

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS DOIS VIZINHOS
CURSO DE ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS

PEDRO HENRIQUE DE ANDRADE GOMES

**OTIMIZAÇÃO DE BASE DE DADOS DE SISTEMA DE
TRAMITAÇÃO DE PROCESSOS GOVERNAMENTAIS**

TRABALHO DE CONCLUSÃO DE CURSO DE ESPECIALIZAÇÃO

DOIS VIZINHOS
2022

PEDRO HENRIQUE DE ANDRADE GOMES

OTIMIZAÇÃO DE BASE DE DADOS DE SISTEMA DE TRAMITAÇÃO DE PROCESSOS GOVERNAMENTAIS

Trabalho de Conclusão de Curso de Especialização apresentado ao Curso de Especialização em Ciência de Dados da Universidade Tecnológica Federal do Paraná, como requisito para a obtenção do título de Especialista em Ciência de Dados.

Orientador: Prof. Dr. Rafael Alves Paes Oliveira

DOIS VIZINHOS
2022



4.0 Internacional

Esta licença permite remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

PEDRO HENRIQUE DE ANDRADE GOMES

OTIMIZAÇÃO DE BASE DE DADOS DE SISTEMA DE TRAMITAÇÃO DE PROCESSOS GOVERNAMENTAIS

Trabalho de Conclusão de Curso de Especialização apresentado ao Curso de Especialização em Ciência de Dados da Universidade Tecnológica Federal do Paraná, como requisito para a obtenção do título de Especialista em Ciência de Dados.

Data de aprovação: 28 de dezembro de 2022

Rafael Alves Paes Oliveira
Doutorado
Universidade Tecnológica Federal do Paraná - Câmpus Dois Vizinhos

Yuri Kazubowski Lopes
Doutorado
Universidade do Estado de Santa Catarina

Francisco Carlos Monteiro Souza
Doutorado
Universidade Tecnológica Federal do Paraná - Campus Dois Vizinhos

DOIS VIZINHOS
2022

RESUMO

Ambientes de processamento de transações online (OLTP), ou ambientes transacionais, são sistemas de gerenciamento de bancos de dados geralmente projetados para otimizar o processamento de transações curtas e em tempo real em que os dados são alterados constantemente. Estas soluções são costumeiramente muito bem projetadas para a recuperação de toda a informação referente a um indivíduo, porém, podem ter perda considerável de desempenho na recuperação baseada em agrupamentos. Neste trabalho, é apresentada uma prova de conceito para a resolução de problemas de recuperação da informação em um conhecido sistema de tramitação de processos governamentais. São apresentadas propostas fundamentadas, assim como o resultado das implantações das mesmas em ambiente de produção, trazendo como ganhos um melhor desempenho do banco de dados, melhoria na experiência dos usuários e economia de recursos computacionais.

Palavras-chave: Bancos de Dados Transacionais; Sistema Eletrônico de Informações (SEI!); Otimização; Recuperação de Conteúdo.

ABSTRACT

On-line Transaction Processing (OLTP) environments, or transactional environments, are database management systems generally designed to optimize the processing of short, real-time transactions where data is constantly changing. These solutions are usually very well designed for the recovery of all information referring to an individual, however, they can have considerable loss of performance in the recovery based on clusters. In this work, a proof of concept is presented for the resolution of information retrieval problems in a well-known governmental process processing system. Substantiated proposals are presented, as well as the result of their deployments in a production environment, bringing gains: better database performance, improved user experience and savings in computational resources.

Keywords: Transactional Databases; Sistema Eletrônico de Informações (SEI!); Tunning; Content Retrieval.

LISTA DE FIGURAS

Figura 1 – Disponibilidade da Aplicação – Destaque para o período de maior instabilidade.	8
Figura 2 – Síntese dos desafios enfrentados e objetivos pretendidos agrupados por etapa.	10
Figura 3 – Ações por período (à esquerda) e Erros registrados por período (à direita) – destaque para o período das semanas 5 à 11 de 2022.	11
Figura 4 – Exemplo de trecho de código que gera as sequências para os IDs dos relacionamentos.	12
Figura 5 – Exemplo de trecho de código que gera as sequências para os IDs dos relacionamentos.	13
Figura 6 – Ações realizadas por semana e Tendência Polinomial de Ordem 2	14
Figura 7 – Interface da Solução DBADash para três dias de execução do SEI.	15
Figura 8 – Interface da Solução sp_Blitz®	15
Figura 9 – Diagnósticos identificados pela solução SQL Alerts	16
Figura 10 – Exemplo de Relatório da Ferramenta SQL Alerts	16
Figura 11 – Exemplificação da solução aplicada para diminuição do tempo de bloqueio.	19
Figura 12 – Resultado da solução aplicada para diminuição do tempo de bloqueio. . . .	20
Figura 13 – Demonstração dos níveis de Isolamento para tratamento de Bloqueios. . . .	21
Figura 14 – Configuração apresentada pela ferramenta <i>SqlServer Management Studio</i> .	22
Figura 15 – Diminuição do tempo de espera <i>CXPacket</i> após alteração do limiar de custo para paralelismo	22
Figura 16 – Exemplo de consulta após criação de índice sugerido.	23

SUMÁRIO

1	INTRODUÇÃO	7
1.1	Problema de Pesquisa	7
1.2	Objetivos	7
1.2.1	Objetivo Geral	7
1.2.2	Objetivos Específicos	8
1.3	Justificativa	8
1.4	Organização do Trabalho	9
2	IMPLANTAÇÃO DA CULTURA DE OTIMIZAÇÃO	10
2.1	Estabilizar	10
2.1.1	Priorização	11
2.1.2	Análise	11
2.1.2.1	Geração das Sequências	12
2.1.2.2	Geração das Sequências Auxiliares	12
2.1.3	Movimentação	12
2.1.4	Exclusão	13
2.2	Monitorar	14
2.2.1	DBADash	14
2.2.2	<i>sp_Blitz</i> ®	15
2.2.3	SQL Alerts	16
2.3	Aperfeiçoar	17
3	RESULTADOS	18
3.1	Materiais e Métodos	18
3.2	Bloqueios	18
3.2.1	Bloqueios nas Tabelas de Geração de Sequências	19
3.2.2	Bloqueios na tabela de Geração de Sequência Auxiliares	19
3.2.3	Bloqueios nas Demais Estruturas	21
3.3	Paralelismo	21
3.4	Criação de Índices	22
4	CONSIDERAÇÕES FINAIS	24
4.1	Limitações	24
4.2	Trabalhos Futuros	24
	REFERÊNCIAS	25

1 INTRODUÇÃO

OLTP, do inglês “*On-line Transaction Processing*”, é o termo usado para se referir aos sistemas transacionais, ou seja, os sistemas operacionais das organizações. São utilizados no processamento dos dados de rotina que são gerados diariamente através dos sistemas informacionais da empresa e dão suporte às funções de execução do negócio organizacional.

Um dos pilares do Programa Eletrônico Nacional ([MINISTÉRIO DA ECONOMIA, 2022a](#)), o Sistema Eletrônico de Informações (SEI) é uma ferramenta de gestão de documentos e processos eletrônicos, e tem como objetivo promover a eficiência administrativa, sem a necessidade de elaboração de documentos em papel ([MINISTÉRIO DA ECONOMIA, 2022b](#)).

