

FERRAMENTA DE BUSCA E DIAGNÓSTICO DE ERROS EM SISTEMAS DE INTEGRAÇÃO

Marcos Paulo Rosa, Marcel Hugo – Orientador

Curso de Bacharel em Ciência da Computação
Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brazil

marcospaulorosa@gmail.com, marcel@furb.br

Resumo: Este artigo tem o objetivo de relatar o desenvolvimento de uma ferramenta de busca e diagnóstico de erros de integração em um sistema proprietário, que envolve um sistema de ERP, sistema de retaguarda de loja, PDV e integradores. A ferramenta proposta e desenvolvida tem como objetivo a redução do tempo despendido pelo suporte na obtenção de dados para investigação de erros de integração que possam ocorrer no dia a dia pelos mais diversos motivos. Conforme os resultados coletados em testes, houve a efetiva redução do tempo despendido pela área de Suporte, aumentando a capacidade de atendimento de seus colaboradores. O uso experimental da ferramenta abriu margem para expansão de novas funcionalidades que poderão ser exploradas futuramente, ampliando o ganho de tempo na resolução de erros. Contudo, esta expansão encontrará dificuldades nos diferentes ambientes encontrados nos clientes, tais como arquiteturas de redes e configurações entre servidores que variam de cliente para cliente.

Palavras-chave: Suporte. Sistemas integrados. Erros de integração.

INTRODUÇÃO

A satisfação do cliente com o serviço de pós-venda de software é tão importante quanto a própria venda do produto, segundo Bentes (2012, p. 109). Isto inclui a área de suporte da empresa desenvolvedora deste software, no qual através de seu atendimento, confirma para o cliente a importância que ele tem para a empresa tanto na hora da venda quanto no momento em que necessita de suporte. Para que isto ocorra, é necessário que a equipe do setor de suporte esteja capacitada, possua processos bem definidos e saiba administrar corretamente o tempo despendido nos atendimentos de requisições de atendimento de suporte (BEISSE, 2014, p. 290).

Uma equipe de suporte capacitada, proativa e eficiente não deve ser subestimada como uma importante ferramenta de satisfação ao cliente. Os clientes mais leais não necessariamente são aqueles que nunca se decepcionaram com o atendimento da área de suporte, mas sim aqueles que estavam aguardando a resolução de um erro ou falha de serviço e então tiveram o problema resolvido de forma rápida, eficaz e cordialmente (HILES; GUNN, 2000, p. 11, tradução nossa).

Em Blumenau, há uma empresa desenvolvedora de software que possui uma solução de ERP integrada com sistema de varejo, composto por um sistema de retaguardas e pontos de vendas, chamados PDV. Cada camada da solução possui um banco de dados próprio e todas as camadas são interligadas por integradores que realizam as trocas de informações do ponto de venda até o ERP. Neste processo, podem haver falhas e ruídos de integração por diversos motivos, tais como queda de conexão gerando má formação de um pacote JSON ou XML, erros de parametrização e má operação do sistema.

Ao ocorrerem falhas de integração no produto, aumenta-se o acionamento à área de suporte do produto. Com mais ocorrências, pode gerar uma demanda onerosa em termos de tempo de diagnóstico e pesquisas em bases de dados para encontrar o ponto de falha e/ou dados relevantes para compor a análise do que pode ter gerado a falha, tendo em vista o grande número de bancos de dados que a solução necessita devido à alta escalabilidade da solução.

Segundo Cohen (2008, p. 131), a rapidez com que o suporte intervém e soluciona as requisições demandadas pelos clientes de software influenciam diretamente na redução do impacto causado pelo erro ou dúvida na operação do sistema instalado ou em implantação. Ainda segundo o autor, a área de suporte de uma empresa está constantemente em busca de novas ferramentas para dinamizar e aumentar a eficiência no atendimento ao cliente e resolução de problemas. Algumas destas ferramentas possibilitam ao cliente o autoatendimento, localizando soluções e orientações por conta própria sem que haja um contato com o serviço de suporte da empresa.

Diante dos fatos expostos, este trabalho propõe o desenvolvimento de uma ferramenta de apoio à área de suporte desta empresa, que atue no âmbito desta aplicação proprietária relatada anteriormente. Esta ferramenta busca facilitar e agilizar o diagnóstico de erros de integração bem como a obtenção de dados que facilitarão a análise e auxiliarão na tomada de decisões para a solução de erros, tendo como objetivos:

- a) reduzir o tempo de análise despendido pela equipe de suporte;
- b) evitar a quantidade de acessos remotos efetuados pela equipe de suporte;

- c) centralizar a análise dos dados de integração, evitando acessos redundantes;
- d) reduzir o tempo de diagnóstico do erro de origem em uma determinada integração.

FUNDAMENTAÇÃO TEÓRICA

A comunicação entre sistemas através de integrações tornou-se um processo irreversível na evolução de sistemas. Um exemplo desta evolução são os sistemas Enterprise Resource Planning (ERP), que tem como principal característica, além da integração dos sistemas internos, a ênfase na colaboração comercial utilizando a internet para conectar processos e/ou sistemas de duas ou mais empresas. (CAIÇARA JUNIOR, 2009, p. 88). Para Meireles (2001, p. 69), sistemas ERP são um recurso vital para a rotina de trabalho em empresas de diferentes portes e segmentos de mercado.

Conforme Prado e Souza (2014, p. 141), as empresas implantam sistemas ERP com vistas à integração dos processos de negócio. Tal sistema viabiliza a construção de uma infraestrutura padronizada eliminando a problemática de se interligar diferentes componentes por meio de variadas interfaces. Ainda segundo o autor, este tipo de aplicação distribuída e integrada propicia a visibilidade do negócio fornecendo informação integrada e consolidada em um único banco de dados, em tempo real e em diferentes áreas da organização.

Segundo Caiçara Júnior (2009, p. 83), muitos benefícios tangíveis e intangíveis são advindos da integração entre sistemas, onde destaca-se a redução de pessoal, aumento da produtividade, padronização de processos, entregas pontuais, entre outros. No entanto, erros de parametrização ou inserção de informações errôneas no sistema são disponibilizadas para outros usuários no momento em que são registradas. Uma vez inserida, esta informação será liberada e propagada para todas as pessoas em uma organização ou mesmo para pessoas fora desta organização, tendo em vista tratar-se de um sistema que pode estar integrado com outras empresas (GURARINO, 2015).

O monitoramento dos erros de integração permite uma rápida intervenção nos incidentes de integração do sistema. Segundo Bruton (2002, tradução nossa), quando o usuário solicita apoio do setor de suporte ele deseja a resposta imediatamente, pois caso o suporte não tenha esta resposta de imediato gera uma insatisfação para o cliente. Desta forma, ainda segundo o autor, tal situação em contrapartida gera um custo para o setor de suporte pois o atendente necessitará escalar a informação ou buscá-la de outros meios tornando o processo mais oneroso.

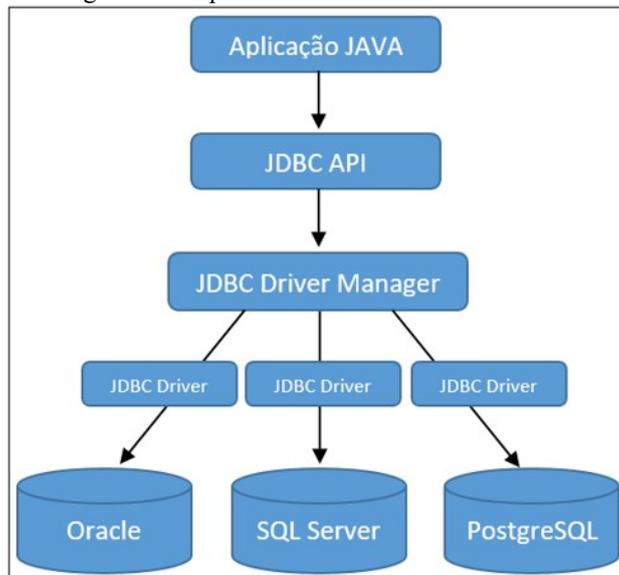
Para Frantz et al. (2014), erros ocorrem em função de falhas, de origem permanente devido a defeitos no software ou transientes devido a indisponibilidade de algum recurso para a solução. A utilização de técnicas baseadas em software para detectar erros evita que estes erros se propaguem pelo sistema e venham a aumentar o dano sistêmico da operação do sistema (PICCOLI, 2006). Desta forma, a detecção de erros de integração e a sugestão de possíveis soluções contribuem na eficácia e rapidez de atendimento nas áreas de suporte.

