



UNIVERSIDADE DO SUL DE SANTA CATARINA
VINICIUS DA SILVA VIANA

INTEROPERABILIDADE DE DADOS ENTRE SISTEMAS
GERENCIADORES DE BANCO DE DADOS RELACIONAIS

Tubarão
2020

VINICIUS DA SILVA VIANA

**INTEROPERABILIDADE DE DADOS ENTRE SISTEMAS
GERENCIADORES DE BANCO DE DADOS RELACIONAIS**

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação, da Universidade do Sul de Santa Catarina, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Luciano Sávio

**Tubarão
2020**

VINICIUS DA SILVA VIANA

**INTEROPERABILIDADE DE DADOS ENTRE SISTEMAS
GERENCIADORES DE BANCO DE DADOS RELACIONAIS**

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo Curso de Ciência da Computação, da Universidade do Sul de Santa Catarina.

_____, _____ de _____ de 20 _____.
Local Dia Mês Ano

Prof. e Orientador Luciano Sávio

Prof. Clávison Zapelini

Prof. Silvana Dal-Bó

RESUMO

A grande dificuldade para se ter disponibilidade em banco de dados, é a de conseguir proteger os dados de qualquer imprevisto que possa acontecer. Descobrir novas maneiras para manter esses dados disponíveis pode garantir uma maior proteção do banco. Para atender esta necessidade, foi elaborado um programa de conversação entre sistemas gerenciadores de banco de dados relacionais, para que eles possam não somente comunicar-se entre si, mas compartilhar e manipular os dados de maneira prática e acessível com outras bases de dados.

Palavras-chave: Banco de dados, Segurança de dados, Comunicação de dados, Manipulação de dados.

ABSTRACT

The great difficulty to have availability in a database, is to be able to protect the data from any unforeseen that may happen. Finding new ways to keep this data available can ensure greater bank protection. To meet this need, a conversation program was developed between relational database management systems, so that they can not only communicate with each other, but share and manipulate the data in a practical and accessible way with other databases.

Keywords: Database, Data Security, Data Communication, Data Manipulation.

LISTA DE ILUSTRAÇÕES

Figura 1 - Figura ilustrativa da seleção de abas.....	14
Figura 2 - Figura ilustrativa da aba Conexões.....	15
Figura 3 - Figura ilustrativa do menu Dados na aba Conexões.....	16
Figura 4 - Figura ilustrativa do menu Configuração na aba Conexões.	17
Figura 5 - Figura ilustrativa da aba Procedimentos.....	18
Figura 6 - Figura ilustrativa do menu Conexão 1 na aba Procedimentos.....	19
Figura 7 - Figura ilustrativa do menu Comandos na aba Procedimentos.....	20
Figura 8 - Casos de Uso.....	21
Figura 9 - Diagrama de Classes.....	25
Figura 10 - Figura ilustrativa exibindo os dados de uma conexão em rede.	29

LISTA DE TABELAS

Tabela 1 - Requisitos Funcionais.....	12
Tabela 2 - Requisitos Não Funcionais	13
Tabela 3 - Caso de Uso: Cadastrar Conexão.	21
Tabela 4 - Caso de Uso: Testar Conexão.	21
Tabela 5 - Caso de Uso: Alterar Conexão.	22
Tabela 6 - Caso de Uso: Remover Conexão.....	22
Tabela 7 - Caso de Uso: Selecionar Conexão.	22
Tabela 8 - Caso de Uso: Exibir Conexão.	23
Tabela 9 - Caso de Uso: Transferir/Receber Conexão.	23
Tabela 10 - Caso de Uso: Remover Conexão.....	23
Tabela 11 - Caso de Uso: Exibir Log Conexão.....	24
Tabela 12 - Caso de Uso: Visualizar Dados.....	24
Tabela 13 - Caso de Uso: Selecionar Tabelas.	24

SUMÁRIO

1	INTRODUÇÃO.....	9
2	OBJETIVOS	10
2.1	GERAL	10
2.2	ESPECÍFICO	10
3	JUSTIFICATIVA	11
4	DESENVOLVIMENTO.....	12
4.1	LEVANTAMENTO DE REQUISITOS	12
4.1.1	Requisitos Funcionais	12
4.1.2	Requisitos Não Funcionais	13
4.2	FUNCIONALIDADES DO SISTEMA	14
4.2.1	Conexões.....	14
4.2.2	Procedimentos.....	17
4.3	DIAGRAMA DE CASOS DE USO	21
4.4	DIAGRAMA DE CLASSES	25
4.5	FERRAMENTAS	26
4.5.1	Java.....	26
4.5.2	Eclipse.....	26
4.5.3	e(fx)clipse.....	27
4.5.4	Sistema Gerenciadores de Bancos de Dados Relacionais	27
4.5.5	Scene Builder	27
4.5.6	Launch4j	27
5	RESULTADOS E DISCUSSÕES.....	28
6	CONCLUSÃO.....	30
7	TRABALHOS FUTUROS	31
7.1	SALVAR E CARREGAR CONEXÕES	31
7.2	REVERTER COMANDOS REALIZADOS	31
8	REFERÊNCIAS.....	32

1 INTRODUÇÃO

O projeto em questão tem como tema principal a manipulação exclusiva dos dados entre os sistemas gerenciadores de banco de dados relacionais, possibilitando então a transferência de dados ou até mesmo a remoção dele.

