

# FERRAMENTA DE COLETA E ANÁLISE DE ESTATÍSTICAS PARA SISTEMA GERENCIADOR DE BANCO DE DADOS POSTGRESQL

Evandro Franzen<sup>1</sup>, Fabiano Tomasini<sup>2</sup>

**Resumo:** O volume cada vez maior de dados tem exigido atenção com a velocidade com que as informações são obtidas e quanto à forma com que os dados são armazenados. É fundamental garantir a integridade das informações e o bom desempenho em seu acesso. Algumas dessas preocupações podem ser amenizadas com algumas ações que podem ser tomadas em relação ao Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido. Este trabalho tem o objetivo de disponibilizar uma ferramenta que possibilite identificar possíveis problemas de desempenho. Por meio da ferramenta desenvolvida, são coletadas estatísticas do banco de dados que permitem detectar impasses que estejam relacionados à administração do banco de dados, podendo, assim, tornar visível a necessidade de ações para melhorar a eficiência do sistema.

**Palavras-chave:** Banco de dados. Administração de dados. Consultas. Transações. Concorrência.

## 1 INTRODUÇÃO

O rápido crescimento da internet na última década fez dela a maior fonte de dados de acesso público em todo o mundo (LIU, 2011). Atualmente, há imensa quantidade de dados disponíveis nos mais diversos locais. Sistemas de informação, páginas de notícias, *blogs* e redes sociais são alguns exemplos de fontes de dados que podem ser utilizadas para extrair informações relevantes.

Com o desenvolvimento tecnológico, os *softwares* exigem mais dos computadores, ou seja, maior poder de processamento de modo que informações importantes sejam obtidas de maneira fácil e rápida. Sendo essas informações a chave para a boa tomada de decisão, podemos considerar que a mesma garante a sobrevivência de uma organização no mercado global, motivando diversos estudos na área de banco de dados (ROB; CORONEL, 2011).

Segundo Silberschatz, Korth e Sudarshan (2012), um banco de dados é considerado uma coleção de dados relacionados que são fornecidos ao usuário final de maneira conveniente, eficiente e segura. Visto que a manipulação de informações é tão importante para a gestão de uma organização, cientistas da computação têm desenvolvido grande conjunto de conceitos e técnicas de gerenciamento de dados visando à otimização da obtenção desses dados.

Para garantir o armazenamento correto e manter a integridade dos dados, é indispensável monitorar o uso do sistema gerenciador de banco de dados, verificando, por exemplo, a ocorrência de impasses (*deadlocks*) que podem afetar a disponibilidade do *software*. Além disso, é fundamental analisar a performance das consultas realizadas, o que requer o conhecimento das estatísticas do

---

1 Mestre em Computação pela Universidade Federal do Rio Grande do Sul (2002). Professor assistente do Centro Universitário UNIVATES.

2 Acadêmico do Centro Universitário UNIVATES.

banco de dados, como número de consultas por tabela que utilizaram ou não utilizaram índices, percentual de consultas por tabela que utilizaram ou não utilizaram cache.

Este trabalho apresenta o desenvolvimento de uma ferramenta que contribui para a manutenção de grandes bancos de dados com ênfase especial na performance das aplicações que dependem desse serviço. Para obter o resultado esperado, foram abordados gargalos de performance relacionados à administração de banco de dados por meio de estatísticas extraídas do Sistema de Gerenciamento de Banco de Dados (SGBD). As informações coletadas servem para medir a performance do banco e alertando o *Database Administrator* (DBA) sobre possíveis pontos de melhorias e possíveis problemas.

A ferramenta conta com um módulo coletor de estatísticas, que deve ter acesso ao banco de dados a ser monitorado. Neste trabalho foi desenvolvido o módulo coletor para o SGBD *PostgreSQL*, porém, a arquitetura desenvolvida permite que seja implementado o monitoramento de outros SGBDs em trabalhos futuros, por meio de coletores específicos.

Com base na importância do monitoramento do SGBD para manter a disponibilidade do serviço, e da análise de estatísticas para garantir a otimização de consultas, é evidenciada a necessidade de haver ferramentas que auxiliam na execução desse trabalho.

Atualmente, existem várias ferramentas que têm como objetivo gerenciar tabelas e acessos, porém, quando se deseja monitorar estatísticas do banco de dados, são encontrados poucos *softwares* que fornecem essas informações de maneira gráfica. Algumas ferramentas atingem o objetivo, mas acabam sendo descartadas por serem pagas, descontinuadas, não suportar mais de um SGBD ou simplesmente não proporcionam o que se deseja de maneira clara.

Tendo como base essa realidade, o presente trabalho visa a proporcionar ao DBA uma ferramenta que possibilita o monitoramento e a análise de estatísticas de maneira gráfica e de fácil manuseio. A ferramenta auxilia a administração, o monitoramento e a otimização de um banco de dados que utilize qualquer SGBD e que tenha grande volume de informação armazenada.

## 2 FUNDAMENTAÇÃO TEÓRICA

Segundo Ramakrishnan e Gehrke (2008), um sistema de gerenciamento de banco de dados é um *software* que foi desenvolvido para auxiliar na manipulação e na organização de vastos conjuntos de dados. A necessidade de um *software* com esse fim tem crescido rapidamente, pois, antes de surgirem, os dados eram armazenados em arquivos e era necessário escrever códigos específicos na própria aplicação para gerenciá-los. Siberschartz, Korth e Sudarshan (2012) também consideram esse sistema como uma coleção de dados inter-relacionados e um compilado de programas para manipulá-los.

Conforme Ramakrishnan e Gehrke (2008), um modelo de dados é considerado uma seleção de construtores de alto nível que ocultam detalhes de baixo nível do armazenamento de informações. O SGBD permite que o usuário final defina as informações a serem armazenadas em relação ao modelo de dados. Grande parte dos sistemas atuais baseia-se no modelo relacional. Um banco de dados que utiliza esse modelo contém um conjunto de tabelas (entidades), em que cada uma recebe uma denominação única que irá representá-la. Cada tabela contém campos (atributos) que também recebem uma denominação única e um tipo específico de dados (SUDARSHAN, 2012).

### 2.1 Catálogo do sistema

Segundo Elmasri e Ramez (2011), um SGBD é um sistema genérico projetado para atender a diversas aplicações de banco de dados. Dessa maneira, sempre que for necessário conhecer a estrutura de um banco de dados específico, deve-se recorrer ao catálogo, pois é lá que o SGBD

armazena os metadados do esquema, como informações sobre tabelas, colunas e informações de controle interno.