1.1 BANCOS DE DADOS

Conforme Reese (2000), a base de dados é o coração de qualquer sistema empresarial. As informações e objetos de negócios de uma empresa precisam de alguma forma serem seguramente guardados e acessados a qualquer momento. O banco de dados fornece esse mecanismo de armazenamento e segurança. Desta forma, a ferramenta proposta por este trabalho irá realizar acessos a bancos de dados de três tipos: SQL Server, Oracle e PostgreSQL. Pois, dependendo do cenário em que o ERP está instalado, a ferramenta deverá acessar os três bancos de dados diferentes.

Para isto a ferramenta utiliza a API Java JDBC, que permite incorporar instruções SQL como argumentos a métodos em JDBC. Ainda segundo Reese (2000), estas implementações roteiam as chamadas SQL para o banco de dados proprietário que as reconhece naturalmente. Na visão do programador, o uso da API elimina a preocupação de como a aplicação está roteando as instruções SQL para o banco de dados. Desta forma, a facilidade fornecida pela API JDBC oferece total liberdade de quaisquer questões específicas de bancos de dados, podendo executar o mesmo código não importando o banco de dados que está presente. Na figura 1 pode-se verificar a estrutura de camadas de conexão utilizadas pela API JDBC da aplicação até o banco de dados. A figura ilustra como a API Java JDBC é utilizada na ferramenta apresentada neste artigo e utilizada para conexão nos diversos bancos de dados utilizados durante a pesquisa do erro de integração, tanto nos bancos de dados Oracle, SQL Server e PostgreSQL.

Figura 1 - Arquitetura de conexão da API JDBC



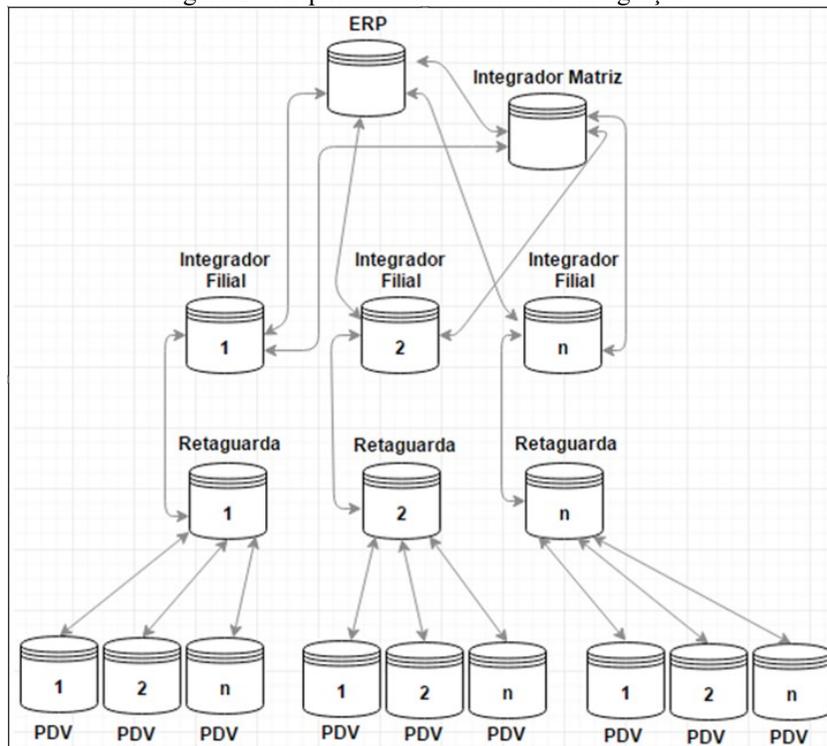
Fonte: <https://docs.oracle.com>

1.2 AMBIENTE ERP PROPRIETÁRIO

Como relatado anteriormente, este trabalho desenvolveu uma ferramenta para detecção de erros e apresentar dados para realizar diagnósticos em um ambiente específico de ERP. Este ambiente específico é composto por um ERP propriamente dito, conectado ao sistema de varejo através de um módulo de retaguarda, que por sua vez está conectado aos caixas conhecidos como PDV, acrônimo de ponto de venda.

Todas as camadas citadas são integradas através de integradores que possuem a função de gerir o tráfego de informações entre as aplicações. Por se tratar de um ambiente escalável, o ERP pode ter inúmeras empresas ou filiais com retaguardas, que por sua vez podem estar ligados a inúmeros PDV, sendo que em cada empresa ou filial haverá um integrador ligando esta empresa ou filial à matriz através do integrador matriz. Na Figura 2 pode-se observar a arquitetura básica de integração e suas camadas interligadas pelos integradores.

Figura 2 - Arquitetura das camadas de integração



Fonte: elaborado pelo autor.

O fluxo de integrações pode ocorrer do ERP para o retaguarda, do retaguarda para o ERP, retaguarda para o PDV e do PDV para o retaguarda. A integração ocorre no sentido ERP para o retaguarda quando uma informação é inserida ou alterada no ERP. Neste momento o integrador da filial de destino identifica que há uma pendência de integração, importa a informação do ERP e grava esta informação no banco de dados do retaguarda. Após isto, o integrador envia uma resposta de integração efetuada ao integrador matriz que por sua vez grava a confirmação de integração no banco de dados do ERP.

No sentido inverso, ao inserir ou alterar uma informação no retaguarda, o integrador filial lê a informação no banco de dados do retaguarda e a envia para o integrador matriz, que por sua vez grava no banco de dados do ERP. Na sequência o integrador matriz envia uma confirmação de integração ao integrador filial que grava esta confirmação no banco de dados do retaguarda.

A integração do retaguarda para o PDV ocorre quando existem atualizações de registros no banco de dados do retaguarda que são necessárias estarem no PDV. O PDV lê a base do retaguarda constantemente para verificar tais atualizações. Todas as movimentações efetuadas no PDV são integradas no retaguarda imediatamente após a conclusão da operação realizada no PDV.

Como a arquitetura é escalável, podendo ter inúmeras filiais, depende da interação do usuário para efetuar constantemente novos cadastros e alterações de parametrizações tanto na manutenção da estrutura cadastral já existente quanto na adaptação às novas necessidades de mercado e legislatórias.

Neste ponto podem surgir erros de integração como por exemplo uma venda efetuada no PDV para um determinado cliente, que por alguma instabilidade de conexão fica pendente de integração com o ERP. Durante a instabilidade este mesmo cliente é inativado no ERP e ao restabelecer a comunicação esta venda não é integrada no ERP pois o cliente está inativo. Caso este cliente desista da mercadoria, uma nota fiscal de devolução é emitida, porém esta nota não irá integrar enquanto a venda de origem não for integrada antes. Desta forma o crédito da devolução não é gerado e integrado ao ERP.

Para estes pontos de erro de integração, este artigo descreve o desenvolvimento de uma ferramenta de diagnóstico, diminuindo consideravelmente o tempo de resposta do atendimento da área de Suporte ao cliente. O diagnóstico fornece dados da inconsistência geradora do erro de integração, bem como suas dependências para que o problema seja resolvido.

