Comparação do desempenho de bancos de dados SQL e NoSQL: Azure SQL Database e Azure Cosmos DB

Tainá Oliveira Soares[1], **Wagner Inácio de Oliveira**[1], **Marcelle Model Alves**[1]

[1]Universidade Federal do Pampa – (UNIPAMPA)
97.546-550 – Alegrete – RS – Brazil

Abstract. *The aim of this paper is to compare and analyze the performance of relational (SQL) and non-relational (NoSQL) databases. We present the development and results of the proposed comparative analysis between the Azure SQL Database and the Azure Cosmos DB through graphical representations and tables considering the behavior of each database for INSERT, UPDATE and DELETE operations.*

Resumo. *Este trabalho tem como objetivo comparar e analisar o desempenho de bancos de dados relacionais (SQL) e não relacionais (NoSQL). Apresentamos o desenvolvimento e resultados da análise comparativa proposta entre o Azure SQL Database e o Azure Cosmos DB através de representações gráficas e tabelas considerando o comportamento de cada banco para operações INSERT, UPDATE e DELETE.*

1. Introdução

Entende-se por banco de dados como o local onde dados são armazenados para que possam ser acessados a qualquer momento quando for necessário. Com base na maneira como esses dados são relacionados, conseguimos classificar bancos de dados em duas categorias: bancos de dados relacionais (*SQL*) e bancos de dados não relacionais (*NoSQL*) [Khasawneh et al. 2020].

Neste trabalho, apresentamos uma análise comparativa do desempenho de banco de dados *SQL* e *NoSQL* com o objetivo de testar o comportamento de dois bancos diferentes, o **Azure SQL Database** e o **Azure Cosmos DB** em relação ao modelo de dados de 4 tabelas proposto e implementado. A análise do desempenho de cada banco foi realizada utilizando as seguintes operações:

1. *INSERT*
2. *UPDATE*
3. *DELETE*

O restante deste trabalho está organizado da seguinte maneira: na seção 2 exploramos os conceitos de bancos de dados relacionais (*SQL*) e bancos de dados não relacionais (*NoSQL*) assim como uma breve descrição dos *frameworks* e outros componentes utilizados no processo de desenvolvimento, na seção 3 apresentamos o *setup* onde são realizados os testes, nas seções 4 e 5 apresentamos representações gráficas do modelo de dados implementado e tabelas, nas seções 6 e 7 demonstramos os resultados obtidos e discutimos sobre as lições aprendidas durante a realização do trabalho e por fim, na seção 8 estão as considerações finais deste projeto.

2. Fundamentação Teórica

Nesta seção são abordados os conceitos de bancos de dados *SQL* e *NoSQL* e apresentadas informações sobre as tecnologias utilizadas para a realização deste trabalho.

2.1. *Banco de Dados SQL*

Os bancos de dados *SQL* são bancos de dados relacionais, ou seja, possuem tabelas com estruturas bem definidas que apresentam forte relação entre os dados [Khasawneh et al. 2020]. As informações são armazenadas em grupos de tabelas ou "relações" que são isoladas em linhas (registros) e segmentos (campos) [MUS 2019] que são analizados através de consultas *SQL*.

Para realização da análise comparativa proposta neste trabalho, o bando de dados relaional utilizado foi o **Azure SQL Database**, que está no mercado desde 2010 e inclui inteligência integrada e se adapta para maximizar o desemprenho, confiabilidade e proteção dos dados.

2.2. *Banco de Dados NoSQL*

Os bancos de dados *NoSQL*, que é a abreviação de *not only SQL*, utilizam de um mecanismo diferente para o armazenamento e recuperação de dados. São conhecidos como bancos de dados não relacionais ou distribuídos pois não exigem que os dados tenham forte relações entre si, o que é o caso dos bancos de dados relacionais.