No catálogo, além de informações sobre a estrutura do banco de dados, são armazenadas também informações estatísticas que são usadas pelo SGBD para otimizar o plano de execução de uma consulta. Além disso, as estatísticas podem ser usadas pelo administrador de banco de dados para identificar problemas de performance, como índices não utilizados e a ocorrência de *deadlocks* (SILBERSCHATZ; KORTH; SUDARSHAN, 1999).

## 2.2 Processamento de consultas

O processamento de consultas refere-se ao conjunto de tarefas incluídas na extração de informações de um banco de dados. As tarefas incluem tradução das consultas em uma linguagem que seja entendida pelo SGBD e que permitem ser usadas no nível físico do sistema de arquivos e uma série de transformações de otimização das consultas (SUDARSHAN, 1999).

Antes de qualquer consulta iniciar o processamento, o SGBD precisa transcrever a consulta para linguagem interna. A linguagem de banco de dados SQL utilizada na maioria dos SGBDs relacionais é ideal para a ação humana, porém, essa representação não é a ideal para o sistema interno de consultas.

Segundo Blumm e Fornari (2006), no processamento de consultas, o otimizador do SGBD desempenha o papel principal. Há dois tipos de otimizações:

- otimização baseada em regras heurísticas: essas regras estão inclusas ao SGBD Oracle e produzem bons resultados, porém não há comprovação que garanta sua correção em todas as consultas (BLUMM; FORNARI, 2006);
- otimização baseada em estatísticas: nessa opção, o otimizador de consultas aplica algumas fórmulas para calcular o custo (tempo de processamento + tempo de acesso aos dados em disco) de várias alternativas possíveis e opta pela que apresentar o menor custo estimado. Os dados estatísticos, como o número de linhas de uma determinada tabela, são mantidos no dicionário de dados do SGBD (BLUMM; FORNARI, 2006).

De acordo com Blumm e Fornari (2006), alguns SGBDs utilizam apenas um dos métodos apresentados e outros permitem a escolha do método que apresentar melhores resultados. Embora a responsabilidade por otimizar uma consulta seja do sistema, o administrador deve compreender o processo de execução e conhecer detalhadamente a estrutura do banco de dados, as tabelas, os índices, além das transações e dos aspectos relacionados ao controle de concorrência.

## 2.3 Transações e concorrência

Segundo Ramakrishnan e Gehrke (2008), uma transação é vista pelo SGBD como uma série ou lista de ações. As ações que podem ser executadas por uma transação incluem leituras e gravações de objetos de banco de dados.

Cada transação, além de ler e gravar, deve especificar como ação final ou a efetivação (incluir com sucesso) ou cancelamento (desfazer todas as ações executadas até o momento). As transações interagem umas com as outras apenas por meio de operações de leitura e gravação do banco de dados. Elas, por exemplo, não podem trocar mensagens (RAMAKRISHNAN; GEHRKE, 2008).

De acordo com Silberschartz, Korth e Sudarshan (1999), um conjunto de várias operações no banco de dados é visto pelo usuário como uma única operação. Por exemplo, a transferência de fundos de uma conta corrente para uma poupança é uma operação única sob o ponto de vista do cliente, porém, internamente no banco de dados, ela envolve várias operações.

Uma transação atualiza vários itens de dados e geralmente é o resultado da execução de um programa de usuário escrito em uma linguagem de programação. Cada transação é delimitada por declarações que marcam seu início e fim. As operações que serão executadas nessa transação são todas as invocadas entre o começo e o fim (SILBERSCHATZ; KORTH; SUDARSHAN, 1999).

Conforme Siberschartz, Korth e Sudarshan (1999), para assegurar a integridade dos dados, é exigido que o sistema de banco de dados mantenha as seguintes propriedades das transações:

- atomicidade: ou todas as operações da transação são executadas com sucesso no banco de dados ou nenhuma será executada;
- consistência: a execução de uma transação isolada, ou seja, sem a execução concorrente de uma outra transação, preserva a consistência do banco de dados;
- isolamento: sabendo que diversas transações podem ser executadas de forma concorrente, o sistema deve garantir que, para cada par de transações  $T_i$  e  $T_j$ ,  $T_j$  tenha terminado sua execução antes de  $T_i$  começar, ou que  $T_j$  tenha começado sua execução após  $T_i$  terminar. Assim, cada transação não toma conhecimento de outras transações concorrentes no sistema;
- durabilidade: depois de a transação ser finalizada com sucesso, as mudanças que ela fez no banco de dados persistem até mesmo se houver falhas no sistema.

Uma das propriedades fundamentais de uma transação é o isolamento. Entretanto, quando várias transações são executadas concorrentemente no banco de dados, essa propriedade pode não ser mantida. Para garantir que essa propriedade seja preservada, o SGBD precisa controlar a interação entre as transações concorrentes. Esse controle é atingido por uma série de procedimentos chamados de esquemas de controle de concorrência (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

De acordo com Siberschartz, Korth e Sudarshan (2012), existe uma variedade de esquemas (protocolos) de controle de concorrência. Entre eles estão protocolos baseados em bloqueios, protocolos de bloqueios baseados em duas fases e protocolo baseado em grafo. O protocolo baseado em bloqueio é o mais utilizado. Nesse esquema, é exigido que os dados sejam acessados de maneira mutuamente exclusiva, ou seja, enquanto uma transação está acessando uma informação ou dado, nenhuma outra transação pode modificar esse dado. O método mais comum para atender a esse requisito é permitir que uma transação acesse esse dado somente se estiver atualmente mantendo um bloqueio sobre ele.

## 2.4 Índices

Segundo Rob e Coronel (2011), um índice pode ser considerado uma estrutura ou arquivo associado a uma tabela e tem como objetivo melhorar o desempenho de acesso a dados de uma ou mais linhas de uma tabela. Dessa maneira, o índice cria ponteiros para os dados armazenados nas colunas, que devem ser específicas em sua criação.

