

**FACULDADE DO VALE DO ITAJAÍ MIRIM – FAVIM
UNIASSELVI – ASSEVIM**

MATEUS DE AVIZ

**ANALISE DE DESEMPENHO COMPARATIVO ENTRE OS SGBDS SQL SERVER E
POSTGRESQL**

BRUSQUE

2022-1

FACULDADE DO VALE DO ITAJAÍ - MIRIM – FAVIM
UNIASSELVI - ASSEVIM
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

ANALISE DE DESEMPENHO COMPARATIVO ENTRE OS
SGBDS SQL SERVER E POSTGRESQL

MATEUS DE AVIZ

Proposta de Trabalho de Graduação submetida à Faculdade do vale do Itajaí – FAVIM – UNIASSELVI - ASSEVIM para a obtenção dos créditos na disciplina Trabalho de Graduação I do curso de Sistemas de Informação — Bacharelado.

Prof. Msc. Fernando César de Ornelas – Orientador

BRUSQUE,
2022-1

FACULDADE DO VALE DO ITAJAÍ - MIRIM – FAVIM
UNIASSELVI - ASSEVIM
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

ANALISE DE DESEMPENHO COMPARATIVO ENTRE OS
SGBDS SQL SERVER E POSTGRESQL

MATEUS DE AVIZ

TRABALHO DE CONCLUSÃO DE CURSO, APRESENTADO AO CURSO DE
GRADUAÇÃO DE SISTEMA DE INFORMAÇÃO, DA SOCIEDADE EDUCACIONAL
DO VALE DO ITAJAÍ MIRIM – ASSEVIM.

AVALIADO EM ____/__/__

ORIENTADOR: FERNANDO ORNELAS

BRUSQUE,
2022-1

LISTA DE FIGURAS

Figura 1 Arquitetura de um banco de dados.....	17
Figura 2 Comunicação Cliente Servidor SQL Server	19
Figura 3 Arquitetura de um Sistema Operacional	19
Figura 4 Arquitetura de Virtualização	21
Figura 5 - Resultado da consulta com coleta de estatísticas de performance no SQLServer ...	43
Figura 6 - Plano de execução no SQLServer.....	44

LISTA DE TABELA

Tabela 1 - Cronograma para o desenvolvimento do TG no 2º semestre	24
Tabela 2 - Especificações de hardware do computador de teste	25
Tabela 3 - Especificações Maquinas Virtuais	26
Tabela 4 - Definições dos ambientes	26
Tabela 5 - Tabela OrderBig	29
Tabela 6 - Tabela OrderDetaisBig	30
Tabela 7 - Tabela CustomersBig	30
Tabela 8 - Consultas do Cenário 1	31
Tabela 9 - Consultas do Cenário 2	32
Tabela 10 - Consultas do Cenário 3	32
Tabela 11 - Consultas do Cenário 4	33
Tabela 12 - Consultas do Cenário 5	34
Tabela 13 - Consultas do Cenário 6	35
Tabela 14 - Consultas do Cenário 7	36
Tabela 15 - Consultas do Cenário 8	36
Tabela 16 - Consultas do Cenário 9	37
Tabela 17 - Consultas do Cenário 10	38
Tabela 18 - Consultas do Cenário 11	39
Tabela 19 - Consultas do Cenário 12	39
Tabela 20 - Consultas do Cenário 13	40
Tabela 21 - Consultas do Cenário 14	41
Tabela 22 - Tabelas e chaves dos índices criados	42

Tabela 23 - Coleta de leituras logicas e tempo de execucao da consulta 1 do cenario 1	45
Tabela 24 - Coleta de leituras logicas e tempo de execucao da consulta 2 do cenario 1	45
Tabela 25 - Coleta de leituras logicas e tempo de execucao da consulta 3 do cenario 1	46
Tabela 26 - Coleta de leituras logicas e tempo de execucao da consulta 4 do cenario 1	46
Tabela 27 - Coleta de leituras logicas e tempo de execucao da consulta 1 do cenario 2	49
Tabela 28 - Coleta de leituras logicas e tempo de execucao da consulta 2 do cenario 2	49
Tabela 29 - Coleta de leituras logicas e tempo de execucao da consulta 3 do cenario 2	49
Tabela 30 - Coleta de leituras logicas e tempo de execucao da consulta 4 do cenario 2	50
Tabela 31 - Coleta de leituras logicas e tempo de execucao da consulta 1 do cenario 3	52
Tabela 32 - Coleta de leituras logicas e tempo de execucao da consulta 2 do cenario 3	52
Tabela 33 - Coleta de leituras logicas e tempo de execucao da consulta 3 do cenario 3	53
Tabela 34 - Coleta de leituras logicas e tempo de execucao da consulta 4 do cenario 3	53
Tabela 35 - Coleta de leituras logicas e tempo de execucao da consulta 1 do cenario 4	56
Tabela 36 - Coleta de leituras logicas e tempo de execucao da consulta 2 do cenario 4	57
Tabela 37 - Coleta de leituras logicas e tempo de execucao da consulta 3 do cenario 4	57
Tabela 38 - Coleta de leituras logicas e tempo de execucao da consulta 1 do cenario 5	60
Tabela 39 - Coleta de leituras logicas e tempo de execucao da consulta 2 do cenario 5	60
Tabela 40 - Coleta de leituras logicas e tempo de execucao da consulta 3 do cenario 5	60
Tabela 41 - Coleta de leituras logicas e tempo de execucao da consulta 4 do cenario 5	61
Tabela 42 - Coleta de leituras logicas e tempo de execucao da consulta 1 do cenario 6	63
Tabela 43 - Coleta de leituras logicas e tempo de execucao da consulta 2 do cenario 6	63
Tabela 44 - Coleta de leituras logicas e tempo de execucao da consulta 3 do cenario 6	64
Tabela 45 - Coleta de leituras logicas e tempo de execucao da consulta 1 do cenario 7	66
Tabela 46 - Coleta de leituras logicas e tempo de execucao da consulta 1 do cenario 8	68

Tabela 47 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 9	69
Tabela 48 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 10	71
Tabela 49 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 11	72
Tabela 50 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 12	73
Tabela 51 - Coleta de leituras logicas e tempo de execução da consulta 2 do cenário 12	74
Tabela 52 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 13	75
Tabela 53 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 14	77
Tabela 54 - Pontuação por ambiente	78
Tabela 55 - Pontuação por SGBD	79
Tabela 56 - Pontuação por Sistema Operacional.....	79

LISTA DE GRAFICOS

Gráfico 1 - Diferença de tempo de execução na consulta 1 do cenário 1	47
Gráfico 2 - Diferença de leituras lógicas na consulta 1 do cenário 1	47
Gráfico 3 - Diferença de tempo de execução na consulta 3 do cenário 1	48
Gráfico 4 - Diferença de tempo de execução na consulta 2 do cenário 2	50
Gráfico 5 - Diferença de leituras lógicas na consulta 2 do cenário 2	51
Gráfico 6 - Diferença de tempo de execução na consulta 1 do cenário 3	54
Gráfico 7 - Diferença de leituras lógicas na consulta 1 do cenário 3	55
Gráfico 8 - Diferença de tempo de execução na consulta 3 do cenário 3	55
Gráfico 9 - Diferença de tempo de execução na consulta 1 do cenário 4	58
Gráfico 10 - Diferença de tempo de execução na consulta 1 do cenário 4	59
Gráfico 11 - Diferença de tempo de execução na consulta 1 do cenário 5	61
Gráfico 12 - Diferença de leituras lógicas na consulta 1 do cenário 5	62
Gráfico 13 - Diferença de tempo de execução na consulta 3 do cenário 6	65
Gráfico 14 - Diferença de número de leituras lógicas na consulta 3 do cenário 6	65
Gráfico 15 - Diferença de tempo de execução na consulta 1 do cenário 7	67
Gráfico 16 - Diferença de tempo de execução na consulta 1 do cenário 8	68
Gráfico 17 - Diferença de tempo de execução na consulta 1 do cenário 9	70
Gráfico 18 - Diferença de tempo de execução na consulta 1 do cenário 10	71
Gráfico 19 - Diferença de tempo de execução na consulta 1 do cenário 11	73
Gráfico 20 - Diferença de tempo de execução na consulta 2 do cenário 12	75
Gráfico 21 - Diferença de tempo de execução na consulta 1 do cenário 13	76
Gráfico 22 - Diferença de tempo de execução na consulta 1 do cenário 14	78

SUMÁRIO

1 INTRODUÇÃO.....	11
1.1 CARACTERIZAÇÃO DO PROBLEMA	12
1.2 JUSTIFICATIVA	13
1.3 OBJETIVO GERAL	14
1.4 OBJETIVO ESPECIFICO	14
2 REVISÃO BIBLIOGRÁFICA E FUNDAMENTAÇÃO	15
2.1 BANCO DE DADOS	15
2.2 SQL	16
2.3 SGBD	16
2.3.1 POSTGRESQL	17
2.3.2 SQL SERVER.....	18
2.4 SISTEMAS OPERACIONAIS	19
2.4.1 LINUX	20
2.4.2 WINDOWS SERVER.....	20
2.5 MAQUINAS VIRTUAIS	21
2.6 FERRAMENTAS DE ACESSO A BANCO DE DADOS	22
2.6.1 SQL SERVER MANAGEMENT STUDIO	22
2.6.2 PGADMIN	22
3 METODOLOGIA.....	23
4 DESENVOLVIMENTO DO TRABALHO.....	25
4.1 PREPARAÇÃO DO AMBIENTE	25
4.2 VIRTUALIZAÇÃO DOS SERVIDORES	25
4.3 INSTALAÇÃO DOS SISTEMAS OPERACIONAIS E SGBDS	27
4.4 CRIAÇÃO DA BASE DE DADOS	27
4.5 CRIAÇÃO DAS CONSULTAS, TABELAS E ÍNDICES	28
4.5.1 Tabelas Utilizadas	29
4.5.2 Consultas Utilizadas.....	31
4.5.3 Índices Criados.....	41
4.6 REALIZAÇÃO DOS TESTE	42

4.7 RESULTADO DOS TESTES	44
4.7.1 Cenário 1	45
4.7.2 Cenário 2	48
4.7.3 Cenário 3	51
4.7.4 Cenário 4	56
4.7.5 Cenário 5	59
4.7.6 Cenário 6	62
4.7.7 Cenário 7	66
4.7.8 Cenário 8	67
4.7.9 Cenário 9	69
4.7.10 Cenário 10	70
4.7.11 Cenário 11	72
4.7.12 Cenário 12	73
4.7.13 Cenário 13	75
4.7.14 Cenário 14	76
4.8 CONSIDERAÇÃO SOBRE OS RESULTADOS DOS TESTES	78
5 CONSIDERAÇÕES FINAIS	80

1 INTRODUÇÃO

O crescente volume de SGBDs disponíveis no mercado pode acarretar dúvidas para as empresas em qual deve ser adotado para desenvolver o seu negócio, muitas empresas escolhem o que possui maior referência, outras optam pelo que tem o melhor custo benefício, ou deixam na mão dos desenvolvedores escolherem o que os mesmos assumem terem maior domínio.

Quando existe uma escolha a tomar de qual SGBD deve ser utilizado para o desenvolvimento de um produto, vários pontos devem ser considerados, um deles é a performance esperada, de acordo com a carga que será demandada.

Neste trabalho será exploraradp sobre os SGBDs PostgreSQL e SQLServer, entendendo seu funcionamento, sistemas operacionais compatíveis e realizar um comparativo performático entre esses dois SGBDs para poder concluir em quais pontos cada um possui vantagens.

1.1 CARACTERIZAÇÃO DO PROBLEMA

Com a crescente demanda de desenvolver diversas soluções em software dentro das empresas, se faz necessário avaliar os SGBDs disponíveis no mercado para atender as necessidades da solução desenvolvida. Com tantas tecnologias de banco de dados disponíveis no mercado, surgem dúvidas por parte dos desenvolvedores e da empresa de qual adotar para atender sua necessidade. Por falta de conhecimento por parte de quem deve tomar a decisão de qual SGBD utilizar, o problema pode ser percebido apenas muito tempo após a implementação, o que pode causar um grande prejuízo para a empresa e muitas vezes uma mão de obra pesada para mudar de SGBD.

1.2 JUSTIFICATIVA

Conseguir orientar melhor os analistas de sistemas, arquitetos e desenvolvedores no momento de escolher um SGBD para o seu produto, tendo em vista que na maior dos casos o desenvolvedor não tem conhecimento aprofundado no assunto de arquitetura e performance de banco de dados.

Essa escolha se faz de extrema importância, para que no futuro não tenha surpresas desagradáveis por ter sido escolhido uma tecnologia que não atende performaticamente a necessidade do produto, destacando que, o intuito do trabalho não é mostrar qual é a melhor, e sim a partir de vários testes demonstrar qual se aplica melhor para determinadas situações para que o analista ou desenvolvedor possa tomar a decisão de forma mais assertiva.

1.3 OBJETIVO GERAL

O objetivo deste trabalho é demonstrar em qual cenário existe uma melhor performance por parte do SGBD PostgreSQL e que outro cenário a melhor performance é pelo SGBD SQL Server, a partir de testes e técnicas de performance.

1.4 OBJETIVO ESPECIFICO

- a) Avaliar a performance entre os dois SGBDs com uma instalação limpa e sem índices.
- b) Avaliar o ganho de performance nos dois SGBDs com a criação de índices.
- c) Avaliar o ganho de performance no dois SGBDs utilizando diferentes sistemas operacionais.
- d) Demonstrar através de gráfico e indicadores o resultado comparativo de performance dos dois SGBDs em situações similares e específicas.

2 REVISÃO BIBLIOGRÁFICA E FUNDAMENTAÇÃO

Para possibilitar a avaliação dos SGBDs, foi necessário fazer uma pesquisa bibliográfica em diversas áreas de tecnologia, como banco de dados, sistemas operacionais, servidores virtualizados e ferramentas de acesso e administração de banco de dados, conforme a seguir.

2.1 BANCO DE DADOS

Bancos de dados e sistema de banco de dados são um componente essencial da vida na sociedade moderna; a maioria de nós encontra diariamente diversas atividades que envolvem alguma interação com um banco de dados. Por exemplo, quando vamos ao banco para depositar ou retirar fundos, quando fazemos uma reserva de hotel ou de voo, quando acessamos o catálogo de uma biblioteca para buscar um livro ou quando compramos algo online - como um livro, um brinquedo ou um computador -, essas atividades provavelmente envolverão alguém ou algum programa de computador que acessa um banco de dados. Até mesmo a compra de produtos em um supermercado em geral atualiza automaticamente o banco de dados que mantém o controle de estoque dos itens. (ELMASRI, 2019)

Conforme destacado por Elmasri (2019) podemos definir um banco de dados da seguinte forma “Um banco de dados é uma coleção de dados relacionados. Com dados queremos dizer fatos conhecidos que podem ser registrados e que possuem significado implícito.”

De acordo com o artigo da DevMedia (2006) na década de 60, os computadores se tornaram parte efetiva do custo das empresas e assim surgiu dois modelos de dados: em rede e hierárquico. Nessa época o acesso a o banco de dados era feito através de operações de ponteiros de baixo nível que unem os registros. Era necessário inclusive conhecer a estrutura física do banco de dados para realizar uma consulta. O modelo relacional surgiu na década de 70, proposto por Edgar Frank Codd, os principais sistemas de banco de dados nessa década

eram o Ingres e o System R. Nos anos 80 a linguagem estruturada de consulta SQL se tornou um padrão mundial e nos anos 90 com a chegada da internet os bancos de dados de código aberto ganham força além de bancos OLTP (On-Line Transaction Process) e OLAP (On-Line Analytical Process). No século 21 temos a predominância da IBM (DB2), Microsoft (SQLServer) e Oracle a frente dos bancos de dados do mercado, além dos bancos open source, como PostgreSQL e Mysql e também a popularização dos bancos de dados NoSQL como MongoDB e Redis.

2.2 SQL

Itzik Ben-Gan (2016) explica o conceito de SQL da seguinte forma. “SQL is both an ANSI and ISO standard language based on the relational model, designed for querying and managing data in a RDBMS. ”

Também destaca a forma em que o SQL trabalha, sendo muito parecido com a língua inglesa. “Interestingly, SQL resembles English and is also very logical. Unlike many programming languages which use an imperative programming paradigm, SQL uses a declarative one. That is, SQL requires you to specify what you want to get and how to get it, letting the RDBMS figure out the physical mechanics required to process your request” (Itzik Ben-Gan, 2016)

2.3 SGBD

Na definição de Elmasri (2019). Um sistema de gerenciamento de banco de dados (SGBD) é um sistema computadorizado que permite que os usuários criem e mantenham um banco de dados. O SGBD é um sistema de software de uso geral que facilita o processo de definição, construção, manipulação e compartilhamento de banco de dados entre diversos usuários e aplicação.