Sendo diretamente relacionada ao nível operacional e viabilidade do negócio nas instituições, e com um enorme volume transacional, associado a estrutura relacional e volatilidade dos dados, tornam esta ferramenta a principal ferramenta OLTP na grande maioria das repartições públicas que a adotaram.

1.1 Problema de Pesquisa

A arquitetura ao qual a ferramenta SEI! foi desenvolvida, associada ao constante mau uso por parte de seus utilizadores, faz com que a tarefa de manutenção da solução se torne um desafio para os administradores e gestores de bancos de dados.

Embora idealizada para atender uma repartição que opera em horário comercial, esta solução é adotada por diversas repartições que possuem regime de trabalho por escala, como Segurança Pública e Saúde, dificultando a tarefa de manutenção da solução, pois não possibilita a execução de ações, que podem ocasionar em diversos bloqueios (*locks*) ou alto tempo de espera (*wait time*), a nível de banco de dados.

Ambas as situações (bloqueios e espera) são previstas e possuem tratativas pelo próprio sistema gerenciador de banco de dados (SGBD). Ocorre que em situações extremas, o SGBD não possui recursos computacionais suficientes para o número de tratativas necessárias, aumentando a fila de processamento, o que gera grande tempo de espera entre a solicitação do usuário e a resposta, podendo em alguns casos, afetar até a disponibilidade da aplicação.

1.2 Objetivos

A seguir são apresentados os objetivos Geral e Específicos para este trabalho.

1.2.1 Objetivo Geral

Espera-se com este trabalho levantar e explorar um conjunto de ferramentas, processos e métodos que apoiem a recuperação da informação de maneira sustentável e que possibilite

a otimização e refinamento dos processos de recuperação de conteúdo em grandes bases de dados transacionais.

1.2.2 Objetivos Específicos

São considerados objetivos específicos deste trabalho:

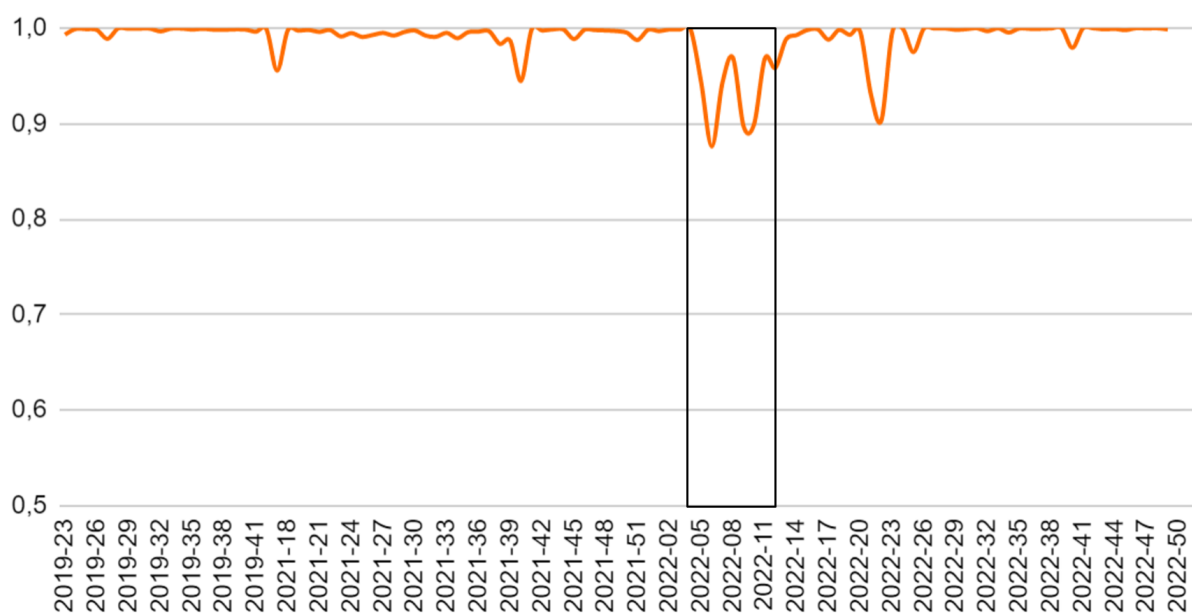
- Demonstrar ferramentas de código-fonte aberto e que podem auxiliar administradores e gestores de bancos de dados;
- Apresentar as soluções adotadas para problemas conhecidos e enfrentados pelos responsáveis das bases de dados que hospedam a aplicação SEI! em diversas repartições;

1.3 Justificativa

A proposta de trabalho veio de uma necessidade do próprio Órgão de Tecnologia da Informação e Comunicação em que o autor trabalha, e da qual tem enfrentado desafios no gerenciamento da base de dados do principal sistema da repartição. A motivação para implantar melhorias também está associada a criticidade da aplicação para o negócio.

A ferramenta SEI! é utilizada, dentre outras coisas, para realizar publicações de decretos, portarias, instruções normativas e editais no Diário Oficial. Também é utilizada para tramitação interna, sendo fundamental para a correta execução orçamentária, gestão de pessoas, etc.

Figura 1 – Disponibilidade da Aplicação – Destaque para o período de maior instabilidade.



Fonte: Do autor.

A Figura 1 apresenta o histórico de disponibilidade da aplicação, onde 1,0 é o valor máximo (100%), sendo possível notar eventuais períodos de queda na estabilidade da aplicação.

1.4 Organização do Trabalho

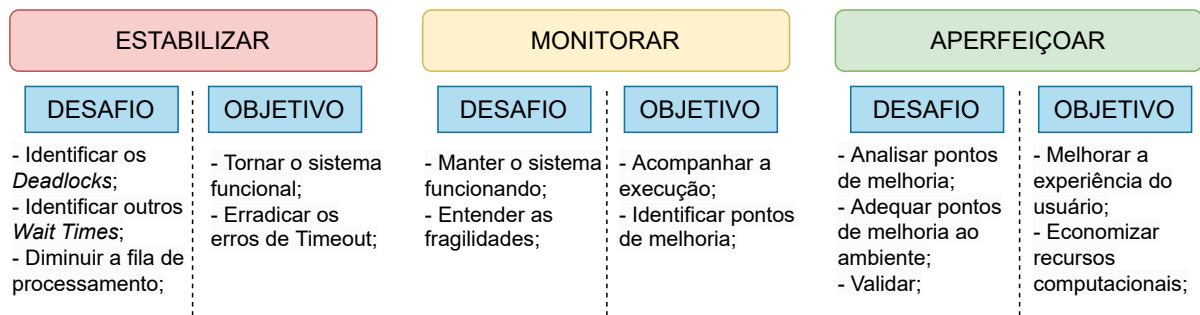
No [Capítulo 1](#) é apresentada uma breve introdução sobre os objetivos deste trabalho, bem como a explicação da abordagem proposta para o Trabalho de Conclusão de Curso. No [Capítulo 2](#), é apresentada a aplicação da abordagem proposta. O [Capítulo 3](#) trás os principais resultados do trabalho. Ao final, são apresentadas algumas Considerações sobre o trabalho no [Capítulo 4](#).

2 IMPLANTAÇÃO DA CULTURA DE OTIMIZAÇÃO

Criar e implantar uma cultura dentro de uma organização pode não ser tarefa fácil. Para implantar uma cultura organizacional forte, é imprescindível a adoção de processos claros e bem definidos e ser transparente quanto aos objetivos.

