

Comparando os banco de dados PostgreSQL e MongoDB

Henrique Fan, Jian Furquim

¹ Universidade Federal do Pampa (UNIPAMPA)
Alegrete – RS – Brasil

henriquefan.aluno@unipampa.edu.br, jianfurquim.aluno@unipampa.edu.br

1. Introdução

Os bancos de dados e os sistemas de bancos de dados se tornaram componentes essenciais no cotidiano da sociedade moderna. No decorrer do dia, a maioria de nós se depara com atividades que envolvem alguma interação com os bancos de dados. Por exemplo, se formos ao banco para efetuarmos um depósito ou retirar dinheiro, se fizermos reservas em um hotel ou para a compra de passagens aéreas[Elmasri et al. 2005]. O volume dados cresce diariamente, com todas essa quantidade de serviço que utilizam bancos de dados. O aumento no tráfego de dados exigiu que empresas procurem soluções para poder suprir a demanda cada vez maior de performance e disponibilidade. Fazendo com que outros requisitos até então indiscutíveis, como consistência de dados, precisem ser revistos [Vogels 2009].

No início dos anos 70 a IBM lançou, a Structured Query Language (SQL), com isso, surgiu os bancos SQLs. Com forte inspiração na álgebra relacional e devido a sua facilidade de aprendizagem e simplicidade tornaram-se a linguagem mais utilizada em bancos de dados. Seu funcionamento é extremamente simples, armazenam os dados em tabelas e utiliza o modelo relacional para criar relações entre as tabelas de dados. Enquanto isso, os bancos de dados Not Only SQL (NoSQL) surgiram no final dos anos 90, apresentam outras características não convencionais, trazendo um paradigma diferente de armazenamento de dados, como por exemplo, armazenamento de documentos sem esquema definido. Tendo como justificativa de uso pelas limitações apresentadas pelo modelo relacional (SQL), quando o volume de dados é elevado e as cargas de trabalhos são grandes [Peres 2018].

2. PostgreSQL

O PostgreSQL é sistema de gerenciamento de banco de dados objeto-relacional de código aberto que usa e estende a linguagem SQL combinada com muitos recursos que armazenam e escalam com segurança as cargas de trabalho de dados mais complexas. As origens do PostgreSQL remontam a 1986 como parte do projeto POSTGRES na Universidade da Califórnia em Berkeley e tem mais de 30 anos de desenvolvimento ativo na plataforma central, ganhando uma forte reputação por sua arquitetura comprovada, confiabilidade, integridade de dados, conjunto de recursos robustos, extensibilidade e dedicação da comunidade de código aberto por trás do software para fornecer soluções inovadoras e de alto desempenho de maneira consistente. PostgreSQL é executado em todos os principais sistemas operacionais[PostgreSQL 2021].

3. MongoDB

O MongoDB inicialmente foi desenvolvido como um componente de serviço pela empresa 10gen em outubro de 2007, tornando-se um software *open source* apenas em 2009,

ano em que foi lançada sua primeira versão. O MongoDB é uma aplicação de código aberto, de alta performance, sem esquemas, orientado a documentos. Baseando os documentos em um formato chamado BSON (Binary JSON). Possui facilidade de migração para um banco de dados relacional, pelo fato de suas consultas serem convertidas facilmente [MongoDB 2021].

4. Diferenças entre PostgreSQL e o MongoDB

A Tabela 1 demonstra as principais diferenças entre PostgreSQL e o MongoDB.

Table 1. Comparação entre MongoDB e PostgreSQL.

Base de comparação	PostgreSQL	MongoDB
Definição	Bando de dados relacional	Bando de dados não relacional
Programação	C	C, C++ e JavaScript
Desenvolvedor	Grupo de desenvolvimento global PostgreSQL	MongoDB Inc.
Orientação	Orientado a Objeto	Orientado a Documentos
Idiomas	Disponível em vários idiomas	Disponível apenas em Inglês
Plataformas	Plataformas do tipo Unix e Windows	Suporta muitas plataformas
Sintaxe	Parecida com outros bancos de dados relacionais	Diferente dos bancos de dados relacionais

Adaptado de [Pedamkar 2021].

Muitos dos termos e conceitos usados no modelo de documento do MongoDB são iguais ou semelhantes ao modelo tabular do PostgreSQL, conforme apresentado na Tabela 1 [MongoDB 2021].

