

Universidade Paulista - UNIP

Peterson Mendonça de Oliveira – D29JHJ-9

**ANÁLISE COMPARATIVA EM AMBIENTE CONTROLADO DE BANCO DE
DADOS RELACIONAL E NÃO RELACIONAL**

Limeira - SP

2020

Universidade Paulista - UNIP

Peterson Mendonça de Oliveira – D29JHJ-9

**ANÁLISE COMPARATIVA EM AMBIENTE CONTROLADO DE BANCO DE
DADOS RELACIONAL E NÃO RELACIONAL**

Trabalho de Conclusão de Curso apresentado à banca examinadora da Universidade Paulista, como requisito parcial à obtenção do Bacharelado em Ciência da Computação sob a orientação do Professor Me. Sergio Eduardo Nunes.

Limeira - SP

2020

Universidade Paulista - UNIP

Peterson Mendonça de Oliveira – D29JHJ-9

**ANÁLISE COMPARATIVA EM AMBIENTE CONTROLADO DE BANCO DE
DADOS RELACIONAL E NÃO RELACIONAL**

Trabalho de Conclusão de Curso apresentado à banca examinadora da Universidade Paulista, como requisito parcial à obtenção do Bacharelado em Ciência da Computação sob a orientação do professor Me. Sergio Eduardo Nunes.

Aprovada em XX de XXXXX de 2020.

BANCA EXAMINADORA

Prof. Dr. Nome completo

Prof. Me. Nome completo

Prof. Esp. Nome completo

DEDICATÓRIA

Dedico a todos que me auxiliaram no desenvolvimento deste trabalho, a meu pai Ailton e minha mãe Ademildes que me fortaleceram e me deram esperança e a força necessária para continuar; assim como minha namorada Jheniffer que me auxiliou e me fortaleceu sempre que foi preciso, aos professores que me apoiaram e orientaram com suas instruções e experiências e pelo café necessário.

“Não há Bala de Prata”.

-Fred Brooks

RESUMO

O trabalho tem como intuito fornecer e apresentar um estudo científico e uma análise comparativa de duas tecnologias de gerenciamento de banco de dados. Sendo elas, o MySQL - My Structured Query Language um sistema de gerenciamento de dados relacional, utilizando de um modelo de cliente-servidor. O MongoDB, um sistema de gerenciamento de dados não relacional orientado a documentos no formato JSON - JavaScript Object Notation. Na construção do trabalho foi necessário o uso de diferentes tecnologias, dentre elas uma linguagem de programação, sendo essa a linguagem Python, uma linguagem de alto nível e orientada a objetos, de fácil construção de códigos, o Visual Studio Code um editor de texto multiplataforma disponibilizado pela Microsoft para o desenvolvimento de aplicações dentre outras tecnologias, onde será colocado à prova as características dos sistemas de gerenciamento de banco de dados em um experimento onde será feito a comparação com métricas de tempo de respostas dessas tecnologias, fornecendo um acervo de informações sobre os dois SGBD's e permitindo ter uma resposta objetiva de como eles trabalham com determinada quantidade de dados.

Palavra-Chave: MongoDB; MySQL; NOSQL; SGBD; SQL.

ABSTRACT

The work aims to provide and present a scientific study and a comparative analysis of two database management technologies. Being them, MySQL - My Structured Query Language is a relational data management system, using a client-server model. MongoDB, a non-relational document-oriented data management system in JSON - JavaScript Object Notation format. In the construction of the work, it was necessary to use different technologies, among them a programming language, this being the Python language, a high level and object-oriented language, easy to build codes, Visual Studio Code a multiplatform text editor made available by Microsoft for the development of applications, among other technologies, where the characteristics of database management systems will be tested in an experiment where the comparison with response time metrics of these technologies will be made, providing a collection of information about the two DBMS's and allowing to have an objective answer of how they work with a certain amount of data.

Key Words: MongoDB; MySQL; NOSQL; SGBD; SQL.

LISTA DE FIGURAS

FIGURA 01 – EXEMPLO DE MODELO RELACIONAL.	18
FIGURA 02 – PRIMEIRO EXEMPLO DE MODELO CLIENTE SERVIDOR.....	19
FIGURA 03 – SEGUNDO EXEMPLO DE MODELO CLIENTE SERVIDOR.	19
FIGURA 04 – EXEMPLO DE ESTRUTURA DE TABELAS EM MYSQL.....	22
FIGURA 05 – EXEMPLO DO FORMATO JSON.....	26
FIGURA 06 – EXEMPLO DE CHAVE-VALOR.....	27
FIGURA 07 – EXEMPLO DE BANCO DE DADOS EM MONGODB	Erro! Indicador não definido.
FIGURA 08 – IMAGEM DO BANCO DE DADOS CRIADO PARA O TRABALHO NO MODELO RELACIONAL	32
FIGURA 09 – IMAGEM DO BANCO DE DADOS CRIADO PARA O TRABALHO NO MODELO NÃO RELACIONAL	33
FIGURA 10 – IMAGEM DO RESULTADO EXPERIMENTAL DOS BANCOS DE DADOS	35
FIGURA 11 – MEDIÇÕES REALIZADAS NO BANCO DE DADOS DE MODELO RELACIONAL.....	36
FIGURA 12 – MANEIRA QUE FOI REALIZADO A BUSCA DE DADOS NO MODELO RELACIONAL.....	37
FIGURA 13 – MEDIÇÕES REALIZADAS NO BANCO DE DADOS DE MODELO NÃO RELACIONAL.....	37

LISTA DE ABREVIATURAS

API	Application Programming Interface – (Interface de Programação de Aplicativos).
BD	Banco de Dados.
IBM	International Business Machines.
IDE	Integrated Development Environment - (Ambiente de Desenvolvimento Integrado).
JSON	JavaScript Object Notation - (Notação de Objetos JavaScript).
mSQL	Mini SQL.
MYSQL	My Structured Query Language - (Minha Linguagem de Consulta Estruturada).
NoSQL	Not Only SQL – (Não Somente SQL).
SGBD	Sistema de Gerenciamento de Banco de Dados.
SGBDR	Sistema de Gerenciamento de Banco de Dados Relacional.
SO	Sistema Operacional.
SQL	Structured Query Language - (Linguagem de Consulta Estruturada).
PaaS	Platform as a Service – (Plataforma como Serviço).
XML	Extensible Markup Language – (Linguagem Extensível de Marcação Genérica).