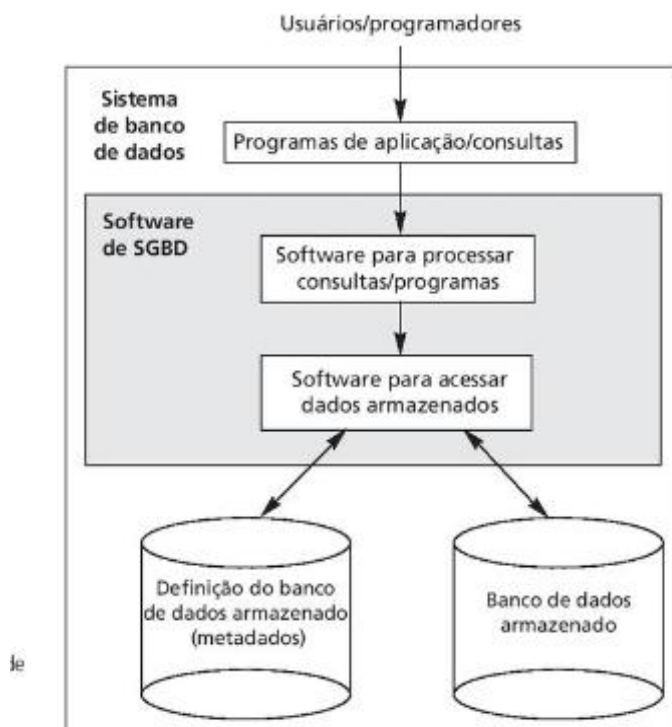


Figura 1 Arquitetura de um banco de dados
Fonte: Elmasri,, 2019

2.3.1 POSTGRESQL

Segundo Vinícius Carvalho (2017) “O PostgreSQL é um poderoso sistema gerenciador de banco de dados objeto-relacional de código aberto.”

O PostgreSQL foi baseado no Postgres 4.2, da Universidade da Califórnia/Berkeley, projeto encerrado em 1994. Ele é dito objeto-relacional por ter suporte a funcionalidades compatíveis com o conceito de orientação de objetos, como herança de tabelas, sobrecarga de operadores e funções e uso de métodos e construtores através de funções. (Fabio Cutt, 2015)

Para Vinicius Caralho as principais funcionalidades do PostgreSQL são:

Como um banco de dados de nível corporativo, o PostgreSQL possui funcionalidades sofisticadas como:

- O controle de concorrência multiversionado (MVCC, em inglês);
- Recuperação em um ponto no tempo (PITR, em inglês), tablespaces;
- Replicação assíncrona;
- Transações agrupadas (savepoints);
- Cópias de segurança quente (online/hot backup);
- Um sofisticado planejador de consultas (otimizador) e registrador de transações sequencial (WAL) para tolerância a falhas;
- Suporta conjuntos de caracteres internacionais;
- Codificação de caracteres multibyte, Unicode e sua ordenação por localização;
- Sensibilidade a caixa (maiúsculas e minúsculas) e formatação;
- É altamente escalável, tanto na quantidade enorme de dados que pode gerenciar quanto no número de usuários concorrentes que pode acomodar. Existem sistemas ativos com o PostgreSQL em ambiente de produção que gerenciam mais de 4TB de dados

2.3.2 SQL SERVER

SQL Server é um dos sistemas gerenciadores de banco de dados mais conhecidos e mais utilizados no mundo, ele tem muita estabilidade e inteligência de negócio.

Microsoft SQL Server is considered to be one of the most commonly used systems for database management in the world. This popularity has been gained by a high degree of stability, security, and business intelligence and integration functionality. Microsoft SQLServer for Linux is a database server that accepts queries from clients, evaluates them, and then internally executes them to deliver results to the client. The client is an application that produces queries through a database provider and communication protocol sends requests to the server and retrieves the result for client-side processing and/or presentation. (AZEMOVIC, 2017)

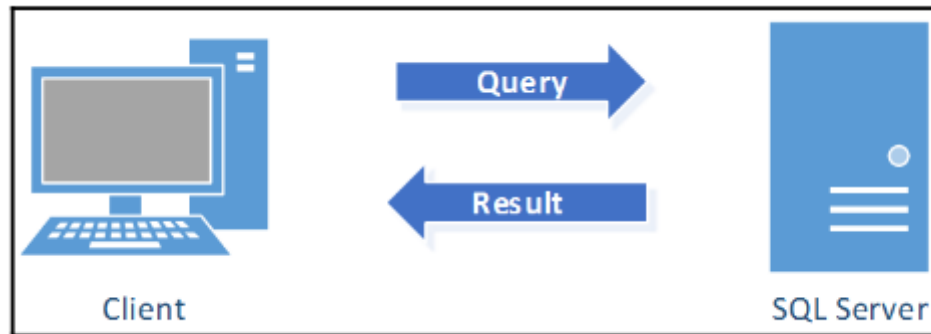


Figura 2 Comunicação Cliente Servidor SQL Server
Fonte: Azemovic, 2017

2.4 SISTEMAS OPERACIONAIS

Na definição de Tanebaum (2017), o sistema operacional é responsável por gerenciar diversos recursos de um sistema computacional.

Um sistema computacional moderno consiste em um ou mais processadores, memória principal, discos, impressoras, teclado, mouse, monitora, interfaces de rede e outros dispositivos de entrada e saída. Enfim, é um sistema complexo. Se cada programador de aplicações tivesse de entender como tudo isso funciona em detalhes nenhum código chegaria a ser escrito.

Além disso, gerenciar todos esses componentes e usá-los de maneira otimizada é um trabalho extremamente difícil. Por isso os computadores têm um dispositivo de software denominado sistema operacional, cujo trabalho é fornecer aos programas do usuário um modelo de computador melhor, mais simples e mais limpo e lidar com o gerenciamento de todos os recursos mencionados. (TANEBAUM, 2017)



Figura 3 Arquitetura de um Sistema Operacional
Fonte: Tanebaum, 2017

2.4.1 LINUX

Na definição de Juliano Ramos (2019) “O Linux é um sistema operacional de código aberto, disponível para alterações, melhorias e novas implementações. Dizer "Linux" envolve um contexto que deve ser observado. É necessário compreender a diferença de "Distribuição GNU/Linux" e "kernel linux", ou em alguns casos, apenas: "Distro Linux””.

Referente as distribuições Linux, (Nemeth, 2007) destaca "A maioria das distribuições do Linux pode fazer praticamente tudo o que você imagina querer fazer com um sistema Linux. Algumas delas podem exigir a instalação de software adicional para serem totalmente funcionais e outras podem facilitar determinadas tarefas; entretanto, suas diferenças não são muito significativas. Na realidade, há certo mistério no motivo de existirem tantas distribuições diferentes, cada uma das quais alardeando sua “instalação fácil” e uma “gigantesca biblioteca de programas” como características distintivas. É difícil evitar a conclusão de que as pessoas simplesmente gostam de criar distribuições do Linux."

2.4.2 WINDOWS SERVER

A documentação oficial da Microsoft traz como definição que o Windows Server é uma plataforma para compilar uma infraestrutura de aplicativos, redes e serviços Web conectados, do grupo de trabalho ao data center. Ele faz a ponte entre os ambientes locais e o Azure, adicionando mais camadas de segurança enquanto ajuda você a modernizar seus aplicativos e sua infraestrutura.

2.5 MAQUINAS VIRTUAIS

De forma simples, tem-se a definição de máquina virtual descrita por Silberschatz, que diz que a máquina virtual é basicamente uma cópia exata do hardware com a intenção de proporcionar um ambiente para execução de um sistema operacional. Dessa forma, é possível rodar várias máquinas virtuais em um determinado hardware (SILBERSCHATZ; GALVIN; GAGNE, 2005).

Tanebaum complementa que a virtualização permite que um único computador seja o hospedeiro de múltiplas máquinas virtuais, cada uma executando potencialmente um sistema operacional completamente diferente (TANEBAUM, 2017)

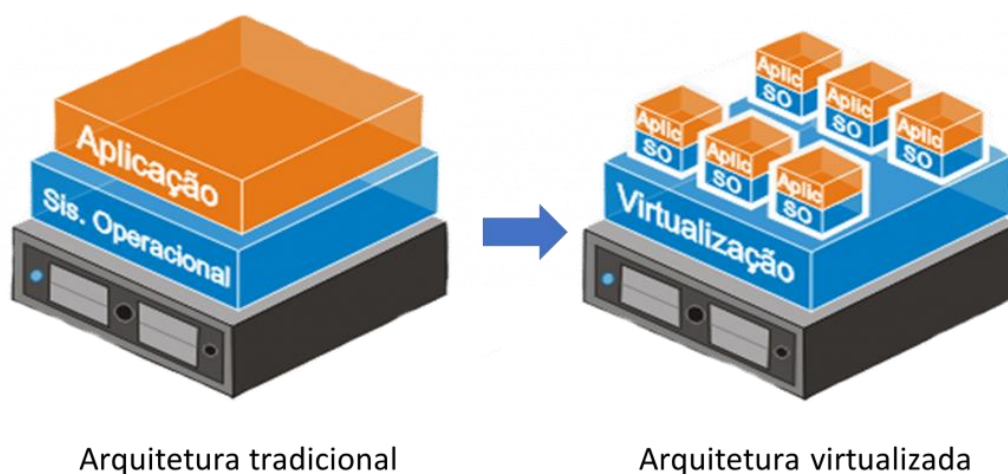


Figura 4 Arquitetura de Virtualização
Fonte: ArtBackup, 2021

2.6 FERRAMENTAS DE ACESSO A BANCO DE DADOS

2.6.1 SQL SERVER MANAGEMENT STUDIO

Conforme a definição encontrada na documentação oficial da Microsoft, desenvolvedora do SQL Server Managed Studio O SSMS (SQL Server Management Studio) é um ambiente integrado para o gerenciamento de qualquer infraestrutura de SQL. Use o SSMS para acessar, configurar, gerenciar, administrar e desenvolver todos os componentes do SQL Server, do Banco de Dados SQL do Azure e do Azure Synapse Analytics. O SSMS fornece um utilitário abrangente que combina um amplo grupo de ferramentas gráficas com vários editores de script avançados, a fim de fornecer acesso ao SQL Server para desenvolvedores e administradores de banco de dados de todos os níveis de habilidade.

2.6.2 PGADMIN

Na página oficial do software open source PGAdmin, tem-se a seguinte definição da ferramenta “pgAdmin is the leading Open Source management tool for Postgres, the world’s most advanced Open Source database. pgAdmin 4 is designed to meet the needs of both novice and experienced Postgres users alike, providing a powerful graphical interface that simplifies the creation, maintenance and use of database objects.”

3 METODOLOGIA

A pesquisa desenvolvida e do tipo aplicada, o conhecimento obtido durante o estudo bibliográfico será utilizado para realizar os testes de desempenho dos SGBDs SQL Server e PostgreSQL.

Segundo Cleber Cristiano Prodanov e Ernani Cesar de Freitas (2013) A pesquisa aplicada objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos. Envolve verdades e interesses locais.

O trabalho será desenvolvido observando as seguintes etapas:

- a) Estudo dos SGBDs: Nessa etapa será realizado um estudo aprofundado dos dois SGBDs, para uma melhor visão geral das possibilidades que podem ser testadas
- b) Preparação do Ambiente de Infraestrutura: Subir duas máquinas virtuais com Windows Server e duas máquinas virtuais com Linux para os testes
- c) Instalação padrão dos SGBDs: Será feita a instalação dos SGBDs nas máquinas virtuais mantendo a configuração padrão.
- d) Criação da base de teste: Será criada e populada duas bases de teste, uma em PostgreSQL e outra em SQL Server, as bases terão a estrutura o mais idêntico possível e a mesma volumetria de dados.
- e) Construção das Consultas: As consultas serão escritas em TSQL e PL/pgSQL da forma mais idêntica possível.
- f) Teste de desempenho (Sem alteração no ambiente): Será feito o teste de consumo de recurso e tempo de execução das consultas sem ter feito nenhuma alteração em ambos os ambientes.
- g) Teste de desempenho (Com índices): Os índices serão criados conforme as consultas e conforme a opção de índice que cada tecnologia possui para comparação.
- h) Teste de desempenho (S.O): Será testado em cada S.O visando a melhor performance e comparado o SGBD SQL Server em Windows e Ubuntu contra o SGBD PostgreSQL em Windows e Ubuntu.

- i) Gráficos de Comparação: Após concluir os testes, será construído gráficos e indicadores com os resultados

Tabela 1 - Cronograma para o desenvolvimento do TG no 2º semestre

etapas / quinzenas	2022									
	fev.		mar.		abr.		maio		jun.	
	1	2	1	2	1	2	1	2	1	2
Estudo dos SGBDs	■	■	■	■	■	■	■	■		
Preparação do Ambiente de Infraestrutura	■	■								
Instalação dos SGBDs		■								
Criação da base de teste	■	■								
Construção das Consultas			■	■						
Teste de desempenho (Sem índices)					■					
Teste de desempenho (Com índices)						■				
Gráficos de Comparação									■	■

Não existe necessidade de orçamento para esse projeto, será utilizado equipamento próprio para realizar os testes. Os servidores serão configurados como maquinas virtuais, utilizando o Vmware Workstation Player, sistema de virtualização gratuito.

Com o equipamento próprio utilizado será possível configurar todos os servidores e subir uma máquina virtual por vez para fazer os testes, disponibilizando 4 núcleos de processamento, 30GB de armazenamento SSD e 8GB de memória RAM. Referente os sistemas operacionais que serão instalados: Para Linux será utilizada a distribuição gratuita Ubuntu e para Windows será utilizado Windows Server com licença de estudante.

Sobre os SGBDs, para PostgreSQL não tem custo de licenciamento e para SQL Server será utilizada a licença gratuita de desenvolvimento que possui os mesmos recursos da licença Enterprise.

4 DESENVOLVIMENTO DO TRABALHO

4.1 PREPARAÇÃO DO AMBIENTE

Como o objetivo desse trabalho é avaliar a performance de duas tecnologias distintas, existe uma importância por trás do hardware utilizado para realizar esses testes. Os mesmos testes realizados em um diferente hardware podem ter resultados diferentes dos testes aqui apresentados. Para ter mais fidelidade nos testes, os SGBDS não foram instalados diretamente no computador, mas sim em máquinas virtuais que possuem as mesmas especificações de hardware e para cada teste foi iniciado uma máquina virtual por vez, para que uma não tenha impacto sobre a outra.

4.2 VIRTUALIZAÇÃO DOS SERVIDORES

Na tabela 2 se encontram as especificações de hardware do computador onde foi virtualizado os servidores de banco de dados.

Tabela 2 - Especificações de hardware do computador de teste

Marca	ACER
Modelo	A515-51G-58VH
Memória RAM	20GB DDR4 2666Mhz
Disco (S.O)	SSD M2 WD Green 120GB Leitura 545 MB/s
Disco (Dados)	SSD Crucial BX500 480GB Leitura 540MB/s Gravação 500MB/s
Processador	Intel Core i5 – 7200U 2.50GHz

Para as máquinas virtuais, as especificações utilizadas estão informadas na tabela 3, elas foram criadas utilizando o sistema VMware Workstation 16 Player, todas com as mesmas configurações, diferenciando somente o sistema operacional de acordo com o teste.

Tabela 3 - Especificações Máquinas Virtuais

Sistema de Virtualização	VMWare Workstation 16 Player Versão 16.2.3
Memória RAM	8 GB
Processadores	2
Espaço em Disco	50GB
Disco Utilizado	SSD Crucial BX500 480GB Leitura 540MB/s Gravação 500MB/s

Foram criados no total de 4 ambientes para que os testes pudessem ocorrer, dois ambientes com as especificações da tabela 3 e com sistema operacional Windows Server 2016 e dois ambientes com as especificações da tabela 3 e com sistema operacional Ubuntu 20.04. A distinção dos ambientes para fim dos testes ocorrem conforme a tabela 4

Tabela 4 - Definições dos ambientes

	SO	Banco	Objetivo
1	Windows Server	PostgreSQL	Testar desempenho no Banco PostgreSQL com SO Windows
2	Windows Server	SQLServer	Testar desempenho no Banco SQLServer com SO Windows
3	Ubuntu	PostgreSQL	Testar desempenho no Banco PostgreSQL com SO Ubuntu
4	Ubuntu	SQLServer	Testar desempenho no Banco SQLServer com SO Ubuntu

Dentro dos servidores, foi liberado o acesso externo, para que as ferramentas utilizadas para os testes pudessem ser acessadas diretamente pelo computador físico, e não pela máquina virtual, evitando assim consumo de recurso por usuário autenticado no servidor e dedicando completamente o hardware alocado para as operações do banco de dados.

4.3 INSTALAÇÃO DOS SISTEMAS OPERACIONAIS E SGBDS

Para instalar os sistemas operacionais, foram utilizadas as ISOs dos respectivos sistemas, Ubuntu 20.04 e Windows Server 2016 Standard versão de avaliação. Em ambos os sistemas somente foi configurado um usuário de acesso e liberado o acesso remoto, após isso foi instalado o SGBD de acordo com o ambiente.

Na instalação do SQL Server no Windows Server não foi realizada nenhuma mudança, foi instalado a partir do instalador encontrado no site oficial da Microsoft. Para a instalação do SQLServer no Ubuntu foi seguido os passos conforme a documentação oficial da Microsoft.

Para a instalação do PostgreSQL no Windows Server foi baixado o executável no site oficial do PostgreSQL e instalado o mesmo sem alterações e para o Ubuntu foi seguido os passos conforme a documentação oficial do PostgreSQL.