Bancos de dados *NoSQL* operam em uma variedade de tipos com base em seu modelo de dados [MUS 2019]. As principais categorias são: *key-value store* ou armazenamento de chave-valor, orientado a coluna, orientado a documentos e orientado a grafos [Khasawneh et al. 2020]. Estas categorias fornecem esquemas flexíveis e operam facilmente com grandes volumes de dados e números de usuários.

Neste trabalho, o banco de dados *NoSQL* escolhido foi o **Azure Cosmos DB**, que está no mercado desde 2017. O **Azure Cosmos DB** é um banco de dados orientado a documentos.

2.2.1. Banco de Dados NoSQL Orientado a Documentos

Banco de dados orientados a documentos é um dos princpais tipos de bancos de dados não relacionais pois são bastante semelhantes aos bancos de dados relacionais tradicionais. Os dados são adicionados em formatos JSON, BSON ou documentos XML e podem ser indexados para possibiliar maior velocidade nas consultas [Mason 2015].

Os documentos podem ser armazenados e recuperados de forma muito mais fiel aos dados utilizados em aplicações e também há a flexibilidade de reestruturação de documentos confirme a necessidade.

2.3. Ferramentas Utilizadas

Além dos banco de dados *SQL* e *NoSQL*, foram utilizadas outras ferramentas para o desenvolvimento deste trabalho.

Os bancos de dados utilizados são disponibilizados como serviço da plataforma Microsoft Azure [1], destinada à execução de aplicativos e serviços, baseada nos conceitos da computação em nuvem.

O *Framework* **.NET** [2], que é uma plataforma para desenvolvimento de aplicações criada e mantida pela *Microsoft*, foi utilizado para implementar os softwares(soluções) utilizadas como interface para realizar operações nos bancos de dados.

A solução para o banco de dados relacional foi uma aplicação *web* (WebApp), construída com a estrutra *ASP.NET Core* do framework *.NET*.

As operações realizadas no banco de dados não relacional, foram feitas por meio de uma aplicação de console, utilizando a linguagem C.

O *Entity Framework Core* [3] (EF) foi empregado no mapeamento das entitdades (classes) para ambos os banco de dados, pois disponibiliza um mecanismo automatizado para acessar e armazenar dados em bancos relacionais e não relacionais. Ademais, o EF, foi utilizado para mapear as entidades definidas, tendo em vista que o *framework* é uma estrutura de Mapeamento de Objeto-Relacional (ORM).

Para garantir a diversidade e qualidade dos dados persitidos durante as operações para os diferentes cenários de testes, utilizou-se a biblioteca *Bogus* [4], gerador de dados fictícios para aplicações desenvolvidas em linguagens .NET como C# , F# e VB.NET.

3. Ambiente de Teste

A execução das operações foram realizadas em uma máquina física, com as seguintes configurações:

- Sistema operacional: Windows 10 Pro;
- Processador: *AMD FX(tm)-8300 Eight Core Processor* 3.30 GHz
- Placa de vídeo: *AMD Radeon R7 200 Series - 2 gigabyte*;
- Memória RAM: 8 *gigabyte*

Os bancos de dados SQL, utilizados para os testes estão armazenas em servidores disponibilizados como serviço pela plataforma Azure. Para garantir que a latência fosse semelhante entre as operações para os diferentes bancos comparados, criou-se um servidor SQL para armazenamento do banco de dados, localizado no Oeste dos Estados Unidos(Azure SQL Database - Oeste/EUA). Pois o banco de dados NoSQL não possui servidores no Brasil, sendo assim, para que a comparação fosse justa, definimos as residência dos dados tanto para leitura e escrita na mesma região.

Também foi utilizado um banco de dados SQL, hospedado em um servidor no sul do Brasil(Azure SQL Database - Sul/BR), para compararmos a latência com o banco de dados relacional hospedado no Oeste dos Estados Unidos.

