

**UNIVERSIDADE FEDERAL DE SERGIPE
CAMPUS ALBERTO CARVALHO
DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO**

CAMILA CARVALHO OLIVEIRA

**UMA FERRAMENTA PARA DETECÇÃO E ANÁLISE DE
IMPACTOS NA EVOLUÇÃO DE ESQUEMAS DE DADOS
EM SGBDs RELACIONAIS**

**ITABAIANA
2015**

**UNIVERSIDADE FEDERAL DE SERGIPE
CAMPUS ALBERTO CARVALHO
DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO**

CAMILA CARVALHO OLIVEIRA

**UMA FERRAMENTA PARA DETECÇÃO E ANÁLISE DE
IMPACTOS NA EVOLUÇÃO DE ESQUEMAS DE DADOS
EM SGBDs RELACIONAIS**

Trabalho de Conclusão de Curso
submetido ao Departamento de Sistemas
de Informação da Universidade Federal de
Sergipe, como requisito parcial para a
obtenção do título de Bacharel em
Sistemas de Informação.

Orientador: Msc. ANDRÉ VINICIUS RODRIGUES PASSOS NASCIMENTO
Coorientador: Dr. METHANIAS COLAÇO RODRIGUES JÚNIOR

**ITABAIANA
2015**

OLIVEIRA, Camila Carvalho.

Uma ferramenta para detecção e análise de impactos na evolução de esquemas de dados em SGBDs relacionais/ Camila Carvalho Oliveira ó Itabaiana: UFS, 2015. 89f.

Trabalho de Conclusão de Curso em Bacharel em Sistemas de Informação ó Universidade Federal de Sergipe, Curso de Sistemas de Informação, 2015.

1. Evolução de Esquema.
2. Banco de Dados.
3. Sistemas de Informação.

I. Uma ferramenta para detecção e análise de impactos na evolução de esquemas de dados em SGBDs relacionais

CAMILA CARVALHO OLIVEIRA

**UMA FERRAMENTA PARA DETECÇÃO E ANÁLISE DE
IMPACTOS NA EVOLUÇÃO DE ESQUEMAS DE DADOS EM
SGBDs RELACIONAIS**

Trabalho de Conclusão de Curso submetido ao corpo docente do Departamento de Sistemas de Informação da Universidade Federal de Sergipe (DSIITA/UFS) como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Itabaiana, 25 de fevereiro de 2015.

BANCA EXAMINADORA:

Prof. André Vinicius Rodrigues Passos Nascimento, Mestre
Orientador
DSIITA/UFS

Prof. Methanias Colaço Rodrigues Júnior, Doutor
Coorientador
DSIITA/UFS

Prof. Eugênio Rubens Cardoso Braz, Doutor
DSIITA/UFS

DEDICATÓRIA

Dedico esse trabalho aos meus maiores educadores, minha mãe Cleide e meu pai Rosembergue. Por todo esforço e dedicação sem medida em me proporcionar a melhor educação, me ensinar a sempre buscar ser uma pessoa melhor e a alcançar meus objetivos. Amo vocês, obrigada por todo amor e carinho.

Ao meu irmão Johansson, obrigada por me apresentar ao curso e me incentivar nessa trajetória. À todos os amigos que conquistei e me conquistaram no curso, obrigada por todo o apoio e companheirismo. E à minha prima irmã Sthefanie, te agradeço pela grande amizade e parceria. Sem o amor e suporte de vocês eu não chegaria tão longe (ou pelo menos não teria graça). Amo vocês!

AGRADECIMENTOS

Agradeço primeiramente e acima de tudo a Deus. Ele é a fonte da minha vida, força, sabedoria e coragem, e graças a Ele cheguei ao final com alegria. Agradeço meu Deus por teu amor e fidelidade.

Aos meus pais agradeço pelo apoio e crédito ao meu esforço e capacidade. Que, mesmo sem entender o que faço, sempre me incentivaram e ajudaram. Sem eles eu não estaria me formando. Amo muito vocês.

Ao meu irmão, obrigada por me aguentar quando eu estava sem paciência. Por me ensinar e acreditar que eu também o podia ajudar. Por todas as loucuras que só a gente entende. Te amo. E a minha cunhada, obrigada por todo carinho e esforço sem medida.

Quero agradecer a todos da minha família Carvalho e Oliveira. Aos meus avós, tios e primos, obrigada por serem o que são e como são, por cada risada e repreensão, por me mostrarem o caminho da verdade e por todos os conselhos. Agradeço a Deus por amar e ser amada por vocês.

Lidianny e Sthefanie, o que falar de vocês sem chorar ou rir com vontade, sem lembrar-se de cada aprendizado e segredo compartilhado, dos momentos difíceis em que nos fortalecemos em Deus e umas nas outras? E Sthefanie, minha companheira de vida, de curso, de família, de todo momento, você que não me deixou ter o mês só pra mim, obrigada por ser minha irmã. Amo vocês, loira e morena.

Agradeço a todos os meus amigos de curso por cada período, cada ajuda, pelo épico congresso, por todas as choradas coletivas em provas e trabalhos, enfim, obrigada a todos os õParanauês de SIõ. Em especial, agradeço a Rodrigo, Thauane, Leonardo e Charles pela amizade, vocês foram importantes em cada momento. E aos componentes do grupo õQuentinhaõ só digo uma coisa: vão continuar ao meu lado e acabou.

Às Luluzinhas agradeço pela amizade, por cada reunião, cada esforço para estarmos juntas. E Vanessa, amiga de sempre, obrigada pela compatibilidade, por ser amiga em todo o tempo, e pelo que ainda há de vir.

Enfim, a todos os amigos que fazem parte da minha vida (mesmo que competindo para saber qual o melhor curso) obrigada por me encontrarem.

Aos meus professores agradeço por todo ensinamento, compreensão e incentivo a nunca desistirmos. E agradeço ao meu orientador, prof. Msc. André Vinícius Rodrigues Passos Nascimento. Obrigada por ter sido um grande orientador e por todo o carinho. Além de ter sido também um psicólogo.

Por tudo isso só tenho a agradecer e a glorificar a Deus. Obrigada Senhor!

OLIVEIRA, Camila Carvalho. **Uma ferramenta para detecção e análise de impactos na evolução de esquemas de dados em SGBDs relacionais**. 2015. Trabalho de Conclusão de Curso ó Curso de Sistemas de Informação, Departamento de Sistemas de Informação, Universidade Federal de Sergipe, Itabaiana, 2015.

RESUMO

Uma das principais atividades relacionadas com a construção e manutenção de sistemas de informação é o ajuste de artefatos em função de mudanças que surgem com o passar do tempo. Um tipo particular de mudança que afeta diretamente o código das aplicações é a evolução de esquemas de banco de dados. Modificar um esquema implica modificar os sistemas que dependem do banco de dados. Para isso é necessário avaliar o impacto da evolução de um esquema, ou seja, identificar todos os locais do sistema que podem ser afetados por cada mudança no esquema, para adaptá-los à nova realidade. Realizar manualmente a detecção de impactos é uma tarefa difícil, que consome tempo e custo e pode causar prejuízos. Neste trabalho propomos uma ferramenta para auxiliar a detecção e análise de impactos na evolução de esquemas de dados em SGBDs relacionais. Através de uma abordagem sistemática para relacionar artefatos e objetos de esquemas, a ferramenta avalia, através da comparação de *baselines*, o impacto que mudanças em um esquema de dados podem causar no ambiente de um sistema de informação.

Palavras-chave:

Evolução de Esquema, Avaliação de Impacto, Sistemas de Informação.

ABSTRACT

One of the main activities related to the construction and maintenance of information systems is the artifacts adjustment due to changes that occur over time. A particular type of change that directly affects the application code is the evolution of database schemas. Modifying a schema implies modifying systems that rely on the database. For this, it is necessary to evaluate the impact of the evolution of a scheme, in other words, identify all system locations that may be affected by every change in the scheme, to adapt them to the new reality. Manually performing the detection of impacts is a difficult task that takes time and costs and can cause damage. In this paper, we propose a tool to aid the detection and analysis of impacts in the evolution of data schemas in relational DBMSs. Through a systematic approach to relate artifacts and schema objects, the tool evaluates, by comparing baselines, the impact that changes in a data schema may cause in the environment of an information system.

Keywords:

Evolution Scheme, Impact Assessment, Information Systems.

LISTA DE FIGURAS

Figura 1: Diagrama de casos de uso da ferramenta	25
Figura 2: Diagrama de classes do caso de uso Manter Esquemas ó Cadastrar Esquema	42
Figura 3: Diagrama de classes do caso de uso Manter Esquemas ó Cadastrar Baseline	44
Figura 4: Diagrama de classes do caso de uso Manter Esquemas ó Editar Esquema.....	45
Figura 5: Diagrama de classes do caso de uso Manter Esquemas ó Importar Esquema	48
Figura 6: Diagrama de classes do caso de uso Manter Artefatos - Sistema.....	49
Figura 7: Diagrama de classes do caso de uso Manter Artefatos ó Módulo	50
Figura 8: Diagrama de classes do caso de uso Manter Artefatos - Artefato	52
Figura 9: Diagrama de classes do caso de uso Manter Fonte de Dados	53
Figura 10: Diagrama de classes do caso de uso Manter Dependência entre Artefatos e Esquemas.....	55
Figura 11: Diagrama de classes do caso de uso Manter Mudanças de Esquemas.....	57
Figura 12: Diagrama de classes do caso de uso Avaliar Impacto de Mudanças em Esquemas.....	59
Figura 13: Diagrama de classes do caso de uso Gerar Relatórios	60
Figura 14: Esquema de dados da ferramenta ó ãEsquemaã	62
Figura 15: Esquema de dados da ferramenta ó ãArtefatoã.....	63
Figura 16: Esquema de dados da ferramenta ó ãFonte de dadosã.....	63
Figura 17: Esquema de dados da ferramenta ó ãArtefato afetadoã	64
Figura 18: Esquema de dados da ferramenta ó ãMudanças de esquemaã.....	65
Figura 19: Esquema de dados da ferramenta.....	66
Figura 20: Tela do Menu Principal da ferramenta.....	68
Figura 21: Tela de Conexão da ferramenta	69
Figura 22: Tela para criação de Fonte de Dados da ferramenta	70
Figura 23: Tela de cadastro de Esquema da ferramenta	71
Figura 24: Tela para criação de Baseline da ferramenta	72
Figura 25: Tela para Importar Esquema da ferramenta	73
Figura 26: Tela para alterar Fonte de Dados da ferramenta.....	74
Figura 27: Tela para Editar Esquema da ferramenta	74
Figura 28: Tela para cadastrar Sistema da ferramenta.....	75
Figura 29: Tela para cadastrar Módulo da ferramenta.....	76
Figura 30: Tela para cadastrar Artefato da ferramenta	77
Figura 31: Tela para vincular Artefatos e Elementos da ferramenta	78
Figura 32: Tela para desvincular Artefatos de Elementos da ferramenta	79
Figura 33: Tela para comparar Baselines da ferramenta.....	80
Figura 34: Tela para avaliar impacto da ferramenta	81
Figura 35: Tela para gerar relatório de diferenças de baselines da ferramenta.....	82
Figura 36: Relatório de diferenças gerado pela ferramenta	82
Figura 37: Tela para gerar relatório de impacto da ferrmaneta	83
Figura 38: Relatório de Impacto gerado pela ferramenta.....	84

LISTA DE QUADROS

Quadro 1: Requisitos funcionais e não funcionais da ferramenta.	24
Quadro 2: Caso de Uso Autenticar Usuário	27
Quadro 3: Caso de Uso Manter Esquemas ó Criar Esquema	28
Quadro 4: Caso de Uso Manter Esquemas ó Editar Esquema.....	29
Quadro 5: Caso de Uso Manter Esquemas ó Cadastrar Baseline	29
Quadro 6: Caso de Uso Manter Esquemas ó Importar Esquema	30
Quadro 7: Caso de Uso Manter Artefato ó Cadastrar Sistema.....	30
Quadro 8: Caso de Uso Manter Artefato ó Editar Sistema	31
Quadro 9: Caso de Uso Manter Artefato ó Remover Sistema.....	31
Quadro 10: Caso de Uso Manter Artefato ó Cadastrar Módulo	32
Quadro 11: Caso de Uso Manter Artefato ó Editar Módulo	32
Quadro 12: Caso de Uso Manter Artefato ó Remover Módulo.....	33
Quadro 13: Caso de Uso Manter Artefato ó Cadastrar Módulo	33
Quadro 14: Caso de Uso Manter Artefato ó Editar Artefato.....	34
Quadro 15: Caso de Uso Manter Artefato ó Remover Artefato.....	34
Quadro 16: Caso de Uso Manter Fonte de Dados ó Cadastrar Fonte de Dados.....	35
Quadro 17: Caso de Uso Manter Fonte de Dados ó Editar Fonte de Dados	36
Quadro 18: Caso de Uso Manter Fonte de Dados ó Remover Fonte de Dados	36
Quadro 19: Caso de Uso Manter Artefato ó Criar Vinculo.....	37
Quadro 20: Caso de Uso Manter Artefato ó Remover Vínculo	37
Quadro 21: Caso de Uso Manter Mudanças ó Comparar Baselines.....	38
Quadro 22: Caso de Uso Avaliar Impacto de Mudanças em Esquema	39
Quadro 23: Caso de Uso Gerar Relatórios ó Gerar Relatórios de Diferenças entre Baselines.....	40
Quadro 24: Caso de Uso Gerar Relatórios ó Gerar Relatórios de Impacto de Mudanças em Esquema.....	41

SUMÁRIO

1. INTRODUÇÃO	12
1.1. Objetivos	14
1.1.1. Geral.....	14
1.1.2. Específicos	14
1.2. Relevância do trabalho	14
1.3. Metodologia	15
1.4. Estrutura do trabalho	15
2. REVISÃO BIBLIOGRÁFICA	17
2.1. Trabalhos relacionados	18
2.1.1. Automating Database Schema Evolution in Information System Upgrades 18	
2.1.2. Visualizing Impacts of Database Schema Changes ó A Controlled Experiment.....	18
2.1.3. Impact Analysis of Database Schema Changes	19
2.1.4. DaSIAn: A Tool for Estimating the Impact of Database Schema Modifications on Web Applications	20
2.2. Trabalhos Relacionados ó Ferramentas	20
2.1.2. DB Change Manager	21
2.2.2. Oracle Enterprise Manager	21
2.2.3. Find it EZ Software	22
3. PROJETO DA FERRAMENTA.....	23
3.1. Requisitos funcionais e não funcionais.....	23
3.2. Casos de Uso	24
3.2.1. Casos de Uso ó Nível de Usuário	24
3.2.2. Casos de Uso ó Nível de Sistema	26
3.2.2.1. Caso de Uso Autenticar Usuário.....	27
3.2.2.2. Caso de Uso Manter Esquemas	27
3.2.2.3. Caso de Uso Manter Artefatos.....	30
3.2.2.4. Caso de Uso Manter Fontes de Dados	35
3.2.2.5. Caso de Uso Manter Relação de Dependência entre Artefatos e Esquemas	36
3.2.2.6. Caso de Uso Manter Mudanças de Esquemas	37
3.2.2.7. Caso de Uso Avaliar Impacto de Mudanças em Esquema.....	38
3.2.2.8. Caso de Uso Gerar Relatórios	39
3.3. Diagrama de Classes.....	41
3.3.1. Caso de uso Manter Esquemas	41
3.3.2. Caso de uso Manter Artefatos	48
3.3.3. Caso de uso Manter Fonte de Dados	53
3.3.4. Caso de uso Manter Dependência entre artefatos e esquemas	54
3.3.5. Caso de uso Manter Mudanças de Esquemas.....	56
3.3.6. Caso de uso Avaliar Impacto de Mudanças em Esquemas	58
3.3.7. Caso de uso Gerar Relatórios	59

3.4. Modelagem de Dados	60
3.4.1. Componentes do modelo de dados	61
4. FERRAMENTA.....	67
5. CONCLUSÕES	85
REFERÊNCIAS	87

1. INTRODUÇÃO

Um dos principais problemas de construir e manter um sistema de informação de médio ou grande porte é a necessidade de mudanças que inevitavelmente surgem ao longo do tempo. Um tipo particular de mudança que pode ter consequências graves para o resto dos sistemas de aplicação é a evolução de esquemas de banco de dados (SJOBERG, 1993).

Projetar um esquema de banco de dados é uma atividade evolutiva que requer várias iterações e, mesmo depois de ser concluído, o esquema de banco de dados continua a mudar ao longo do tempo (ZANIOLO et al. 1997). Um esquema de banco de dados precisa estar em evolução, a fim de se adaptar às mudanças de requisitos, corrigir erros e melhorar o seu desempenho. Embora a evolução do esquema torne mais fácil o entendimento e manutenção do banco de dados, por outro lado, é um risco que muitas empresas preferem evitar.

Segundo (CAMOLESI; TRAINA, 1996 apud MORAES, 2009), a evolução de esquemas de banco de dados se faz necessária por diferentes motivos, dentre os quais se destacam:

- a) Correção do esquema de dados diante de problemas encontrados durante a fase de operação com o banco de dados (falta de testes ou falhas humanas);
- b) Adaptação do esquema de dados a novos componentes ou circunstâncias que envolvem o sistema de banco de dados e seu ambiente de utilização (interfaces ou processos);
- c) Refinamento de componentes do esquema que foram um pouco detalhados durante a fase de projeto, e, portanto, estavam semanticamente pobres (especificação deficiente);
- d) Atualização de componentes do esquema de modo a acompanharem as constantes modificações de requisitos dos usuários;
- e) Experimentação de novas concepções do esquema de dados em momentos quando se buscam alternativas para a modelagem do banco de dados; e
- f) Incorporação de novos elementos ao esquema de dados.

A evolução de um esquema envolve a modificação da estrutura do banco de dados, o que traz uma problemática para os desenvolvedores e empresas em relação aos efeitos de cada mudança sobre as aplicações ao redor do sistema. Quando uma mudança de esquema ocorre, a consistência entre o esquema, o banco de dados e as aplicações deve permanecer (KARAHASANOVIC; SJØBERG, 2001). A complexidade envolvida, o custo e o tempo utilizados para conciliar a mudança de esquema com todo o sistema, dificultam o processo de evolução de esquemas de banco de dados.

Uma pequena mudança no esquema de banco de dados, como, por exemplo, a remoção de um atributo em uma tabela, pode causar um grande impacto em partes do próprio esquema, nos dados do banco de dados e nas aplicações que dependem do banco de dados. Consultas e dados de formulários podem ser invalidados, programas de aplicação podem falhar (resultando na falha total do fluxo de trabalho), e várias páginas de sistemas Web podem tornar-se invisíveis (PAPASTEFANATOS et al. 2007).

Muitas vezes a avaliação dos efeitos do impacto é feita manualmente pelos administradores de banco de dados (DBAs), tarefa essa que é um tanto frágil e difícil (LAW; ROTHERMEL, 2003), e pode estar frequentemente incorreta (LINDVAL; SANDAHL, 1998). Antes da mudança de esquema ser realizada, o DBA precisa analisar a viabilidade da mesma. Se a mudança tem grande impacto sobre as aplicações, a melhor opção pode ser deixar o esquema inalterado, a fim de evitar efeitos negativos sobre as aplicações e, conseqüentemente, os custos da reparação. Como solução muitas empresas criam um novo esquema para reparar os problemas existentes, e, como consequência, criam visões unindo os dois esquemas, o que torna a leitura do banco de dados difícil e confusa. Se a alteração é viável, os DBAs precisam localizar as áreas afetadas em todo o sistema e conciliá-los com o novo esquema. Da mesma forma, não encontrar todos os locais afetados pode trazer efeitos negativos sobre as aplicações.

A evolução de esquema ainda é uma tarefa complexa que consome muito tempo, pois os DBAs não possuem ferramentas automáticas para gerenciar e automatizar este esforço para (i) prever e avaliar os efeitos das alterações no esquema proposto, (ii) reescrever consultas e aplicações para operar no novo esquema, e (iii) migrar a base de dados (CURINO et al. 2009). Esse fato transforma a necessidade de evolução de esquema um grande obstáculo para as empresas e administradores de banco de dados.

Além disso, o problema torna-se mais grave com o crescimento de sistemas de informação web e dos bancos de dados de projetos de *big science*, onde o rápido

avanço do conhecimento e a participação ativa de muitos interessados têm acelerado o ritmo de evolução dos esquemas (CURINO et al. 2009).

O objetivo desse trabalho é projetar e implementar uma ferramenta para auxiliar a detecção e análise de impactos na evolução de esquemas de dados. A ferramenta facilitará o processo de avaliação do impacto de uma mudança de esquema de dados, identificando partes dos sistemas dependentes que possam ser afetados.

Esperamos, com esse trabalho, reduzir o número de erros na tarefa de identificação de áreas impactadas pela mudança de esquema e auxiliar o processo de evolução de esquema.

1.1. Objetivos

1.1.1. Geral

O objetivo desse trabalho é projetar e implementar uma ferramenta para auxiliar a detecção e análise de impactos na evolução de esquemas de dados.

1.1.2. Específicos

- Realizar revisão bibliográfica sobre evolução de esquema de dados;
- Realizar levantamento de requisitos;
- Projetar a ferramenta de avaliação de impacto de evolução de esquema;
- Desenvolvimento da ferramenta de avaliação do impacto de evolução de esquema.

1.2. Relevância do trabalho

O presente trabalho apresenta uma proposta de ferramenta para a detecção e análise de impacto na evolução de esquemas de dados. Apesar de existirem outros

trabalhos com propostas de ferramentas com o mesmo objetivo, as ferramentas não são utilizadas na prática. Os resultados parciais de um survey realizado em paralelo a esse trabalho demonstram claramente que um alto percentual das empresas consultadas não faz avaliação de impacto e, quando realizam, o fazem de forma manual. Esse resultado, ainda que preliminar, demonstra a carência de ferramentas dessa natureza que possam ser efetivamente utilizadas.

1.3. Metodologia

Inicialmente, será realizada a revisão da literatura e pesquisa bibliográfica sobre impactos na evolução de esquemas de dados.

Após a revisão da literatura, será feito o levantamento de requisitos. Serão levantados requisitos funcionais e não funcionais que devem descrever todos os aspectos significativos do sistema.

Após o levantamento de requisitos, a ferramenta será projetada. Inicialmente será modelado o diagrama de casos de uso, em nível de usuário e em nível de sistema. Para auxiliar na análise e melhoria dos dados do sistema, será realizada a modelagem de dados. Adicionalmente, serão construídos diagramas de classe para representar a estrutura das classes que o sistema precisa e as relações entre as mesmas.

Após o projeto, será realizado o desenvolvimento da ferramenta, de forma iterativa e incremental. Cada incremento será responsável por uma funcionalidade do sistema.

1.4. Estrutura do trabalho

O restante do trabalho está estruturado como segue. No capítulo 2 são apresentados os principais artigos e ferramentas encontrados relacionados com o tema do nosso trabalho. No capítulo 3, são descritas as principais atividades realizadas durante o projeto da Ferramenta, levantamento de requisitos, diagrama de casos de uso, diagramas de classes e esquema de dados lógico. O capítulo 4 descreve um estudo de

caso para apresentação das funcionalidades da ferramenta. Finalmente, no capítulo 5 são apresentadas as conclusões do trabalho.

2. REVISÃO BIBLIOGRÁFICA

Ao longo do tempo é inevitável que mudanças de requisitos ocorram em sistemas, principalmente nos de grande porte. Para garantir que o sistema reflita a nova realidade com maior precisão, é preciso que os esquemas de banco de dados sejam atualizados.

Um esquema de banco de dados precisa estar em evolução, a fim de se adaptar às mudanças de requisitos, corrigir erros e melhorar o seu desempenho. Evolução de esquema consiste em modificar o esquema, preocupando-se com a consistência dos dados. Portanto, quando uma mudança de esquema ocorre, a consistência entre o esquema, o banco de dados e as aplicações deve permanecer (KARAHASANOVIC; SJØBERG, 2001).

A evolução de um esquema de dados implica mudanças em todos os níveis da aplicação. Uma pequena mudança no esquema de banco de dados pode causar um grande impacto em partes do próprio esquema, nos dados do banco de dados e nas aplicações que dependem do banco de dados. Por exemplo, se uma tabela for removida do esquema, várias classes da aplicação podem ser afetadas por fazerem uso dessa tabela. A complexidade envolvida, o custo e o tempo utilizados para conciliar a mudança de esquema com todo o sistema, dificultam o processo de evolução de esquemas de banco de dados.

Muitas vezes a avaliação dos efeitos do impacto é feita manualmente pelos administradores de banco de dados (DBAs). Encontrar todas as partes do sistema que possam ser afetadas por cada alteração feita no esquema de dados é uma tarefa difícil, e pode gerar muitos erros. Não realizar essa tarefa com o auxílio de ferramentas automáticas consome tempo e custo para a empresa. Isso significa, para muitas delas, que a avaliação não será feita, o que pode resultar em falhas na aplicação.

Portanto, é de essencial importância uma ferramenta para reduzir o número de erros na tarefa de identificação de áreas impactadas pela mudança de esquema e auxiliar o processo de evolução de esquema.

Nas subseções a seguir serão apresentados alguns trabalhos relacionados à evolução de esquema, bem como ferramentas com proposta semelhante ao nosso trabalho.

2.1. Trabalhos relacionados

Nesta seção apresentaremos algumas propostas que possuem objetivo semelhante ao nosso trabalho: analisar o impacto de mudanças na evolução de esquemas. As propostas apresentadas representam trabalhos moderadamente e fortemente relacionados.

2.1.1. Automating Database Schema Evolution in Information System Upgrades

Em (CURINO et al. 2009) é proposto um projeto que procura desenvolver métodos e ferramentas para automatizar o processo de evolução de esquema de dados. Para atingir este objetivo os autores criaram o sistema PRISM, que fornece uma linguagem de Operadores de Modificação de Esquema (*Schema Modification Operators* - SMO) para expressar de forma concisa as mudanças de esquema, e, ferramentas para avaliar os efeitos de tais mudanças. O objetivo é que os administradores de banco de dados utilizem a nova linguagem para especificar as alterações do novo esquema e, em seguida, faça uso do PRISM para (i) prever e avaliar os efeitos de alterações de esquema, (ii) reescrever consultas antigas, realizar atualizações on-line e adaptar os aplicativos para trabalhar com a nova versão do esquema, (iii) migrar os dados e (iv) documentar todas as mudanças realizadas. Segundo (CURINO et al. 2009), sua proposta reduz efetivamente o custo e o tempo de trabalho de evoluções de esquema de banco de dados.

