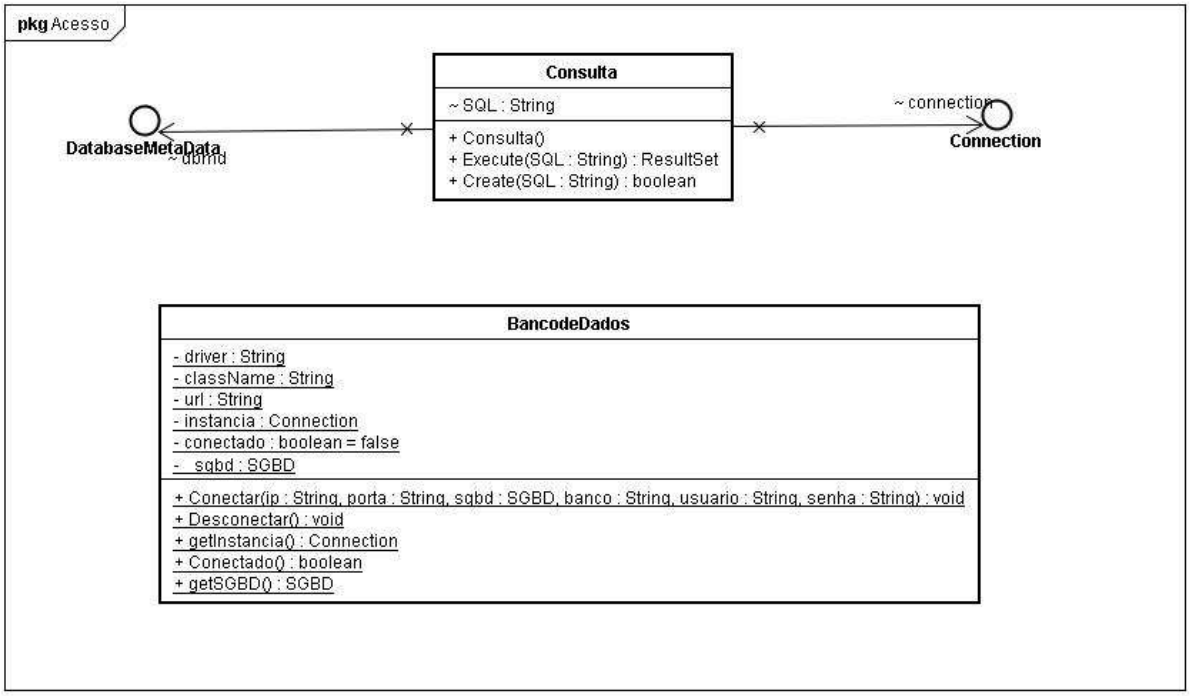

Figura 6: Diagrama de classe do pacote de Acesso.

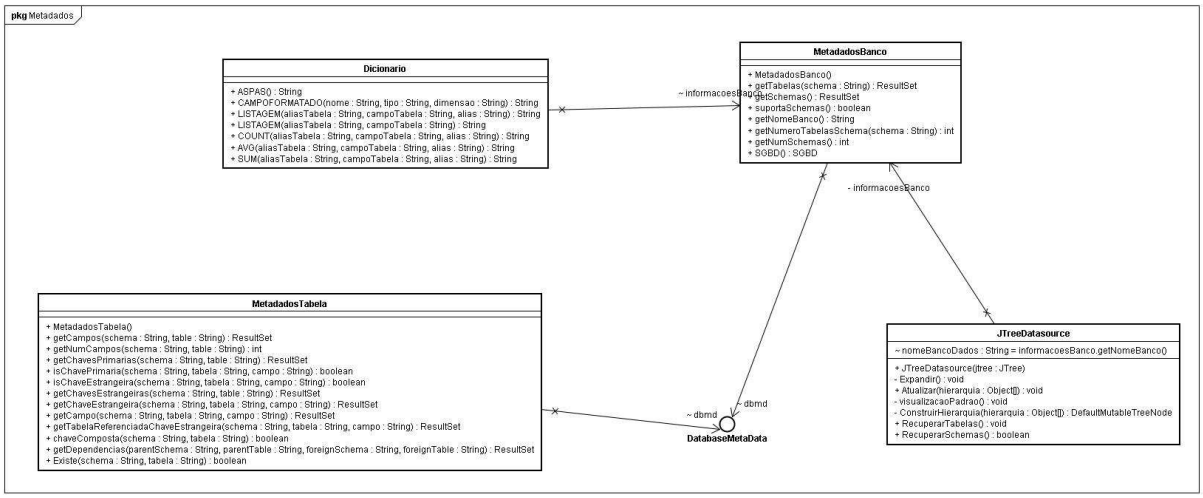

Figura 7: Diagrama de classe do pacote de Metadados.

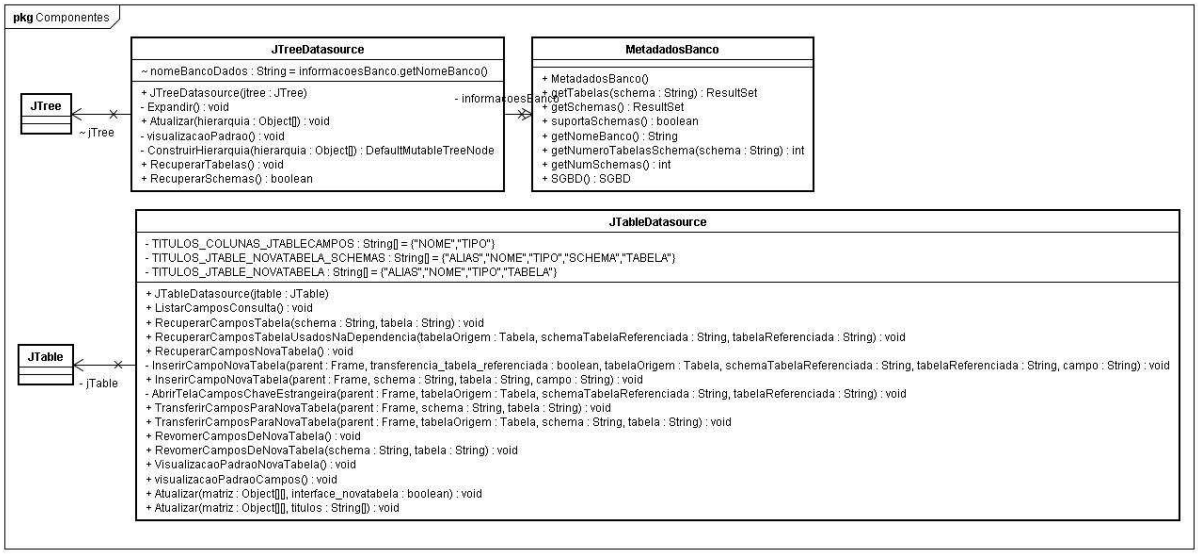

Figura 8: Diagrama de classe do pacote de Componentes.

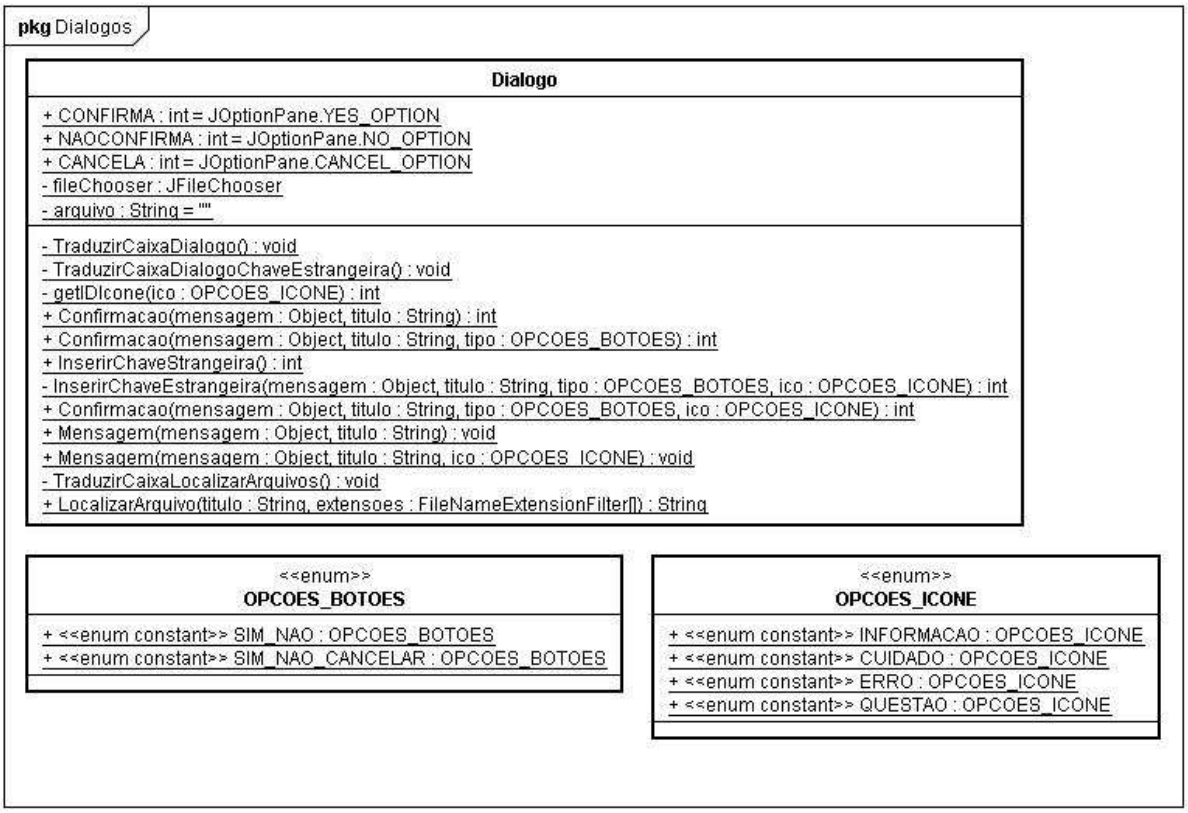

Figura 9: Diagrama de classe do pacote de Dialogos.

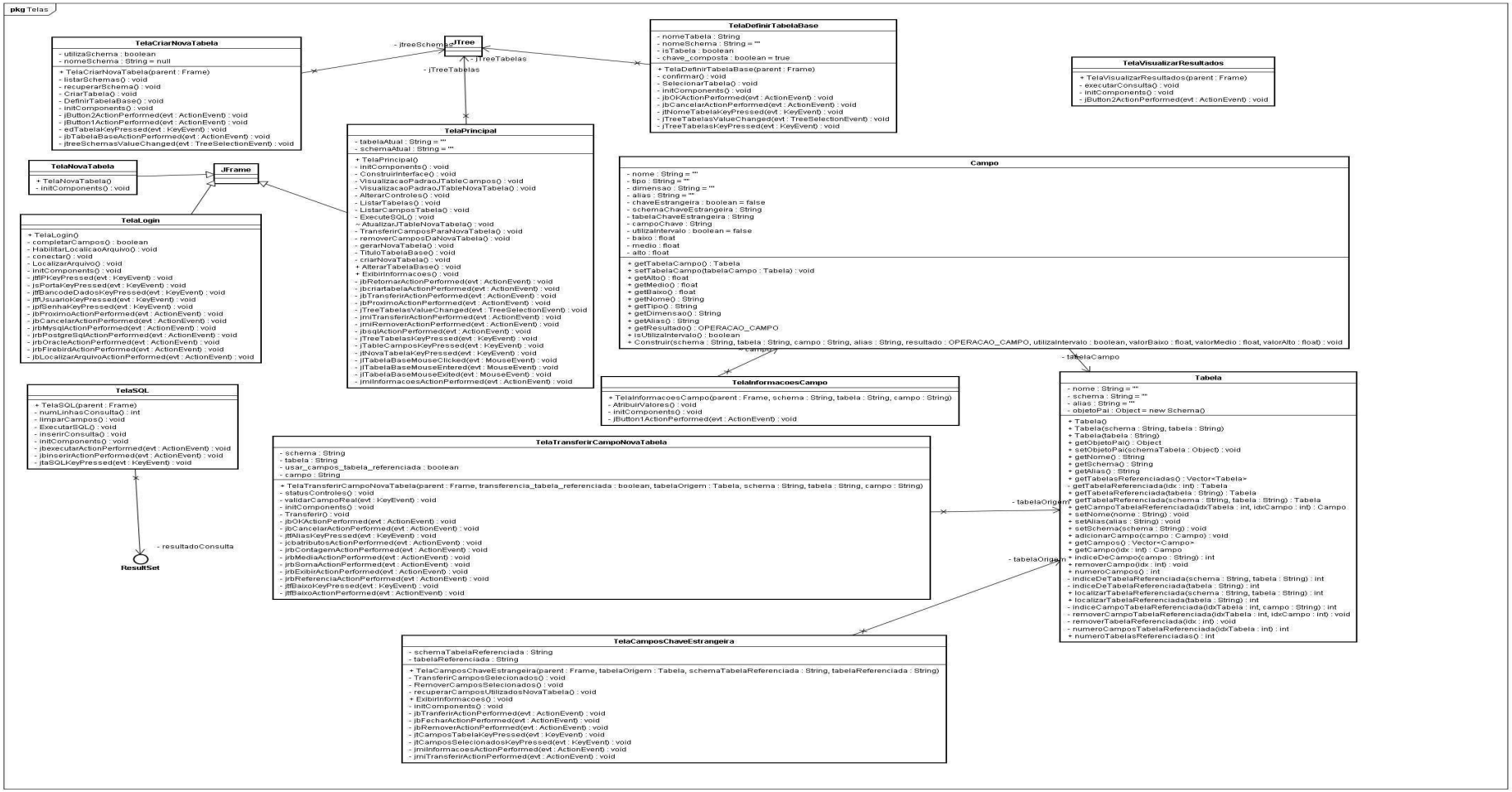

Figura 10: Diagrama de classe do pacote de Telas.