SUMÁRIO

1. INTRODUÇÃO	13
1.1 OBJETIVO	14
1.2 JUSTIFICATIVA.....	14
1.3 METODOLOGIA	15
2. BANCO DE DADOS RELACIONAL	17
2.1 ARQUITETURA CLIENTE SERVIDOR.....	18
2.2 BANCO DE DADOS.....	20
2.3 SGBD.....	21
2.3.1 HISTÓRIA DO MYSQL	23
3. BANCO DE DADOS NÃO RELACIONAL	25
3.1 CHAVE-VALOR	26
3.2 MONGODB.....	27
3.2.1 HISTÓRIA DO MONGODB.....	28
4. PYTHON	29
5. EXPERIMENTO	31
6. RESULTADOS E ANÁLISES EXPERIMENTAIS	35
7. CONCLUSÃO	38
REFERÊNCIAS BIBLIOGRÁFICAS	39

1. INTRODUÇÃO

Com o passar dos anos a tecnologia vem se aprimorando e estabelecendo seu local de poder no mercado, atualmente ela está presente direta ou indiretamente no nosso cotidiano, tanto no nosso ambiente profissional, quanto no não profissional. Fornecendo com ela diversos benefícios como praticidade, agilidade, organização, dentre muitas outras formas de auxílio em ambientes profissionais e pessoais. Servindo não só como uma forma de exercer um trabalho, mas também como alimento intelectual e formas de entretenimento.

Por conta deste crescimento descomunal da tecnologia e de suas aplicações em diversas áreas, é facilmente notado o aumento massivo de dados que vem sendo gerado pelas diversas plataformas de tecnologias, tais dados tendem a crescer cada vez mais com o passar do tempo dificultando seu armazenamento e sua organização, gerando problemas para tratar tais dados.

Para evitar possíveis problemas que venham a ocorrer por conta do não tratamento dos dados, são utilizados SGBD's (Sistemas de Gerenciamento de Banco de Dados), esses sistemas têm como objetivo gerenciar os dados de uma forma que facilite o acesso a eles de maneira mais dinâmica e direta.

Segundo Heuser (1998, p. 16) "Um modelo de (banco de) dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados.". Atualmente no mercado existem vários tipos de sistemas de banco de dados, cada um visando uma característica distinta para um melhor gerenciamento dos dados, dentre os sistemas de gerenciamento de dados temos os mais conhecidos como os modelos relacionais e os mais novos como os não relacionais.

Os quais têm características distintas um do outro, fornecendo a possibilidade de serem aplicados em diferentes ocasiões, para o solucionamento de diversos problemas visando o melhor resultado possível.

1.1 OBJETIVO

Este trabalho tem como objetivo gerar uma comparação entre dois sistemas de banco de dados, elaborando assim um resultado e fornecendo um comparativo com o tempo de resposta entre os dois sistemas.

Serão abordados dois modelos de sistemas de gerenciamento de banco de dados, utilizando de um ambiente isolado para gerar medições mais precisas. Essas análises comparativas serão realizadas entre um banco de dados relacional (MySQL) e o banco de dados não relacional (MongoDB).

O MySQL é um modelo relacional, que também é de código aberto, o mesmo usa a linguagem SQL (Structure Query Language - Linguagem de Consulta Estruturada), por ser um modelo relacional o MySQL relacionar suas tabelas e dados para proporcionar um melhor resultado, sendo necessário a criação destas tabelas previamente. O MongoDB diferente do MySQL é um modelo não relacional e orientado a documentos, também conhecido como NoSQL (Not Only SQL - Não Somente SQL), sua estrutura de dados é adaptável e seu documento será criado no momento da inserção dos dados, ao contrário do MySQL não é necessário definir a estrutura previamente.

1.2 JUSTIFICATIVA

Nos dias atuais a tecnologia está sendo consumida em proporções nunca vista antes, tanto na área de entretenimento, quanto na área profissional seu consumo vem aumentando cada vez mais, cada plataforma com seus sistemas próprios. As múltiplas redes sociais que vêm sendo lançadas, gerando quantidades massivas de dados, que a população consome e gera informações muito rápido.

Dessa forma é necessário efetuar o gerenciamento de tais dados gerados, organizando e administrando as informações para que possam ser usadas de forma rápida e ágil quando necessário.

Para obter essa demanda será necessário utilizar sistemas de gerenciamento de dados conhecidos como SGBD's, e para realizar a organização dos dados existem os dois modelos principais, sendo eles o MySQL e o MongoDB, cada um com seus respectivos funcionamentos e características.

Em situações de aplicação profissional, conhecer qual dos modelos de gerenciamento de banco de dados (Relacional e NoSQL) é o mais adequado para o ambiente o qual será implementado.

1.3 METODOLOGIA

Inicialmente, o trabalho irá tratar do Banco de Dados Relacional, onde será abordado sobre o sistema de gerenciamento de banco de dados relacional, seu esquema de estruturação e construção, como características técnicas da mesma. Também será abordado as vantagens e desvantagens do uso desta tecnologia. Dentre os possíveis sistemas relacionais será abordado sobre o escolhido para este trabalho, no caso MySQL, abordando sua história profundamente como criação e desenvolvedores abrangendo suas características e curiosidades.

Na segunda etapa, será abordado sobre o sistema de gerenciamento de banco de dados não relacional, descrevendo como funciona os modelos a estruturação e a arquitetura, abordando também as características técnicas, além de suas vantagens e desvantagens. O sistema de banco de dados não relacional escolhido para este trabalho foi o MongoDB, onde irá ser abordado mais a fundo suas informações como sua criação abrangendo sua história, características e curiosidades.

Na terceira etapa, será abordado sobre a Linguagem de Programação utilizada para a criação dos dados. A linguagem escolhida foi, Python uma linguagem de orientação a objetos, onde será abrangido sua história e funcionamento, como também sua estruturação. Também será mencionado sobre a IDE utilizada para a construção do código.

Na quarta parte, será abordado a execução do Experimento, onde será realizado uma vasta bateria de testes nos dois bancos, utilizando de um ambiente controlado para melhor tempo de resposta de suas buscas, sendo necessário também o desenvolvimento de um algoritmo de geração de dados randômicos para alimentar os dois bancos de dados e para realizar as medições de seus tempos de resposta.

A quinta e sexta parte, trataram sobre os Resultados Experimentais, onde apresentará os resultados obtidos após a realização da aferição do tempo. A análise dos resultados onde será realizado a comparativa dos dados experimentais obtidos para obter a conclusão do trabalho.

2. BANCO DE DADOS RELACIONAL

Um sistema de banco de dados relacional é uma ferramenta de armazenamento da qual proporciona uma segurança das informações para que as mesmas estejam salvas, permitindo assim que sejam recuperadas para uso quando necessário, além de permitir se preciso a implementação de funcionalidades.