2.1.2. Visualizing Impacts of Database Schema Changes ó A Controlled Experiment

Em (KARAHASANOVIC; SJØBERG, 2001) é desenvolvida uma ferramenta de visualização para análise de impacto de evolução de esquema de dados em aplicações orientada a objetos. A ferramenta *Schema Evolution Management Tool* (SEMT) recebe como entrada arquivos de código Java e de banco de dados, e, em seguida, identifica as relações entre os componentes (pacotes, classes, interfaces, métodos e campos), os quais são exibidos como um gráfico. Quando o esquema é alterado, SEMT identifica objetos potencialmente afetados e altera a forma deles no gráfico. Foram criadas duas versões da ferramenta: uma versão identifica partes impactadas na granularidade de pacotes, classes e interfaces, enquanto a outra versão identifica as partes afetadas na granularidade mais fina de campos e métodos. Os autores realizaram um experimento controlado com estudantes para avaliar as duas versões da ferramenta em relação à produtividade e satisfação do usuário. Segundo (KARAHASANOVIC; SJØBERG, 2001), os resultados indicaram que identificar os impactos na granularidade mais fina reduz o tempo e os erros cometidos para realizar as mudanças de esquema, e, além disso, os usuários aprovaram a ideia de visualização dos impactos.

2.1.3. Impact Analysis of Database Schema Changes

Em (MAULE et al. 2008) foi desenvolvida uma abordagem para prever o impacto de mudanças de esquema de banco de dados relacionais sobre aplicações orientadas a objetos. Essa abordagem consiste em analisar o fluxo de dados e com os resultados da análise calcular o impacto da evolução de esquema. Os autores utilizaram a análise do fluxo de dados para prever os valores de quaisquer consultas que podem ser executadas no banco de dados. Para obter uma maior precisão de análise sem aumentar o custo computacional, (MAULE et al. 2008) criou um algoritmo para um corte de programa. O objetivo é reduzir o tamanho do programa a ser analisado, mantendo apenas as partes que podem ser afetadas pela mudança de banco de dados. Em seguida, essas partes são analisadas e os resultados, juntamente com as informações do esquema, utilizados para prever os possíveis efeitos de alterações de esquema de banco de dados. Essa tarefa é chamada de Cálculo de Impacto e é feita usando a ferramenta CrocoPat (BEYER et al. 2005). Para realizar as etapas dessa abordagem, os autores criaram um protótipo de sistema chamado SUITE (*Schema Update Impact Tool Environment*). Foi realizado um estudo de caso para avaliar essa abordagem e, segundo os autores, os

resultados mostraram que a precisão da análise de fluxo de dados em trabalhos relacionados é insuficiente, e, que a combinação da análise com o corte de programa produz resultados exatos e precisos em menos de 2 minutos em uma máquina desktop padrão.

2.1.4. DaSIAn: A Tool for Estimating the Impact of Database Schema Modifications on Web Applications

Em (GARDIKIOTIS; MALEYRIS, 2006) foi desenvolvida uma ferramenta, chamada DaSIAn (*Database Schema Impact Analyzer*), para estimar o impacto que modificações de esquema de banco de dados implicam sobre a operacionalidade de uma aplicação web. A proposta dos autores é baseada em uma metodologia que envolve quatro fases. Durante a primeira fase as formas mais comuns de instruções SQL são agrupadas e correlacionadas com as modificações de esquema de banco de dados. A segunda fase inclui a classificação das páginas web entre dependentes (direta ou indiretamente) e independentes de banco de dados. Na terceira fase é feita uma análise mais detalhada de cada página web de acordo com sua correlação com os objetos de banco de dados, incluindo informações como nome e tipo do objeto de banco de dados, tipo de instrução SQL, entre outras. Na última fase as modificações de esquema de banco de dados são aplicadas e seu impacto sobre a aplicação web é estimado em termos do número de páginas web afetadas. Foi realizado um conjunto de experiências para avaliar a eficiência da ferramenta DaSIAn. Segundo (GARDIKIOTIS; MALEYRIS, 2006), na maior parte dos casos a estimativa do número de páginas afetadas foi muito próxima ou igual às páginas reais afetadas. Além disso, os autores afirmam que DaSIAn fornece um conjunto de métricas úteis que indicam a dependência entre o código fonte da aplicação e banco de dados, bem como uma série de estatísticas de uso para cada objeto de esquema de banco de dados.

2.2. Trabalhos Relacionados ó Ferramentas

Nesta seção apresentaremos algumas ferramentas comerciais que possuem objetivo semelhante ao nosso trabalho: analisar o impacto de mudanças na evolução de esquemas.

2.1.2. DB Change Manager

O DB Change Manager é uma ferramenta de gerenciamento de mudança de bancos de dados criada pela empresa Embarcadero. Ele oferece processos de comparação e sincronização automatizadas para simplificar operações de mudança. Apresenta entre suas características a capacidade de identificar e notificar as mudanças nos dados, esquemas e nas configurações do banco de dados e comparar as diferenças resultantes, restaurar objeto a um estado anterior, bem como gerenciar dependências entre os objetos. O DB Change Manager sincroniza bancos de dados, identifica e cria relatórios sobre as mudanças do banco de dados em determinados momentos ou entre momentos específicos. (DB Change Manager. Disponível em: <<https://www.embarcadero.com/br/products/db-change-manager>>. Acesso em: 4 de dezembro de 2014.).

2.2.2. Oracle Enterprise Manager

A Oracle Enterprise Manager é uma ferramenta da Oracle criada para gerenciar servidores de banco de dados Oracle. Dentre as suas funcionalidades, está o gerenciamento de mudanças em bancos de dados. Ela apresenta três principais recursos que permitem aos desenvolvedores e administradores de banco de dados gerenciar as mudanças (Oracle Enterprise Manager. Disponível em: <<http://docs.oracle.com/en/enterprise-manager/>>. Acesso em: 4 de dezembro de 2014.):

- **Baseline ó Define** o banco de dados e seus objetos associados. A ferramenta permite capturar versões de baseline, resultantes de mudanças em banco de dados, em intervalos regulares. Quando uma baseline possui mais de uma versão é possível controlar as alterações às definições de objetos de banco de dados ao longo do tempo e examinar as diferenças entre as versões.

- Comparação ó Apresenta uma lista completa das diferenças entre uma baseline ou um banco de dados e outra baseline ou um banco de dados, ou dois esquemas dentro de um único banco de dados. É possível identificar diferenças de qualquer valor de atributo entre os objetos de qualquer tipo e criar várias versões de uma comparação.
- Sincronização ó Promove as mudanças nas definições de objetos de banco de dados para coincidir com os de outro banco de dados ou um baseline. É possível criar várias versões de sincronização, bem como definir uma sincronização a partir de uma comparação.

Em resumo, a ferramenta identifica, compara e sincroniza mudanças entre bancos de dados ou entre um baseline e um banco de dados.

2.2.3. Find it EZ Software

É uma ferramenta de gerenciamento de mudança de software. A principal função da ferramenta é encontrar artefatos que apresentam determinada expressão. O usuário informa uma expressão e a ferramenta pesquisa arquivos de código-fonte, bancos de dados, relatórios e documentos. Ao final, é produzida uma lista com todos os artefatos que possuem a expressão sendo procurada. (Find it EZ Software. Disponível em: <<http://www.finditez.com/databasedevelopers.aspx>>. Acesso em: 4 de dezembro de 2014.).

3. PROJETO DA FERRAMENTA

Após a revisão bibliográfica, iniciou-se o projeto da ferramenta, no qual foram analisados e definidos quais seriam seus requisitos funcionais e não funcionais. Além disso, foram modelados o digrama de casos de uso e o projeto lógico de dados.

3.1. Requisitos funcionais e não funcionais

Os requisitos funcionais da ferramenta, resumidos no Quadro 1, são: 1) Autenticar usuário. Toda e qualquer manutenção na ferramenta só pode ser efetuada por usuários autenticados; 2) Manter esquemas. Esse requisito inclui criar, alterar, consultar e remover tabelas, colunas, restrições e índices; 3) Importar Esquemas. Esse requisito permite ao usuário conectar-se a outras fontes de dados para importar esquemas diretamente do SGBD; 4) Criar *baselines*. Os *baselines* representam versões de um esquema de dados, utilizadas para controlar mudanças no esquema e possibilitar comparações; 5) Definir fontes de dados, que serão utilizadas para importação de esquemas; 6) Gerenciar artefatos, que são elementos de um sistema que podem sofrer impactos com a evolução de esquema, tais como: códigos, documentações ou testes; 7) Relacionar esquemas e artefatos, indicando de quais elementos do esquema o artefato depende; 8) Gerenciar mudanças de Esquemas, comparando *baselines* e incluindo manualmente as alterações em determinados elementos do esquema; 9) Avaliar o impacto de mudanças, a qual verifica quais artefatos são afetados por cada alteração realizada no esquema. ; 10)Gerar Relatórios de diferenças de *baselines*; 11) Gerar Relatórios de avaliação de impacto.

Requisitos Funcionais
R1: Autenticar usuário
R2: Manter Esquemas (Criar, Editar, Consultar, Remover Esquemas, Tabelas, Colunas, Índices e Restrições).
R3: Importar Esquemas a partir de conexões com SGBDs.
R4: Criar BaseLine de Esquemas.
R5: Definir Fontes de Dados.

R6: Manter Artefatos (Criar, Editar, Consultar, Remover Sistemas, Módulos, Código, Documentação, Casos de Teste, etc.)
R7: Manter relação de dependência entre Artefatos e Esquemas.
R8: Comparar diferenças de Baselines
R8: Avaliar Impacto de Mudanças em Esquemas.
R10: Gerar Relatório de Impacto de Mudanças
R11: Gerar Relatório de Diferenças entre BaseLine e Atualizações.
Requisitos Não-Funcionais
R12: O Sistema só deve permitir acesso através de login e senha.
R13: O Sistema deve ser desenvolvido utilizando a linguagem Java.
R14: O Sistema deve ser desenvolvido para Desktop
R15: O Sistema deve ser executado em qualquer plataforma (Linux, Unix, Windows).
R16: O Sistema deve ter como banco de dados o SQL Server.

Quadro 1: *Requisitos funcionais e não funcionais da ferramenta.*

Além desses requisitos citados anteriormente, a ferramenta foi desenvolvida para ser Desktop, na linguagem Java e utilizar o SQL Server como banco de dados. O Sistema deve ser executado em qualquer plataforma e o seu acesso só deve ser permitido através de usuário e senha.

3.2. Casos de Uso

3.2.1. Casos de Uso ó Nível de Usuário

Após a identificação e análise inicial dos requisitos essenciais para a ferramenta, foi modelado o diagrama de casos de uso, que é mostrado na Figura 1. Com ele, pode-se verificar quais são as funções do sistema.

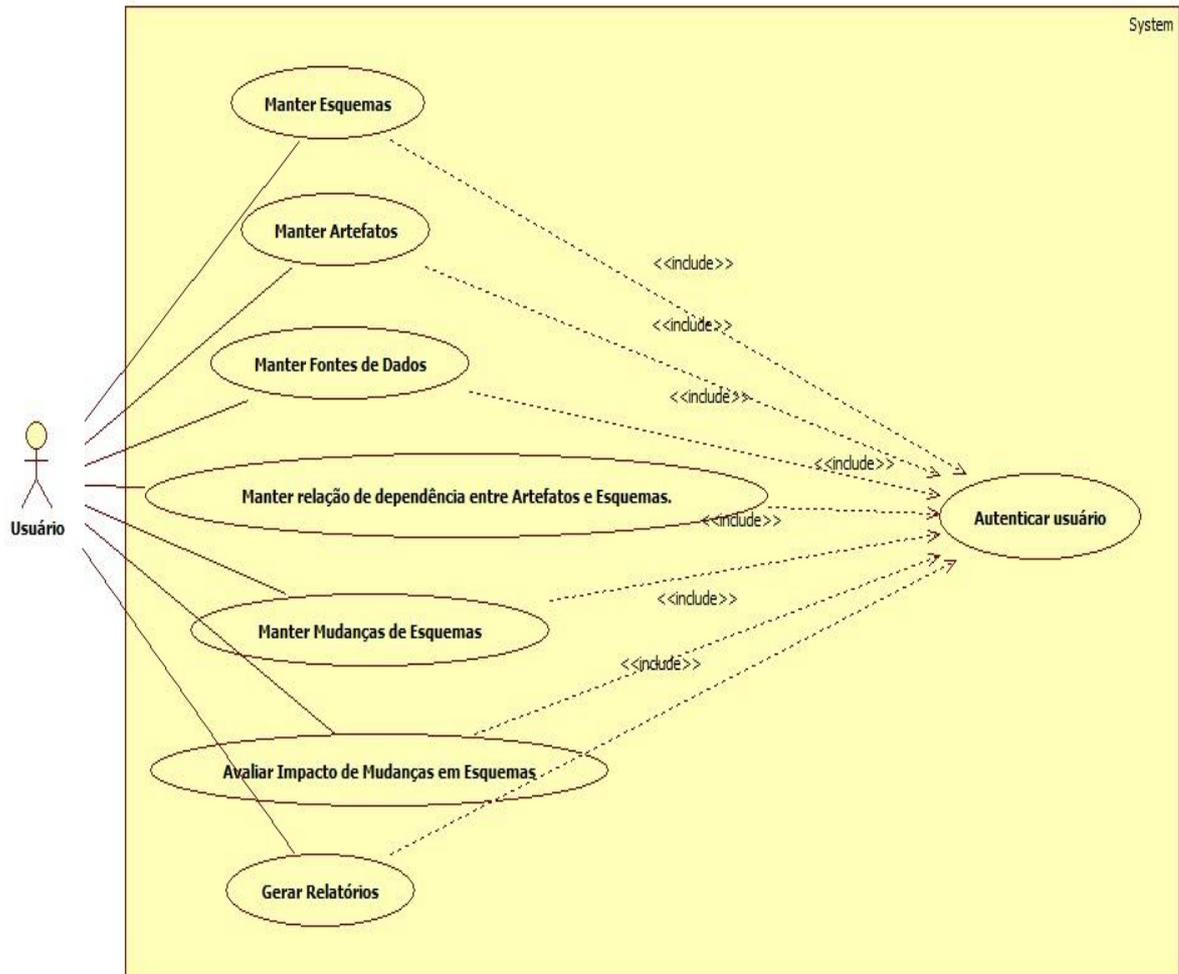


Figura 1: *Diagrama de casos de uso da ferramenta*

O caso de uso Autenticar Usuário valida os dados do usuário, nome de usuário e senha, para permitir seu acesso ao sistema. Somente com a autenticação válida é possível ao usuário acessar as demais funcionalidades da ferramenta. Associa-se ao requisito R1.

O caso de uso Manter Esquemas tem como objetivo gerenciar os dados do Esquema no sistema. O usuário acessa o sistema e nele cadastra o esquema, bem como um *baseline* para o mesmo. Adicionalmente, pode-se alterar, remover, importar e consultar os dados de cada *baseline* do esquema. Possui os requisitos R2, R3 e R4 associados.

O caso de uso Manter artefatos objetiva manter os dados de um sistema que terá relação com o esquema de dados. Um sistema pode possuir módulos, que, por sua vez, possuem artefatos. Esse caso de uso inclui as funcionalidades de cadastrar, editar,

consultar e remover cada sistema, módulo ou artefato. Tem R6 como requisito associado.

Manter Fontes de Dados consiste no registro de conexões com o SGBD relacional, as quais serão utilizadas para importar esquemas e monitorar mudanças. O usuário cadastra os dados para cada conexão necessária e pode, também, alterar e remover. O requisito R5 está associado a esse caso de uso.

Manter relação de dependência entre artefatos e esquemas tem como objetivo criar um vínculo entre os elementos do esquema e os artefatos que podem ser afetados por esse esquema. O sistema permite escolher o artefato e selecionar dentre esquemas, tabelas, colunas, restrições e índices, aqueles dos quais o artefato depende. Adicionalmente, é possível remover os vínculos criados. Tem como requisito associado o R7.

Manter mudanças de Esquemas objetiva cadastrar as alterações realizadas no esquema. Por exemplo, alterar o nome de um atributo de determinada tabela consiste em uma mudança que precisa ser registrada para posteriormente avaliar o impacto sobre os artefatos. Apresenta a opção de comparar *baselines* de um determinado esquema para verificar as diferenças, e, conseqüentemente, analisar a evolução do esquema, bem como a opção de realizar as alterações de forma manual. O requisito R8 está associado.

O caso de uso Avaliar o impacto de mudanças em Esquema consiste na análise de quais artefatos foram afetados por determinada mudança no esquema. O usuário, após registrar as mudanças realizadas em tabelas, colunas, índices ou restrições de um esquema, pode avaliar o impacto de determinada mudança. Está associado ao requisito R9.

Por fim, o caso de uso Gerar Relatórios tem como objetivo gerar um documento com as informações da evolução do esquema e do impacto da mesma no sistema cadastrado. Associa-se aos requisitos R10 e R11.

3.2.2. Casos de Uso ó Nível de Sistema

Nesta seção apresentaremos os casos de uso em nível de sistema. Nestes são apresentados o objetivo de cada função, os usuários indicados, pré-condição e pós-

condição para realização do caso de uso. Além disso, os fluxos principais, alternativos e de exceção são detalhados para conhecimento das interações de cada funcionalidade do sistema.

3.2.2.1. Caso de Uso Autenticar Usuário

Objetivo	Autenticar no sistema para usar funcionalidades disponíveis no sistema.	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Usuário deve estar autorizado para acessar as funcionalidades do sistema.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator seleciona a opção Conectar do Menu Início.	2. Sistema exibe tela para preenchimento dos dados do usuário.	
3. Ator informa login e senha e clica em Conectar	4. Sistema verifica preenchimento e validade dos dados.	
	5. Sistema permite o acesso do usuário às funcionalidades.	
Fluxo Alternativo		
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.		
Fluxo Exceção		
Linha 4: Dados não válidos. Sistema exibe mensagem "Usuário/Senha incorretos" e exibe os campos que estão com problemas retornando ao passo 3.		
Linha 4: Campos vazios. Sistema exibe mensagem "Há campos vazios", retornando ao passo 3.		

Quadro 2: *Caso de Uso Autenticar Usuário*

3.2.2.2. Caso de Uso Manter Esquemas

Seção: Criar Esquema

Objetivo	Criar um novo esquema para importação/inclusão de tabelas, colunas, restrições e índices
Ator	DBA (Administrador de banco de dados) e desenvolvedores

Pré-Condição	O Ator precisa estar autenticado no sistema
Pós-Condição	Esquema deve ter sido criado e armazenado numa estrutura de dados específica do sistema.
Fluxo Principal	
Ação do Ator	Resposta do Sistema
1. Ator seleciona a opção Criar Esquema do Menu Esquemas.	2. Sistema exibe o formulário para preenchimento dos dados de Esquema.
3. Ator informa os dados e clica em Cadastrar.	4. Sistema verifica preenchimento e validade dos dados.
	5. Sistema grava dados informados em meio persistente.
Fluxo Alternativo	
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.	
Fluxo Exceção	
Linha 4: Dados não válidos. Sistema exibe mensagem <i>“Dados inválidos”</i> e exibe os campos que estão com problemas retornando ao passo 3.	
Linha 4: Campos vazios. Sistema exibe mensagem <i>“Há campos vazios”</i> , retornando ao passo 3.	

Quadro 3: *Caso de Uso Manter Esquemas ó Criar Esquema*

Seção: Editar Esquema

Objetivo	Altera os dados do esquema em meio persistente.
Ator	DBA (Administrador de banco de dados) e desenvolvedores
Pré-Condição	O Ator precisa estar autenticado no sistema
Pós-Condição	Esquema deve ter sido atualizado no sistema.
Fluxo Principal	
Ação do Ator	Resposta do Sistema
1. Ator seleciona a opção Editar Esquema do Menu Esquema.	2. Sistema exibe tela com opções para escolha do esquema, <i>baseline</i> e tipo de artefato que deseja editar.
3. Ator seleciona esquema e sua <i>baseline</i> da listagem, bem como o tipo de Artefato que deseja editar.	4. Sistema exibe tabela preenchida com os elementos do esquema correspondentes ao tipo de artefato e ao esquema selecionado.
5. Se o Ator deseja inserir um novo registro no esquema, ele preenche a linha da tabela com os dados e clica no botão Inserir. Se o Ator deseja editar algum dado, ele seleciona a linha da tabela desejada, altera os campos necessários e clica no botão Editar com a linha alterada selecionada. Se o Ator deseja remover algum elemento	6. Sistema verifica validade dos dados.

do esquema, seleciona a linha da tabela desejada e clica em Excluir.	
	7. Sistema exibe tabela atualizada com os elementos do esquema.
Fluxo Alternativo	
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.	
Linha 5: Ator clica em Fechar. Retorna ao passo 3.	
Fluxo Exceção	
Linha 4. Dados não válidos. Sistema exibe mensagem <i>“Dados inválidos”</i> e exibe os campos que estão com problemas retornando ao passo 3.	

Quadro 4: *Caso de Uso Manter Esquemas ó Editar Esquema*

Seção: Cadastrar Baseline

Objetivo	Criar um novo <i>baseline</i> de um esquema para controle de versão.	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	<i>Baseline</i> deve ter sido criado e armazenado numa estrutura de dados específica do sistema.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator seleciona a opção Cadastrar <i>Baseline</i> do Menu Esquema.	2. Sistema exibe o formulário para preenchimento dos dados de <i>Baseline</i> .	
3. Ator informa os dados e clica em Cadastrar.	4. Sistema verifica preenchimento e validade dos dados.	
	5. Sistema grava dados informados em meio persistente.	
Fluxo Alternativo		
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.		
Fluxo Exceção		
Linha 4. Dados não válidos. Sistema exibe mensagem <i>“Dados inválidos”</i> e exibe os campos que estão com problemas retornando ao passo 3.		
Linha 4: Campos vazios. Sistema exibe mensagem <i>“Há campos vazios”</i> , retornando ao passo 3.		

Quadro 5: *Caso de Uso Manter Esquemas ó Cadastrar Baseline*

Seção: Importar Esquema

Objetivo	Importar os dados do esquema em meio persistente.
Ator	DBA (Administrador de banco de dados) e desenvolvedores
Pré-Condição	O Ator precisa estar autenticado no sistema
Pós-Condição	Esquema do SGBD deve ter sido importado para o sistema.
Fluxo Principal	

Ação do Ator	Resposta do Sistema
1. Ator seleciona a opção Importar Esquema do Menu Esquema.	2. Sistema exibe tela para escolha da fonte de dados e do esquema e <i>baseline</i> para o qual será importado o esquema do SGBD, bem como o esquema a importar.
3. Ator seleciona os dados desejados.	4. Sistema mostra tabelas existentes no esquema a importar selecionado.
5. Ator clica em Importar.	6. Sistema importa tabelas, colunas, índices e restrições para o esquema selecionado.
Fluxo Alternativo	
Linha 5: Ator clica em Fechar. Retorna ao Menu Principal.	
Fluxo Exceção	
Linha 5. Esquema não selecionado. Sistema exibe mensagem <i>“É preciso selecionar um esquema, retornando ao passo 3.”</i>	

Quadro 6: *Caso de Uso Manter Esquemas ó Importar Esquema*

3.2.2.3. Caso de Uso Manter Artefatos

Seção: Cadastrar Sistema

Objetivo	Cadastrar sistema que será relacionado a um esquema
Ator	DBA (Administrador de banco de dados) e desenvolvedores
Pré-Condição	O Ator precisa estar autenticado no sistema
Pós-Condição	Sistema deve ter sido cadastrado no sistema.
Fluxo Principal	
Ação do Ator	Resposta do Sistema
1. Ator seleciona a opção Cadastrar Sistema do Menu Artefato e sub-menu Sistema.	2. Sistema exibe o formulário para preenchimento dos dados de Sistema.
3. Ator informa os dados e clica em Cadastrar.	4. Sistema verifica preenchimento e validade dos dados.
	5. Sistema grava dados informados em meio persistente.
Fluxo Alternativo	
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.	
Fluxo Exceção	
Linha 4. Dados não válidos. Sistema exibe mensagem <i>“Dados inválidos”</i> e exibe os campos que estão com problemas retornando ao passo 3.	
Linha 4: Campos vazios. Sistema exibe mensagem <i>“Há campos vazios”</i> , retornando ao passo 3.	

Quadro 7: *Caso de Uso Manter Artefato ó Cadastrar Sistema*

Seção: Editar Sistema

Objetivo	Altera os dados do Sistema em meio persistente.	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Sistema deve ter sido atualizado no sistema.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator seleciona a opção Editar Sistema do Menu Artefato e sub-menu Sistema.	2. Sistema exibe tela com lista de sistemas cadastrados.	
3. Ator seleciona o sistema que deseja editar e clica em Editar.	4. Sistema exibe tela com os dados do sistema selecionado.	
5. Ator informa os dados que deseja alterar e clica em Salvar.	6. Sistema verifica validade dos dados.	
	7. Sistema atualiza dados informados em meio persistente.	
Fluxo Alternativo		
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.		
Linha 5: Ator clica em Fechar. Retorna ao passo 3.		
Fluxo Exceção		
Linha 6. Dados não válidos. Sistema exibe mensagem "Dados inválidos" e exibe os campos que estão com problemas retornando ao passo 3.		

Quadro 8: *Caso de Uso Manter Artefato ó Editar Sistema*

Seção: Remover Sistema

Objetivo	Remover o sistema em meio persistente.	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Sistema deve ter sido removido do sistema.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator seleciona a opção Editar Sistema do Menu Sistema	2. Sistema exibe tela com lista de sistemas, módulos e artefatos cadastrados.	
3. Ator seleciona o sistema que deseja remover e clica em Excluir.	4. Sistema exibe mensagem para confirmação da remoção.	
5. Ator clica em Excluir.	6. Sistema remove o sistema do meio persistente.	
Fluxo Alternativo		
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.		
Linha 5: Ator clica em Cancelar. Retorna a linha 3.		

Quadro 9: *Caso de Uso Manter Artefato ó Remover Sistema*

Seção: Cadastrar Módulo

Objetivo	Cadastrar módulo de um sistema
-----------------	--------------------------------

Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Módulo deve ter sido cadastrado no sistema.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator seleciona a opção Cadastrar Módulo do Menu Artefato e sub-menu Módulo.	2. Sistema exibe o formulário para preenchimento dos dados de Módulo.	
3. Ator informa os dados e clica em Cadastrar.	4. Sistema verifica preenchimento e validade dos dados.	
	5. Sistema grava dados informados em meio persistente.	
Fluxo Alternativo		
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.		
Fluxo Exceção		
Linha 4: Dados não válidos. Sistema exibe mensagem "Dados inválidos" e exibe os campos que estão com problemas retornando ao passo 3.		
Linha 4: Campos vazios. Sistema exibe mensagem "Há campos vazios", retornando ao passo 3.		