Table 2. PostgreSQL vs MongoDB Terminologia e Conceitos.

PostgreSQL	MongoDB
Transações ACID	Transações ACID
Tabela	Coleção
Linha	Documento
Coluna	Campo
Índice secundário	Índice secundário
JOINS, UNIONs	Documentos incorporados, \$lookup e \$graphLookup, \$unionWith
Visualizações Materializadas	Visualizações Materializadas Sob Demanda
GROUP_BY	Agregação Pipele

Adaptado de [MongoDB 2021].

5. Objetivo

O objetivo deste trabalho é apresentar uma análise comparativa de desempenho entre o banco de dados SQL PostgreSQL e o banco de dados NoSQL MongoDB, para isso, serão aplicados testes de desempenho para cada um deles por meio da realização das operações básicas de INSERT, UPDATE e DELETE, medindo os seguintes tamanhos de volumes de dados: 100, 1.000, 10.000, 100.000 e 1.000.000 de registros para cada uma das operações.

Em nossos testes foram criadas seguintes tabelas nos dois bancos (os campos marcados com * são chaves primárias e os sublinhados chaves estrangeiras):

- Cliente (IdCliente*, Nome, Sexo, Endereco, Cidade, UF)

- ClienteDetalhe (IdCliente*, DataNasc, LimiteAprovado, ClienteDesde, ValorAcumulado)
- ClienteProduto (idClienteProduto*, IdCliente, IdProduto, Qtde, Valor)
- Produto (IdProduto*, Nome, Sigla, Tipo, Descricao, Valor)

6. Metodologia

Para implementar os testes, a linguagem Python versao 3.8 foi utilizada juntamente com as respectivas interfaces para a conexão com os bancos. Para o PostgreSQL, foi utilizada a interface Psycopg2 e para o MongoDB foi utilizada a Pymongo.

Para replicar os testes basta usar o comando (-c 5, é a quantidade de vezes que irá executar, depois uma média dos tempo é tirada):

```
python3 main.py -c 5
```

A Tabela 3 apresenta o ambiente de nossos teste.

Table 3. Parâmetros do ambiente de teste.

Descrição	Valor(es)
Sistema Operacional	Ubuntu 20.04.3 LTS (Focal Fossa)
Kernel	Linux 5.11.0-36-generic
Processador	Intel Core i5-5200U - 2.20GHz
Memória	8GB
HD	1.000GB
Versão PostgreSQL	9.3.25
Versão MongoDB	3.6.8

Figure 1. Interface de conexão com o PostgreSQL.

```
host = "localhost"
dbname = "trabalho_final"
user = "postgres"
password = "postgres"
sslmode = "require"

conn_string = "host={0} user={1} dbname={2} password={3} sslmode={4}".format(host, user, dbname, password, sslmode)
conn = psycopg2.connect(conn_string)
db = conn.cursor()
```

Comandos que criam as tabelas no PostgreSQL:

- CREATE TABLE client(idclient integer not null, name varchar(10), sex varchar(1), address varchar(10), city varchar(10), uf varchar(2), primary key(idclient))
- CREATE TABLE clientDetails(idclient integer not null, dateBirth DATE, approvedLimit REAL, sinceClient DATE, accumulatedValue REAL, primary key(idclient), constraint fk_client foreign key (idclient) references client(idclient))

Figure 2. Interface de conexão com o MongoDB.

```
host = "localhost"
dbname = "trabalho_final"
port = '27017'

conn_string = "mongodb://localhost:27017/"
conn = pymongo.MongoClient(conn_string)

db = conn['trabalho_final']
```

- CREATE TABLE product(idproduct integer not null, name varchar (10), initials varchar(5), type varchar(10), description varchar(10), value REAL, primary key(idproduct))
- CREATE TABLE clientProduct(idclientProduct integer not null, idclient integer not null, idproduct integer not null, amount integer not null, value REAL, primary key(idclientProduct), constraint fk_client foreign key (idclient) references client(idclient), constraint fk_product foreign key (idproduct) references product(idproduct))

A Figura 3 demonstra a criação das coleções em MongoDB.

Figure 3. Comandos que criam as Coleções em MongoDB.

```
client = db['client']
clientDetails = db['clientDetails']
product = db['product']
clientProduct = db['clientProduct']
```

7. Resultados

Os gráficos resultantes das operações INSERT e UPDATE realizados no PostgreSQL com os tipos de dados STRING e NUMBER são apresentados nas Figuras 4, 5, 6 e 7 respectivamente e na Tabela 4.