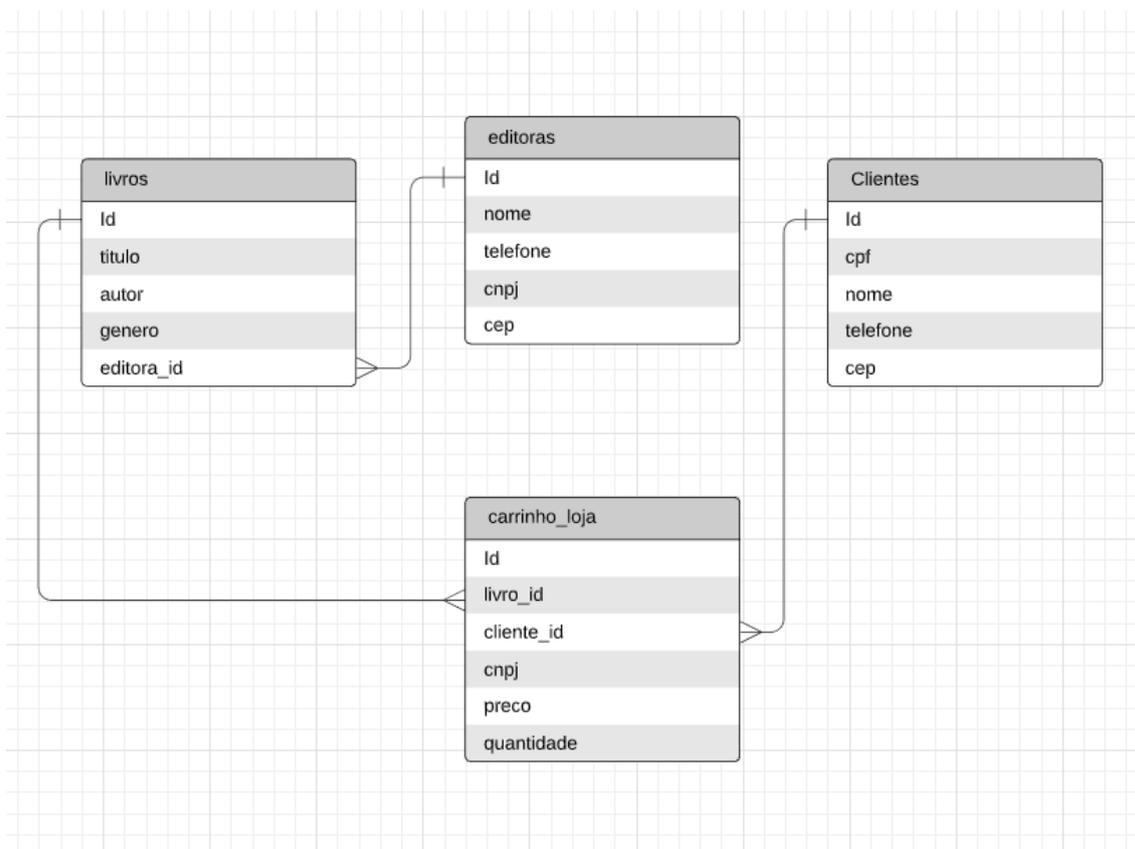
“O modelo de dados relacional foi introduzido inicialmente por Ted Codd, da IBM Research, em 1970, em um artigo clássico (Codd, 1970), que atraiu atenção imediata devido a sua simplicidade e base matemática.” (ELMASRI; NAVATHC; p.38, 2011)

Artigo conhecido como *Relational Model of Data for Large Shared Data Banks* (Modelo relacional de dados para grandes bancos de dados compartilhados) criado por Edgar Frank Ted Codd e futuramente aperfeiçoado por Chris Dare e Hugh Darwen num modelo geral de dados.

No ano de 1995 demonstraram que o modelo relacional poderia trabalhar com a orientação a objetos sem afetar seus elementos principais.

Entidades são conhecidas como “tabelas” ou “Ficheiros” sendo retratadas graficamente na forma de um retângulo. Associação é a relação que atribui uma conexão de uma tabela X com uma tabela Y. Na Figura 01 demonstra um exemplo de estrutura do modelo relacional.

Figura 01 – Exemplo de Modelo Relacional.



Fonte: Elaborado pelo Autor.

A figura permite demonstrar uma estrutura de quatro tabelas onde é feito o relacionamento delas pelas colunas de sua estrutura.

2.1 ARQUITETURA CLIENTE SERVIDOR

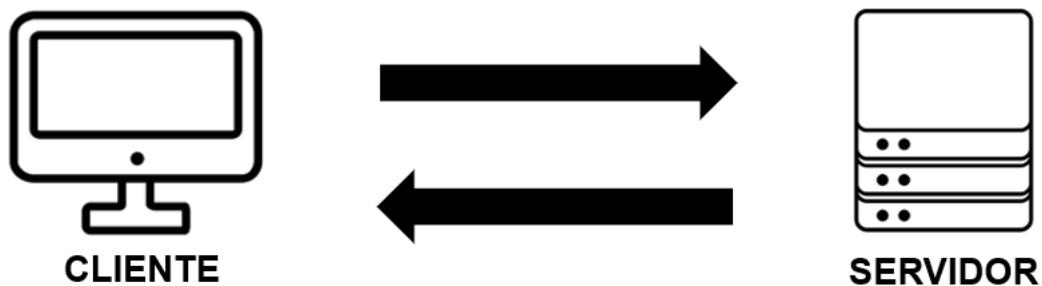
O modelo de arquitetura de cliente-servidor efetua seu processamento dos dados de maneira em que se é dividido o trabalho realizado em módulos, que são conhecidos como cliente e servidor.

“Em uma arquitetura básica de SGBD cliente/servidor a funcionalidade do sistema é distribuída entre dois tipos de módulos.” (ELMASRI; NAVATHC; p.19, 2011)

Conforme defende Elmasri e Navathc (2011), a arquitetura cliente-servidor é um sistema distribuído, sendo composto de fornecedores ou gerenciadores de recursos delimitados como “servidores” e por atores de requisição de serviços ou recursos conhecidos como “clientes”. Cada um deles