1.3 TRABALHOS CORRELATOS

Nas pesquisas realizadas não foram encontradas aplicações que se destinem unicamente à função de identificação de falhas de integração, como no caso da ferramenta deste artigo. As aplicações apresentadas nesta seção possuem funções semelhantes e compõem uma de várias funcionalidades da ferramenta desenvolvida. Portanto serão descritos, para efeitos comparativos, apenas as funcionalidades correlatas.

O primeiro trabalho correlato apresentado no Quadro 1 chama-se Ensemble Application Integration Tool Technology (INTERSYSTEMS, 2017). É um software proprietário da InterSystems Ensemble, que tem como objetivo ser uma plataforma que auxilia na integração de sistemas diferentes entre si. O segundo correlato apresentado no Quadro 2 refere-se ao trabalho de Ribeiro (2003), que desenvolveu um sistema de monitoramento de QoS distribuído em redes de computadores. O terceiro correlato que segue no Quadro 3 refere-se à solução da Oracle, Oracle Application Integration Architecture (Oracle, 2011), que oferece tecnologia similar ao primeiro correlato, disponibilizando uma plataforma pré-concebida e posteriormente customizável orientada a facilitar a integração de diversos sistemas diferentes, gerenciando fluxos de dados e sincronizações de dados entre aplicativos.

Quadro 1 – Ensemble Application Integration Tool Technology

Referência	www.intersystems.com (INTERSYSTEMS, 2017)
Objetivos	Efetuar a integração entre as diferentes aplicações dentro de um ambiente empresarial, não importando o tipo de linguagem em que a aplicação foi construída.
Principais funcionalidades	Permite integrar sistemas diferentes entre si, permite criar regras de monitoramento, relatórios e identificação de problemas de integração nas trocas de informação entre as diferentes aplicações conectadas, em tempo real
Ferramentas de desenvolvimento	Não disponível, por se tratar de software proprietário.
Resultados e conclusões	Não disponível, por se tratar de software proprietário.

Fonte: elaborado pelo autor.

* A ausência de referências e citações quanto a fundamentação teórica, no que diz respeito ao sistema proprietário e ambiente ERP Varejo, deve-se ao fato de respeitar a política de confidencialidade da empresa desenvolvedora deste sistema.

Embora com propósitos diferentes, a Ensemble Application Integration Tool Technology assemelha-se à ferramenta desenvolvida neste artigo no fato de oferecer ao usuário dados sobre problemas de integração, para que este possa tomar uma ação mais rapidamente em busca da solução.

Quadro 2 – Um Sistema Para Monitoramento de Redes IP Baseado em Políticas

Referência	Universidade Federal do Rio Grande do Sul, Porto Alegre. (Ribeiro, 2003)
Objetivos	Fornecer facilidades de monitoramento e controle da qualidade de serviço (QoS) em redes de computadores.
Principais funcionalidades	Permite criar regras de monitoramento, aplicar políticas de rede baseada em monitoramento, detecta degradação do serviço de rede em tempo real avisando desta anomalia ao administrador de rede.
Ferramentas de desenvolvimento	Desenvolvida na linguagem de programação C, adotando como protocolo de comunicação o SNMP.
Resultados e conclusões	Os testes realizados pelo autor foram bem-sucedidos e validaram a escolha das bibliotecas e ferramentas utilizadas.

Fonte: elaborado pelo autor.

O trabalho “Um Sistema Para Monitoramento de Redes IP Baseado em Políticas” possui semelhanças com a ferramenta apresentada neste artigo no ponto em que a aplicação traz ao usuário dados de problemas que estão ocorrendo naquele momento. Embora a aplicação do autor seja em tempo real baseada em regras e a ferramenta deste artigo necessite ser acionado pelo usuário para que a busca seja realizada.

Quadro 3 – Oracle Application Integration Architecture

Referência	www.oracle.com (Oracle, 2011)
Objetivos	Efetuar a integração entre as diferentes aplicações dentro de um ambiente empresarial, não importando o tipo de linguagem em que a aplicação foi construída.
Principais funcionalidades	Permite integrar sistemas diferentes entre si, permite criar regras de monitoramento, relatórios e identificação de problemas de integração nas trocas de informação entre as diferentes aplicações conectadas, em tempo real.
Ferramentas de desenvolvimento	Não disponível, por se tratar de software proprietário.
Resultados e conclusões	Não disponível, por se tratar de software proprietário.

Fonte: elaborado pelo autor.

Da mesma forma que o quadro 1, a aplicação Oracle Application Integration Architecture assemelha-se à ferramenta desenvolvida neste artigo apenas no fato de oferecer ao usuário dados sobre problemas de integração, para que este possa tomar uma ação mais rapidamente em busca da solução.

DESCRIÇÃO DA FERRAMENTA

1.4 PRINCIPAIS REQUISITOS

A seguir são apresentados os principais requisitos da ferramenta:

- a) permitir ao usuário selecionar tipos de integração na busca a ser realizada (Requisito Funcional - RF);
- b) permitir ao usuário realizar uma busca utilizando uma data de referência como opcional (RF);
- c) permitir ao usuário realizar a busca por informações de integração independentemente do tipo de banco de dados (RF);
- d) permitir ao usuário salvar em banco de dados os ambientes com dados de acesso às bases em que a busca será realizada (RF);
- e) permitir ao usuário o gerenciamento das informações referente aos ambientes pesquisados (RF);
- f) permitir a visualização de informações referente a busca realizada (RF);
- g) permitir ao usuário salvar as requisições XML retornadas da busca (RF);
- h) permitir o monitoramento de falhas em tempo real, acusando problemas de integração através de alertas/mensagens (RF);
- i) permitir efetuar buscas nos bancos de dados dos ambientes cadastrados (RF);
- j) efetuar testes de conexão de todos os bancos cadastrados do ambiente (RF);
- k) utilizar o Sistema Gerenciador de Banco de Dados (SGBD) H2 para persistir os dados dos ambientes a serem pesquisados (RNF);
- l) efetuar conexão aos SGBDs SQL Server, Oracle e PostgreSQL, que são utilizados pelo sistema ERP e pelo sistema de Varejo (RNF);
- m) utilizar a linguagem de programação Java (RNF).

O fluxo principal da ferramenta inicia com o recebimento de dados do usuário, que deseja buscar no ambiente da integração ERP-varejo algum problema de integração que está investigando naquele momento. O usuário informa a filial, o tipo de integração a ser pesquisada e um intervalo de datas em que a pesquisa se baseará. Após inseridas estas informações a busca é iniciada.

A ferramenta primeiramente realiza uma pesquisa na base de dados do retaguarda. Em seguida utiliza as informações coletadas para realizar uma varredura nas bases de dados do integrador filial e do integrador matriz. Depois de coletados os dados no sistema de varejo, a ferramenta de busca faz a pesquisa no ERP para encontrar as chamadas dos webservices envolvidos nesta integração, com o objetivo de obter as informações enviadas na requisição do webservice e qual foi a resposta devolvida pelo ERP.

Caso a ferramenta não encontre as requisições e respostas no ERP, a pesquisa é feita no próprio integrador matriz, que contém os dados básicos utilizados na montagem da requisição XML enviada ao ERP pelo integrador matriz, pois a empresa pode desativar a gravação em banco de dados das requisições e respostas em XML trocadas entre os sistemas varejo e ERP.

Tendo em vista que a ferramenta opera sobre um sistema proprietário instalado em diferentes clientes comerciais, são necessárias configurações para a identificação dos servidores e conexões de banco de dados. Sempre que necessário, o usuário do suporte pode inserir, alterar ou remover dados de configuração através do botão de Configurações, conforme pode ser observado na figura 3 que ilustra a tela de configurações da ferramenta.