Os gráficos resultantes das operações INSERT e UPDATE realizados no MongoDB com os tipos de dados STRING e NUMBER são apresentados nas Figuras 8, 9, 10 e 11 respectivamente e na Tabela 5.

Table 4. Resultados PostgreSQL.

Banco de Dados PostgreSQL				
Número de registros	Tempo em Segundos			
	STRINGS		NUMBERS	
	INSERT	UPDATE	INSERT	UPDATE
100	0,15	0,17	0,16	0,17
1.000	1,92	2,03	2,01	2,16
10.000	19,46	19,93	20,32	20,54
100.000	253,47	452,06	437,05	452,64
1.000.000	4199,14	4787,8	4668,1	4824,63

References

- Elmasri, R., Navathe, S. B., Pinheiro, M. G., et al. (2005). Sistemas de banco de dados.
- MongoDB (2021). MongoDB. <https://www.mongodb.com/pt-br>. (Acesso em 27/09/2021).
- Pedamkar, P. (2021). MongoDB vs postgresql. <https://www.educba.com/mongodb-vs-postgresql/>. (Acesso em 27/09/2021).
- Peres, R. (2018). Sql vs nosql. *Revista Programador*.
- PostgreSQL (2021). Postgresql. <https://www.postgresql.org/>. (Acesso em 27/09/2021).
- Vogels, W. (2009). Eventually consistent. *Communications of the ACM*, 52(1):40–44.

Table 5. Resultados MongoDB.

Banco de Dados MongoDB				
Número de registros	Tempo em Segundos			
	STRINGS		NUMBERS	
	INSERT	UPDATE	INSERT	UPDATE
100	0,02	0,03	0,02	0,03
1.000	0,25	0,34	0,24	0,33
10.000	2,43	3,23	2,32	3,27
100.000	28,17	33,57	27,03	34,64
1.000.000	283,35	380,84	312,54	377,63

Figure 4. INSERT STRINGS PostgreSQL.

INSERT STRINGS - PostgreSQL

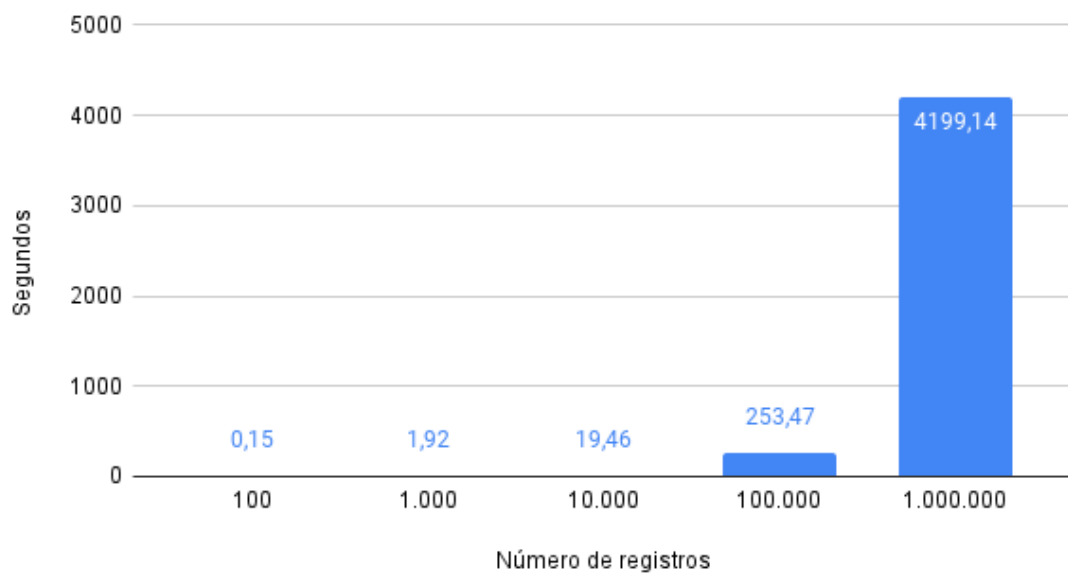


Figure 5. UPDATE STRINGS PostgreSQL.