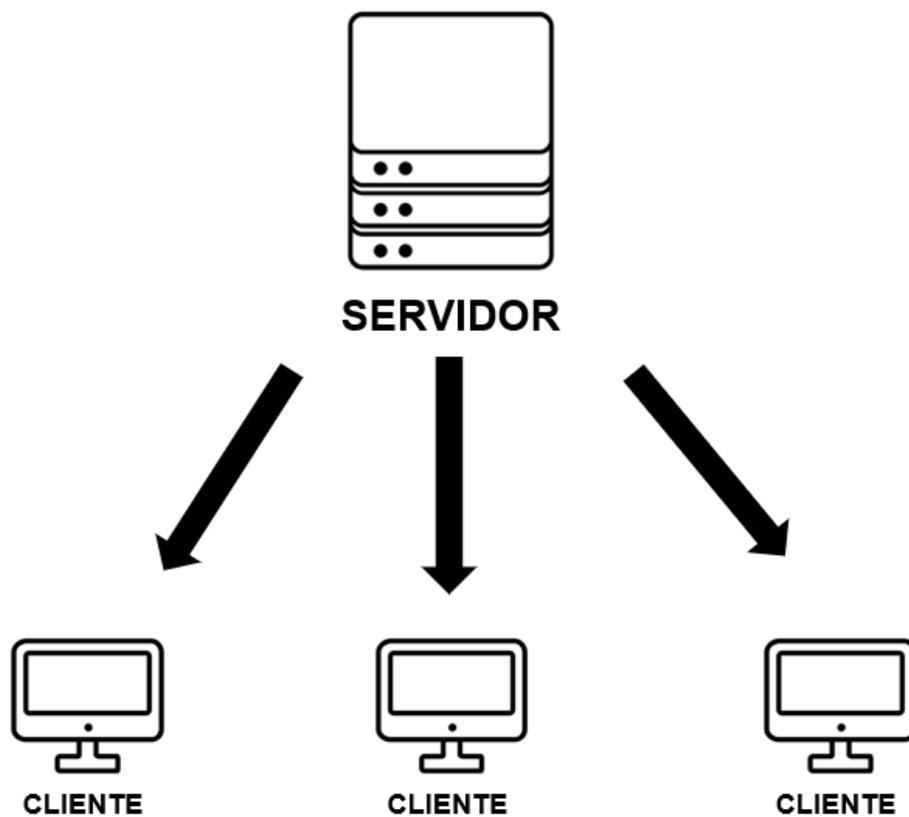
tendo sua respectiva característica e responsabilidade, exemplo de cliente-servidor na Figura 02 e Figura 03:

Figura 02 – Primeiro Exemplo de Modelo Cliente Servidor.



Fonte: Elaborado pelo Autor.

Figura 03 – Segundo Exemplo de Modelo Cliente Servidor.



Fonte: Elaborado pelo Autor.

“O módulo cliente normalmente é projetado para executar em uma estação de trabalho ou computador pessoal.” (ELMASRI; NAVATHC; p.19, 2011)

O módulo cliente é responsável por efetuar as requisições dos dados sendo necessário proporcionar uma interação com o usuário, fornecendo de maneira ágil interfaces de fácil compreensão, como interfaces gráficas, formulários, dentre outros. Assim como descrito por Elmasri e Navathc (2011), o módulo cliente é utilizado em plataformas que sejam acessíveis a usuários, sendo na maioria das vezes um computador pessoal ou corporativo.

“O outro tipo de módulo, chamado de módulo servidor, é normalmente responsável pelo armazenamento de dados, acesso, pesquisa e outras funções.” (ELMASRI; NAVATHC; p.19, 2011)

Diferente do módulo cliente, que envia requisições de busca de dados para serem realizadas, o servidor é quem vai realizar o processamento dessas buscas, onde realizará o gerenciamento das informações requisitadas pelo cliente e assim retornando o resultado obtido.

Salientando que em uma arquitetura cliente-servidor, o processamento gerado tanto pelo servidor, quanto pelo cliente deve ser equilibrado para serem considerados um modelo cliente-servidor, caso haja um peso para algum dos lados ele não é o modelo descrito.

2.2 BANCO DE DADOS

Os bancos de dados fazem parte do nosso dia-a-dia há muito tempo, um exemplo pode ser a própria lista telefônica. Os bancos de dados são um amplo agrupamento de dados que serão organizados e armazenados de forma estruturada.

Não só hoje como antigamente é comum que empresas e sistemas gerem diversos dados, tais dados devem ser armazenados, organizados e disponibilizados para qualquer necessidade futura que possa acontecer. Para facilitar o gerenciamento desses dados e do próprio banco, é utilizado os SGBD's (Sistema de Gerenciamento de Banco de Dados), tais sistemas

facilitam a manipulação e o gerenciamento dos dados tornando o serviço mais fácil e prático.

2.3 SGBD

Todo o processo realizado nos bancos de dados passará pelo sistema SGBD, sendo responsável por todos os procedimentos realizados assim como, salvar, fornece uma interface de acesso, efetua a conexão de dados e metadados, dentre muitos outros.

“O Sistema Gerenciador de banco de dados (SGBD) ou Data Base Management System (DBMS) é um software utilizado para administrar um banco de dados, permitindo criar, modificar, eliminar e inserir dados no mesmo.” (GARCIA; SOTTO; p.13, 2019)

Como dito por Garcia e Sotto (2019), um SGBD ou Sistema de Gerenciamento de Banco de Dados é um agrupamento de softwares que se tornam responsáveis pelo gerenciamento dos Dados trabalhados, ele é responsável pela organização e recuperação dos dados, inserção e manipulação desses dados.

O primeiro surgimento de um SGBD no mercado foi ao final da década de 60, utilizando como base de sua estrutura, sistemas de arquivos arcaicos os quais não conseguiam efetuar o controle de concorrência sobre múltiplos acessos ou mesmo processos.

A evolução dos Sistemas de gerenciamento de Dados se vê possível utilizando as estruturas de sistemas de armazenamento em disco, permitindo a criação de estruturas de dados melhores para o armazenamento de informações. Com sua evolução os SGBD's começaram a empregar múltiplas formas de representação de modelos de dados para retratar tais informações que estariam contidas nos bancos.

O MySQL é um grande servidor de banco de dados relacional, SGBDR (Sistemas de gerenciamento de banco de dados relacional), utilizando do modelo SQL (Structured Query Language - Linguagem Estruturada para Pesquisas) sendo altamente veloz, o mesmo funciona no modelo cliente servidor que também é um software de código aberto. Abaixo na Figura 04, demonstra como é feita a estrutura do banco de dados MySQL em estrutura de código.

Figura 04 – Exemplo de Estrutura de tabelas em MySQL

```
1 create table if not exists clientes(  
2     id bigint unsigned primary key not null auto_increment,  
3     cpf varchar (255) not null,  
4     nome varchar (255) not null,  
5     telefone varchar (255) not null,  
6     cep varchar (255) not null  
7 );  
8  
9 create table if not exists editoras(  
10    id bigint unsigned primary key not null auto_increment,  
11    nome varchar (255) not null,  
12    telefone varchar (255) not null,  
13    cnpj varchar (255) not null,  
14    cep varchar (255) not null  
15 );  
16  
17 create table if not exists livros(  
18    id bigint unsigned primary key not null auto_increment,  
19    titulo varchar (255) not null,  
20    autor varchar (255) not null,  
21    genero varchar (255) not null,  
22    editora_id bigint unsigned not null,  
23    constraint editora_id foreign key (editora_id) references editoras(id)  
24 );  
25  
26 create table if not exists carrinho_loja(  
27    id bigint unsigned primary key not null auto_increment,  
28    livro_id bigint unsigned not null,  
29    cliente_id bigint unsigned not null,  
30    cnpj varchar (255) not null,  
31    preco double (8,2) not null,  
32    quantidade int (10) not null,  
33    constraint livro_id foreign key (livro_id) references livros(id),  
34    constraint cliente_id foreign key (cliente_id) references clientes(id)  
35 );
```

Fonte: Elaborado pelo Autor.