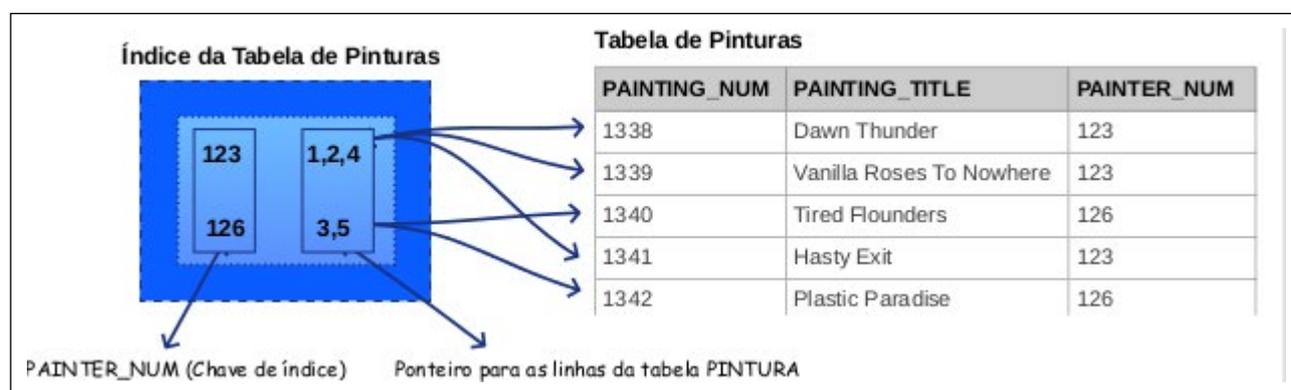
Os índices podem ser utilizados de diferentes maneiras e levar a planos de execução mais rápidos do que qualquer outro que não utiliza índices (RAMAKRISHNAN; GEHRKE 2008).

Um índice é uma disposição ordenada utilizada para acessar logicamente uma tabela. Considerando do ponto de vista conceitual, é formado de uma chave de índice e de um conjunto de ponteiros, em que a chave de índice é o ponto de referência do mesmo. Cada chave aponta para a localização dos dados identificados por ela. Os índices executam papel importante na estrutura de um SGBD, pois, ao definir a chave primária de uma tabela, o SGBD cria automaticamente um índice

exclusivo para a(s) coluna(s) dessa chave (ROB; CORONEL, 2011). Uma tabela pode ter vários índices, porém, cada um deles está associado a apenas uma tabela.

Por exemplo, caso seja necessário procurar todas as pinturas criadas por um determinado pintor no banco de dados, sem um índice, é necessário ler todas as linhas da tabela pintura e ver se o atributo correspondente ao código do pintor é do pintor solicitado. No entanto, se for criado um índice utilizando-se o código do pintor, basta procurar o valor adequado desse atributo no índice e encontrar os ponteiros correspondentes. Em termos conceituais, o índice se assemelha à representação ilustrada na Figura 4 (ROB; CORONEL, 2011).

Figura 1 – Exemplo de um índice para uma tabela de pinturas



Fonte: modificado de Rob e Coronel (2011).

### 3 METODOLOGIA

Esta seção objetiva apresentar os princípios metodológicos, as tecnologias e as técnicas que foram utilizados para o desenvolvimento da ferramenta proposta.

A pesquisa desenvolvida neste trabalho pode ser caracterizada como exploratória. A pesquisa exploratória visa a aumentar o conhecimento do pesquisador sobre determinado problema. Ela pode ser composta por revisão de literatura, entrevistas, entre outros (GIL, 2002).

Considerando o objetivo geral, que é disponibilizar uma ferramenta *web* de coleta e análise de estatísticas para o sistema gerenciador de banco de dados PostgreSQL, foram realizados estudos sobre os temas relacionados, com a finalidade de desenvolver uma ferramenta em conformidade com os conceitos já existentes. Foram analisadas ferramentas existentes e as tecnologias empregadas no seu desenvolvimento.

Após os estudos e a avaliação dos *softwares* existentes, seguiu-se o trabalho de especificação e projeto da ferramenta, culminando posteriormente com a implementação do sistema. A ferramenta desenvolvida foi utilizada e testada em um ambiente de produção, em uma base de dados com volume considerável de informações armazenadas e grande volume de acessos.

Na etapa de especificação, foram definidos os requisitos funcionais e não funcionais. Optou-se pela criação de uma ferramenta baseada em plataforma *web*, *open source* que possibilite ao usuário DBA extrair e visualizar estatísticas do *software* PostgreSQL e futuramente de qualquer SGBD, contribuindo, assim, para o monitoramento e a tomada de ações para garantir a performance da base de dados. Uma das funcionalidades essenciais é a coleta de estatísticas do banco de dados, que

permite detectar impasses que estejam relacionados à administração do banco de dados, tornando visível a necessidade de criação de índices e alterações em configurações do sistema utilizado.

Para o desenvolvimento da ferramenta proposta, são utilizadas diversas tecnologias, sendo as principais:

- PHP (*Hypertext Preprocessor*): é uma linguagem de *script open source* de uso geral, muito utilizada e adequada para o desenvolvimento *web*. O PHP foi utilizado para desenvolver o agente coletor e os processamentos relacionados à ferramenta *web* (PHP, 2015);
- PostgreSQL: é um SGBD, objeto relacional de código aberto extremamente robusto e confiável. O PostgreSQL foi utilizado para armazenar as estatísticas e as informações coletadas, os usuários que acessarão a ferramenta e as configurações do sistema (PostgreSQL, 2015);
- JavaScript: é uma linguagem de programação do lado cliente processada pelo próprio navegador. O JavaScript foi utilizado para criar efeitos nas interfaces da ferramenta para proporcionar maior interatividade (JAVASCRIPT, 2015);
- Bootstrap: é um framework para facilitar a criação de *sites* ou páginas *web* (responsivo). O Bootstrap foi utilizado para deixar a ferramenta mais intuitiva e utilizável em qualquer dispositivo que tenha acesso à *web* (BOOTSTRAP, 2015);
- Morris: é uma biblioteca leve que usa jQuery e utilizada para tornar fácil a tarefa de desenhar gráficos simples. O Morris foi utilizado para representar graficamente as estatísticas coletadas.

