



**INSTITUTO FEDERAL**  
Bahia  
Campus Vitória da Conquista



Quézia Menezes Filadelfo  
Efânio Jeferson Amaral de Oliveira

Clécio Silva Ferreira  
Pablo Freire Matos

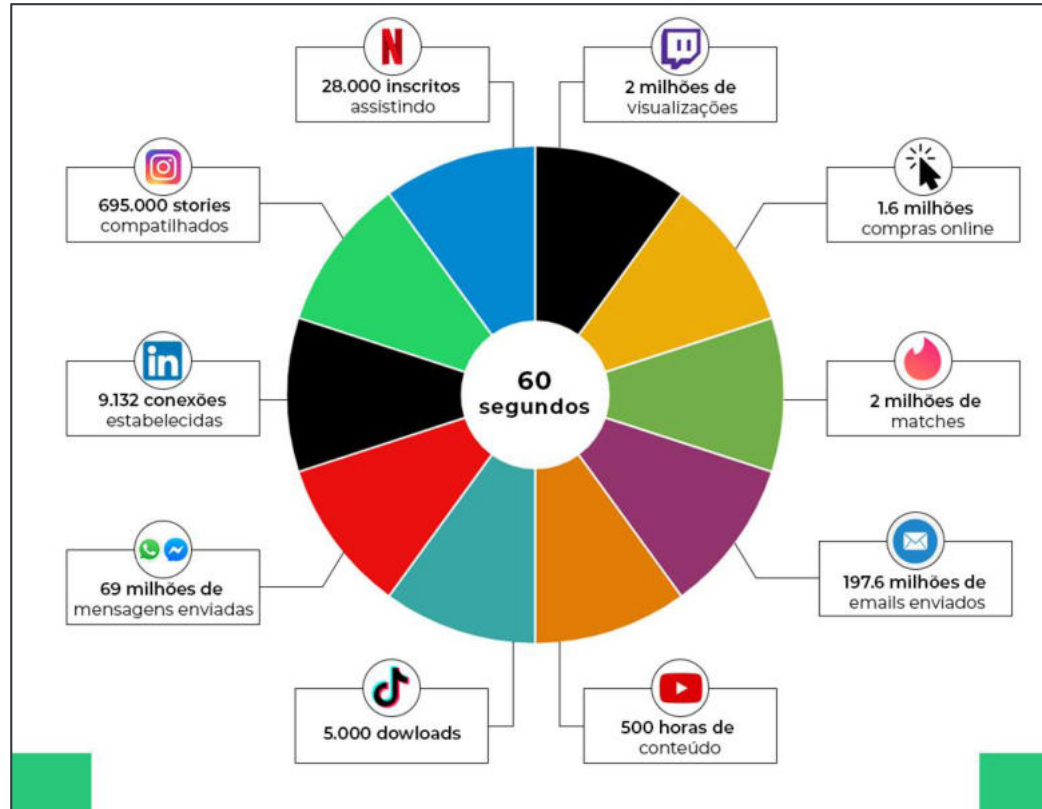
---

# Análise Comparativa de Desempenho entre Bases de Dados Relacional e Não Relacional Orientada a Documentos

# AGENDA

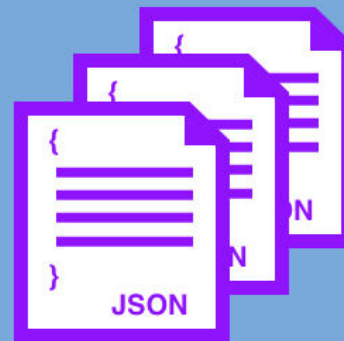
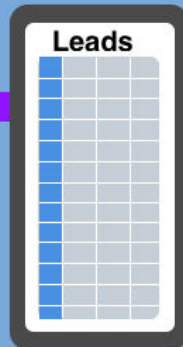
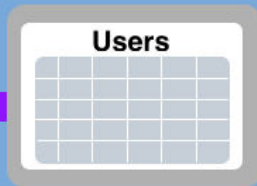
1. Introdução
2. Material e Métodos
3. Resultados e Discussão
4. Conclusão

## O QUE ACONTECE A CADA MINUTOS NA INTERNET



Fonte: Jenik (2021).