2.3.1 História do MySQL

O MySQL foi desenvolvido por Michael Widenius da companhia suíça TcX. Que inicialmente criou um bando de dados conhecido como UNIREG, onde o mesmo foi sendo reescrito em diversas linguagens diferentes, o problema principal do UNIREG seria overhead que no caso é a sua sobrecarga, sendo um problema muito grande no cenário de aplicações web em que as páginas deveriam ser dinâmicas.

“O MySQL foi originalmente desenvolvido pela empresa sueca TCX, que necessitava de um servidor de banco de dados que operasse com grandes escalas de dados rapidamente sem exigir caríssimas plataformas de hardware. No início eles utilizavam o mSQL, mas depois de alguns testes chegaram à conclusão que o mSQL não era rápido nem flexível o suficiente para as necessidades existentes.” (QUESADA et al. p. 17, 2001)

Sendo assim a TcX teve que recorrer a outras tecnologias da época, dentre elas o mSQL o qual era baseado no sistema SQL, porém, o mesmo possuía peculiaridades dentre elas, não possuía suporte a índices e com um desempenho abaixo do sistema UNIREG.

Por este motivo Michael entrou em contato com David Hughes o desenvolvedor do sistema mSQL com o objetivo de criar um sistema que juntaria os pontos positivos de ambas as tecnologias, com David animado e o interesse da empresa TcX foi iniciado o desenvolvimento de uma nova tecnologia, porém, sem reinventar a roda, utilizando da estrutura do UNIREG o qual já estava feita, e uma grande massa de utilitários feitos para o mSQL. Sendo então em maio de 1995 que a primeira versão MySQL foi lançada oficialmente, um dos parceiros da empresa TcX sugeriu que seria de interesse geral a distribuição do servidor na internet, o resultado foi um sucesso com o MySQL distribuído de forma gratuita.

Será abordado algumas das principais características pertencentes ao sistema de gerenciamento de banco de dados MySQL. Assim como Quesada et al. (2001) fornece e exemplifica uma das características pertencentes ao MySQL:

“O servidor de banco de dados MySQL é extremamente rápido, confiável, e fácil de usar. O Servidor MySQL também tem um conjunto de recursos muito práticos desenvolvidos com a cooperação dos próprios usuários.” (QUESADA et al. p. 17, 2001)

Além da característica apresentada por Quesada (2001), sua base de construção é feita em cima da linguagem de programação C e C++, podendo ser realizado os testes por diferentes compiladores e disponibilizados por seus criadores para acesso livre, o MySQL é um sistema de código aberto, e um sistema multiplataforma, possibilitando suportar diferentes SO's dentre eles os, Win32, unix, Linux, dentre outros. Permitindo suporte a API's de diferentes linguagens como, Python, Java, PHP e outros.

3. BANCO DE DADOS NÃO RELACIONAL

Divergindo do modelo relacional, não há a necessidade da estruturação de dados previamente como é feito em modelos relacionais, isso porque o modelo não relacional se adapta e gera um novo campo na estrutura de dados atuais caso ela não exista sem ter que modificar dados já existentes.

“Surgiu uma nova categoria de Banco de Dados, chamada NoSQL (Not Only SQL), que foi proposta com o objetivo de atender aos requisitos de gerenciamento de grandes volumes de dados, semi-estruturados ou não estruturados, que necessitam de alta disponibilidade e escalabilidade.” (LÓSCIO, p.01, 2016)

Como retrata LÓSCIO (2016), os bancos de dados não relacionais são conhecidos pelo termo NoSQL (Não apenas SQL), porém, há quem diga que o termo correto é “Não SQL”, onde como seu próprio nome já diz, o mesmo não utiliza do SQL. Os bancos não-relacionais diferentemente do pensamento popular podem armazenar dados relacionais, porém, de maneira diferente do modelo relacional em si.

Nos Bancos de dados orientados a documentos sua estrutura tem como base o método de chave-valor sendo um esquema extremamente flexível, o mesmo utiliza do formato JSON.

O modelo de banco orientado a documentos é composto por um conjunto de campos de chaves e valores. Salientando que o modelo orientado a documentos não exige que o esquema estrutural do banco de dados seja previamente pensado, o mesmo é adaptável, que nada mais é que, a cada inserção de um valor caso não haja um determinado campo, o mesmo irá criá-lo em sua estrutura de armazenamento, exemplo do modelo JSON na Figura 05:

Figura 05 – Exemplo do formato JSON

```
{
  "cpf": "45137449809",
  "nome": "Peterson Mendonça",
  "telefone": "34249871",
  "cep": "13487013",
  "pedidos": [
    {
      "preco": 39.90,
      "quantidade": 40
    }
  ]
}
```

Fonte: Elaborado pelo Autor.

Um banco de dados não relacional não fica limitado ao relacionamento de tabelas como no modelo relacional, permitindo que o mesmo seja flexível em sua estrutura.

O modelo não relacional permite uma facilidade muito grande caso seja necessário a inserção de novos dados a uma estrutura já existente, por exemplo, se uma estrutura tiver três campos e a inserção dos novos dados tiver quatro campos, a estrutura não relacional irá adicionar o novo campo, não sendo necessário a alteração dos dados já existentes. Também conta com uma facilidade no gerenciamento de uma grande massa de dados. Contando com menos manutenção e custo bem reduzido.

3.1 CHAVE-VALOR

O modelo de Chave-valor, é um estilo de banco de dados não relacional, onde é armazenado um conjunto que compõem um par de chave-valor, a chave prestará o serviço de identificador na estrutura. Esta chave-valor não tem um valor fixo pré-definido, permitindo abranger desde objetos complexos ou simples. “O conteúdo pode ser armazenado em qualquer formato, seja uma string, inteiro, matriz, ou objeto.” (TOTH, p. 3, 2011).

Assim como diz Toth (2011), o armazenado pode ser realizado com diferentes formatos desde inteiro, strings, objetos e outros assim como na figura 06.

Figura 06 – Exemplo de chave-valor

KEY	VALUE
KEY 1	AAAA
KEY 2	AAAA, BBBB
KEY 3	AAAA, BBBB, CCCC
KEY 4	1245

Fonte: Elaborado pelo Autor.

3.2 MONGODB

O MongoDB é um SGBD (Sistema de Gerenciamento de banco de dados) onde o mesmo é orientado a documentos, sendo um sistema de código aberto e multiplataforma, utiliza do formato JSON (JavaScript Object Notation - Notação de Objetos JavaScript), proporcionando uma tolerância a falhas como também fornecendo consistência.