4.4 CRIAÇÃO DA BASE DE DADOS

Para os testes serem fidedignos, a estrutura do banco e a quantidade de dados utilizada foi a mesma para todos os ambientes e ambos SGBDs. Para isso foi utilizado o banco de dados Northwind, um banco de dados de exemplo disponibilizado pela Microsoft, esse banco

possui estrutura e dados fictícios baseados em um sistema de vendas. O banco foi baixado no formato de backup do SQL Server (.bak) e restaurado nos ambientes SQLServer.

Para o PostgreSQL foi utilizado a mesma estrutura, banco de dados Northwind, publicado no GitHub pelo desenvolvedor pthom de forma gratuita. Foi validado que a estrutura se manteve a mesma utilizada no banco do SQLServer e que os tipos de dados utilizados foram próximos.

Tendo a estrutura do banco de dados pronta para as duas tecnologias, foi necessário aumentar a quantidade de dados de algumas tabelas para que os testes fossem mais próximos de um banco de dados de uma empresa que já utiliza o sistema a alguns anos. Para isso foi utilizado o script disponibilizado por Fabiano Amorim no curso Dicas de Performance no SQLServer, que se baseia nos dados originais do banco de dados Northwind e cria quatro novas grandes tabelas.

Após criar as novas tabelas no SQLServer, foi realizado uma exportação de dados de cada tabela para um arquivo .sql e posteriormente criado estas tabelas nos ambientes PostgreSQL e importado os dados do arquivo .sql.

4.5 CRIAÇÃO DAS CONSULTAS, TABELAS E ÍNDICES

Diversos cenários foram criados para validar a performance de cada ambiente, será apresentado as tabelas e as consultas utilizadas junto ao objetivo que elas se propõem a atender.

4.5.1 Tabelas Utilizadas

As tabelas utilizadas para as consultas foram OrderBig, OrderDetailsBig, CustomerBig e ProductBig, com a estrutura conforme abaixo.

Tabela OrdersBig, com chave primaria no campo OrderID e todos os campos não nulos, ela foi populada com 10 milhões de registros. Tipagem de dados conforme tabela a seguir

Tabela 5 - Tabela OrderBig

Coluna	Tipo de dado
OrderID	Inteiro
CustomerID	Inteiro
OrderDate	Data
Value	Numérico (18,2)

Tabela OrderDetailsBig, com chave primaria composta nos campos OrderId e ProductID e chave estrangeira com a tabela OrderBig no campo OrderID, campo Quantity podendo receber valores nulos, e demais campos não nulos. Foi populada com 9.901.012 de registros e a tipagem de dado conforme tabela abaixo

Tabela 6 - Tabela OrderDetaisBig

Coluna	Tipo de dado
OrderId	Inteiro
ProductId	Inteiro
Shipped_Date	Data e Hora
Quantity	Inteiro

Tabela CustomersBig, com chave primaria em CustomerID, todos os campos, fora a chave, podem receber valores nulos. Tabela populada com 5 milhões de registros tipagem de dados na tabela

Tabela 7 - Tabela CustomersBig

Coluna	Tipo de dado
CustomerId	Inteiro
CityId	Inteiro
CompanyName	Caracter Variavel
ContactName	Caracter Variavel
Col1	Caracter Variavel
Col2	Caracter Variavel
CNPJ	Caracter Variavel

A última tabela é a ProductsBig, com chave primária no campo ProductId e com outros dois campos nulos, recebeu 50 mil registros. Tipagem de dados da tabela abaixo

Coluna	Tipo de dado
ProductId	Inteiro
ProductName	Carcter Variavel (209)
Col1	Caracter Variavel (250)

4.5.2 Consultas Utilizadas

Sobre as consultas utilizadas, será listado a seguir todas as consultas e sua numeração que será apresentada nos gráficos gerados.

Cenário 1

O Cenário 1 tem como intuito demonstrar o tempo levado para buscar todos os registros de cada tabela, levando em consideração o tipo de dados e a quantidade de registro que cada uma possui. Dessa forma se trata de 4 consultas, uma em cada tabela

Tabela 8 - Consultas do Cenário 1

Consulta 1	SELECT * FROM OrdersBig
Consulta 2	SELECT * FROM Order_DetailsBig
Consulta 3	SELECT * FROM CustomersBig
Consulta 4	SELECT * FROM ProductsBig

Cenário 2

No Cenário 2 será realizado 4 consultas, cada uma no campo que é a chave primária de cada tabela, pois em ambos os SGBDs, mesmo que não especifique um índice, por padrão é criado um índice na chave primaria.

Tabela 9 - Consultas do Cenário 2

Consulta 1	SELECT * FROM OrdersBig WHERE OrderID = 500000
Consulta 2	SELECT * FROM Order_DetailsBig WHERE OrderID = 86 AND ProductID = 37944
Consulta 3	SELECT CustomerID FROM CustomersBig WHERE CustomerID = 23
Consulta 4	SELECT * FROM ProductsBig WHERE ProductID = 417

Cenário 3

Neste cenário será inicialmente realizado uma consulta em um campo que não contem índice, depois será criado um índice nesse campo e comparado todos os cenários entre os ambientes. São 4 consultas nas 4 tabelas com variações de tipo de dado.

Tabela 10 - Consultas do Cenário 3

Consulta 1	SELECT * FROM OrdersBig WHERE CustomerID = 658
Consulta 2	SELECT * FROM Order_DetailsBig WHERE Quantity = 1002
Consulta 3	SELECT * FROM CustomersBig WHERE CNPJ = '11017104790000'
Consulta 4	SELECT * FROM ProductsBig WHERE ProductName = 'Rogede sild 4A7F6A98'

Cenário 4

O Cenário 4 apresenta 2 consultas realizando filtro por coluna de data, uma em uma coluna com apenas data e outra em uma coluna com data e hora. Será realizado primeiro a consulta sem índice no campo de data e depois criando o índice.

Tabela 11 - Consultas do Cenário 4

Consulta 1	SELECT * FROM OrdersBig WHERE OrderDate = '2024-11-17'
Consulta 2	SELECT * FROM Order_DetailsBig WHERE Shipped_Date = '2025-06-10 00:00:00.000'
Consulta 3	SELECT * FROM OrdersBig WHERE OrderDate > '2024-11-17'

Cenário 5

Neste Cenário será validado o uso de junção de tabelas, a partir de colunas que são chaves primárias. Será realizado 4 consultas, junção da OrderDetailsBig com ProductBig, junção da CustomersBig com OrdersBig, junção da OrdersDetailsBig com OrdersBig e junção das quatro tabelas em uma só consulta.

Tabela 12 - Consultas do Cenário 5

Consulta 1	<pre>SELECT CustomersBig.CompanyName, CustomersBig.ContactName, OrdersBig.Value,OrdersBig.OrderDate FROM OrdersBig JOIN CustomersBig ON OrdersBig.CUSTOMERID = CustomersBig.CustomerID WHERE CustomersBig.CustomerID = 17476 OR CustomersBig.CustomerID = 370</pre>
Consulta 2	<pre>SELECT ProductsBig.ProductName, Order_DetailsBig.Quantity FROM Order_DetailsBig JOIN ProductsBig ON Order_DetailsBig.ProductID = ProductsBig.ProductID WHERE ProductsBig.ProductID = 143</pre>
Consulta 3	<pre>SELECT OrdersBig.OrderId, OrdersBig.Value, Order_DetailsBig.ProductID, Order_DetailsBig.Quantity FROM Order_DetailsBig JOIN OrdersBig ON Order_DetailsBig.OrderID = OrdersBig.OrderID WHERE OrdersBig.OrderId = 156</pre>
Consulta 4	<pre>SELECT CustomersBig.CompanyName, CustomersBig.ContactName, OrdersBig.Value, OrdersBig.OrderDate, ProductsBig.ProductName, Order_DetailsBig.Quantity FROM OrdersBig JOIN CustomersBig ON OrdersBig.CUSTOMERID = CustomersBig.CustomerID JOIN Order_DetailsBig ON Order_DetailsBig.OrderID = OrdersBig.OrderID JOIN ProductsBig ON Order_DetailsBig.ProductID = ProductsBig.ProductID WHERE OrdersBig.OrderId = 156</pre>

Cenário 6

No Cenário 6 existirá a junção de tabelas, mas com campos do tipo data que não são chave e inicialmente não possuem índice, depois será criado os índices e validado a performance em cada ambiente, nesse cenário também será testado o uso do BETWEEN.

Tabela 13 - Consultas do Cenário 6

Consulta 1	<pre>SELECT CustomersBig.CompanyName, CustomersBig.ContactName, OrdersBig.Value, OrdersBig.OrderDate FROM OrdersBig JOIN CustomersBig ON OrdersBig.CUSTOMERID = CustomersBig.CustomerID WHERE OrdersBig.OrderDate BETWEEN '2025-10-15' AND '2026-03-20'</pre>
Consulta 2	<pre>SELECT ProductsBig.ProductName, Order_DetailsBig.Quantity, Order_DetailsBig.Shipped_Date FROM Order_DetailsBig JOIN ProductsBig ON Order_DetailsBig.ProductID = ProductsBig.ProductID WHERE Order_DetailsBig.Shipped_Date BETWEEN '2025-10-15' AND '2026-03-20'</pre>
Consulta 3	<pre>SELECT CustomersBig.CompanyName, CustomersBig.ContactName, OrdersBig.Value, OrdersBig.OrderDate, ProductsBig.ProductName, Order_DetailsBig.Quantity FROM OrdersBig JOIN CustomersBig ON OrdersBig.CUSTOMERID = CustomersBig.CustomerID JOIN Order_DetailsBig ON Order_DetailsBig.OrderID = OrdersBig.OrderID JOIN ProductsBig ON Order_DetailsBig.ProductID = ProductsBig.ProductID WHERE OrdersBig.OrderDate BETWEEN '2025-10-15' AND '2025-12-15'</pre>

Cenário 7

Será validado no Cenário 7 a performance ao utilizar Order By em uma query com junção de várias tabelas e o ganho ao criar um índice na clausula do Order By em cada ambiente.

Tabela 14 - Consultas do Cenário 7

Consulta 1	<pre> SELECT CustomersBig.CompanyName, CustomersBig.ContactName, OrdersBig.Value, OrdersBig.OrderDate, ProductsBig.ProductName, Order_DetailsBig.Quantity FROM OrdersBig JOIN CustomersBig ON OrdersBig.CUSTOMERID = CustomersBig.CustomerID JOIN Order_DetailsBig ON Order_DetailsBig.OrderID = OrdersBig.OrderID JOIN ProductsBig ON Order_DetailsBig.ProductID = ProductsBig.ProductID WHERE OrdersBig.OrderDate BETWEEN '2025-10-15' AND '2025-12-15' ORDER BY CustomersBig.ContactName, Order_DetailsBig.Quantity </pre>
-------------------	---

Cenário 8

No cenário 8 será validado a performance ao utilizar o limitador de cada SGBD, no caso do SQL Server o TOP e no PostgreSQL o Limit e a diferença de ter ou não índice nesse cenário.

Tabela 15 - Consultas do Cenário 8

Consulta 1	<pre> SELECT TOP 100 CustomersBig.CompanyName, CustomersBig.ContactName, OrdersBig.Value, OrdersBig.OrderDate, ProductsBig.ProductName, Order_DetailsBig.Quantity FROM OrdersBig JOIN CustomersBig ON OrdersBig.CUSTOMERID = CustomersBig.CustomerID JOIN Order_DetailsBig ON Order_DetailsBig.OrderID = OrdersBig.OrderID JOIN ProductsBig ON Order_DetailsBig.ProductID = ProductsBig.ProductID WHERE OrdersBig.OrderDate BETWEEN '2025-10-15' AND '2025-12-15' ORDER BY CustomersBig.ContactName, Order_DetailsBig.Quantity </pre>
-------------------	---

Cenário 9

Neste cenário será testado o uso de filtros por valores nulos e o ganho ao criar índice em campos com valores nulos.

Tabela 16 - Consultas do Cenário 9

Consulta 1	<pre>SELECT CustomersBig.CompanyName, CustomersBig.ContactName, OrdersBig.Value, OrdersBig.OrderDate, ProductsBig.ProductName, Order_DetailsBig.Quantity FROM OrdersBig JOIN CustomersBig ON OrdersBig.CUSTOMERID = CustomersBig.CustomerID JOIN Order_DetailsBig ON Order_DetailsBig.OrderID = OrdersBig.OrderID JOIN ProductsBig ON Order_DetailsBig.ProductID = ProductsBig.ProductID WHERE OrdersBig.OrderDate BETWEEN '2025-10-15' AND '2025-12-15' AND Order_DetailsBig.Quantity IS NOT NULL ORDER BY CustomersBig.ContactName, Order_DetailsBig.Quantity</pre>
-------------------	---

Cenário 10

O Cenário 10 apresentará o uso da expressão CASE WHEN e como impacta na performance em ambos SGBDs.

Tabela 17 - Consultas do Cenário 10

Consulta 1	<pre> SELECT CustomersBig.CompanyName, CustomersBig.ContactName, OrdersBig.Value, OrdersBig.OrderDate, ProductsBig.ProductName, Order_DetailsBig.Quantity, CASE WHEN Order_DetailsBig.Quantity >= 1500 THEN 'Muitos Itens em Estoque' WHEN Order_DetailsBig.Quantity <= 10 THEN 'Necessário Repor Estoque' WHEN Order_DetailsBig.Quantity = 0 THEN 'Sem Estoque' ELSE 'Estoque Suficiente' END AS Alerta_Estoque FROM OrdersBig JOIN CustomersBig ON OrdersBig.CUSTOMERID = CustomersBig.CustomerID JOIN Order_DetailsBig ON Order_DetailsBig.OrderID = OrdersBig.OrderID JOIN ProductsBig ON Order_DetailsBig.ProductID = ProductsBig.ProductID WHERE OrdersBig.OrderDate BETWEEN '2025-10-15' AND '2025-12-15' AND Order_DetailsBig.Quantity IS NOT NULL ORDER BY CustomersBig.ContactName, Order_DetailsBig.Quantity </pre>
-------------------	--

Cenário 11

No Cenário 11 apresentará a performance do COUNT para ambos SGBDs e a diferença em ter o campo do COUNT indexado.

Tabela 18 - Consultas do Cenário 11

Consulta 1	<pre>SELECT ProductsBig.ProductName,Order_DetailsBig.Shipped_Date, Count(Order_DetailsBig.Quantity) AS Total_Data FROM Order_DetailsBig JOIN ProductsBig ON Order_DetailsBig.ProductID = ProductsBig.ProductID WHERE Order_DetailsBig.Shipped_Date BETWEEN '2023-10-15' AND '2026-03-20' GROUP BY Order_DetailsBig.Shipped_Date, ProductsBig.ProductName ORDER BY Total_data</pre>
-------------------	--

Cenário 12

Neste cenário será realizado teste com sub consultas, uma opção ao uso de junções e que em muitos casos pode ser a melhor solução, será validado a performance do uso das sub consultas e a diferença ao realizar uma sub consulta em colunas que possuem índices.

Tabela 19 - Consultas do Cenário 12

Consulta 1	<pre>SELECT CustomersBig.ContactName FROM CustomersBig WHERE CityID IS NULL AND EXISTS (SELECT * FROM OrdersBig WHERE OrdersBig.CustomerID = CustomersBig.CustomerID)</pre>
Consulta 2	<pre>SELECT CustomersBig.ContactName FROM CustomersBig WHERE CityID IS NULL AND NOT EXISTS (SELECT * FROM OrdersBig WHERE OrdersBig.CustomerID = CustomersBig.CustomerID)</pre>

Cenário 13

O Cenário 13 apresentará o uso do LIKE na condição do WHERE utilizando o “%” no final da busca, como se comporta sem índice e como se comporta com a criação de índice no campo do LIKE.

Tabela 20 - Consultas do Cenário 13

Consulta 1	<pre> SELECT CustomersBig.CompanyName, CustomersBig.ContactName, OrdersBig.Value, OrdersBig.OrderDate, ProductsBig.ProductName, Order_DetailsBig.Quantity, CASE WHEN Order_DetailsBig.Quantity >= 1500 THEN 'Muitos Itens em Estoque' WHEN Order_DetailsBig.Quantity <= 10 THEN 'Necessário Repor Estoque' WHEN Order_DetailsBig.Quantity = 0 THEN 'Sem Estoque' ELSE 'Estoque Suficiente' END AS Alerta_Estoque FROM OrdersBig JOIN CustomersBig ON OrdersBig.CUSTOMERID = CustomersBig.CustomerID JOIN Order_DetailsBig ON Order_DetailsBig.OrderID = OrdersBig.OrderID JOIN ProductsBig ON Order_DetailsBig.ProductID = ProductsBig.ProductID WHERE ProductsBig.ProductName LIKE 'Maxilaku%' AND Order_DetailsBig.Quantity IS NOT NULL ORDER BY CustomersBig.ContactName, Order_DetailsBig.Quantity </pre>
-------------------	---

Cenário 14

Seguindo o conceito do cenário anterior, no Cenário 14 será testado novamente o LIKE, porém com o uso do “%” no início da busca e testado se a criação de índice nesse caso apresenta ganhos e quais soluções cada SGBD nos fornece.