Os dois bancos SQL, configurados possuem as mesmas caracteríticas, sendo 2 *Gigabyte* de memória de armazenamento máximo e 5 DTU's. Uma DTU (unidade de

[1] https://azure.microsoft.com
[2] https://dotnet.microsoft.com
[3] https://docs.microsoft.com/pt-br/ef/core/
[4] https://github.com/bchavez/Bogus

transação do banco de dados) representa uma medida combinada de CPU, memória, leituras e gravações.

O banco de dados NoSQL da Azure, não oferece servidores dedicados para o armazenamento das bases, pois se trata de um banco distribuido globalmente. Portanto, para que não houvesse custos monetários durante as execuções, configurou-se o banco de dados dentro dos paramêtros gratuitos da plataforma Azure. o Nível gratuito da plataforma para bancos NoSQL, disponibiliza 25 *Gigabyte* de armazemamento para documentos e com taxa de transferência provisionada a 1000 RU's (Unidade de Solicitação). Ademais, para a adesão do nível gratuito, foi necessário definir o local de leuitura e gravação dos dados, a região escolhida foi o Oeste dos Estados Unidos (Azure Cosmos DB - Oeste/EUA). Vale ressaltar que, os bancos de dados NoSQL da Azure, são armazendos em container, e que as taxas de transferencias são compartilhadas entre todos os containers, portanto, matemos apenas uma instância do banco de dados.

Além disso, todos os bancos utilizados para axecução dos testes, foram configurados no modo Test/Development no ato da criação das bases, ou seja, nenhum teste de banco operou em status de produção.

4. Modelagem de Dados

Para a realização desse trabalho, foi modelado quatro entitdades, sendo elas: Fornecedor, Endereço, Produto e Venda. Para cada entidade definiu-se pelo menos um atributo, como apresentado na figura 1.