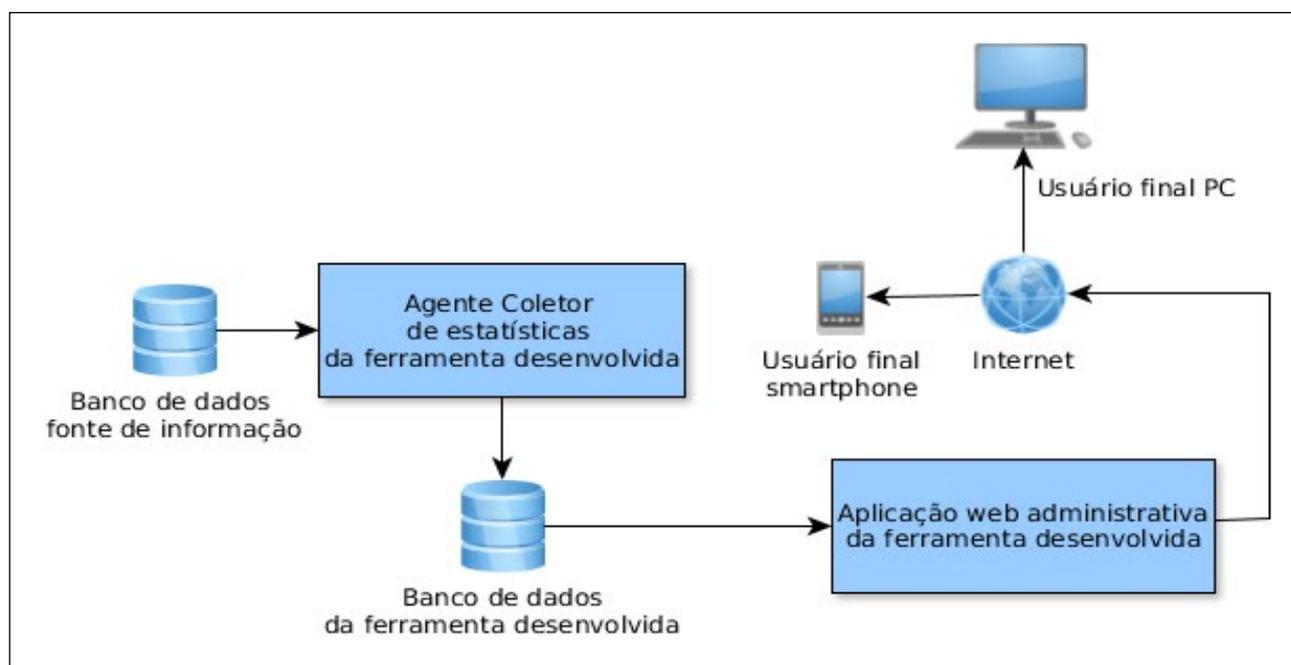
## 4 RESULTADOS E DISCUSSÃO

Esta seção descreve a ferramenta desenvolvida, sua arquitetura, suas funcionalidades e sua interface e apresenta a demonstração da utilização do referido sistema em uma base de dados real.

### 4.1 Arquitetura e módulos da ferramenta

A Figura 2 mostra a arquitetura da ferramenta desenvolvida, na qual existe um agente coletor que se comunica com a base de dados monitorada e o banco de dados responsável por armazenar as estatísticas. A estrutura desenvolvida possibilita que seja monitorado um outro SGBD, bastando que seja implementado o módulo coletor específico para o sistema gerenciador desejado.

Figura 2 - Arquitetura da ferramenta desenvolvida



Fonte: Dos autores.

O agente é a parte da ferramenta responsável pela coleta das estatísticas da base de dados que está sendo monitorada, e é por meio dessa rotina que a base de dados do SMBD vai ser populada. A rotina desenvolvida utiliza a linguagem de programação PHP e se comunica com o sistema de monitoramento pelo protocolo Simple Object Access Protocol (SOAP). O agente não precisa necessariamente estar na mesma máquina em que o SGBD se encontra, bastando que haja comunicação entre os dois pontos.

O agente coletor obtém os dados a partir de diversas consultas, que são enviados para o SMBD. Um exemplo de consulta é apresentado na Figura 3, na qual consta o código utilizado para obter informações gerais da base de dados.

Figura 3 - Consulta que obtém informações da base de dados

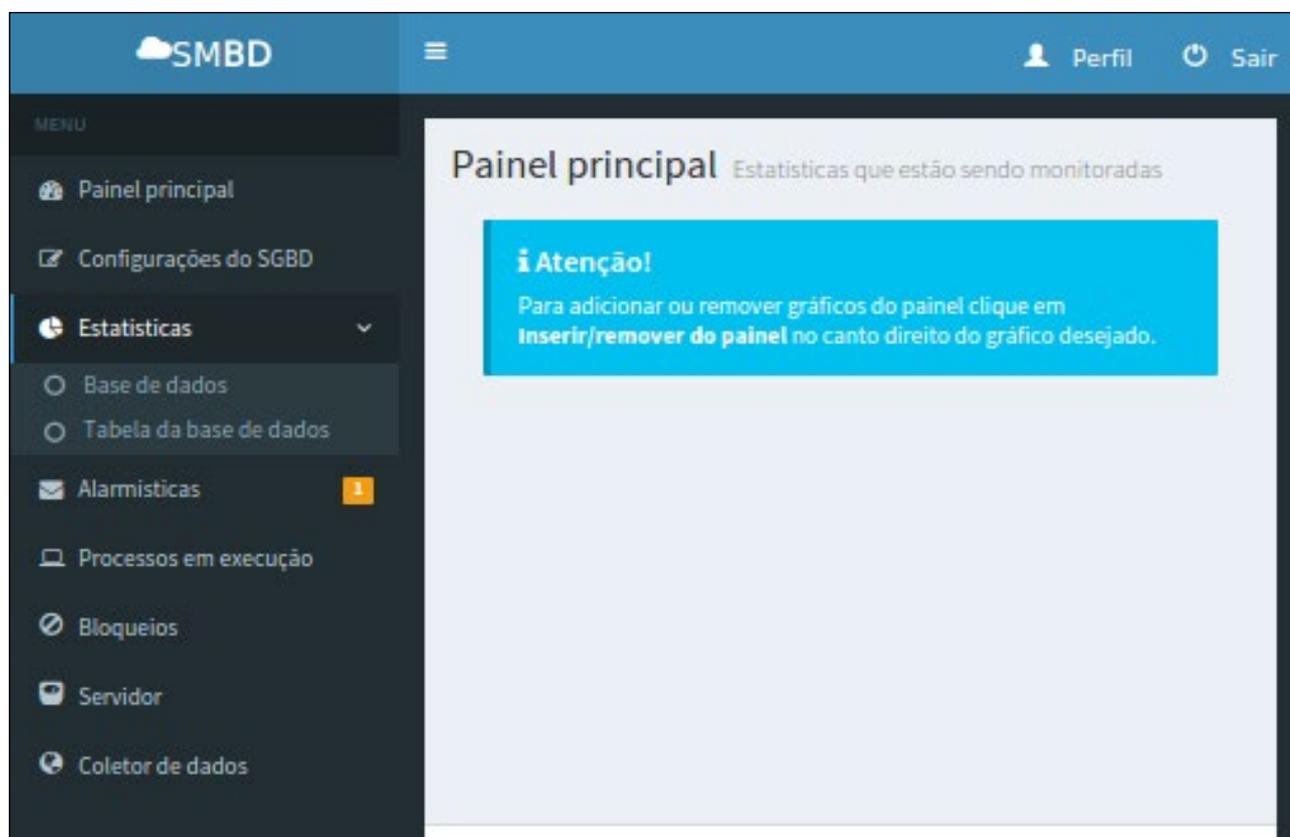
```
SELECT round(AVG(
(heap_blks_hit + COALESCE(toast_blks_hit, 0))::numeric /
(heap_blks_read + COALESCE(toast_blks_read, 0) +
heap_blks_hit + COALESCE(toast_blks_hit,0))::numeric * 100), 3) as hit_ratio
FROM pg_statio_user_tables
WHERE (heap_blks_read + COALESCE(toast_blks_read, 0)) + heap_blks_hit + COALESCE(toast_blks_hit)
> 0) as hit_ratio,
pg_size_pretty(pg_database_size(pg_stat_database.datname)) as tamanho_base_de_dados_formatado,
pg_database_size(pg_stat_database.datname) as tamanho_base_de_dados
FROM pg_stat_database
INNER JOIN pg_stat_database_conflicts
ON pg_stat_database.datname = pg_stat_database_conflicts.datname
WHERE pg_stat_database.datname= '$bdNome'
```