Figura 3 - Tela de configurações

The screenshot shows a web application interface for configuration. At the top, there are three icons: a gear for 'Configurações', a magnifying glass for 'Pesquisar', and a status icon for 'Status'. Below these is a tab labeled 'Configurações'. The main content area is divided into three sections:

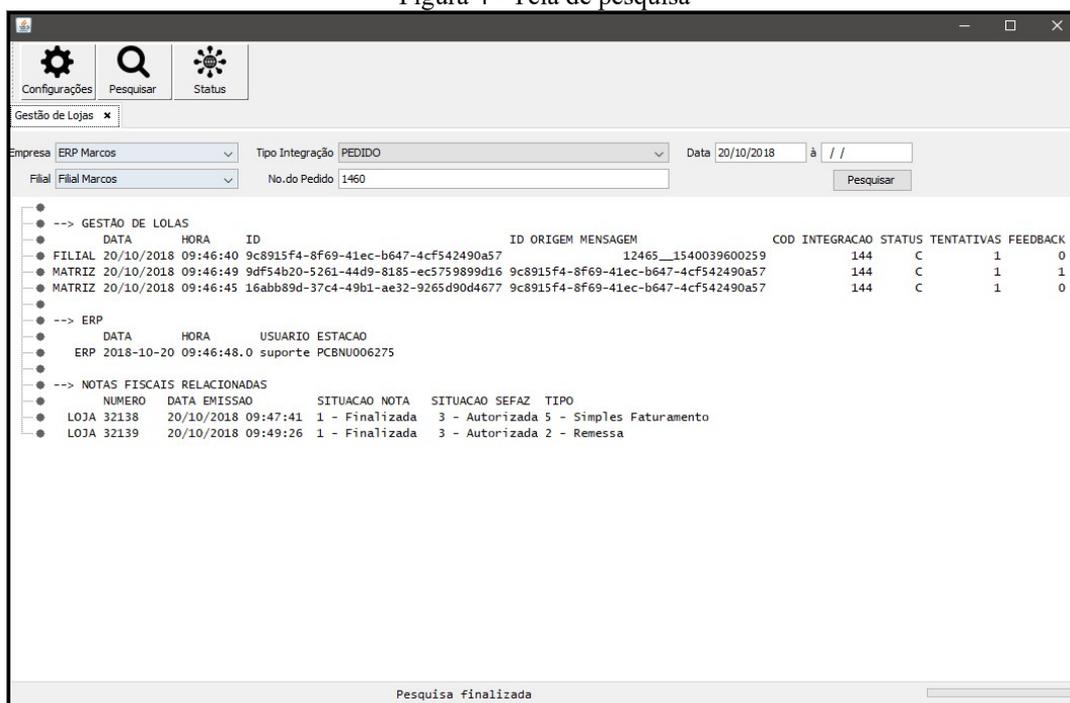
- Empresas cadastradas:** A dropdown menu shows 'ERP Marcos'. Below it are input fields for 'Nome da empresa' (ERP Marcos) and 'Código da empresa' (1). Buttons for 'Excluir', 'Gravar', and 'Adicionar' are at the bottom.
- Filiais da empresa:** A dropdown menu shows 'Filial Marcos'. Below it are input fields for 'Nome da Filial' (Filial Marcos) and 'Código da Filial' (1). Buttons for 'Excluir', 'Gravar', and 'Adicionar' are at the bottom.
- Bases relacionadas à Filial:** A dropdown menu shows 'xe'. Below it are several input fields: 'Tipo da aplicação' (ERP), 'Nome da base' (xe), 'Host da aplicação' (localhost), 'Porta de comunicação' (1521), 'Usuário do banco' (erp), and 'Senha do banco' (masked with three dots). Buttons for 'Excluir', 'Gravar', and 'Adicionar' are at the bottom.

At the bottom of the window, a status bar reads 'Aguardando dados para pesquisar.' with a progress indicator on the right.

Fonte: elaborado pelo autor.

Ao iniciar a busca, um objeto com os parâmetros da pesquisa é criado contendo os dados a serem buscados juntamente com os dados de conexão da filial escolhida. Os parâmetros da busca são definidos na tela de pesquisa, conforme ilustrado pela figura 4.

Figura 4 - Tela de pesquisa



Fonte: elaborado pelo autor.

Os usuários do suporte podem a qualquer momento verificar a conectividade dos acessos aos bancos de dados de todas as filiais da empresa selecionada, utilizando para isto o botão Status, conforme ilustrado na figura 5. Ao clicar no botão Testar, a ferramenta verifica se há conectividade em cada base de dados, ou seja, se a base de dados está acessível e pronta para que possa realizar as pesquisas necessárias.

Figura 5 - Tela de status

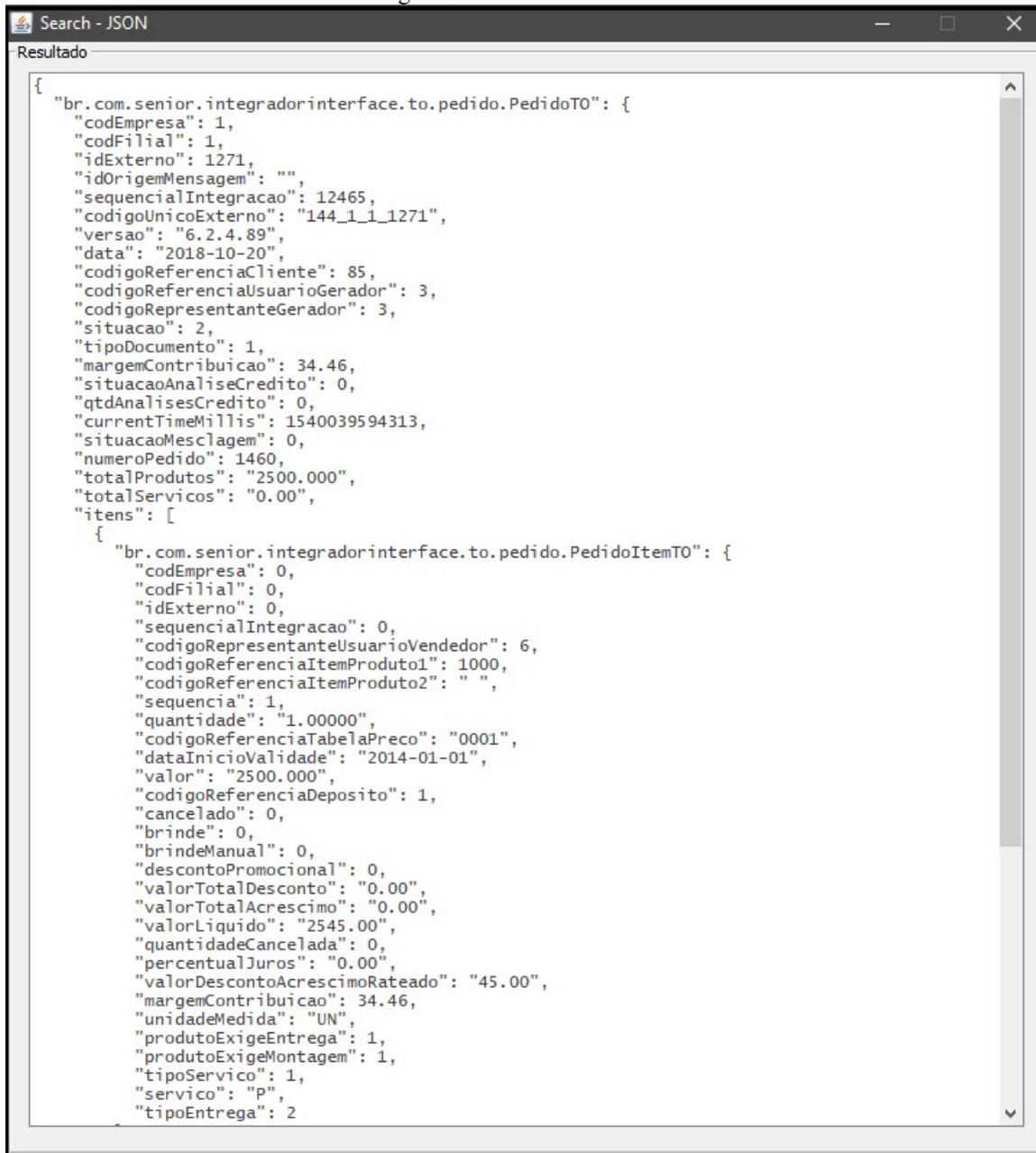


Fonte: elaborado pelo autor.