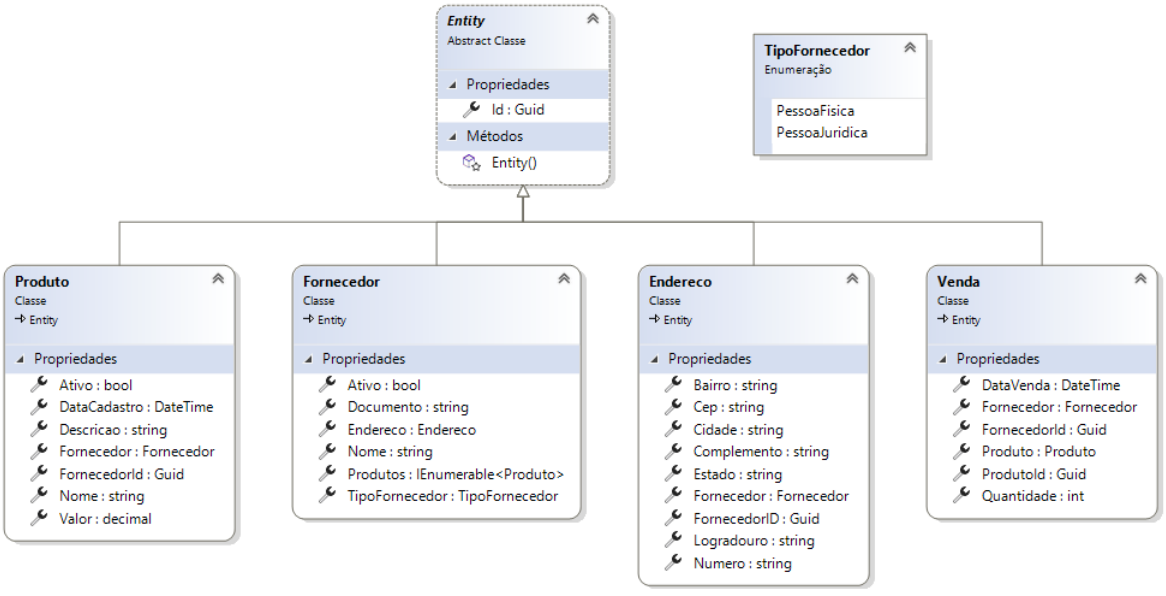

Figura 1. Diagrama de Classes da aplicação

Para o mapeamento das entidades para o banco de dados, utilizou-se e *Entity Framework Core*, o EF Core é um mapeador relacional de objeto (ORM). O Mapeamento Objeto-Relacional é uma técnica que permite aos desenvolvedores trabalhar com dados de maneira orientada a objetos, realizando o trabalho necessário para mapear entre objetos definidos na linguagem de programação de um aplicativo e dados armazenados em

fontes de dados relacionais. A modalegem para o banco de dados é aprensatado na Figura 2.

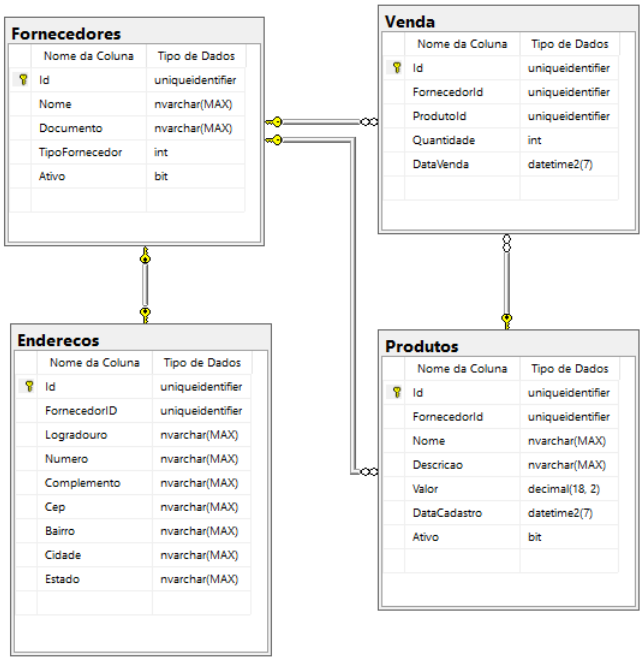

Figura 2. Diagrama ORM do banco de dados

5. Execução dos Testes

Foi realizada a execução das operações de *Insert, Update e Delete*, para diferentes volumes de dados: 100, 1000, 10000,100000 e 1000000. Para cada cenário considerando operação e volume, foram feitas três execuções, exceto para as operações com volumes de dados de 1000000, que foram realizadas uma única vez.

Nas operações de *Insert*, com o utilização da biblioteca *Bogus*, foram gerados dados fictícios para todas as tabelas, garantindo que todo os campos fossem preenchidos.

Nas operações de *Update*, exceto os campos de identificador único, todos os outros foram atualizados com novos registros. Para os operações de *Delete*, todos os registros foram excluídos das tabelas de forma sequuêncial, sem empregar nenhuma cláusula para defnição de critérios.

O tempo decorrido para cada operação nos diferentes cenários de volumes de dados, foram registrados em segundos, à partir dos arquivos de*logs* gerados ao final de cada operação executada.

A execução das operações para os bancos relacioanais, foram feitos via interface de uma aplicação web que operou em um ambiente de hospedagem local. A aplicação web, foi executada em modo *release* (fora do ambiente de depuração) em uma versão oficial do navegador Google Chrome de 64 bits. Destaca-se que para cada conjunto de operações executadas, todos os sistemas de hardware e software foram reinicializados.

6. Análise dos Resultados

A análise dos resultados foi realizada em duas etapas. Primeiramente, foram analisados os resultados por Banco de Dado, em 6.1, para comparar as operações de *Insert*, *Update* e *Delete* em cada um deles. Depois disso, em 6.2, a análise se deu de forma comparativa entre os Bancos de Dados.