Quadro 10: *Caso de Uso Manter Artefato ó Cadastrar Módulo*

Seção: Editar Módulo

Objetivo	Altera os dados do módulo em meio persistente.	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Módulo deve ter sido atualizado no sistema.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator seleciona a opção Editar Módulo do Menu Artefato e sub-menu Módulo.	2. Sistema exibe tela com lista de sistemas e módulos cadastrados	
3. Ator seleciona o módulo do sistema que deseja editar e clica em Editar.	4. Sistema exibe tela com os dados do módulo selecionado.	
5. Ator informa os dados que deseja alterar e clica em Salvar.	6. Sistema verifica validade dos dados.	
	7. Sistema atualiza dados informados em meio persistente.	
Fluxo Alternativo		
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.		
Linha 5: Ator clica em Fechar. Retorna ao passo 3.		
Fluxo Exceção		
Linha 6: Dados não válidos. Sistema exibe mensagem "Dados inválidos" e exibe os campos que estão com problemas retornando ao passo 3.		

Quadro 11: *Caso de Uso Manter Artefato ó Editar Módulo*

Seção: Remover Módulo

Objetivo	Remover o módulo em meio persistente.	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Módulo deve ter sido removido do sistema.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator seleciona a opção Editar Módulo do Menu Artefato e sub-menu Módulo.	2. Sistema exibe tela com lista de sistemas e módulos cadastrados.	
3. Ator seleciona o módulo que deseja remover e clica em Excluir.	4. Sistema exibe mensagem para confirmação da remoção.	
5. Ator clica em Excluir.	6. Sistema remove o módulo do meio persistente.	
Fluxo Alternativo		
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.		
Linha 5: Ator clica em Cancelar. Retorna a linha 3.		

Quadro 12: *Caso de Uso Manter Artefato ó Remover Módulo***Seção: Cadastrar Artefato**

Objetivo	Cadastrar artefato correspondente a um módulo	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Artefato deve ter sido cadastrado no sistema.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator seleciona a opção Cadastrar Artefato do Menu Artefato e sub-menu Artefato.	2. Sistema exibe o formulário para preenchimento dos dados do Artefato.	
3. Ator informa os dados e clica em Cadastrar.	4. Sistema verifica preenchimento e validade dos dados.	
	5. Sistema grava dados informados em meio persistente.	
Fluxo Alternativo		
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.		
Fluxo Exceção		
Linha 4: Dados não válidos. Sistema exibe mensagem <i>“Dados inválidos”</i> e exibe os campos que estão com problemas retornando ao passo 3.		
Linha 4: Campos vazios. Sistema exibe mensagem <i>“Há campos vazios”</i> , retornando ao passo 3.		

Quadro 13: *Caso de Uso Manter Artefato ó Cadastrar Módulo*

Seção: Editar Artefato

Objetivo	Altera os dados do artefato em meio persistente.	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Artefato deve ter sido atualizado no sistema.	
Fluxo Principal		
Ação do Ator		Resposta do Sistema
1. Ator seleciona a opção Editar Artefato do Menu Artefato e sub-menu Artefato.	2. Sistema exibe tela com lista de sistemas, módulos e artefatos cadastrados.	
3. Ator seleciona o artefato correspondente ao módulo e sistema que deseja editar e clica em Editar.	4. Sistema exibe tela com os dados do artefato selecionado.	
5. Ator informa os dados que deseja alterar e clica em Salvar.	6. Sistema verifica validade dos dados.	
	7. Sistema atualiza dados informados em meio persistente.	
Fluxo Alternativo		
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.		
Linha 5: Ator clica em Fechar. Retorna ao passo 3.		
Fluxo Exceção		
Linha 6. Dados não válidos. Sistema exibe mensagem <i>“Dados inválidos”</i> e exibe os campos que estão com problemas retornando ao passo 3.		

Quadro 14: *Caso de Uso Manter Artefato ó Editar Artefato***Seção: Remover Artefato**

Objetivo	Remover o artefato em meio persistente.	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Artefato deve ter sido removido do sistema.	
Fluxo Principal		
Ação do Ator		Resposta do Sistema
1. Ator seleciona a opção Editar Artefato do Menu Artefato e sub-menu Artefato.	2. Sistema exibe tela com lista de sistemas, módulos e artefatos cadastrados.	
3. Ator seleciona o artefato correspondente ao módulo e sistema que deseja remover e clica em Excluir.	4. Sistema exibe mensagem para confirmação da remoção.	
5. Ator clica em Excluir.	6. Sistema remove o artefato do meio persistente.	
Fluxo Alternativo		
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.		
Linha 5: Ator clica em Cancelar. Retorna a linha 3.		

Quadro 15: *Caso de Uso Manter Artefato ó Remover Artefato*

3.2.2.4. Caso de Uso Manter Fontes de Dados

Seção: Cadastrar Fonte de Dados

Objetivo	Cadastrar uma nova fonte de dados para ser usada na importação de esquema e monitoração de mudanças.	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Fonte de dados deve ter sido criada e armazenada numa estrutura de dados específica do sistema.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator seleciona a opção Cadastrar Fonte de Dados do Menu Início.	2. Sistema exibe o formulário para preenchimento dos dados da Fonte de Dados.	
3. Ator informa os dados e clica em Cadastrar.	4. Sistema verifica preenchimento e validade dos dados.	
	5. Sistema grava dados informados em meio persistente.	
Fluxo Alternativo		
Linha 3: Ator clica em Fechar. Retorna ao Menu Principal.		
Fluxo Exceção		
Linha 4: Dados não válidos. Sistema exibe mensagem "Dados inválidos" e exibe os campos que estão com problemas retornando ao passo 3.		
Linha 4: Campos vazios. Sistema exibe mensagem "Há campos vazios", caso exista algum campo sem preenchimento, retornando ao passo 3.		

Quadro 16: Caso de Uso Manter Fonte de Dados ó Cadastrar Fonte de Dados

Seção: Editar Fonte de Dados

Objetivo	Altera os dados da fonte de dados em meio persistente.	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Fonte de dados deve ter sido atualizada no sistema.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator seleciona a opção Editar Fonte de Dados do Menu Início.	2. Sistema exibe opção de selecionar dentre as fontes de dados cadastradas.	
3. Ator seleciona a fonte de dados que deseja alterar e clica em Editar	4. Sistema exibe tela com os dados da fonte selecionada.	
5. Ator informa os dados que deseja alterar e clica em Salvar.	6. Sistema verifica validade dos dados.	
	7. Sistema atualiza dados informados em meio persistente.	

Fluxo Alternativo	
Linha 3:	Ator clica em Fechar. Retorna ao Menu Principal.
Linha 5:	Ator clica em Fechar. Retorna ao passo 3.
Fluxo Exceção	
Linha 6.	Dados não válidos. Sistema exibe mensagem “Dados inválidos” e exibe os campos que estão com problemas retornando ao passo 3.

Quadro 17: *Caso de Uso Manter Fonte de Dados ó Editar Fonte de Dados*

Seção: Remover Fonte de Dados

Objetivo	Remover uma fonte de dados do meio persistente.	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Fonte de dados deve ter sido removida do sistema.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator seleciona a opção Remover Fonte de Dados do Menu Início.	2. Sistema exibe opção de selecionar dentre as fontes de dados cadastradas.	
3. Ator seleciona a fonte de dados que deseja remover e clica em Excluir	4. Sistema exclui os dados do meio persistente.	
Fluxo Alternativo		
Linha 3:	Ator clica em Fechar. Retorna ao Menu Principal.	

Quadro 18: *Caso de Uso Manter Fonte de Dados ó Remover Fonte de Dados*

3.2.2.5. Caso de Uso Manter Relação de Dependência entre Artefatos e Esquemas

Seção: Criar Vínculo

Objetivo	Relacionar artefato a elementos de esquema	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Vínculo deve ter sido armazenado no sistema.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator seleciona a opção Criar Vínculo do Menu Vincular.	2. Sistema exibe tela com opções para escolher artefato correspondente a determinado módulo e sistema, o tipo de elemento e o esquema, juntamente com sua versão, que se quer vincular.	
3. Ator seleciona o artefato, a versão do esquema desejada e o tipo de elemento, que pode ser esquema, tabela, coluna,	4. Sistema exibe lista com os elementos do esquema correspondentes ao tipo e versão do esquema selecionada.	

restrição ou índice.	
5. Ator seleciona os elementos listados que se deseja relacionar ao artefato e clica em Salvar.	6. Sistema verifica se os elementos já estão vinculados ao artefato selecionado
	7. Sistema grava dados informados em meio persistente.
Fluxo Alternativo	
Linha 5: Ator clica em Fechar. Retorna ao Menu Principal.	
Fluxo Exceção	
Linha 6. Elementos já vinculados. Sistema exibe mensagem de erro: Artefato já está vinculado a esse elemento, retornando ao passo 5.	

Quadro 19: *Caso de Uso Manter Artefato ó Criar Vinculo*

Seção: Remover Vínculo

Objetivo	Remover relação entre artefatos e elementos de esquema do meio persistente.	
Ator	DBA (Administrador de banco de dados) e desenvolvedores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Artefato deve ter sido atualizado no sistema.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator seleciona a opção Remover Vínculo do Menu Vincular.	2. Sistema exibe tela com opções para escolher artefato correspondente a determinado módulo e sistema, o tipo de elemento e o esquema, juntamente com sua versão, que se quer vincular.	
3. Ator seleciona o artefato, a versão do esquema desejada e o tipo de elemento, que deseja desvincular.	4. Sistema exibe lista com os elementos vinculados ao artefato selecionado.	
5. Ator seleciona os elementos listados que se deseja desvincular do artefato e clica em Remover.	6. Sistema remove os vínculos do meio persistente.	
Fluxo Alternativo		
Linha 5: Ator clica em Fechar. Retorna ao Menu Principal.		

Quadro 20: *Caso de Uso Manter Artefato ó Remover Vínculo*

3.2.2.6. Caso de Uso Manter Mudanças de Esquemas

Seção: Comparar *Baselines*

Objetivo	Comparar baselines para identificar diferenças
Ator	DBA (Administrador de banco de dados) e desenvolvedores
Pré-Condição	O Ator precisa estar autenticado no sistema
Pós-Condição	Diferenças devem ser armazenadas no sistema.
Fluxo Principal	
Ação do Ator	Resposta do Sistema
1. Ator seleciona a opção Comparar <i>baselines</i> do Menu Impacto.	2. Sistema exibe tela com opções para escolher esquema e os dois <i>baselines</i> desejados para comparação.
3. Ator seleciona o esquema e os <i>baselines</i> e clica em Comparar.	4. Sistema exibe tabela com as diferenças encontradas entre os dois <i>baselines</i> .
5. Se houver diferenças entre os dois <i>baselines</i> , o Ator clica em Salvar.	6. Sistema exibe tela para preenchimento do nome e data da comparação.
7. Ator informa os dados e clica em Salvar.	8. Sistema verifica preenchimento e validade dos dados
	9. Sistema grava dados informados em meio persistente.
Fluxo Alternativo	
Linha 5: Ator clica em Fechar. Retorna ao Menu Principal.	
Linha 7: Ator clica em Cancelar. Retorna ao passo 5.	
Fluxo Exceção	
Linha 8. Dados não válidos. Sistema exibe mensagem <i>“Dados inválidos”</i> e exibe os campos que estão com problemas, retornando ao passo 3.	
Linha 8: Campos vazios. Sistema exibe mensagem <i>“Há campos vazios”</i> , retornando ao passo 7.	

Quadro 21: Caso de Uso Manter Mudanças ó Comparar Baselines

3.2.2.7. Caso de Uso Avaliar Impacto de Mudanças em Esquema

Objetivo	Identificar os artefatos que podem ser afetados por alterações em esquema.
Ator	DBA (Administrador de banco de dados) e desenvolvedores
Pré-Condição	O Ator precisa estar autenticado no sistema
Pós-Condição	Avaliação deve ser apresentada e pode ser salva em documento.
Fluxo Principal	
Ação do Ator	Resposta do Sistema
1. Ator seleciona a opção Avaliar Impacto <i>baselines</i> do Menu Impacto.	2. Sistema exibe tela com opção para escolher as comparações de <i>baselines</i> cadastradas.
3. Ator seleciona a comparação.	4. Sistema exibe na tela esquema, <i>baselines</i> e data da comparação selecionada.
5. Ator clica em Avaliar.	6. Sistema verifica preenchimento dos dados.

	7. Sistema exibe tabela com dados do elemento alterado e artefatos afetados pelo mesmo.
8. Se Ator desejar salvar relatório ele clica em Gerar Relatório.	9. Sistema exibe um seletor de arquivos para seleção do diretório.
10. Ator escolhe o diretório, informa o nome do arquivo e clica em Salvar.	11. Sistema verifica validade dos dados.
	12. Sistema grava dados informados em meio persistente.
	13. Sistema volta à tela de Impacto.
Fluxo Alternativo	
Linha 5: Ator clica em Fechar. Retorna ao Menu Principal.	
Linha 10: Ator clica em Cancelar. Retorna ao passo 8.	
Fluxo Exceção	
Linha 6: Campos vazios. Sistema exibe mensagem <i>“Há campos vazios”</i> , retornando ao passo 5.	
Linha 11. Dados não válidos. Sistema exibe mensagem <i>“Dados inválidos”</i> e exibe os campos que estão com problemas, retornando ao passo 8.	

Quadro 22: *Caso de Uso Avaliar Impacto de Mudanças em Esquema*

3.2.2.8. Caso de Uso Gerar Relatórios

Seção: Gerar Relatórios de Diferenças entre Baselines

Objetivo	Gerar arquivo pdf com dados da comparação entre baselines.
Ator	DBA (Administrador de banco de dados) e desenvolvedores
Pré-Condição	O Ator precisa estar autenticado no sistema
Pós-Condição	Arquivo deve ser armazenado no sistema.
Fluxo Principal	
Ação do Ator	Resposta do Sistema
1. Ator seleciona a opção Relatório de Diferenças entre <i>Baselines</i> do Menu Relatórios.	2. Sistema exibe tela com opção para escolher as comparações de baselines cadastradas.
3. Ator seleciona a comparação.	4. Sistema exibe na tela esquema, <i>baselines</i> e data da comparação selecionada.
5. Se o Ator deseja mudar o caminho de destino, onde o arquivo será salvo, ele clica em Procurar	6. Sistema exibe um seletor de arquivos para seleção do diretório.
7. Ator escolhe o diretório, informa o nome do arquivo e clica em Salvar.	8. Sistema volta à tela de relatórios.
9. Ator clica em Gerar Relatório.	10. Sistema verifica preenchimento e validade dos dados.

	11. Sistema grava arquivo em meio persistente.
Fluxo Alternativo	
Linha 5: Ator clica em Fechar. Retorna ao Menu Principal.	
Linha 9: Ator clica em Fechar. Retorna ao Menu Principal.	
Linha 7: Ator clica em Cancelar. Retorna ao passo 5.	
Fluxo Exceção	
Linha 10. Dados não válidos. Sistema exibe mensagem <i>“Dados inválidos”</i> e exibe os campos que estão com problemas, retornando ao passo 9.	
Linha 10: Campos vazios. Sistema exibe mensagem <i>“Há campos vazios”</i> , retornando ao passo 9.	

Quadro 23: *Caso de Uso Gerar Relatórios ó Gerar Relatórios de Diferenças entre Baselines*

Seção: Gerar Relatórios de Impacto de Mudanças em Esquema

Objetivo	Gerar arquivo pdf com dados do impacto de mudanças em esquema.
Ator	DBA (Administrador de banco de dados) e desenvolvedores
Pré-Condição	O Ator precisa estar autenticado no sistema
Pós-Condição	Arquivo deve ser armazenado no sistema.
Fluxo Principal	
Ação do Ator	Resposta do Sistema
1. Ator seleciona a opção Relatório de Impacto de Mudanças do Menu Relatórios.	2. Sistema exibe tela com opção para escolher as comparações de <i>baselines</i> cadastradas.
3. Ator seleciona a comparação.	4. Sistema exibe na tela esquema, baselines e data da comparação selecionada.
5. Ator informa nome do arquivo.	
6. Se o Ator deseja mudar o caminho de destino, onde o arquivo será salvo, ele clica em Escolher.	7. Sistema exibe um seletor de arquivos para seleção do diretório.
8. Ator escolhe o diretório e clica em Salvar.	9. Sistema volta à tela de relatórios.
10. Ator clica em Gerar Relatório.	11. Sistema verifica preenchimento e validade dos dados.
	12. Sistema grava arquivo em meio persistente.
Fluxo Alternativo	
Linha 6: Ator clica em Fechar. Retorna ao Menu Principal.	
Linha 10: Ator clica em Fechar. Retorna ao Menu Principal.	
Linha 8: Ator clica em Cancelar. Retorna ao passo 6.	

Fluxo Exceção
Linha 11. Dados não válidos. Sistema exibe mensagem "Dados inválidos" e exibe os campos que estão com problemas, retornando ao passo 9.
Linha 11: Campos vazios. Sistema exibe mensagem "Há campos vazios", retornando ao passo 9.

Quadro 24: *Caso de Uso Gerar Relatórios ó Gerar Relatórios de Impacto de Mudanças em Esquema*

3.3. Diagrama de Classes

É o diagrama central da modelagem orientada a objeto. Exibe as classes que irão compor o sistema com seus respectivos métodos, atributos e relacionamentos.

Após a criação dos casos de uso da ferramenta, foram modelados os diagramas de classes do Sistema. Nesta seção apresentaremos os diagramas separados por casos de uso, detalhando as classes que o sistema necessita possuir e os relacionamentos entre si.

3.3.1. Caso de uso Manter Esquemas

GUICadastrarEsquema ó Representa a interface (fronteira) por onde o usuário fornece os dados necessários para a criação de um novo esquema, juntamente com seu *baseline*. Os métodos *cadastrarEsquema()* e *cadastrarBaseline()* permitem ao usuário efetivar a criação de um esquema, com seu primeiro *baseline* (versão), para o sistema. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela. Relaciona-se com as classes *Esquema*, para criação de um objeto esquema com os dados a serem cadastrados, e *DAOEsquema* e *DAOBaseline*, para efetivar a inserção dos novos esquema e *baseline* no banco de dados. A Figura 2 apresenta o diagrama de classes da parte de criação de esquemas do caso de uso Manter Esquemas.

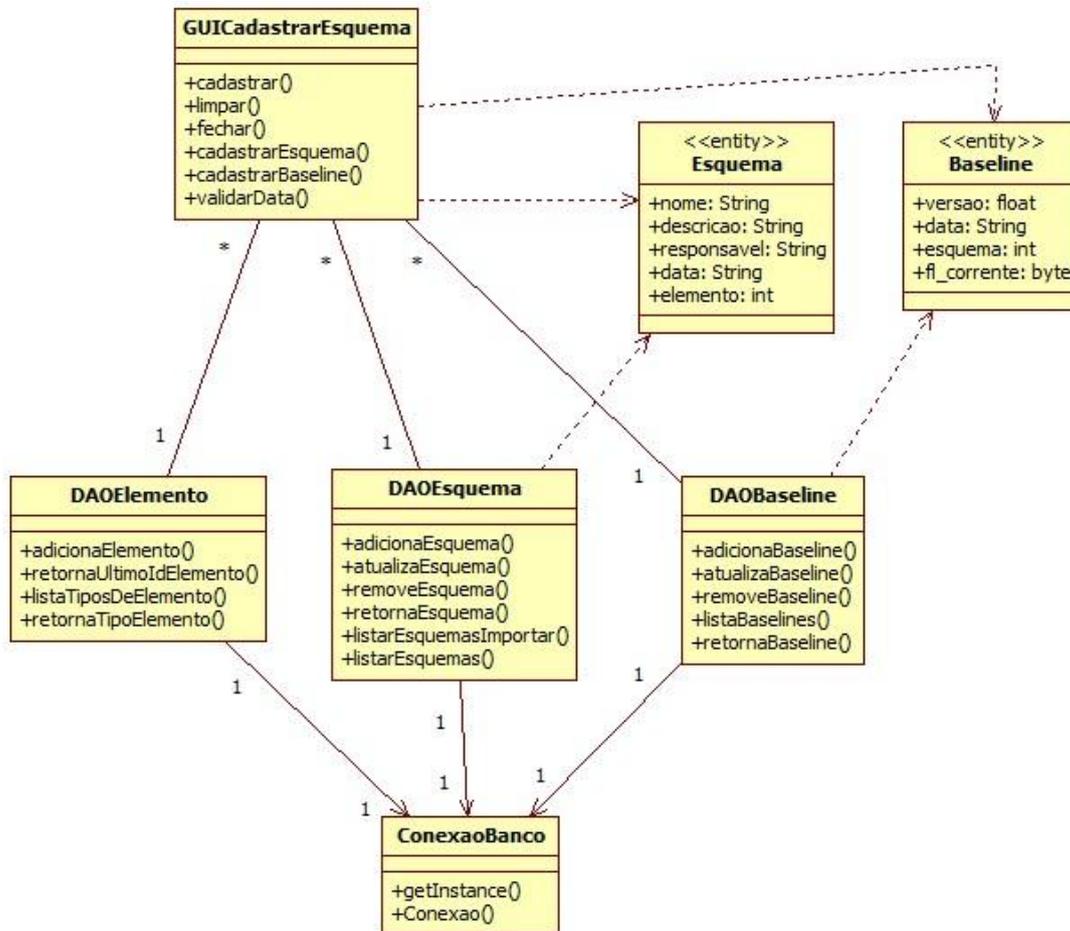


Figura 2: *Diagrama de classes do caso de uso Manter Esquemas ó Cadastrar Esquema*

DAOEsquema ó Representa a classe de controle que, conectado ao SGBD alvo, gerencia as operações de inserção de novos esquemas, bem como deleção, edição e recuperação de um esquema no banco de dados. Isso é feito através dos métodos *adicionaEsquema()*, *atualizaEsquema()*, *removeEsquema()*, *listaEsquemas()*, *retornaEsquema()*. O método *listaEsquemasImportar()* retorna os esquemas registrados no banco de dados que o usuário deseja importar. Relaciona-se com a classe *Esquema* para obter os dados do objeto, informados na interface, e, então, gravá-los, por meio de seus métodos, na tabela determinada para esquema na base de dados.

DAOBaseline ó Representa a classe de controle para inserção, deleção, edição e recuperação de *baselines* de um esquema. Essa classe se conecta ao SGBD alvo por

meio da classe de conexão, *ConexaoBanco*, para persistir os dados do *baseline* na base de dados. Isso é feito através dos métodos *adicionaBaseline()*, *atualizaBaseline()*, *removeBaseline()*, *listaBaselines()*, *retornaBaseline()*. Relaciona-se com a classe *Baseline* para obter os dados, informados pelo usuário na tela, e, então, gravá-los, por meio de seus métodos, na tabela determinada para *baseline*.

DAOElemento ó Representa a classe de controle de elementos. Associa-se a esquemas, tabelas, colunas, restrições e índices, e, portanto, relaciona-se com as classes de interface desses elementos. Gerencia as operações de inserção e recuperação de elementos, juntamente com seu tipo, por meio dos métodos *adicionaElemento()*, e *retornaElemento()*. À medida que novos esquemas, tabelas, colunas, restrições ou índices são criados, novos elementos são persistidos no banco de dados. Adicionalmente, quando elementos de um *baseline* são importados para uma nova versão, os elementos já cadastrados são retornados para associação com a nova versão.

Esquema ó Classe que corresponde aos dados dos esquemas (nome, descrição, responsável, data e elemento) para persistir na Base de Dados.

Baseline ó Classe que corresponde aos dados de *baseline* (versão, data, fl_corrente e esquema) a serem persistidos na Base de Dados.

ConexaoBanco - Classe que representa uma conexão com determinado SGBD. A conexão é feita através dos métodos *getInstance()*, que instancia um novo objeto da classe, caso não o mesmo exista, e *Conexao()*, que efetiva a conexão. Relaciona-se com todas as classes DAO, possibilitando às mesmas o acesso ao banco de dados.

GUICadastrarBaseline ó Representa a interface por onde o usuário fornece os dados necessários para a criação de um novo *baseline* para um determinado esquema. O método *cadastrarBaseline()* permite ao usuário efetivar a criação de uma nova versão do esquema para o sistema. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela. Relaciona-se com as classes *Baseline*, para criação de um objeto

baseline com os dados a serem cadastrados, e *DAOBaseline*, para efetivar a inserção do novo *baseline* no banco de dados. A Figura 3 apresenta o diagrama de classes da parte de criação de *baselines* do caso de uso Manter Esquemas.

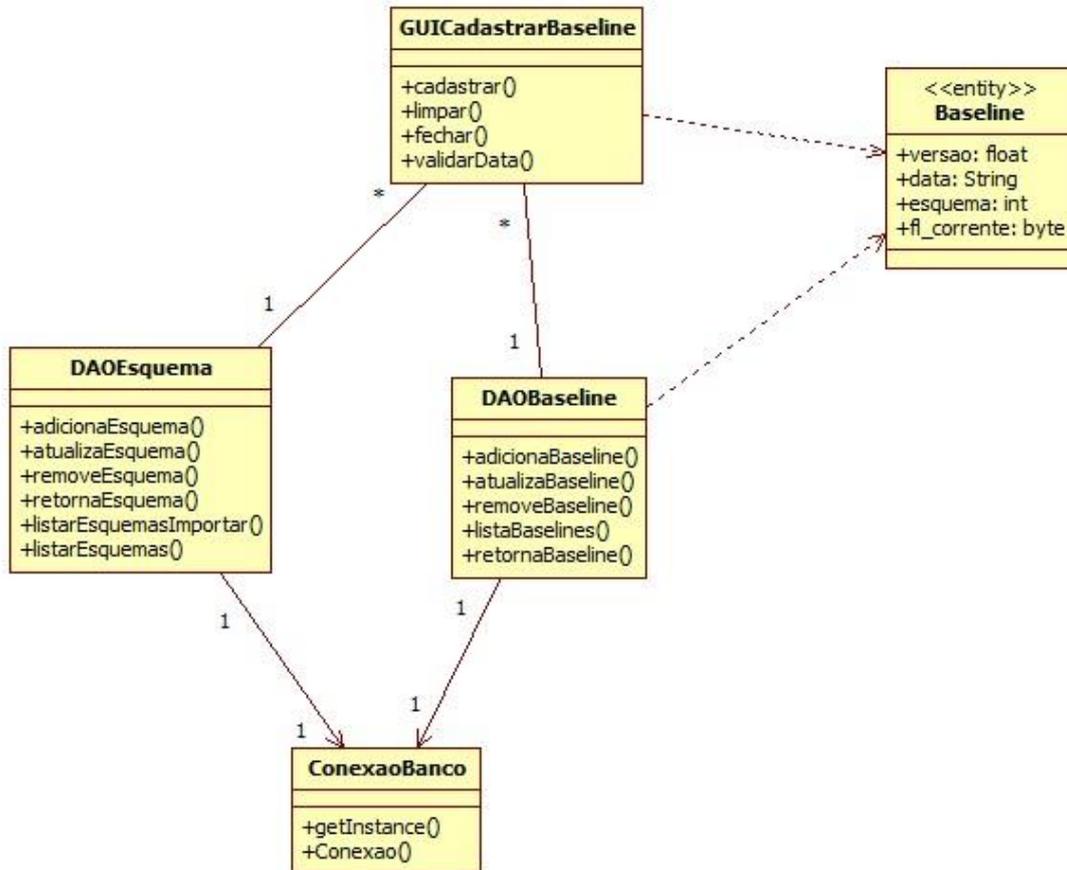


Figura 3: Diagrama de classes do caso de uso Manter Esquemas ó Cadastrar Baseline

GUIEditarEsquema ó É a classe representante da tela que possibilita ao usuário alterar dados nos elementos de determinada versão do esquema. Registros de esquema, tabela, coluna, restrições e índices podem ser adicionados, alterados ou excluídos por meio dos métodos *inserir()*, *editar()* e *excluir()*, respectivamente. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela. A Figura 4 apresenta o diagrama de classes da parte de editar esquemas do caso de uso Manter Esquemas.