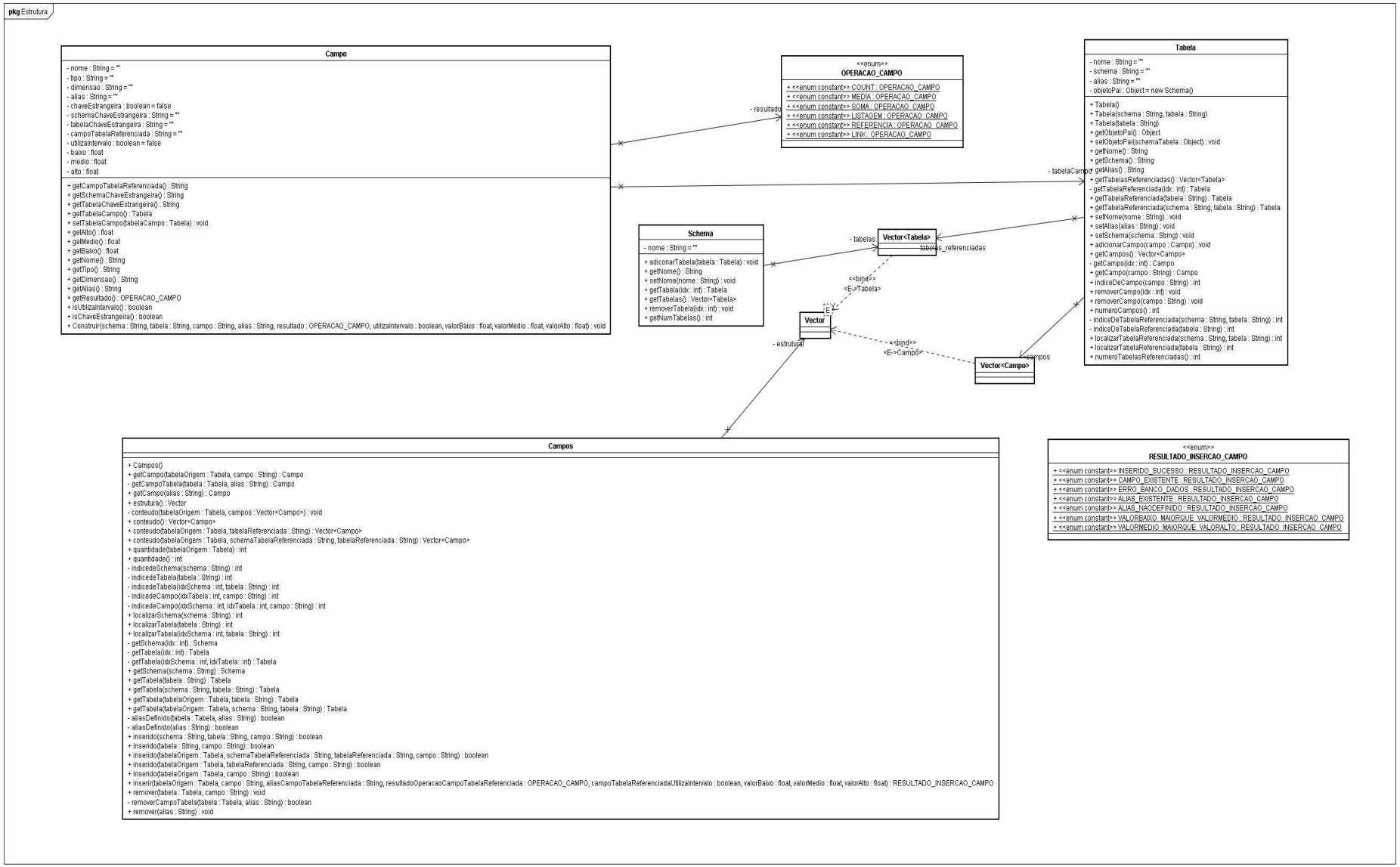

Figura 11: Diagrama de classe pacote de Estrutura.

4.3.3 Diagramas de Pacote.

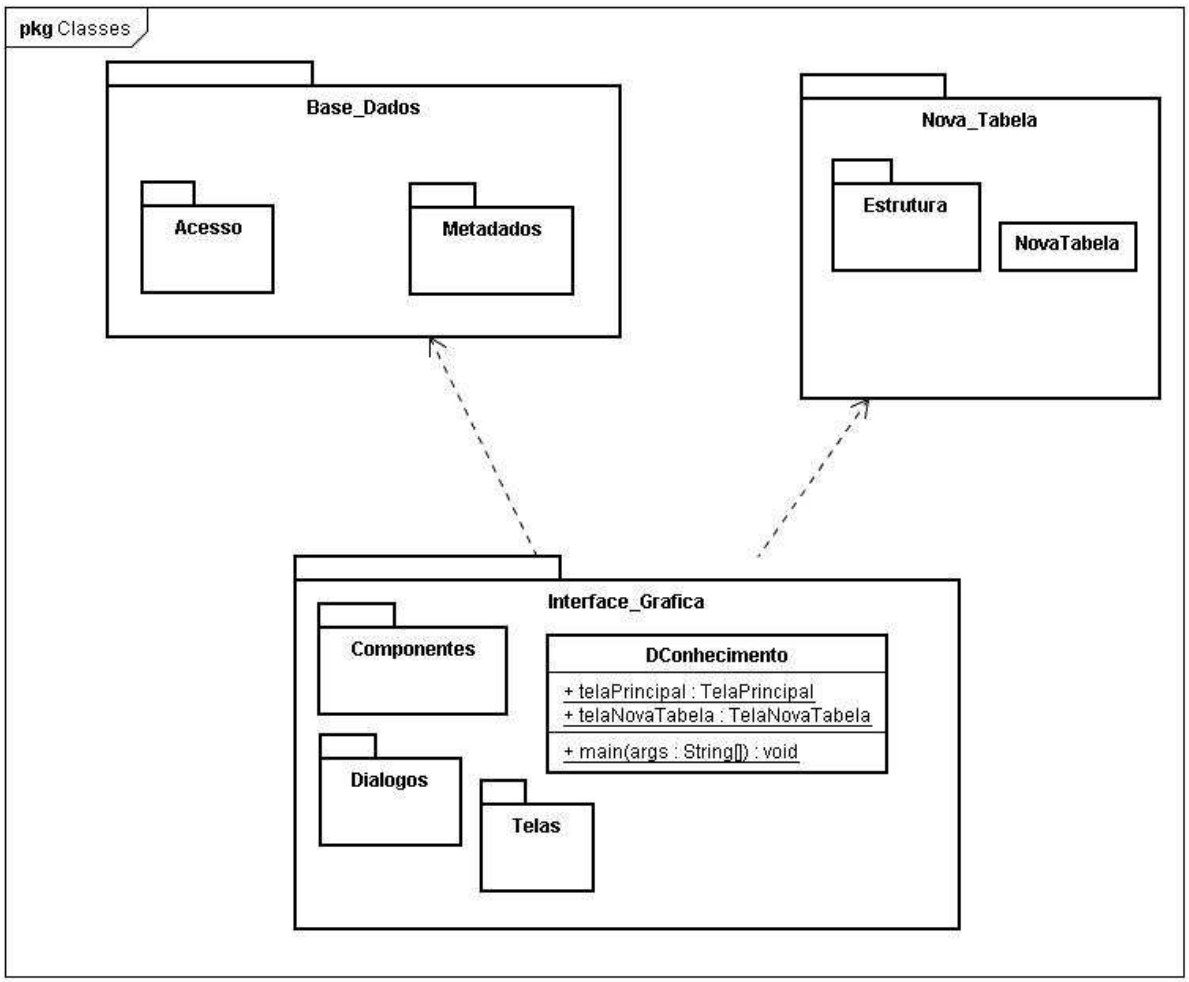

Figura 12 – Diagrama de Pacote do sistema.

4.3.4 Diagramas de Seqüência

- Diagrama de Seqüência referente à inicialização do sistema.

Acessar Tela Principal.

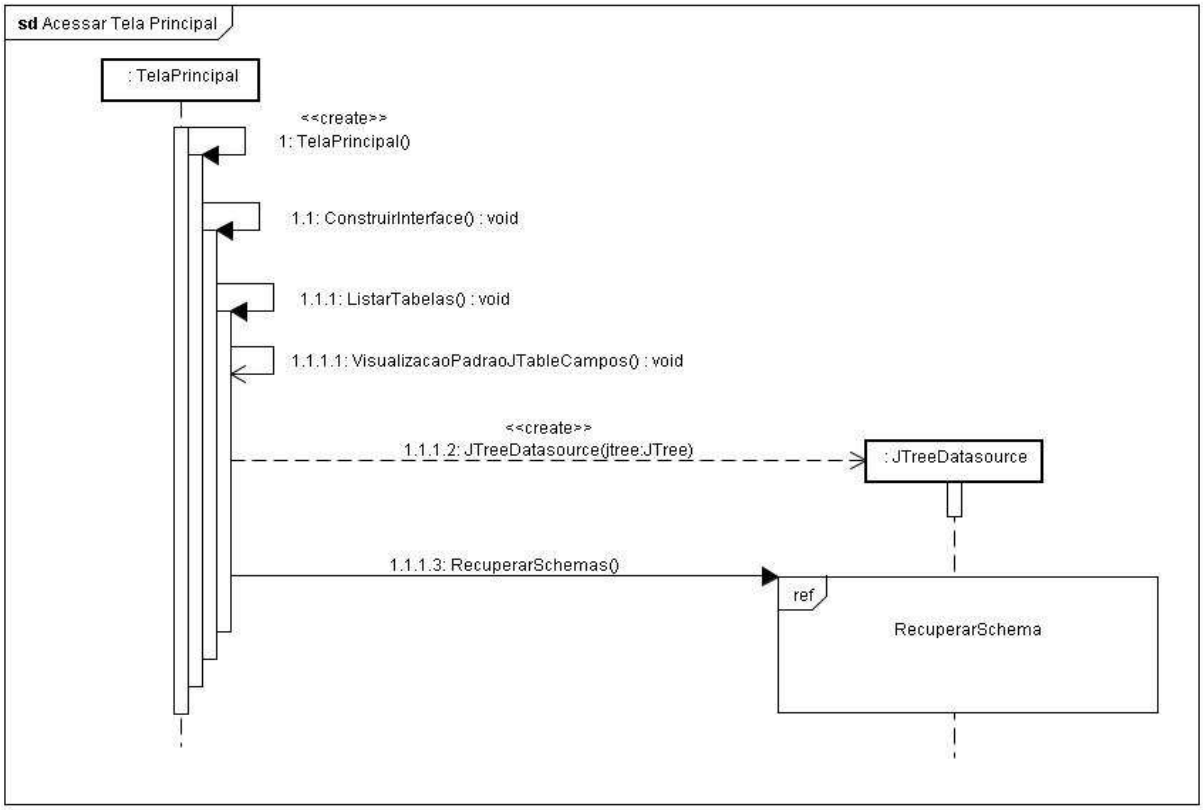

Figura 13 - Acessar Tela Principal.

✓ Realizar conexão com Banco de Dados.

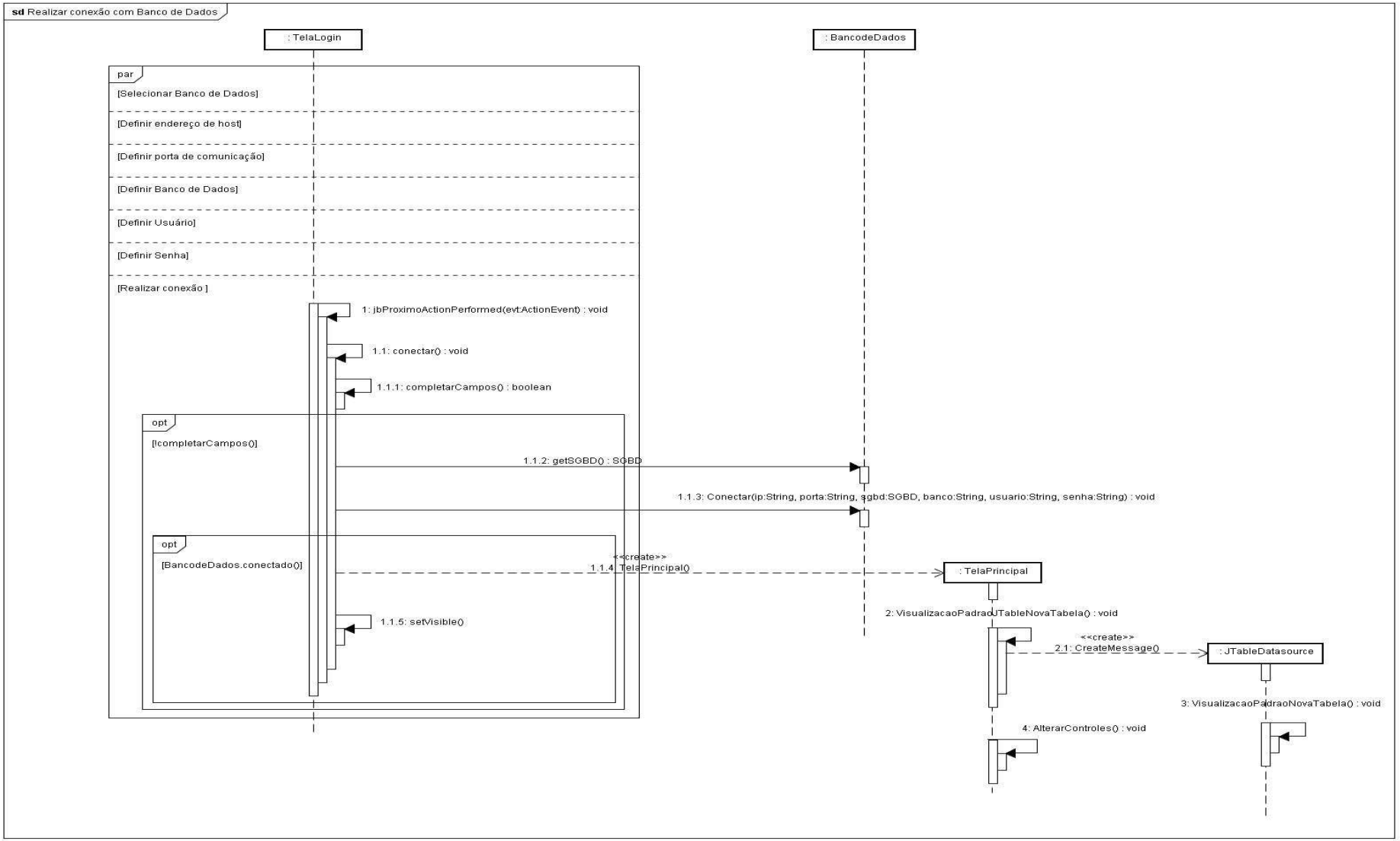

Figura 14- Realizar conexão com Banco de Dados.