6.1. Operações

Nas Tabelas 1, 2 e 3 temos os resultados de tempo das operações quando usados os Bancos de Dados Azure SQL Database - Oeste/EUA, Azure SQL Database - Sul/BR e Azure Cosmos DB - Oeste/EUA.

Tabela 1. Tempo (s) para Azure SQL Database (Oeste/EUA)

INSERT					
Tabela	**100**	**1K**	**10K**	**100K**	**1M**
Fornecedor-Endereço	85,44	883,61	5808,52	90671,92	807685,65
Produto	62,32	604,11	6976,93	70821,33	547170,6
Venda	63,7	712,55	7112,41	68028,25	713154,86
UPDATE					
Tabela	**100**	**1K**	**10K**	**100K**	**1M**
Fornecedor	61,55	604,79	7022,39	66516,32	756822,86
Endereço	60,52	673,04	6742,42	63774,26	664280,86
Produto	61,71	632,15	6825,85	66496,46	489144,13
Venda	63,86	665,07	6578,79	60517,89	751334,16
DELETE					
Tabela	**100**	**1K**	**10K**	**100K**	**1M**
Tabela	**100**	**1K**	**10K**	**100K**	**1M**
Fornecedor	62,2	545,56	7043,5	67575,03	595257,03
Endereço	61,43	573,55	5908,78	83530,46	624513,97
Produto	62,17	616,42	6815,8	61558,97	639784,27
Venda	261,22	774,99	5868,27	64809,65	533547,8

A operação *Delete* foi a mais rápida na maioria dos casos para os três bancos de dados, pois a implementação do método que executa essa operação não define nenhuma condição de verificação ou cláusula para especifiação de critérios.

Enquanto isso, a operação de *Insert* foi a mais demorada. Essa operação é tipicamente mais demorada do que as outras porque é nela em que o SGBD (Sistema de Gerenciamento de Dados) reserva recursos para a criação de registros. Vale ressaltar que no caso do *Insert*, os registros das entidades Fornecedor e Endereço são persistidos em

Tabela 2. Tempo (s) para Azure SQL Database (Sul/BR)

INSERT					
Tabela	**100**	**1K**	**10K**	**100K**	**1M**
Fornecedor-Endereço	14,18	144,14	1470,88	12681,13	102091,77
Produto	9,84	94,23	1085,49	9959,95	112002,33
Venda	9,64	83,14	1019,23	9065,25	85937,23
UPDATE					
Tabela	**100**	**1K**	**10K**	**100K**	**1M**
Fornecedor	9,81	105,93	644,13	11036,49	107722,6
Endereço	9,79	89,01	1094,85	9461,55	84449,78
Produto	10,14	110,57	989,37	9132,11	109779,11
Venda	9,53	84,424	1083,49	9265,15	106478,68
DELETE					
Tabela	**100**	**1K**	**10K**	**100K**	**1M**
Fornecedor	9,17	101,41	1025,94	7546,66	92796,33
Endereço	9,15	85,33	999,92	11480,74	80621,74
Produto	9,24	84.16	951,07	9197,15	124328,88
Venda	9,52	98,38	922,05	8464,44	103182,85

Tabela 3. Tempo (s) para Azure Cosmos DB (NoSQL - Oeste/EUA)

INSERT					
Tabela	**100**	**1K**	**10K**	**100K**	**1M**
Fornecedor-Endereço	44,41	500,58	4076,99	41948	408892,23
Produto	22,28	171,28	2306,04	14334,915	229929,07
Venda	21,74	242,62	1923,77	21421,49	239145,65
UPDATE					
Tabela	**100**	**1K**	**10K**	**100K**	**1M**
Fornecedor	22,65	200,69	2283,13	18132,15	202614,53
Endereço	23,45	254,99	2838,27	23670,85	274258,12
Produto	23,54	236,89	3044,49	21560,83	257820,1
Venda	21,17	190,48	2627,36	19191,38	208476,55
DELETE					
Tabela	**100**	**1K**	**10K**	**100K**	**1M**
Fornecedor	21,46	228,46	2385,27	23074,21	212909,32
Endereço	22,41	199,81	2262,26	24232,49	221063,35
Produto	21,72	245,85	1894,62	23162,89	201431
Venda	20,85	214,88	2645,12	22557,09	131922,47