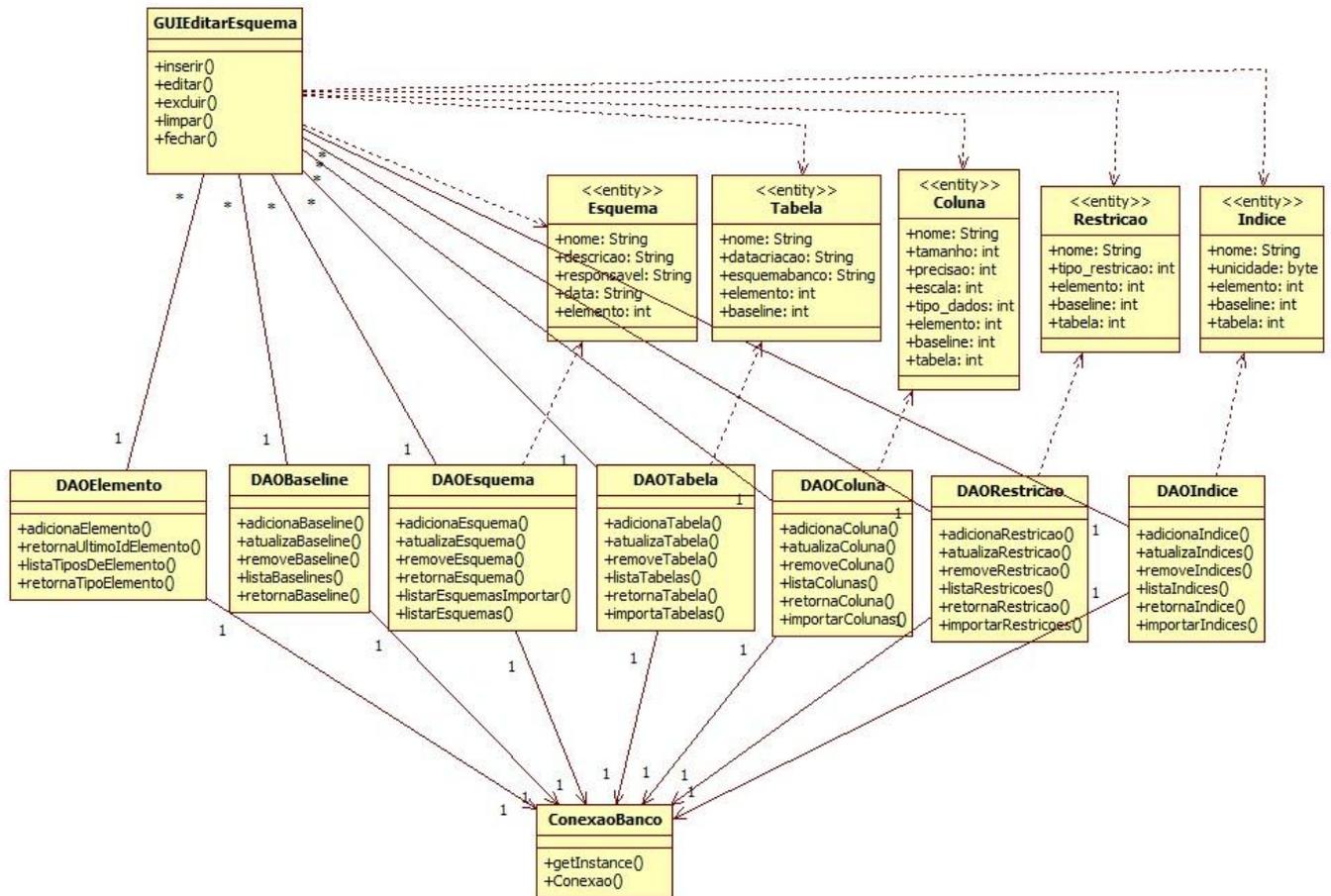


Figura 4: *Diagrama de classes do caso de uso Manter Esquemas ó Editar Esquema*

DAOTabela ó Representa a classe de controle para inserção, deleção, edição e recuperação de tabelas de um esquema. Essa classe se conecta ao SGBD alvo para persistir os dados de tabelas na base de dados. Isso é feito através dos métodos *adicionaTabela()*, *atualizaTabela()*, *removeTabela()*, *listaTabelas()*, *retornaTabela()* e *importaTabelas()*, entre outros. Relaciona-se com a classe Modelo *Tabela* para obter os dados necessários, e, então, os grava, por meio de seus métodos, na tabela determinada para tabela.

Tabela ó Classe que corresponde aos dados das tabelas (nome, data de criação, esquema do banco, *baseline* e elemento) para persistir na Base de Dados.

DAOColuna ó Classe que controla as operações de inserção, atualização, remoção e consulta de colunas que fazem parte de um esquema. *DAOColuna* efetiva essas

operações junto ao SGBD alvo, com o qual se conecta por meio da classe *ConexaoBanco*. Os principais métodos dessa classe são: *adicionaColuna()*, *atualizaColuna()*, *removeColuna()*, *listaColuna()*, *retornaColuna()* e *importaColunas()*, os quais, por meio de instruções SQL, permitem persistir os dados de colunas na tabela correspondente do banco. A classe *Coluna* é a responsável pelo acesso e modificação dos dados das colunas.

Coluna *ó* Apresenta os dados da coluna (nome, tamanho, precisão, escala, tabela, tipo de dados, *baseline* e elemento) para serem recuperados ou modificados.

DAORestricao *ó* Apresenta os métodos *adicionaRestricao()*, *adicionaRestricaoColuna()*, *atualizaRestricao()*, *removeRestricao()*, *listaRestricao()*, *retornaRestricao()* e *importaRestricoes()*, com os quais insere, remove, atualiza e recupera dados de restrições do banco de dados. A partir de conexão feita com o SGBD, essa classe efetiva a persistência de restrições relacionadas a tabelas e a colunas determinadas. Relaciona-se com a classe *Restricao* para obter os dados informados pelo usuário na tela, e, então, gravá-los na base.

Restricao *ó* Classe de dados correspondente a uma restrição (nome, tipo de restrição, tabela, *baseline* e elemento).

DAOIndice *ó* Representa a classe de controle de índices, os quais são associados a tabelas e a colunas. Essa relação é gerenciada através da conexão com o banco de dados e das operações de inserção de novos índices, bem como deleção, edição e recuperação de um índice no banco de dados. Essas operações são realizadas por meio dos métodos *adicionaIndice()*, *adicionaIndiceColuna()*, *atualizaIndice()*, *removeIndice()*, *listaIndices()*, *retornaIndice()*, *retornaIndiceColuna()*, *importaIndices()*. Essa classe possui relação com a classe *Indice*, classe do pacote Modelo, responsável por recuperar e modificar os dados de um objeto índice. Os métodos da classe *DAOIndice* obtém os dados informados na interface (guardados em um objeto índice), e, então, persiste-os nas tabelas determinadas na base de dados.

Indice é Classe que corresponde aos dados dos índices (nome, unicidades, tabela, baseline e elemento) para persistir na Base de Dados. Possibilita às classes GUI e DAO guardar e recuperar, respectivamente, os dados necessários do índice.

GUIImportarEsquema é Classe de interface que apresenta ao usuário as opções de importar um esquema completo de um SGBD determinado, bem como alterar a fonte de dados para essa importação. O método *alterarFonte()* permite modificar o local (servidor, instância e banco de dados) onde se localiza o esquema de dados a ser importado. O método *importar()* efetiva a recuperação de tabelas, colunas, restrições e índices do esquema escolhido e grava no esquema do sistema. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela. Possui associação com as classes de controle *DAOElemento*, *DAOBaseline*, *DAOEsquema*, *DAOTabela*, *DAOColuma*, *DAORestricao*, *DAOColuna*, através das quais acessa os métodos de importação de tabelas, colunas, restrição e índices. Adicionalmente, relaciona-se com as classes de dados *Tabela*, *Coluna*, *Restricao* e *Indice*. A Figura 5 apresenta o diagrama de classes da parte de importação de esquemas do caso de uso Manter Esquemas.

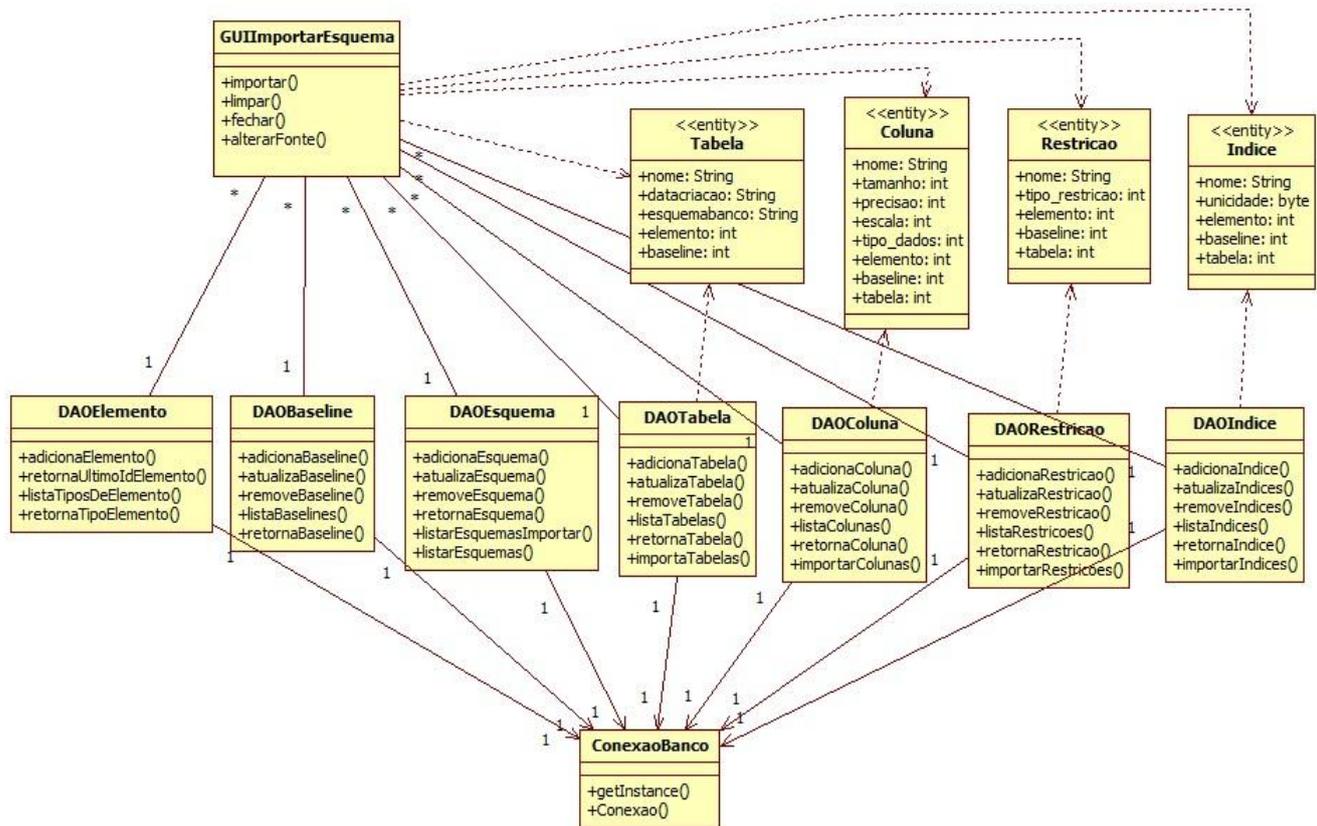


Figura 5: *Diagrama de classes do caso de uso Manter Esquemas ó Importar Esquema*

3.3.2. Caso de uso Manter Artefatos

GUICadastrarSistema ó Representa a interface (fronteira) por onde o usuário fornece os dados necessários para a criação de um novo sistema. Na ferramenta proposta, os artefatos possuem módulos associados, que por sua vez, possuem sistemas relacionados. Essa hierarquia oferece detalhamento do artefato que será avaliado no impacto de evolução de esquema, permitindo uma facilidade na sua localização. O método *cadastrarSistema()* permite ao usuário efetivar a criação de um sistema. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela. Relaciona-se com as classes *Sistema*, para criação de um objeto com os dados de um sistema a serem cadastrados, e *DAOSistema*, para efetivar a inserção dos novos sistemas banco de dados. A Figura 6 apresenta o diagrama de classes que envolve Sistema do caso de uso Manter Artefatos.

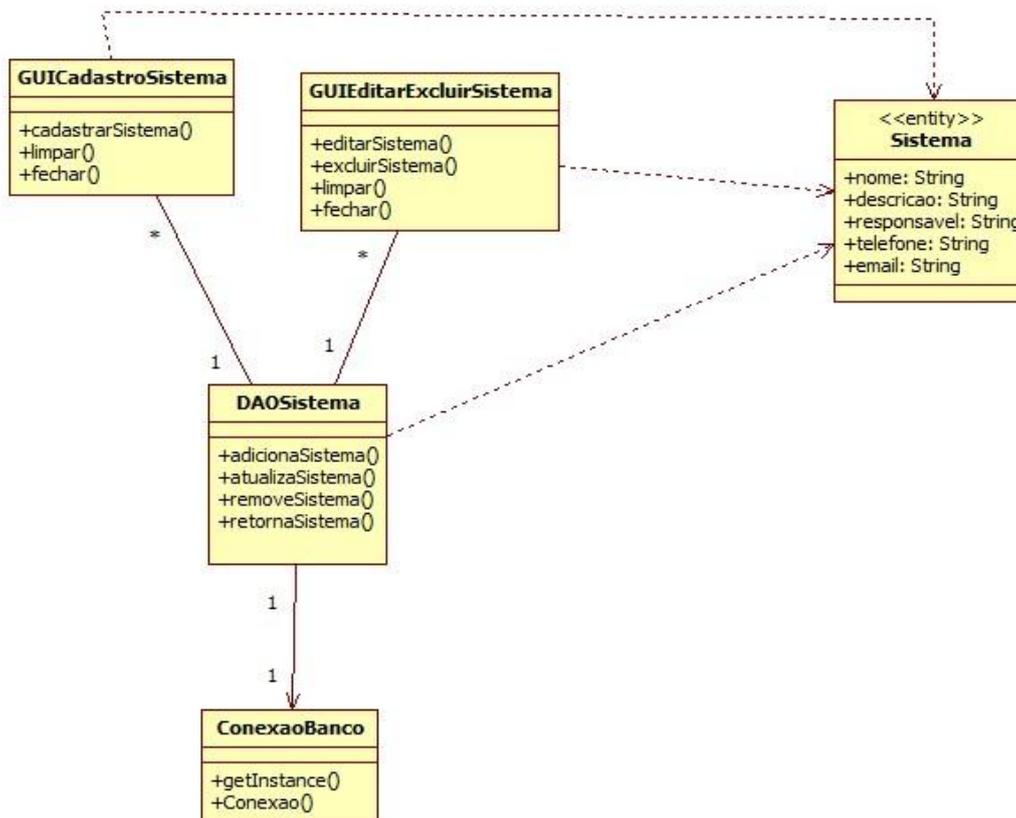


Figura 6: *Diagrama de classes do caso de uso Manter Artefatos - Sistema*

GUIEditarExcluirSistema é a classe representante da tela que possibilita ao usuário alterar dados nos sistemas cadastrados. Além de atualizar dados do sistema, a interface oferece a opção de excluir determinado sistema por meio dos métodos *editarSistema()* e *excluirSistema()*, respectivamente. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela. As classes *Sistema* e *DAOSistema*, relacionadas a essa interface, acessam o objeto do sistema editado ou excluído para persistir na base.

DAOSistema representa a classe de controle que, conectado ao SGBD alvo, gerencia as operações de inserção de novos sistemas, bem como deleção, edição e recuperação de um sistema no banco de dados. Isso é feito através dos métodos *adicionaSistema()*, *atualizaSistema()*, *removeSistema()*, *listaSistemas()* e *retornaEsquema()*. Relaciona-se com a classe *Sistema* para obter os dados do objeto, informados na interface, e, então, gravá-los, por meio de seus métodos, na tabela determinada para sistema na base de dados.

Sistema é Classe que corresponde aos dados de sistema (nome, descrição, responsável, telefone e e-mail) para persistir na base através da classe *DAOSistema*.

GUICadastrarModulo é Classe que apresenta tela de criação de um novo módulo ao usuário. Apresenta os métodos *cadastrarModulo()*, que permite ao usuário efetivar a criação de um módulo correspondente a um sistema, o método *limpar()*, que esvazia os campos da tela, e o *fechar()*, que fecha a interface. Relaciona-se com as classes *Modulo*, para guardar os dados de um módulo a serem cadastrados no banco, *DAOSistema*, para recuperar os dados necessários do sistema do qual faz parte o módulo, e *DAOModulo*, para efetivar a inserção dos novos módulos. A Figura 7 apresenta o diagrama de classes que envolve Módulo do caso de uso Manter Artefatos.

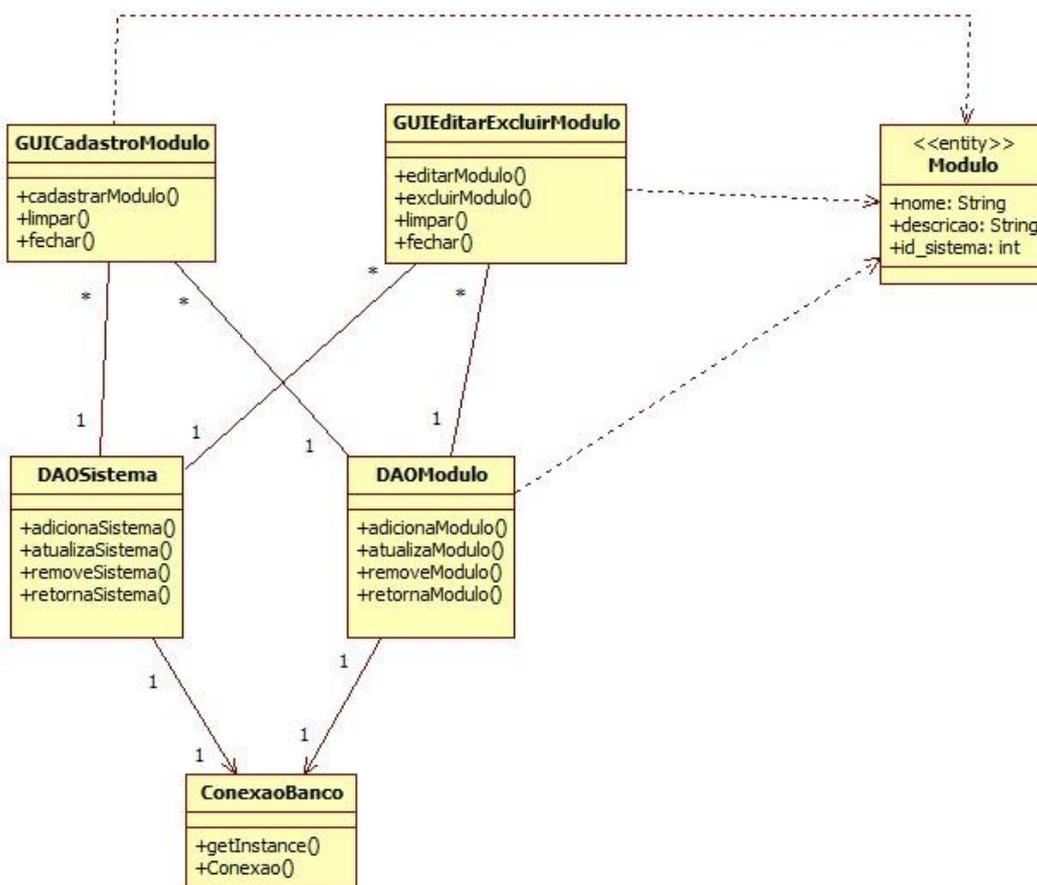


Figura 7: Diagrama de classes do caso de uso Manter Artefatos é Módulo

GUIEditarExcluirModulo ó É a classe de interface que apresenta aos usuários as opções de editar e excluir módulo correspondente a determinado sistema. Isso é feito por meio dos métodos *editarModulo()* e *excluirModulo()*, que acessa determinados métodos da classe *DAOModulo* para efetivar essas operações no banco de dados. Bem como a interface de cadastro de módulo, essa classe também relaciona-se a classe de dados *Modulo* para acessar seus campos, necessários a qualquer registro de módulo. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela.

DAOModulo ó Classe que controla as operações de inserção, atualização, remoção e consulta de módulos de um sistema. *DAOModulo* efetiva essas operações junto ao SGBD alvo, com o qual se conecta por meio da classe *ConexaoBanco*. Os principais métodos dessa classe são: *adicionaModulo()*, *atualizaModulo()*, *removeModulo()*, *listaModulos()* e *retornaModulo()*, os quais permitem persistir os dados dos módulos na tabela correspondente do banco. A classe *Modulo* é a responsável pelo acesso e modificação dos dados das colunas.

Modulo ó Classe de dados que permite o acesso e modificação dos dados do módulo (nome, descrição e sistema). Relaciona-se com a classe *GUICadastroModulo*, permitindo-o guardar os dados do módulo, e com a classe *DAOModulo*, a qual acessa os dados do módulo.

GUICadastrarArtefato ó Classe de interface que apresenta aos usuários a tela de criação de um novo artefato. Registros de novos artefatos de um módulo são inseridos por meio do método *cadastrarArtefato()*. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela. Relaciona-se com as classes *Artefato*, para guardar os dados do artefato a serem inseridos no banco, *DAOModulo*, para recuperar os dados necessários do módulo, do qual faz parte o artefato, e *DAOArtefato*, para efetivar o registro do novo artefato. A Figura 8 apresenta o diagrama de classes que envolve artefato do caso de uso Manter Artefatos.

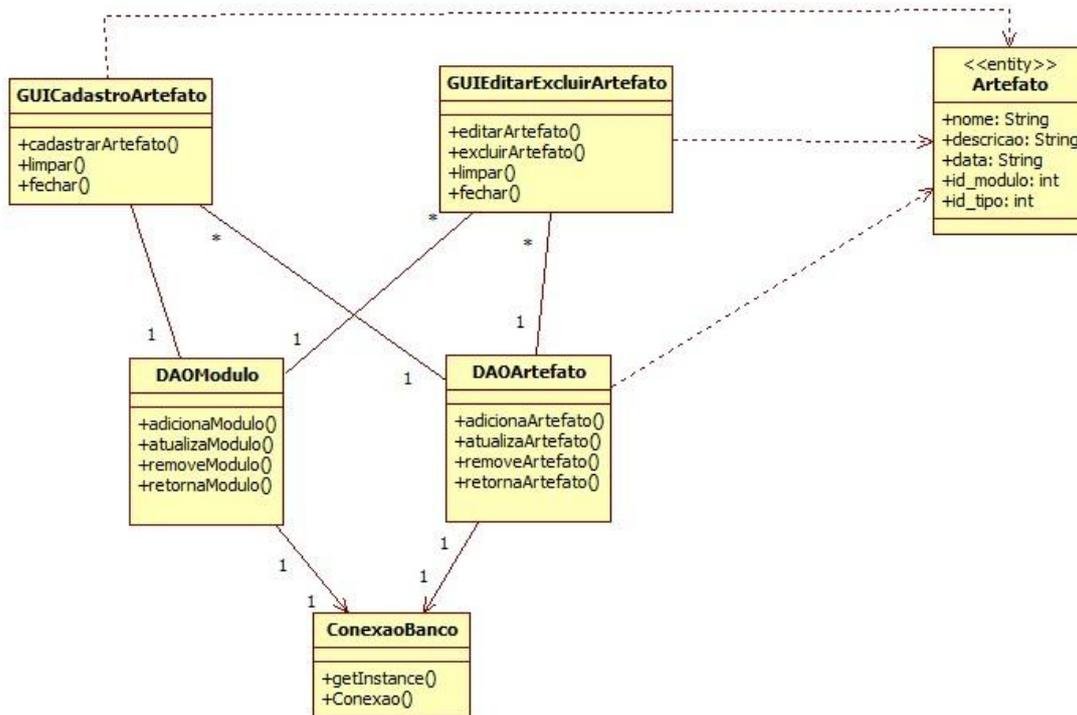


Figura 8: *Diagrama de classes do caso de uso Manter Artefatos - Artefato*

GUIEditarExcluirArtefato ó Representa a classe de fronteira por onde o usuário fornece os dados necessários para alteração de determinado artefato ou exclusão do mesmo. Os métodos *editarArtefato()* e *excluirArtefato()* efetivam as operações de edição e exclusão do artefato no sistema. Isso é possível por conta da associação com a classe *DAOArtefato*, a qual, de fato, gerencia os registros dos artefatos no banco de dados. Apresenta, também, dependência com a classe *Artefato*, por conta dos dados de artefato que a mesma apresenta. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela.