A mais de uma década os bancos de dados representam cada vez mais importância no ambiente empresarial, visto que essa é a melhor maneira de se manter dados salvos para serem usados futuramente de maneira prática e eficiente assim como é citado por Cavassini Matioli (2010):

[...] Atualmente, com a tecnologia, surgem necessidades para as empresas. Com o aumento da demanda, dos funcionários, clientes e da empresa, muitos dados são manipulados a todo o momento. Inclusão de um novo cliente, alteração, exclusão e outros processos a todo instante, passa a ser um problema para a empresa, pois não pode perder os dados e como DBA, facilitar a rotina é um objetivo. Quando um cliente, por exemplo, faz um pedido de 100 latas de cerveja, rapidamente temos que consultar se temos 100 latas no estoque, se o cliente tem cadastro na empresa, se o endereço dele no cadastro está correto para a entrega e muitas outras coisas. Por isso um banco de dados. Com os dados no banco de dados, as consultas são rápidas e caso haja a necessidade de alteração ou exclusão, isso será rápido. [...]
(CAVASSINI MATIOLI, Danilo. 2010)

Porém esses dados podem ser perdidos com qualquer problema que venha a ocorrer com a base de dados, sendo então de suma importância a necessidade de se manter uma base segura desses dados a salvo em caso de falhas ou contingências.

Tendo em mente essa necessidade, foi elaborado um programa para a manipulação destes dados, visando sempre em manter a segurança dos dados a salvo de qualquer imprevisto, e que possam ser utilizados o mais rápido possível, levando em conta que ferramentas de banco de dados ao apresentarem falhas principalmente na questão dos dados, perde-se grande parte da confiança do usuário final sobre o negócio, tendo então uma grande relevância em manter o seu controle garantido assim como é citado por Crepaldi Martins e Candido Junior (2014):

[...] a confiabilidade está ligada à proteção dos dados, algo importantíssimo, já que a exposição de certas informações pode ter como consequências o constrangimento dos envolvidos, a perda da confiança ou numa ação contra a instituição. [...]
(CREPALDI MARTINS, Fábio; CANDIDO JUNIOR, Eli. 2014)

2 OBJETIVOS

2.1 GERAL

Desenvolver um software da qual faça a transferência e limpeza de dados em sistemas gerenciadores de bancos de dados relacionais, objetivando aumentar a facilidade no controle de dados.

2.2 ESPECÍFICO

- Permitir que se obtenha melhoria na qualidade dos dados transferidos.
- Permitir a transferência de dados em sistemas gerenciadores de banco de dados.
- Permitir que produtos diferentes de sistemas gerenciadores de banco de dados comuniquem-se entre si.

3 JUSTIFICATIVA

Enviar dados entre sistemas gerenciadores de bancos de dados relacionais pode ser um processo muito lento, segundo Fernanda Malacrida, Pereira Zaupa, Augusto Pazoti (2014).

[...] Ao utilizar uma ferramenta que realiza a migração de dados entre bases, há uma redução significativa da necessidade de digitação das informações, pois a tarefa de digitar dados manualmente torna-se inviável dependendo do volume de dados armazenados. Primeiramente pelo tempo gasto, mas principalmente pelo comprometimento da integridade dos dados armazenados. [...]

Considerando o comprometimento necessário para realizar a migração de dados, é muito importante que seja utilizado uma ferramenta que torne esta tarefa mais prática e fácil. Deve-se então levar em conta o uso de um software especializado para realizar essa tarefa.

De acordo com (Bevilacqua Mello, 2009) As empresas realizam a migração de dados para novos sistemas gerenciadores de banco de dados quando se deparam com problemas relacionados a fatores de custos, desempenho e usabilidades. Isso faz com que seja necessário ferramentas adequadas para realizar a migração de dados.

Tendo em mente os assuntos descritos, acredita-se que o software após implementado, possa facilitar e melhorar a manipulação de informações entre os bancos de dados, além de garantir um meio de proteção ao ambiente que utiliza o banco de dados, por permitir que outro sistema de banco de dados relacional possa assumir o controle em caso de falhas, reduzindo então os custos operacionais da empresa e garantindo maior proteção aos dados armazenados.

4 DESENVOLVIMENTO

4.1 LEVANTAMENTO DE REQUISITOS

Para inicializar qualquer projeto de software, é muito importante que se realize o levantamento de requisitos necessários, segundo Chichinelli (2017) “Desenvolver um sistema, sem possuir um entendimento completo e correto dos requisitos pode trazer problemas, pois, certamente não atenderá a todas as necessidades dos usuários.” Tendo em mente essa citação, foi necessário elaborar o levantamento de requisitos para obter um maior desempenho do software desenvolvido.

4.1.1 Requisitos Funcionais

Tabela 1 - Requisitos Funcionais

RF01. Trocar Janelas	Possibilitar o usuário trocar entre as janelas Conexões, Procedimentos.
RF01. Conexões: Cadastro	Permitir que o usuário possa realizar o cadastro de uma conexão no software, informando um Nome, URL, Usuário e Senha.
RF02. Conexões: Alterar	Alterar os dados cadastrados em uma conexão.
RF03. Conexões: Remover	Remover uma conexão cadastrada.
RF04. Conexões: Testar	Consultar uma conexão para verificar se ela está sendo corretamente conectada ao software.
RF05. Conexões: Mensagem	Descrever qualquer ação tomada na janela de Conexões, exibindo se cada comando foi realizado com sucesso ou não, descrevendo as falhas ocorridas, caso tenha.
RF06. Procedimentos: Conexões	Permitir ao usuário selecionar uma das conexões cadastradas no programa. Mostrar 2 campos de conexões onde serão usados para realizar outros procedimentos.