“O MongoDB é considerado um dos projetos mais notáveis NoSQL devido sua linguagem de consulta de alta performance, baseada em documentos e facilidade de migração de um banco de dados relacional, pelo fato de suas consultas serem convertidas facilmente.” (SOLAGNA; LAZZARETTI, p. 8, 2016)

Assim como diz Solagna e Lazzaretti, o MongoDB é amplamente eficiente e sua possibilidade de fornecer a migração de um modelo relacional para o NoSQL é facilmente realizado, facilitando caso necessário a mudança da estrutura inicialmente feita em um projeto relacional.

Diferente de um bando de dados relacional o MongoDB não fica preso a características estruturais de seus dados, como no relacional onde deve-se declarar as tabelas e colunas previamente, o MongoDB se adapta conforme a aplicação desejar, a cada inserção de novos dados, assim criando uma

estrutura com todos os dados necessários, vale frisar que, quando realizado uma nova inserção de dados no MongoDB, os valores previamente inseridos não terão mudanças, evitando a necessidade de alterar os dados que já se encontram no sistema.

3.2.1 História do MongoDB

O MongoDB é um banco de dados não relacional orientado a documentos, utilizando do formato JSON de código aberto. O MongoDB é produzido pela empresa MongoDB Inc. antigamente conhecida como 10gen, empresa americana responsável não só pelo desenvolvimento como também pelo suporte prestado comercialmente.

“O MongoDB inicialmente foi desenvolvido como um componente de serviço pela empresa 10gen em outubro de 2007, passando a ser um software open source em 2009.” (SOLAGNA; LAZZARETTI, p. 7, 2016)

Assim como diz Solagna e Lazzaretti, a empresa 10gen surgiu por volta do ano de 2007 com seu cofundador Dwight Merriman e pelo CEO Kevin P. Ryan e pelo engenheiro Eliot Horowitz. Inicialmente a empresa buscava o desenvolvimento de uma aplicação de computação em nuvem conhecida como PaaS (Plataforma como Serviço), mas a empresa sofreu alguns contratemplos em seu desenvolvimento, pois, não conseguiam adequar um sistema de banco de dados para computação em nuvem.

Por conta desse inconveniente foi iniciado o desenvolvimento de um banco de dados não relacional onde fora chamado de MongoDB, vendo seu potencial foi então descartado o sistema em nuvem anterior para ser melhor trabalhado o MongoDB, foi só em 2009 que a empresa 10gen anunciou oficialmente o MongoDB com código aberto.

O MongoDB é um sistema com orientação a documentos, diferente do modelo relacional que lida com registros, o MongoDB trabalha com documentos e conta com um código aberto escrito em C++, sendo um sistema multiplataforma no formato JSON.

4. PYTHON

Python é uma linguagem de programação orientada a objetos, sendo uma linguagem *high level Language* (Linguagem de Alto Nível). De acordo com Luzzardi (2017):

Python é uma linguagem de programação interpretada (onde cada linha é compilada e executada, uma por vez), ou seja, não é compilada (onde todo o texto é traduzido para linguagem de máquina e posteriormente executado). (apud, DA SILVA, p. 96, 2018)

Assim como dito por da Silva (2018), o Python executa linha por linha de seu código, diferente de outras linguagens, que realizam a leitura do seu conteúdo de forma completa para só assim executarem.

Uma linguagem de programação orientada a objetos tem a finalidade de realizar a estabilidade e o controle de projetos em larga escala, necessitando de uma programação mais complexa que supra as necessidades requisitadas. Mesmo que seja uma linguagem orientada a objetos, o Python permite ser trabalhado de maneira mais fácil, utilizando de códigos sequenciais sem a orientação de objetos inserida no código chegando ao objetivo visado.

A linguagem de programação Python assim como diversas linguagens podem ser utilizadas logo após sua instalação pelo terminal próprio fornecido pela linguagem, ou pelo terminal do próprio SO (Sistema Operacional). Porém, existem diversas IDE's (Integrated Development Environment - Ambiente de Desenvolvimento Integrado), essas IDE's são interfaces de fácil manuseio, proporcionando uma facilidade visual e no código já que muitas delas fornecem atalhos de comandos.

O Python teve seu desenvolvimento próximo da década de 90 no ano de 1989, desenvolvido por Guido van Rossum em um campo focado em matemática e em ciência da computação chamado CWI Software que se

encontra na Holanda, a linguagem de programação Python vem como sucessora da antiga linguagem ABC. Segundo da Silva:

Ele escolheu este nome pois a empresa onde trabalhava remetia o nome de seus projetos a programas, séries ou qualquer outra designação que remetesse à televisão onde, num momento de humor "irreverente" em sua vida, escolheu este nome pois era fã da série britânica Monty Python's Flying Circus, porém Python também é um gênero de serpentes da família Pythonidae, que inclusive estampa a capa do primeiro livro publicado sobre a linguagem pela editora O'Reilly e esta editora sempre usa, estampado na capa de seus livros publicados, a imagem de um animal. (p. 97, 2018)

Sim como dito seu nome foi inspirado pelo seu grande apreço quanto ao grupo humorístico britânico, conhecido como Monty Python grupo conhecido no meio cinematográfico em vários locais do mundo.

O Python atualmente pertence a Python Software Foundation, como dito anteriormente o mesmo é uma linguagem de código aberto, a mesma é uma linguagem de multiparadigmas sendo capaz de realizar uma solução de diversas formas.

5. EXPERIMENTO

O experimento realizado tem como objetivo, o levantamento de dois sistemas de banco de dados os quais foram escolhidos pelo convívio no meio institucional e profissional, o que forneceu um desejo de ver como tais sistemas se comportam caso fossem colocados a diversos testes, os sistemas escolhidos foram, um deles do modelo SGBDR, logo um banco de dados relacional, o sistema escolhido foi o MySQL. O outro modelo foi o NoSQL, que no caso seria um modelo de banco de dados não relacional, o sistema escolhido foi o MongoDB.

Para ser possível a realização deste trabalho, foi necessário o desenvolvimento de uma aplicação onde utilizaria da linguagem de programação Python a qual já mantinha um nível de conhecimento da mesma, porém, foi utilizado do Python *documentation* (2020), para melhor estruturação do código, seguindo a normativa correta da linguagem. O Python fornece uma construção ágil e fácil, permitindo gerar um código que tem como finalidade criar a estrutura dos dois bancos de dados.

Para realizar os testes dos dois sistemas foi necessário o desenvolvimento de um banco de dados o qual teria (N) quantidades de campos, os bancos deveriam ter a maior proximidade possível tanto no modelo relacional, quanto no modelo não relacional, fornecendo assim uma comparação mais real.