# SQL vs. NoSQL



Fonte: Mete (2020)

## ESTUDO DE CASO

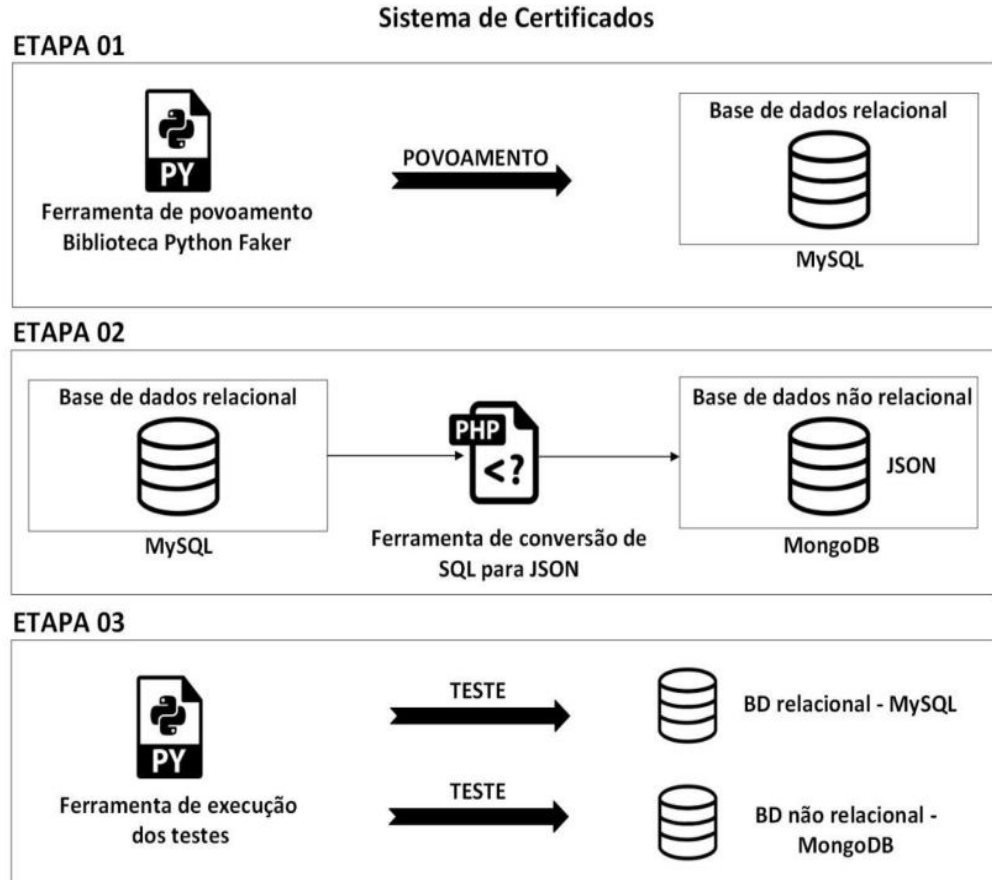


The image shows a web form for searching certificates. At the top, there is a logo consisting of a rolled-up document with a red ribbon seal, followed by the text 'certificados' in green and 'sistema de emissão e validação' in black. Below the logo are four input fields: a text box for 'Informe seu CPF', a text box for 'Informe sua data de nascimento', a dropdown menu for 'Selecione o evento', and a reCAPTCHA verification box with a green checkmark and the text 'Não sou um robô'. At the bottom of the form is a large green button with the text 'Buscar certificados'.