- Diagrama de Seqüência referente à definição Nova Tabela.
- ✓ Abrir tela para definir Nova Tabela.

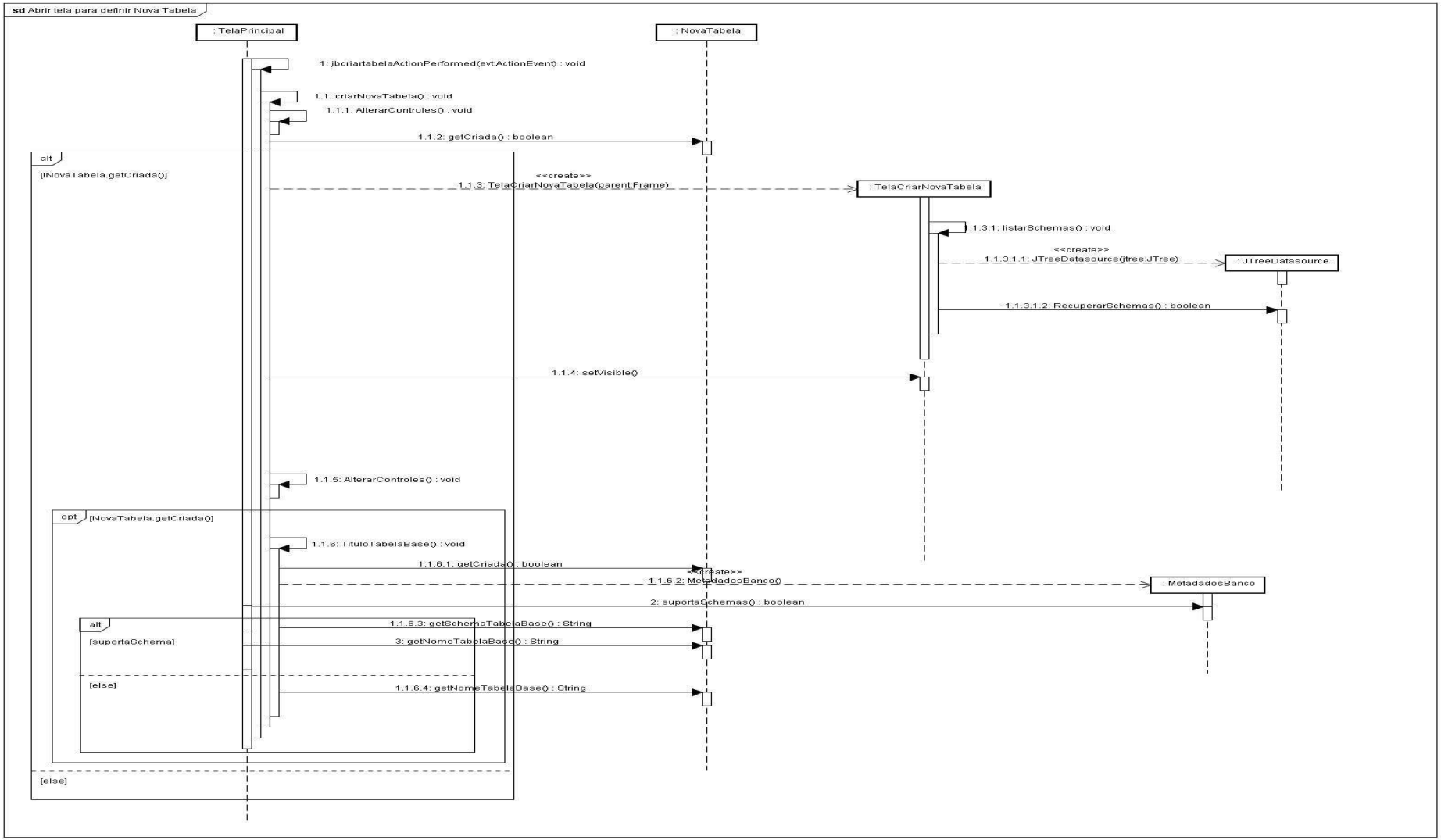

Figura 15 – Abrir tela para definir Nova Tabela.

✓ Abrir tela para definir tabela base.

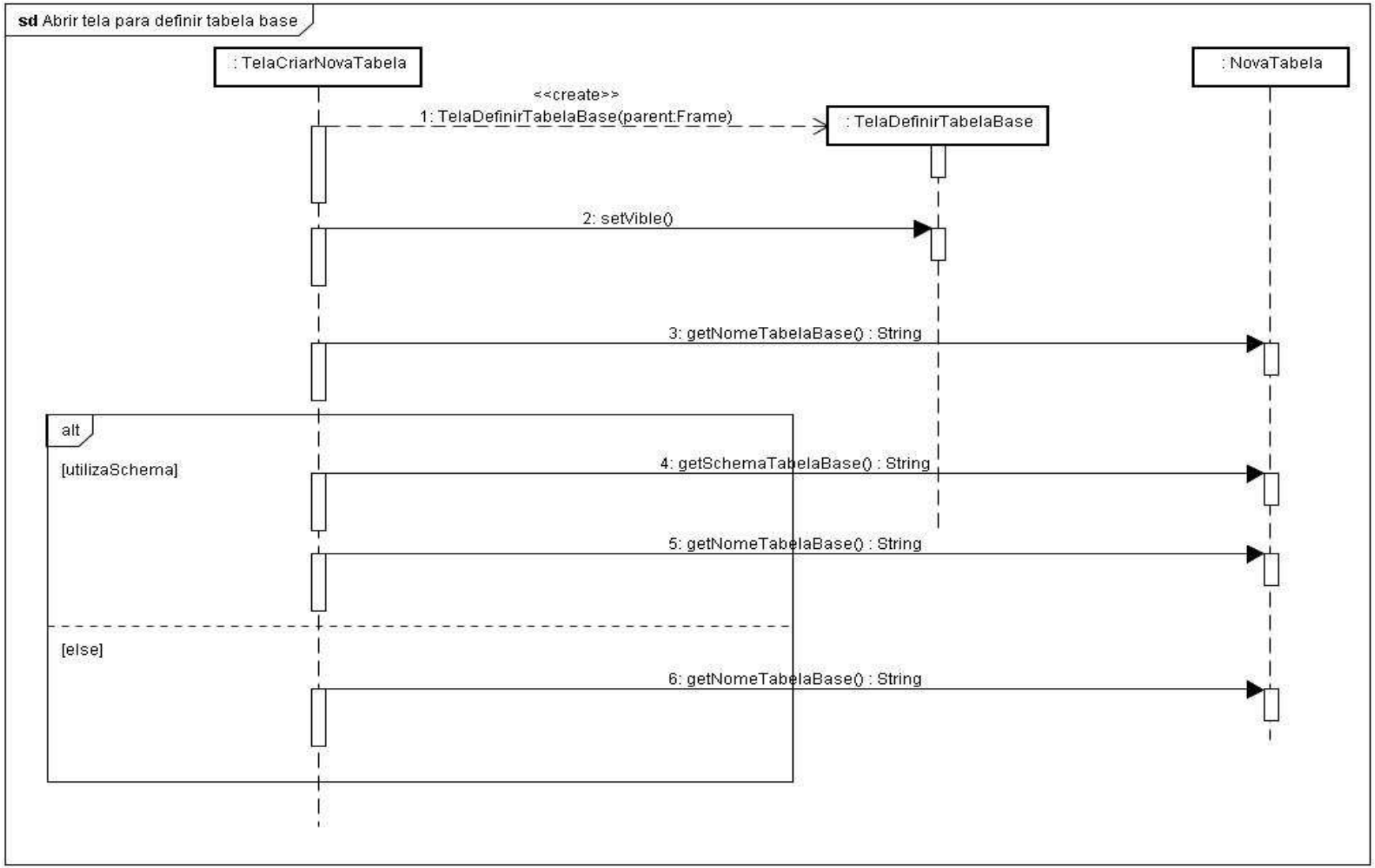

Figura 16 - Abrir tela para definir tabela base.

✓ Modificar Tabelas Base.

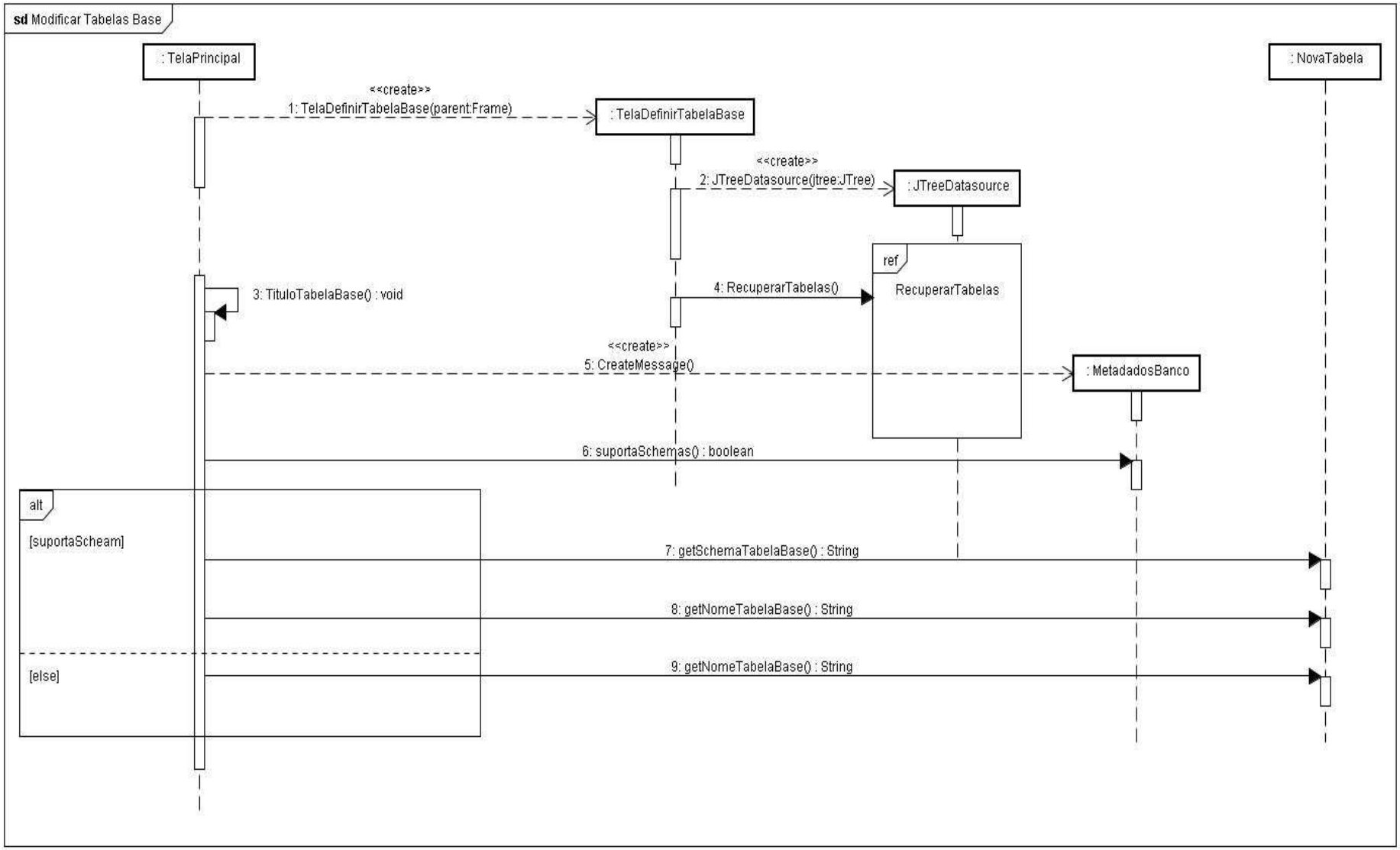

Figura 17 - Modificar Tabelas Base.

- Diagrama de Seqüência referente à manipulação de informação de Nova Tabela.

✓ Transferir campo para Nova Tabela.