O banco de dados desenvolvido no modelo relacional deveria conter interações entre suas tabelas proporcionando o relacionamento delas, conforme pode ser observado na Figura 08:

Figura 08 – Imagem do Banco de Dados Criado para o Trabalho no Modelo Relacional

```
1 ▼ create table if not exists clientes(  
2     id bigint unsigned primary key not null auto_increment,  
3     cpf varchar (255) not null,  
4     nome varchar (255) not null,  
5     telefone varchar (255) not null,  
6     cep varchar (255) not null  
7 );  
8  
9 ▼ create table if not exists editoras(  
10    id bigint unsigned primary key not null auto_increment,  
11    nome varchar (255) not null,  
12    telefone varchar (255) not null,  
13    cnpj varchar (255) not null,  
14    cep varchar (255) not null  
15 );  
16  
17 ▼ create table if not exists livros(  
18    id bigint unsigned primary key not null auto_increment,  
19    titulo varchar (255) not null,  
20    autor varchar (255) not null,  
21    genero varchar (255) not null,  
22    editora_id bigint unsigned not null,  
23    constraint editora_id foreign key (editora_id) references editoras(id)  
24 );  
25  
26 ▼ create table if not exists carrinho_loja(  
27    id bigint unsigned primary key not null auto_increment,  
28    livro_id bigint unsigned not null,  
29    cliente_id bigint unsigned not null,  
30    cnpj varchar (255) not null,  
31    preco double (8,2) not null,  
32    quantidade int (10) not null,  
33    constraint livro_id foreign key (livro_id) references livros(id),  
34    constraint cliente_id foreign key (cliente_id) references clientes(id)  
35 );
```

Fonte: Elaborado pelo Autor.

Para a criação do banco de dados do modelo não relacional, o mesmo deveria conter o máximo possível de campos para gerar a maior semelhança possível com o banco desenvolvido para o modelo relacional, sendo ele:

Figura 09 – Imagem do Banco de Dados Criado para o Trabalho no Modelo Não Relacional

```
1 db.clientes.insert({
2   cpf: "45137449809",
3   nome: "Peterson Oliveira",
4   telefone: "34249871",
5   cep: "13497013",
6   pedidos:
7     {
8       preco: 39.90,
9       quantidade: 40,
10      livros:
11        {
12          titulo: "A volta dos que não foram 2",
13          autor: "Jacks Blacksu",
14          genero: "Aventura",
15          editora:
16            {
17              nome: "The Time 2",
18              telefone: "97865421",
19              cnpj: "56842521384985",
20              cep: "13254287"
21            }
22        }
23      }
24  })
```

Fonte: Elaborado pelo Autor.

Após ao desenvolvimento da estrutura de banco de dados onde foi realizado os testes, tanto no modelo relacional quanto no não relacional, foi necessário a inserção de uma grande massa de dados para possibilitar a realização dos testes comparativos entre os dois bancos de dados, para sanar tal necessidade, foi necessário o planejamento e o desenvolvimento de uma aplicação, a linguagem escolhida foi o Python, da qual já contava com um certo conhecimento da mesma, possibilitando a criação de um código ágil e limpo. O algoritmo criado permite a geração de dados para os bancos, podendo estrutura-lo de forma que se adapte a estrutura do banco necessário, o mesmo fornecesse uma quantidade de dados massiva de maneira randômica, permitindo definir quantos dados seriam criados para a inserção nos dois bancos.

Após a primeira versão do algoritmo, foi aprimorado a aplicação inicial em Python utilizando do pacote de nome *Fake* (Faraglia, 2014) o qual tem como objetivo a geração de dados falsos, o mesmo pertencendo à linguagem Python e acessível a outras linguagens, a qual auxiliaria na geração de dados de maneira randômica, o que permitiu a criação de um código ágil e rápido que supriria a necessidade em questão.

Foi utilizado da interface Visual Studio Code para o desenvolvimento do código em Python. O código desenvolvido tem a finalidade não só de gerar os dados, mas o de criar o banco, gerar os dados de maneira randômica, inserir os dados gerados nos bancos e medir o tempo de resposta que os mesmos levam para gerar a busca dos dados tanto o relacional quanto o não relacional, o código estará disponível no GitHub: <https://github.com/PetersonMendonca/Gerador-de-Dados.git>

6. RESULTADOS E ANALISES EXPERIMENTAIS

Os testes foram realizados em uma máquina isolada, onde a mesma iria apenas realizar as medições dos bancos, a medição dos dois bancos foi feita utilizando a aplicação em Python disponibilizada no GitHub já descrito neste trabalho. Tal máquina era composta dos seguintes requisitos:

- Processador: Intel Core i5-9300H CPU @ 2.40GHz 240Ghz.
- Memória RAM: 24 GB 2666MHz e DDR4.
- SSD M2 120 GB.

Após a realização de diversas repetição e medições, foram efetuadas comparações com resultados iniciais obtidos onde constava uma diferença grande de tempo entre os dois sistemas de banco de dados. Inicialmente os dois sistemas trabalham com o mesmo valor de dados e a medições foram efetuadas em uma máquina focada apenas em realizar as medições, foi definido um *range*, onde seria realizado 10 medições e tirado o menor e maior valor retornado.

Figura 10 – Imagem do resultado experimental dos bancos de dados

Tipo BD	Quantidade de Dados	Maior tempo (seg.)	Menor Tempo (seg.)
NÃO RELACIONAL	100000	0.976	0.833
RELACIONAL	100000	132.372	113.147

Fonte: Elaborado pelo Autor.

Após uma comparação e medição entre os bancos de dados dos modelos relacional e não relacional, chega o momento de efetuar uma análise em cima dos resultados obtidos. Foram realizadas múltiplas medições dos bancos, com diferentes quantidades de dados neles, em um ambiente isolado, o qual proporciona uma medição mais assertiva em cima das medições realizadas. Os bancos se comportaram de maneira totalmente diferente um do outro.

Como dito anteriormente, foi realizado como ponto de partida um experimento com 100 000 dados nos dois bancos, os quais retornaram o tempo

já demonstrado na Figura 10. Utilizando esse comparativo inicial foi então preparado um primeiro experimento em que teria o objetivo de verificar quantos dados seriam necessários para o mesmo banco relacional se equiparar com o tempo de resposta do banco não relacional.

Deve ser destacado que os testes realizados foram em um ambiente isolado de outras aplicações em andamento no mesmo tempo, além de, os dois bancos estarem sendo utilizados com seu meio natural de uso, sem nenhum aprimoramento de funcionamento. Na figura 11, é demonstrado o tempo obtido:

Figura 11 – Medições Realizadas no Banco de dados de Modelo Relacional

PRIMEIRO EXPERIMENTO			
Tipo BD	Quantidade de Dados	Maior tempo (seg.)	Menor Tempo (seg.)
NÃO RELACIONAL	100000	0.976	0.833
RELACIONAL	100000	132.372	113.147
RELACIONAL	80000	79.294	73.297
RELACIONAL	60000	49.456	46.477
RELACIONAL	40000	16.437	15.144
RELACIONAL	25350	0.960	0.856
RELACIONAL	20000	0.444	0.420
RELACIONAL	10000	0.278	0.191

Fonte: Elaborado pelo Autor.