UPDATE STRINGS - PostgreSQL

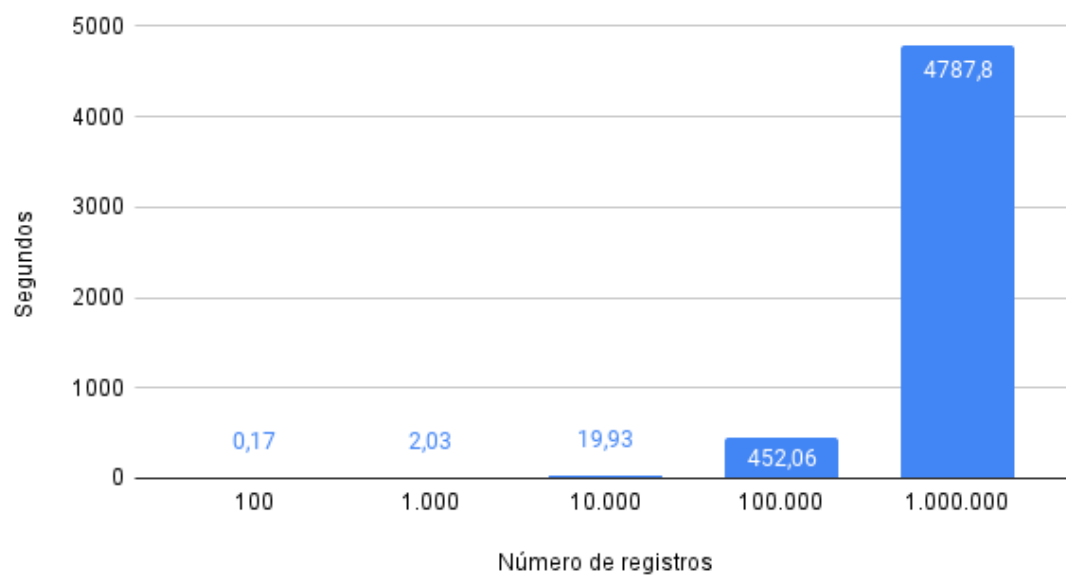


Figure 6. INSERT NUMBERS PostgreSQL.

INSERT NUMBERS - PostgreSQL

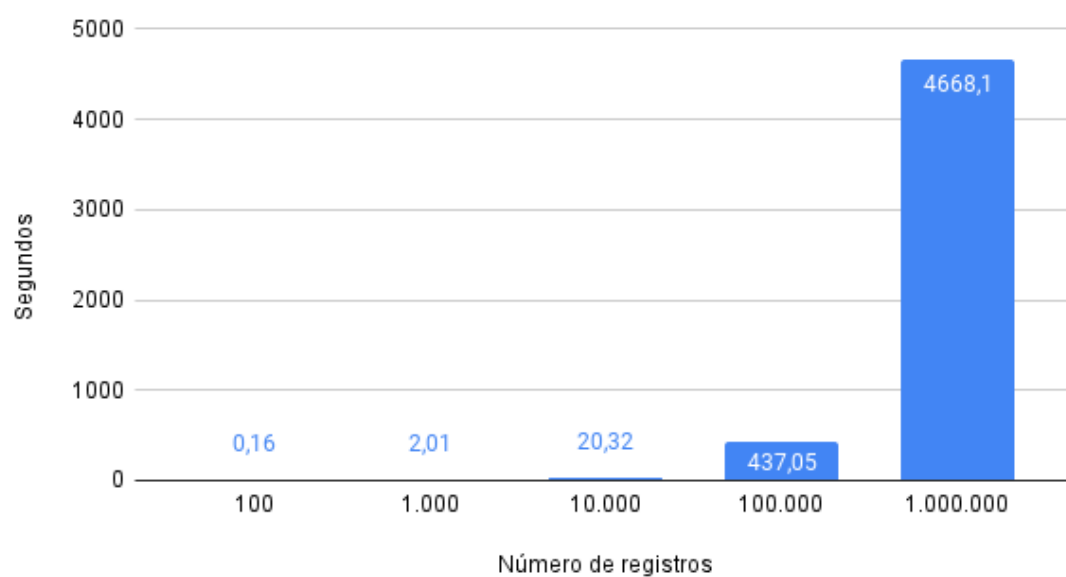


Figure 7. UPDATE NUMBERS PostgreSQL.