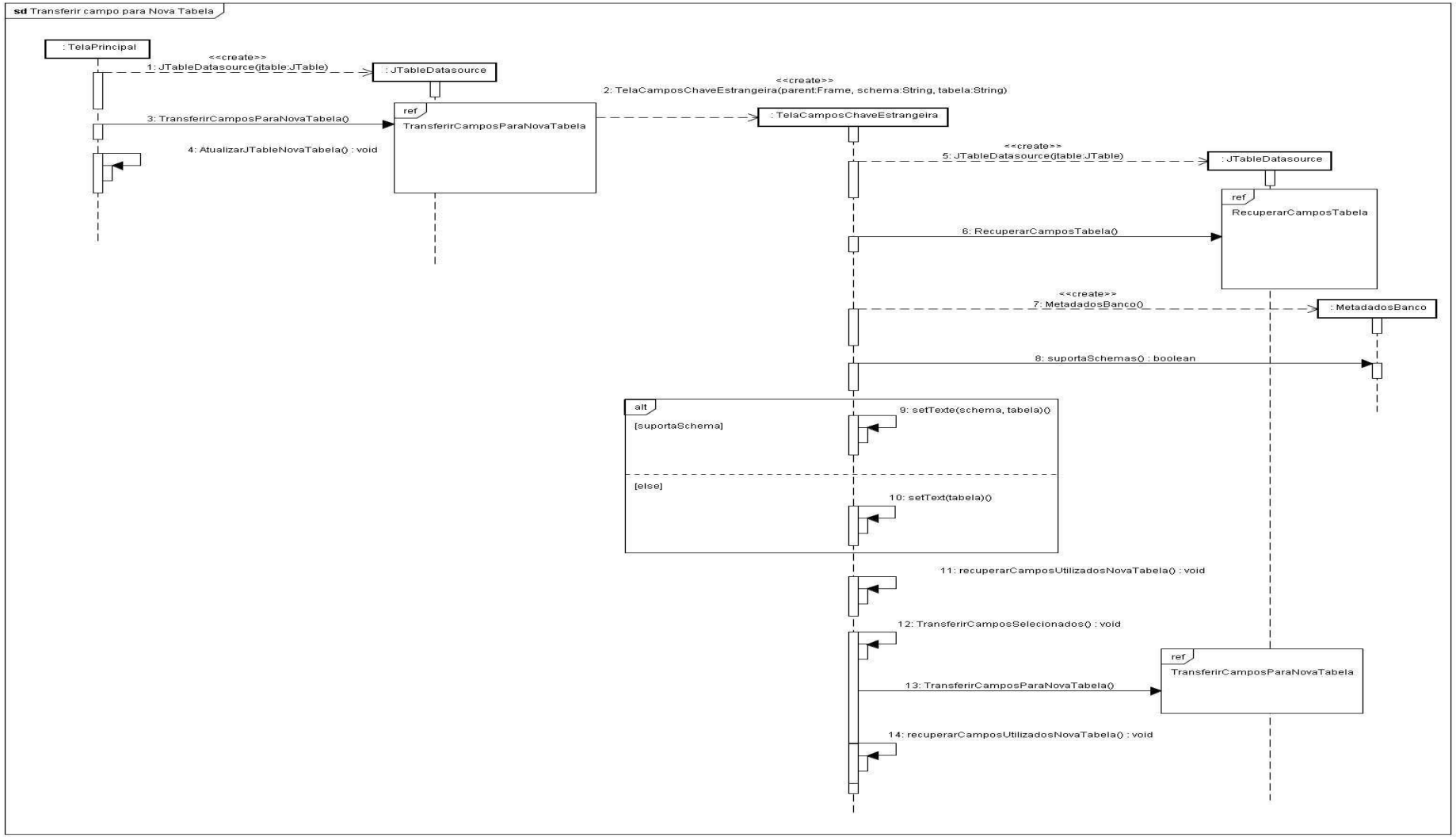

Figura 18 - Transferir campo para Nova Tabela.

✓ Inserir Campo Chave Estrangeira.

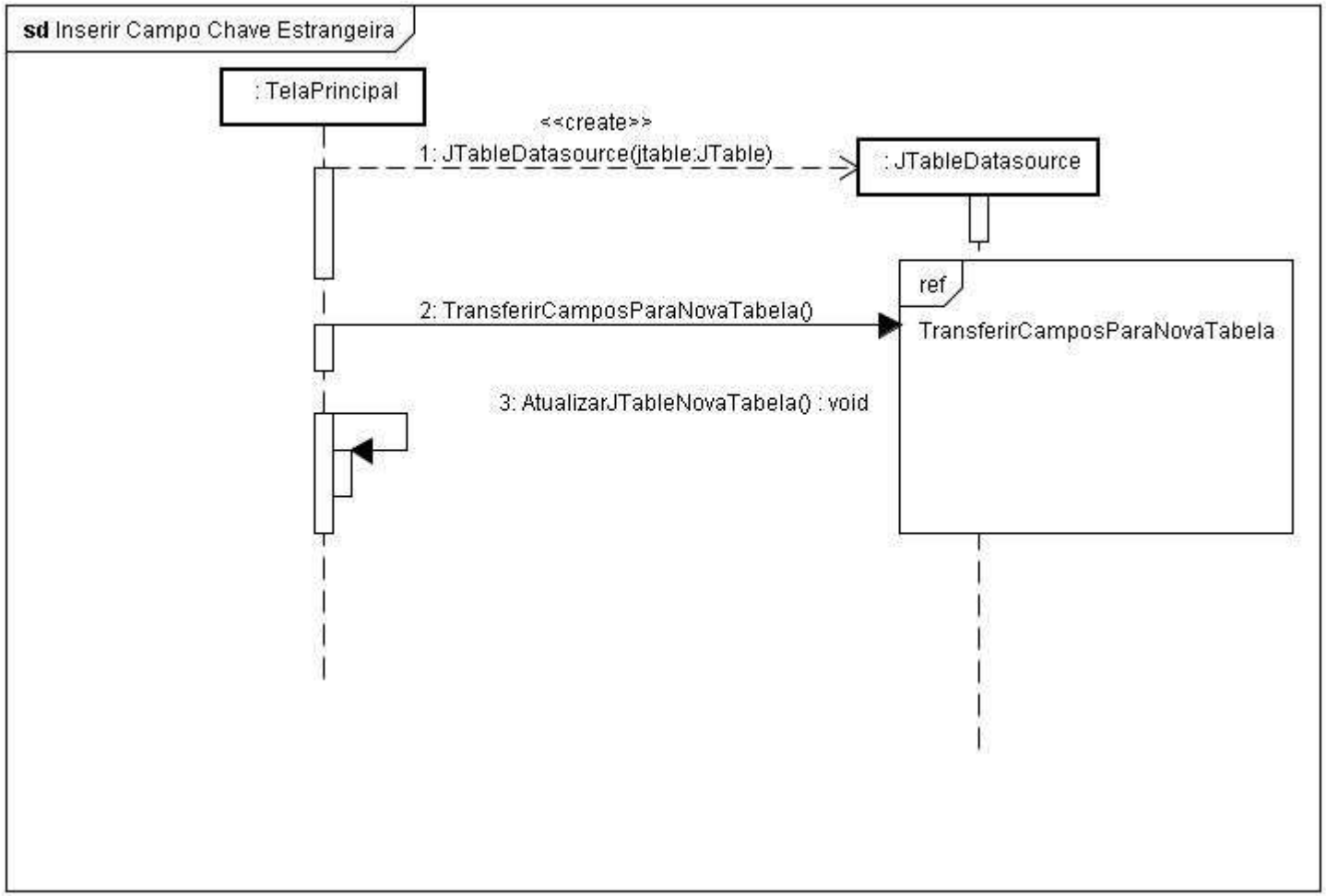

Figura 19 - Inserir Campo Chave Estrangeira.

✓ Remover Campo da Nova Tabela.

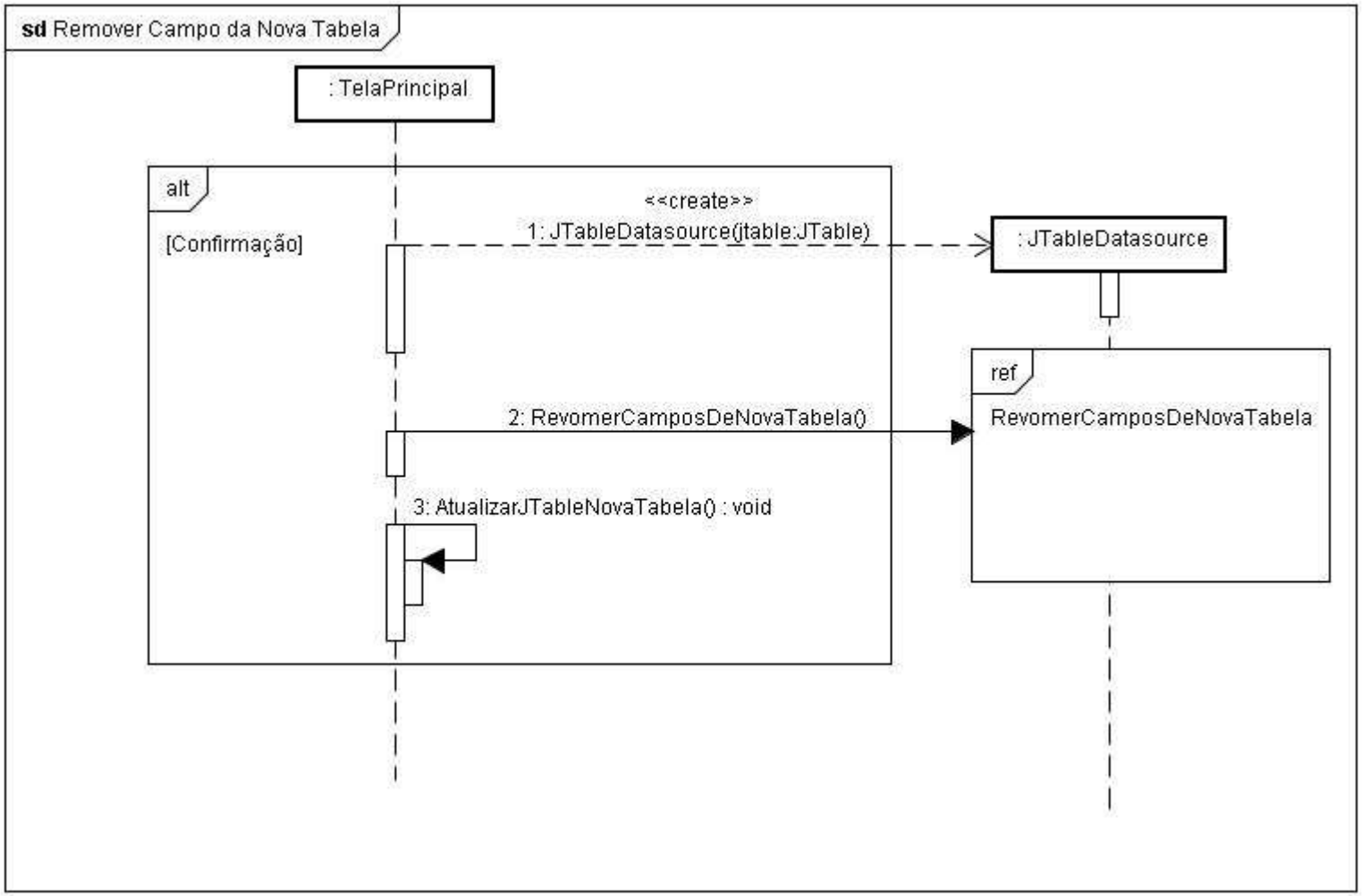

Figura 20 - Remover Campo da Nova Tabela.

✓ Exibir Informação do campo da Nova Tabela.

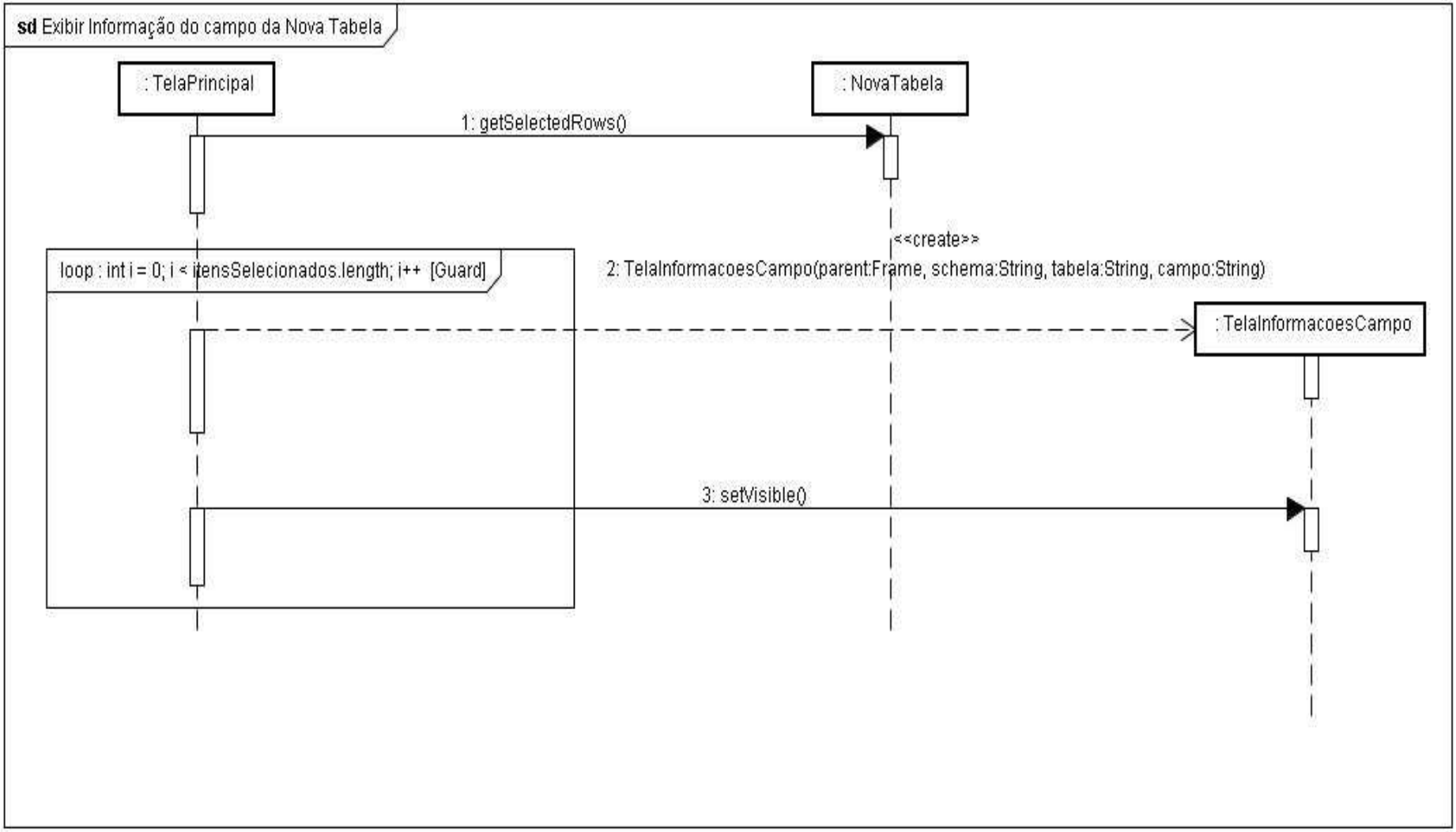

Figura 21 - Exibir Informação do campo da Nova Tabela.