DAOArtefato ó Representa a classe responsável por gerenciar os registros de artefatos na base de dados. Gerencia as operações de inserção, atualização, remoção e consulta de artefatos de um módulo. Isso é feito através dos métodos *adicionaArtefato()*, *atualizaArtefato()*, *removeArtefato()*, *listaArtefatos()* e *retornaArtefato()*. A classe *ConexaoBanco* permite a conexão dessa classe ao SGBD alvo, e, conseqüentemente, a persistência dos dados do artefato. A classe *Artefato* é a responsável pelo acesso e modificação dos dados dos artefatos.

Artefato 6 Classe que corresponde aos dados de artefato (nome, descrição, data, modulo e tipo de artefato) a serem persistidos na Base de Dados.

3.3.3. Caso de uso Manter Fonte de Dados

GUIConectarFonteDeDados - Classe que apresenta tela para o usuário informar os dados para conexão com o SGBD. Apresenta os métodos *conectar()*, que realiza a conexão com a base de dados alvo, permitindo que o usuário execute as persistências desejadas correspondentes a cada função do sistema, o método *limpar()*, que esvazia os campos da tela, e o *fechar()*, que fecha a interface. Relaciona-se com as classes *FonteDeDados*, para acessar os dados necessários para a conexão, *DAOFonteDeDados*, para efetivar a ligação ao banco de dados. A Figura 9 apresenta o diagrama de classes do caso de uso Manter Fonte de Dados.

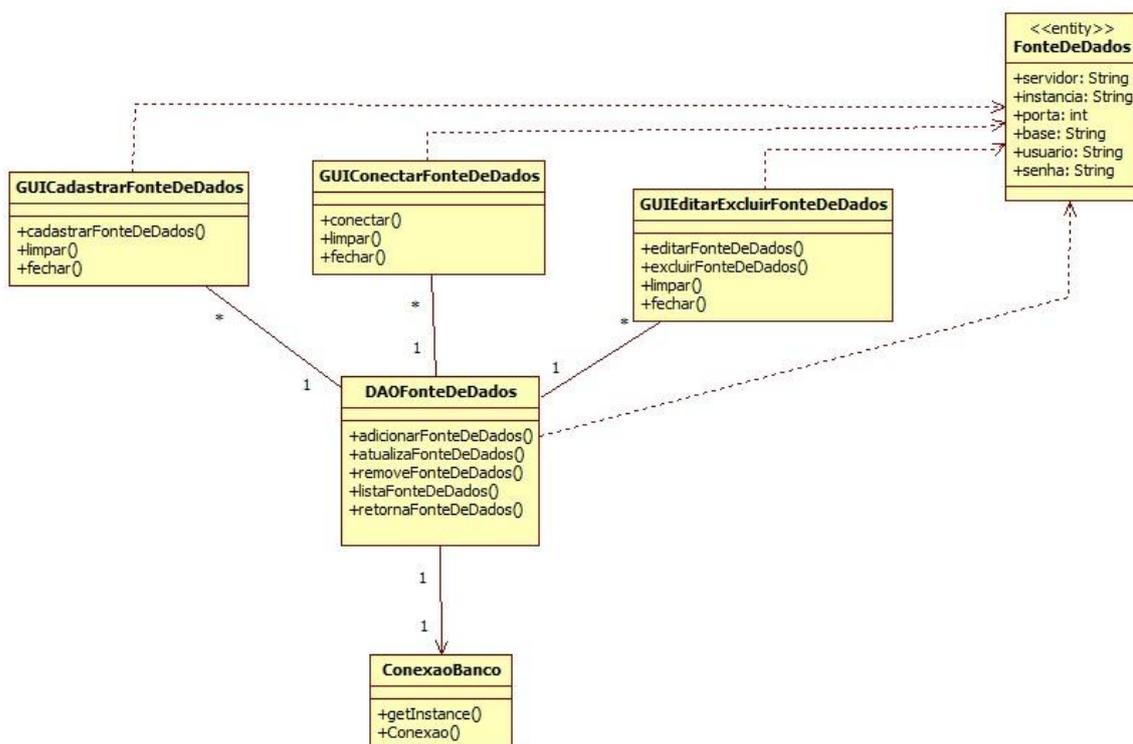


Figura 9: Diagrama de classes do caso de uso Manter Fonte de Dados

GUICadastrarFonteDeDados - É a classe representante da interface que possibilita ao usuário criar novas fontes de dados no sistema. Isso é feito através do método *cadastrarFonteDeDados()* que registra a nova fonte para uso posterior na importação de esquemas. Além disso, há os métodos *limpar()*, que esvazia os campos da tela, e *fechar()*, que fecha a tela. As classes *FonteDeDados* e *DaoFonteDeDados*, relacionadas a essa interface, permitem acessar o objeto da nova fonte com os dados a persistir na base e registrar efetivamente na base, respectivamente.

DAOFonteDeDados - Classe que executa os comandos SQL para atualizações das fontes de dados no SGBD. Os métodos *adicionaFonte()*, *atualizaFonte()*, *removeFonte()*, *listaFontes()* e *retornaFonte()* são utilizados para os comandos de inserção, atualização, remoção, consulta e recuperação diretamente no banco de dados, e o *verificaLogin()* confere se o login informado pelo usuário está contido nos registros da base. Possui relação com a classe *FonteDeDados*, para obter os dados necessários da fonte, e com a classe *ConexaoBanco*, associação que permite o seu acesso ao SGBD.

FonteDeDados ó Classe de dados que apresenta os dados de fonte de dados (servidor, instância, porta, base, usuário e senha) utilizados para conexão com o SGBD.

GUIEditarExcluirFonteDeDados ó Representa a classe de fronteira por onde o usuário fornece os dados necessários para alteração de determinada fonte de dados ou exclusão da mesma. Os métodos *editarFonteDeDados()* e *excluirFonteDeDados()* efetivam as operações de edição e exclusão da fonte no sistema. Isso é possível por conta da sua associação com a classe *DAOFonteDeDados*, a qual, de fato, gerencia os registros das fontes de dados no SGBD. Apresenta, também, dependência com a classe *FonteDeDados*, por conta dos dados da fonte que a mesma apresenta. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela.

3.3.4. Caso de uso Manter Dependência entre artefatos e esquemas

GUICriarVinculo é Classe fronteira que faz interação com o usuário para cadastro de relacionamento entre artefatos e elementos de esquema. O método *vincular()* é responsável por obter os dados dos elementos e artefato informados em tela e realizar a sua vinculação. Associa-se a classe *DAOArtefatoElemento* para efetivar a vinculação dos elementos, bem como com as classes DAO correspondentes a artefatos e a todos os elementos de esquema, para recuperar os dados necessários para a operação. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela. A Figura 10 apresenta o diagrama de classes do caso de uso Manter Dependência entre Artefatos e Esquemas.

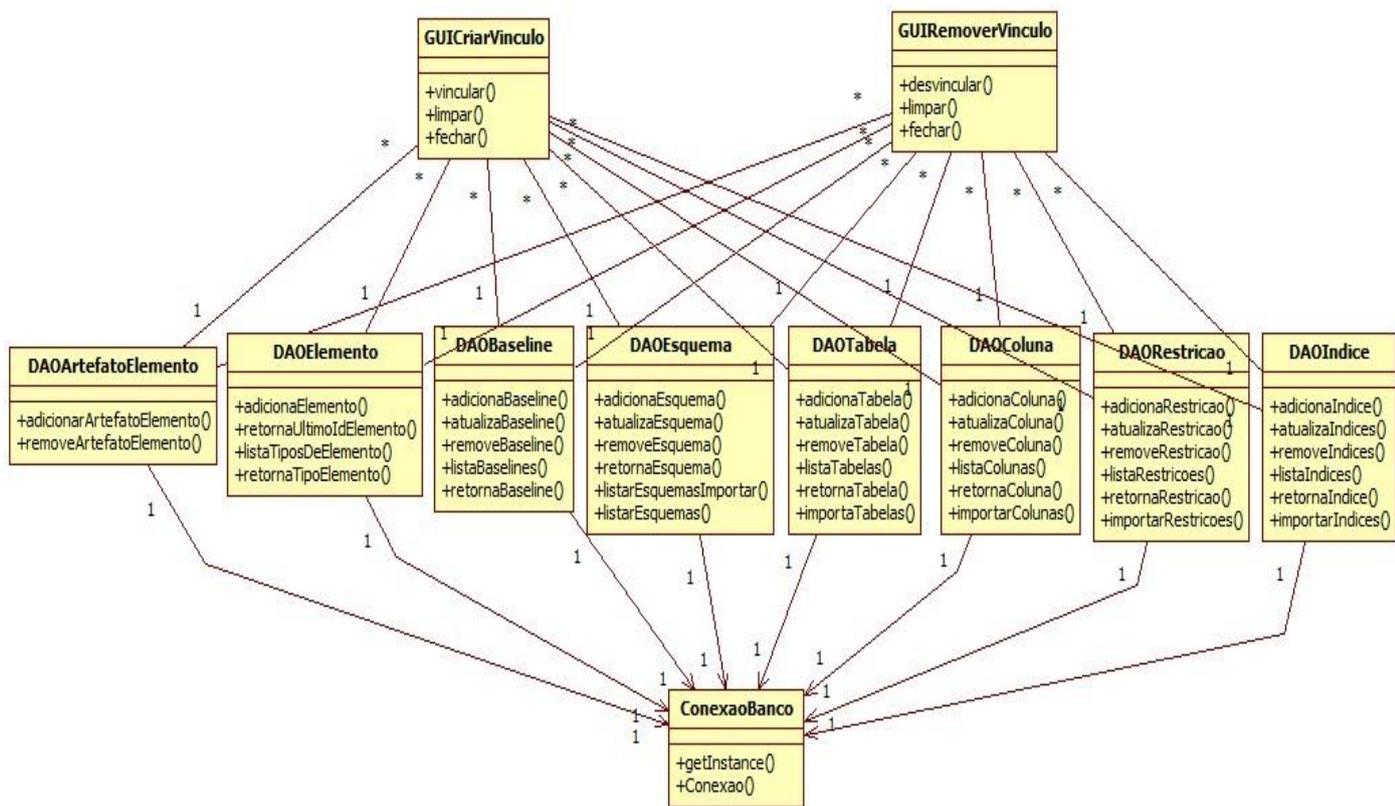


Figura 10: *Diagrama de classes do caso de uso Manter Dependência entre Artefatos e Esquemas*

GUIRemoverVinculo - Classe que apresenta tela ao usuário para o mesmo realizar a remoção de dependência entre artefatos e elementos de esquema. A tela apresenta os elementos vinculados a determinado artefato e o método *desvincular()* é responsável por quebrar essa ligação e remover a associação do artefato com os elementos selecionados em tela. Sua associação com a classe *DAOArtefatoElemento* permite a execução de

comandos no SGBD para efetivar a desvinculação. Relaciona-se também com as classes DAO correspondentes a artefatos e a todos os elementos de esquema. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela.

DAOArtefatoElemento ó Essa classe representa as operações de atualização no banco de dados para os artefatos que tem associação com elementos. O método *adicionaArtefatoElemento()* vincula elementos do tipo esquema, tabela, coluna, restrição ou índice a artefatos. Essa relação permite encontrar os artefatos que podem ser afetados caso haja alguma modificação no esquema. O método *removeElementoDeArtefato()* efetiva a remoção do vínculo criado.

3.3.5. Caso de uso Manter Mudanças de Esquemas

GUIDiferencasBaselines ó Interface que salva as informações dos baselines utilizados em comparação para detectar mudanças nos seus elementos. O método *cadastrarDiferencas()* armazena esses dados informados pelo usuário no sistema. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela. Relaciona-se com as classes *DAODiferencasBaselines*, a qual persiste os dados nas tabelas do SGBD, e *DiferencasBaselines*, classe de dados. A Figura 11 apresenta o diagrama de classes do caso de uso Manter Mudanças de Esquemas.

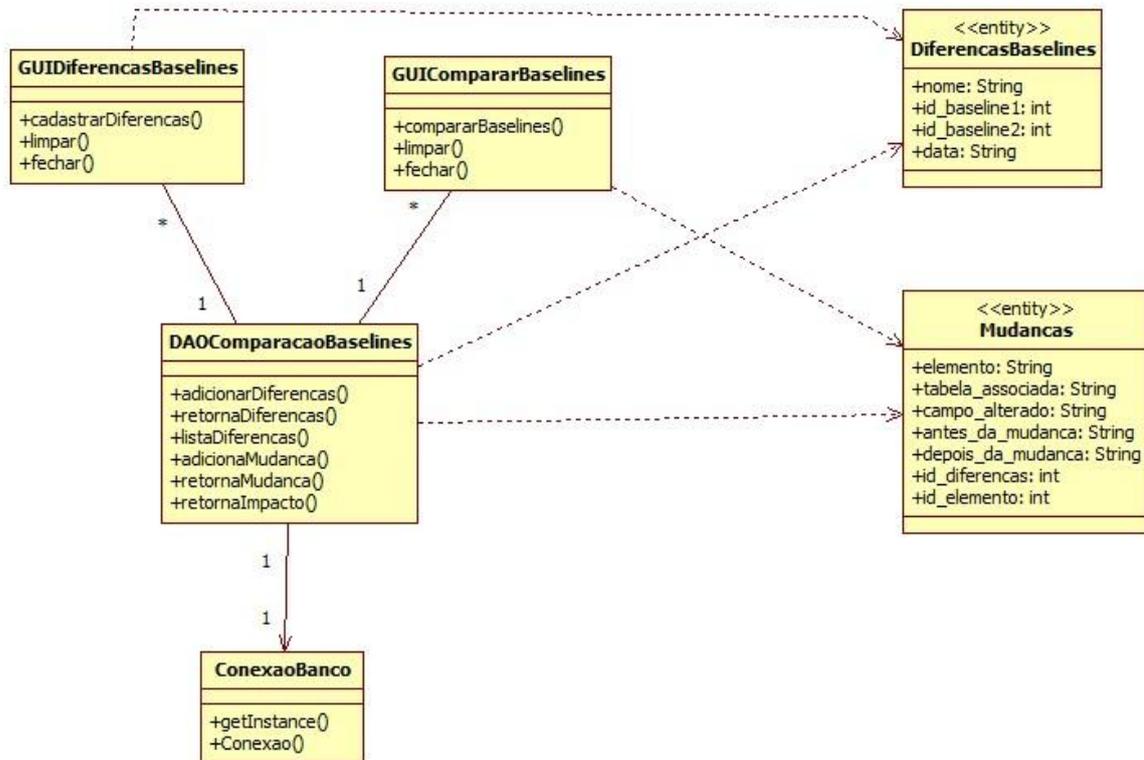


Figura 11: *Diagrama de classes do caso de uso Manter Mudanças de Esquemas*

GUICompararBaselines ó Representa a classe de interface onde o usuário informa os baselines de um esquema, com os quais deseja realizar comparação para verificar se há mudanças no esquema em questão. O método *compararBaselines()* acessa a classe *DAODiferencasBaselines* para recuperar os dados dos elementos ligados aos baselines e compará-los, a fim de verificar se houve evolução de uma versão para outra. Essas mudanças podem ser salvas no sistema com o método *salvar()*, para, posteriormente, o usuário poder recuperar com a geração de relatórios. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela.

DAODiferencasBaselines ó Concentra operações de inserção, recuperação e remoção de dados sobre mudanças entre baselines. Para tais operações, permite a comunicação do sistema com a base de dados por meio dos métodos *adicionaDiferencas()*, *retornaDiferencas()*, *listaDiferencas()* e *removeDiferencas*. Estes métodos registram os baselines que possuem diferenças. Adicionalmente, os métodos *adicionaMudanca()* e *retornaMudanca()* são responsáveis por persistir as mudanças encontradas entre as

versões do esquema. Os dados são obtidos por meio das instâncias das classes *DiferencasBaselines* e *Mudancas*, e, depois de conectado ao SGBD, os mesmos são registrados em tabela. Essa classe também permite executar a avaliação de impacto das mudanças registradas. O método *retornaImpacto()* consulta os artefatos que foram relacionados aos elementos alterados, identificando, assim, o impacto resultante das mudanças.

DiferencasBaselines é Classe que manipula os dados referentes a comparação de baselines (nome, data, primeiro baseline, segundo baseline). Esses dados representam os baselines envolvidos na identificação de alterações de esquema.

Mudancas é Classe básica que corresponde aos dados de cada mudança de elemento (elemento, tabela associada, campo alterado, conteúdo antes da mudança e conteúdo depois da mudança) a serem gravados no banco de dados.

3.3.6. Caso de uso Avaliar Impacto de Mudanças em Esquemas

GUIAvaliarImpacto é Classe que representa a tela de interface, na qual o usuário informa a comparação de baselines que deseja que seja avaliada. Como dito anteriormente, avaliar o impacto de evolução de esquema consiste em identificar quais artefatos podem ser afetados por uma alteração no esquema. O relacionamento entre artefatos e elementos persistido em tabela no banco de dados facilita essa tarefa. O método *avaliarImpacto()* acessa a classe *DAODiferencasBaselines* para efetivar a avaliação de impacto de mudanças em esquemas. A classe *DAODiferencasBaselines*, conectada ao SGBD alvo, verifica o relacionamento dos elementos alterados com os artefatos do sistema por meio de uma consulta, identificando quais podem ser afetados pelas mudanças. O método *salvar()* permite salvar um relatório do impacto gerado. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela. A Figura 12 apresenta o diagrama de classes do caso de uso Avaliar Impacto de Mudanças em Esquemas.

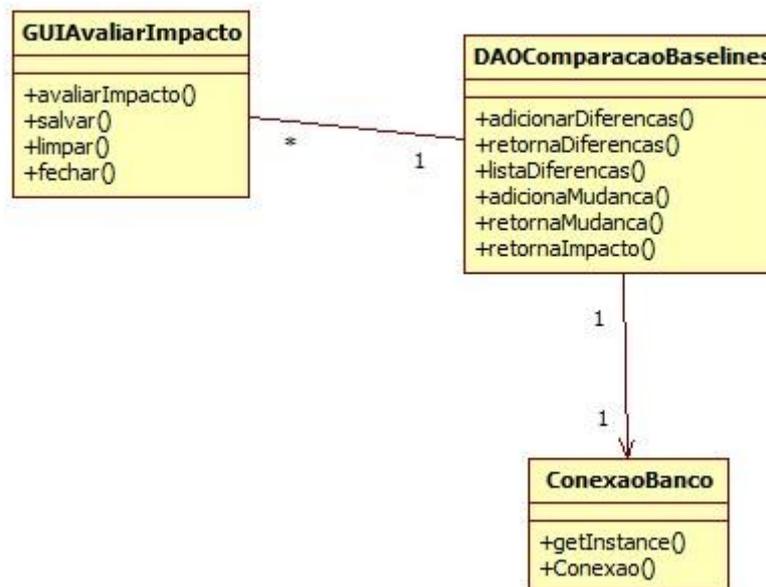


Figura 12: *Diagrama de classes do caso de uso Avaliar Impacto de Mudanças em Esquemas*

3.3.7. Caso de uso Gerar Relatórios

GUIRelatorioComparacao é Classe que apresenta os relatórios de comparação entre baselines de um esquema. Fornece ao usuário a lista das comparações salvas entre baselines para poder gerar relatórios das mesmas. Possui o método `gerarRelatorio()` que salva um documento no formato pdf com informações sobre os baselines do esquema que foram comparados, e sobre as mudanças encontradas nos elementos desse esquema. O método `limpar()` esvazia os campos da tela, enquanto que `fechar()` fecha a tela. Relaciona-se com a classe `DAODiferencasBaselines`, para recuperar os dados das mudanças no banco de dados. A Figura 13 apresenta o diagrama de classes do caso de uso Manter Relatórios.

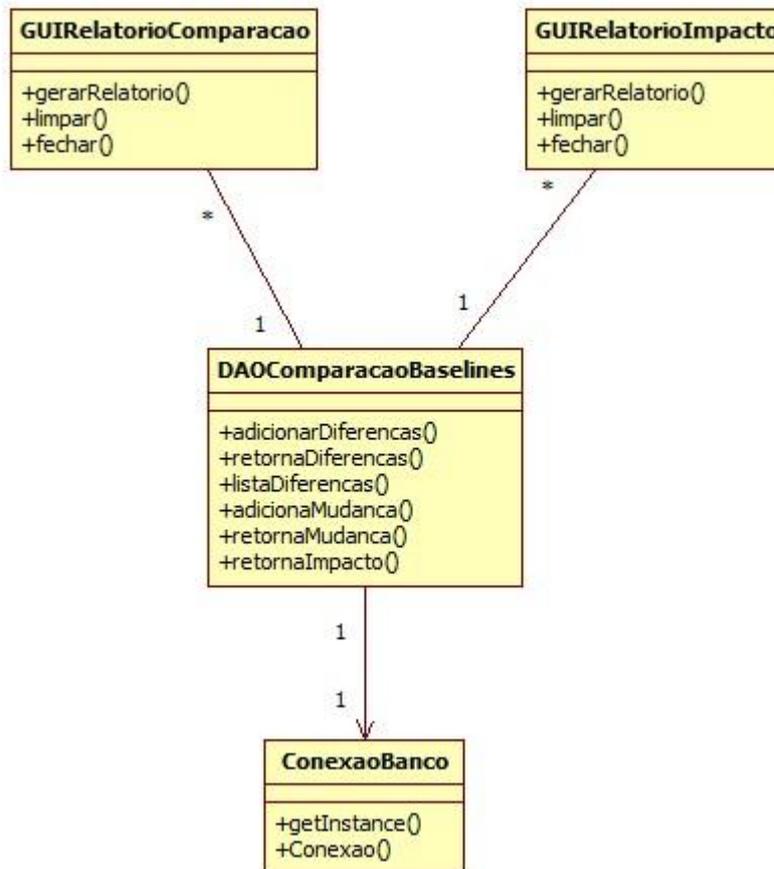


Figura 13: *Diagrama de classes do caso de uso Gerar Relatórios*

GUIRelatorioImpacto é Representa a classe de interface que permite ao usuário gerar relatórios de impacto de evolução de esquema. Fornece ao usuário a lista de comparações salvas entre *baselines* para, então, avaliar o impacto das mudanças, encontrando os artefatos afetados. O método *gerarRelatorio()* salva um documento no formato pdf com informações sobre os elementos modificados e os artefatos impactos por estes. O método *limpar()* esvazia os campos da tela, enquanto que *fechar()* fecha a tela. Relaciona-se com a classe *DAODiferencasBaselines*, para retornar a avaliação das mudanças do banco de dados.

3.4. Modelagem de Dados

Após a criação dos diagramas de classe, foi modelado o projeto lógico do Sistema. A Figura 19 mostra a modelagem de dados da ferramenta. Nesta seção detalharemos as tabelas do modelo.

3.4.1. Componentes do modelo de dados

Caracterização de esquemas: Todas as informações sobre os esquemas cadastrados são persistidas na tabela TB_ESQUEMA. Para que haja controle de histórico, e, conseqüentemente, controle sobre a evolução de esquema, um esquema deve possuir um ou mais *baselines*. Os dados sobre os baselines de cada esquema são armazenados na tabela TB_BASELINE. Para importar corretamente um esquema de um SGBD alvo, a ferramenta precisa saber quais tabelas, colunas, restrições e índices estão relacionados a cada *baseline* de um esquema determinado. Todas essas informações são armazenadas nas tabelas TB_TABELA, TB_COLUNA, TB_RESTRICOES, TB_COLUNA_RESTRICAO, TB_INDICE e TB_COLUNA_INDICES. Essas tabelas também são utilizadas para armazenar os dados de um esquema inseridos manualmente na ferramenta.

Caracterização de elementos: A tabela TB_ELEMENTO foi criada para facilitar o relacionamento de dependência entre artefatos e esquemas. Um elemento pode ser um esquema, tabela, coluna, índice ou restrição. Quando uma tabela, por exemplo, for cadastrada e armazenada em TB_TABELA, será registrado também um elemento do tipo TABELA e relacionado à tabela cadastrada. Dessa forma, quando for preciso vincular um artefato a uma tabela, o relacionamento será feito do artefato para o elemento correspondente a tabela desejada. Logo, não é preciso que a tabela TB_ARTEFATO esteja ligada a todas as dependências de um esquema, mas somente a tabela TB_ELEMENTO. A Figura 14 representa a modelagem das tabelas relacionadas a esquema e elemento.

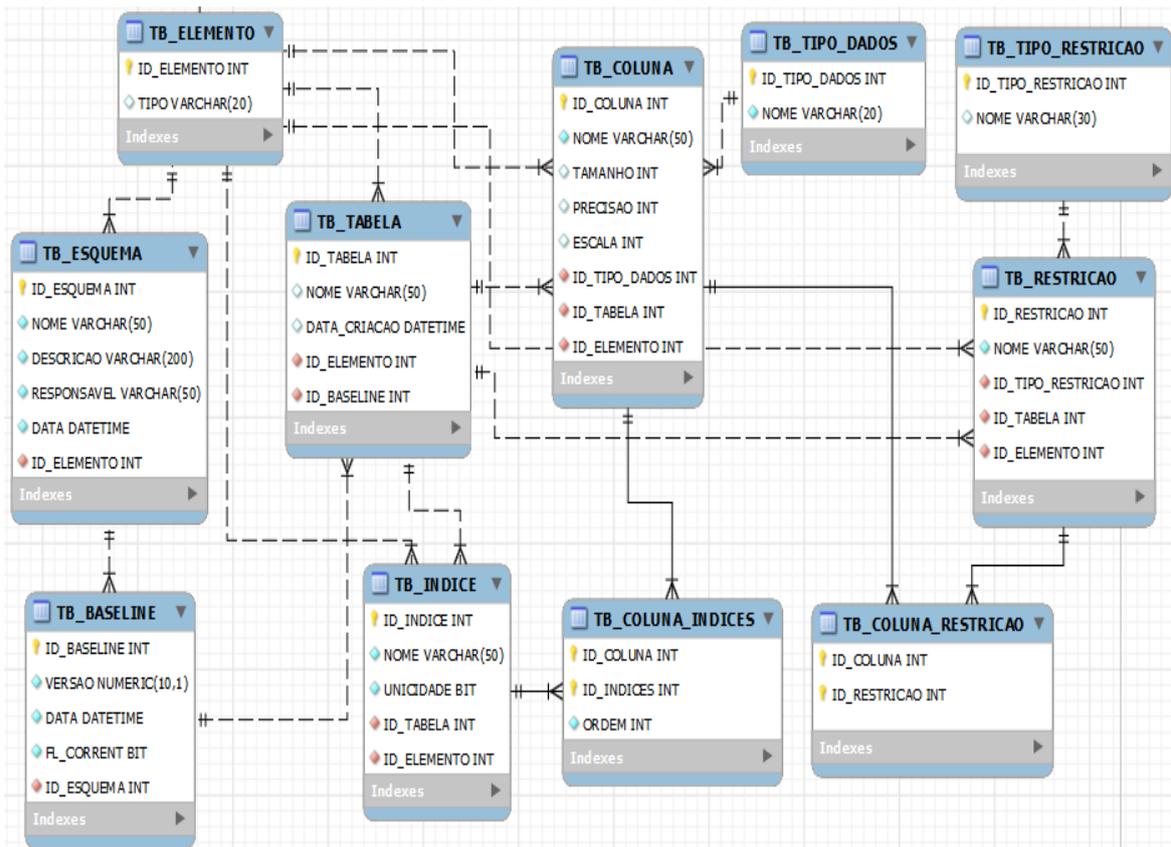


Figura 14: *Esquema de dados da ferramenta ó Esquemaö*

Caracterização de artefatos: A ferramenta precisa persistir dados sobre artefatos de um projeto de desenvolvimento, que vão ser relacionados a esquema de dados e, por fim, avaliados sobre o impacto da evolução de esquema. Um artefato pode ser qualquer classe de código, documentação ou casos de teste. O mesmo está relacionado a um módulo de um sistema. Portanto, a tabela TB_TIPO_ARTEFATO persiste os possíveis tipos de artefato, a tabela TB_SISTEMA armazena as informações sobre um sistema, TB_MODULO as informações sobre um módulo relacionado a um sistema, e TB_ARTEFATO os dados sobre um artefato relacionado a um tipo e um módulo. A Figura 15 representa a modelagem das tabelas citadas anteriormente.