conjunto, pois assim que um novo fornecedor faz parte do banco de dados, o endereço dele também faz. Essa característica fez com que o tempo para essas entidades fosse significativamente mais demorado do que nos outros casos.

6.2. Bancos de Dados

Os gráficos das figuras 3, 4 e 5 apresentam os resultados de tempo das operações para os bancos localizados nos EUA, para facilitar a comparação entre SQL e NoSQL. Já os gráficos das figuras 6, 7 e 8 mostram os tempos das operações para os bancos SQL localizados nos EUA e no Brasil, para destacar a diferença oriunda da localização.

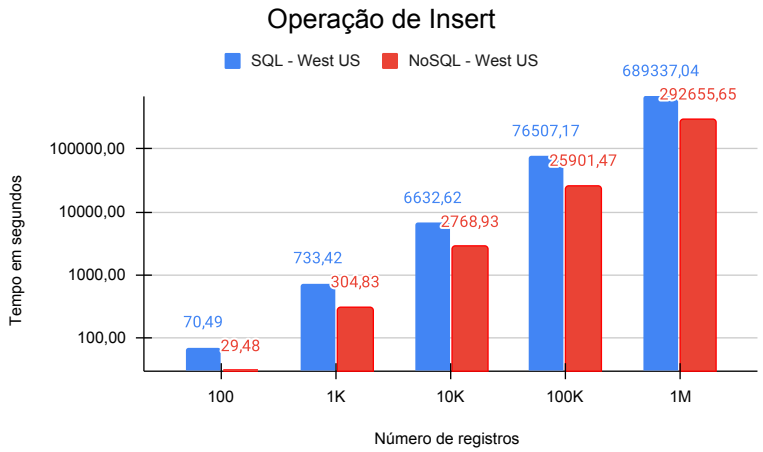

Figura 3. Insert - EUA

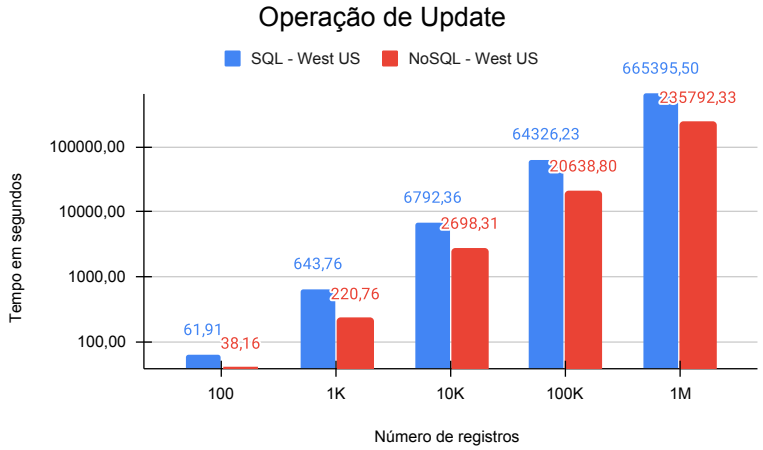

Figura 4. Update - EUA

Ao comparar o banco SQL e o banco NoSQL, ambos localizados nos EUA, o SQL foi mais demorado do que o NoSQL em todos os casos neste estudo. Os bancos SQL, em sua maioria (incluindo o Azure SQL Database), garantem ACID (Atomicidade, Consistência, Isolamento e Durabilidade) e por isso tendem a demorar mais do que os bancos NoSQL. Essas garantias exigem uma série de processos internos do SGBD, que levam tempo.