✓ Visualizar Resultados.

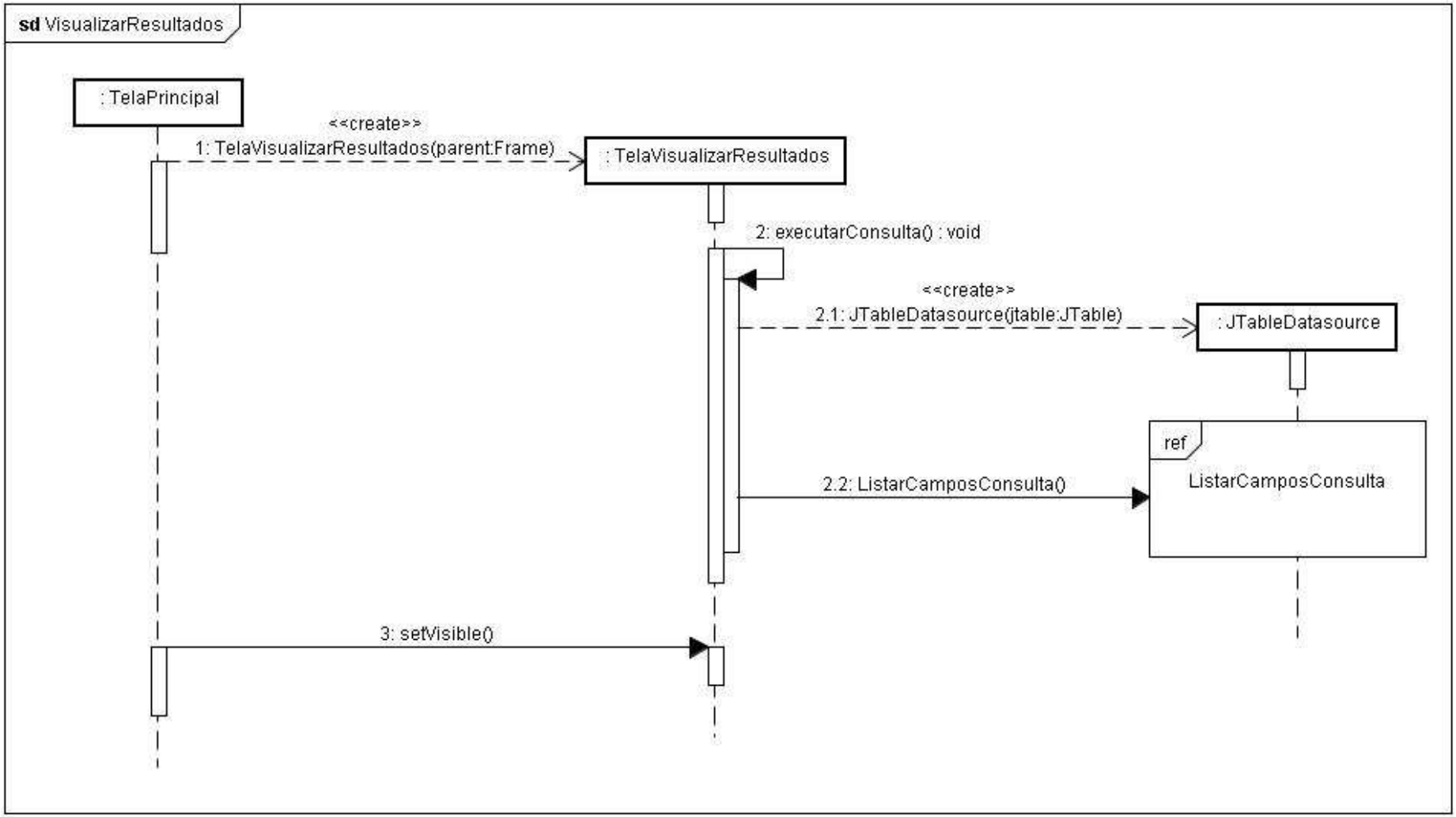

Figura 22 – Visualizar Resultados.

✓ Executar SQL.

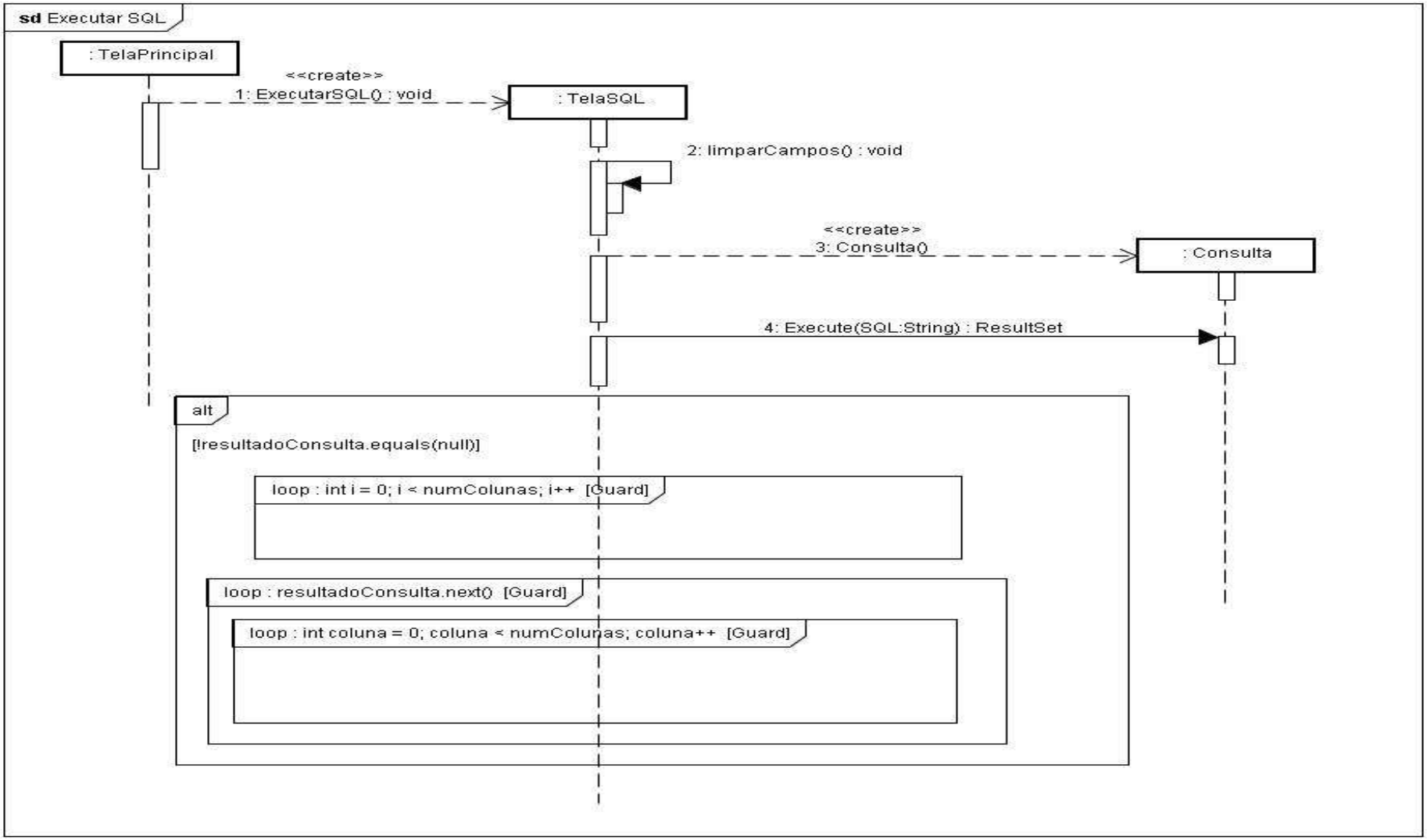

Figura 23 - Executar SQL.

- Diagrama de Seqüência referente a estruturas de JTableDataSourse.

✓ JTableDataSource Listar Campos Consulta.

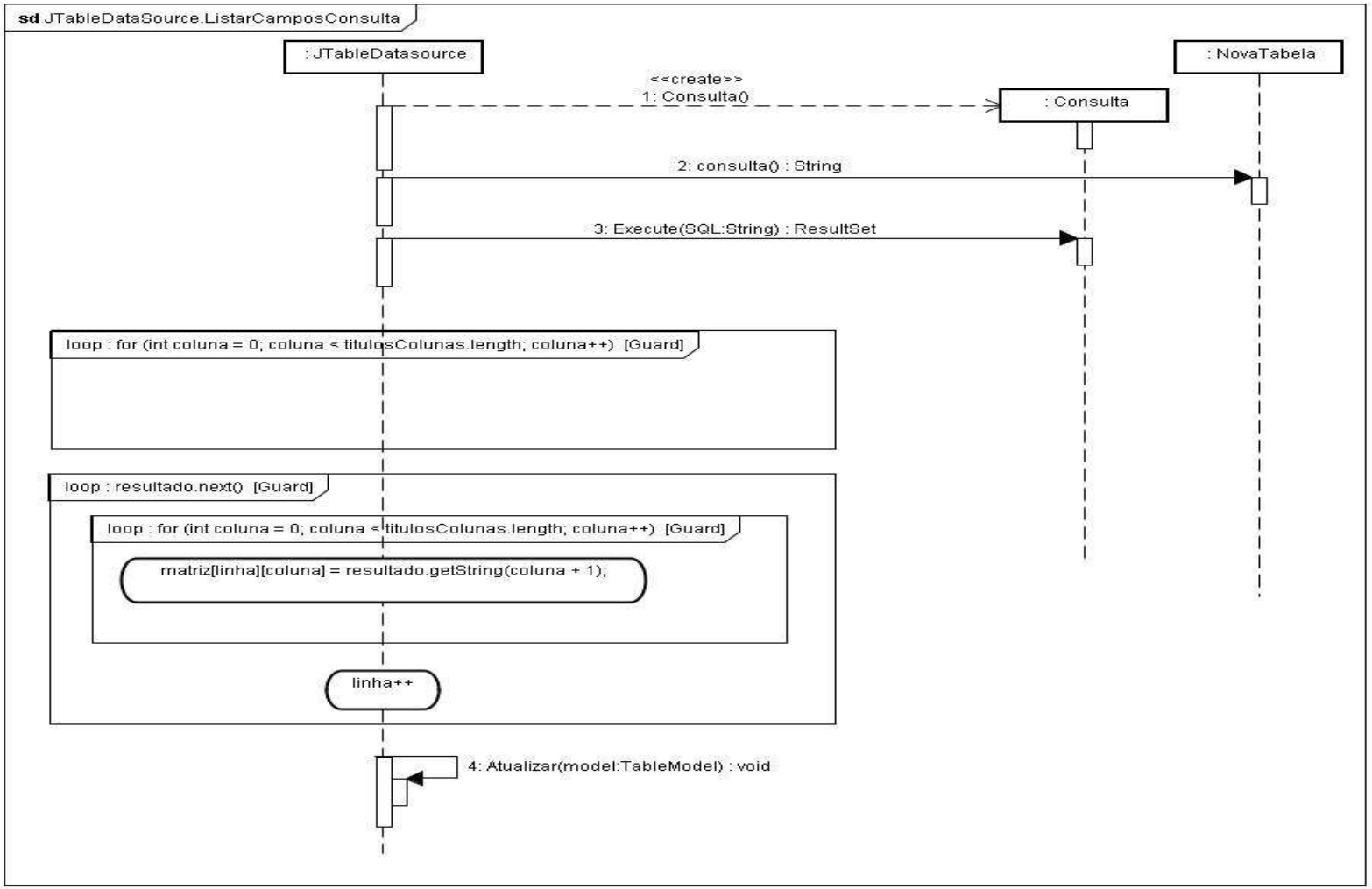

Figura 24 – JtableDataSource Listar Campos Consulta.

- ✓ JTableDataSource Remover Campos de Nova Tabela.

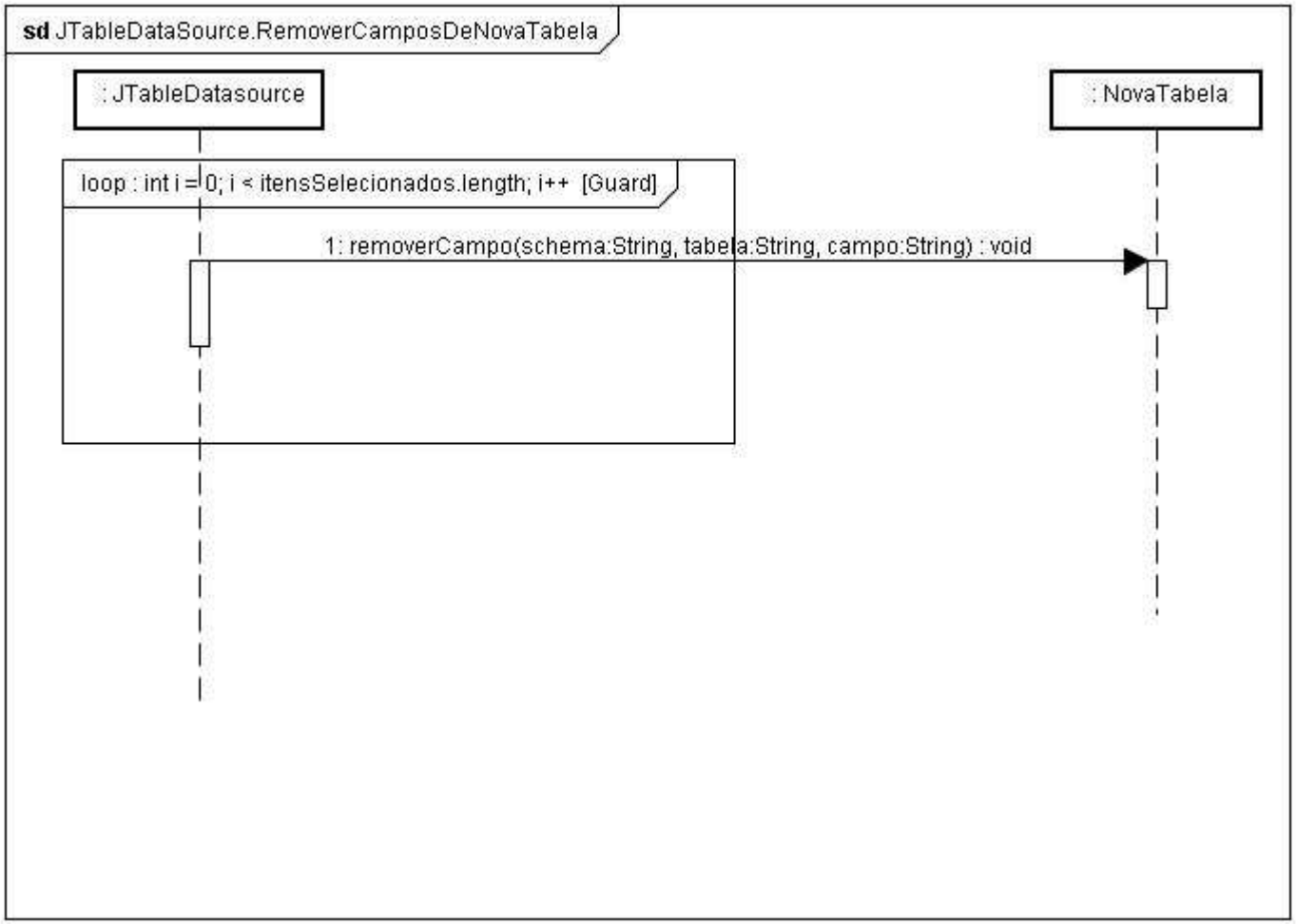

Figura 25 - JTableDataSource Remover Campos de Nova Tabela.