Tabela 21 - Consultas do Cenário 14

Consulta 1	<pre> SELECT CustomersBig.CompanyName, CustomersBig.ContactName, OrdersBig.Value, OrdersBig.OrderDate, ProductsBig.ProductName, Order_DetailsBig.Quantity, CASE WHEN Order_DetailsBig.Quantity >= 1500 THEN 'Muitos Itens em Estoque' WHEN Order_DetailsBig.Quantity <= 10 THEN 'Necessário Repor Estoque' WHEN Order_DetailsBig.Quantity = 0 THEN 'Sem Estoque' ELSE 'Estoque Suficiente' END AS Alerta_Estoque FROM OrdersBig JOIN CustomersBig ON OrdersBig.CUSTOMERID = CustomersBig.CustomerID JOIN Order_DetailsBig ON Order_DetailsBig.OrderID = OrdersBig.OrderID JOIN ProductsBig ON Order_DetailsBig.ProductID = ProductsBig.ProductID WHERE ProductsBig.ProductName LIKE '%7b' AND Order_DetailsBig.Quantity IS NOT NULL ORDER BY CustomersBig.ContactName, Order_DetailsBig.Quantity </pre>
-------------------	---

4.5.3 Índices Criados

Será realizado a criação de alguns índices com o intuito de demonstrar o ganho, independente do SGBD, ao se criar um índice. Os índices serão utilizados em diversos cenários e atenderam mais de uma consulta, a tabela a seguir apresenta os índices criados.

A base utilizada para criar os índices é uma cópia da base sem índices, porém com a adição dos índices. A base compartilhara a mesma instância onde já existe a base sem índice em cada ambiente de teste.

Tabela 22 - Tabelas e chaves dos índices criados

Índice	Tabela	Chave	Campo de Cobertura
Índice 1	OrdersBig	Customer_ID	
Índice 2	Order_DetailsBig	Quantity	
Índice 3	CustomersBig	CNPJ	
Índice 4	ProductsBig	ProductName	
Índice 5	OrdersBig	Order_Date	
Índice 6	Order_DetailsBig	Shipped_Date	
Índice 7	OrdersBig	Order_Date	Customer_ID, Value
Índice 8	Order_DetailsBig	Shipped_Date	Quantity
Índice 9	Order_DetailsBig	Product_ID,Shipped_Date	Quantity
Índice 10	CustomersBig	City_ID	Contact_Name

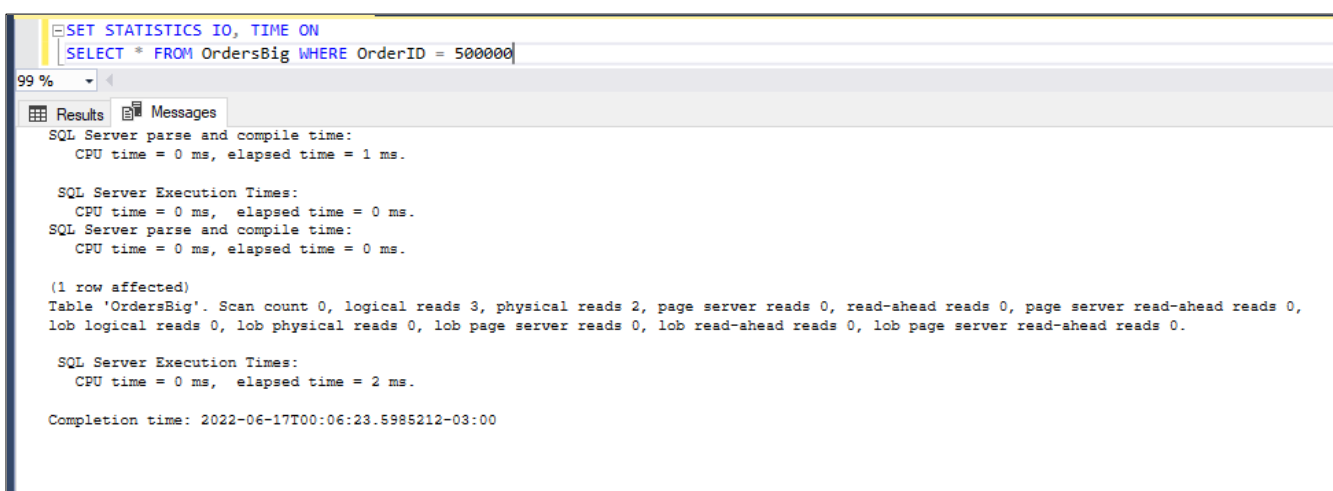
4.6 Realização dos Teste

Para realização dos testes, será iniciado um ambiente por vez, evitando assim consumo de recurso compartilhado por outras maquinas virtuais. Para os testes nos ambientes SQL Server, será utilizado o SQL Server Management Studio 18, a partir dele será conectado no ambiente e executa as consultas criadas, uma por vez, utilizando o parâmetro “SET

STATISTICS IO, TIME ON” para coletar a quantidade de páginas que precisaram ser lidas para a execução da consulta e o tempo que levou para ser executada.

Esse processo será executado no mínimo três vezes por consulta, pegando o valor médio, para que seja evitado casos pontuais de oscilação no banco, ou até situações em que a consulta não estava no cache e levou mais tempo para executar pois precisou ser inserida na memória.

A seguir pode-se ver o resultado de uma consulta com os parâmetros informados



```
SET STATISTICS IO, TIME ON
SELECT * FROM OrdersBig WHERE OrderID = 500000

Results Messages
99 %

SQL Server parse and compile time:
  CPU time = 0 ms, elapsed time = 1 ms.

SQL Server Execution Times:
  CPU time = 0 ms, elapsed time = 0 ms.
SQL Server parse and compile time:
  CPU time = 0 ms, elapsed time = 0 ms.

(1 row affected)
Table 'OrdersBig'. Scan count 0, logical reads 3, physical reads 2, page server reads 0, read-ahead reads 0, page server read-ahead reads 0,
lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

SQL Server Execution Times:
  CPU time = 0 ms, elapsed time = 2 ms.

Completion time: 2022-06-17T00:06:23.5985212-03:00
```

Figura 5 - Resultado da consulta com coleta de estatísticas de performance no SQLServer

Outro ponto que será observado durante a realização dos testes, é o plano de execução, onde é apresentado a trajetória que o SGBD vai realizar para buscar os dados solicitados. A partir do plano de execução será possível analisar e conseguir sugestões para os índices que serão criados. Abaixo é possível ver um exemplo de um plano de execução no SQL Server.

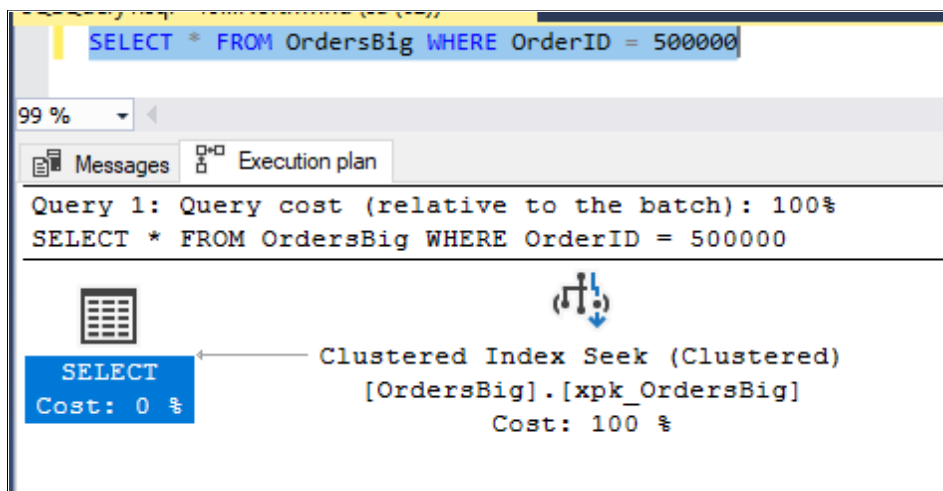


Figura 6 - Plano de execução no SQLServer

Para os testes que serão realizados no PostgreSQL, será realizado a partir da ferramenta PgAdmin 4, como ela é possível acessar o banco e realizar as consultas, assim como medir o número de páginas utilizadas e tempo de execução, para isso será utilizado o parâmetro “EXPLAIN (ANALYZE, BUFFER)”.

Todos os dados coletados serão armazenados em uma planilha, separando por cenário, consulta, SGBD, Sistema Operacional e se a base é com ou sem índices. Posteriormente será gerado gráficos para análise a partir dos dados coletados.

4.7 Resultado dos Testes

Será apresentado os resultados dos testes realizados, junto com observações sobre o desempenho obtido em cada cenário.

4.7.1 Cenário 1

Neste cenário é esperado um alto volume de leitura e uma demora acima dos demais cenários, por se tratar de um select na tabela inteira. O uso de índice nesse caso não iria mudar pois todos os dados contidos na tabela precisam ser varridos. Os dados coletados para o Cenário 1 seguem conforme as tabelas a seguir

Tabela 23 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 1

Tecnologia	S.O	Leituras Lógicas (Total)	Tempo de Execução(ms)
SQLServer	Windows	35905	117329
PostgreSQL	Ubuntu	63599	234562
PostgreSQL	Windows	61391	300450
SQLServer	Ubuntu	35905	75060

Tabela 24 - Coleta de leituras logicas e tempo de execução da consulta 2 do cenário 1

Tecnologia	S.O	Leituras Lógicas (Total)	Tempo de Execução(ms)
SQLServer	Windows	35566	74015
PostgreSQL	Ubuntu	53519	6181
PostgreSQL	Windows	53071	16822
SQLServer	Ubuntu	35566	77134

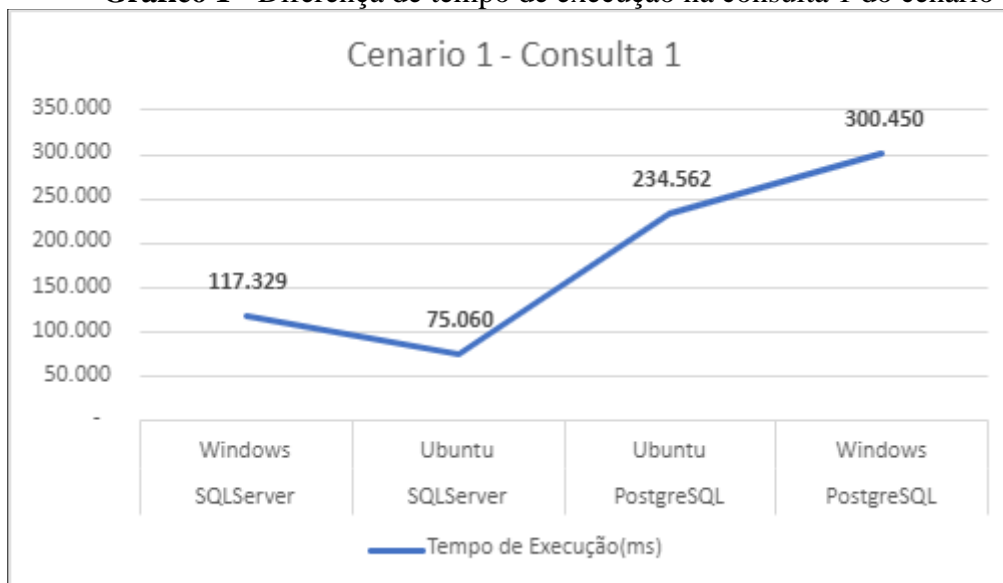
Tabela 25 - Coleta de leituras lógicas e tempo de execução da consulta 3 do cenário 1

Tecnologia	S.O	Leituras Lógicas (Total)	Tempo de Execução(ms)
SQLServer	Windows	187195	6313
PostgreSQL	Ubuntu	85972	9390
PostgreSQL	Windows	85907	37000
SQLServer	Ubuntu	187195	70023

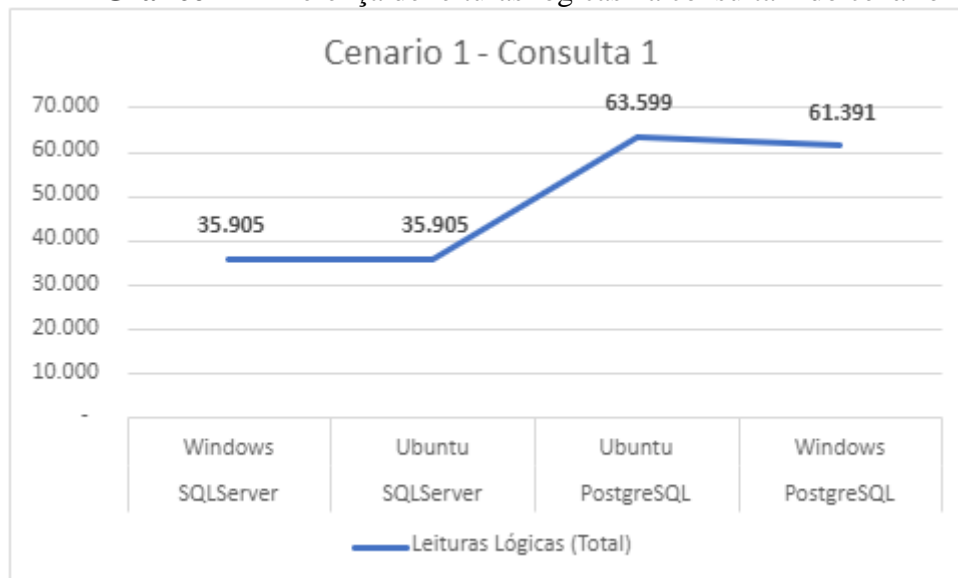
Tabela 26 - Coleta de leituras lógicas e tempo de execução da consulta 4 do cenário 1

Tecnologia	S.O	Leituras Lógicas (Total)	Tempo de Execução(ms)
SQLServer	Windows	502	381
PostgreSQL	Ubuntu	610	10
PostgreSQL	Windows	610	274
SQLServer	Ubuntu	502	327

Pode-se observar conforme o gráfico a seguir que se teve um alto tempo de execução em todos os casos, porém observa-se vantagem no SQL Server e ainda mais no SQL Server que é executado no sistema operacional Ubuntu.

Gráfico 1 - Diferença de tempo de execução na consulta 1 do cenário 1

Essa vantagem maior entre o SGBD SQL Server e PostgreSQL apresenta ter relação com a quantidade de páginas lidas conforme o gráfico a seguir

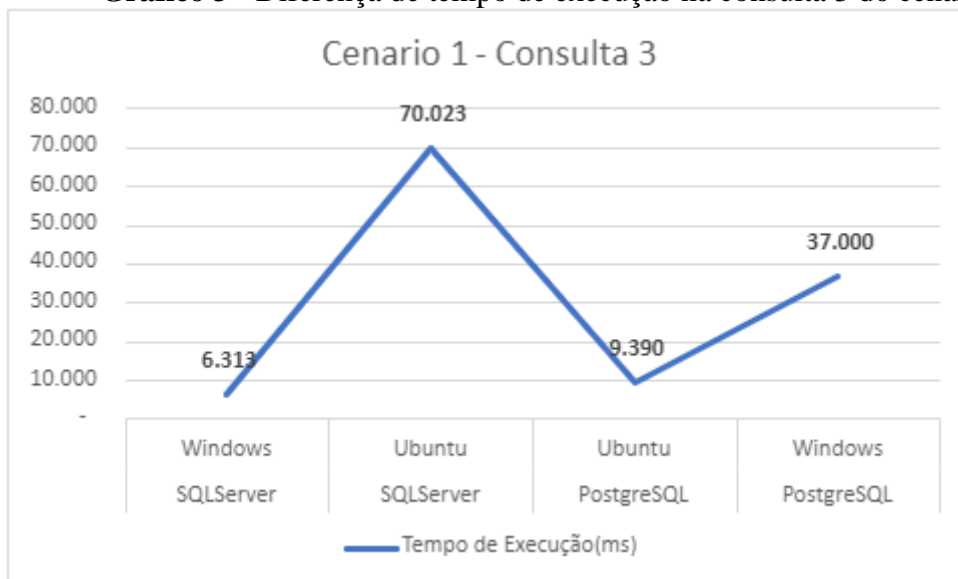
Gráfico 2 - Diferença de leituras lógicas na consulta 1 do cenário 1

Percebe-se também que o SQLServer mantém a mesma quantidade de leitura de páginas independente do Sistema Operacional, o que indica ter uma arquitetura de

armazenamento muito similar em ambos os sistemas operacionais, já no PostgreSQL se perceber uma leve variação.

Na consulta três, onde se tem menos dados retornando, porém se trata de uma tabela que possui campos do tipo Identificador Exclusivo que se trata de uma cadeia de caracteres no formato xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx, em que cada x é um dígito hexadecimal no intervalo 0–9 ou a–f. Isso faz com que a coluna acabe sendo maior que das outras tabelas e pese mais para retornar o resultado. Nesse quesito se tem o SQL Server no Windows e o PostgreSQL no Ubuntu performando muito melhor do que os mesmos nos outros sistemas operacionais.

Gráfico 3 - Diferença de tempo de execução na consulta 3 do cenário 1



4.7.2 Cenário 2

No Cenário 2 todos os ambientes tiveram bom desempenho, nesse como se trata de buscar em chave primária, nos dois SGBDs por padrão a chave primária é um índice, mesmo que não seja especificado.