<http://certificados.conquista.ifba.edu.br/>

Fonte: Ramos *et al.* (2018)

## ETAPAS REALIZADAS



## AMBIENTE DE TESTES

### HOST:

- Dell G15 5511
- Processador: Intel Core i5-11400H (clock de 4,50ghz)
- 8GB de memória RAM (frequência de 3200mhz)
- Nvidia RTX 3050 (GDDR, 4GB)

### Sistema operacional:

- Windows 11 Home Edition

## TECNOLOGIAS

### MODELO RELACIONAL:

- MySQL Community Server
- Versão 8.0.28

### MODELO NÃO RELACIONAL:

- MongoDB Community Server
- Versão 5.0.6

## FERRAMENTA DE POVOAMENTO ALEATÓRIO

```
28 def insert():
29
30     cursor, conn = conecta()
31
32     insert_db = []
33     for i in range(100_000):
34         nome = f'{fake.first_name()} {fake.last_name()} {fake.last_name()}'
35
36         insert_db.append((
37             generateCpf(i),
38             nome,
39             '2017-12-23',
40             'random-email@hotmail.com',
41             None,
42             'Sim',
43             '2018-02-04 20:43:02'))
44
45     query = """ INSERT INTO participante (
46         CPF, nome_completo, data_nascimento, email, email_valido, instituicao_ifba_vca, cadastrado_em)
47     VALUES (%s,%s,%s,%s,%s,%s,%s)
48     """
49
50     cursor.execute('SET foreign_key_checks = 0')
51     try:
52         cursor.executemany(query, insert_db)
53         conn.commit()
54         print(cursor.rowcount, "registos inseridos.")
55
56     except Exception as e:
57         print(e)
58
```

<https://github.com/certificadosifba/Ferramenta-Povoamento-Aleatorio>



## FERRAMENTA DE MIGRAÇÃO SQL - JSON

```
1 <?php
2 include 'connection.php';
3
4 $sql = "SELECT id FROM participante limit 1";
5
6 $resultado = mysqli_query($connect, $sql);
7
8 while ($aux = mysqli_fetch_ASsoc($resultado)) {
9     $participante_ids[] = $aux;
10 }
11 for ($i = 0; $i < sizeof($participante_ids); $i++) {
12     $query1 = "SELECT * FROM participante WHERE id=" . implode(',', $participante_ids[$i]) . ";";
13     $query2 = "SELECT * FROM participacao WHERE participante_id=" . implode(',', $participante_ids[$i]) . ";";
14     $result1 = mysqli_query($connect, $query1);
15     $result2 = mysqli_query($connect, $query2);
16
17     // enquanto houver resultados no participante, adiciona os dados pessoais em um documento
18     while ($row = mysqli_fetch_ASsoc($result1)) {
19         $data[$i]['dados_pessoais'] = $row;
20         // enquanto houver resultados no participante, adiciona as participacoes em um array
21         while ($row = mysqli_fetch_ASsoc($result2)) {
22             $json_array[] = $row;
23             $data[$i]['participacao'] = $json_array;
24         }
25         $json_array = [];
26     }
27 }
28
29 $fp = fopen('slide.json', 'w');
30 fwrite($fp, json_encode($data, JSON_PRETTY_PRINT));
31 fclose($fp);
32
```

## FERRAMENTA DE EXECUÇÃO DE TESTES

```
30 def teste_velocidade_sql():
31     cursor, conn = conecta_sql()
32     cursor.execute(c1_sql)
33
34
35 def testa_sql():
36     total_sql = []
37
38     for i in range(6):
39         inicio = timeit.default_timer()
40         teste_velocidade_sql()
41         fim = timeit.default_timer()
42         total_sql.append(fim - inicio)
43
44     total_sql.remove(max(total_sql)) and remove(min(total_sql))
45
46     return total_sql
47
48
49 def teste_velocidade_mongo():
50     cliente, database, mycol = conecta_mongo()
51     mydoc = mycol.aggregate(c1_mongo_ref)
52
53
54 def testa_mongo():
55     total_mongo = []
56
57     for i in range(6):
58         inicio = timeit.default_timer()
59         teste_velocidade_mongo()
60         fim = timeit.default_timer()
61         total_mongo.append(fim - inicio)
62
63     total_mongo.remove(max(total_mongo)) and remove(min(total_mongo))
64
65     return total_mongo
```