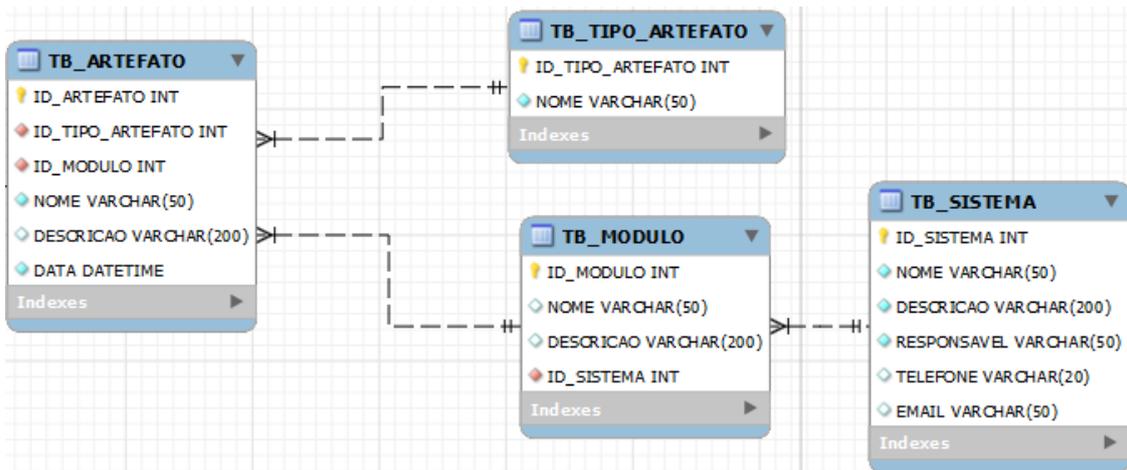


Figura 15: *Esquema de dados da ferramenta ó Artefatoö*

Caracterização de fonte de dados: Através de informações armazenadas na tabela TB_FONTE_DE_DADOS, a ferramenta consegue identificar quais servidores, instâncias e usuários (com seus respectivos login e senha) possuem esquemas para importação, bem como as conexões possíveis para monitoramento de mudanças de esquema. A Figura 16 representa a tabela de fonte de dados do modelo lógico da ferramenta.

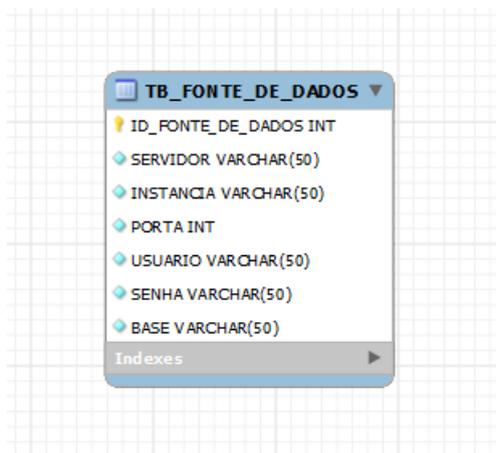


Figura 16: *Esquema de dados da ferramenta ó Fonte de dadosö*

Caracterização de artefatos afetados: Para a ferramenta conseguir identificar quais artefatos podem sofrer impactos por uma alteração no esquema de dados, ela precisa relacionar os artefatos aos elementos de um esquema. Se em um sistema, por exemplo, uma classe de código tem interação com duas tabelas do esquema, na ferramenta o usuário precisa vincular o artefato em questão a essas tabelas. Essas relações são

armazenadas na tabela TB_ARTEFATO_AFETADO. Dessa forma, quando for preciso avaliar o impacto de mudanças em um esquema, as informações armazenadas nessa tabela nos mostra quais artefatos podem ser afetados por cada elemento atualizado. A Figura 17 apresenta o modelo de dados com o relacionamento da tabela TB_ARTEFATO_AFETADO.

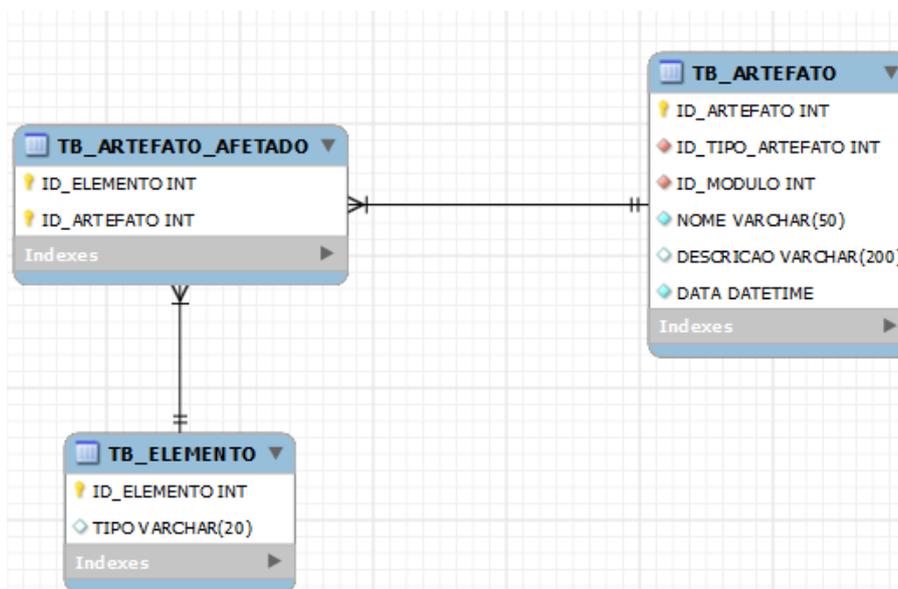


Figura 17: *Esquema de dados da ferramenta ó ãArtefato afetadoö*

Caracterização de mudanças: Para avaliar o impacto de mudanças em esquemas, é preciso ter controle sobre as mudanças realizadas em uma versão (*baseline*) do esquema. É possível descobrir as alterações comparando uma versão com outra de um mesmo esquema de dados. As informações sobre as mudanças encontradas, tais como elemento modificado, conteúdo do elemento antes e depois da mudança, são persistidas na tabela TB_MUDANÇAS. A tabela TB_DIFERENCAS_BASELINE registra quais *baselines* foram comparados e a data desse feito. Dessa forma, com conhecimento de quais elementos foram atualizados, é possível identificar os artefatos potencialmente impactados. Relatórios sobre diferenças de *baseline* e avaliação de impacto podem ser gerados com as informações sobre a evolução de esquema, armazenadas nas tabelas citadas acima. A modelagem desses componentes está representada na Figura 18.

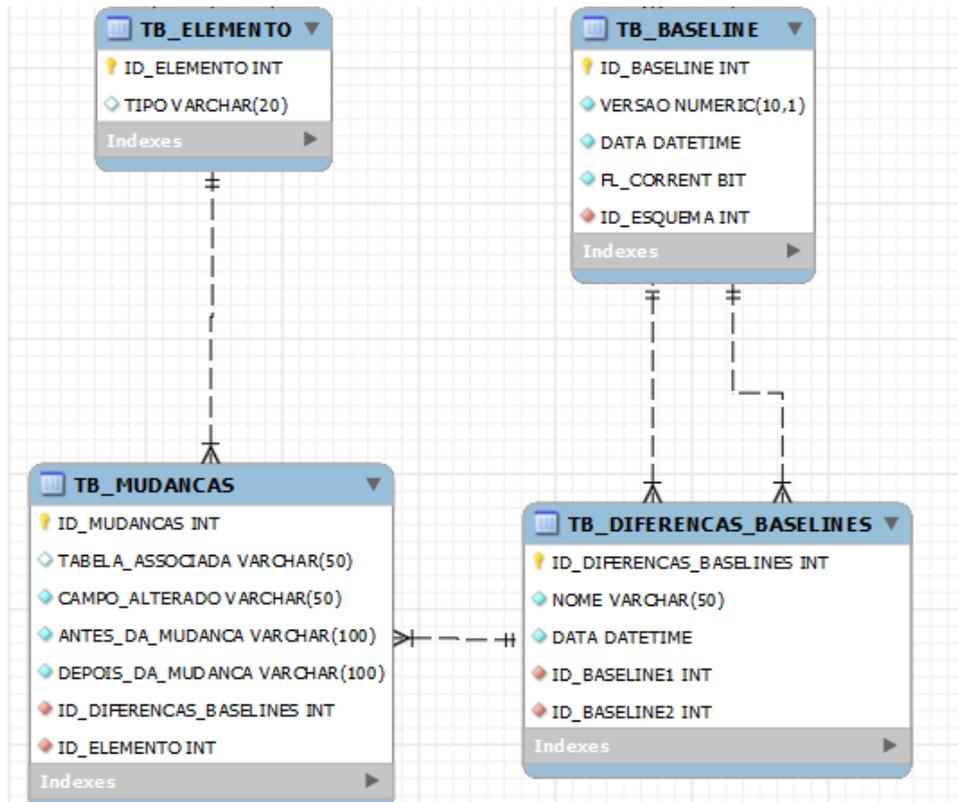


Figura 18: *Esquema de dados da ferramenta ó Mudanças de esquema*

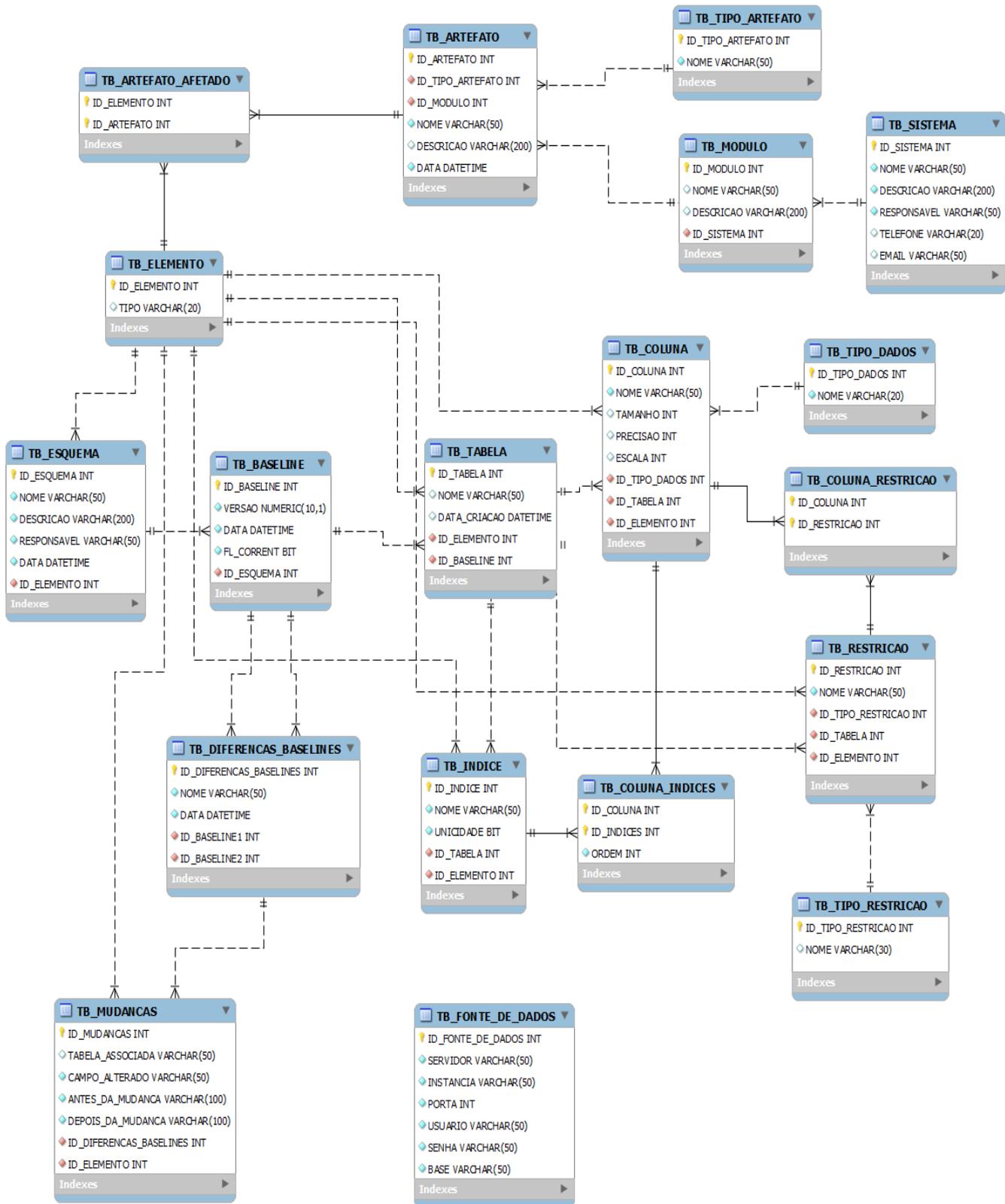


Figura 19: Esquema de dados da ferramenta

4. FERRAMENTA

A ferramenta desenvolvida tem como objetivo a detecção e análise de impactos na evolução de esquemas em SGBDs relacionais.

Nesta seção apresentaremos um estudo de caso com um esquema de dados criado para uma rede de cinemas, cujas tabelas fornecem informações sobre diversas transações de venda de ingressos e produtos. Esse esquema foi criado durante a disciplina Sistemas de Informação, para um projeto de *Data Warehouse*. Faremos uso do esquema do ambiente operacional, *CINEMA_OLTP*, para demonstrar o funcionamento da ferramenta.

Menu Principal

A Figura 20 apresenta o menu principal da ferramenta. As opções do menu *Esquema*, *Artefato*, *Vincular*, *Impacto* e *Relatórios* permanecem inativos até que o usuário se conecte ao SGBD alvo.

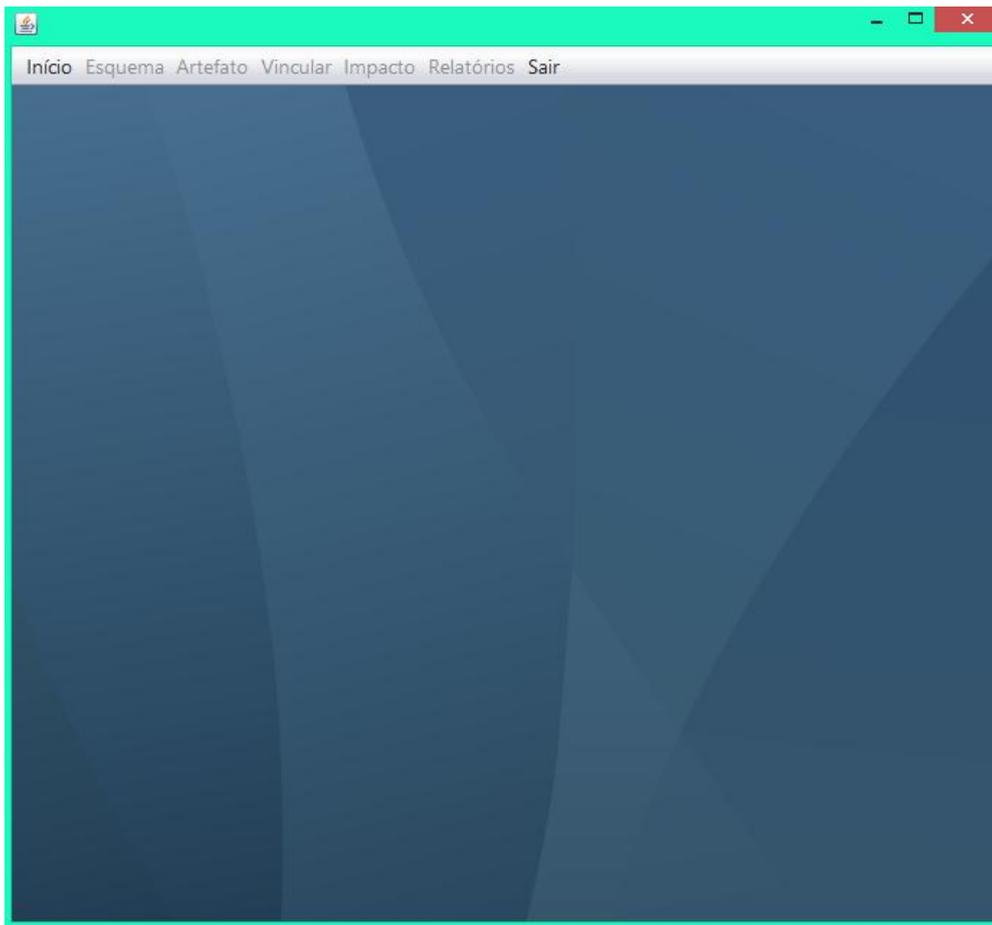


Figura 20: *Tela do Menu Principal da ferramenta*

Conectar

O usuário acessa a opção *Conectar* do menu *Iniciar* para conexão com o banco de dados da ferramenta. A Figura 21 apresenta a tela na qual o usuário informa o servidor, instância do banco, o nome da base, e seu usuário e senha, usados para conexão com o banco de dados *DatabaseEvolution*, esquema de dados da ferramenta. Depois que a conexão é realizada, as demais funcionalidades do sistema tornam-se ativas.

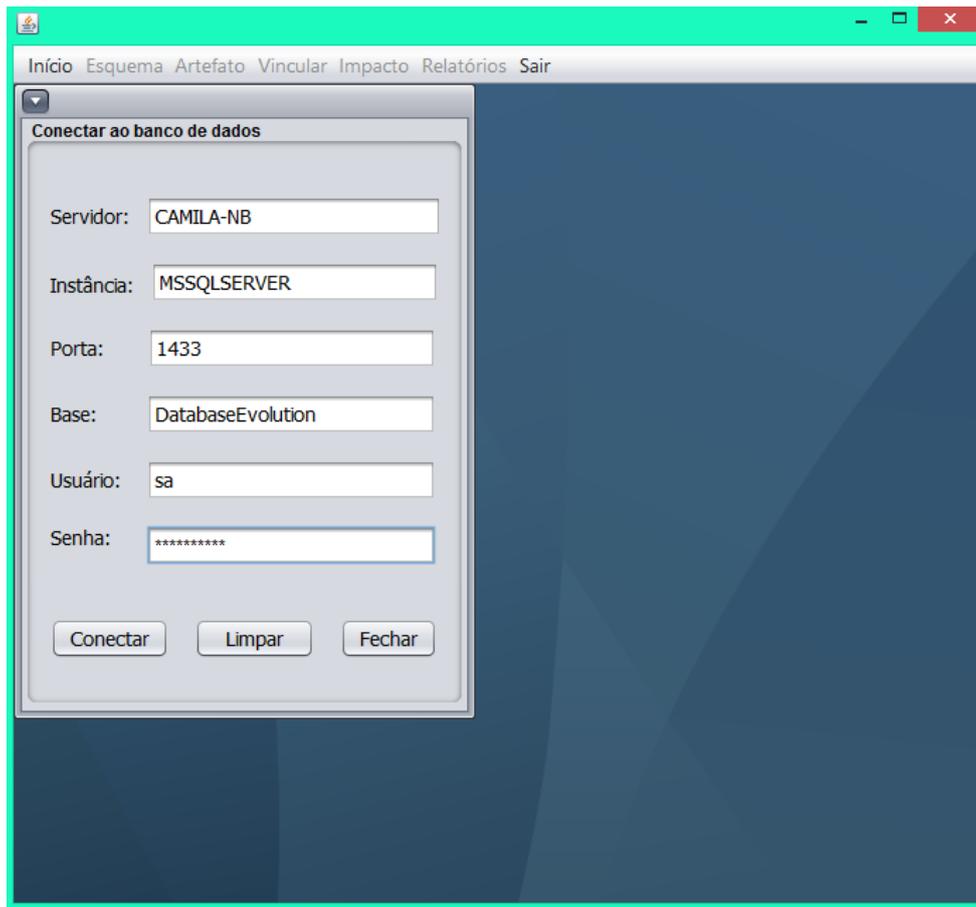


Figura 21: *Tela de Conexão da ferramenta*

Criar Fonte de Dados

No menu *Iniciar* existe a função de criar uma nova fonte de dados, a ser usada para importação de esquemas de dados. A Figura 22 mostra a tela com os dados para criação da fonte de dados para acessar a base *CINEMA_OLTP*.

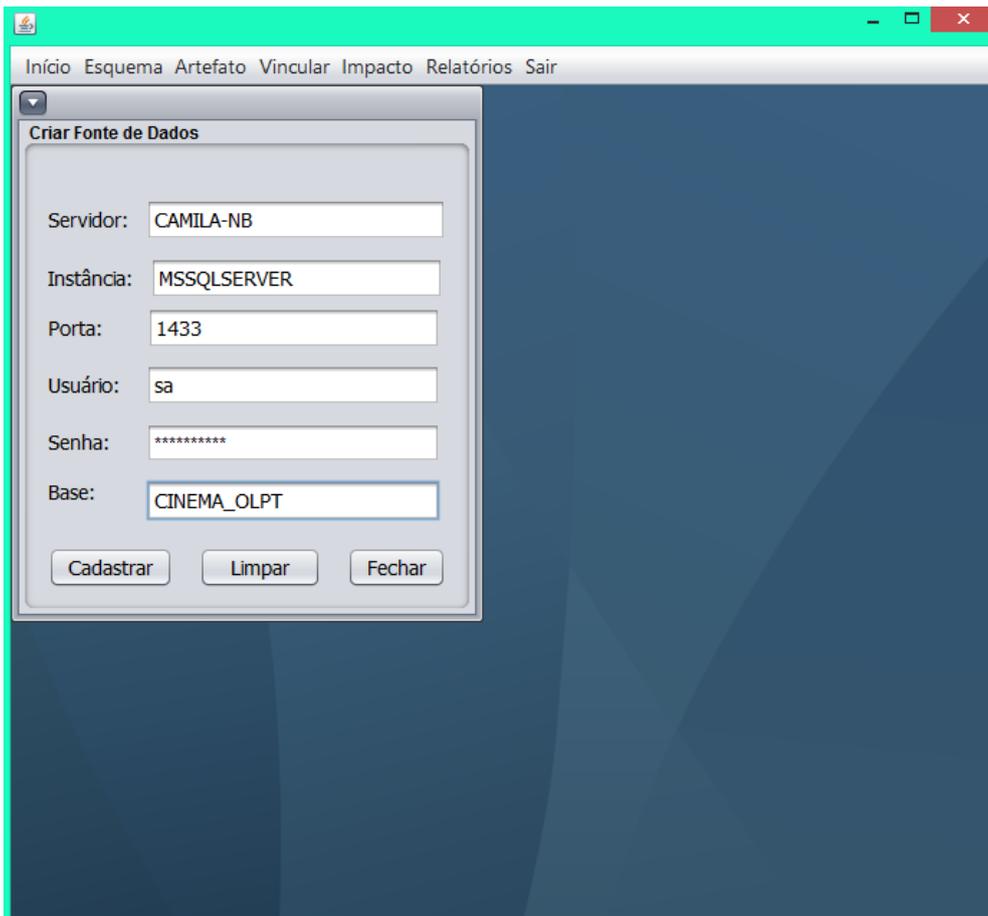


Figura 22: *Tela para criação de Fonte de Dados da ferramenta*

Criar Esquema

Nessa tela o usuário cadastra um novo esquema, no qual serão inseridos tabelas, colunas, restrições e índices, ou importa um esquema do banco de dados. Quando um novo esquema é criado, seu primeiro *baseline* (versão) também é criado. O esquema criado na Figura 23 foi *Cinema*.

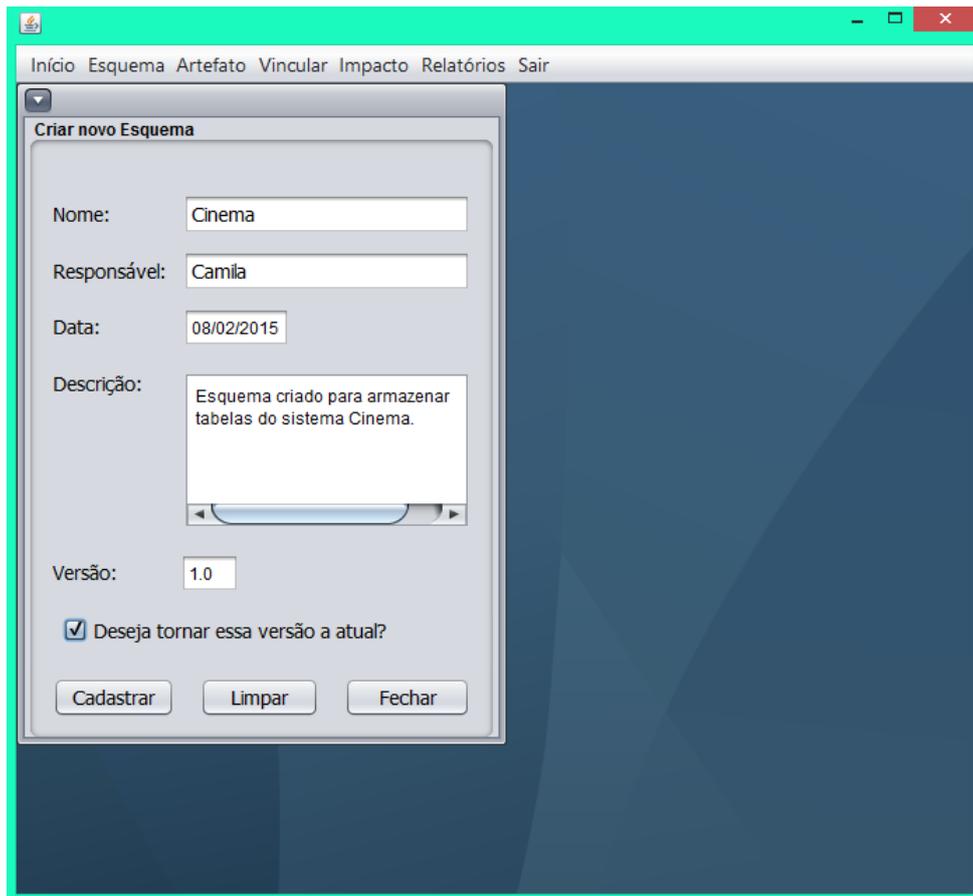


Figura 23: *Tela de cadastro de Esquema da ferramenta*

Criar Baseline

Nessa tela o usuário cadastra uma nova versão para o esquema que deseja. O esquema desejado é selecionado, no caso o esquema *Cinema*, e ao lado são apresentadas as versões já existentes para o esquema, e abaixo são informadas versão, data e se deseja tornar esse *baseline* o atual. O *baseline* 1.1 criado na Figura 24 para o esquema *Cinema*.