RF06. Procedimentos: Exibir	Conectar-se ao banco de dados da conexão que for selecionado pelo o usuário e extrair informações.
RF07. Procedimentos: Selecionar Tabelas	Após conectado a uma conexão, deve exibir as seguintes informações neste campo: o nome do banco de dado, o nome das suas tabelas. Deve-se permitir a seleção das tabelas para a realização de outros procedimentos.
RF08. Procedimentos: Visualizar Dados	Após conectado a uma conexão, deve exibir as seguintes informações neste campo: O nome do banco de dado, o nome das suas tabelas, o nome das colunas da tabela, e as informações contidas naquela coluna.
RF09. Procedimentos: Transferir/Receber Dados	Após definição da primeira e da segunda conexão, permitirá transferir ou receber os dados que estiverem nas respectivas conexões. Apenas deve se transferir/receber os dados das tabelas que foram selecionadas. Deve-se também permitir atribuição de condições para esse procedimento.
RF10. Procedimentos: Remover Dados	Permitir que todos os dados de uma conexão escolhida pelo usuário, possam ser removidos. Apenas deve se remover os dados das tabelas que foram selecionadas. Deve-se também permitir atribuição de condições para esse procedimento.
RF11. Procedimentos: Log	Mostrar um registro de todos os comandos query utilizados no software.
RF12. Procedimentos: Mensagem	Descrever qualquer ação tomada na janela de Mensagem, exibindo se cada comando foi realizado com sucesso ou não, descrevendo as falhas ocorridas caso tenha.

4.1.2 Requisitos Não Funcionais

Tabela 2 - Requisitos Não Funcionais

RNF01. Restrição de Cadastro e Alteração	A usuário só poderá confirmar o Cadastro/Alteração com uma conexão, caso os campos além da senha não estejam vazios.
RNF02. Restrição de Alteração/Remoção/Testar	Só será permitido efetuar a Alteração/Remoção/Testar caso tenha uma conexão selecionada pelo usuário.

RNF03. Restrição de Exibição/Limpar	Para poder realizar esse procedimento, é necessário que a conexão tenha sido escolhida pelo usuário.
RNF04. Restrição para Transferência/Receber	Para poder realizar esse procedimento, é necessário que duas conexões tenham sido escolhidas pelo usuário.
RNF05. Restrição para a Conexão	Para poder realizar procedimentos de Exibição/Limpeza/Transferência/Receber é necessário que seja possível acessar a conexão.

4.2 FUNCIONALIDADES DO SISTEMA

As funcionalidades do programa são separadas por abas que podem ser abertas pelo usuário, elas são classificadas em duas, das quais podem ser vistas na Figura 1 - Figura ilustrativa da seleção de abas.

Figura 1 - Figura ilustrativa da seleção de abas.



Fonte: Elaboração do autor (2020).

4.2.1 Conexões

Para a realização da manipulação dos dados nos sistemas de banco de dados relacionais, é necessário que este esteja conectado ao programa, a fim de realizar essa comunicação, o programa tem um local exclusivo para lidar com esta conexão, da qual é possível visualizar na Figura 2 - Figura ilustrativa da aba Conexões.

Figura 2 - Figura ilustrativa da aba Conexões.

Fonte: Elaboração do autor (2020).

4.2.1.1 Dados

Contem 4 caixas de texto a serem preenchidas pelo usuário, das quais são: **Nome** da qual requer que o usuário atribua um nome fictício ao banco em questão com o intuito de evitar problemas com nomes repetidos, a **URL** que refere-se ao local onde o projeto irá estabelecer a conexão com o banco de dados, o **Usuário** e **Senha**, da qual são responsáveis por garantir o acesso ao banco com os direitos de realizar a manipulação dos dados, e a fim de concluir a adição dos dados, possui um botão **Adicionar**, que ao clicar ele mostrará erros de digitação caso existam, abaixo das caixas de texto com erros, e caso não tenha nenhum problema, ele adicionará a conexão ao programa. Essas funcionalidades podem ser visualizadas na Figura 3 - Figura ilustrativa do menu Dados

Figura 3 - Figura ilustrativa do menu Dados na aba Conexões.



O formulário, intitulado "Dados", possui um fundo cinza com uma textura de losangos. Ele contém quatro campos de entrada e um botão. Os campos são: "Nome:" com o valor "Post1 TESTE"; "URL:" com o valor "jdbc:postgresql://localhost:5432/TrabalhoTC"; "Usuário:" com o valor "postgres"; e "Senha:" com caracteres ocultos por pontos. O botão "Adicionar" está localizado na base do formulário.

Dados	
Nome:	Post1 TESTE
URL:	jdbc:postgresql://localhost:5432/TrabalhoTC
Usuário:	postgres
Senha:
<input type="button" value="Adicionar"/>	

Fonte: Elaboração do autor (2020).

4.2.1.2 Configuração

Neste local, 4 funcionalidades estão inclusas, das quais são: **Conexão** onde será o local que irá ser selecionado pelo usuário qual conexão será configurada, **Testar** da qual se refere a verificar se os dados preenchidos estão corretos e conectando ao sistema de banco de dados, **Alterar** da qual irá possibilitar realizar edições nos dados da conexão selecionada, onde irá alterar o local reservado para adição de dados para que se torne em um local para alteração de dados, e este local alterado pode ser cancelado caso seja clicado novamente no botão alterar, e por fim o botão **Remover** que será usado para remover a conexão selecionada do programa. Essas funcionalidades podem ser visualizadas na Figura 4 - Figura ilustrativa do menu Configuração na aba Conexões..

Figura 4 - Figura ilustrativa do menu Configuração na aba Conexões.