Ao finalizar a busca, os resultados encontrados nas bases de dados são apresentados na própria tela de pesquisa em forma de linhas, conforme figura 4, em que cada linha representa um registro existente no banco de dados pesquisado. Ao clicar em qualquer linha de resultado de pesquisa, a ferramenta abrirá uma nova tela com as informações detalhadas dos dados trafegados. Caso esta pesquisa tenha sido realizada nos integradores, o formato das

informações será em JSON (figura 6). Caso o resultado seja a troca de informações entre integradores e ERP, o formato das informações será em XML (figura 7).

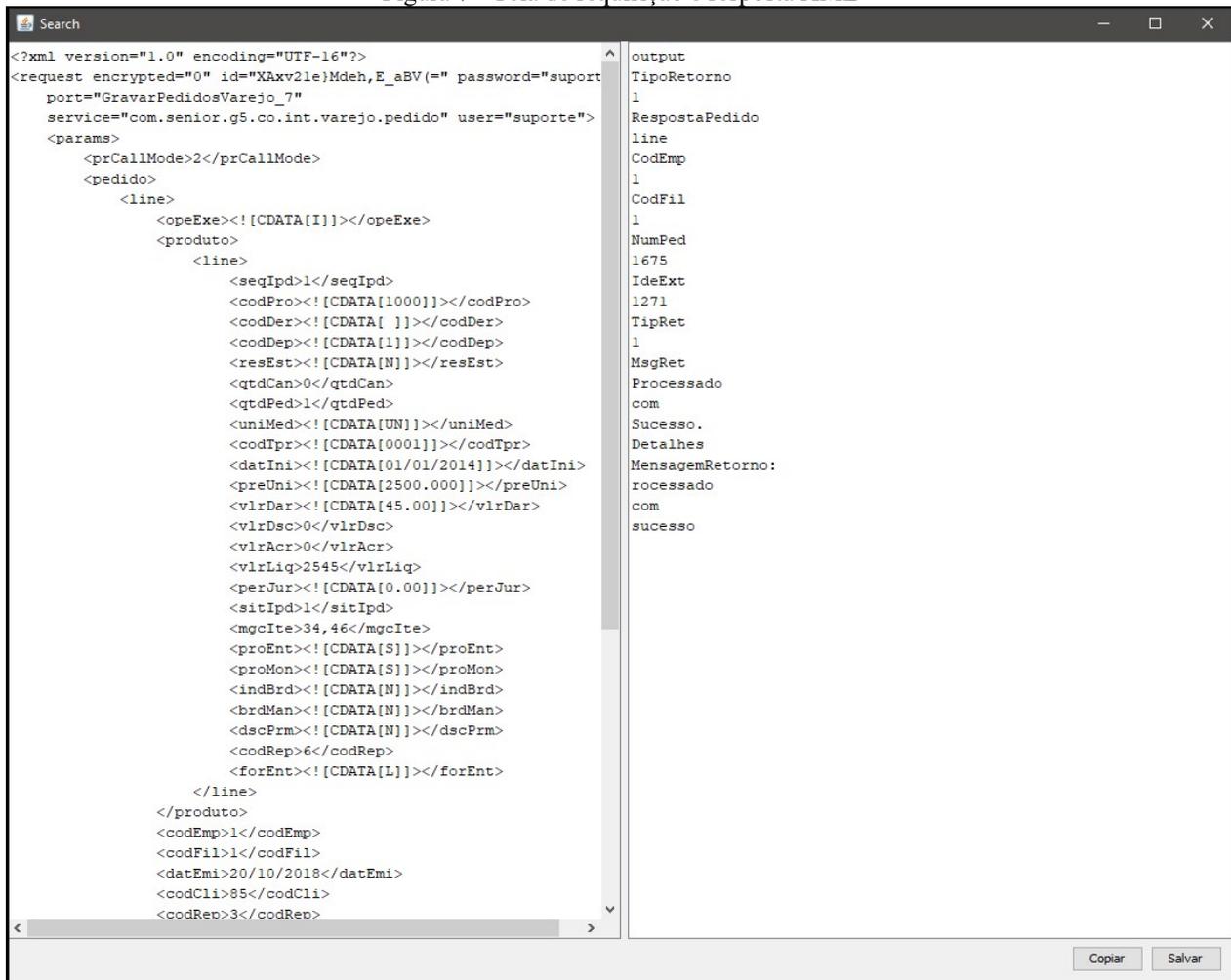
Figura 6 - Tela de dados JSON



```
{
  "br.com.senior.integradorinterface.to.pedido.PedidoTO": {
    "codEmpresa": 1,
    "codFilial": 1,
    "idExterno": 1271,
    "idOrigemMensagem": "",
    "sequencialIntegracao": 12465,
    "codigoUnicoExterno": "144_1_1_1271",
    "versao": "6.2.4.89",
    "data": "2018-10-20",
    "codigoReferenciaCliente": 85,
    "codigoReferenciaUsuarioGerador": 3,
    "codigoRepresentanteGerador": 3,
    "situacao": 2,
    "tipoDocumento": 1,
    "margemContribuicao": 34.46,
    "situacaoAnaliseCredito": 0,
    "qtdAnalisesCredito": 0,
    "currentTimeMillis": 1540039594313,
    "situacaoMesclagem": 0,
    "numeroPedido": 1460,
    "totalProdutos": "2500.000",
    "totalServicos": "0.00",
    "itens": [
      {
        "br.com.senior.integradorinterface.to.pedido.PedidoItemTO": {
          "codEmpresa": 0,
          "codFilial": 0,
          "idExterno": 0,
          "sequencialIntegracao": 0,
          "codigoReferenciaItemProduto1": 1000,
          "codigoReferenciaItemProduto2": "",
          "sequencia": 1,
          "quantidade": "1.00000",
          "codigoReferenciaTabelaPreco": "0001",
          "dataInicioValidade": "2014-01-01",
          "valor": "2500.000",
          "codigoReferenciaDeposito": 1,
          "cancelado": 0,
          "brinde": 0,
          "brindeManual": 0,
          "descontoPromocional": 0,
          "valorTotalDesconto": "0.00",
          "valorTotalAcrescimo": "0.00",
          "valorLiquido": "2545.00",
          "quantidadeCancelada": 0,
          "percentualJuros": "0.00",
          "valorDescontoAcrescimoRateado": "45.00",
          "margemContribuicao": 34.46,
          "unidadeMedida": "UN",
          "produtoExigeEntrega": 1,
          "produtoExigeMontagem": 1,
          "tipoServico": 1,
          "servico": "P",
          "tipoEntrega": 2
        }
      }
    ]
  }
}
```

Fonte: elaborado pelo autor.

Figura 7 - Tela de requisição e resposta XML



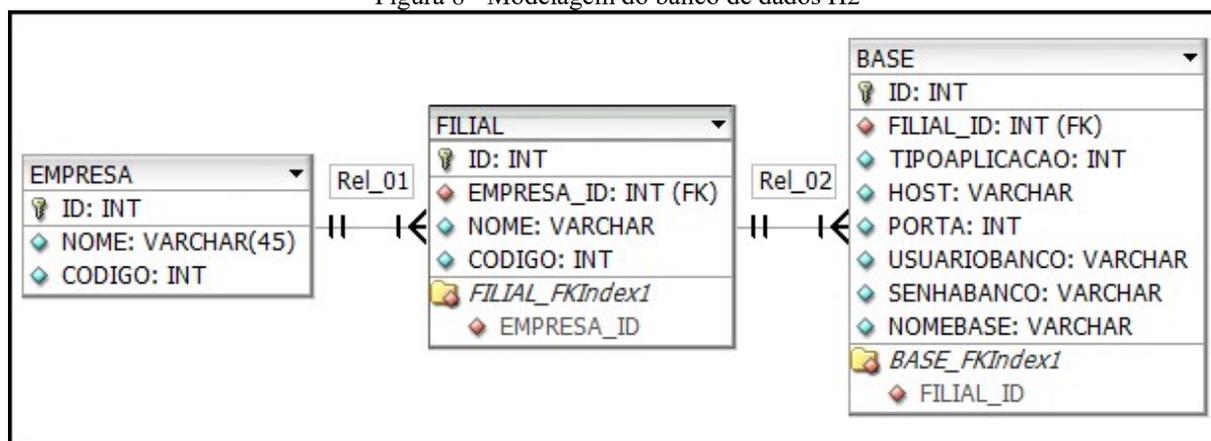
Fonte: elaborado pelo autor.