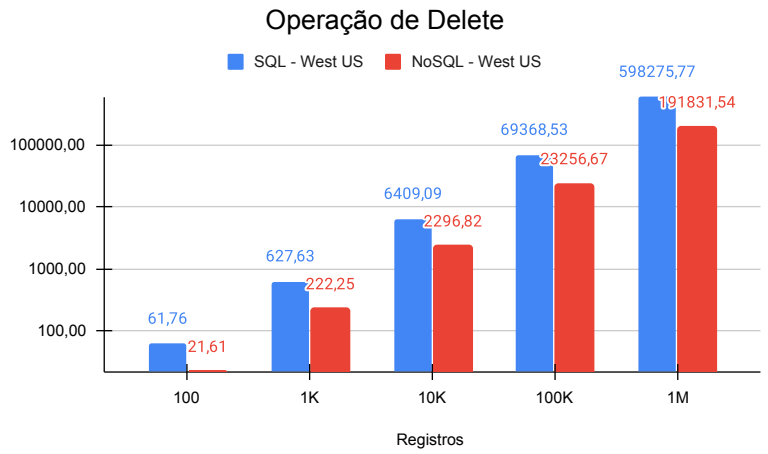

Figura 5. Delete - EUA

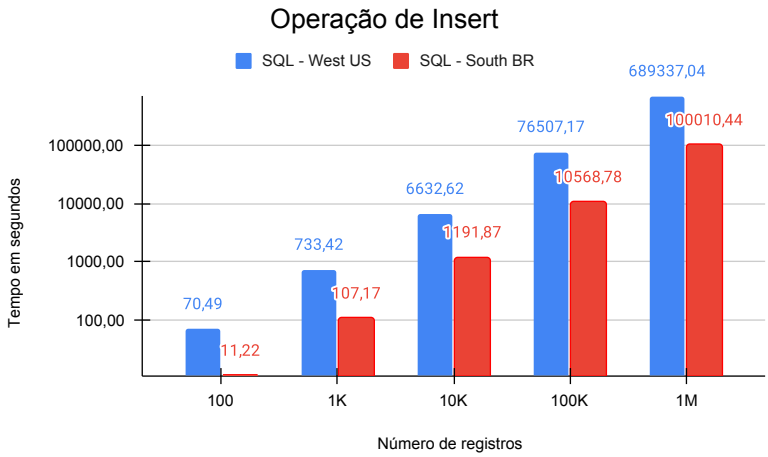

Figura 6. Insert - SQL

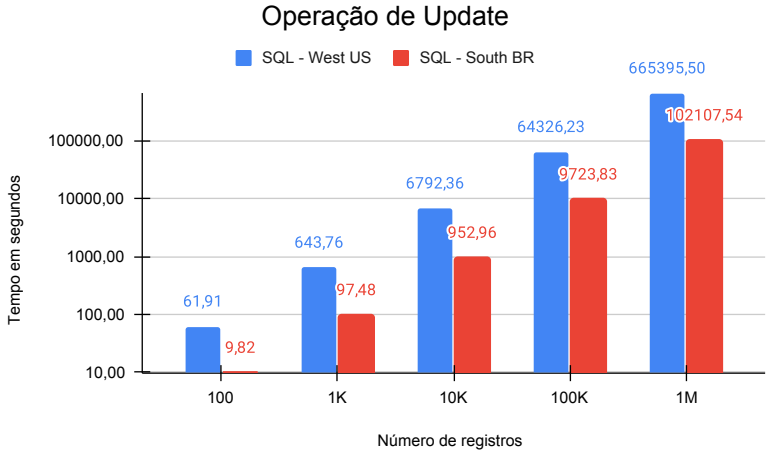

Figura 7. Update - SQL

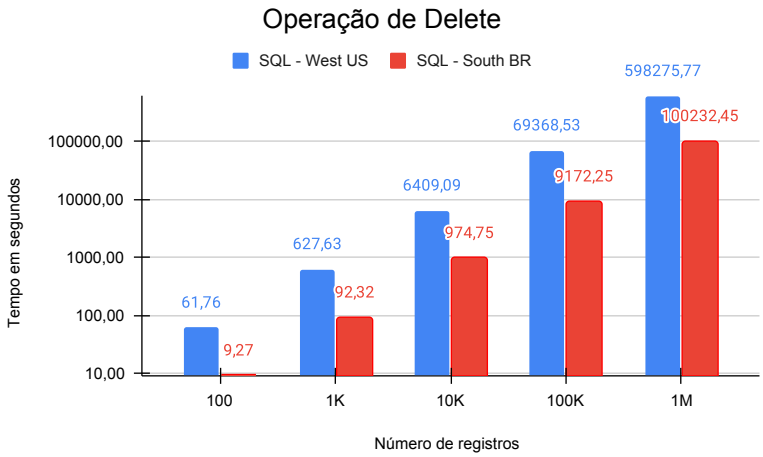

Figura 8. Delete - SQL

Quanto a localização dos bancos de dados, o tempo das operações foi diretamente proporcional a distância entre o servidor do banco de dados e o local dos testes (Sul do Brasil). Assim, o tempo levado para o banco localizado no Brasil é menor do que para o banco localizado nos EUA. Esse resultado já era esperado devido a latência da conexão. No caso do nosso estudo, a latência foi tão alta, que até o banco NoSQL (que de acorodo com os resultados foi consideravelmente mais rápido do que o SQL quando comparados sem considerar diferenças de distância) localizado nos EUA, demorou mais do que o banco SQL localizado no Brasil. Além disso, a velocidade da rede de *internet* pode impactar na latência das operações.

7. Considerações Finais

As operações básicas de *Insert*, *Update* e *Delete* apresentaram comportamento esperado já que *Insert*, que se preocupa com mais recursos, foi a mais demorada e *Delete*, que teve o método que usa menos funções, foi a mais rápida.

Através dos resultados obtidos, ficaram claros alguns aspectos que impactam no tempo das operações de banco de dados, como a latência de conexão. A latência é oriunda principalmente de aspectos como velocidade de rede e da distância entre o servidor do banco de dados e o local onde as operações são definidas.