## BANCO DE DADOS ORIGINAL (5 mil)

Consulta	MySQL	Mongo DBRef	Mongo Embedded
1	0,0026	0,0340	0,0282
2	0,0105	0,1229	0,0234
3	0,0051	0,0673	0,0245
4	0,0048	0,1210	0,0364
5	0,0022	0,0142	0,0103

## 10 e 100 MIL REGISTROS

Consulta	10.000			100.000		
	MySQL	Mongo DBRef	Mongo Embedded	MySQL	Mongo DBRef	Mongo Embedded
1	0,0029	0,0279	0,1630	0,0030	0,0258	1,2890
2	0,1202	0,8387	0,1106	8,7962	8,2877	1,6289
3	0,0031	0,0809	0,1970	0,0039	0,0769	1,5416
4	0,0142	0,8338	0,1477	0,7227	5,8717	1,1630
5	0,0026	0,0277	0,0428	0,0027	0,1061	0,0922

## 500 MIL e 5 MILHÕES DE REGISTROS

Consulta	500.000			5.000.000		
	MySQL	Mongo DBRef	Mongo Embbeed	MySQL	Mongo DBRef	Mongo Embedded
1	0,0035	0,0255	4,0485	0,0024	0,0266	15,8320
2	13,392	28,7846	4,6558	50,596	63,9515	17,6758
3	0,0037	0,0899	4,9523	0,0024	0,0853	14,0339
4	0,7081	35,662	3,6187	8,2209	56,2658	11,6542
5	0,0031	0,3944	0,3221	0,0019	0,8512	1,9476

## CONCLUSÃO

Tendo como objetivo realizar uma comparação entre duas bases de dados, sendo uma relacional e outra não, se conclui que o estudo realizado nesse artigo é útil para determinar qual base de dados utilizar na construção de determinados sistemas, levando em conta a demanda do sistema e suas restrições.

## CONTRIBUIÇÕES

- Ferramenta de povoamento aleatório
- Ferramenta de migração SQL – JSON
- Ferramenta de execução de testes
- Bases de dados geradas
- Resultados

## TRABALHOS FUTUROS

- Uso de indexação
- Quantidade maior de dados
- Testes com outras operações DML
- Desenvolvimento de uma ferramenta mais simples de migração

# REFERÊNCIAS CITADAS NA APRESENTAÇÃO

JENIK, Claire. **A Minute on the Internet in 2021**. Statista.[S. l.], 20 jul. 2021. Disponível em: <https://www.statista.com/chart/25443/estimated-amount-of-data-created-on-the-internet-in-one-minute/>. Acesso em: 22 jul. 2022.

METE, Frederico. **Relational model vs Document model**. 13 de Jan 2020. Disponível em: <https://medium.com/analytics-vidhya/relational-model-vs-document-model-a71cddd6782d>. Acesso em: 22 Set 2022.

RAMOS, Leandro Lopes; SILVA, Joabe Pinheiro da; SOBREIRA, André Daniel; MATOS, Pablo Freire. Sistema Web e Open Source de Gerenciamento de Emissão e Validação de Certificado nos Institutos Federais. In: CONGRESSO NORTE NORDESTE DE PESQUISA E INOVAÇÃO (CONNEPI), XII, 2018, Recife, PE. **Anais [...]**, 2018. p. 1-10. Disponível em: <http://pablofmatos.pro.br/papers>. Acesso em: 22 Set. 2022.



Quézia Menezes Filadelfo  
Efânio Jeferson Amaral de Oliveira

Clécio Silva Ferreira  
Pablo Freire Matos



Análise Comparativa de Desempenho entre Bases  
de Dados Relacional e Não Relacional Orientada a  
Documentos

**DÚVIDAS?**