Estes diferentes formatos devem-se às diferentes tecnologias utilizadas no armazenamento das informações trafegadas entre os sistemas. No varejo as informações trafegam no formato JSON, porém o ERP utiliza webservice XML para a troca de informações. Desta forma no momento em que o integrador matriz comunica-se com o ERP o JSON é convertido em XML para a correta entrega dos dados ao ERP.

A ferramenta foi desenvolvida na linguagem de programação Java incorporando bibliotecas JDBC para conexão com os diversos tipos de bancos de dados que podem ser encontrados nos clientes, que são: Oracle, PostGreSQL e SQL Server. Também foi adicionado o JDBC para o banco do tipo H2. Este banco de dados funciona de forma interna à aplicação e é utilizado para armazenar os dados cadastrais das filiais para que a ferramenta tenha os dados de conexão e tipo de banco de dados a ser utilizado na conexão.

Na figura 8 pode-se verificar a arquitetura das tabelas do banco hospedado no H2. A arquitetura é simples e enxuta, pois não requer muitos dados persistidos porque a grande maioria dos dados necessários estão na própria base do cliente. O diagrama de classes segue no Apêndice A, figura 11.

Figura 8 - Modelagem do banco de dados H2



Fonte: elaborado pelo autor.

Os dados a serem utilizados nas pesquisas são armazenados em uma instância do objeto `ParamPesquisaGL`, que conterá, além da informação inserida pelo usuário, as informações de acesso aos bancos de dados da filial escolhida, o intervalo de data e tipo de integração.

Como existem mais que uma centena de tipos de integração, foram agrupadas as integrações similares para que o usuário escolha a principal e a ferramenta se encarregue em verificar as demais integrações relacionadas. Exemplo: notas fiscais de saída, notas fiscais de entrada, cancelamento de notas fiscais, devolução de venda, etc. Todas são notas fiscais. Ao selecionar Notas Fiscais, a ferramenta cria uma lista com as integrações relacionadas através de um mapa com base em enumeradores. Nos enumeradores cada integração relacionada possui um código e é este código que será utilizado para compor o comando SQL durante as pesquisas nas bases de dados. O quadro 4 ilustra como é criado o mapa com as integrações relacionadas.

Quadro 4 - Criação do MAP com integrações relacionadas

```

public Set<TipoCodigoIntegracaoErp> getDependencias() {
    if (integracoesRelacionadas == null) {
        synchronized(TipoCodigoIntegracaoErp.class) {
            if (integracoesRelacionadas == null) {
                integracoesRelacionadas = new HashMap<>();
                integracoesRelacionadas.put(CLIENTE, EnumSet.of(CLIENTE_RETAGUARDA)); //
                integracoesRelacionadas.put(CUPOM_FISCAL, EnumSet.of(CUPOM_FISCAL_CANCELAMENTO_RETAGUARDA, //
                    CUPOM_FISCAL_ELETRONICO_SAT_CANCELADO_RETAGUARDA, //
                    CUPOM_FISCAL_ELETRONICO_SAT_RETAGUARDA, //
                    CUPOM_FISCAL_RETAGUARDA)); //
                integracoesRelacionadas.put(NOTA_FISCAL, EnumSet.of(NOTA_FISCAL_SAIDA_CUPOM_RETAGUARDA, //
                    NOTA_FISCAL_SAIDA_RETAGUARDA, //
                    NOTA_FISCAL_CONSUMIDOR_ELETRONICA_CANCELADA_RETAGUARDA, //
                    NOTA_FISCAL_CONSUMIDOR_ELETRONICA_RETAGUARDA, //
                    NOTA_FISCAL_ENTRADA_RETAGUARDA)); //
                integracoesRelacionadas.put(ENTREGA, EnumSet.of(ENTREGA_RETAGUARDA1, //
                    ENTREGA_RETAGUARDA2)); //
                integracoesRelacionadas.put(PEDIDO, EnumSet.of(PEDIDO_ALTERACAO_FRETE_CD, //
                    PEDIDO_RETAGUARDA, //
                    PEDIDO_RETAGUARDA_CANCELAMENTO)); //
                integracoesRelacionadas.put(ASSISTENCIA, EnumSet.of(ASSISTENCIA_RETAGUARDA)); //
            }
        }
    }
    return integracoesRelacionadas.get(this);
}
    
```

Fonte: elaborado pelo autor.

No quadro 5 pode-se observar como é efetuada a extração dos dados das integrações e sua posterior utilização na composição do comando SQL a ser aplicado na pesquisa nos bancos de dados.

Quadro 5 - Extração dos códigos de integração e utilização dos códigos

```
private List<Long> obterIntegracaoERP(ParamPesquisaGL param, Base baseRetaguarda) throws ClassNotFoundException, SQLException {
    TipoCodigoIntegracaoErp tipo = param.getTipoIntegracao();
    List<Integer> codIntegracoes = new ArrayList<>();
    for (String s : tipo.getDependenciasToString(tipo.getName())) {
        codIntegracoes.add(TipoCodigoIntegracaoErp.getByNome(s));
    }
    return RetaguardaRN.getInstance().obterIntegracaoERP(param, baseRetaguarda, codIntegracoes);
}

StringBuilder codigos = new StringBuilder();
for (Integer codigo : codIntegracoes) {
    if (codigos.length() > 0) {
        codigos.append(",");
    }
    codigos.append(codigo);
}

List<ResultFila> listaResult = new ArrayList<>();

String info = Util.ajustarListaLongParaParametroIn(listaIdIntegracaoERP);

String sql = "SELECT \"ID\", \"CODINTEGRACAO\", \"IDORIGEM_MENSAGEM\", to_char(\"DATETIMEINC\", 'dd/mm/yyyy hh24:mi:ss') as \"DATETIMEINC\", \" +
    \"CONTEUDOFILA\", \"STATUS\", \"QTDENTATIVAS\", \"FEEDBACK\" \" + //
    \" FROM \"FILA\" \" + //
    \" WHERE \"DATETIMEINC\" >= TO_TIMESTAMP('\" + param.getDataInicial() + \"', 'DD/MM/YYYY hh24:mi:ss') \" + //
    \" AND \"CODINTEGRACAO\" in (\" + codigos.toString() + \") \" + //
    \" AND split_part(\"IDORIGEM_MENSAGEM\", '_', 1) in (\" + info + \") \" + //
    \" UNION ALL \" + //
    \"SELECT \"ID\", \"CODINTEGRACAO\", \"IDORIGEM_MENSAGEM\", to_char(\"DATETIMEINC\", 'dd/mm/yyyy hh24:mi:ss') as \"DATETIMEINC\", \" +
    \"CONTEUDOFILA\", \"STATUS\", \"QTDENTATIVAS\", \"FEEDBACK\" \" + //
    \" FROM \"FILA_HISTORICO\" \" + //
    \" WHERE \"DATETIMEINC\" >= TO_TIMESTAMP('\" + param.getDataInicial() + \"', 'DD/MM/YYYY hh24:mi:ss') \" + //
    \" AND \"CODINTEGRACAO\" in (\" + codigos.toString() + \") \" + //
    \" AND split_part(\"IDORIGEM_MENSAGEM\", '_', 1) in (\" + info + \"); //

conexao = BancoPostgre.abrirConexao(baseIntegradorFilial);
PreparedStatement pstmt = conexao.prepareStatement(sql);
ResultSet rs = pstmt.executeQuery();
```

Fonte: elaborado pelo autor.