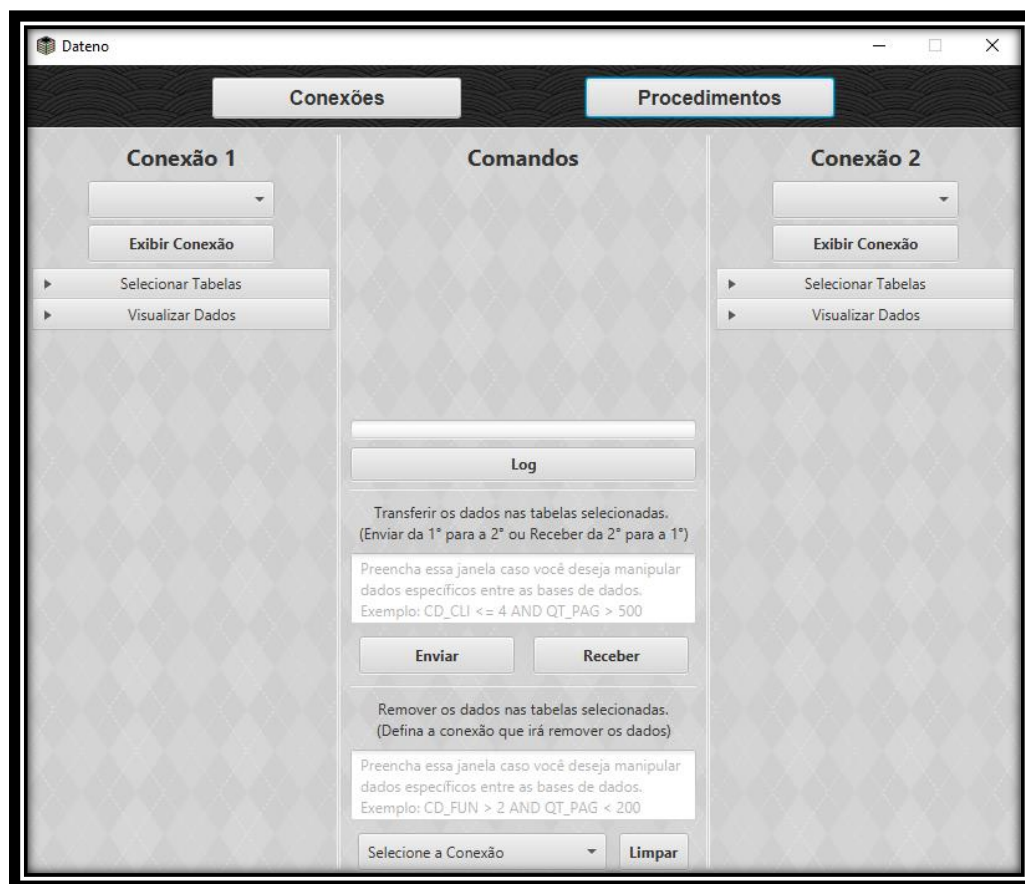


Fonte: Elaboração do autor (2020).

4.2.2 Procedimentos

Este é o local onde irá realizar a manipulação dos dados, da qual necessitará das conexões atribuídas na aba Conexões para que possa realizar os comandos, esse local é também visualmente mostrado na Figura 5 - .

Figura 5 - Figura ilustrativa da aba Procedimentos.

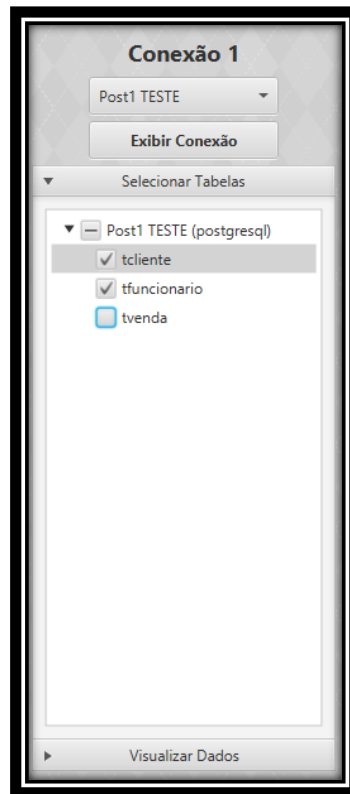


Fonte: Elaboração do autor (2020).

4.2.2.1 Conexão

Conterá 2 locais quase iguais com diferencial na numeração e posição que servirá para **Selecionar** a conexão que será utilizada para ser manipulada no passo seguinte, este também pode realizar o comando **Exibir Conexão**, para que seja possível **Selecionar Tabelas** e **Visualizar Dados** da conexão na lista abaixo deles. Essas funcionalidades podem ser visualizadas na Figura 3 - Figura ilustrativa do menu Dados

Figura 6 - Figura ilustrativa do menu Conexão 1 na aba Procedimentos.



Fonte: Elaboração do autor (2020).

4.2.2.2 Comandos

Esta é a parte principal do projeto onde irá realizar a manipulação dos dados, neste local possui 3 funcionalidades: **Enviar**, da qual irá pegar os dados das tabelas selecionadas pelo usuário que estão na conexão 1 para serem enviados para a conexão 2 além de também enviar condições adicionadas pelo usuário na caixa de texto abaixo, para possibilitar o envio de dados específicos, este processo só será possível caso o usuário tenha atribuído a conexão com direito a esses comandos, **Receber**, da qual fará o mesmo processo da funcionalidade transferir, mas que será feito da conexão 2 para a conexão 1, **Limpar**, onde dará a opção ao usuário de limpar todos os dados que estão armazenados dentro da conexão selecionadas por ele, além de também possuir uma caixa de texto abaixo para possibilitar a adição de condições para o envio de dados específicos, e por fim o comando **Log**, da qual irá exibir uma lista com todos os procedimentos Query utilizados no software . Essas funcionalidades podem ser visualizadas na Figura 7 - Figura ilustrativa do menu Comandos na aba Procedimentos..

Figura 7 - Figura ilustrativa do menu Comandos na aba Procedimentos.