Fonte: Dos autores.

As demais consultas envolvem a busca de informações sobre as tabelas, os índices, as configurações do banco de dados e do servidor (carga de trabalho, memória), além de dados relacionados a transações, bloqueios e outras atividades em andamento no SGBD.

Depois ou durante a extração de dados, o resultado pode ser consultado por meio de uma ferramenta em que as informações são integradas. Por meio dela o DBA pode identificar, de forma visual, possíveis problemas de desempenho em relação ao SGBD, podendo, assim, tornar visível a necessidade de criação de índices e alterações em configurações do sistema utilizado. O módulo de monitoramento pode ser utilizado para outros SGBDs, desde que seja implementado o módulo coletor específico para o sistema gerenciador escolhido. A Figura 4 mostra a interface principal da aplicação.

Figura 4 – Interface do módulo de monitoramento



Fonte: Dos autores.

É possível visualizar os seguintes recursos:

- painel principal: a partir dessa interface, o DBA pode visualizar as estatísticas que estão sendo monitoradas. Para monitorar uma estatística, o usuário deve clicar em publicar no canto superior direito do *dashbord* desejado;
- configurações do SGBD: a partir dessa interface, o DBA pode visualizar as configurações do SGBD da base de dados monitorada;
- estatísticas: esse item de menu possui duas opções: base de dados e tabela da base de dados. Para visualizar os *dashboards* relacionados à base de dados, deve ser selecionada a primeira opção. Se o DBA desejar visualizar alguma informação específica de uma tabela, deve ser selecionada a segunda opção;
- alarmísticas: por meio dessa interface, o usuário pode visualizar os alertas identificados pela ferramenta;
- menu processos em execução: interface que possibilita visualizar todas as tarefas que estão sendo executadas no banco de dados monitorado. Para melhor eficácia dessa funcionalidade, o tempo de coleta dessa estatística deve ser pequena;
- bloqueios: por meio dessa funcionalidade, o DBA pode visualizar todos os processos que estão efetuando bloqueios na base de dados;

- servidor: por meio dessa interface, o usuário pode visualizar os gráficos de uso de memória e carga do servidor do SGBD do banco de dados monitorado;
- coletor de dados: por meio dessa ação, o DBA pode baixar o módulo coletor para instalar em um servidor de banco de dados que deseja monitorar.

#### 4.2 Avaliação da ferramenta em um ambiente de produção

A base de dados monitorada armazena informações de um *software* para gestão de ensino superior que contém cerca de 806 tabelas. A instituição utiliza o sistema desde 2007, portanto, contém um volume considerável de dados e já sofre com alguns problemas de performance em processos que envolvem tabelas com maior quantidade de registros.

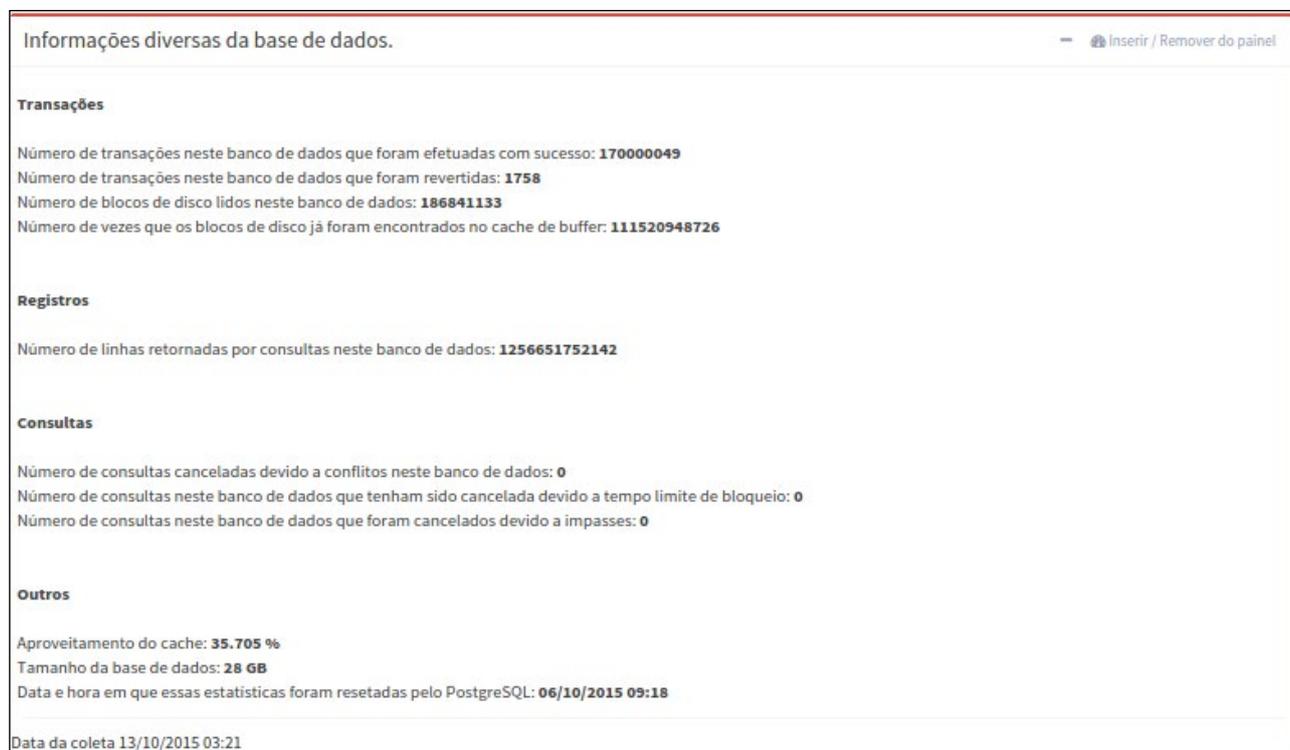
Com o objetivo de facilitar a configuração de acesso à base de dados monitorada o agente coletor foi instalado na mesma máquina em que o SGBD se encontra. Após a instalação, o mesmo foi colocado em execução durante 24 horas e foram feitas coletas de todas as estatísticas previstas pelo coletor.