RESULTADOS

No quadro 4 são apresentados os dados coletados nos testes da ferramenta realizados pelos usuários do suporte que, costumeiramente realizam as buscas de forma manual ou por scripts, com objetivo de avaliar a diferença entre a pesquisa manual de informações na base de dados e a pesquisa através da ferramenta desenvolvida. Os testes foram realizados em ambientes de produção e se categorizam entre testes executando comandos de banco manualmente, preenchendo scripts pré-prontos e o uso da ferramenta desenvolvida.

Os testes realizados para obter o tempo gasto em cada método foram obtidos cronometrando-se as ações de um mesmo usuário do suporte com média experiência em sua função. O tempo decorrido teve seu início registrado a partir do momento em que um erro era apresentado ao usuário para investigação. Os testes foram executados sempre na mesma estação e sempre no período vespertino, entre 13:30h e 15:00h, para que o teste fosse influenciado sempre com a mesma carga de uso do banco de dados em produção. O tempo cronometrado era encerrado no momento em que todas as informações coletadas nos bancos de dados envolvidos na investigação eram obtidas.

No quadro 6, existem três tipos de integração: pedido, nota fiscal e cupom fiscal. Para cada tipo foram testados os três métodos de busca: manual, por script pré-pronto e pela ferramenta de busca, por três vezes cada um, para que fosse obtido uma média de tempo.

Quadro 6 – Testes comparativos

Tipo de Integração	Método	Tempo decorrido (em minutos)*
Pedido	Manual	3:10 a 4:42
	Script	1:45 a 2:10
	Ferramenta de busca	0:15 a 1:04
Nota fiscal	Manual	2:55 a 3:04
	Script	1:40 a 1:49
	Ferramenta de busca	0:13 a 1:02
Cupom fiscal	Manual	3:04 a 4:10
	Script	1:32 a 2:20
	Ferramenta de busca	0:06 a 0:54

* Os resultados são uma média que variam de acordo com a agilidade do usuário, qualidade da conexão de internet e carga de uso do banco de dados.

Fonte: elaborado pelo autor.

Pode-se observar no quadro 6 que o ganho de desempenho é evidente se comparado com interação do usuário do suporte com o banco de dados. Os testes foram realizados com usuários de média experiência para que os dados

fossem condizentes com a experiência média da equipe de suporte. Como resultado, observou-se uma redução média de 75,53% do tempo despendido pelo usuário, considerando a média de execução utilizando scripts e manualmente.

CONCLUSÕES

O processo de desenvolvimento da ferramenta e os testes realizados no ambiente de suporte, onde a necessidade surgiu, proporcionaram uma experimentação das dificuldades encontradas pelos usuários do suporte para obter dados investigatórios acerca de problemas de integração. Evidenciaram também quão complexo é um sistema de integração distribuído e escalável, com sua base de dados em constante expansão.

Neste ponto, verificou-se que o ganho de tempo em obter dados de diversos banco de dados e a concentração de informações em único ponto, reduziu consideravelmente em pelo menos um terço, no pior caso, o tempo despendido pelo usuário do suporte, que deste modo pode tanto aumentar a sua produtividade quanto realizar um número maior de atendimentos. Vale salientar que a ferramenta desenvolvida e descrita neste artigo já está em uso pela equipe de suporte.

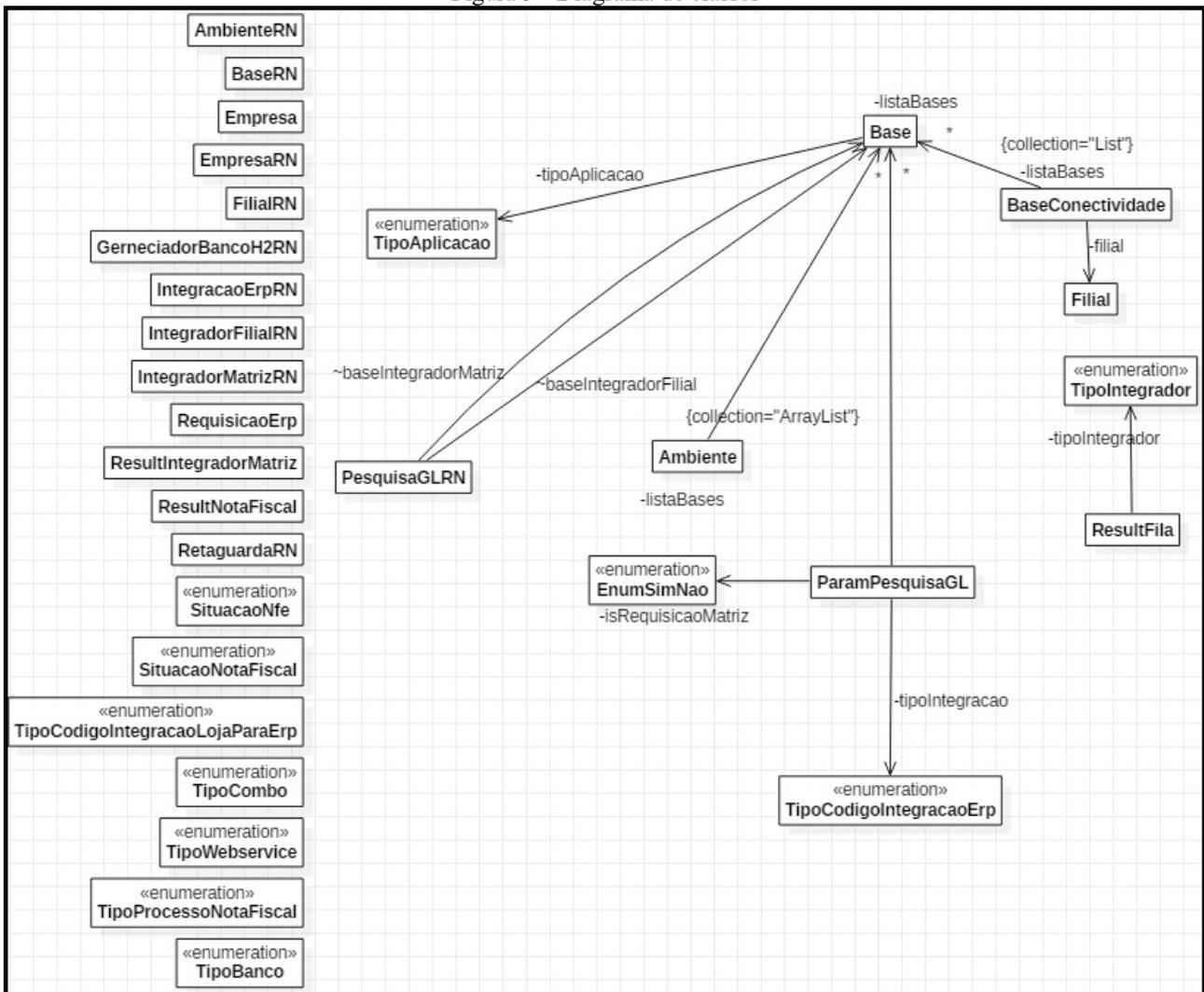
Apesar das dificuldades encontradas, tais como adaptar comandos SQL aos diversos tipos de bancos de dados e tornar estes comandos os mais genéricos possível devido as grandes possibilidades de variações, os objetivos foram alcançados e ao longo do desenvolvimento perceberam-se várias possibilidades de melhorias, tais como portar a aplicação para a nuvem e fornecer novas ferramentas, como por exemplo, a possibilidade de efetuar ajustes na base de dados pela própria ferramenta de busca, tendo em vista que a ferramenta tem acesso a todas as bases de dados do sistema ERP e varejo.