✓ JTableDataSource Transferir Campos para Nova Tabela.

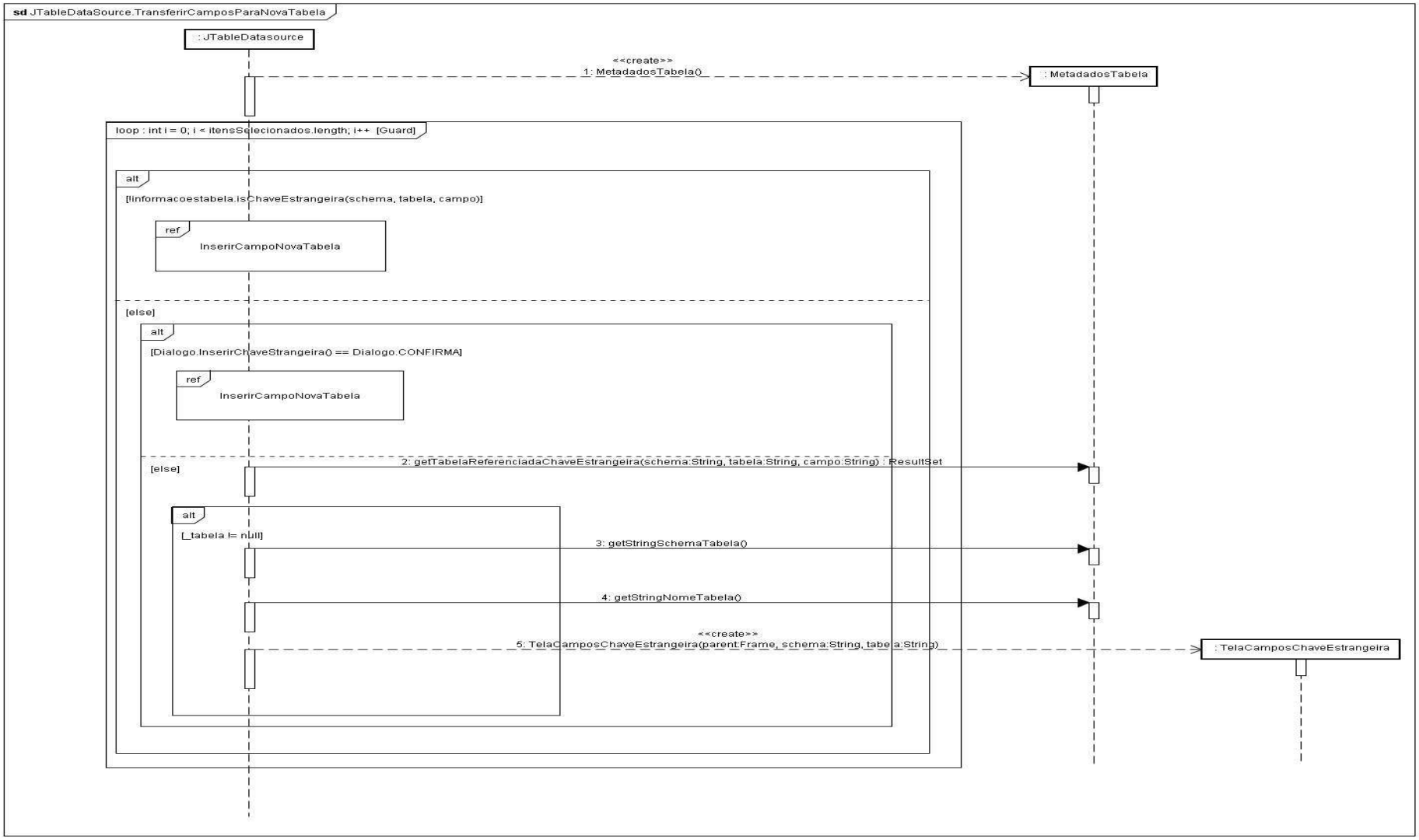

Figura 26 – JtableDataSource Transferir Campos para Nova Tabela.

✓ JTableDataSource Recuperar Campos Tabela.

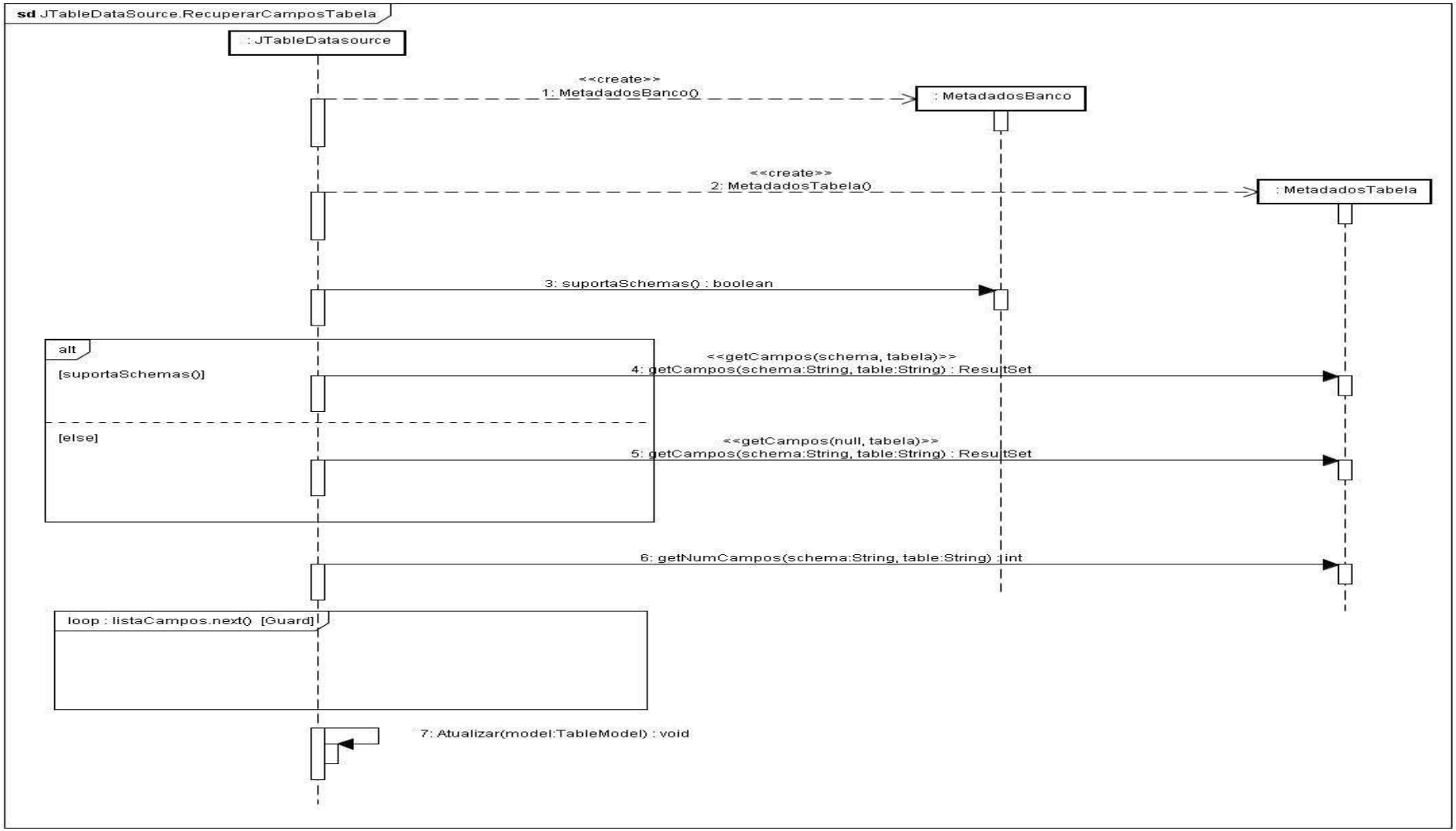

Figura 27 – JtableDataSource Recuperar Campos Tabela.

✓ JTreeDataSource Recuperar Tabelas.

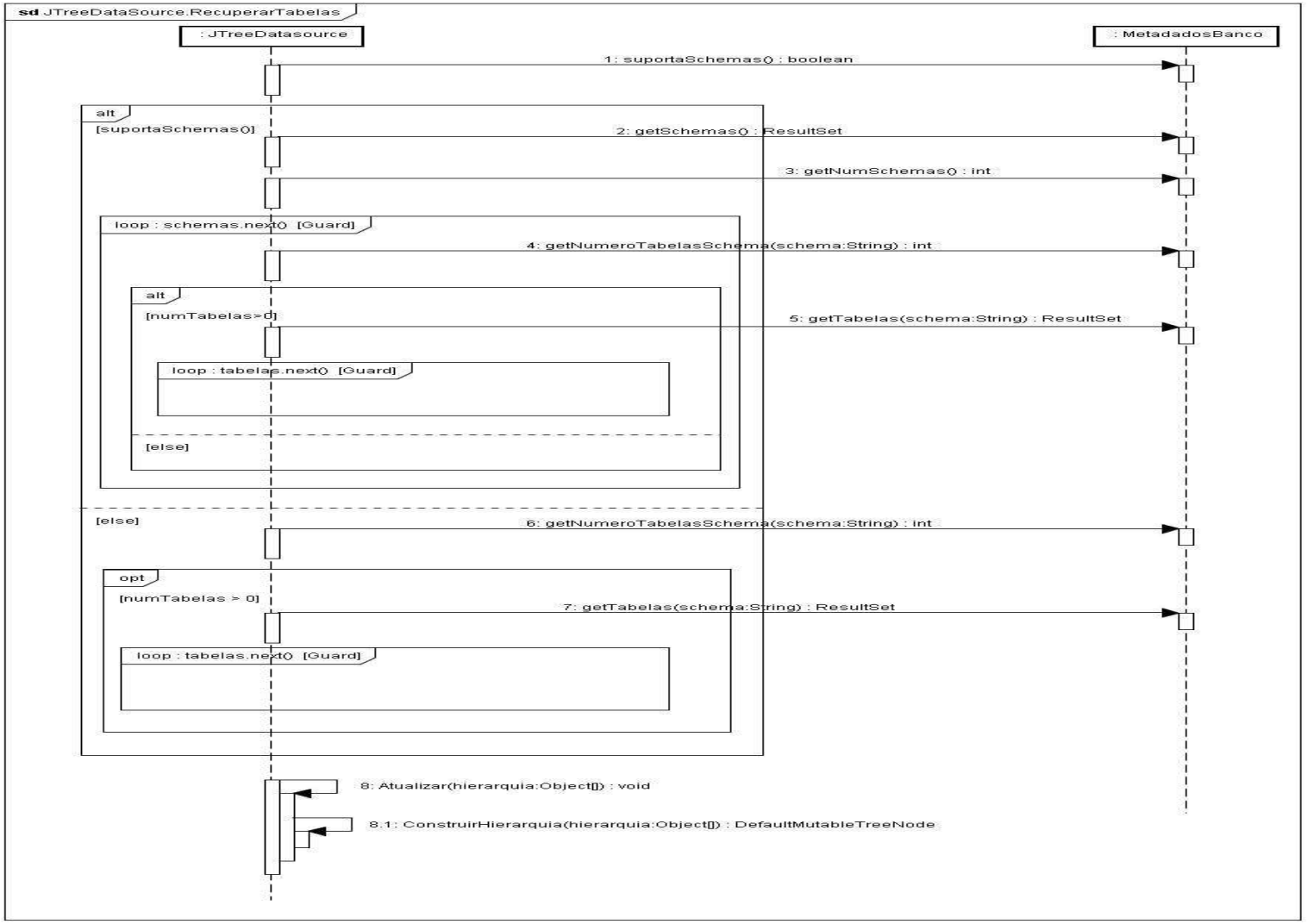

Figura 28 - JTreeDataSource Recuperar Tabelas.

✓ JTreeDataSource Recuperar Schemas.