Para haver bom acompanhamento dos processos executados na base de dados, bloqueios, carga e uso de memória do servidor, foram feitas coletas no intervalo de um minuto. Já para informações que não variam tanto, como informações da base de dados, estatísticas de índices, tabelas, versão do servidor e configurações do SGBD, as coletas tiveram intervalo configurado de 200 minutos.

Com a configuração apresentada, foi possível obter bom volume de dados e identificar alguns problemas da base monitorada. Nas próximas seções são apresentados dados relativos a essas coletas e os recursos disponibilizados pela ferramenta por meio de um ambiente real. Dentre os recursos serão vistos os *dashboards* disponibilizados em cada item de menu da ferramenta. A seguir são apresentados e explicados alguns dos painéis e as informações que constam neles.

A Figura 5 apresenta informações gerais da base de dados monitorada. Com base nas informações coletadas com essa funcionalidade, foi possível identificar que as coletas de estatísticas foram iniciadas pelo SGBD no dia 06/10/2015. Consequentemente, todas as informações disponíveis nesse painel e nos demais que serão apresentados nas próximas seções foram iniciadas nessa data. Foi possível também identificar que a base de dados monitorada tem tamanho total de 28 GB. Por meio desse mesmo painel o usuário DBA pode consultar informações relacionadas ao número de transações efetivadas no banco de dados, ao número de conflitos e ao percentual do uso do cache.

Figura 5 - Informações da base de dados



Fonte: Dos autores.

Além das informações gerais, o sistema mostra também a distribuição das tabelas no banco de dados, com relação ao seu tamanho. Os gráficos mostrados na Figura 6 exibem essa distribuição considerando apenas o tamanho da tabela (a) e a tabela mais os índices existentes.

Figura 6 – Tamanho das tabelas na base de dados



Fonte: Dos autores.

Outras informações sobre tabelas e índices indicam, por exemplo, o percentual de consultas feitas sobre a tabela, que usaram índices, o que permite ao administrador verificar se os índices existentes são ou não efetivos. Caso uma tabela apresenta tamanho maior em função dos índices e apresentar baixo percentual de pesquisas com índice, o DBA pode optar por excluir um ou mais índices existentes.

Conhecer a instruções executadas na base e saber quais têm desempenho insatisfatório são igualmente importantes para que o administrador do banco de dados possa dar atenção especial às instruções citadas, ou seja, tentar otimizá-las ou refatorá-las. Tomando essa ação, pode-se ter ganho considerável no desempenho do sistema. A Figura 7 exibe a lista das instruções com pior desempenho e os tempos correspondentes.

Figura 7 – Instruções que apresentam pior desempenho

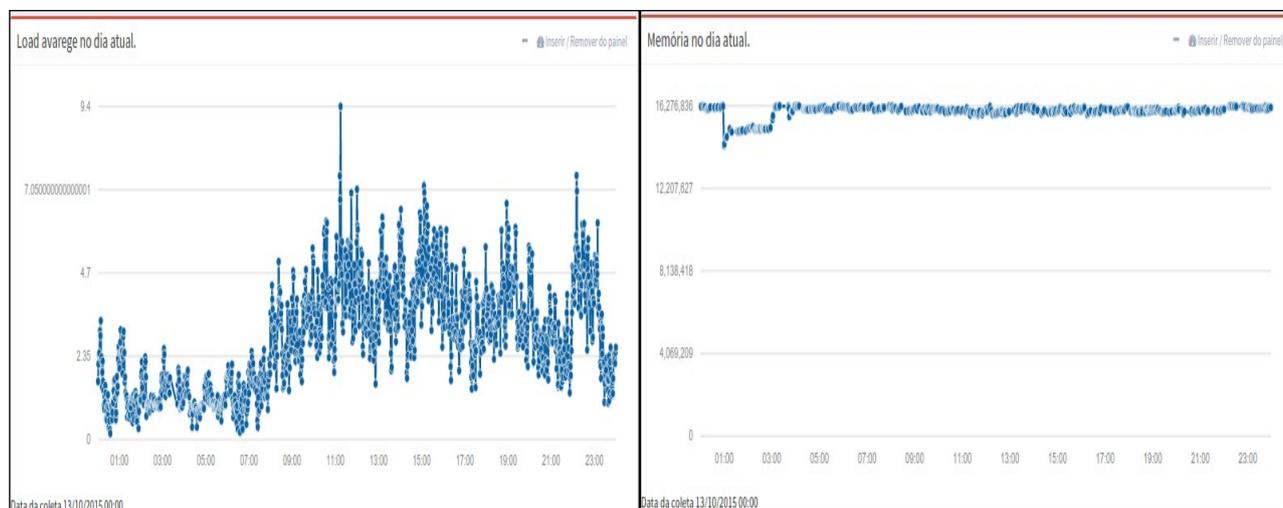
| Comandos mais lentos do dia.   |          | Inserir / Remover do painel |
|--|----------|-----------------------------|
| Comando  |          | Tempo de execução           |
| VACUUM FULL VERBOSE ANALYZE  |          | 00:46:17                    |
| SELECT array_agg(UPPER(replace(convert_from(convert_to(replace(replace(replace(replace(replace(replace(X.complementarybibliography, 'A', ''), 'A', ''), 'A&AM', 'XX'), 'A (\u0080\u0093', '')), 'AM', 'ÖE'), 'LATIN1'), 'UTF8'), 'XX', 'ÇÓ'))) AS complementarybibliography FROM (SELECT acdGroup.groupId, UNNEST(complementarybibliography) AS complementarybibliography FROM acdGroup INNER JOIN acdLearningPeriod ON acdLearningPeriod.learningPeriodId = acdGroup.learningPeriodId WHERE acdLearningPeriod.periodId = '2015/2' AND array_to_string(complementarybibliography, ',') <> '' AND groupId = 27269 ORDER BY groupId ) X;   | 00:29:28 |                             |
| SELECT acdGroup.groupId, UNNEST(complementarybibliography) AS complementarybibliography FROM acdGroup INNER JOIN acdLearningPeriod ON acdLearningPeriod.learningPeriodId = acdGroup.learningPeriodId WHERE acdLearningPeriod.periodId = '2015/2' AND array_to_string(complementarybibliography, ',') <> '' AND groupId = 27269 ORDER BY groupId ;  | 00:27:22 |                             |
| SELECT acdGroup.groupId, UNNEST(complementarybibliography) AS complementarybibliography FROM acdGroup INNER JOIN acdLearningPeriod ON acdLearningPeriod.learningPeriodId = acdGroup.learningPeriodId WHERE acdLearningPeriod.periodId = '2015/2' AND array_to_string(complementarybibliography, ',') <> '' AND groupId = 27269 ORDER BY groupId limit 1 offset 28;   | 00:24:13 |                             |
| INSERT INTO bi_rematricula SELECT COALESCE(SUM(X.matriculados), 0) AS matriculados, COALESCE(SUM(X.renovacoes), 0) AS renovacoes , COALESCE(SUM(X.reingressos), 0) AS reingressos, COALESCE(SUM(X.transferencia_de), 0) AS transferencia_de, COALESCE(SUM(X.ndas_matriculados), 0) AS ndas_matriculados, COALESCE(SUM(X.portadores_diploma), 0) AS portadores_diploma, COALESCE(SUM(X.nao_renovados), 0) AS nao_renovados, COALESCE(SUM(X.prematriculados), 0) AS pre_matriculados FROM ( SELECT codigo_curso AS courseid, nome_curso AS NAME, codigo_turno AS turnid, descricao_turno AS description, (CASE WHEN situacao_no_periodo = 'Pré-matriculado' THEN COUNT(codigo_contrato) ELSE 0 END) AS prematriculados, (CASE WHEN situacao_no_periodo = 'Portador de diploma' THEN COUNT(codigo_contrato) ELSE 0 END) AS portadores_diploma, (CASE WHEN situacao_no_periodo = 'Vestibulando' THEN COUNT(codigo_contrato) ELSE 0 END) AS matriculados, (CASE WHEN situacao_no_periodo = 'Renovado' THEN COUNT(codigo_contrato) | 00:21:31 |                             |