Tabela 27 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 2

Tecnologia	S.O	Leituras Lógicas (Total)	Tempo de Execução(ms)
SQLServer	Windows	3	3
PostgreSQL	Ubuntu	4	0
PostgreSQL	Windows	4	4
SQLServer	Ubuntu	3	0

Tabela 28 - Coleta de leituras logicas e tempo de execução da consulta 2 do cenário 2

Tecnologia	S.O	Leituras Lógicas (Total)	Tempo de Execução(ms)
SQLServer	Windows	3	0
PostgreSQL	Ubuntu	4	3
PostgreSQL	Windows	3	7
SQLServer	Ubuntu	3	0

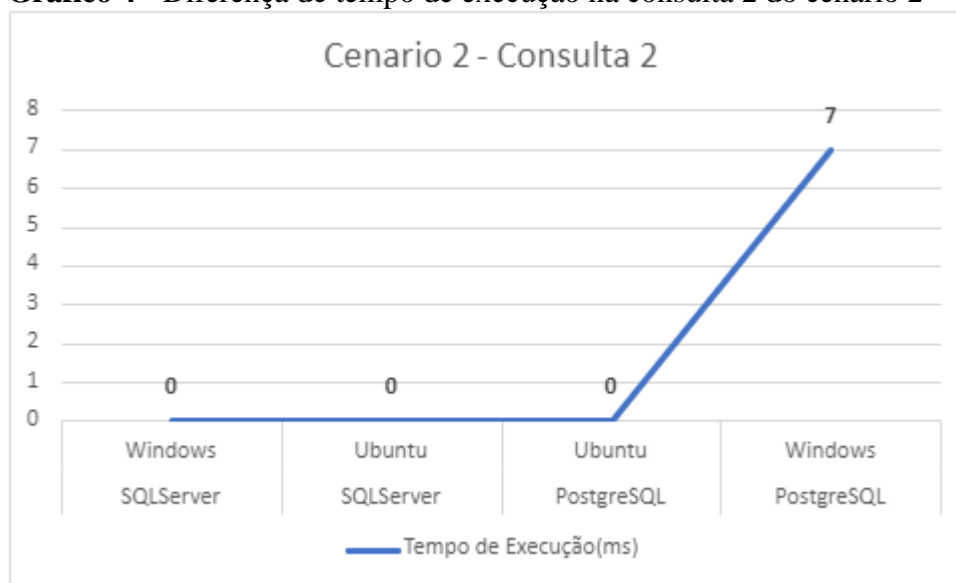
Tabela 29 - Coleta de leituras logicas e tempo de execução da consulta 3 do cenário 2

Tecnologia	S.O	Leituras Lógicas (Total)	Tempo de Execução(ms)
SQLServer	Windows	3	0
PostgreSQL	Ubuntu	4	3
PostgreSQL	Windows	4	8
SQLServer	Ubuntu	3	0

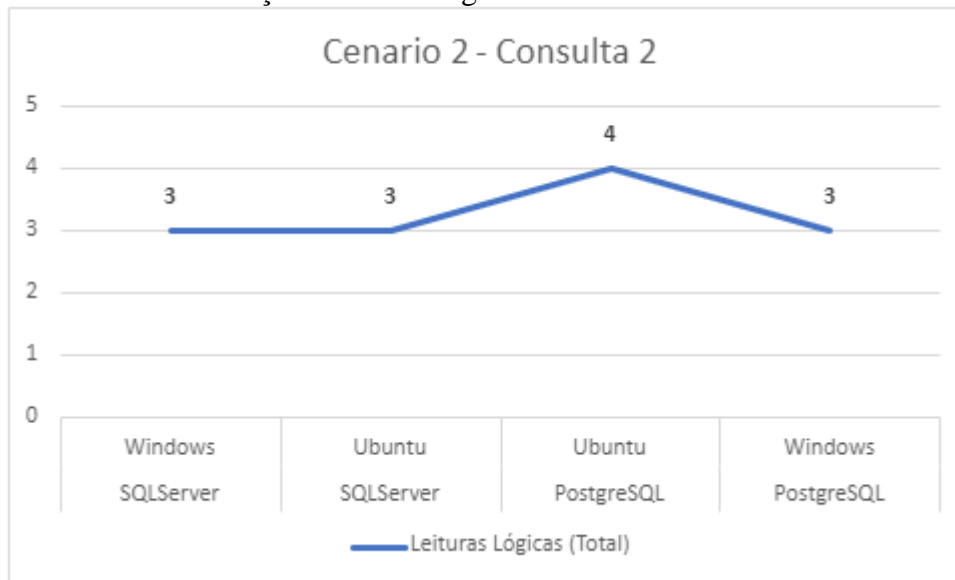
Tabela 30 - Coleta de leituras lógicas e tempo de execução da consulta 4 do cenário 2

Tecnologia	S.O	Leituras Lógicas (Total)	Tempo de Execução(ms)
SQLServer	Windows	2	0
PostgreSQL	Ubuntu	2	2
PostgreSQL	Windows	4	7
SQLServer	Ubuntu	2	0

Pode-se perceber apenas uma leve diferença no ambiente PostgreSQL no sistema operacional Windows, mas levando em consideração que se está medindo em milissegundos, não tem impacto significativo.

Gráfico 4 - Diferença de tempo de execução na consulta 2 do cenário 2

Percebe-se também, que em relação a quantidade de páginas lidas, é um número muito baixo em todos os ambientes, pois o banco sabe exatamente qual página de dado precisa ler.

Gráfico 5 - Diferença de leituras lógicas na consulta 2 do cenário 2

4.7.3 Cenário 3

Neste cenário é perceptível a diferença ao executar uma consulta sem índice e com índice. Dificilmente uma base sem índice vai se sobressair sobre a base com índice, independente do SGBD ou do Sistema Operacional usado.

Tabela 31 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 3

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	881	36139
SQLServer	Windows	Com índices	0	21
SQLServer	Ubuntu	Sem índices	342	36139
SQLServer	Ubuntu	Com índices	0	21
PostgreSQL	Ubuntu	Sem índices	706	63471
PostgreSQL	Ubuntu	Com índices	123	9
PostgreSQL	Windows	Sem índices	976	61519
PostgreSQL	Windows	Com índices	75	6

Tabela 32 - Coleta de leituras logicas e tempo de execução da consulta 2 do cenário 3

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	1269	35868
SQLServer	Windows	Com índices	297	23733
SQLServer	Ubuntu	Sem índices	453	35868
SQLServer	Ubuntu	Com índices	75	13958
PostgreSQL	Ubuntu	Sem índices	665	53487
PostgreSQL	Ubuntu	Com índices	450	4356
PostgreSQL	Windows	Sem índices	937	53007
PostgreSQL	Windows	Com índices	296	4357

Tabela 33 - Coleta de leituras logicas e tempo de execução da consulta 3 do cenário 3

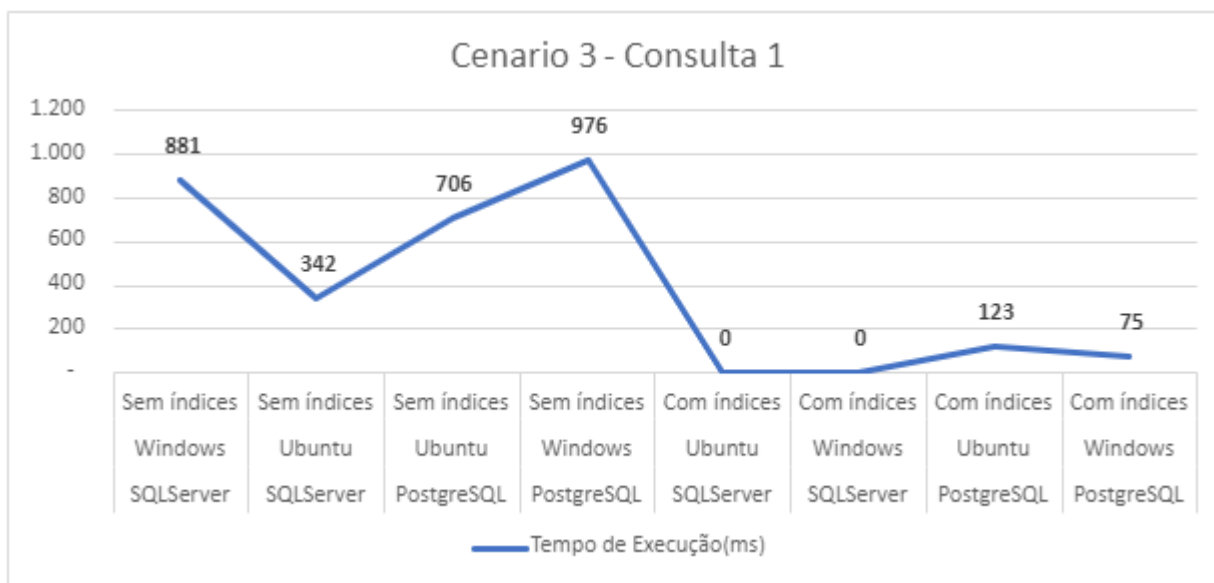
Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	390	188.997
SQLServer	Windows	Com índices	0	6
SQLServer	Ubuntu	Sem índices	503	188997
SQLServer	Ubuntu	Com índices	0	6
PostgreSQL	Ubuntu	Sem índices	818	85940
PostgreSQL	Ubuntu	Com índices	0	7
PostgreSQL	Windows	Sem índices	1.072	85875
PostgreSQL	Windows	Com índices	0	7

Tabela 34 - Coleta de leituras logicas e tempo de execução da consulta 4 do cenário 3

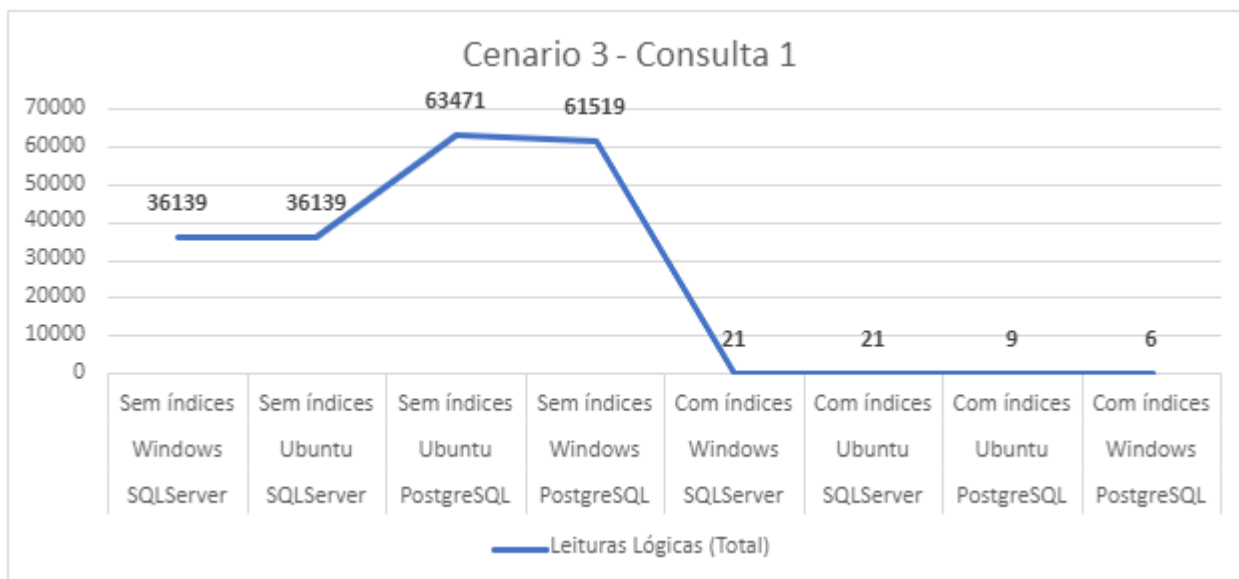
Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	5	502
SQLServer	Windows	Com índices	0	5
SQLServer	Ubuntu	Sem índices	5	502
SQLServer	Ubuntu	Com índices	0	5
PostgreSQL	Ubuntu	Sem índices	6	610
PostgreSQL	Ubuntu	Com índices	0	3
PostgreSQL	Windows	Sem índices	73	610
PostgreSQL	Windows	Com índices	0	4

No gráfico a seguir é possível ver claramente a diferença de tempo de execução nas bases com índice, se considerar a necessidade de a consulta executar, por exemplo, 10 vezes por segundo, sem utilização do índice isso não seria possível.

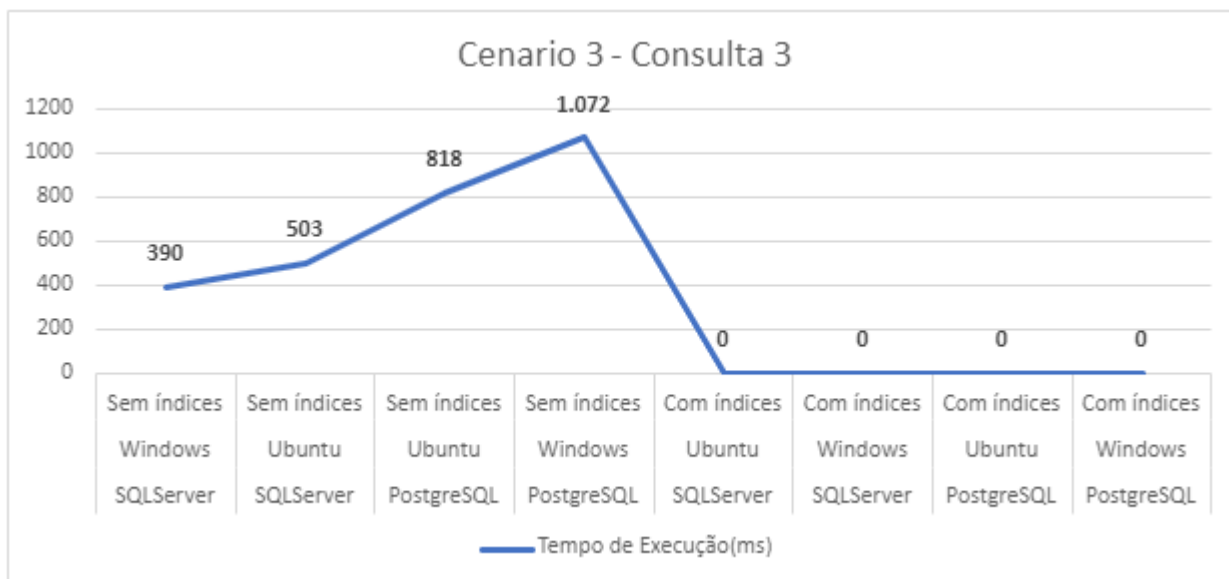
Gráfico 6 - Diferença de tempo de execução na consulta 1 do cenário 3



Pode-se perceber também, que não se tem somente uma redução no tempo de execução, mas também uma redução considerável no número de páginas que precisam ser lidas para retornar o resultado das consultas

Gráfico 7 - Diferença de leituras lógicas na consulta 1 do cenário 3

No gráfico número x referente a consulta 3 também nota-se que o SQL Server no sistema Windows, mesmo sem índice, apresentou um tempo satisfatório comparado aos demais ambientes.

Gráfico 8 - Diferença de tempo de execução na consulta 3 do cenário 3

4.7.4 Cenário 4

Percebe-se muitas variações no cenário 4, pois como se trata de filtros utilizando datas, o resultado pode ser muito diferente dependendo da quantidade de registros que a consulta irá retornar. Um filtro de um mês pode ser muito ou pouco, tudo depende de quantos dados estão dentro daquele mês, muitas vezes o próprio SGBD irá analisar que existem muitos dados para retornar e optará por descartar o uso do índice e ler toda a tabela. Por conta disso se pode perceber pouco ou nenhum efeito em algumas consultas comparando a base com e sem índice.