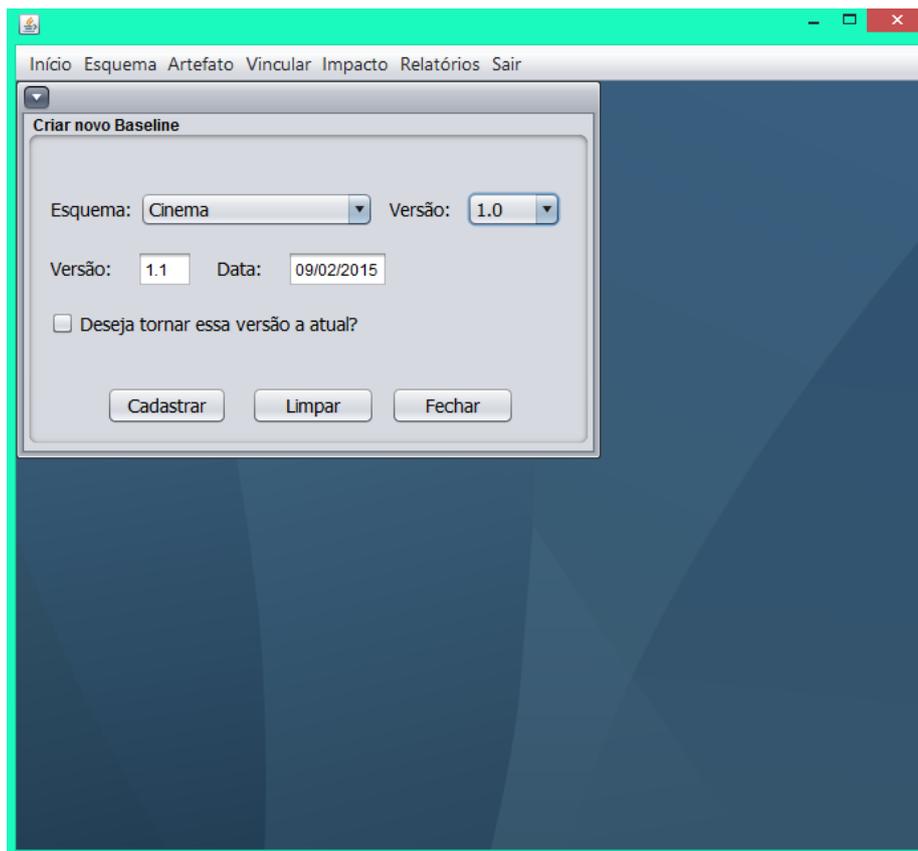


Figura 24: Tela para criação de Baseline da ferramenta

Importar Esquema

Na tela de importação de esquema há um campo no topo da tela informando a fonte de dados conectada no momento, inicialmente conectada à base de dados da ferramenta *DatabaseEvolution*. O usuário pode alterar a fonte de dados clicando em *alterar*, que abre uma nova tela, Figura 26, com as fontes de dados cadastradas para seleção. O usuário conecta-se à base *CINEMA_OLTP* para importação do esquema criado para rede de cinemas. Na Figura 25 o usuário escolhe, então, o esquema que irá receber a importação, *Cinema*, e o esquema a importar da base conectada. São apresentadas em tela as tabelas do esquema a importar selecionado, e o usuário pode, então, efetivar a importação.

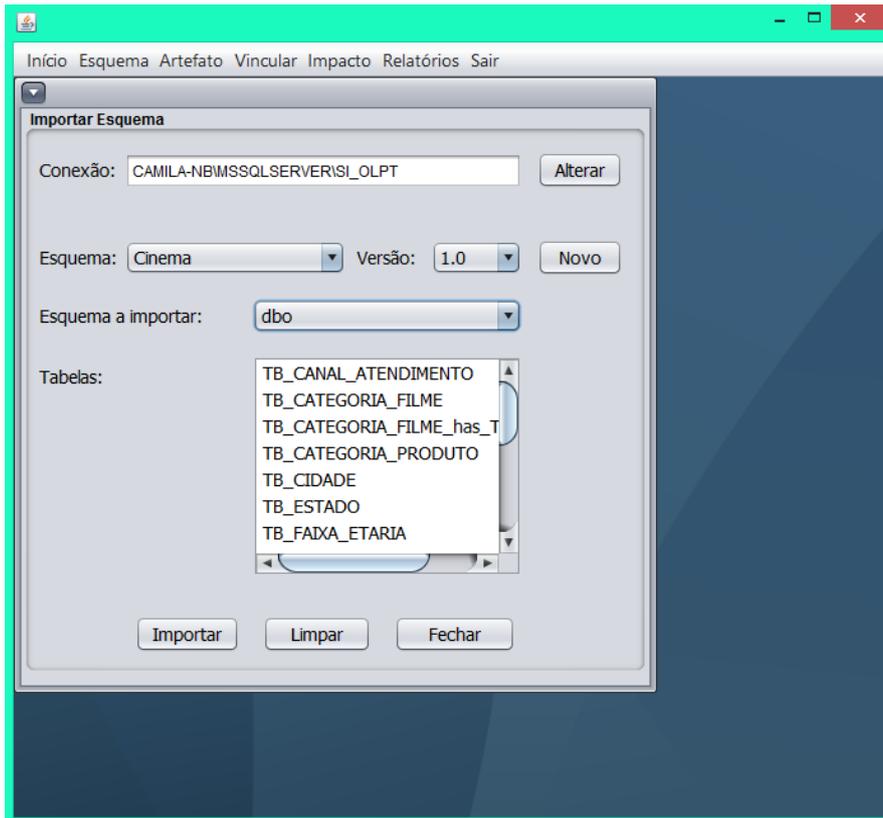


Figura 25: Tela para Importar Esquema da ferramenta

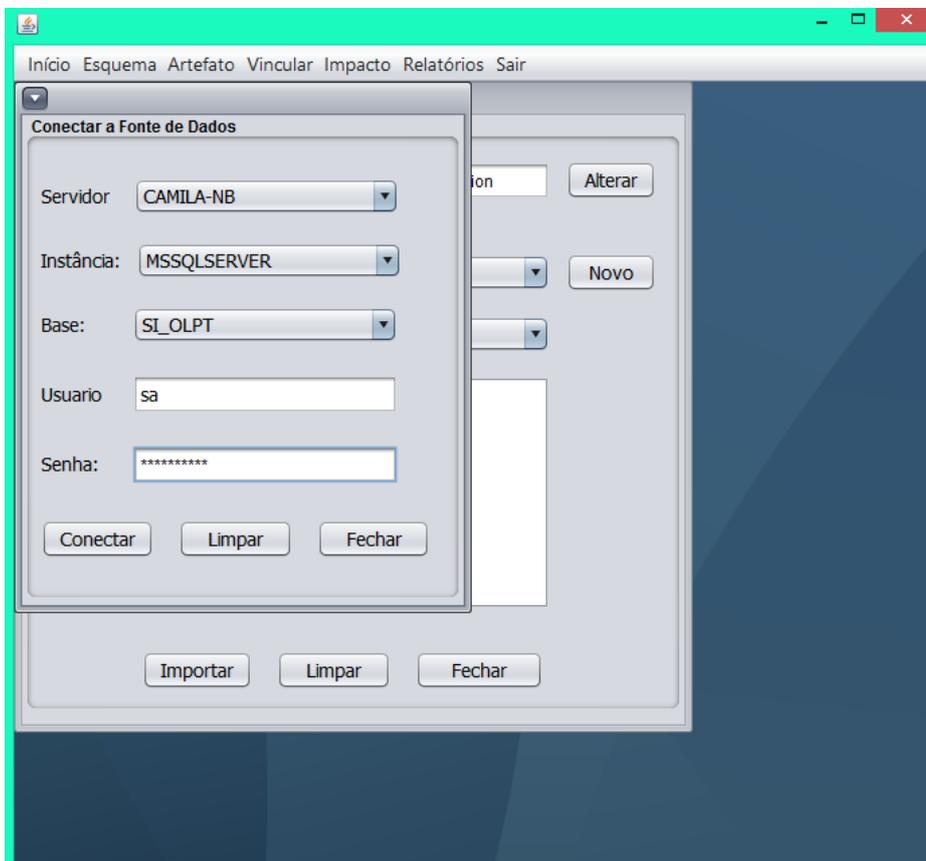


Figura 26: Tela para alterar Fonte de Dados da ferramenta

Editar Esquema

O usuário pode inserir, editar ou remover tabelas, colunas, restrições ou índices para a versão do esquema desejada. Adicionalmente, é possível criar um novo esquema ou atualizar os existentes. Inicialmente, o usuário seleciona o tipo de elemento que deseja editar. Dependendo do tipo selecionado, é preciso escolher o esquema e versão, e no caso de colunas, restrições ou índices, seleciona-se a tabela associada. A Figura 27 apresenta a tela de edição para o esquema *Cinema*. O Tipo escolhido foi *TABELA*, e abaixo são apresentadas as tabelas cadastradas no baseline em questão.

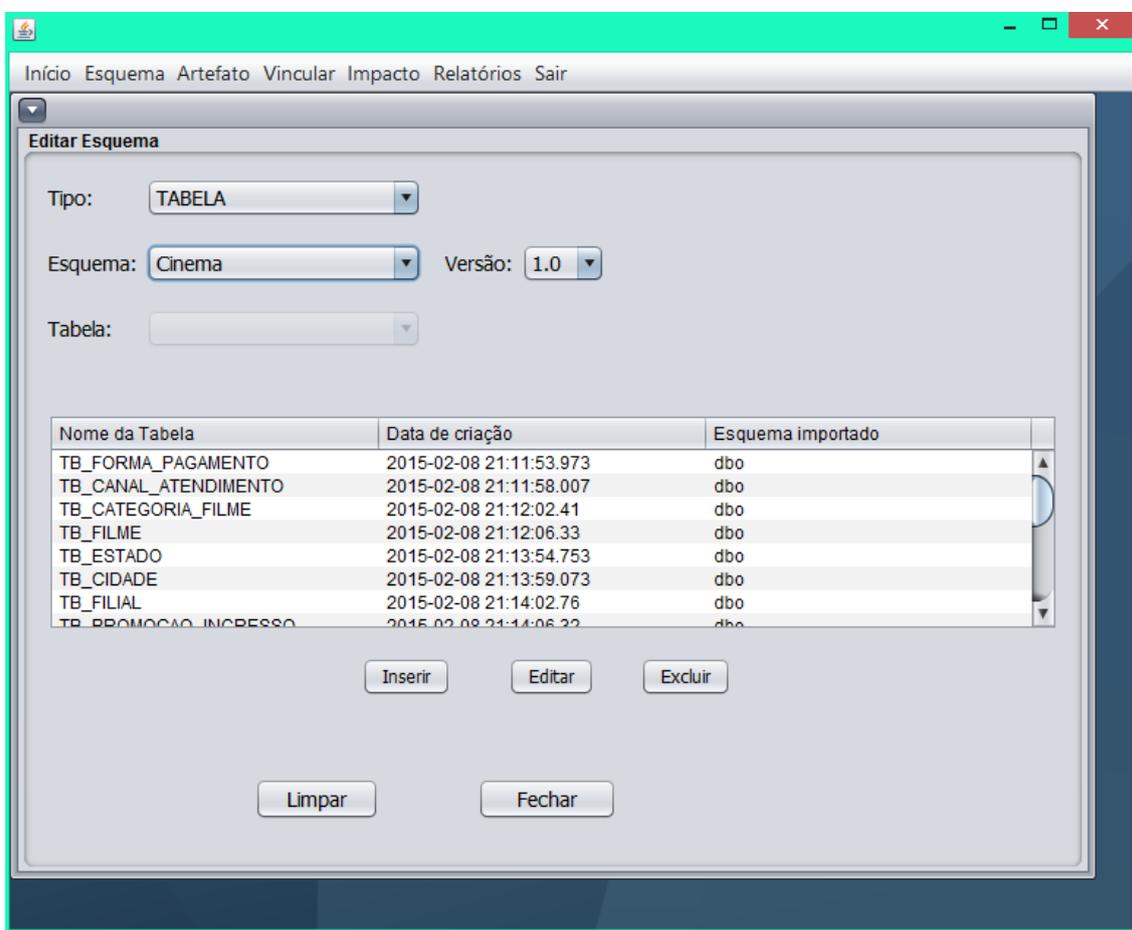


Figura 27: Tela para Editar Esquema da ferramenta

Cadastrar Sistema

Na parte de artefatos, a ferramenta apresenta funcionalidades para cadastro, edição e remoção de sistemas, módulos e artefatos em si, os quais serão alvo de

avaliação de impacto de mudanças de esquema. Para o estudo de caso, iremos cadastrar o Sistema *CinemaX*. A Figura 28 apresenta a tela de cadastro de sistemas.

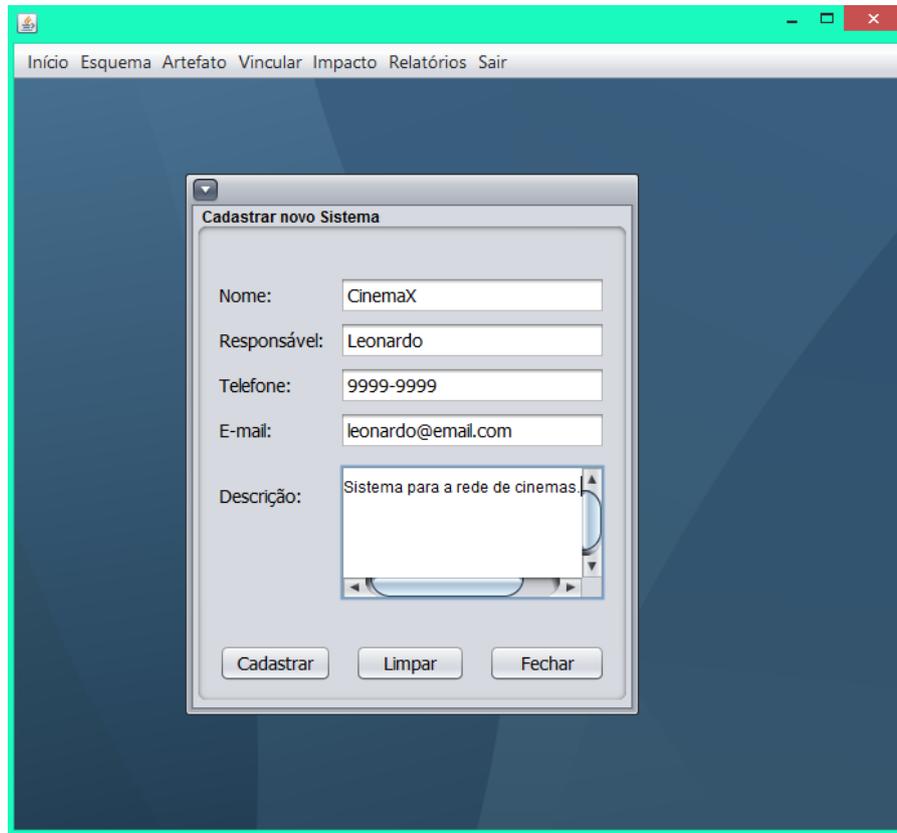


Figura 28: Tela para cadastrar Sistema da ferramenta

Cadastro de Módulo

Os módulos Vendas será cadastrado para o sistema *CinemaX*. A Figura 29 apresenta a tela para esse cadastro.

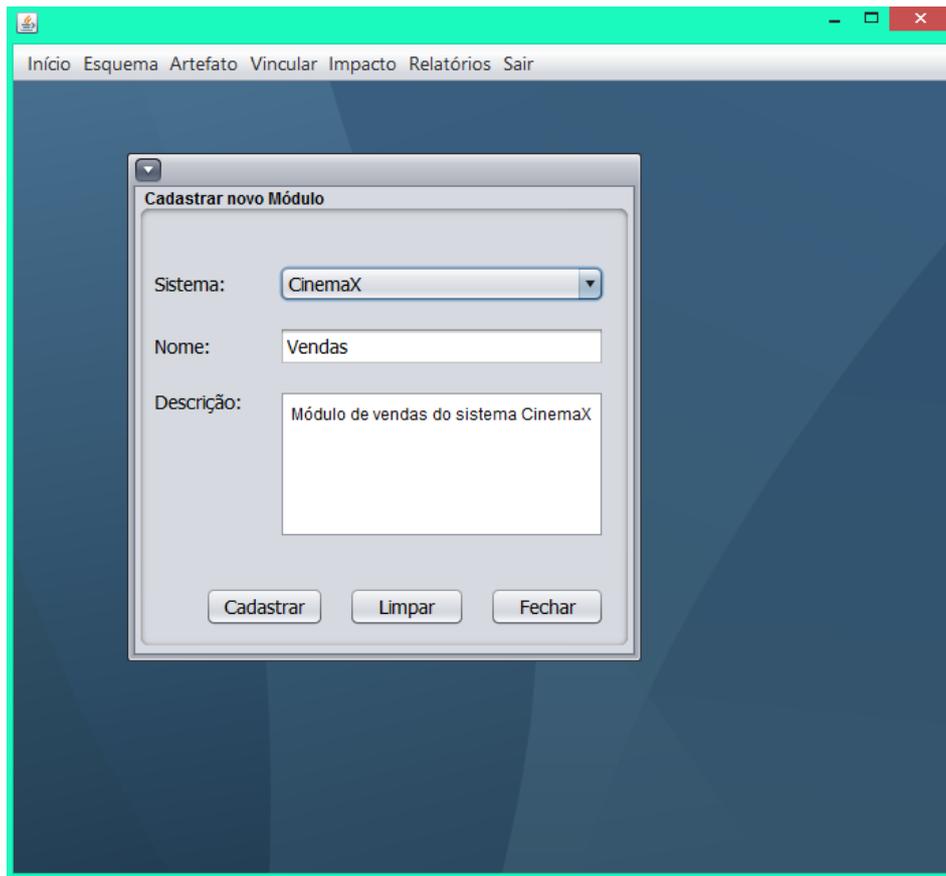


Figura 29: Tela para cadastrar Módulo da ferramenta

Cadastrar Artefato

Na tela de cadastro de artefato, o usuário escolhe o sistema e módulo desejado e informa o tipo do artefato, que pode ser Código, Documentação ou Teste. Os artefatos a serem criados serão *Venda de Produto* e *Venda de Ingresso*, do tipo código, os quais, posteriormente, serão vinculados a elementos do esquema *Cinema*. A Figura 30 mostra a tela de criação para o artefato *Venda de Ingresso*.

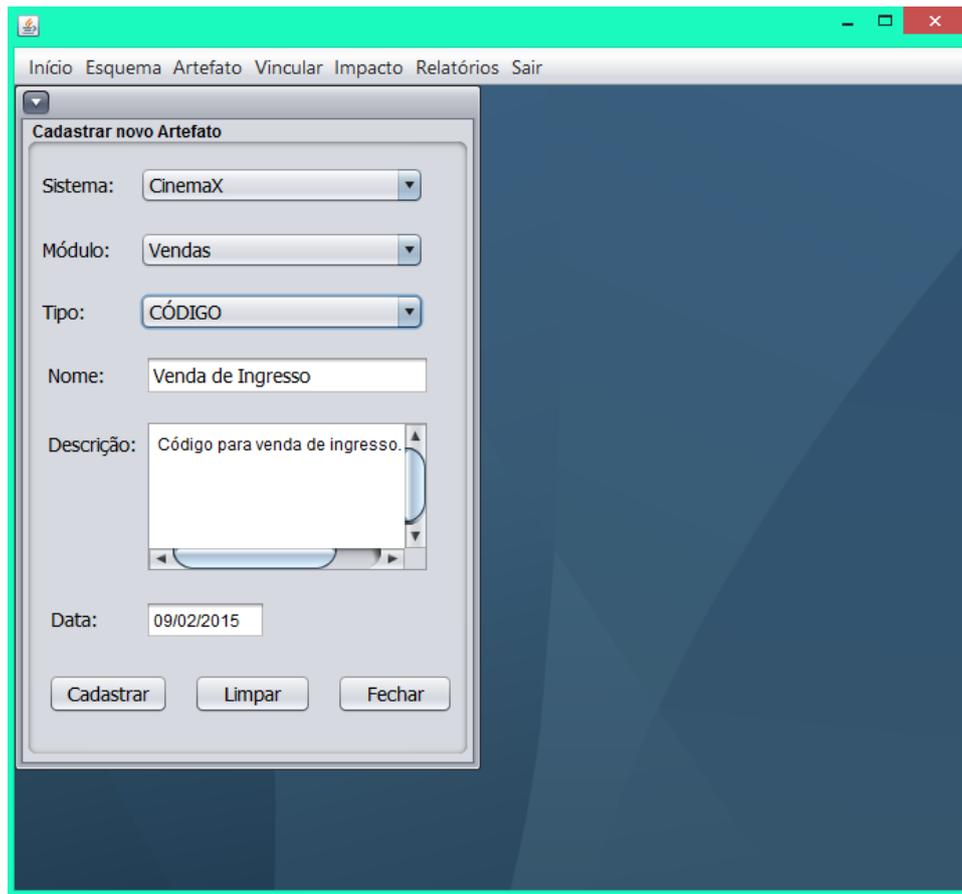


Figura 30: Tela para cadastrar Artefato da ferramenta

Criar Vínculo

Essa função consiste em relacionar artefatos e elementos de um esquema, gerando dependência entre os mesmos. Essa dependência é importante para o objetivo da ferramenta. Quando os elementos modificados forem identificados, os vínculos que esses elementos possuem vão dizer quais artefatos vão ser afetados pelas modificações. A Figura 31 mostra a tela de vinculação com o artefato *Venda de Ingresso* selecionado. Quando o usuário seleciona o esquema e sua versão, são listadas tabelas, colunas, índices ou restrições dependendo do tipo de elemento selecionado. O artefato *Venda de Ingresso* vai ser vinculada à coluna de código de cada tabela.

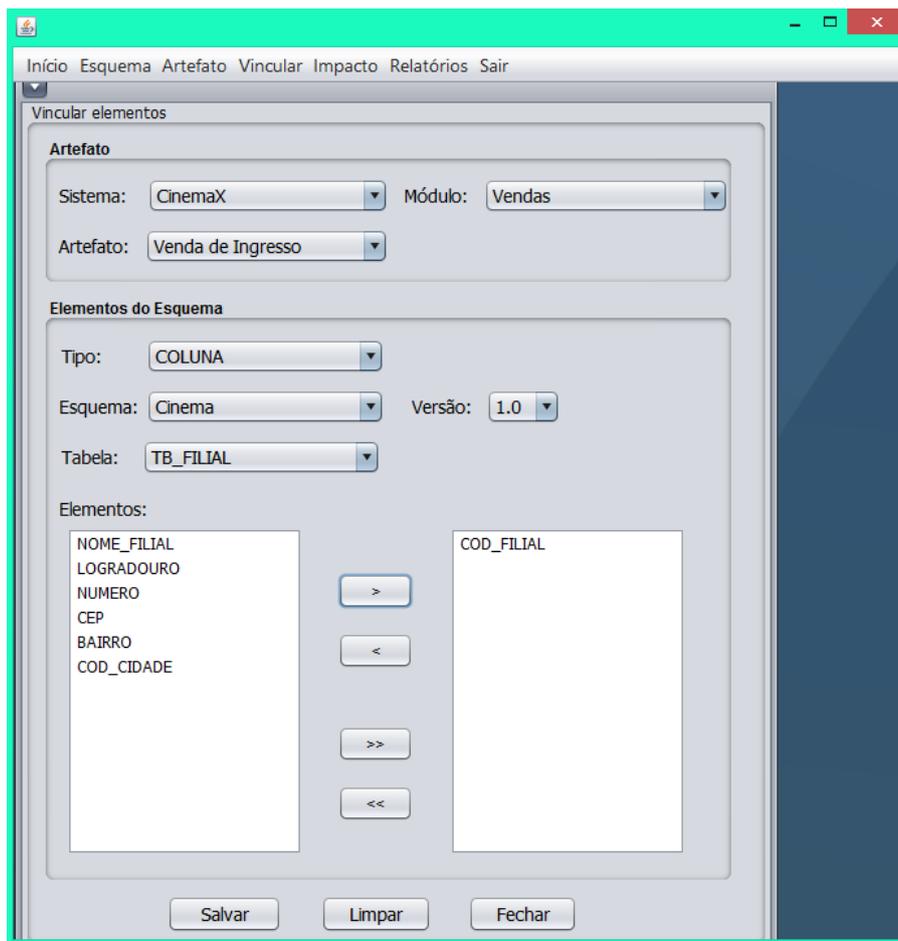


Figura 31: Tela para vincular Artefatos e Elementos da ferramenta

Remover Vínculo

Ao contrário da função anterior, essa função consiste em remover os relacionamentos entre artefatos e elementos de um esquema. Essa desvinculação é necessária quando uma dependência não se faz mais necessária ou a mesma foi criada erroneamente. A Figura 32 mostra a tela de desvinculação, com o artefato *Venda de Ingresso* selecionado. Quando o usuário seleciona o tipo de elemento, são listados os elementos vinculados ao artefato para esse tipo. O artefato *Venda de Ingresso* vai ser desvinculado de alguns campos de código que são desnecessários, tais como *COD_CIDADE* e *COD_ESTADO*, pois o artefato já é dependente do código da tabela filial, o qual já apresenta os dados correspondentes a estado e cidade.