Data da coleta 13/10/2015

Fonte: Dos autores.

Estatísticas relacionadas ao servidor incluem a carga de trabalho e o consumo de memória durante o período de coleta de dados. Os gráficos mostrados na Figura 8 indicam a evolução da carga e o uso da memória ao longo do tempo.

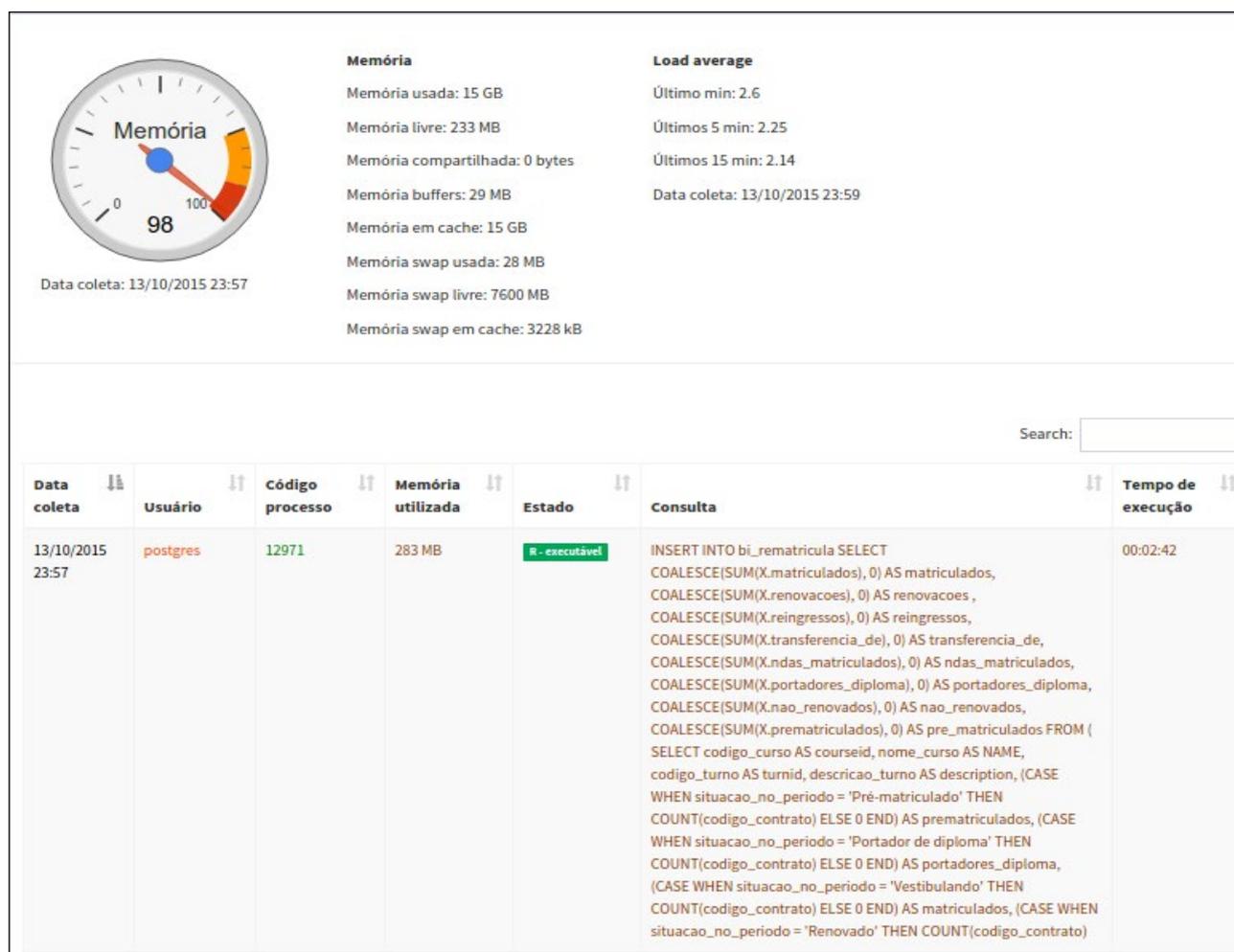
Figura 8 – Evolução da carga de trabalho e uso da memória do servidor



Fonte: Dos autores.

Em um sistema de banco de dados, vários processos podem estar em execução ao mesmo tempo. Dessa maneira, é possível que um em específico possa estar onerando a execução de uma porção de outros comandos. É importante que o DBA visualize todos esses comandos e o tempo que eles estão sendo executados para que possa tomar alguma ação caso a carga do servidor estiver muito elevada.