A seguir, são apresentadas as soluções implantadas com o objetivo de fornecer ao usuário a melhor experiência na utilização do Sistema Eletrônico de Informação. Para que a aplicação se mantenha operacional, tornou-se necessária a adoção de uma cultura de otimização da base de dados. No processo de implantação, pode-se identificar três cenários advindos de momentos diferentes na perspectiva organizacional – Estabilizar, Monitorar e Aperfeiçoar – explicados a seguir.

Figura 2 – Síntese dos desafios enfrentados e objetivos pretendidos agrupados por etapa.



Fonte: Do Autor.

Na [Figura 2](#), são demonstradas as etapas e seus respectivos desafios e objetivo. As abordagens adotadas em cada etapa são demonstradas a seguir.

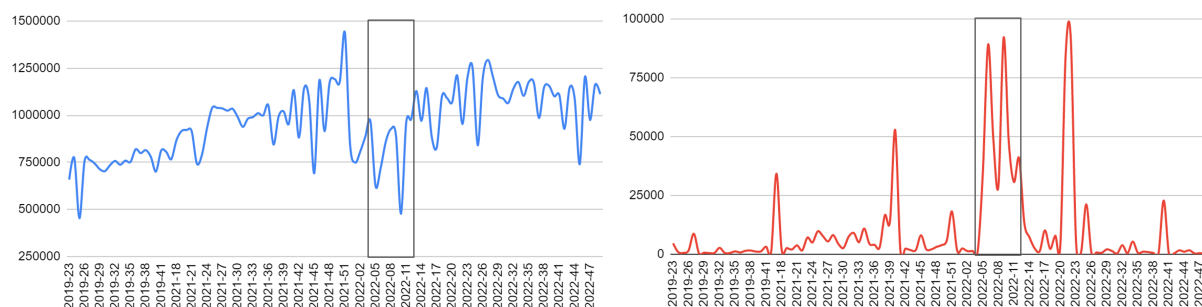
2.1 Estabilizar

O primeiro momento, indicado na [Figura 1](#) teve por objetivo manter a aplicação funcional, ou seja, lentidão era aceitável desde que não acarretasse em indisponibilidade total ou parcial do sistema. Este período ocorreu entre a semana 5 até a semana 11 do ano de 2022.

Diversos problemas enfrentados fizeram com que a taxa de disponibilidade da aplicação (que fica em cerca de 99,99% – com mais de 180 mil ações por dia) atingisse a taxa de disponibilidade de 87,7%, causando grandes transtornos aos usuários e sendo motivadores do início deste trabalho. Diversas ações foram realizadas, sendo o foco no saneamento dos dados, o expurgo.

Pode-se considerar o expurgo como uma das etapas comuns do processo de saneamento de dados ([SQL Magazine, 2010](#)). Neste contexto, o expurgo de dados é a remoção de dados antigos (informações históricas). Decisões de projeto e conseqüentemente da implementação

Figura 3 – Ações por período (à esquerda) e Erros registrados por período (à direita) – destaque para o período das semanas 5 à 11 de 2022.



(a) Ações realizadas por semana – com queda (b) Erros registrados por semana – com grande crescimento no período destacado.

Fonte: Do Autor.

do código-fonte, também podem acabar influenciando no volume de dados (muitas vezes desnecessários) armazenados nas bases.

Esta é uma problemática constantemente enfrentada pelas bases de dados do SEII, principalmente as hospedadas nos bancos de dados *Microsoft SQL Server*, explicado em detalhes na [Subseção 2.1.2](#). Em síntese, a aplicação gera uma quantidade imensa de dados das ações de usuário dentro da aplicação, fundamentais para a realização de auditorias, mas também dados de controle/temporários, como contadores e geradores de índices, considerados lixo, apenas para fins de controle da aplicação.

Para dar a destinação correta conforme o tipo de dado armazenado (auditoria e controle/temporários), esta ação foi dividida em quatro fases, sendo elas Priorização, Análise, Movimentação e Exclusão, explicadas a seguir.

2.1.1 Priorização

Ao analisar os registros do Monitor de Atividades da Base de Dados, identificou-se que um pequeno conjunto de ações no sistema prejudicava a grande maioria dos usuários. Neste sentido, definimos por priorizar pequenas ações na aplicação, matando todo processo do qual a consulta associada perdurava por tempo superior a um minuto.

Tornou-se possível realizar tarefas básicas, que envolviam principalmente a inserção de registros no banco, sendo consideravelmente frustrante ao usuário realizar buscas ou atividades mais complexas, pois essas ações ocasionalmente perduraram por mais de um minuto.

2.1.2 Análise

Após a identificação das consultas problemáticas (com maiores *wait-times*). O desafio passou a ser verificar a possibilidade de realizar alguma ação no banco de dados que pudessem compensar um possível problema de implementação da aplicação.

2.1.2.1 Geração das Sequências

Toda a estrutura de organização dos dados é baseada na modelagem relacional. Ao aplicarmos o modelo físico em ferramentas que permitem visualização do modelo lógico de banco de dados, identificamos potenciais problemas que foram confirmados após realização de análise minuciosa do código.

Figura 4 – Exemplo de trecho de código que gera as sequências para os IDs dos relacionamentos.

```
public function getValorSequencia($sequencia){
    $rs = $this->consultarSql('SELECT ' . $sequencia . '.NEXTVAL FROM DUAL');
    return $rs[0]['nextval'];
}
```

(a) Trecho de Código para Oracle.

```
public function getValorSequencia($sequencia) {
    $rs = $this->consultarSql('INSERT INTO '.$sequencia.
        ' OUTPUT CAST(INSERTED.id as VARCHAR) as \'id\' VALUES (null);');
    return $rs[0]['id'];
}
```

(b) Trecho de código para SQLServer.

Fonte: Do Autor.

A Figura 4 demonstra o trecho de código responsável pela ação *getValorSequencia*, para o SGBD da Oracle (Figura 4a) e da Microsoft (Figura 4b). É possível notar que além de ser bem mais oneroso, o trecho de código para SQL Server gera grande quantidade de lixo.

São mais de 100 tabelas, todas com *identity* ativado, nas quais a cada nova iteração no sistema, são realizadas inserções nessas tabelas, com o objetivo de atribuir o valor sequencial do objeto.

2.1.2.2 Geração das Sequências Auxiliares

Durante a análise, percebemos que a tabela *infra_sequencia* estava constantemente bloqueada. A estrutura da aplicação faz com que determinadas linhas desta tabela sejam constantemente atualizadas gerando bloqueios exclusivos (*Exclusive Lock*) e em alguns casos, o impasse (*deadlock*) acarretava em falha na transação (*timeout*).

Mesmo tendo menos de 20 linhas, uma simples consulta de seleção nessa tabela, em horário comercial, levava cerca de 5 minutos, com casos de até 15 minutos de espera.

2.1.3 Movimentação

Após estudo, foram identificadas tabelas das quais os registros seriam necessários apenas para fins de auditoria da aplicação. Foram desenvolvidas rotinas diárias de expurgo das quais moviam todos os registros deste cenário para uma base de dados separada da base

Figura 5 – Exemplo de trecho de código que gera as sequências para os IDs dos relacionamentos.

```
$sql = 'update infra_sequencia set num_atual = num_atual + qtd_incremento
| OUTPUT Inserted.num_atual where nome_tabela = ' . BancoInfra::getInstance()->formatarGravacaoStr($strTabela);
$rsProx = BancoInfra::getInstance()->consultarSql($sql);
```

(a) Trecho de Código para todos os SGBDs.