Além disso, os resultados também mostraram que o banco Azure SQL Database tende a ser mais demorado do que o banco Azure Cosmos DB (NoSQL). No entanto, dependendo da localização do servidor do banco, vale mais a pena, em questão de tempo levado para as operações, usar o banco SQL se ele é localizado mais próximo ao local onde a aplicação é executada. Ao usar o SQL, a confiabilidade aumenta também.

Por fim, vale ressaltar que nenhum banco de dados é melhor ou pior do que o outro. Cada um tem suas propriedades e a melhor escolha sempre depende da aplicação [Jatana et al. 2012]. Os resultados e conclusões apresentados nesse trabalho são exclusivamente particulares à nossa aplicação e experimento.

Referências

[Jatana et al. 2012] Jatana, N., Puri, S., Ahuja, M., Kathuria, I., and Gosain, D. (2012). A survey and comparison of relational and non-relational database.

[Khasawneh et al. 2020] Khasawneh, T. N., AL-Sahlee, M. H., and Safia, A. A. (2020). Sql, newsql, and nosql databases: A comparative survey. In *2020 11th International Conference on Information and Communication Systems (ICICS)*, pages 013–021. IEEE.

[Mason 2015] Mason, R. T. (2015). Nosql databases and data modeling techniques for a document-oriented nosql database. In *Proceedings of Informing Science & IT Education Conference (InSITE)*, volume 3, pages 259–268.

[MUS 2019] MUS, M. (2019). Comparison between sql and nosql databases and their relationship with big data analytics.