Comandos

O comando de limpeza foi realizado com sucesso na conexão 'Post1 TESTE'

Log

Transferir os dados nas tabelas selecionadas.
(Enviar da 1ª para a 2ª ou Receber da 2ª para a 1ª)

Preencha essa janela caso você deseja manipular dados específicos entre as bases de dados.
Exemplo: CD_CLI <= 4 AND QT_PAG > 500

Enviar **Receber**

Remover os dados nas tabelas selecionadas.
(Defina a conexão que irá remover os dados)

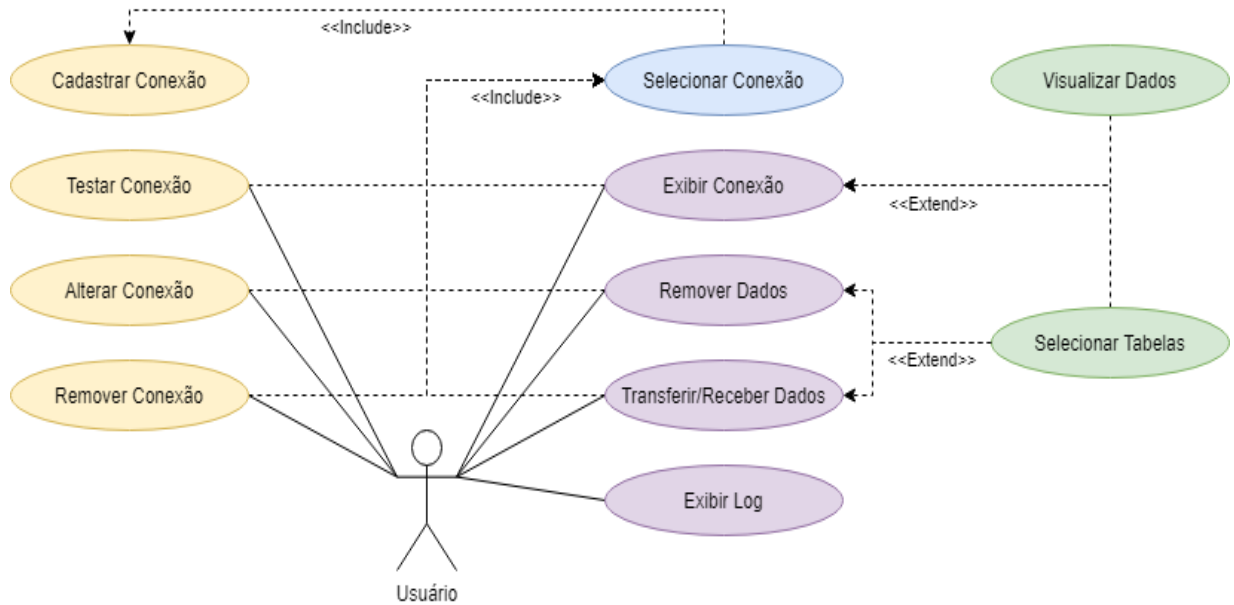
Preencha essa janela caso você deseja manipular dados específicos entre as bases de dados.
Exemplo: CD_FUN > 2 AND QT_PAG < 200

Conexao 1 **Limpar**

Fonte: Elaboração do autor (2020).

4.3 DIAGRAMA DE CASOS DE USO

Figura 8 - Casos de Uso.



Fonte: Elaboração do autor (2020).

Tabela 3 - Caso de Uso: Cadastrar Conexão.

Cadastrar Conexão	
Descrição	O ator poderá realizar o cadastro de conexões no programa
Atores	Usuário
Pré-Condições	Nenhum
Cenário Principal	O ator irá acessar a janela de “Conexões” e preencherá todos os dados necessários para cadastrar a conexão

Tabela 4 - Caso de Uso: Testar Conexão.

Testar Conexão	
Descrição	O ator poderá realizar um teste na conexão selecionada
Atores	Usuário

Pré-Condições	1. É necessário selecionar uma conexão
Cenário Principal	O ator pressiona o botão “Testar” para verificar se a conexão selecionada por ele está funcionando corretamente

Tabela 5 - Caso de Uso: Alterar Conexão.

Alterar Conexão	
Descrição	O ator poderá alterar uma conexão cadastrada
Atores	Usuário
Pré-Condições	1. É necessário selecionar uma conexão
Cenário Principal	O ator pressiona o botão “Alterar” para atualizar os dados de uma conexão selecionada por ele.

Tabela 6 - Caso de Uso: Remover Conexão.

Remover Conexão	
Descrição	O ator poderá remover uma conexão cadastrada
Atores	Usuário
Pré-Condições	1. É necessário selecionar uma conexão
Cenário Principal	O ator pressiona o botão “Remover” para remover a conexão selecionada por ele.

Tabela 7 - Caso de Uso: Selecionar Conexão.

Selecionar Conexão	
Descrição	O ator poderá escolher entre as conexões cadastradas no programa.

Atores	Usuário
Pré-Condições	1. É necessário ter pelo menos uma conexão cadastrada
Cenário Principal	O ator abre o menu de seleção de conexões e escolhe qual será utilizado entre todas as que estão cadastradas.

Tabela 8 - Caso de Uso: Exibir Conexão.

Exibir Conexão	
Descrição	O ator poderá exibir dados que estão em uma conexão.
Atores	Usuário
Pré-Condições	1. É necessário ter pelo menos uma conexão cadastrada 2. É necessário escolher uma conexão
Cenário Principal	O ator pressiona o botão “Exibir Conexão” da qual irá possibilitar mostrar os dados contidos naquela conexão.

Tabela 9 - Caso de Uso: Transferir/Receber Conexão.

Transferir/Receber Dados	
Descrição	O ator poderá transferir ou receber os dados de uma conexão para outra conexão
Atores	Usuário
Pré-Condições	1. É necessário ter pelo menos uma conexão cadastrada 2. É necessário escolher duas conexões
Cenário Principal	O ator pressiona o botão “Transferir” ou “Receber” da qual irá possibilitar Transferir ou Receber os dados contidos em uma conexão para outra.