The screenshot shows a SQL query in a text editor and its results in a table. The query is:

```
1 /***** Script for SelectTopNRows command from SSMS *****/
2 SELECT * from infra_sequencia
3 order by 3 desc;
```

The results table is as follows:

	nome_tabela	qtd_incremento	num_atual	num_maximo
1	usuario_externo	1	1025435	9999999999
2	seq_2021_uni_sei_91	1	616556	9999999999
3	seq_2019_uni_sei_91	1	563838	9999999999

(b) Exemplo de dados persistidos na base de dados, resultados de operações realizadas na aplicação.

Fonte: Do Autor.

transacional. A criação dessa base de dados de expurgo permitiu a retirada de ambiente de produção de cerca de 500 Gb de dados, aproximadamente 1 bilhão de registros.

Tabela 1 – Resultado dos testes.

	TableName	RowCounts	TotalSpaceMB
Exemplo 1	auditoria_protocolo	329.881.188	51.544,15
Exemplo 2	seq_auditoria_protocolo	329.881.188	6.140,91

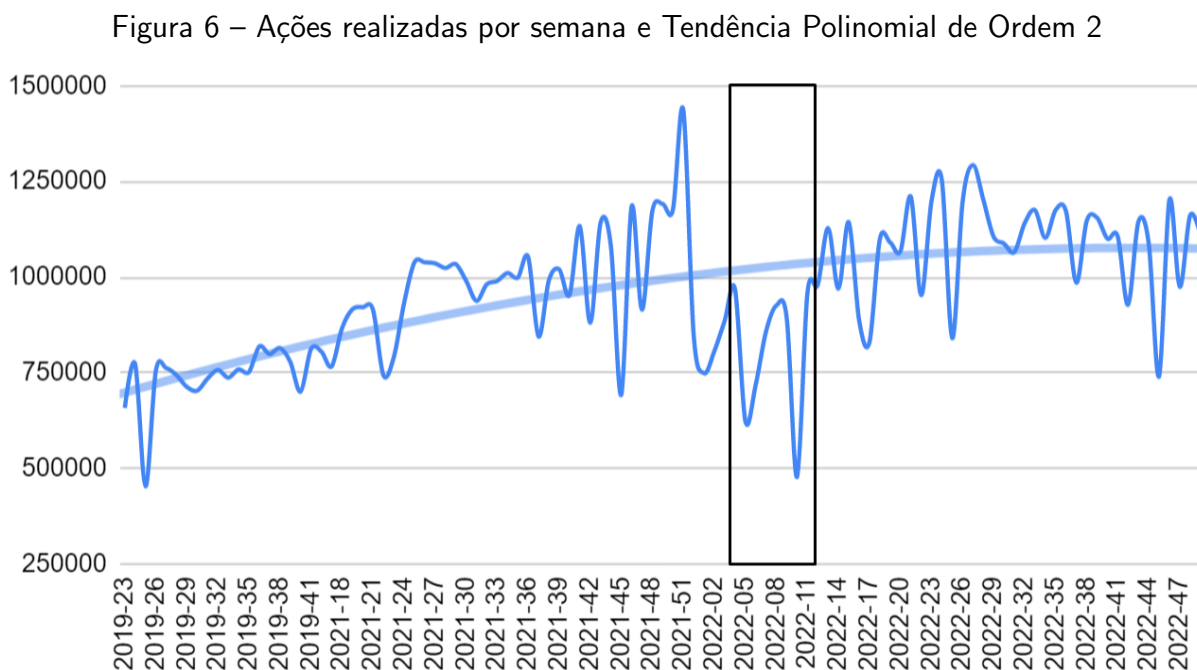
Fonte: Do Autor.

2.1.4 Exclusão

A parte mais delicada do processo, exclusão definitiva de todo registro que poderia ser considerado como temporário. A primeira execução permitiu a remoção definitiva (exclusão) de cerca de 750 milhões de registros, liberando aproximadamente 80 Gb de espaço em disco, sem prejuízo ao usuário.

Após avaliação, teste e homologação, foram criadas rotinas periodicamente executadas para exclusão de todos os registros de controle e que não afetam a execução da aplicação, além dos registros temporários, que não afetam a experiência do usuário.

Toda essa primeira fase foi realizada dentro do intervalo destacado na [Figura 6](#). A rápida execução se deu pela criticidade do sistema e prejuízos associados a indisponibilidade do mesmo. Ao considerar a projeção de tendência, é possível estimar que mais de um milhão de ações deixaram de ser executadas, causando prejuízo para o Estado.



Fonte: Do Autor.

Ao assegurar que os usuários conseguiram utilizar satisfatoriamente o sistema, o foco passou a ser o acompanhamento da base de dados, a partir da adoção de ferramentas de monitoramento, para garantir que o cenário anterior não ocorresse novamente.

2.2 Monitorar

Sanados os problemas que acarretavam na instabilidade do sistema, a etapa seguinte consistiu na consolidação das ações, com foco no monitoramento constante e adoção de ferramentas que apoiem tal objetivo.

Foram priorizadas ferramentas conhecidas no mercado, de fácil implantação e preferencialmente de código aberto. Dentre essas ferramentas, destacam-se as apresentadas abaixo.

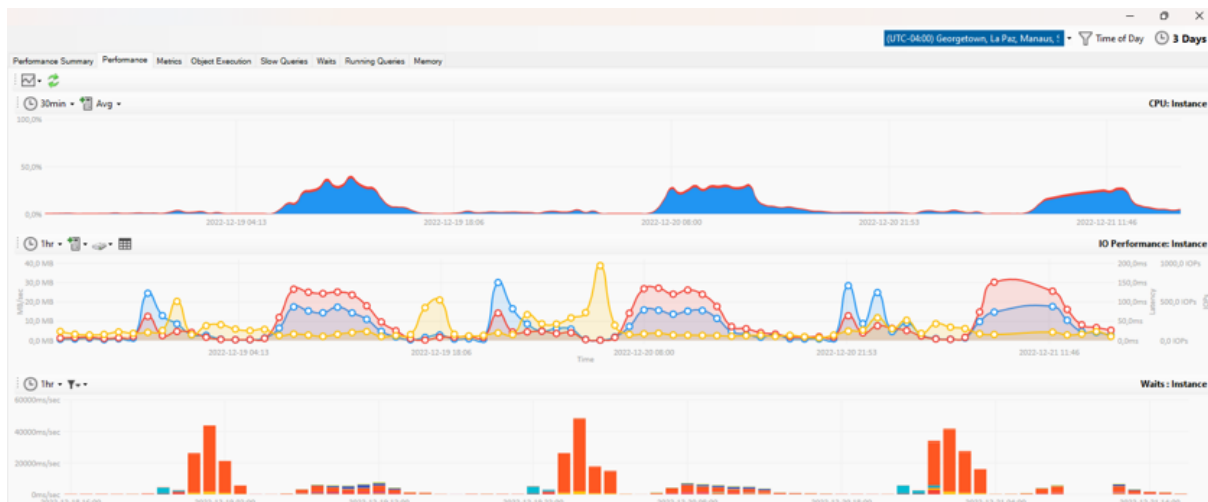
2.2.1 DBADash

Projetada com foco nas atividades de administradores de bases de dados (DBAs) a ferramenta DBADash apoia o DBA nas atividades de monitoramento da saúde das bases e rastreamento das atividades. Trata-se de uma ferramenta completamente gratuita e de código aberto similar a ferramentas consolidadas no mercado como SolarWinds, Power Tools e DataDog.