UPDATE NUMBERS - PostgreSQL

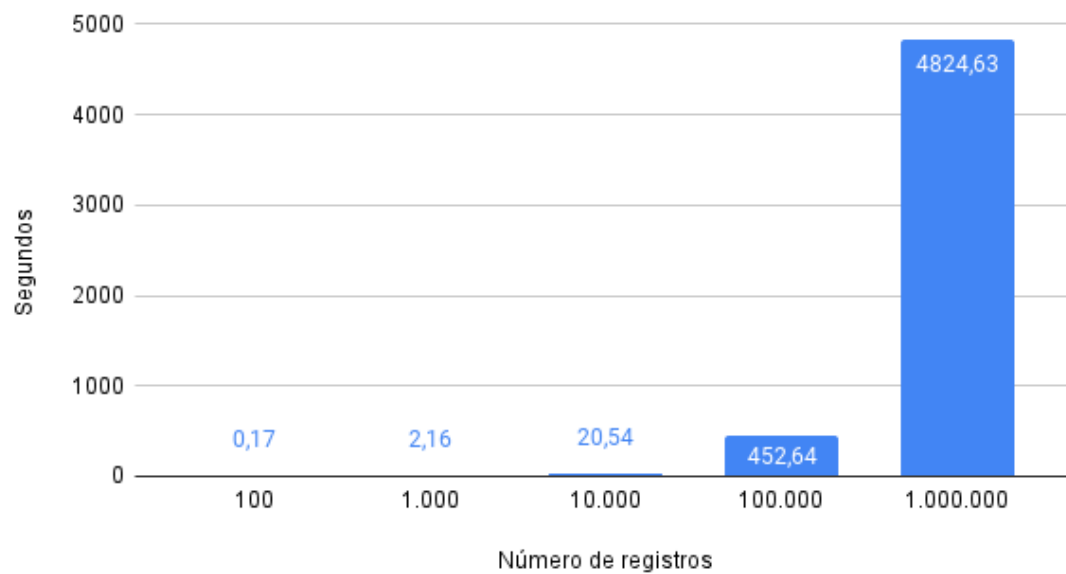


Figure 8. INSERT STRINGS MongoDB.

INSERT STRINGS - MongoDB

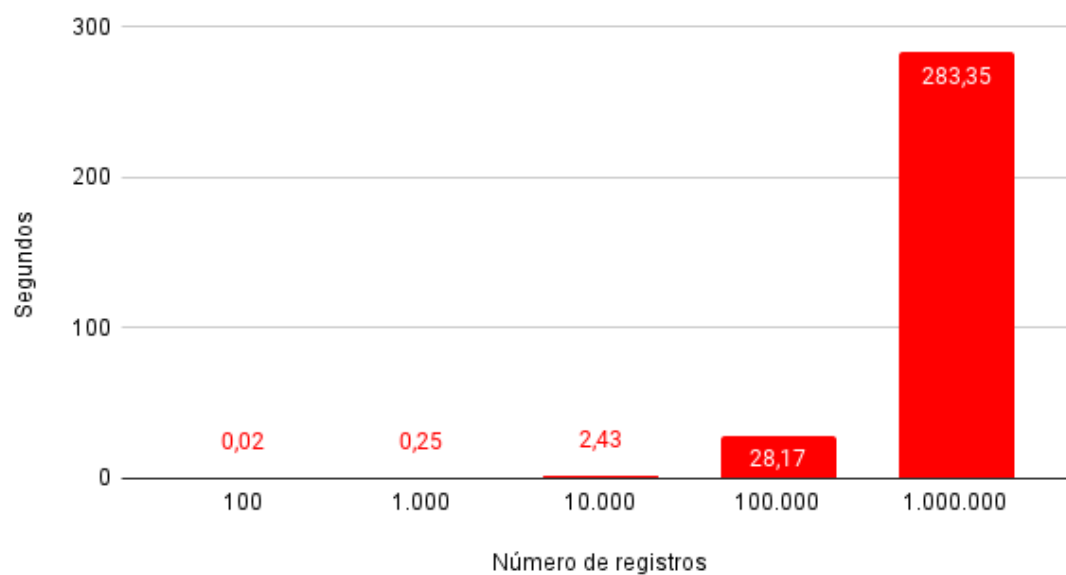


Figure 9. UPDATE STRINGS MongoDB.