Tabela 10 - Caso de Uso: Remover Conexão.

Remover Dados	
Descrição	O ator poderá realizar a remoção dos dados contidos em uma conexão.

Atores	Usuário
Pré-Condições	1. É necessário ter pelo menos uma conexão cadastrada 2. É necessário escolher uma conexão
Cenário Principal	O ator pressiona o botão “Remover” da qual irá possibilitar Limpar os dados contidos em uma conexão.

Tabela 11 - Caso de Uso: Exibir Log Conexão.

Exibir Log	
Descrição	O ator poderá visualizar um log de todos os comandos query realizados no programa.
Atores	Usuário
Pré-Condições	Nenhum
Cenário Principal	O ator pressiona o botão “Log” e aparecerá uma nova janela mostrando todos os comandos query realizados no programa

Tabela 12 - Caso de Uso: Visualizar Dados.

Visualizar Dados	
Descrição	O ator poderá realizar o cadastramento de conexões no programa
Atores	Usuário
Pré-Condições	1. É necessário ter pelo menos uma conexão cadastrada 2. É necessário escolher uma conexão 3. É necessário Exibir Conexão
Cenário Principal	O ator irá visualizar todos os dados de uma conexão cadastrada, da qual será mostrado em formato de lista.

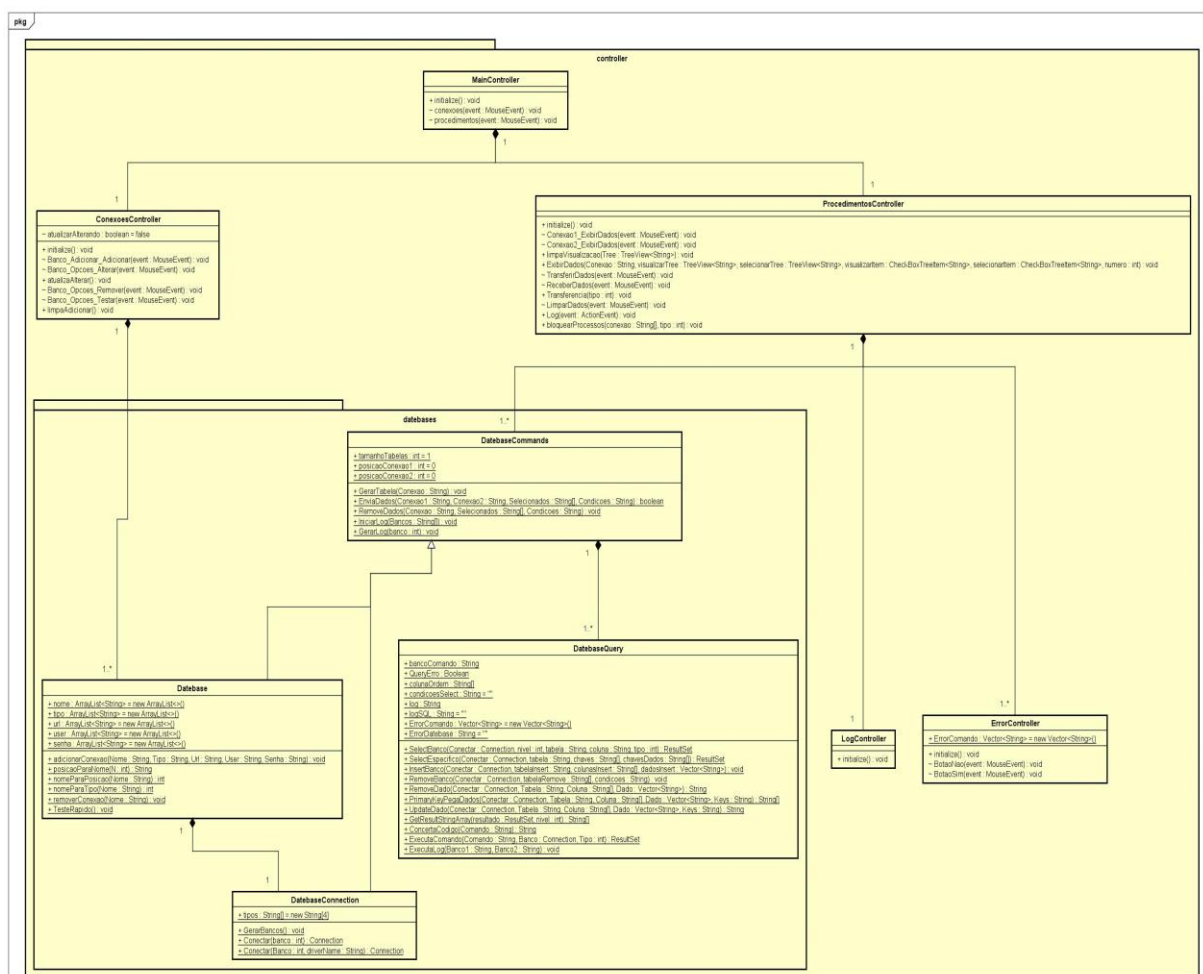
Tabela 13 - Caso de Uso: Selecionar Tabelas.

Selecionar Tabelas

Descrição	O ator poderá selecionar as tabelas que vão ser utilizadas
Atores	Usuário
Pré-Condições	<ol style="list-style-type: none"> 1. É necessário ter pelo menos uma conexão cadastrada 2. É necessário escolher uma conexão 3. É necessário Exibir Conexão
Cenário Principal	O ator irá selecionar as tabelas que ele desejar de uma conexão para utilização em outros procedimentos.

4.4 DIAGRAMA DE CLASSES

Figura 9 - Diagrama de Classes.