A ferramenta exibe em tempo real as informações do servidor de banco de dados, como utilização de processador, memória, rede e disco. Também é possível acompanhar a execução de consultas, procedimentos e agendamentos. Com esse acompanhamento, torna-se

de fácil identificação situações como lentidão, interrupções, bloqueios e até vazamento de memória e dados corrompidos.

Figura 7 – Interface da Solução DBADash para três dias de execução do SEI.



Fonte: Do Autor.

2.2.2 sp_Blitz®

Conforme definição do próprio idealizador ¹, sp_Blitz® é uma solução gratuita focada na verificação de integridade do SQL Server. Trata-se de um conjunto de *Stored Procedures* que identifica problemas comuns de integridade e desempenho.

Figura 8 – Interface da Solução sp_Blitz®

1 EXEC sp_Blitz @CheckUserDatabaseObjects = 0

Priority	FindingsGroup	Finding	DatabaseName	URL	Details
0	sp_Blitz Feb 22 ...	SQL Server First Responder Kit	NULL	http://FirstResponderKit.org/	To get help or add your own contributions, join
2	1 Backup	Backups Not Performed Recently	DBAtools	https://BrentOzar.com/go/nobak	Last backed up: May 30 2020 3:48AM
3	1 Backup	Backups Not Performed Recently	master	https://BrentOzar.com/go/nobak	Last backed up: never
4	1 Backup	Backups Not Performed Recently	model	https://BrentOzar.com/go/nobak	Last backed up: never
5	1 Backup	Backups Not Performed Recently	msdb	https://BrentOzar.com/go/nobak	Last backed up: never
6	1 Backup	Backups Not Performed Recently	StackOverflow	https://BrentOzar.com/go/nobak	Last backed up: never
7	1 Backup	Backups Not Performed Recently	StackOverflow2010	https://BrentOzar.com/go/nobak	Last backed up: Nov 27 2020 3:06AM
8	1 Backup	Backups Not Performed Recently	StackOverflow2013	https://BrentOzar.com/go/nobak	Last backed up: Jul 14 2020 11:14AM
9	10 Performance	Plan Cache Erased Recently	NULL	https://BrentOzar.com/askbrent/pl...	The oldest query in the plan cache was created
10	10 Reliability	Server restarted in last 24 hours	NULL		Surprise! Your server was last restarted on: Feb
11	20 File Configuration	TempDB on C Drive	tempdb	https://BrentOzar.com/go/cdrive	The tempdb database has files on the C drive.
12	20 Reliability	Memory Dumps Have Occurred	NULL	https://BrentOzar.com/go/dump	That ain't good. I've had 5 memory dumps betw

Fonte: Do Autor

¹ Disponível em: <<https://www.brentozar.com/blitz/documentation/>>

2.2.3 SQL Alerts

Baseada em soluções conhecidas no mercado como a **sp_Blitz**² e **sp_whoisactive**², a ferramenta *SQL Alerts*³ é um projeto nacional, criado em 2010 e amparado pela comunidade *open-source*, que compõe um conjunto de *Scripts* que permitem o monitoramento do SQL Server. Seu grande diferencial se comparado a ferramenta anterior, é que esta monitora de forma autônoma, em tempo real e ainda é capaz de enviar relatórios por e-mail, enquanto que a primeira funciona como um diagnóstico, sempre que executada.

Figura 9 – Diagnósticos identificados pela solução SQL Alerts



The screenshot shows the 'Alertas' (Alerts) interface of the SQL Alerts tool. It features a header with the logo of 'Fabrício Lima' and the title 'Alertas'. Below the header, there are four columns of alerts:

- Alertas Atuais** (Current Alerts): 1) Processo Bloqueado, 2) Arquivo de Log FULL, 3) Espaço em Disco, 4) Consumo de CPU, 5) Tempdb Utilização Arquivo, 6) Conexão SQL Server, 7) Status Database, 8) Banco de Dados Corrompido, 9) Página Corrompida, 10) Queries Demoradas, 11) Falha de JOB, 12) SQL Server Reiniciado, 13) Database Criada, 14) Database sem Backup, 15) Alertas de Severidade.
- Alertas Novos** (New Alerts): 16) Erros no Banco de Dados, 17) Lentidão de I/O Dia Anterior, 18) Lentidão de I/O Hora em Hora, 19) Fila de Disco Agora, 20) Processo executando a muito Tempo, 21) Crescimento Lento de Arquivos do BD, 22) Alertas que não foram Resolvidos, 23) Falha de Logins, 24) Bases Sem Backup do Log, 25) Baseline de Informações Mensal do BD, 26) Consumo de memória, 27) Jobs Desabilitados, 28) LDF Size 50% do MDF, 29) Falha de Rebuild por Lock já com o Motivo, 30) Deadlocks.
- Alertas Novos** (New Alerts): 31) CheckList Diário do BD (diário), 32) Processo Bloqueado 2 (mais de 20 min), 33) Crescimento Exagerado da Base, 34) MaxSize Arquivos, 35) Utilização de CPU no Managed Instance, 36) Monitora o Status do Database Mirroring, 37) Mudança do nó Ativo do Cluster, 38) Algum nó do Cluster não está UP, 39) Monitora o Status do AlwaysOn AG, 40) Índice Fragmentado por muito tempo.
- Alertas Futuros** (Future Alerts): 41) Estatísticas muito antigas? (Breve), 42) Degradação de Performance – CPU (Breve), 43) Degradação de Performance – Queries Demoradas (Breve), 44) Degradação de Performance – PLE (Breve), 45) Degradação de Performance – Disco (Breve).

Fonte: Disponível em: <https://github.com/soupowertuning/Script_SQLServer_Alerts>

Figura 10 – Exemplo de Relatório da Ferramenta SQL Alerts

Nome Alerta	Mensagem	Data
Jobs que Falharam nas últimas 24 Horas	ALERTA: Jobs que Falharam nas últimas 24 Horas no Servidor: FABRICIOLIMA\SQLSERVER2016	2017-03-19 06:50:00
Arquivo de Log Full	CLEAR: Não existe mais algum Arquivo de Log com mais de 85% de utilização no Servidor: FABRICIOLIMA\SQLSERVER2016	2017-03-19 03:00:30
Arquivo de Log Full	ALERTA: Existe algum Arquivo de Log com mais de 85% de utilização no Servidor: FABRICIOLIMA\SQLSERVER2016	2017-03-19 02:59:31

Fonte: <https://github.com/soupowertuning/Script_SQLServer_Alerts>

Como terceira parte do desafio, tanto no primeiro, quanto no segundo momento, foram identificados itens de melhoria que poderiam ser posteriormente adotados. Estes itens são explicados a seguir.

² Disponível em: <<http://whoisactive.com/>>

³ Disponível em: <https://github.com/soupowertuning/Script_SQLServer_Alerts>

2.3 Aperfeiçoar

Com a adoção das ferramentas *sp_Blitz* e *SQL Alerts*, identificou-se que diversos índices estavam com alta taxa de fragmentação. Com isso, faz-se necessária a execução de *Scripts* de Reorganização e Recriação de Índices, além da atualização das estatísticas.