Figura 9 – Acompanhamento da carga e instruções mais executadas



Fonte: Dos autores.

Por meio do painel apresentado na Figura 9, o DBA pode visualizar os comandos que estão em execução na base de dados monitorada. Na mesma interface é possível acompanhar o uso de memória e a carga da máquina na última coleta. O painel apresentado é recarregado automaticamente a cada minuto para que as informações que estão sendo exibidas estejam sempre sincronizadas com a última coleta. No instante da captura da imagem foi possível identificar que um comando do tipo INSERT estava em execução há dois minutos e 42 segundos e estava utilizando 283 MB de memória do servidor.

Analisando os resultados obtidos, verifica-se que o uso do SMBD facilitou o monitoramento de uma base de dados real, visto que foi possível identificar momentos em que o servidor esteve sobrecarregado, consultas que tiveram execução lenta, tabelas que tiveram poucas consultas ou que utilizaram índices e identificação de índices não utilizados.

Com base nos resultados obtidos no ambiente real foram identificadas algumas ações que podem ser tomadas pelo DBA para melhorar o desempenho e evitar problemas no SGBD monitorado:

- efetuar manutenções no banco de dados no horário sugerido pelo SGBD a partir das alarmísticas;
- refatorar ou identificar ponto de melhoria nas consultas que tiveram execução lenta;
- considerar a possibilidade de criar índices nas tabelas que tiveram poucas consultas que utilizaram índices;
- avaliar a possibilidade de deletar os índices que não estão sendo utilizados pela base de dados.

## 5 CONSIDERAÇÕES FINAIS

Por meio dos estudos realizados, foi possível verificar a importância que o monitoramento e a análise das estatísticas disponibilizadas pelo SGBD empregam no funcionamento ideal e na performance de um banco de dados. Para evitar a perda de informação e garantir que os dados possam ser acessados de maneira eficaz, empresas procuram cada vez mais profissionais que tenham conhecimento em sistemas de banco de dados.

A ferramenta desenvolvida demonstrou ser capaz de auxiliar o profissional a perceber diversas situações, como identificar pontos de melhoria em consultas que tiveram execução lenta, possibilidade de criação de índices em tabelas que tiveram poucas consultas utilizaram esse recurso.

Após a conclusão do desenvolvimento, foi possível validar o sistema em um ambiente em produção. Ficou evidenciado que a mesma pode ser usada em cenários reais e que cumpriu com o objetivo de disponibilizar uma ferramenta para auxiliar na administração, no monitoramento e na otimização de um banco de dados que tenha grande volume de informação armazenada.

O painel gerenciador das informações coletadas permite visualizar o monitoramento de apenas uma base de dados para cada conta de usuário. Em trabalhos futuros deseja-se que o mesmo usuário possa acompanhar os dados coletados de mais de uma base e efetuar comparativos entre elas. Ainda com o objetivo de tornar o aplicativo mais proativo, podem ser implementados novos alarmes que sugerem formas de resolver problemas automaticamente identificados.

A coleta de estatísticas por meio de rotinas automatizadas suportou apenas o sistema gerenciador de banco de dados PostgreSQL, possibilitando a evolução da ferramenta com desenvolvimento de outros módulos coletores que permitam o monitoramento de outros SGBDs em trabalhos vindouros. Cabe ressaltar aqui que este trabalho é parte de uma iniciativa maior, com o objetivo de criar uma ferramenta que possa monitorar qualquer SGBD.

## REFERÊNCIAS

BERKUS, J. Checklist de Performance do PostgreSQL 8.0. **PostgreSQL WIKI**, 2005. Disponível em: <[https://wiki.postgresql.org/wiki/Checklist\\_de\\_Performance\\_do\\_PostgreSQL\\_8.0](https://wiki.postgresql.org/wiki/Checklist_de_Performance_do_PostgreSQL_8.0)>. Acesso em: 06 mai. 2015.

BIAZUS, D.O. PostgreSQL. **PostgreSQL WIKI**, 2003. Disponível em: <[https://wiki.postgresql.org/wiki/Introdu%C3%A7%C3%A3o\\_e\\_Hist%C3%B3rico](https://wiki.postgresql.org/wiki/Introdu%C3%A7%C3%A3o_e_Hist%C3%B3rico)>. Acesso em: 03 mai. 2015.

BLUMM, C; FORNARI, M. R. Dúvidas frequentes sobre Banco de Dados. **Revista SQL Magazine**, 28. ed., 2006.

BORELLO, F. ; KNEIPP, R. E. Álgebra Relacional. **Revista SQL Magazine**, p. 1 - 1, 2008.

CEDRUS. Cedrus: PostgreSQL Manager. Disponível em: <<http://sourceforge.net/projects/cedrus/>>. Acesso em: 19 mai. 2015.

CYBERTEC. Pgwatch Cybertec Enterprise PostgreSQL Monitor. Disponível em: <[http://www.cybertec.at/postgresql\\_produkte/pgwatch-cybertec-enterprise-postgresql-monitor/](http://www.cybertec.at/postgresql_produkte/pgwatch-cybertec-enterprise-postgresql-monitor/)>. Acesso em: 19 mai. 2015.

DATE, C. J. **Introdução a sistemas de bancos de dados**. 8.ed. Rio de Janeiro: Campus, 2004.

ELMASRI, R; NAVATHE, S. B. **Sistemas de banco de dados**. 6. ed. Pearson, 2011.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002.

MILANI, A. **PostgreSQL: Guia do Programador**. São Paulo: Novatec Editora, 2008.

ORACLE. MySQL Enterprise Monitor. Disponível em: <<https://www.mysql.com/products/enterprise/monitor.html>>. Acesso em 19 maio. 2015.

POSTGRESQL. Documentation. Disponível em <<http://www.postgresql.org/docs/manuals/archive/>>. Acesso em: 18 abr. 2015.

SANTOS, A. R. **Metodologia científica: a construção do conhecimento**. 2. ed. Rio de Janeiro: DP&A editora, 1999.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistemas de Bancos de Dados**. 3. ed. Makron Books, 1999.

SILBERSCHATZ, A.; KORTH, H.F.; SUDARSHAN, S; **Sistemas de Bancos de Dados**. 6. ed. Makron Books, 2012.

SMANIOTO, C. E. PostgreSQL. **Revista SQL Magazine**, 37. ed., 2007.

RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de Gerenciamento de Banco de Dados**. 3. ed. Mc Graw Hill, 2008.

ROB, P.; CORONEL, C. **Sistemas de Banco de Dados: Projeto, implementação e administração**. 8. ed. CENGATE Learning, 2011.