Fonte: Elaboração do autor (2020).

4.5 FERRAMENTAS

Com o intuito de gerar bons resultados, este projeto foi desenvolvido com uma série de ferramentas especializadas da qual se encaixam melhor na necessidade do projeto, mas em nenhum momento foram utilizadas ferramentas proprietárias, que necessitassem de algum tipo de licença paga.

4.5.1 Java

Neste projeto foi adotada a linguagem de programação Java, essa escolha se deve a vários fatores relacionados às funcionalidades pretendidas com o software, tendo em vista as muitas ferramentas e mecanismos que possibilitam a comunicação, manuseio e interpretação dos dados que serão necessários para o funcionamento desejado da ferramenta, além de ser uma linguagem multi-threading da qual possibilita o manuseio de vários processos ao mesmo tempo sendo quase indispensável em um processo de verificação de dados simultâneos. Tudo isso faz com que a linguagem Java seja a melhor escolha quando trabalhado com um software que se deseja realizar trabalhos simultâneos com vários dados, além de manter uma velocidade e interface agradável ao seu usuário que irá necessitar devido a área em questão trabalhada.

4.5.2 Eclipse

Por motivos de abrangência e necessidade de uma IDE (Ambiente de Desenvolvimento Integrado) gratuita para realizar a programação dos requisitos do projeto, a ferramenta Eclipse foi utilizada no processo de criação do programa, sendo que com ela pode-se conseguir facilmente obter-se drivers e mecanismos necessários para a integração dos banco de dados, além de também ser uma ferramenta vastamente utilizada pela comunidade da área com vários meios para se conseguir informações que auxiliem no processo de criação do programa que foi desenvolvido.

4.5.3 e(fx)clipse

Para poder manipular o programa Scene Builder adequadamente dentro da IDE Eclipse, fez-se necessário a instalação do plugin E(fx)clipse para o seu funcionamento, sem isso o programa não permitiria o funcionamento do aplicativo, e exibiria diversos erros pelo código.

4.5.4 Sistema Gerenciadores de Bancos de Dados Relacionais

Como o tema do projeto é a conversação entre sistemas gerenciadores de bancos de dados relacionais, foram utilizados sistemas mais atuais para o desenvolvimento do software, e estes são: PostgreSQL, MySQL, SQL Server e Oracle SQL.

Essas ferramentas foram utilizadas para a primeira versão do programa devido a sua fácil maneabilidade e configurabilidade do sistema, esse tipo de comportamento são fundamentais para a conclusão adequada do projeto, visando então evitar erros fora do escopo.

4.5.5 Scene Builder

Com a necessidade de desenvolver um programa que possa realizar de forma simples e clara as necessidades visuais do usuário final, o programa Scene Builder foi escolhido para a criação de painéis visuais para o desenvolvimento do projeto, devido a sua ampla facilidade e manuseio simplificado do seu conteúdo visual, acaba por ser uma ferramenta realmente útil para o desenvolvimento deste projeto.

4.5.6 Launch4j

Objetivado a criar um executável que seja mostrado com uma boa qualidade ao seu usuário final, o programa Launch4j foi utilizado para realizar automaticamente os argumentos necessários ao lidar com o plugin e(fx)clipse, além disso ele oferece um pacote completo para testar e executar o código em diversas versões do JDK (Java Development Kit) e JRE (Java Runtime Environment) do Java, isso contribui para a portabilidade do programa, tornando ele acessível a diversas versões do Java, e também por fim, gerar um executável que pode ser configurado.

5 RESULTADOS E DISCUSSÕES

De acordo com o desenvolvimento apresentado e seguido para a construção do software, foi possível verificar que cada um dos sistemas gerenciadores de bancos de dados relacionais possuem maneiras diferentes para se trabalhar, e cada um destes tem suas especificações e necessidades tanto de implementação quanto de organização, ao utilizar o sistema gerenciador MySQL por exemplo, foi possível destacar que todos os dados que eram enviados para essa base automaticamente eram postos em ordem crescente com base em cada coluna da tabela, esse tipo de resultado poderia acarretar no desentendimento dos dados por parte do usuário, levando ele a pensar que estes não estavam corretos, só que este tipo de problema não é possível de ser consertado sem a inserção lógica igual aos dados para cada banco utilizado.

Algumas bases de dados podem apresentar falhas nos processos de comunicação sem apresentar nenhum alerta específico, e isso pode dificultar na busca de informações sobre o erro, o Oracle SQL por exemplo, em alguns casos pode simplesmente não permitir que nenhum comando seja realizado em uma conexão específica, necessitando que o programa seja fechado e aberto novamente.

Também é importante ressaltar que como é um aplicativo que trabalha com conexões diretas, pode ser então utilizados bancos de dados em rede ou apenas localmente, como é o caso exibido na Figura 10 - Figura ilustrativa exibindo os dados de uma conexão em rede., onde o software retorna uma resposta positiva ao testar sua conexão com uma base online.

Figura 10 - Figura ilustrativa exibindo os dados de uma conexão em rede.

The screenshot displays the Dateno application window, which is divided into two main sections: 'Dados' (Data) and 'Configuração' (Configuration). The 'Dados' section contains four input fields: 'Nome:' with the value 'PostOnline1 TESTE', 'URL:' with the value 'jdbc:postgresql://204.2.195.79:18881/Banco1', 'Usuário:' with the value 'admin', and 'Senha:' with a masked password '.....'. Below these fields is an 'Adicionar' button. The 'Configuração' section features a 'Conexão:' dropdown menu currently set to 'PostOnline1 TESTE'. Below the dropdown are three buttons: 'Testar' (highlighted with a blue border), 'Alterar', and 'Remover'. At the bottom of the 'Configuração' section, the text 'Nenhum erro encontrado' is displayed. The application window has a title bar with the name 'Dateno' and standard window controls.