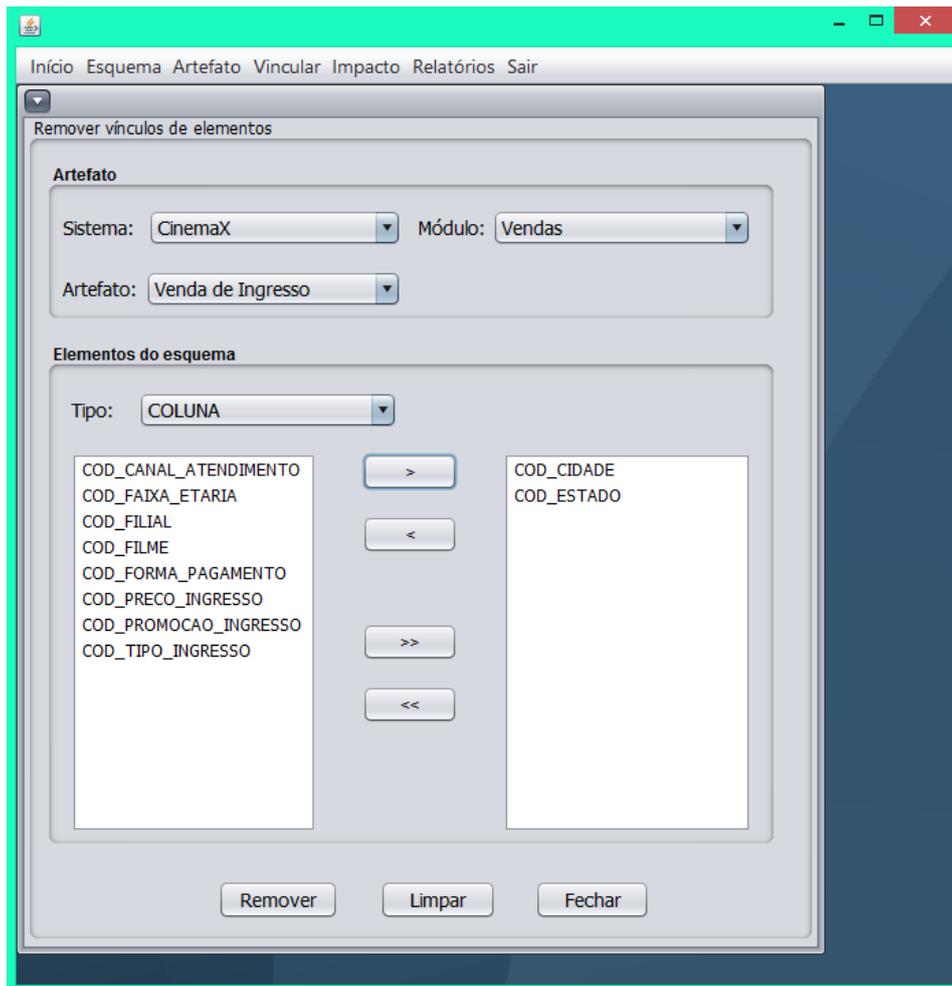


Figura 32: Tela para desvincular Artefatos de Elementos da ferramenta

Comparar Baselines

Foi feita algumas alterações no esquema de dados da base *CINEMA_OLTP* diretamente no SGBD. Para avaliar o impacto dessas mudanças é preciso salvá-las na ferramenta. Para isso foi importado o esquema atualizado para a nova baseline 1.1. Na Tela de comparação, presente na Figura 33, o usuário seleciona os baselines do esquema *Cinema* que possuem elementos. Em seguida, seleciona a opção de comparar. Na tabela no centro da tela são listadas as mudanças encontradas, identificando o elemento alterado, o tipo de elemento, o tipo de operação (inserção, atualização ou remoção) e as modificações efetuadas. O usuário então salva essa comparação, informando o nome desejado, nesse caso "Comparação do esquema Cinema 1.0 e 1.1" e a data da realização.

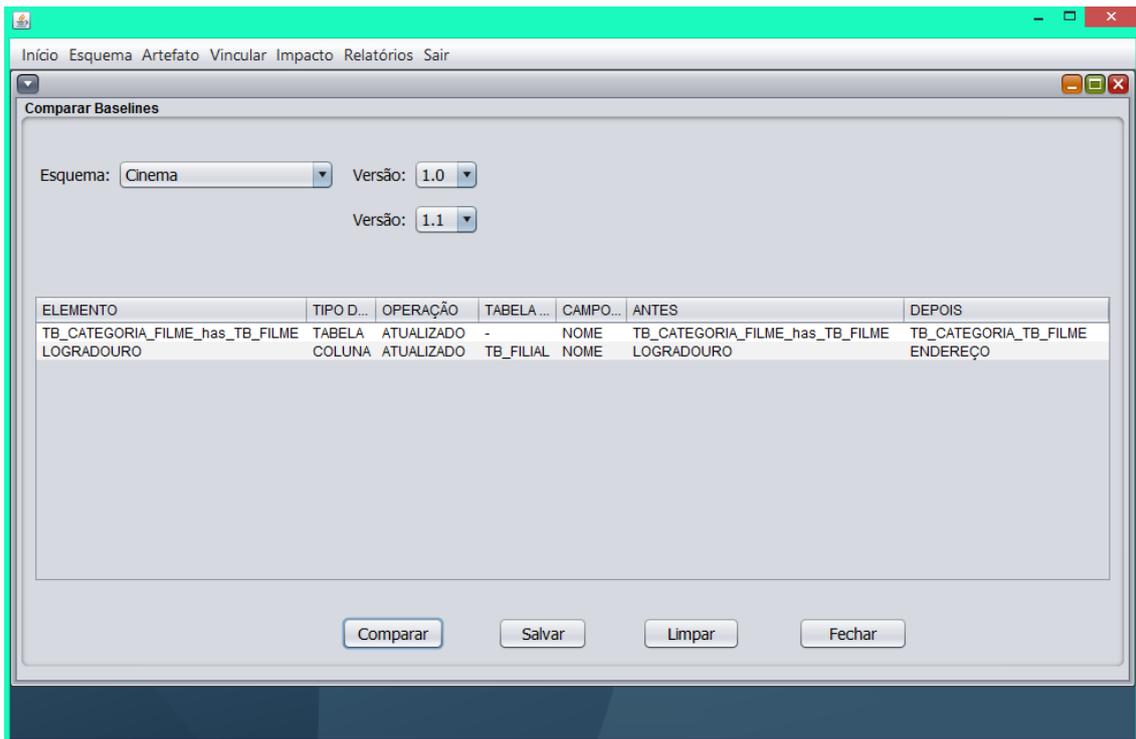


Figura 33: Tela para comparar Baselines da ferramenta

Avaliar o impacto

Com a comparação entre as versões do esquema salvas, é possível encontrar os artefatos potencialmente afetados. Na tela de avaliação (Figura 34), o usuário seleciona a comparação, anteriormente salva. São apresentados o esquema, os baselines e a data registrada na comparação para efeito de conhecimento por parte do usuário e escolha correta da comparação. Feito isso, a tabela no centro da tela apresenta as mudanças encontradas na comparação. O usuário escolhe as modificações das quais deseja detectar o impacto e clica em *Avaliar Impacto*. Caso deseje avaliar o impacto de as alterações, basta clicar no botão sem selecionar nenhuma linha da tabela. São exibidos na tela os dados do elemento alterado associado ao artefato relacionado. É possível salvar a avaliação de impacto gerando um relatório nessa tela ou na própria tela de Relatórios de Impacto (Figura 37).

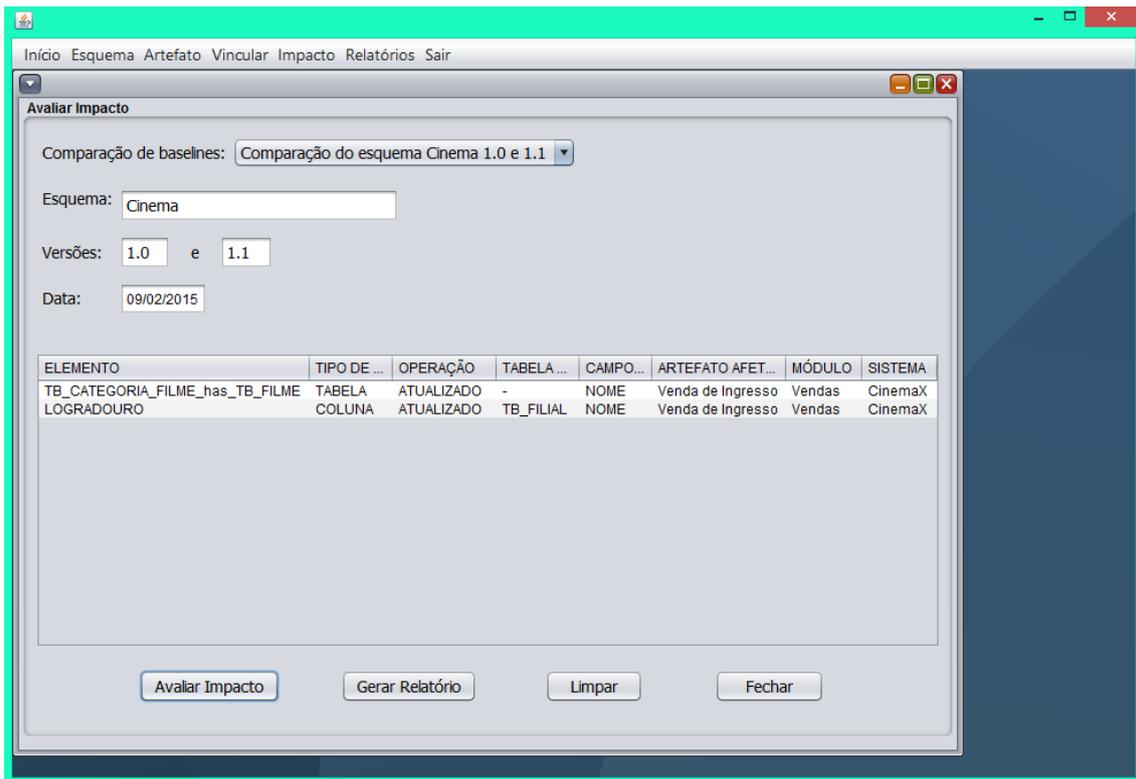


Figura 34: Tela para avaliar impacto da ferramenta

Gerar Relatório de Comparação de Baselines

Consiste em escolher uma comparação de baselines salva no sistema e gerar um documento pdf com as informações sobre essa comparação e mudanças encontradas. A Figura 35 apresenta a tela com as opções de escolha da comparação a ser registrada e o diretório para salvar o documento. A comparação escolhida foi a "Comparação do esquema Cinema 1.0 e 1.1" e o diretório foi a Área de Trabalho. A Figura 36 mostra o relatório que é gerado.

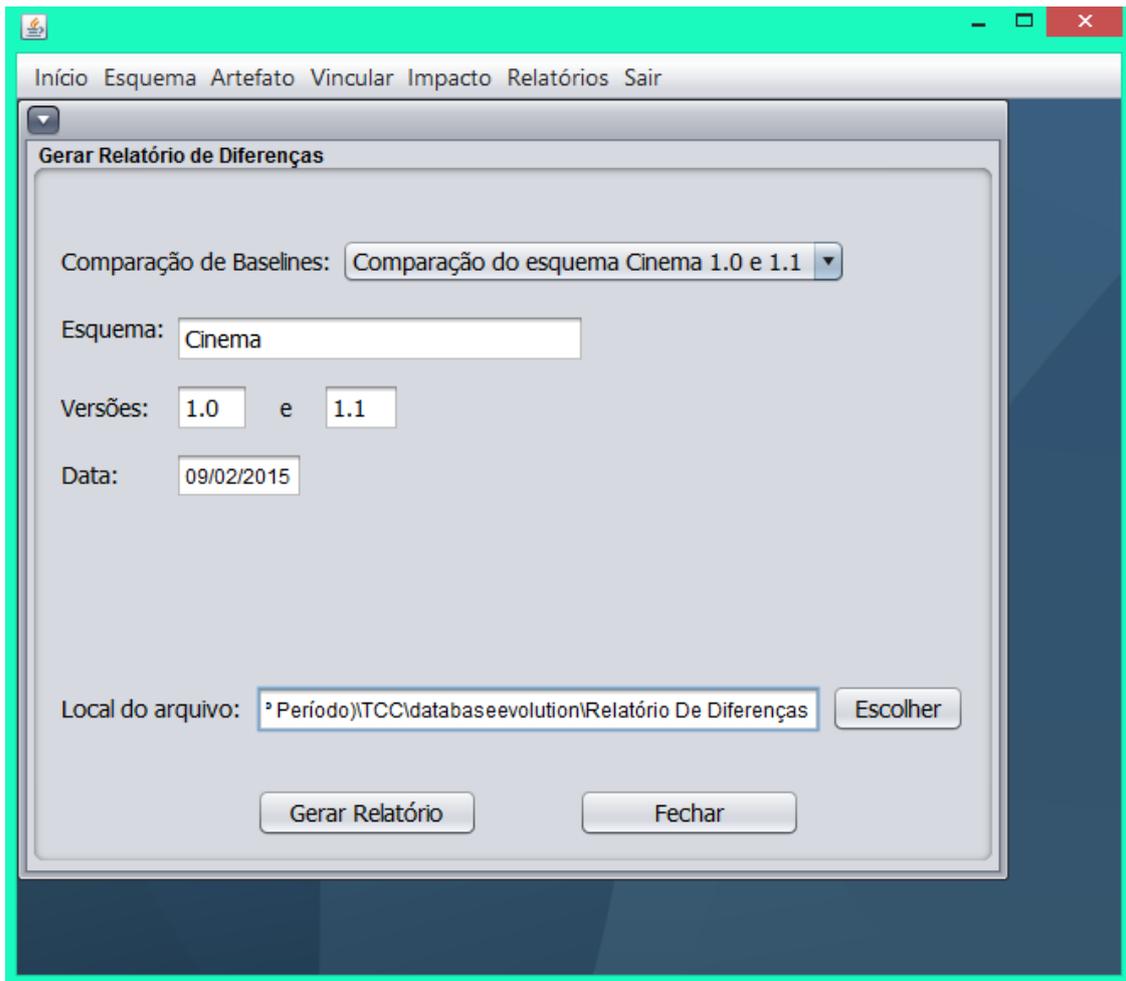


Figura 35: Tela para gerar relatório de diferenças de baselines da ferramenta

Relatório de Diferenças entre Baselines

Esquema: Cinema
 Versão Antes: 1.0
 Versão Depois: 1.1

Id do Elemento	Elemento	Tipo de Elemento	Operação	Tabela Associada	Campo Alterado	Antes	Depois
2536	TB_CATEGORIA_FILME_has_TB_FILME	TABELA	ATUALIZADO	-	NOME	TB_CATEGORIA_FILME_has_TB_FILME	TB_CATEGORIA_TB_FILME
2563	LOGRADOURO	COLUNA	ATUALIZADO	TB_FILIAL	NOME	LOGRADOURO	ENDEREÇO

Figura 36: Relatório de diferenças gerado pela ferramenta

Gerar Relatório de Avaliação de Impacto

Semelhantemente à tela anterior, a tela da Figura 37 apresenta a tela com as opções de escolha da comparação a ser registrada e o diretório para salvar o documento. Nessa função será gerado um documento com o relatório de impacto de mudanças em esquemas. Serão registradas no documento as mudanças encontradas na comparação juntamente com os artefatos vinculados a estes. A comparação escolhida foi a Comparação do esquema Cinema 1.0 e 1.1 e o diretório foi a Área de Trabalho. A Figura 38 mostra o relatório gerado.

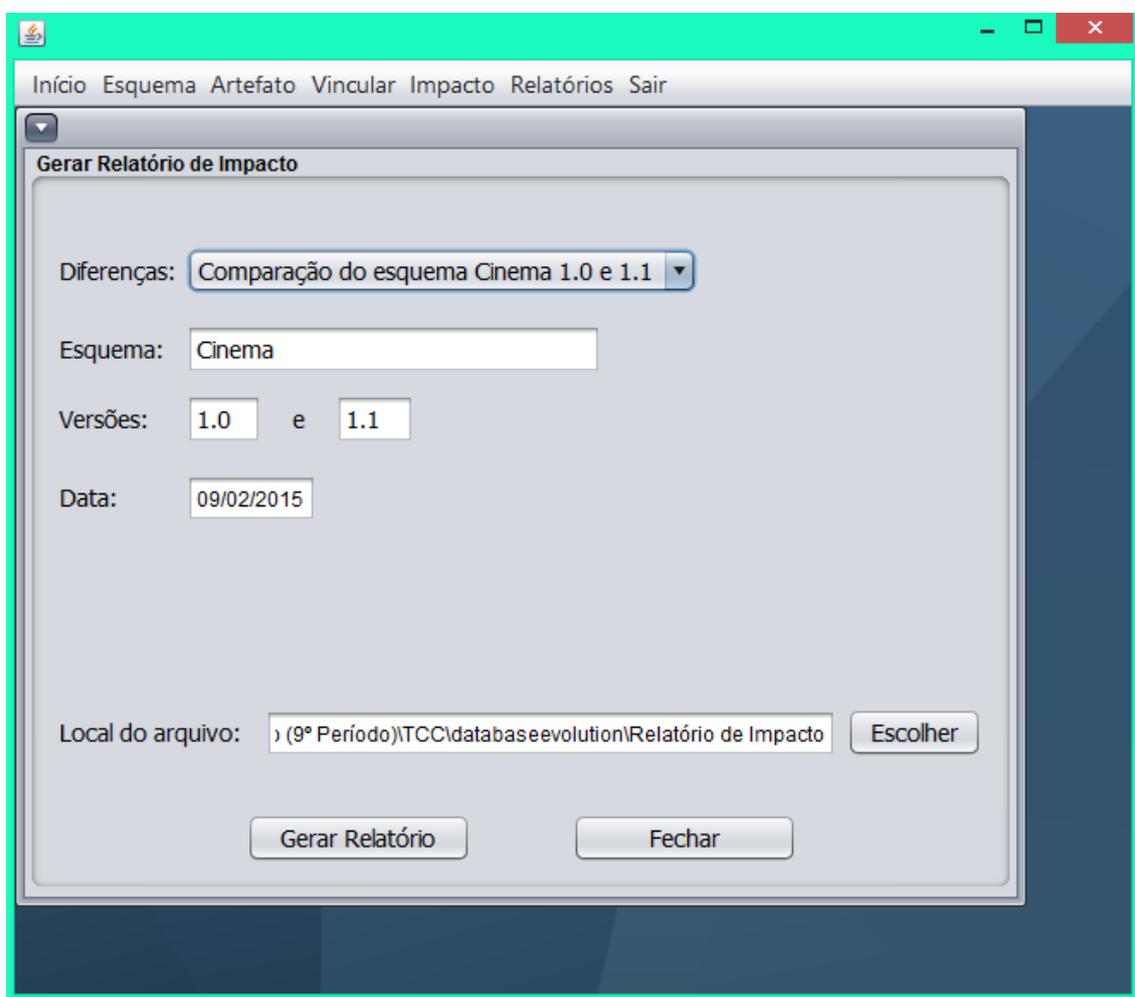


Figura 37: Tela para gerar relatório de impacto da ferrmaneta

Relatório de Avaliação de Impacto

Esquema: Cinema

Versão Antes: 1.0

Versão Depois: 1.1

Id do Elemento	Elemento	Tipo de Elemento	Operação	Tabela Associada	Campo Alterado	Id do Artefato	Artefato Afetado	Módulo	Sistema
2536	TE_CATEGORIA_FILME_has_TE_FILME	TABELA	ATUALIZADO	-	NOME	6	Venda de Ingresso	Vendas	CinemaX
2563	LOGRADOURO	COLUNA	ATUALIZADO	TB_FILIAL	NOME	6	Venda de Ingresso	Vendas	CinemaX

Figura 38: Relatório de Impacto gerado pela ferramenta

5. CONCLUSÕES

Analisar e identificar os impactos de evolução de esquema em banco de dados é um grande desafio enfrentado pelas grandes empresas no processo de evolução de esquemas. Dentre os vários fatores que contribuem para a má avaliação dos efeitos de mudanças nos esquemas de dados, está a falta de ferramentas automáticas para prever e avaliar os efeitos das alterações no esquema proposto. Nesse trabalho, propomos uma ferramenta de avaliação de impacto automática que pode substituir a análise manual e contribuir para melhoria da qualidade da evolução de esquema e suas propagações em todo o sistema.

Dentre os principais desafios para evolução de esquemas está a atualização dos sistemas. Quando um esquema é alterado, é preciso identificar os artefatos dos sistemas que podem ser afetados pelas modificações e adaptá-las, de forma a manter a consistência dos dados. Consultas e dados de formulários podem ser invalidados, programas de aplicação podem falhar devido a inconsistência entre o esquema de dados, o banco de dados e aplicação. Portanto, detectar manualmente os impactos de uma evolução de esquema é uma tarefa complexa que consome muito tempo e pode causar efeitos negativos sobre as aplicações. Através de revisão da literatura, pesquisa de usuários e muitas discussões, chegamos a alguns requisitos necessários e satisfatórios para a realização da avaliação de impacto pela ferramenta.

Outros trabalhos moderadamente relacionados também buscam soluções para a análise e detecção de impactos de mudanças em esquemas. Em (CURINO et al. 2009) é proposto um projeto para automatizar o processo de evolução de esquema de dados, com uso de linguagem própria para especificação das alterações de esquema. Em (KARAHASANOVIC; SJØBERG, 2001) é desenvolvida uma ferramenta gráfica para visualização dos impactos de mudanças no esquema de dados. Em (MAULE et al. 2008) foi desenvolvida uma abordagem para prever o impacto de mudanças de esquema de banco de dados, na qual havia redução do programa, análise do fluxo de dados e cálculo de impacto. Em (GARDIKIOTIS; MALEYRIS, 2006) foi desenvolvida uma ferramenta para estimar o impacto de modificações de esquema de banco em uma aplicação web, cujo objetivo era encontrar o número de páginas web afetadas.

O presente trabalho difere dos trabalhos anteriores na medida em que apresenta uma abordagem sistemática para a avaliação de impacto em função da evolução de esquemas. Ao invés de utilizar algoritmos para tentar identificar nos códigos das aplicações possíveis impactos em função da evolução de esquemas, a ferramenta adota uma abordagem baseada em cadastro de metadados. Todo e qualquer artefato ao ser produzido, pode ser vinculado na ferramenta a elementos de um esquema de dados, sejam eles um esquema, uma tabela, um atributo, um índice ou uma restrição. Uma vez vinculados, comparações entre *baselines* de esquemas, podem identificar as mudanças que ocorreram e selecionar os artefatos diretamente ou indiretamente afetados pelas mudanças.

Logo, se durante o processo de desenvolvimento e manutenção de sistemas for adotada a prática de vinculação de elementos de dados a artefatos, toda e qualquer mudança nos esquemas de dados poderão ser acompanhadas por uma avaliação de impacto, dada pela ferramenta, com alto grau de confiança.

O presente trabalho pode servir de base para trabalhos futuros. A conexão com outros SGBDs é uma extensão importante que pode confirmar os benefícios da ferramenta para uma maior cobertura de esquema de dados. Outra extensão possível seria a criação de novos módulos para monitorar as mudanças efetuadas diretamente nos SGBDs.

REFERÊNCIAS

BEYER, D.; NOACK, A.; LEWERENTZ C. **Efficient Relational Calculation for Software Analysis**. IEEE TSE, 31(2):1376149, Feb. 2005.

CURINO, C.; MOON, H. J.; ZANIOLO, C. **Automating Database Schema Evolution in Information System Upgrades**. In: HotSWUp, pp. 1-5, 2009.

DB Change Manager. Disponível em: <<https://www.embarcadero.com/br/products/db-change-manager>>. Acesso em: 4 de dezembro de 2014.

Find it EZ Software. Disponível em: <<http://www.finditez.com/databasedevelopers.aspx>>. Acesso em: 4 de dezembro de 2014.

GARDIKIOTIS, S. K.; MALEYRIS, N. **DaSIAn: A Tool for Estimating the Impact of Database Schema Modifications on Web Applications**. In Proceedings of AICCSA, pp.188-195, IEEE International Conference on Computer Systems and Applications, 2006.

KARAHASANOVIC, A.; SJØBERG, D.I.K. **Visualizing Impacts of Database Schema Changes -- A Controlled Experiment**. Proceedings of the IEEE Symposium on Visual/Multimedia Approaches to Programming and Software Engineering, Human Centric Computing 2001 (pp. 358-365) Stresa, Italy, 2001

LAW, J.; ROTHERMEL, G. **Whole program Path-Based dynamic impact analysis**. In ICSE 03: Proceedings of the 25th International Conference on Software Engineering, pages 3086318. IEEE Computer Society, 2003.

LINDVAL, M.; SANDAHL, K. **How well do experienced software developers predict software change?** Journal of Systems and Software, 43:19627(9), October 1998.

MAULE, A; EMMERICH, W; ROSENBLUM, D. S. **Impact Analysis of Database Schema Changes**, in Proceedings of the 30th International Conference on Software Engineering, May. 2008, pp. 451-460.

MORAES, A. J. C. **AutonomousDB: uma ferramenta para propagação autônoma de atualizações de esquemas de dados.** 2009. 104 f. Dissertação (Mestrado em Ciência da Computação) ó Centro de Informática, Universidade Federal de Pernambuco, Recife.

Oracle Enterprise Manager. Disponível em: <<http://docs.oracle.com/en/enterprise-manager/>>. Acesso em: 4 de dezembro de 2014.

PAPASTEFANATOS, G.; VASSILIADIS, P.; ANAGNOSTOU, F.; VASSILIOU, Y. **Hecataeus: A What-If Analysis Tool for Database Schema Evolution.** In 9th International Conference on Data Warehousing and Knowledge Discovery (DaWaK '07), Regensburg, Germany, 3-7 September, 2007, LNCS 4654, pp. 23633, 2007.

SJOBERG, D. **Quantifying Schema Evolution.** Information and Software Technology, vol. 35, no. 1, pp. 35-44, 1993.

ZANIOLO, C.; CERI, S.; FALOUTSOS, C.; SNODGRASS, R. T.; SUBRAHMANIAN, V.; ZICARI, R. **Advanced Database Systems.** Morgan Kaufmann Publishers Inc., San Francisco, California, Chapter 17, pp. 437-438, 1997.