Tabela 35 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 4

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	1.134	36.127
SQLServer	Windows	Com índices	0	11
SQLServer	Ubuntu	Sem índices	714	36139
SQLServer	Ubuntu	Com índices	6	11
PostgreSQL	Ubuntu	Sem índices	685	63375
PostgreSQL	Ubuntu	Com índices	4	2309
PostgreSQL	Windows	Sem índices	1.057	63183
PostgreSQL	Windows	Com índices	3	2304

Tabela 36 - Coleta de leituras logicas e tempo de execução da consulta 2 do cenário 4

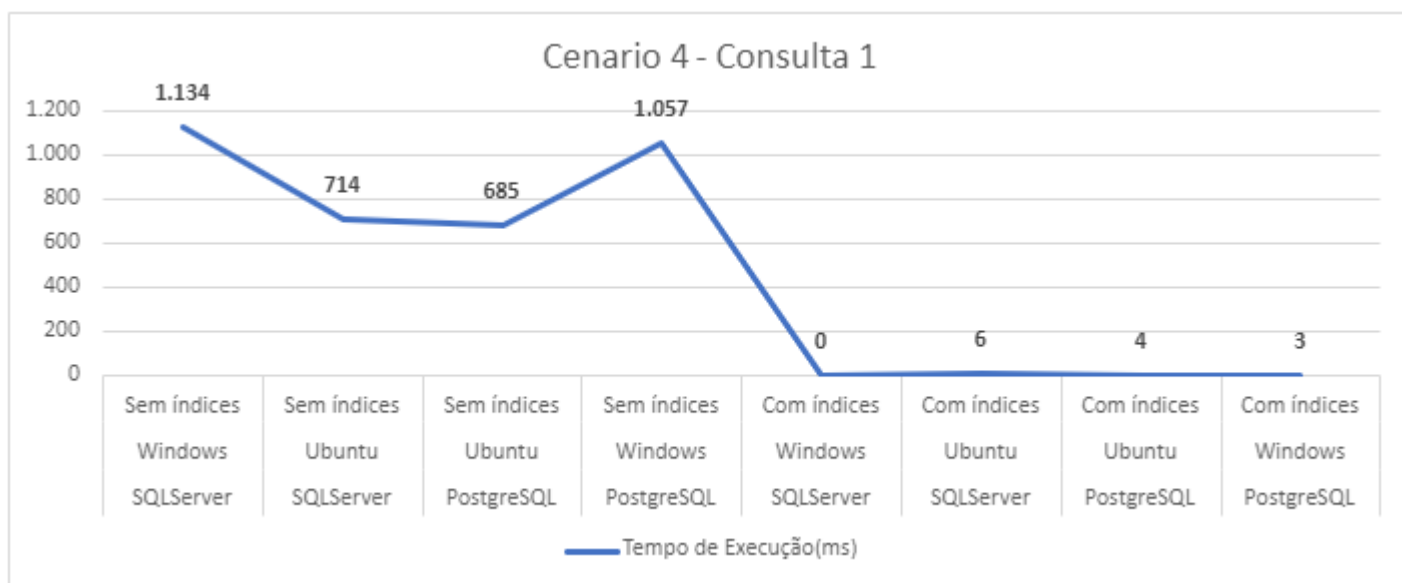
Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	391	35868
SQLServer	Windows	Com índices	7	11
SQLServer	Ubuntu	Sem índices	711	35868
SQLServer	Ubuntu	Com índices	7	11
PostgreSQL	Ubuntu	Sem índices	682	53390
PostgreSQL	Ubuntu	Com índices	24	1992
PostgreSQL	Windows	Sem índices	879	52814
PostgreSQL	Windows	Com índices	188	1990

Tabela 37 - Coleta de leituras logicas e tempo de execução da consulta 3 do cenário 4

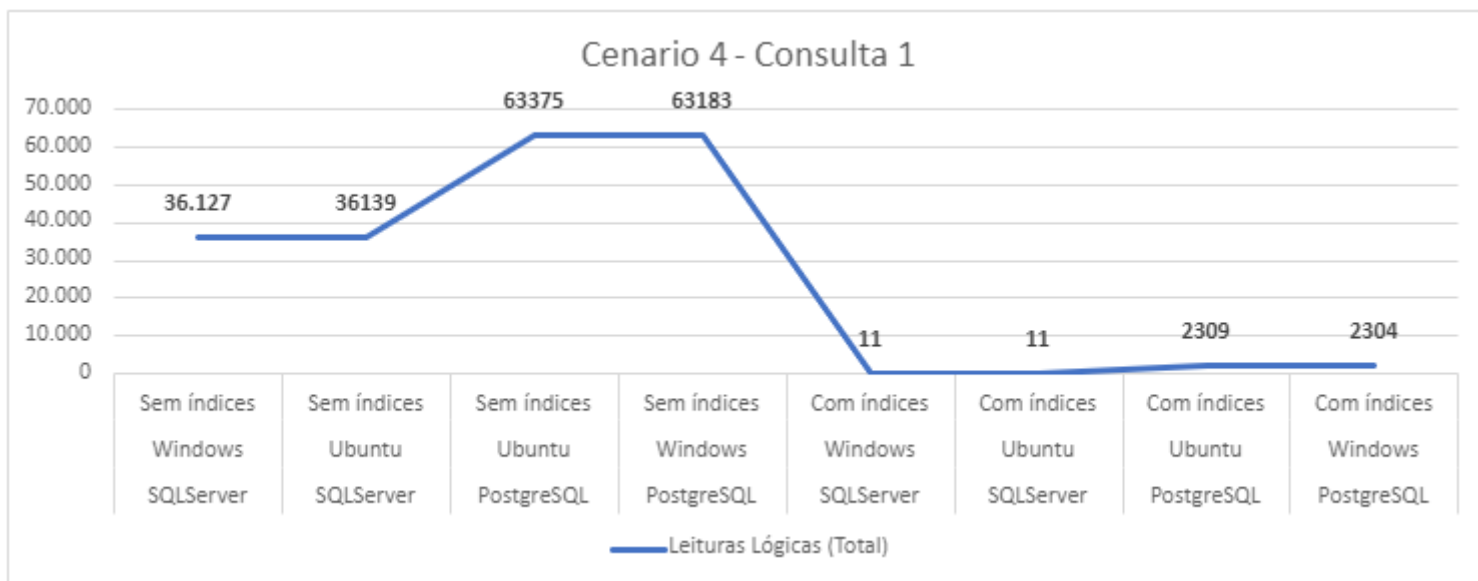
Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	25.273	36063
SQLServer	Windows	Com índices	19182	9285
SQLServer	Ubuntu	Sem índices	19634	36043
SQLServer	Ubuntu	Com índices	19191	9285
PostgreSQL	Ubuntu	Sem índices	7657	63279
PostgreSQL	Ubuntu	Com índices	7651	66046
PostgreSQL	Windows	Sem índices	8.754	62991
PostgreSQL	Windows	Com índices	7.628	68560

Na consulta 1 se pode perceber o ganho efetivo do uso dos índices, também Percebe-se que sem o índice o retorno nos sistemas operacionais Ubuntu foi mais efetivo.

Gráfico 9 - Diferença de tempo de execução na consulta 1 do cenário 4



A diferença na quantidade de leituras de página justifica o ganho de tempo de consultas

Gráfico 10 - Diferença de tempo de execução na consulta 1 do cenário 4

4.7.5 Cenário 5

Tem-se nesse cenário uma demonstração o uso de JOINS, realizando em campo que são chaves primárias, na consulta 1 percebe-se um esforço maior para buscar os dados, or conta da grande quantidade de leituras na tabela CustomersBig, o SQL Server em ambos os sistemas operacionais ficou com valores muito próximos se destacando dos demais ambientes. Mesma coisa acontece na consulta 2, onde é necessário buscar um volume um pouco maior de dados. Nas consultas 3 e 4 onde foi possível ir direto na página requisitada em todos os ambientes o retorno foi imediato.

Tabela 38 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 5

Tecnologia	S.O	Leituras Lógicas (Total)	Tempo de Execução(ms)
SQLServer	Windows	36144	458
PostgreSQL	Ubuntu	63254	938
PostgreSQL	Windows	62935	1188
SQLServer	Ubuntu	36144	501

Tabela 39 - Coleta de leituras logicas e tempo de execução da consulta 2 do cenário 5

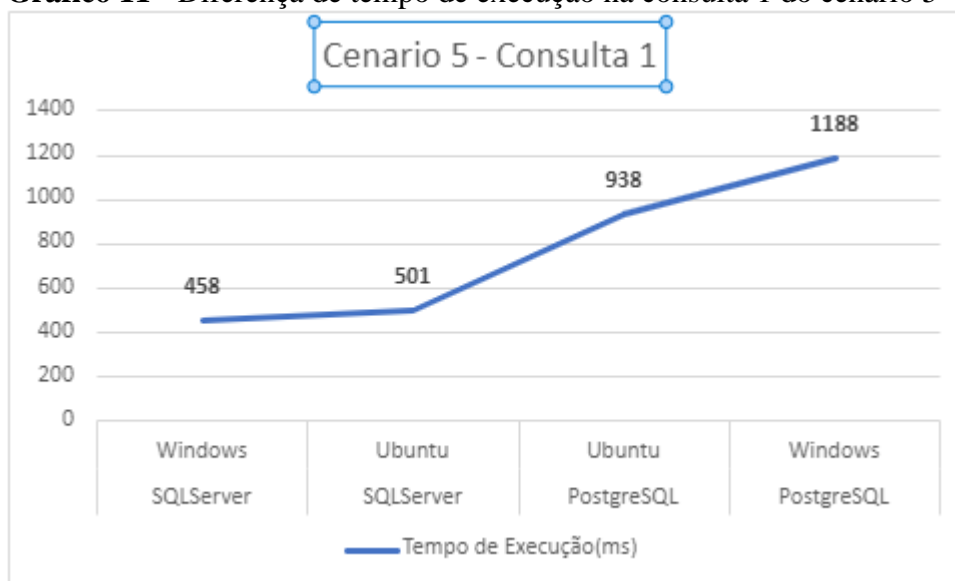
Tecnologia	S.O	Leituras Lógicas (Total)	Tempo de Execução(ms)
SQLServer	Windows	35870	325
PostgreSQL	Ubuntu	53197	646
PostgreSQL	Windows	52622	903
SQLServer	Ubuntu	35870	358

Tabela 40 - Coleta de leituras logicas e tempo de execução da consulta 3 do cenário 5

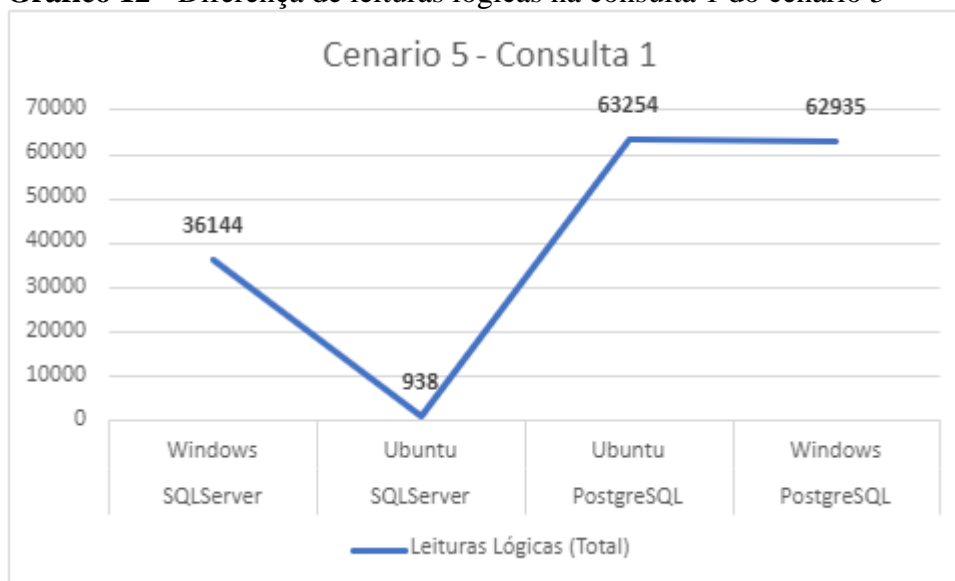
Tecnologia	S.O	Leituras Lógicas (Total)	Tempo de Execução(ms)
SQLServer	Windows	6	0
PostgreSQL	Ubuntu	4	0
PostgreSQL	Windows	5	0
SQLServer	Ubuntu	6	0

Tabela 41 - Coleta de leituras logicas e tempo de execução da consulta 4 do cenário 5

Tecnologia	S.O	Leituras Lógicas (Total)	Tempo de Execução(ms)
SQLServer	Windows	19	0
PostgreSQL	Ubuntu	92	0
PostgreSQL	Windows	98	0
SQLServer	Ubuntu	19	0

Gráfico 11 - Diferença de tempo de execução na consulta 1 do cenário 5

Tem-se um ponto interessante nesse cenário, onde o SQLServer no Ubuntu realizou menos leituras do que nos demais ambientes, mas mesmo assim não desempenhou tão rápido quanto O SQLServer no Windows, apesar de chegar muito próximo.

Gráfico 12 - Diferença de leituras lógicas na consulta 1 do cenário 5

4.7.6 Cenário 6

Neste cenário temos exemplos de consulta que mesmo utilizando índice e filtros, vão levar mais tempo e necessitar de muita leitura, por buscar um grande volume de dados em várias tabelas. Pode-se destacar as consultas 1 e 2 que trazem mais de 300 mil registros como resultado da busca.

Tabela 42 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 6

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	3.735	223.704
SQLServer	Windows	Com índices	5394	188761
SQLServer	Ubuntu	Sem índices	4061	223.704
SQLServer	Ubuntu	Com índices	3112	188761
PostgreSQL	Ubuntu	Sem índices	4.678	198272
PostgreSQL	Ubuntu	Com índices	4229	174749
PostgreSQL	Windows	Sem índices	5.725	296062
PostgreSQL	Windows	Com índices	4.577	85779

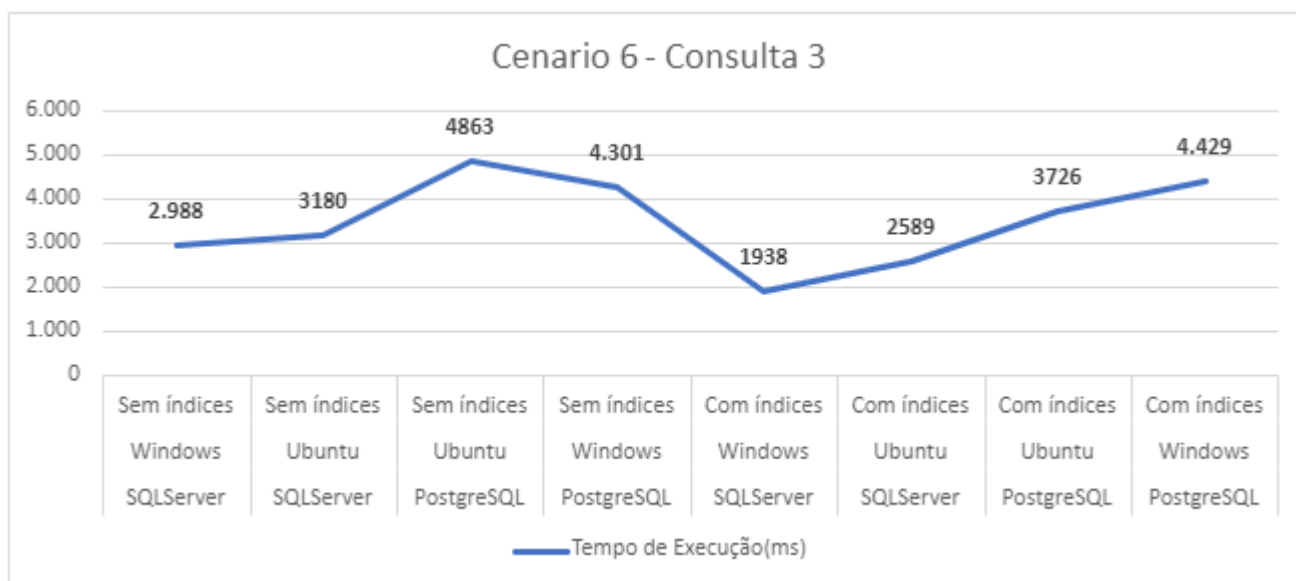
Tabela 43 - Coleta de leituras logicas e tempo de execução da consulta 2 do cenário 6

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	4.970	71.863
SQLServer	Windows	Com índices	1192	1417
SQLServer	Ubuntu	Sem índices	2658	71.863
SQLServer	Ubuntu	Com índices	1742	1417
PostgreSQL	Ubuntu	Sem índices	2783	53101
PostgreSQL	Ubuntu	Com índices	1572	51400
PostgreSQL	Windows	Sem índices	5.730	228216
PostgreSQL	Windows	Com índices	2.293	50801

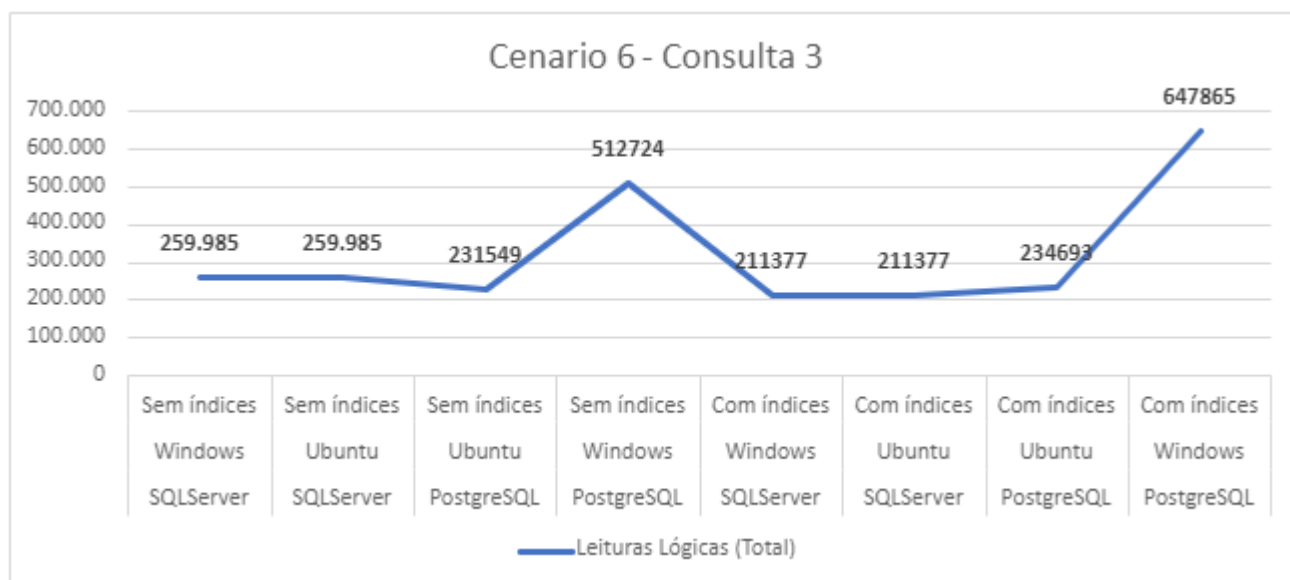
Tabela 44 - Coleta de leituras lógicas e tempo de execução da consulta 3 do cenário 6

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	2.988	259.985
SQLServer	Windows	Com índices	1938	211377
SQLServer	Ubuntu	Sem índices	3180	259.985
SQLServer	Ubuntu	Com índices	2589	211377
PostgreSQL	Ubuntu	Sem índices	4863	231549
PostgreSQL	Ubuntu	Com índices	3680	234693
PostgreSQL	Windows	Sem índices	4.301	512724
PostgreSQL	Windows	Com índices	4.429	647865

Conforme o próximo gráfico, pode-se perceber que se tem um pequeno ganho com a criação de índices nos ambientes, mas comparando o ambiente PostgreSQL com índice, percebe-se que teve uma demora maior do que todos os ambientes SQL Server, mesmo os que não possuíam índice. Isso ocorre por conta da tomada de decisão do algoritmo de cada SGBD em situações onde é necessário buscar e retornar um grande volume de dados.