Fonte: Elaboração do autor (2020).

Deve-se lembrar também que a segurança relacionada aos dados de onde provém as conexões, não podem ser assegurados por este software, pois ele apenas utilizará as conexões pra manipular os dados, da qual necessita do URL, Usuário e Senha, o que no caso não se diz a respeito da estrutura do local onde estão armazenados os bancos de dados, não podendo ser levado em conta então a segurança deste, já que não faz parte do escopo do projeto.

Com base em todas as informações obtidas na construção e desenvolvimento do projeto, foi então possível chegar ao seu objetivo de manipular dados entre sistemas gerenciadores de bancos de dados, mas deve-se levar em consideração que cada sistema gerenciador de banco de dados possui uma maneira para armazenamento de dados como dito anteriormente, e o processo de comunicação poderá omitir falhas diferentes para cada conexão, sendo então difícil de encontrar os erros específicos para uma determinada conexão.

6 CONCLUSÃO

De forma geral, o projeto apresentou sucesso nos seus objetivos propostos, obteve-se uma maior facilidade de realizar a comunicação entre sistemas gerenciadores de bancos de dados relacionais, mas também surgiram alguns problemas que não poderiam ser facilmente resolvidos devido as funcionalidades de cada sistema.

No decorrer do desenvolvimento do projeto, alguns problemas surgiram no desenvolvimento da aplicação, como cada sistema gerenciador de banco de dados possui as suas próprias funcionalidades, necessitou-se criar comandos específicos para cada um deles, precisando-se então procurar pelos comandos query que realizem determinado comando corretamente.

O software desenvolvido possui a habilidade de trabalhar localmente ou remotamente. Ao atribuir uma conexão local, a aplicação irá responder rapidamente aos procedimentos que forem realizados, mas isso se difere das conexões remotas que necessitam de pelo menos alguns segundos até conseguirem executar os comandos regularmente, isso acontece devido a conexão à internet que necessita enviar os comandos até o usuário remoto e ainda precisar trazer todas essas informações de volta.

Uma outra qualidade do software, é que ele possibilita a seleção das tabelas e adição de condições para a inserção e remoção, já que todos os sistemas gerenciadores de bancos de dados relacionais utilizados possuem o mesmo esquema de condições para os dados manipulados, o que possibilitou impor esta funcionalidade sem nenhum problema específico.

Finalizando, acredita-se que a ferramenta pode ser muito útil para a transferência e limpeza de dados entre as diversas conexões, além de reduzir drasticamente o tempo necessário para a realização deste processo, ela proporciona uma facilidade para a manipulação de dados em diferentes sistemas gerenciadores de bancos de dados relacionais, não necessitando que o usuário entenda os comandos query específicos para cada um deles.

7 TRABALHOS FUTUROS

Com a finalização do projeto baseado no desenvolvimento proposto, pode-se então pensar na possibilidade de trazer novas atualizações, buscando uma maneira de melhor enfatizar a qualidade e praticidade do software. Dentre essas possibilidades de atualizações, pode-se citar a adição de dois novos métodos.

7.1 SALVAR E CARREGAR CONEXÕES

Quando o software for fechado ele não irá salvar as conexões cadastradas nele, necessitando que a cada vez que o usuário abrir o programa ele deverá cadastrar todas as conexões que irá utilizar, esse tipo de comportamento traz consigo uma redução da praticidade do programa em termos de agilidade, mesmo que não seja algo relativamente necessário para o funcionamento do projeto, ainda assim, se faz importante que em updates futuros possa existir a possibilidade de salvar e carregar conexões no software.

7.2 REVERTER COMANDOS REALIZADOS

Ao utilizar o software para realizar um procedimento, pode existir o caso do usuário realizar comandos indesejados, e caso esse tipo de ação aconteça, pode ser difícil retornar ao estado anterior dependendo das instruções impostas nestes comandos, mesmo que não seja uma necessidade para o funcionamento do projeto, ela ainda traz consigo uma melhor qualidade em termos de segurança no controle dos dados por parte do usuário, sendo então uma função de grande importância para adicionar ao software.

8 REFERÊNCIAS

BEVILACQUA MELLO, Rafael. **Migração entre sistemas gerenciadores de banco de dados**, 2009. Disponível em: <https://cepein.femanet.com.br/BDigital/arqTccs/0611270062.pdf>. Acesso em: 23 dezembro. 2020.

CAVASSINI MATIOLI, Danilo. **Importância da Segurança em Banco de Dados**, São Paulo, p. 15, 2010. Disponível em: <https://cepein.femanet.com.br/BDigital/arqTccs/0811060524.pdf>. Acesso em: 19 outubro. 2020.

CHICHINELLI, Micheli. **A importância das técnicas de levantamento de requisitos no processo de desenvolvimento de software**, 2017. Disponível em: <http://www.ojs.toledo.br/index.php/gestaoetecnologia/article/view/2653/220>. Acesso em: 23 dezembro. 2020.

CREPALDI MARTINS, Fábio; CANDIDO MARTINS, Eli. **Segurança em Banco de Dados: Conceitos e Aplicações**, 2014. Disponível em: <http://intertemas.toledoprudente.edu.br/index.php/ETIC/article/download/4412/4172>. Acesso em: 13 mai. 2020.

FERNANDA MALACRIDA, Tamires; PEREIRA ZAUPA, Aglaê; AUGUSTO PAZOTI, Mário. **Desenvolvimento de uma ferramenta para migração de dados entre bancos de dados relacionais**, 2013. Disponível em: <http://revistas.unoeste.br/index.php/ce/article/view/1015/1106>. Acesso em: 23 dezembro. 2020.