É possível notar depois das diversas medições, e a retirada dos maiores e menores valores das medições realizadas, que o modelo de banco de dados relacional só foi capaz de realizar uma medição onde estaria no mesmo *range* de tempo do modelo não relacional, utilizando de 25 350 dados inseridos nesse banco. Deve-se ser também demonstrado como foi feito a busca desses dados no modelo relacional, sendo realizado um *select* onde seria necessário o uso de *inner join* pelo relacionamento das tabelas do modelo relacional.

Figura 12 – Maneira que foi realizado a busca de dados no Modelo Relacional

```

46 select clientes.cpf, clientes.nome, clientes.telefone,
47 clientes.cep, editoras.nome, editoras.telefone,
48 editoras.cnpj, editoras.cep, livros.titulo, livros.autor,
49 livros.genero, carrinho_loja.id, carrinho_loja.livro_id,
50 carrinho_loja.cliente_id, carrinho_loja.cnpj, carrinho_loja.preco,
51 carrinho_loja.quantidade
52 from carrinho_loja
53 inner join clientes on carrinho_loja.cliente_id = clientes.id
54 inner join livros on carrinho_loja.livro_id = livros.id
55 inner join editoras on editoras.id = livros.editora_id;

```

Fonte: Elaborado pelo Autor.

Após essa descoberta de desempenho do modelo relacional, gerou-se a pergunta, “Se o modelo relacional levou esse tempo, quanto o modelo não relacional levaria?”, com isso, foi desenvolvido um segundo experimento, o qual teria o propósito de medir o desempenho do não relacional em uma bateria de testes e medições, as quais determinariam sua performance.

Figura 13 – Medições Realizadas no Banco de dados de Modelo Não Relacional

SEGUNDO EXPERIMENTO			
Tipo BD	Quantidade de Dados	Maior tempo (seg.)	Menor Tempo (seg.)
NÃO RELACIONAL	100000	0.976	0.833
RELACIONAL	100000	132.372	113.147
NÃO RELACIONAL	200000	1.743	2.078
NÃO RELACIONAL	500000	4.531	5.268
NÃO RELACIONAL	1000000	10.143	9.248
NÃO RELACIONAL	2000000	20.043	17.999
NÃO RELACIONAL	13000000	122.421	116.943

Fonte: Elaborado pelo Autor.

Após a bateria de testes realizada no modelo não relacional, o mesmo proporcionou um resultado um tanto curioso, o mesmo veio a ter o mesmo desempenho de tempo de resposta do modelo relacional com 100 000 apenas quando realizado a medição com o valor de 13 000 000 de dados, suportando trabalhar de maneira muito rápida com uma quantidade tão grande de dados.

7. CONCLUSÃO

Este trabalho permitiu ser colocado a prova duas tecnologias de banco de dados MySQL e MongoDB, cada uma delas passando por baterias de testes em que se visava o seu tempo de busca, as tecnologias reagiram de maneiras muito diferentes, sem sombra de dúvidas é possível destacar o desempenho demonstrado pelo MongoDB com a grande massa de dados na realização de suas buscas, com seu tempo de resposta incrivelmente rápido.

O MySQL é um ótimo sistema de banco de dados, permitindo realizar as relações de suas tabelas, porém, não é construído para lidar bem com uma massa tão grande de dados, como o MongoDB, caso seja necessário efetuar uma busca em grande escala seu tempo de resposta pode ser um fator em que se deva levar em consideração, dependendo do cenário o qual será implementado.

Desta forma, o trabalho forneceu um rico conhecimento sobre o funcionamento dos dois bancos de dados, proporcionando a possibilidade onde pode ser colocado como um fator decisivo em uma situação de dúvidas sobre a adesão de determinada tecnologia de banco de dados, onde o mesmo pode utilizar destas medições realizadas neste projeto como exemplo. Para assim poder tomar a decisão que lhe forneça o melhor desempenho para a sua necessidade.

REFERÊNCIAS BIBLIOGRÁFICAS

DA SILVA, D. M. **Phyton: História e Ascendência**. Revista portuguesa de programação - Programar. 59. ed. p. 96 – 99, 2018. Disponível em: <<https://www.revista-programar.info/static/downloads/download.php?t=site&e=59>>. Acesso em 27 de Outubro de 2020.

ELMASRI, R.; NAVATHC, S. B. **Sistemas de banco de dados**. 6. ed. - São Paulo: Pcarson Addison Wcsley, 2011.

FARAGLIA, D. Faker's documentation. 2014. Disponível em: <<https://faker.readthedocs.io/en/master/>>. Acessado em 27 de Setembro de 2020.

GARCIA, V. S.; SOTTO, E. C. S. **COMPARATIVO ENTRE OS MODELOS DE BANCO DE DADOS RELACIONAL E NÃO-RELACIONAL**. Revista Interface Tecnológica, [S. l.], v. 16, n. 2, p. 12-24, 2019. DOI: 10.31510/infa.v16i2.673. Disponível em: <<https://revista.fatectq.edu.br/index.php/interfacetecnologica/article/view/673>>. Acessado em 19 de Outubro de 2020.

HEUSER, C. A. **Projeto de Banco de Dados**. 4. ed. – Digital Source & Viciados em Livros. 1998.

LÓSCIO, B. F. **NoSQL no desenvolvimento de aplicações Web colaborativas**. Disponível em: <https://www.researchgate.net/profile/Bernadette_Loscio/publication/268201466_NoSQL_no_desenvolvimento_de_aplicacoes_Web_colaborativas/links/576aa72008aef2a864d1ef8c/NoSQL-no-desenvolvimento-de-aplicacoes-Web-colaborativas.pdf>. Acessado em 21 de Outubro de 2020.

Python 3.9.0 documentation. Python Software Foundation, 2020. Disponível em: <<https://docs.python.org/3/>>. Acessado em 20 de Setembro de 2020.

QUESADA, A. et al. **Apostila de Mysql**. Cge coordenadoria do governo eletrônico. São Paulo, 2001.

SOLAGNA, E. A.; LAZZARETTI, A. T. **Um estudo comparativo entre O mongodb e o postgresql**. Trabalho de Conclusão de Curso (Graduação em Tecnólogo em Sistemas para Internet). Curso de Tecnologia em Sistemas para Internet. Instituto Federal Sul-Rio-Grandense, Campus Passo Fundo, p. 23 2016. Disponível em:

<<https://painel.passofundo.ifsul.edu.br/uploads/arq/201607111805501015914198.pdf>>. Acesso em 26 de Outubro de 2020.

TOTH, R. M. **Abordagem NoSQL – uma real alternativa**. Sorocaba, São Paulo, Brasil. 2011.