Gráfico 13 - Diferença de tempo de execução na consulta 3 do cenário 6

Pode-se perceber também neste caso um volume bem maior de páginas a serem lidas nos ambientes PostgreSQL no Windows Server.

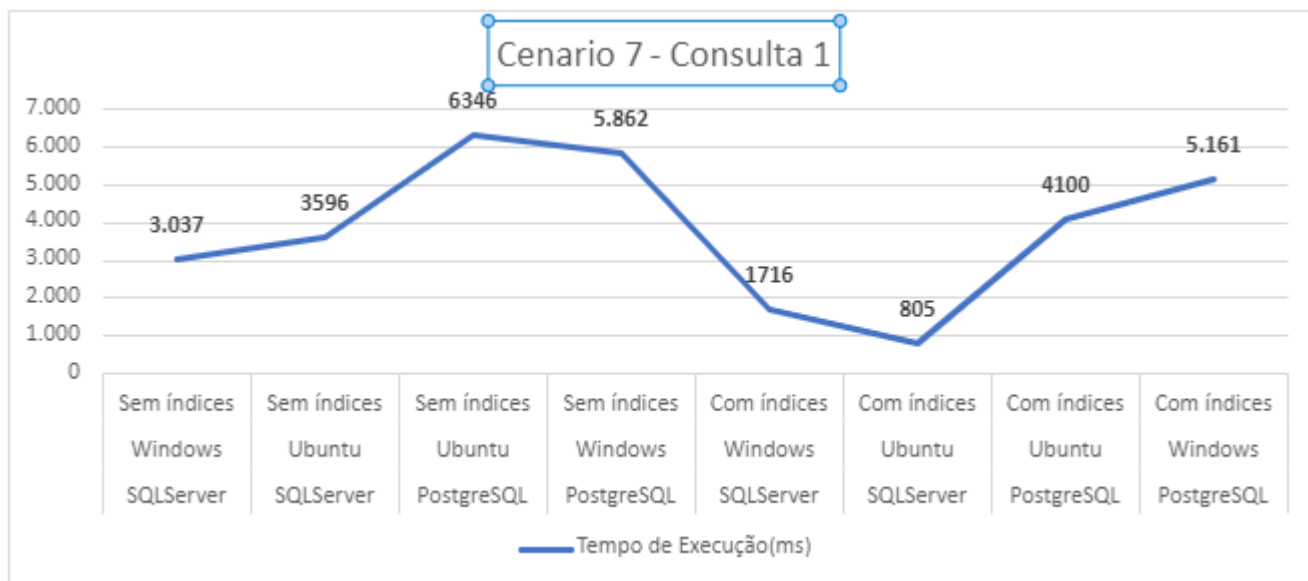
Gráfico 14 - Diferença de número de leituras lógicas na consulta 3 do cenário 6

4.7.7 Cenário 7

No cenário 7 percebe-se uma vantagem considerável no ambiente SQL Server com Ubuntu utilizando índices, e em um contexto geral uma vantagem na utilização do índice que foi criado para cobrir o campo solicitado no Order by.

Tabela 45 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 7

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	3.037	260.481
SQLServer	Windows	Com índices	1716	212857
SQLServer	Ubuntu	Sem índices	3596	260.481
SQLServer	Ubuntu	Com índices	805	212857
PostgreSQL	Ubuntu	Sem índices	6346	518038
PostgreSQL	Ubuntu	Com índices	4100	647712
PostgreSQL	Windows	Sem índices	5.862	425892
PostgreSQL	Windows	Com índices	5.161	593060

Gráfico 15 - Diferença de tempo de execução na consulta 1 do cenário 7

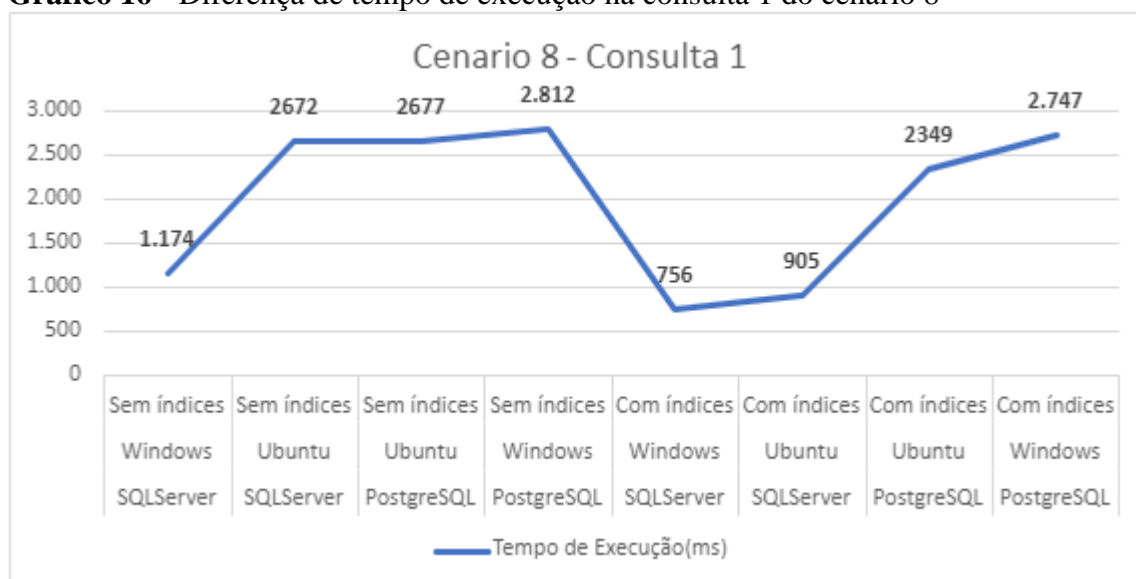
4.7.8 Cenário 8

Aqui é esperado um retorno mais rápido, independente do cenário, pois é utilizado funções internas de cada SGBD para limitar a quantidade de registros retornados. Os SGBDs vão estimar uma consulta completa, porém vão parar a buscar quando atingir o limite solicitado.

Tabela 46 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 8

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	1.174	260.857
SQLServer	Windows	Com índices	756	212857
SQLServer	Ubuntu	Sem índices	2672	260.857
SQLServer	Ubuntu	Com índices	905	212857
PostgreSQL	Ubuntu	Sem índices	2677	219384
PostgreSQL	Ubuntu	Com índices	2349	429762
PostgreSQL	Windows	Sem índices	2.812	355056
PostgreSQL	Windows	Com índices	2.747	773175

Pode-se perceber que o uso do índice fez diferença somente no SQL Server, no PostgreSQL não teve uma mudança considerável. E de todos os ambientes o TOP no SQLServer com Windows se saiu melhor.

Gráfico 16 - Diferença de tempo de execução na consulta 1 do cenário 8

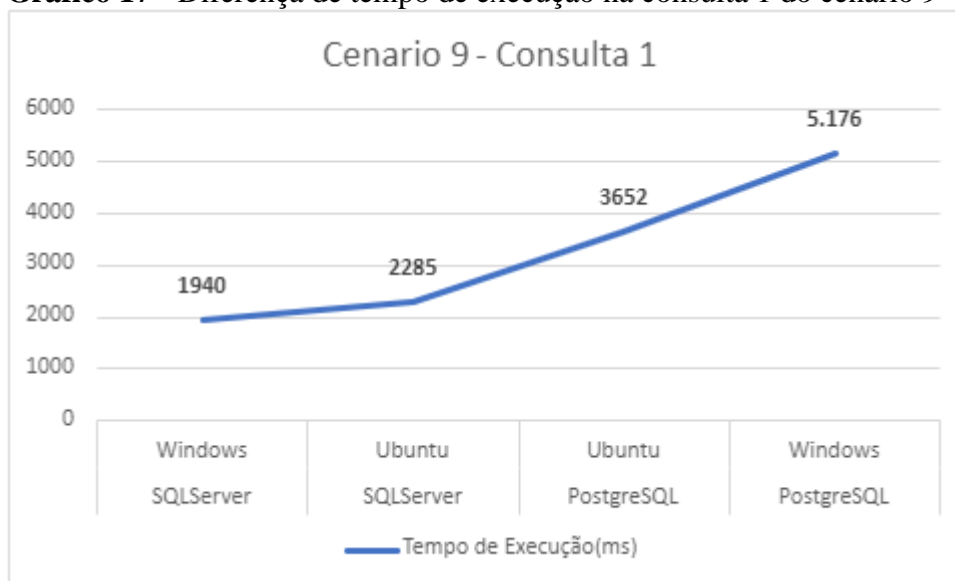
4.7.9 Cenário 9

Realizando o filtro com NULLs, se consegue descartar uma quantidade considerável de dados, mas não será possível ir direto nas páginas requeridas sem fazer uma busca mais abrangente.

Tabela 47 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 9

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	2.673	260.301
SQLServer	Windows	Com índices	1940	212857
SQLServer	Ubuntu	Sem índices	2998	260.301
SQLServer	Ubuntu	Com índices	2285	212857
PostgreSQL	Ubuntu	Sem índices	4992	516721
PostgreSQL	Ubuntu	Com índices	3652	448064
PostgreSQL	Windows	Sem índices	5.631	507855
PostgreSQL	Windows	Com índices	5.176	899412

Novamente se tem vantagem do SQL Server, o gráfico a seguir demonstra somente o resultado final com índices.

Gráfico 17 - Diferença de tempo de execução na consulta 1 do cenário 9

4.7.10 Cenário 10

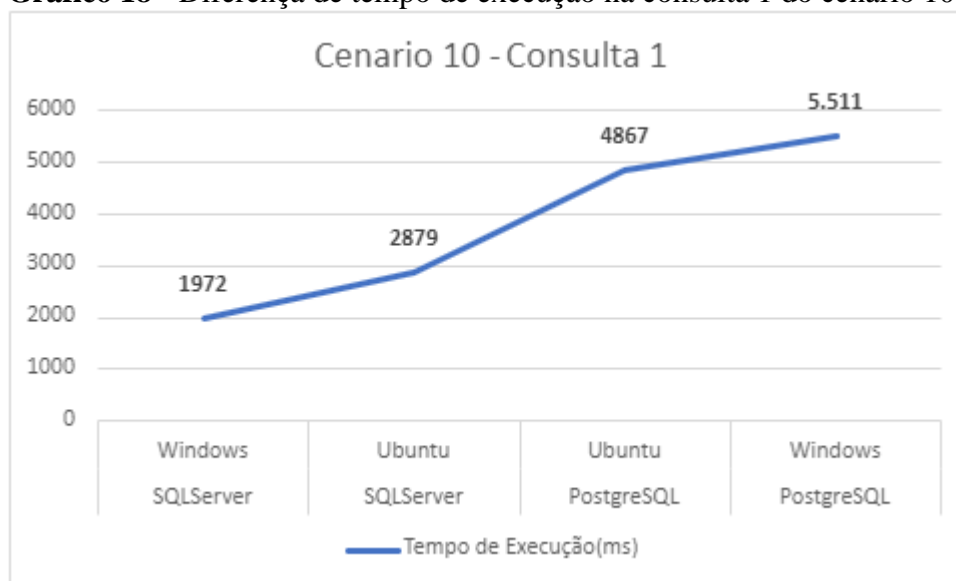
No uso do CASE, não se tem muita diferença nos resultados, a função não afeta a performance em nenhum dos casos, por trabalhar em cima dos dados que já teria que ser buscados.

Tabela 48 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 10

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	2.954	261.409
SQLServer	Windows	Com índices	1972	212857
SQLServer	Ubuntu	Sem índices	4624	261.409
SQLServer	Ubuntu	Com índices	2879	212857
PostgreSQL	Ubuntu	Sem índices	5182	515953
PostgreSQL	Ubuntu	Com índices	4530	841328
PostgreSQL	Windows	Sem índices	5.830	507087
PostgreSQL	Windows	Com índices	5.511	503971

Pode-se observar no gráfico de comparação das bases com índice, que novamente tem-se vantagem para o SGBD SQLServer

Gráfico 18 - Diferença de tempo de execução na consulta 1 do cenário 10



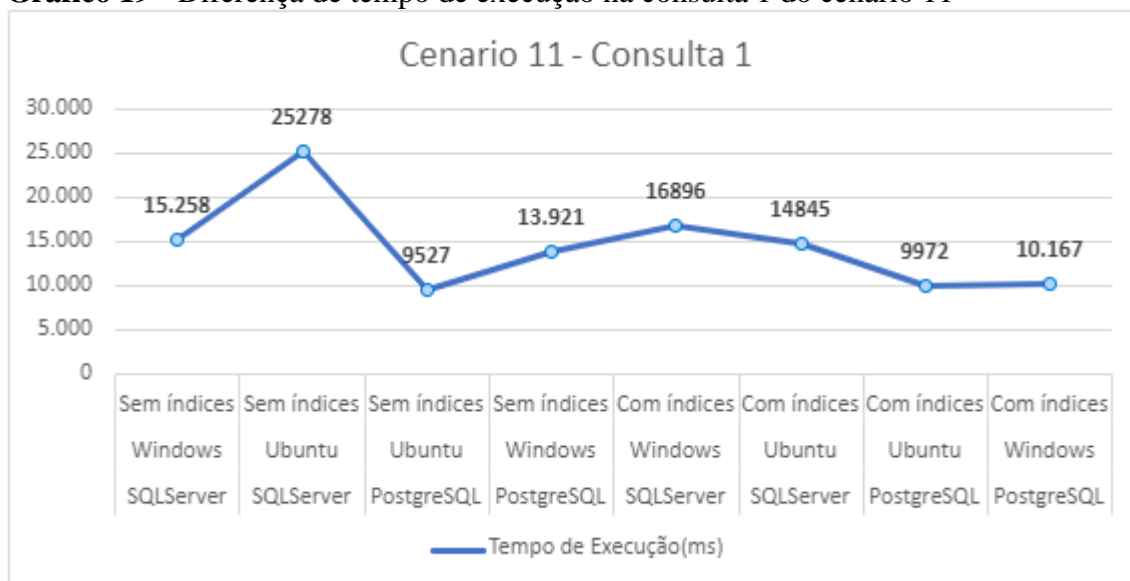
4.7.11 Cenário 11

O uso do count adiciona uma operação a mais para ser processada no momento da execução da consulta, nesse cenário o PostgreSQL se mostrou mais eficiente.

Tabela 49 - Coleta de leituras logicas e tempo de execução da consulta 1 do cenário 11

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	15.258	36.385
SQLServer	Windows	Com índices	16896	7703
SQLServer	Ubuntu	Sem índices	25278	36.385
SQLServer	Ubuntu	Com índices	14845	7703
PostgreSQL	Ubuntu	Sem índices	9527	52204
PostgreSQL	Ubuntu	Com índices	9972	113599
PostgreSQL	Windows	Sem índices	13.921	88233
PostgreSQL	Windows	Com índices	10.167	113599

Conforme demonstra o gráfico, com ou sem utilização de índices, nesse cenário o PostgreSQL foi superior ao SQLServer

Gráfico 19 - Diferença de tempo de execução na consulta 1 do cenário 11

4.7.12 Cenário 12

Como alternativa ao uso de JOINS e um dos métodos bastante utilizados antigamente, a subconsulta apresenta resultados diferentes de acordo com os ambientes.