Devido à quantidade de índices, nível de fragmentação e espaço ocupado pelos mesmos (cerca de 600 Gb na época da adoção) os procedimentos padrões de manutenção levavam um tempo muito grande para serem executados, sendo necessário abortar o processo na segunda-feira pela manhã (quando ocorria o aumento repentino da carga de consultas ao banco de dados). Após três tentativas frustradas de execução, tornou-se o desafio, procurar ferramentas que pudessem nos auxiliar nessa demanda.

Para tal, utilizamos a ferramenta SQL Server Maintenance Solution, gratuita e desenvolvida pela comunidade de DBAs chamada Hallengren⁴, constantemente referenciadas em fóruns de discussões. Com a ferramenta, conseguimos definir a seguinte rotina de manutenção:

- **MANUTENÇÃO - Rebuild ou Reorganize de Índices Fragmentados e Atualização completa de Estatísticas** → Agendado para ser executado toda sexta-feira às 21 horas. Composto de 3 passos, cito:
 - Faz um *Snapshot* da estrutura dos índices;
 - *Rebuild or Reorganize*;
 - Update completo de estatísticas;
- **MANUTENÇÃO - Update Somente Estatísticas Modificadas** → Agendado para ser executado todo dia às 21 horas. Composto apenas de 1 passo, onde faz a atualização das estatísticas onde ocorre modificação de registros;
- **DBA - Load Index Fragmentation** → Agendado para ser executado de segunda a sexta-feira às 02 horas da madrugada. Tira um “retrato” de toda a estrutura de fragmentação dos índices, para posterior acompanhamento da percentagem de fragmentação de cada índice.

⁴ Disponível em: <<https://ola.hallengren.com/>>

3 RESULTADOS

A seguir, são apresentados os materiais e métodos utilizados, assim como os principais resultados da aplicação da proposta.

3.1 Materiais e Métodos

Como insumo para a condução do trabalho, foi utilizada a base de dados do SEI!, de uma das repartições do poder executivo cooperadas com o TRF-4 e Ministério da Economia, e filiada ao Processo Eletrônico Nacional (PEN). A base de dados na aplicação, contava com cerca de 9Tb de dados em formato de texto e numérico. Todas as operações demonstradas foram executadas primeiro em ambiente de teste, a partir de um sistema gerenciador de banco de dados, *Microsoft SQL Server 2019 - Developer Edition*¹, instalado em um nó virtualizado na infraestrutura *Nutanix*², tendo a disposição os recursos de infraestrutura, processamento e memória necessários. Após validação e homologação, as soluções eram implantadas em ambiente de produção, também *Microsoft SQL Server 2019*. Os ambientes de aplicação também estão virtualizados e utilizam sistema operacional baseado em Unix, devidamente atualizado assim como as demais dependências. Assim como no ambiente do servidor de banco de dados, os recursos de infraestrutura, processamento e memória serão disponibilizados conforme necessidade. Por limitações do termo de cooperação, nenhuma atividade foi realizada nos ambientes de aplicação.

A seguir são apresentadas as soluções implantadas, agrupadas conforme problema identificado.

3.2 Bloqueios

O bloqueio é uma característica inevitável e por design de qualquer SGBD com simultaneidade baseada em bloqueio. Sua ocorrência se dá quando uma sessão mantém um bloqueio em um recurso específico e uma segunda transação tenta adquirir um tipo de bloqueio conflitantes no mesmo recurso. Normalmente, o período para o qual a primeira requisição bloqueia o recurso é pequeno. Quando a sessão de propriedade libera o bloqueio, a segunda conexão fica livre para adquirir seu próprio bloqueio no recurso e continuar o processamento. O bloqueio é um comportamento normal e pode acontecer muitas vezes ao longo de um dia sem nenhum efeito perceptível no desempenho do sistema.

Ocorre que devido a possíveis problemas de projeto, a estrutura a qual a aplicação foi desenvolvida realiza diversas transações de bloqueios, como exemplos, existem tabelas contadoras para Ids de chaves primárias e para numerações de documentos e processos. Nesse

¹ Disponível em: <<https://www.microsoft.com/pt-br/sql-server/sql-server-downloads>>

² Disponível em: <<https://www.nutanix.com/br>>

segundo caso, é realizado bloqueio a nível de linha, fazendo com que os usuários precisem aguardar a atribuição da numeração.

3.2.1 Bloqueios nas Tabelas de Geração de Sequências

Conforme mencionado na [Subsubseção 2.1.2.1](#), para o controle dos relacionamentos lógicos na aplicação, no caso do SGBD Microsoft SQL Server, são utilizadas tabelas auxiliares ([Figura 4b](#)), ao contrario da implementação para Oracle ([Figura 4a](#)), que utiliza o *DUAL* ([DOCS, 2017](#)) e da implementação para ([DOCS, 2022](#)) que adota *SEQUENCES*, embora a Microsoft também forneça uma solução similar ([DOCS, 2022](#)), os desenvolvedores optaram por adotar uma abordagem diferente.

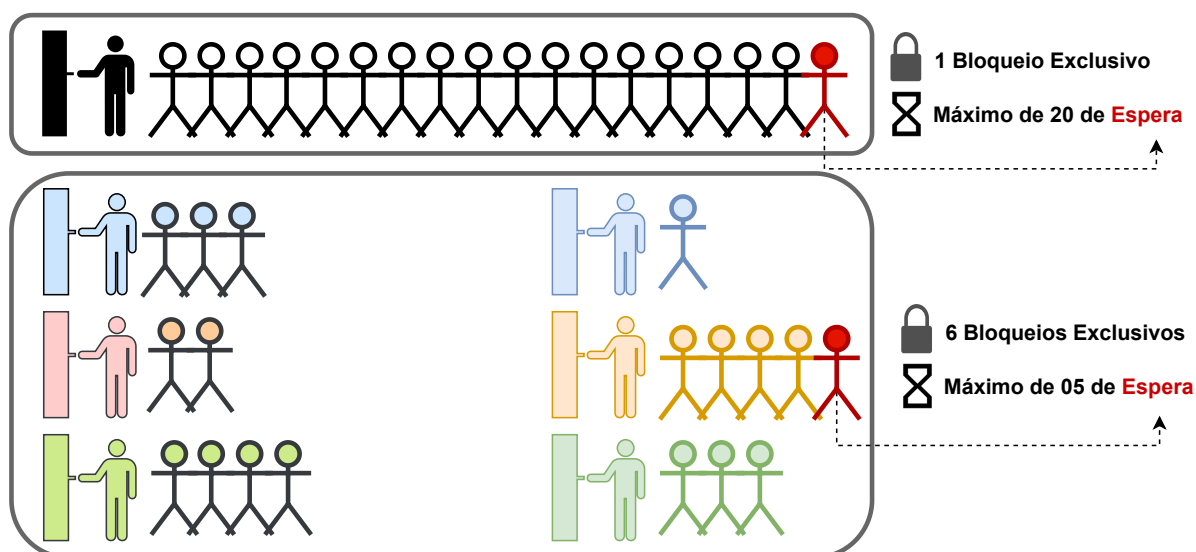
Infelizmente, só foi possível realizar a exclusão diária do lixo gerado pela abordagem para Microsoft, mas bloqueios são constantes nessas tabelas.

3.2.2 Bloqueios na tabela de Geração de Sequência Auxiliares

Identificou-se que embora não seria possível acabar com os bloqueios exclusivos, ao analisar o código e comportamento da aplicação durante as ações de criação de documentos e processos, uma alteração em massa espalhou os contadores entre diversas linhas, aumentando a quantidade de bloqueios, porém, diminuindo o tempo entre eles e o tempo total de bloqueios.