REFERÊNCIAS

- BENTES, Otavio M. **Atendimento ao Cliente**. Curitiba: IESDE, 2012, p. 108.
- BEISSE, Fred. **A Guide to Computer User Support: for Help Desk & Supporte Specialists**. 5th ed. USA: Carnage Learning, 2014, p. 290.
- BRUTON, Noel. **How to Manage the IT Helpdesk: A guide for user support and call centre**. 2nd ed. USA: Butterworth Heinemann, 2002, p. 286.
- CAIÇARA JUNIOR, Cícero. **Sistemas Integrados de Gestão: Uma abordagem gerencial**. 3. ed. IBPEX, Curitiba, 2009.
- COHEN, Roberto. **Implantação de Help Desk e Service Desk: Como construir e manter pequenos e médios centros de suporte técnico, help desk e service desk**. São Paulo: Novatec, 2008, p. 131.
- FRANTZ, Rafael Z., et al. Desafios para a Implantação de Soluções de Integração de Aplicações Empresariais em Provedores de Computação em Nuvem. In: JORNADA DE PESQUISA, XIX, 2014. Ijuí. **Anais...** Ijuí: Unijui, 2014, p. 8. Disponível em <<https://www.publicacoeseventos.unijui.edu.br/index.php/salaconhecimento/article/view/4088>>. Acesso em: 21/05/2017.
- GUARINO, José Carlos. **Sistemas Integrados de Gestão: Simplíssimo livros digitais**, 2015. Disponível em: <<https://itunes.apple.com/br/book/sistemas-integrados-gestao/id696599005?mt=11>>. Acessado em: 14/05/2017.
- HILES, Andrew; GUNN, Yvonne. **Creating A Customer-Focused Help Desk: How to Win and Keep Your Customers**. USA: Paperback, 2000, p. 11.
- INTERSYSTEMS, **Ensemble Platform for connected applications**. USA, 2017. Disponível em: <<http://www.intersystems.com/our-products/ensemble/enterprise-service-bus-esb-service-orientated-architecture-soa/ensembles-enterprise-service-bus-esb-capabilities>>. Acessado em: 23/03/2017.
- MEIRELES, Manuel. **Sistemas de Informação: Quesitos de excelência dos sistemas de informações**. São Paulo: Arte & Ciência, 2001.
- ORACLE, **Oracle Fusion Middleware: Concepts and Technologies Guide for Application Integration Architecture Foundation Pack**. USA, 2011. Disponível em: <<http://www.oracle.com/technetwork/apps-tech/aia/documentation/index.html>>. Acessado em: 24/03/2017.
- PICCOLI, Leonardo B. **Soluções Híbridas de Hardware/Softwre Para a Detecção de Erros em Systems-On-Chip (SoC) de Tempo Real**, 2006, 159 f. Dissertação (Pós-Graduação de Engenharia Elétrica) – Programa de Pós-Graduação de Engenharia Elétrica - PPGE, Pontifícia Universidade Católica do Rio Grande do Sul – Faculdade de Engenharia. Disponível em: <<http://tede2.pucrs.br/tede2/bitstream/tede/3006/1/385283.pdf>>. Acesso em: 30/03/2017.
- PRADO, Edmir P. V.,; SOUZA, Cesar Alexandre de. **Fundamentos de Sistemas de Informação**. Rio de Janeiro: Elsevier, 2014.
- REESE, George. **Database Programming with JDBC and JAVA**. 2. ed. EUA: O'Reilly, 2000, p. 9-10.
- RIBEIRO, Marcelo B. **Um Sistema para monitoração de redes IP baseado em políticas**. 2003, 99 f. Dissertação (Mestrado em Ciência da Computação) – Pós-Graduação em Ciências da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre.

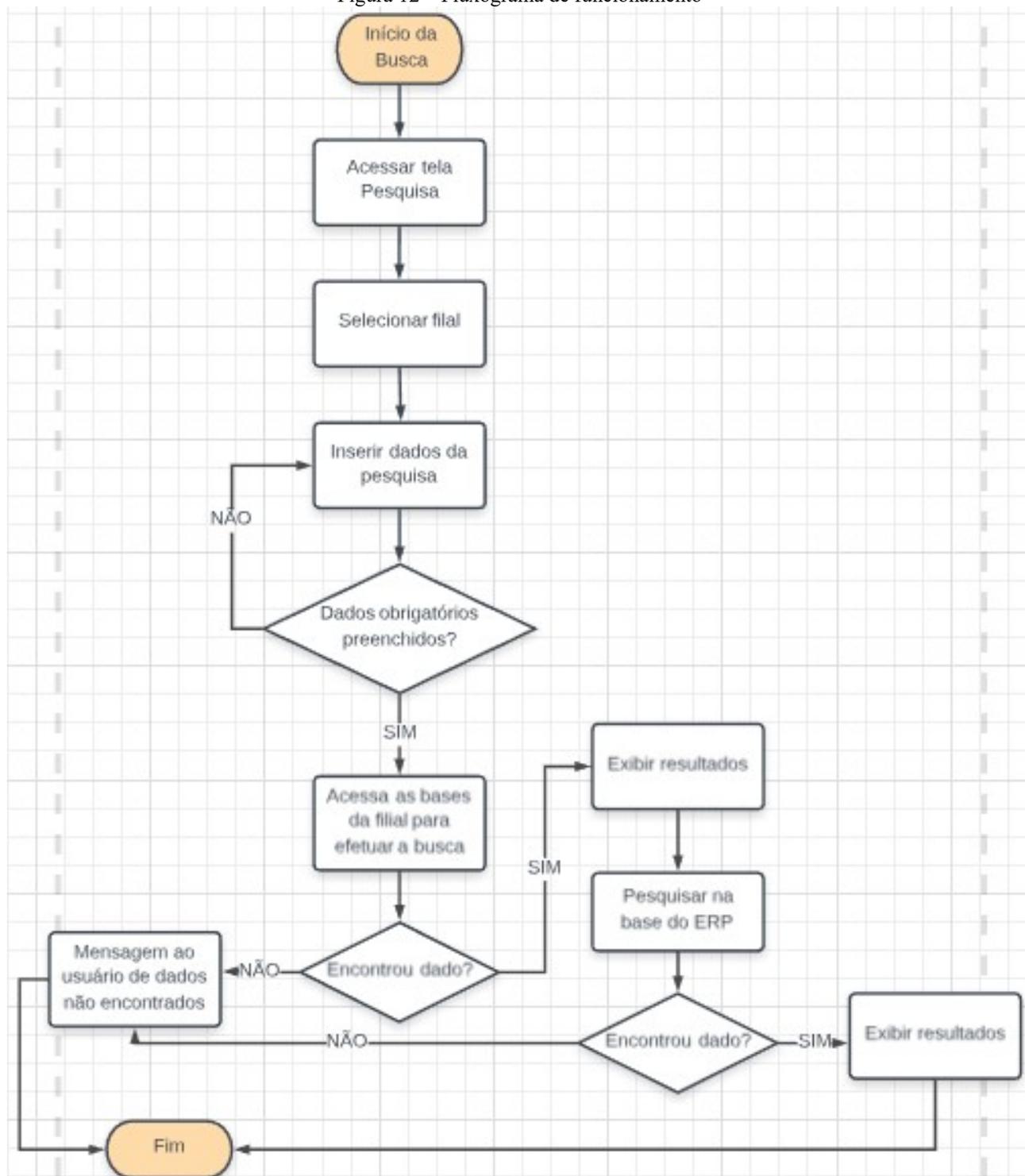
APÊNDICE A – DIAGRAMAS DE ESPECIFICAÇÃO

Figura 9 - Diagrama de classes



Fonte: elaborado pelo autor.

Figura 12 – Fluxograma de funcionamento



Fonte: elaborado pelo autor..