Tabela 50 - Coleta de leituras lógicas e tempo de execução da consulta 1 do cenário 12

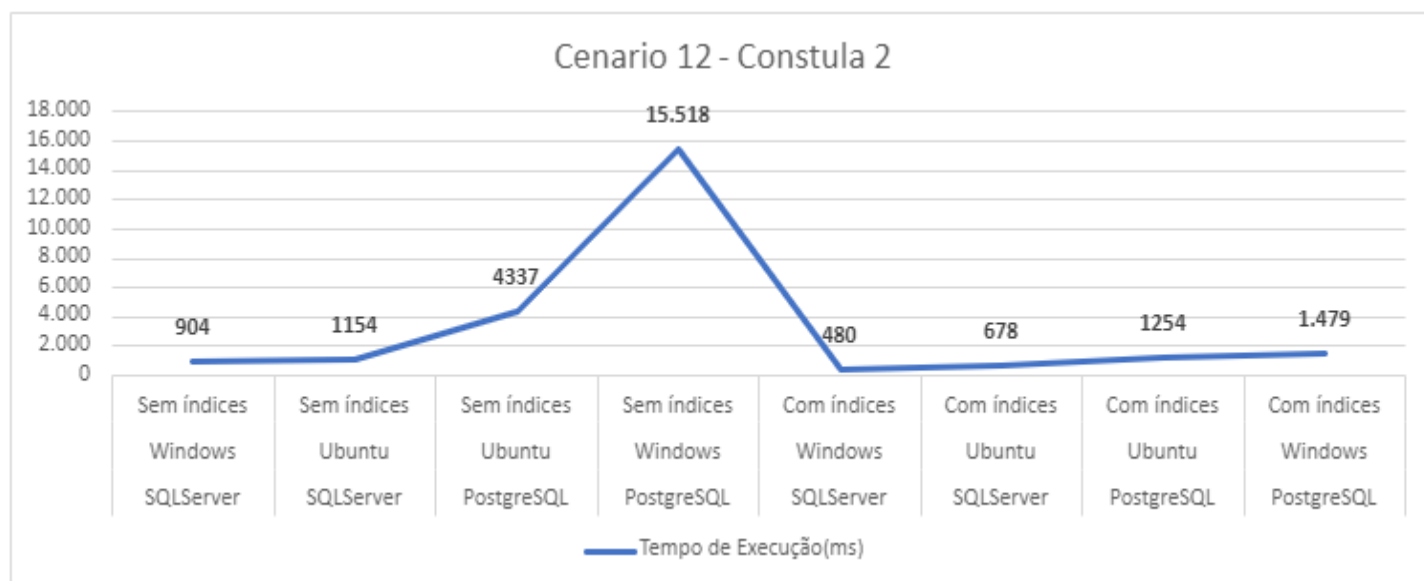
Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	2.988	225.136
SQLServer	Windows	Com índices	989	4577
SQLServer	Ubuntu	Sem índices	2935	225.136
SQLServer	Ubuntu	Com índices	1542	4577
PostgreSQL	Ubuntu	Sem índices	3972	295104
PostgreSQL	Ubuntu	Com índices	1885	176484
PostgreSQL	Windows	Sem índices	11.536	178532

PostgreSQL	Windows	Com índices	2.597	176484
------------	---------	-------------	-------	--------

Tabela 51 - Coleta de leituras logicas e tempo de execução da consulta 2 do cenário 12

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	904	225.136
SQLServer	Windows	Com índices	480	4577
SQLServer	Ubuntu	Sem índices	1154	225.136
SQLServer	Ubuntu	Com índices	678	4577
PostgreSQL	Ubuntu	Sem índices	4337	294720
PostgreSQL	Ubuntu	Com índices	1254	85779
PostgreSQL	Windows	Sem índices	15.518	178099
PostgreSQL	Windows	Com índices	1.479	85779

No gráfico a seguir se pode perceber uma efetividade dos índices em todos os cenários, e que o PostgreSQL no Windows não teve um bom desempenho na consulta 2.

Gráfico 20 - Diferença de tempo de execução na consulta 2 do cenário 12

4.7.13 Cenário 13

É bastante comum o uso do LIKE, principalmente quando se trata de sistemas com mecanismo de procura, no cenário 13 percebe-se a efetividade de cada ambiente ao realizar uma consulta usando LIKE, passando o coringa (%) no final da palavra, essa forma é sempre mais efetiva, pois facilita a utilização de índice para realizar a busca.

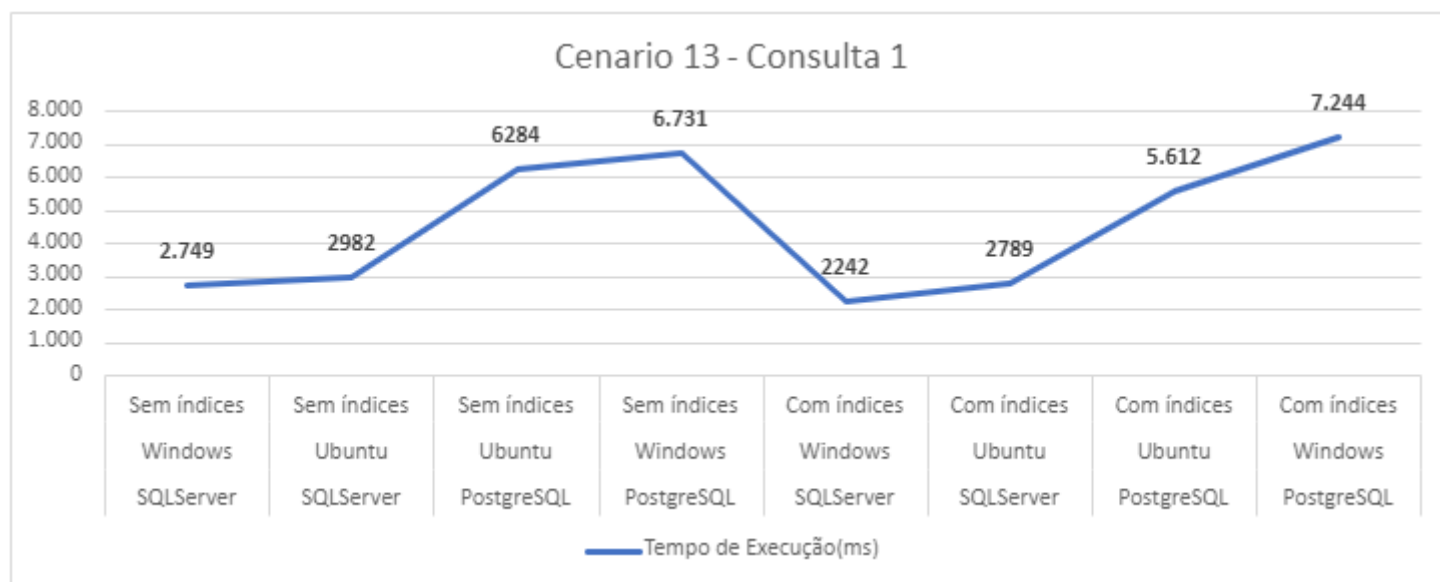
Tabela 52 - Coleta de leituras lógicas e tempo de execução da consulta 1 do cenário 13

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	2.749	260.581
SQLServer	Windows	Com índices	2242	242959
SQLServer	Ubuntu	Sem índices	2982	260.581
SQLServer	Ubuntu	Com índices	2789	242959
PostgreSQL	Ubuntu	Sem índices	6284	597319

PostgreSQL	Ubuntu	Com índices	3.861	716682
PostgreSQL	Windows	Sem índices	6.731	766851
PostgreSQL	Windows	Com índices	7.244	776526

Conforme o próximo gráfico, nota-se que novamente o SQLServer se saiu melhor no tempo de execução da consulta.

Gráfico 21 - Diferença de tempo de execução na consulta 1 do cenário 13



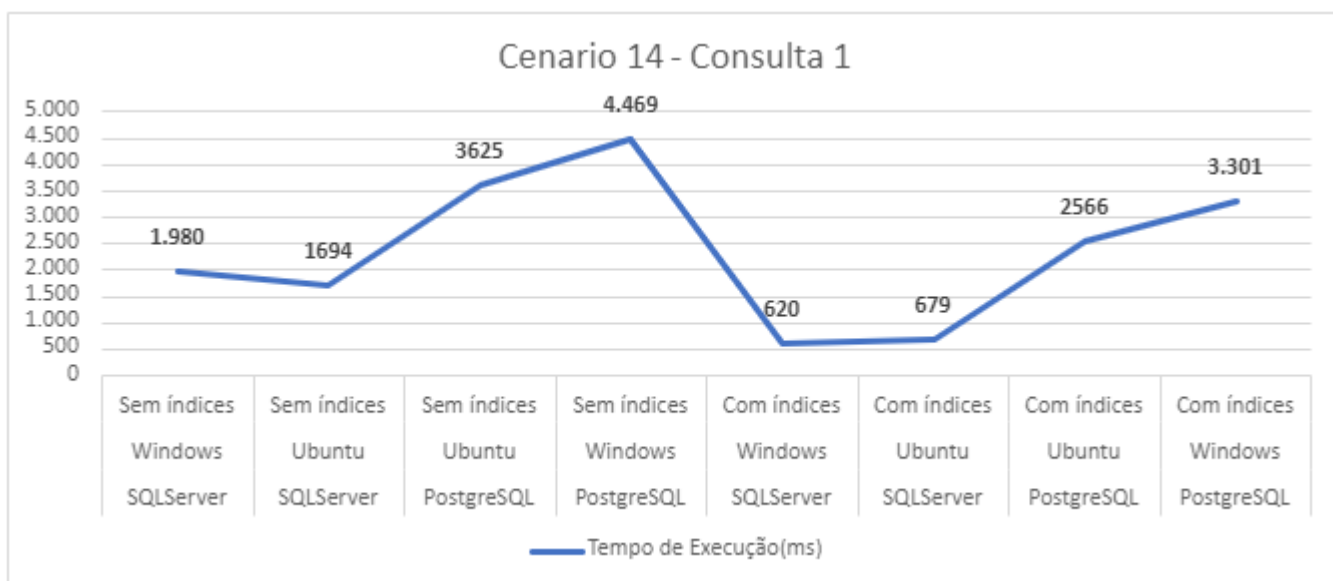
4.7.14 Cenário 14

A situação contrária do cenário anterior, aqui será feito o LIKE, mas com o coringa (%) no início da condição do WHERE, o que dificulta muito para o SGBD, pois a maioria dos índices se tornam inutilizáveis.

Tabela 53 - Coleta de leituras lógicas e tempo de execução da consulta 1 do cenário 14

Tecnologia	S.O	Base	Tempo de Execução(ms)	Leituras Lógicas (Total)
SQLServer	Windows	Sem índices	1.980	173.735
SQLServer	Windows	Com índices	620	209394
SQLServer	Ubuntu	Sem índices	1694	173.735
SQLServer	Ubuntu	Com índices	679	209394
PostgreSQL	Ubuntu	Sem índices	2307	485830
PostgreSQL	Ubuntu	Com índices	1335	1876
PostgreSQL	Windows	Sem índices	4.469	374332
PostgreSQL	Windows	Com índices	3.301	405344

Percebem-se que mesmo assim, tem-se uma vantagem no SQLServer, e foi feito proveito dos índices.

Gráfico 22 - Diferença de tempo de execução na consulta 1 do cenário 14

4.8 CONSIDERAÇÃO SOBRE OS RESULTADOS DOS TESTES

Ao todo foram realizados 14 cenários, que totalizaram 31 consultas testadas, a partir dos dados levantados, será apresentado qual SGBD e Sistema Operacional se saiu melhor em um contexto geral. Para demonstrar o resultado, considera-se que cada consulta equivale a 1 ponto para o ambiente que se saiu melhor, em situações de empate, 1 ponto para cada.

Os resultados ficam conforme as tabelas a seguir:

Tabela 54 - Pontuação por ambiente

Ambiente	Pontos
SQLServer – Ubuntu	14
PostgreSQL – Ubuntu	9
SQLServer – Windows	22
PostgreSQL – Windows	4

Tabela 55 - Pontuação por SGBD

SGBD	Pontos
SQLServer	36
PostgreSQL	13

Tabela 56 - Pontuação por Sistema Operacional

Sistema Operacional	Pontos
Windows	26
Ubuntu	23

Existe uma vantagem aparente do SQLServer, dado em consideração as versões utilizadas do SGBDs, os Sistemas Operacionais e os cenários testados. Ressalta-se também que o ideal é que esse tipo de teste seja realizado sempre que possível antes de realizar a tomada de decisão de que SGBD se deve utilizar para determinada solução de software, como pode-se ver existiu pontos de vantagens de cada SGBD em determinada situação, assim como pode existir diversos cenários que não foram testados nesse trabalho em que um SGBD vai se sobressair do outro.

5 CONSIDERAÇÕES FINAIS

Foram apresentados os SGBDs PostgreSQL e SQLServer, comparando as principais diferenças entre os dois SGBDs no quesito performance de consultas.

Também foi realizada a implementação dos dois SGBDs em ambientes de igual infraestrutura utilizando bases com a estrutura idêntica e a mesma carga de dados e a partir desse ponto foi possível realizar diversos testes de performance, validando diferentes formas de armazenamento de cada tecnologia, funções SQL que cada linguagem possui, criação de índices e coleta de estatísticas.

Ao fim, foi possível ser concluído em quais aspectos cada SGBD se mostrou mais apto e performático.

Alguns aspectos foram validados, como trabalhar com funções de limitação, diferentes sistemas operacionais, trabalhar com grande volume de dados em uma única tabela, desempenho de índices e aproveitamento de leitura em disco.

Os testes foram realizados utilizando maquinas virtuais, com os sistemas operacionais Ubuntu e Windows Server, o software de acesso ao banco que foi utilizado foi o SQL Server Managed Studio para SQL Server e PGAdmin para PostgreSQL.

Como atividades futuras, é possível realizar testes voltados para as operações de escrita e utilização de funções específicas da linguagem de cada SGBD. Também fica em aberto para testes de desempenho, podendo ser os mesmos utilizados neste trabalho, após aplicar técnicas de otimização em ambos os SGBDs e Sistemas Operacionais.

REFERÊNCIAS BIBLIOGRÁFICAS

- ARTBACKUP, **O que é virtualização?** Disponível em: <https://www.artbackup.com.br/data-center/1120-2/> Acesso em 30 nov. 2021
- AZEMOVIC, J. **SQL Server on Linux**. Birmingham, UK: Packt Publishing, 2017. 44p
- BEN-GAN, Itzik **T-SQL Fundamentals, Third Edition**. Washington: Microsoft Press, 2016.
- CAIUT, Fabio. **Administração de Banco de Dados**. Rio de Janeiro: Rede Nacional de Ensino e Pesquisa – RNP, 2015.
- CARVALHO, Vinicius **PostgreSQL : Banco de dados para aplicações web modernas**. Casa do Código, 2017
- GHANAYEM, Mark, **O que é o SSMS (SQL Server Management Studio)?** Microsoft, 2021. Disponível em: <https://docs.microsoft.com/pt-br/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15>. Acesso em: 30 nov. 2021.
- GUARIENTI, Graciyeli Santos Souza. **Fundamentos em Sistemas de Computação** Mato Grosso, 2019. Disponível em: https://educapes.capes.gov.br/bitstream/capes/553325/2/FASCICULO_Fundamentos_Sistema_Computacao.pdf. Acesso em 29 nov. 2021.
- ELMASRI, Ramez; Navathe, Shamkant B. **Sistemas de Banco de Dados 7ª Edição**. São Paulo: Pearson Education do Brasil, 2019.
- MICROSOFT, Docs. **Introdução ao Windows Server**. Disponível em: <https://docs.microsoft.com/pt-br/windows-server/get-started/get-started-with-windows-server>. Acesso em: 29 nov. 2021.

NEMETH, Evi; Snyder, Garth; Hein, Trent. **Manual Completo de Linux: guia do administrador – 2ª edição**. São Paulo: Pearson Prentice Hall, 2007. 24 p.

PGADMIN **pgAdmin 4**. Disponível em:

<https://www.pgadmin.org/docs/pgadmin4/6.0/index.html>. Acesso em: 30 nov 2021.

PRODANOV, Cleber Crisitano; Freitas, Ernani Cesar de. **Metodologia do Trabalho Científico: Métodos e Técnicas de Pesquisa e do Trabalho Acadêmico 2ª Edição**. Rio Grande do Sul: Editora Feevale, 2013.

SILBERSCHATZ A; GALVIN P; GAGNE G. **Sistemas Operacionais com JAVA**. Editora Campus, 2004

TANENBAUM, Andrew S. **Sistemas Operacionais Modernos 4ª Edição**. São Paulo: Perason Education do Brasil, 2015.

VERGARA, Sylvia C. **Projetos e relatórios de pesquisa em administração**. 3. ed. São Paulo: Atlas, 2000.

RICARDO. **A História dos Banco de Dados**. 2006. Disponível em:

<https://www.devmedia.com.br/a-historia-dos-banco-de-dados/1678> Acesso em: 4 jun 2022

MICROSOFT, Docs. **Obter os bancos de dados de exemplo ADO.NET exemplos de código**. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/framework/data/adonet/sql/linq/downloading-sample-databases>

Acesso em: 26 mar. 2022.

pthom. **Northwind database for Postgres**. Disponível em:

https://github.com/pthom/northwind_psql Acesso em: 27 mar. 2022.

AMORIM, Fabiano. **25 Dicas de Performance no SQL Server**. Disponível em:

<https://cursos.powertuning.com.br/course?courseid=25-dicas-de-performance-no-sql-server-parte-1> Acesso em: 12 fev. 2022.

MICROSOFT, Docs. **uniqueidentifier (Transact-SQL)** Disponível em:

<https://docs.microsoft.com/pt-br/sql/t-sql/data-types/uniqueidentifier-transact-sql?view=sql-server-ver16> Acesso em: 11 jun. 2022.

MICROSOFT, Docs. **Início Rápido: Instalar o SQL Server e criar um banco de dados no**

Ubuntu. Disponível em: <https://docs.microsoft.com/pt-br/sql/linux/quickstart-install-connect-ubuntu?view=sql-server-ver16> Acesso em: 11 jun. 2022.