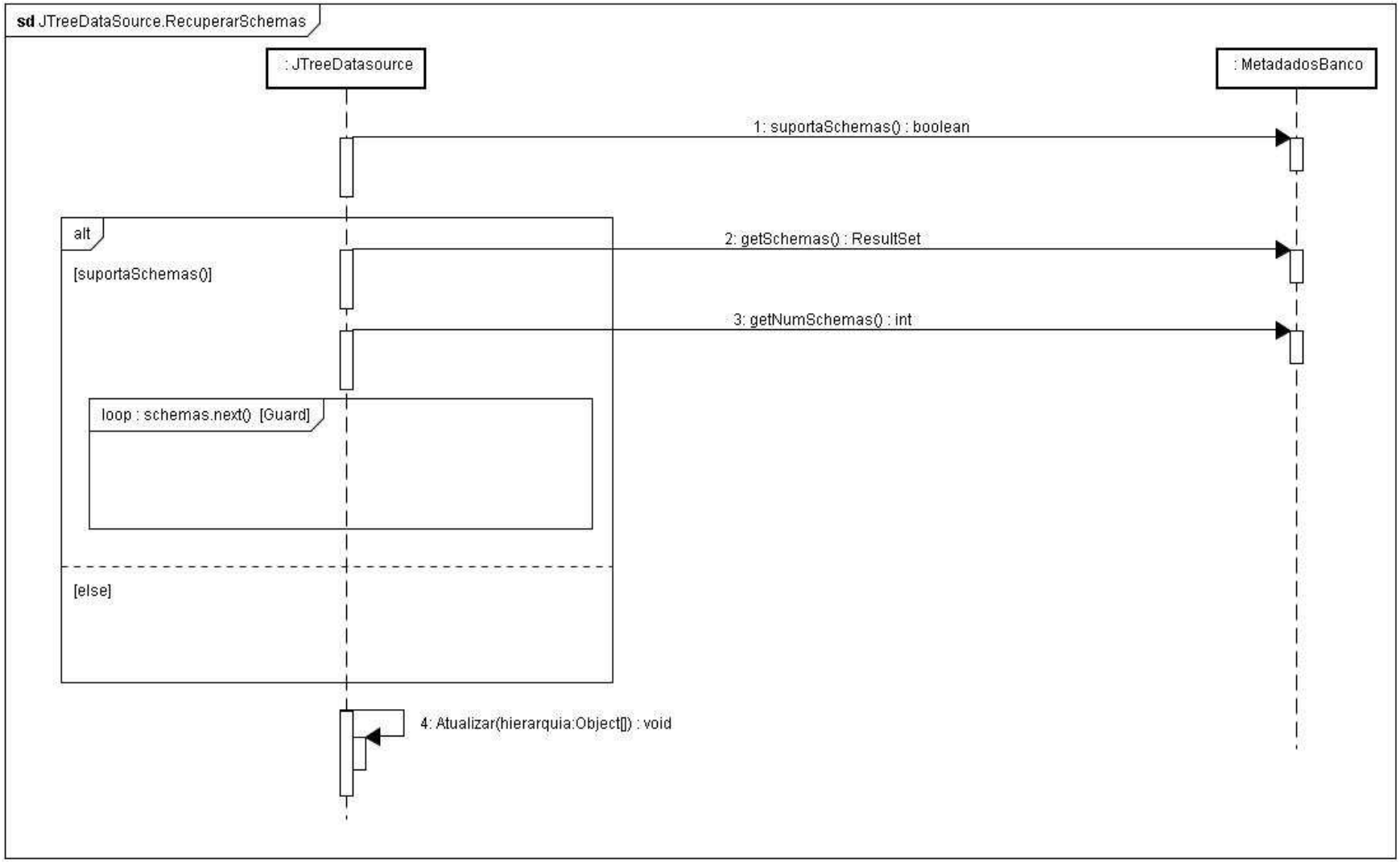

Figura 29 - JTreeDataSource Recuperar Schemas.

4.4 Interfaces do Sistema

A interface inicial do sistema apresenta uma tela para obter parâmetros para realizar a conexão com o banco de dados, como mostrado na figura 30.

- Informações coletadas:
- Banco de dados selecionado.
- IP – referente à localização do banco.
- Porta – referente à localização do banco.
- Baco de dados.
- Usuário.
- Senha.

Figura 30 – Tela de Login do sistema.

Após a passagem dos parâmetros corretos e realizada a conexão, a tela principal do sistema se abre, como pode ser visto na figura 31. Nesta tela o usuário

define a tabela base para sua consulta, insere os campos relevantes, visualiza os dados referentes à nova tabela, e tem disponível um editor de SLQ.

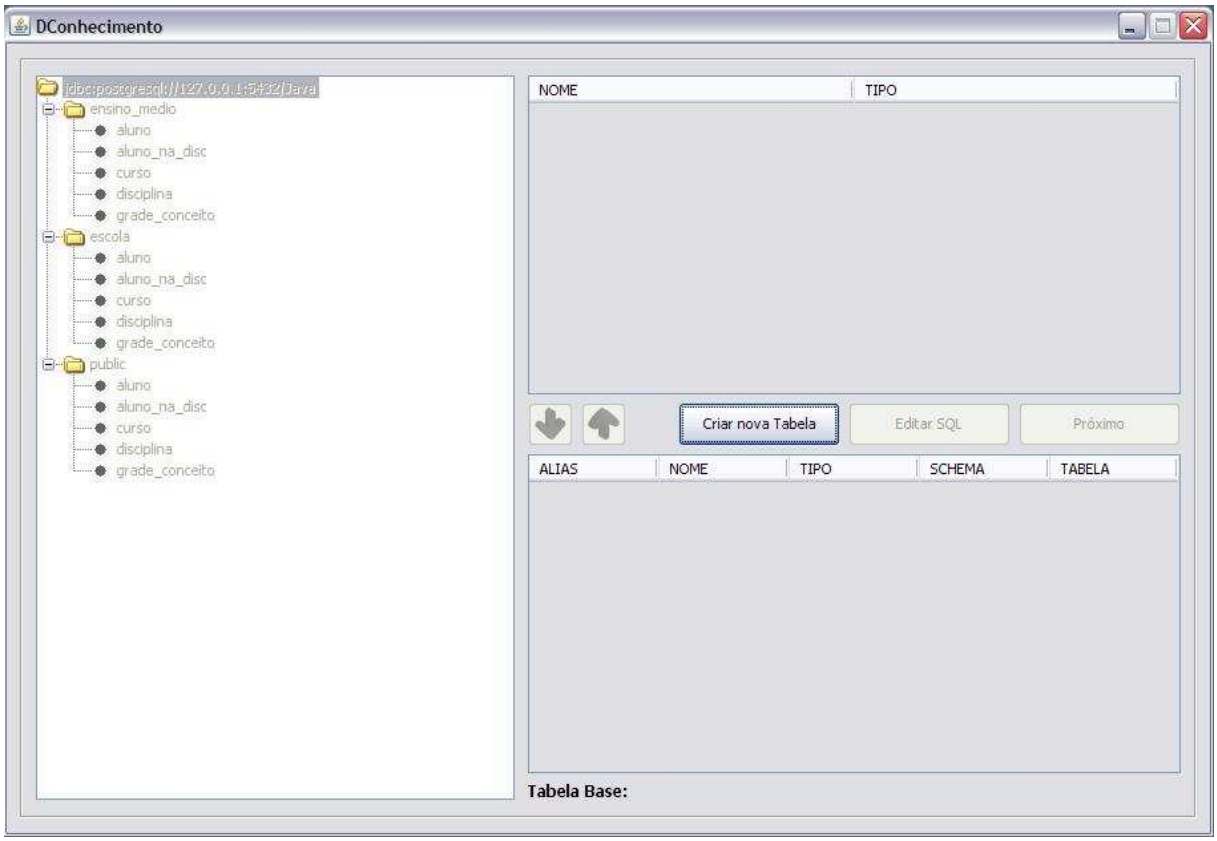

Figura 31 – Tela principal do sistema.

A primeira ação a ser tomada pelo usuário é a definição de sua tabela base, esta será o ponto principal para definir o contexto da nova tabela a ser criada.

Na Figura 32 é apresentada a tela utilizada para a definição da tabela base, os parâmetros utilizados são:

Para banco de dados que utiliza schema.

- Schema.
- Nome da nova tabela.
- Nome da tabela base.

Para Banco de dados que não utilize schema.

- Nome da nova tabela.
- Nome da tabela base.

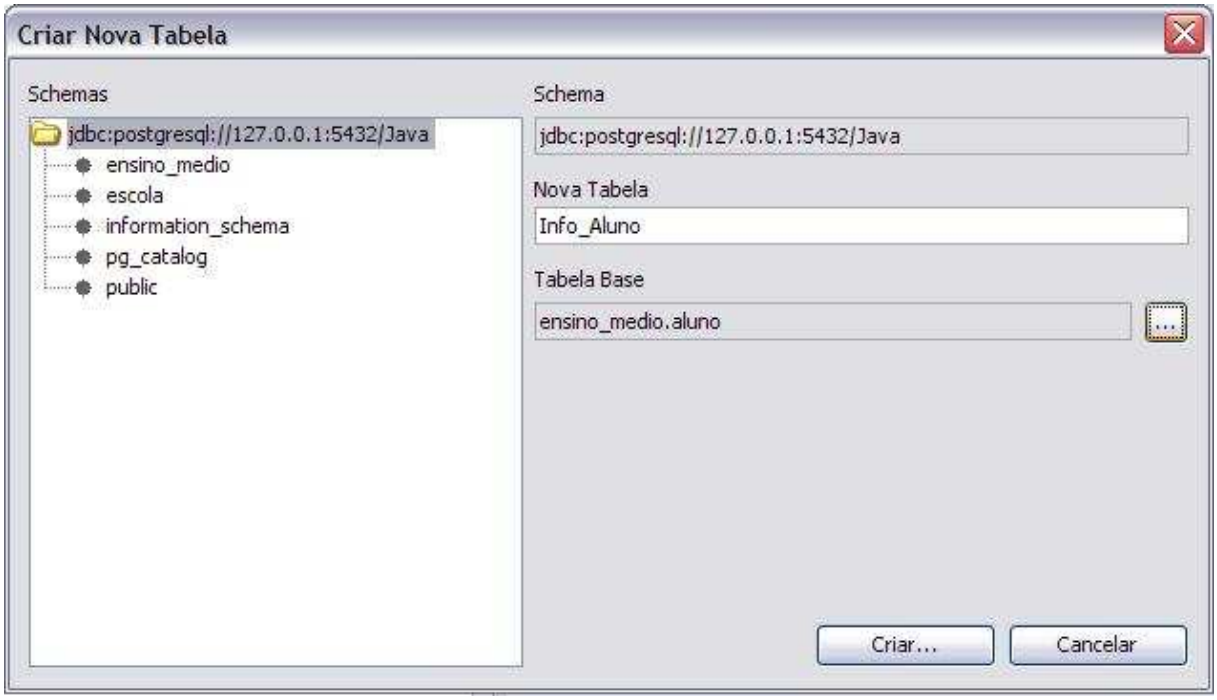

Figura 32 – Tela para definir parâmetros de nova tabela.

Para auxiliar na seleção da tabela base o sistema oferece uma tela apresentando as tabelas existentes no banco de dados, como é mostrado na Figura 33.

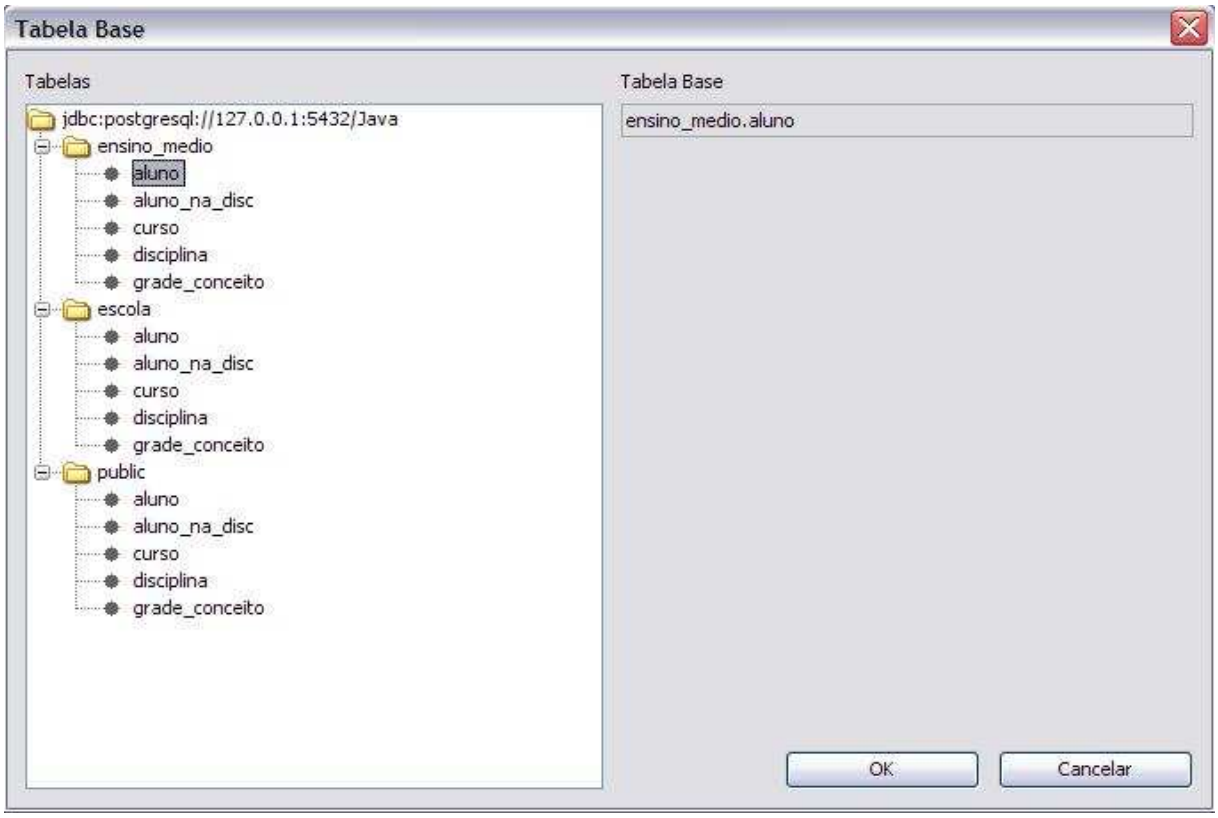

Figura 33 – Tela para definir tabela base.