A ideia foi motivada por um artigo ([PURSWANI, 2022](#)), que permitiu o entendimento de que a quantidade de bloqueios não é necessariamente um problema, e sim o tempo de espera enquanto o SGBD tenta resolve-lo.

Figura 11 – Exemplificação da solução aplicada para diminuição do tempo de bloqueio.

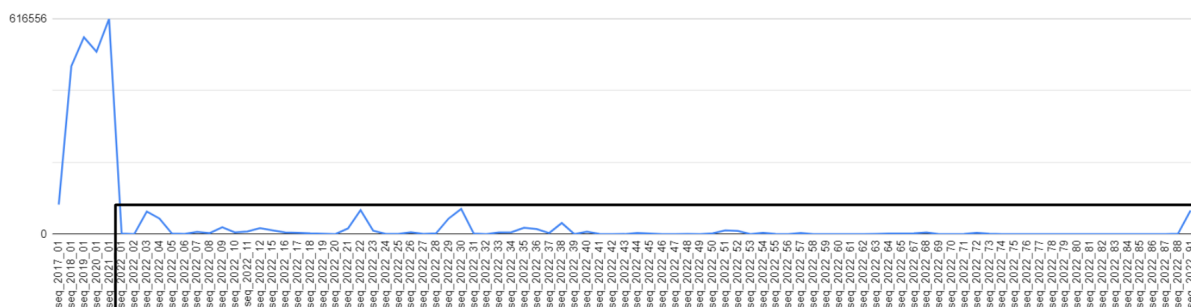


Fonte: Do Autor.

É como imaginar uma entrada em um grande teatro ou cinema, com apenas uma 'ticketeria', fazendo com que o maior tempo fosse aguardando o *ticket* (que na aplicação seria

o ID do documento/processo). Ao aumentar os recursos computacionais, o basicamente o que estaria crescendo era a sala de espera, e não a quantidade de 'ticketerias'. A [Figura 11](#) demonstra essa situação.

Figura 12 – Resultado da solução aplicada para diminuição do tempo de bloqueio.



Fonte: Do Autor.

Na [Figura 12](#), são apresentados os resultados da aplicação da solução de espalhar os bloqueios. É possível notar que no ano de 2021, ocorreram mais de 600 mil bloqueios em uma mesma linha, situação bem diferente do ano de 2022, embora este teve mais operações de atualização.

Tabela 2 – Resultado após a tentativa de espalhar os bloqueios.

Sequência	Linhas	Atualizações (TOTAL)	(MAX)	Média	Desvio Padrão
seq_2017	1	84826	—	—	—
seq_2018	1	481287	—	—	—
seq_2019	1	563838	—	—	—
seq_2020	1	522327	—	—	—
seq_2021	1	616556	—	—	—
seq_2022	86	623339	72172	31782,12088	112731,8025

Fonte: Do Autor.

Considerando os dados apresentados na [Tabela 2](#)³ e na [Figura 12](#) e tomando como exemplo a ideia apresentada na [Figura 11](#), em 2021 ocorreu 616 mil bloqueios sendo este em uma mesma linha, ocasionando em um tempo máximo teórico também de 616 mil de espera.

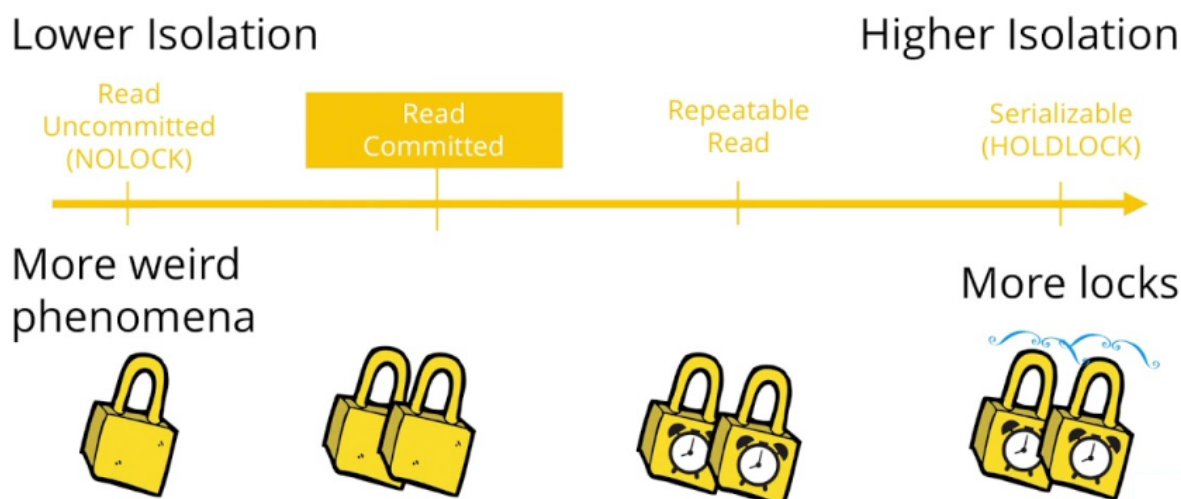
Entretanto, em 2022, com o espalhamento aplicado, o total de bloqueios (*tickets*) foi de 623 mil. Embora maior, o tempo máximo teórico de espera caiu para 72 mil, uma queda de 88,42% em relação ao tempo máximo teórico de espera do ano anterior e 94,90% menor se considerarmos o tempo médio.

³ Sequências com uma linha possuem valor máximo igual ao total, tendo este e os campos estatísticos suprimidos.

3.2.3 Bloqueios nas Demais Estruturas

A solução implantada foi alterar o nível de isolamento de instantâneo (*SNAPSHOT ISOLATION*) e o controle de versão em linha e leitura confirmada (*READ COMMITTED*).

Figura 13 – Demonstração dos níveis de Isolamento para tratamento de Bloqueios.



Fonte: <<https://littlekendra.com/course/read-committed-is-bonkers/why-read-committed-is-the-default-isolation-level-and-what-to-do-next/>>

Conforme exemplo apresentado pelo GeekForGeeks⁴, este nível de tratamento de bloqueios pode gerar fenômenos de aberrações nos dados. Porém, a aplicação consegue identificar e tratar essas situações.

Foi possível notar um maior uso de recursos computacionais após a ativação do recurso, porém, com uma execução bem mais fluída.

3.3 Paralelismo

Desde à adoção efetiva do Sistema Eletrônico de Informação pelos diversos órgãos do ente, um dos tempos de espera que sempre se fez presente foi o *CXPacket*. Trata-se de um *Wait Time* de sincronização, sendo relacionado com a sincronização das rotinas em paralelo existentes numa consulta, sendo assim, caso ele esteja presente é por que existe uma execução de consulta sendo paralelizada e não necessariamente problemas no tempo de resposta.

Ocorre que com o aumento do número de usuários, órgãos, unidades, entre outros, notoriamente, apesar de ter sido aumentada consideravelmente a capacidade de processamento, os tempos de espera relacionados ao *CXPacket* também aumentavam.

A documentação oficial sugere um valor inicial de 5 unidades de custo como limiar para paralelismo, o que significa que toda consulta abaixo de custo deverá ser executada em

⁴ Disponível em: <<https://www.geeksforgeeks.org/what-is-snapshot-isolation/>>

apenas uma *thread*. A única documentação encontrada foi um artigo de 2014 do DevMedia⁵, ao qual apontava para outro artigo que sugeria um valor próximo a 50 unidades de custo.