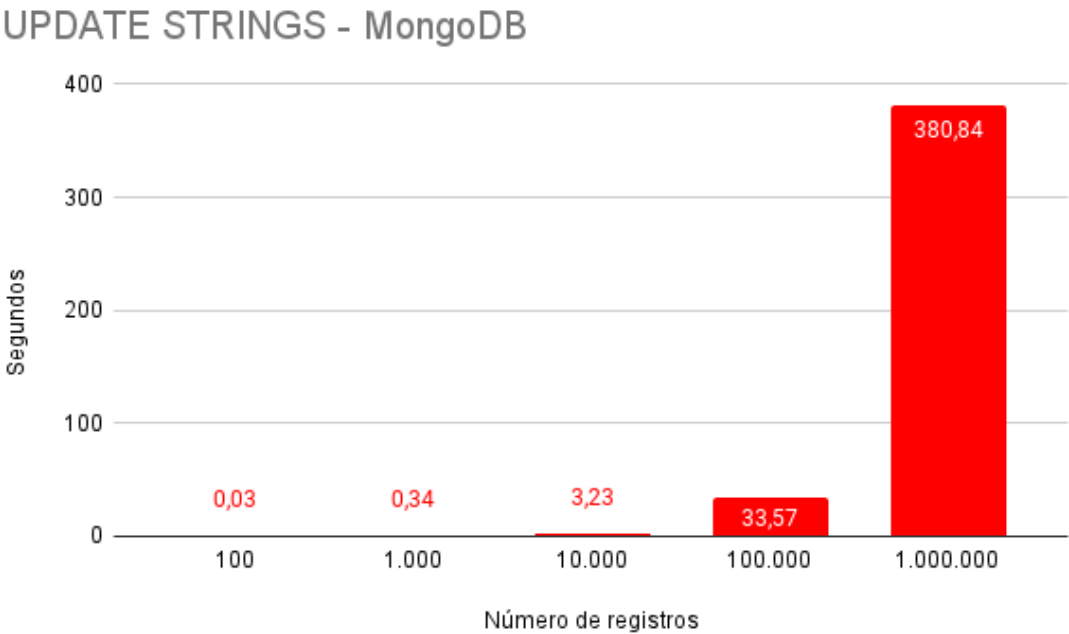


Figure 10. INSERT NUMBERS MongoDB.

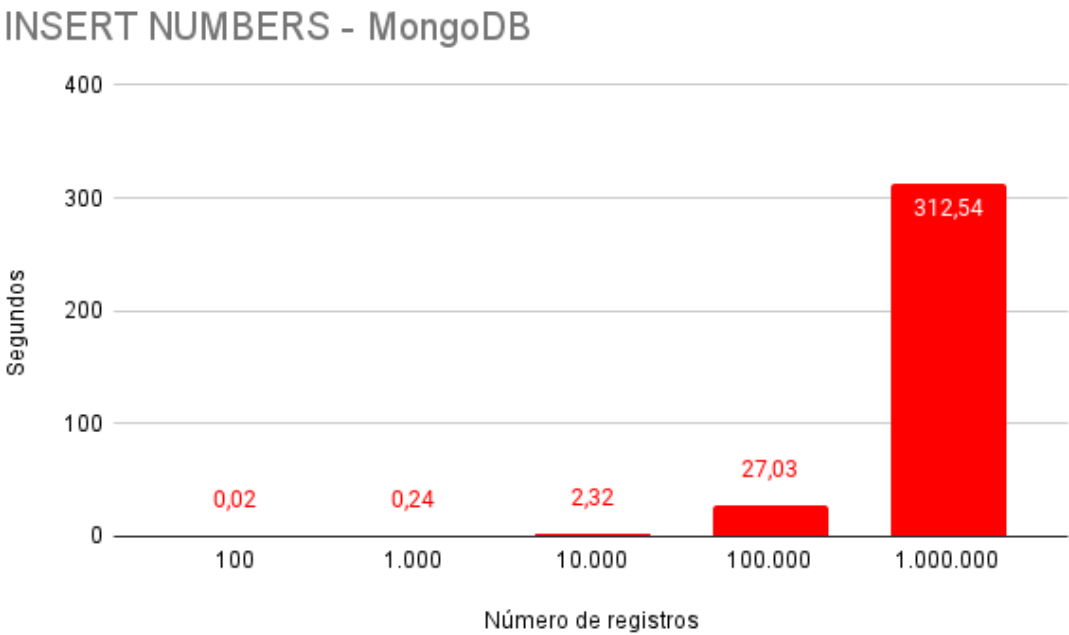


Figure 11. UPDATE NUMBERS MongoDB.