Após definir a tabela base o usuário pode inserir os campos que sejam relevantes para a sua pesquisa, a tela apresentada na Figura 34, auxilia o usuário na inserção destes campos, o valor do campo inserido pode ser:

- Contagem – Conta todas as ocorrências do campo no contexto.
- Média – Média de todas as ocorrências do campo no contexto.
- Soma – Soma de todas as ocorrências do campo no contexto.
- Exibir – Exibe o campo, e inclui na definição do contexto.
- Referência – Inclui na definição do contexto, sem exibir o campo, quando o valor é passado desta maneira, este tem o objetivo de compor a definição da organização dos dados selecionados.

Para os campos do tipo Contagem, Média e Soma, é possível transformar os valores obtidos nestes agrupamentos, em atributos com a nomenclatura de Baixo, Médio e Alto, seguindo a faixa valor determinada pelo usuário para cada item citado acima.

Figura 34 – Tela de inserção de campos.

Na inserção dos campos o usuário pode se deparar com um campo que é chave estrangeira dentro da tabela selecionada, neste caso ele tem duas opções, que são apresentadas como mostra a Figura 35.

Figura 35 – Mensagem referente à seleção de campo chave estrangeira.

A opção “Inserir Chave Estrangeira”, traz ao usuário a tela de inserção de dados, como foi apresentado na Figura 36. No caso de selecionar a opção “Abrir tabela referenciada”, o sistema retornará a tabela que é referenciada pela chave estrangeira, assim permitindo ao usuário selecionar os campos referentes a esta, como pode ser visto na Figura 9. Se existir campos selecionados desta tabela, esta tela disponibiliza estes campos, proporcionado ao usuário um retorno do que já foi selecionado.

Figura 36 – Seleção de campos da tabela Chave estrangeira selecionada.

Após inserir os dados, o usuário pode obter algumas informações sobre este, com um clique utilizando o botão direito do mouse sobre o campo, o sistema irá fornecer a opção de remover o campo ou obter informações, como pode ser observado na Figura 37 e Figura 38.

Figura 37 – Clique sobre o campo Nova Tabela.

Figura 38 – Informações sobre campo de Nova Tabela.

Durante o processo de definição dos campos da nova tabela, o sistema oferece um retorno para o usuário, possibilitando uma visualização prévia da nova tabela construída, como pode ser visto na Figura 39.

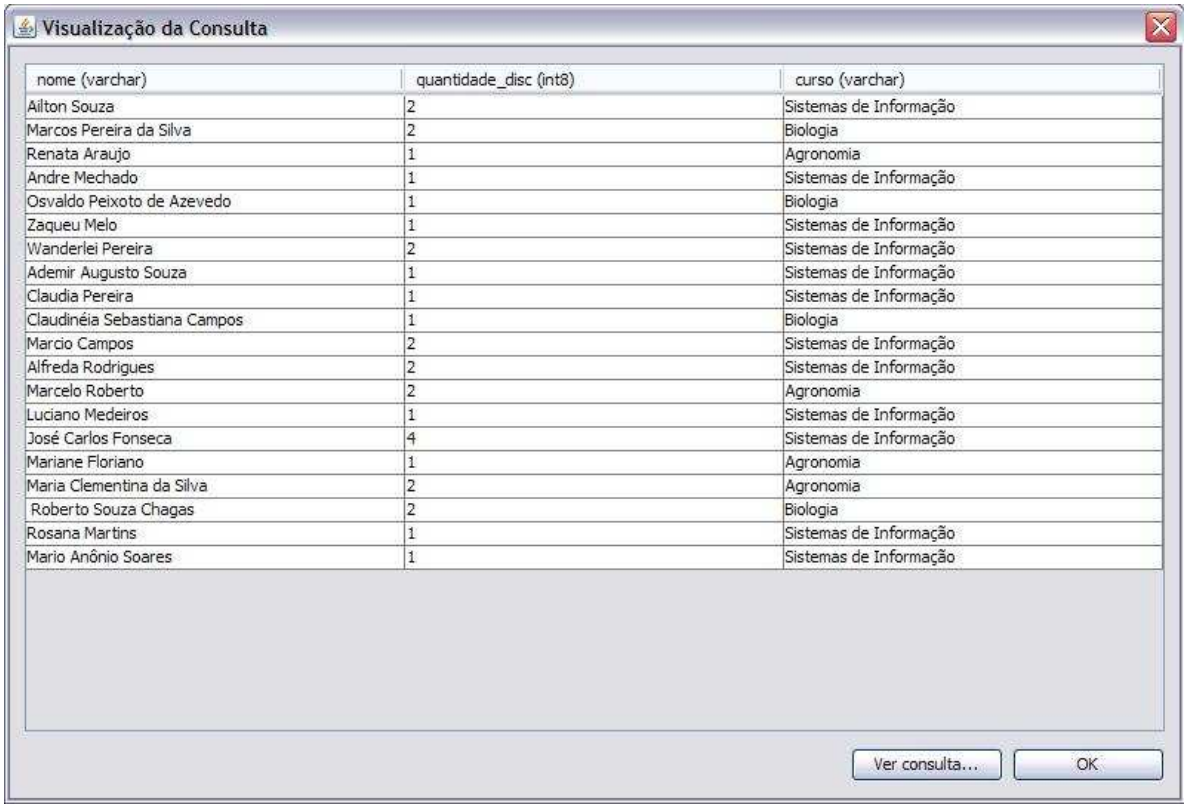

Figura 39 – Visualização da nova tabela.

O sistema oferece um editor de SLQ, que permite ao usuário realizar uma consulta no banco utilizando comandos de SQL, como pode ser visto na Figura 40. Este recurso oferecido pelo sistema está parcialmente construído, tendo suas funcionalidades ainda limitadas, uma vez que o resultado da consulta realizado pelo usuário não pode ser inserido na nova tabela.

Após finalizar o processo de seleção o usuário pode gerar sua Nova Tabela no banco, isto acontece após o comando “Próximo” for acionado. O protótipo reuni as informações necessárias e cria a tabela no banco e apresenta para o usuário uma tela com informações sobre a Nova Tabela gerada, como pode ser visto na Figura 41, com isso as tarefas realizadas com o protótipo são finalizadas.

Editor de Consultas

Executar Inserir

```
select * from public.aluno
```

Operação realizada com sucesso!

matric (bpchar)	nome (varchar)	endereco (varchar)	bairro (varchar)	dt_nasc (date)	sexo (bpchar)
A25	Marcio Campos Belo	Rua 5	Centro	1986-06-15	M
A39	Andrelino Mechado	Rua 2	Centro	1976-06-21	M
A26	Reinaldo Pereira da Silva	Rua 6	Centro	1979-09-11	M
A29	Marcelo Roberto	Rua 9	Centro	1987-06-25	M
A30	Rafaela Gomes	Rua 10	Centro	1978-02-26	M
A31	C□sar Augusto	Rua 11	Centro	1986-10-23	M
A35	Pamela Oliveira	Rua 9	Coisa Mansa	1979-06-01	F
A38	Renata Pastre	Rua 8	Centro	1980-08-20	F
A21	Jos□ Carlos Fonseca	Rua 1	Bairro Pedra Limpa	1987-03-25	M
A27	Gen□zio Roberto Belo	Rua 7	Bairro Pedra Limpa	1980-04-12	M
A37	Mriana Ferreira	Rua 7	Bairro Pedra Limpa	1977-07-03	F
A22	Ailton Santos	Rua 2	Vila Galdino	1978-11-26	M
A24	Alfreda Vieira	Rua 4	Vila Galdino	1989-02-28	F
A28	Maria Clementina da Silva	Rua 8	Vila Galdino	1988-03-25	F
A34	Luciano Cruz	Rua 2	Vila Galdino	1979-04-21	M
A40	Cristiane Matias	Rua 8	Vila Galdino	1988-05-24	F
A23	Wanderson Pereira	Rua 3	Vila Esperanca	1986-04-23	M
A32	Luiz Carlos Braz	Rua 9	Vila Esperanca	1987-08-11	M
A33	Zaqueu Melo Panta	Rua 2	Vila Esperanca	1978-11-05	M
A36	Angelina Gomes	Rua 8	Vila Esperanca	1977-05-01	F

Figura 40 – Editor de SQL.

DConhecimento

Nome da Nova Tabela: public.Aluno__base

Tabela Base: public.aluno

nome (varchar)	quantidade_disc (int8)	curso (varchar)
Ailton Souza	2	Sistemas de Informação
Marcos Pereira da Silva	2	Biologia
Renata Araujo	1	Agronomia
Andre Mechado	1	Sistemas de Informação
Osvaldo Peixoto de Azevedo	1	Biologia
Zaqueu Melo	1	Sistemas de Informação
Wanderlei Pereira	2	Sistemas de Informação
Ademir Augusto Souza	1	Sistemas de Informação
Claudia Pereira	1	Sistemas de Informação
Claudinéia Sebastiana Campos	1	Biologia
Marcio Campos	2	Sistemas de Informação
Alfreda Rodrigues	2	Sistemas de Informação
Marcelo Roberto	2	Agronomia
Luciano Medeiros	1	Sistemas de Informação
José Carlos Fonseca	4	Sistemas de Informação
Mariane Floriano	1	Agronomia
Maria Clementina da Silva	2	Agronomia
Roberto Souza Chagas	2	Biologia
Rosana Martins	1	Sistemas de Informação
Mario Anônio Soares	1	Sistemas de Informação

Concluir

Figura 41 – Informações sobre a tabela criada.

5. CONCLUSÕES E TRABALHOS FUTUROS

Diante da grande importância empregada a decisões tomadas com base em um conjunto de dados, a proposta deste protótipo é tentar agilizar e proporcionar ao administrador uma visão mais simplificada na seleção de dados relevantes para realizar a aplicação do processo de descoberta de conhecimento. Esta ferramenta consiste em uma aplicação com interfaces de fácil assimilação ao usuário, e operações com o banco de dados utilizado uma navegação simplificada com poucos movimentos, procurando assim diminuir a faixa de erros e alcançar uma maior precisão no novo conjunto de dados definido, e realizar a preparação para que seja aplicado o Algoritmo Genético Difuso Multiobjetivo Para Descoberta De Conhecimento.

O protótipo fornece ao usuário a opção de se conectar com diferentes bancos, sem a necessidade de utilizar ferramentas próprias para cada banco, procurando generalizar as ações realizadas pelo usuário de maneira a focar o processo de seleção somente na questão do contexto a se definir, e não com realizar tal operação com comandos específicos de Banco de Dados.

O protótipo já permite ao usuário definir um contexto e selecionar os dados que sejam relevantes para a sua pesquisa. Para que esta ferramenta esteja completa é preciso implementar a discretização e os tratamentos necessários para atender os requisitos para que seja possível a aplicação do Algoritmo Genético Difuso Multiobjetivo Para Descoberta De Conhecimento.

Dentre as sugestões para trabalhos futuros, pode-se colocar o desenvolvimento de aplicações para as outras etapas do Processo de Descoberta de Conhecimento, procurando unificar estas ferramentas e gerar uma ferramenta completa para o Processo de Descoberta de Conhecimento, que atenda todas as suas etapas chegando ao final desse processo fornecendo informações de grande potencial.

6 - REFERÊNCIAS

BATISTA, G. E. A. P. A. Pré-processamento de Dados em Aprendizado de Máquina Supervisionado. Serviço de Pós-graduação do ICMC-USP, São Carlos - SP. Tese de doutorado, USP, 2003.

BOENTE, A. N. P.; OLIVEIRA, F. S. G.; ROSA, J. L. A.. SEGeT – Simpósio de Excelência em Gestão e Tecnologia. 2007.

CARVALHO, F. P; JUNIOR, A. F; SILVEIRA, J. G. KDD – NMS Um Sistema de Descoberta de Conhecimento e Mineração em Bases de Dados de Sistemas de Gerência de Redes. PUCRS – Porto Alegre. 2003.

COELHO, R. G. Um Algoritmo Genético Difuso Multiobjetivo Para Descoberta De Conhecimento. Dissertação de Pós-Graduação em Ciência da Computação, Universidade Estadual de Maringá, 2004.

CORREIA, AMÉRICO ZUCCOLOTTO. Comparação entre os Paradigmas Imperativos e Orientado a objetos. Universidade do Vale do Rio dos Sinos, 2007.

FAYYAD, U.M.; PIATETSKY-SHAPIRO, G. & SMYTH, The KDD Process for Extracting Useful Knowledge from Volumes of Data, Communications of the ACM. November 1996.

GUSTAVO, E. A. P. BATISTA, A. MONARD. Uma Proposta Para O Tratamento de Valores desconhecidos Utilizando o Algoritmo K-Vizinhos mais Próximos. Universidade de São Paulo/ILTC. 2001.

RABELO, E. Avaliação de Técnicas de Visualização para Mineração de Dados. Dissertação de Pós-Graduação em Ciência da Computação, Universidade Estadual de Maringá. Maringá, 2007.

REZENDE, SOLANGE OLIVEIRA. Sistemas Inteligentes – Fundamentos e Aplicações. Editora Manole Ltda, 2005.

RICARTE, IVAN LUIZ MARQUES.Programação Orientada a Objetos: Uma Abordagem com Java. Universidade Estadual de Campinas, 2001.

ROMÃO, W. Descoberta de conhecimento relevante em banco de dados sobre ciência e tecnologia. Programa de Pós-Graduação em Engenharia de Produção, Florianópolis – SC. Tese de doutorado, UFSC, 2002.

SANTOS, R. S. Ambiente para Extração de Informação através de Mineração das Bases de Dados do Sitema Único de Saúde. Tese de Doutorado da Escola de Medicina, Universidade Federal de São Paulo. São Paulo. 2007.

SOARES. J. A. Pré-Processamento em Mineração de Dados: Um Estudo Comparativo em Complementação. Resumo da Tese de Doutor em Ciências (D. Sc.). Universidade Federal do Rio de Janeiro, COPPE. 2007.

7 - Anexo

ANEXO 1 – Mídia com o código fonte do Protótipo de uma Ferramenta para Realização da etapa de Pré-Processamento no Processo de Descoberta de Conhecimento não trivial em base de dados