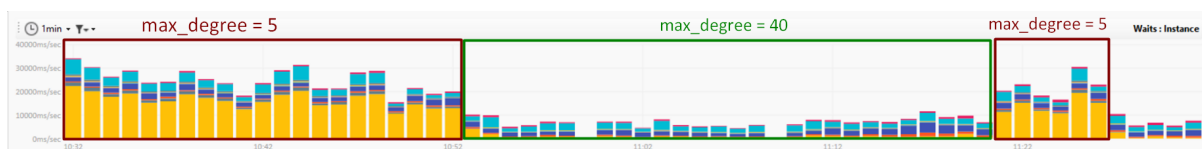
Figura 14 – Configuração apresentada pela ferramenta *SqlServer Management Studio*

Parallelism	
Cost Threshold for Parallelism	5
Locks	0
Max Degree of Parallelism	0
Query Wait	-1

Cost Threshold for Parallelism
Specify the threshold where Microsoft SQL Server creates and executes parallel plans.

Fonte: Do Autor.

Figura 15 – Diminuição do tempo de espera *CXPacket* após alteração do limiar de custo para paralelismo



Fonte: Do Autor.

Após diversos testes e acompanhamento, chegou-se à conclusão que um valor atribuído de 40 como custo limiar para paralelismo se mostrou com uma ótima relação de custo de processamento e tempo de espera. É possível notar que o *CXPacket* (em amarelo) caiu de cerca de 2000 milissegundos (com picos entre 2500 e 1500 milissegundos) para 50 milissegundos (com picos entre 100 e 0 milissegundos). Uma diminuição de 97,5% no tempo de espera.

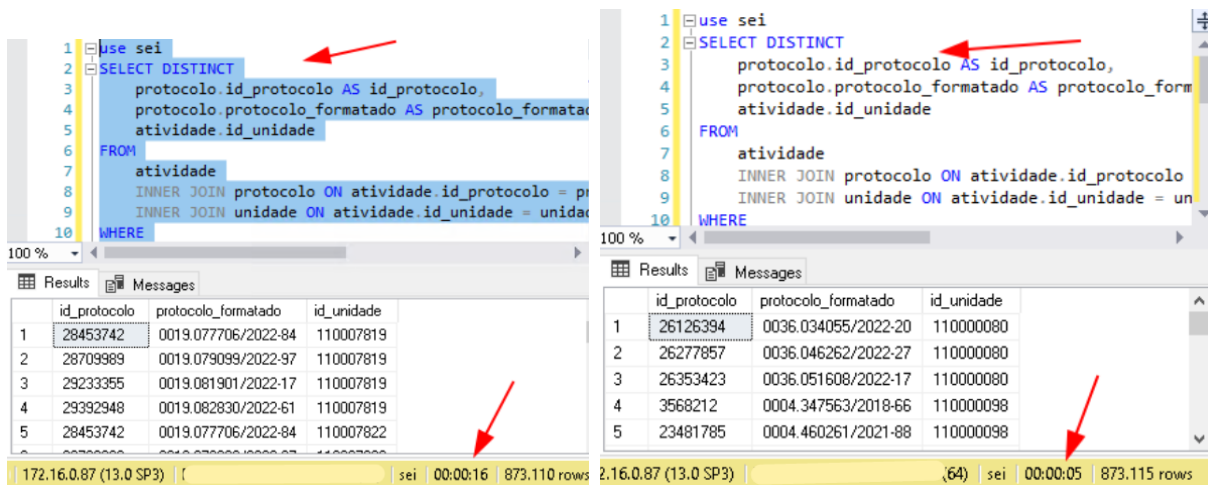
3.4 Criação de Índices

A execução autônoma da ferramenta SQL Alerts, permitiu a identificação de consultas as quais poderiam ser criados novos índices com objetivo de torná-las mais performáticas. A performance de uma consulta é medida através do seu custo computacional para recuperação do conteúdo e a percepção dá-se em função do tempo de espera do usuário.

A seguir é apresentado um exemplo de criação de índice e como isso impactou no tempo de recuperação da informação. Para este trabalho, os custos, embora menores, não serão levados em conta, devido à complexidade que envolve a explicação de seu cálculo.

⁵ Disponível em: <<https://www.devmedia.com.br/paralelismo-do-sql-server-com-o-cxpacket/31608>>

Figura 16 – Exemplo de consulta após criação de índice sugerido.



Fonte: Do Autor.

Ao observar a [Figura 16](#), é possível notar uma redução de 68,75% no tempo de recuperação da informação.

4 CONSIDERAÇÕES FINAIS

A seguir são apresentadas as considerações finais sobre o trabalho.

4.1 Limitações

Como limitações do estudo, além dos entraves legais, que não permitem a alteração no código-fonte da aplicação, o tempo relativamente curto para o desenvolvimento do projeto acaba por se tornar um fator limitante do escopo deste trabalho.

Vale ressaltar também a especificidade do caso, devido a criticidade do problema, a adoção de alguns materiais (e.g. quais ferramentas testar), assim como parte dos métodos adotados (e.g. parâmetros de teste e critérios de validação) foram advindos de conhecimento empírico do autor.

4.2 Trabalhos Futuros

Como trabalhos futuros, torna-se interessante condensar as soluções em um pacote e/ou arcabouço de soluções específicas para bases de dados da aplicação SEI!, auxiliando as demais entidades que também utilizam a aplicação. Outra linha interessante é a identificação e proposição de melhorias a nível de projeto e implementação em código-fonte.

Referências

- DOCS, O. **Database SQL language reference**. Oracle, 2017. Disponível em: <<https://docs.oracle.com/database/121/SQLRF/queries009.htm>>. Citado na página 19.
- DOCS, P. **9.17.Sequence manipulation functions**. PostgreSQL, 2022. Disponível em: <<https://www.postgresql.org/docs/current/functions-sequence.html>>. Citado na página 19.
- DOCS, S. S. **Create sequence (transact-SQL) - SQL server**. Microsoft, 2022. Disponível em: <<https://learn.microsoft.com/en-us/sql/t-sql/statements/create-sequence-transact-sql?view=sql-server-ver16>>. Citado na página 19.
- MINISTÉRIO DA ECONOMIA. **Processo Eletrônico Nacional (PEN)**. 2022. 1 p. Disponível em: <<https://www.gov.br/economia/pt-br/assuntos/processo-eletronico-nacional/conteudo/processo-eletronico-nacional-pen>>. Acesso em: 15 de maio de 2022. Citado na página 7.
- MINISTÉRIO DA ECONOMIA. **Sistema Eletrônico de Informações**. 2022. 1 p. Disponível em: <<https://www.gov.br/economia/pt-br/aceso-a-informacao/sei>>. Acesso em: 15 de maio de 2022. Citado na página 7.
- PURSWANI, A. **Difference between shared lock and Exclusive Lock**. 2022. Disponível em: <<https://www.geeksforgeeks.org/difference-between-shared-lock-and-exclusive-lock/>>. Citado na página 19.
- SQL Magazine. **Expurgo de dados - Organização, planejamento e realização da técnica de remoção de dados históricos chamada expurgo em bancos de dados de médio e grande porte**. 2010. 1 p. Disponível em: <<https://www.devmedia.com.br/expurgo-de-dados-sql-magazine-76/17189>>. Acesso em: 15 de maio de 2022. Citado na página 10.