

---

# IN MEMORY DB

Projeto de Ingestão e Armazenamento de Dados do QUIZ com Redis  
e Data Warehouse

João Luiz RM 353504

Anita Luri Hino RM 353829

Fhelippe Minucci RM 353968

Ronei Fronta Belem RM 354181

Carolina da Cruz Silva RM353695

---

---

# Objetivo

Desenvolver um mecanismo de ingestão de dados para um sistema de Quiz, utilizando Redis como banco intermediário e PostgreSQL como Data Warehouse (DW), com foco na estruturação dos dados e geração de indicadores

---

---

# 1. Definição das Estruturas de Dados no Redis

A estrutura de dados no Redis é implementada utilizando o tipo de dado **Hash**, adequado para armazenar registros planos com pares campo-valor.

Cada pergunta é salva com uma chave no formato **question:<question\_id>**, contendo:

- question\_text
- alternativa\_a
- alternativa\_b
- alternativa\_c
- alternativa\_d
- alternativa\_correta
- dificuldade
- assunto

Cada respostas dos usuários com chaves **<usuario>:<question\_id>**

contendo campos:

- alternativa escolhida
  - data/hora da resposta
  - usuario
  - número da tentativa.
-

## 2. Exemplos de Carga de Dados

- No exemplo de carga de dados apresentado, foram inseridas manualmente via Swagger em formato Json.
- Os dados são lidos e cada função (save\_question() e create\_answers()) utiliza o comando HSET do Redis para armazenar os dados no formato de Hash.

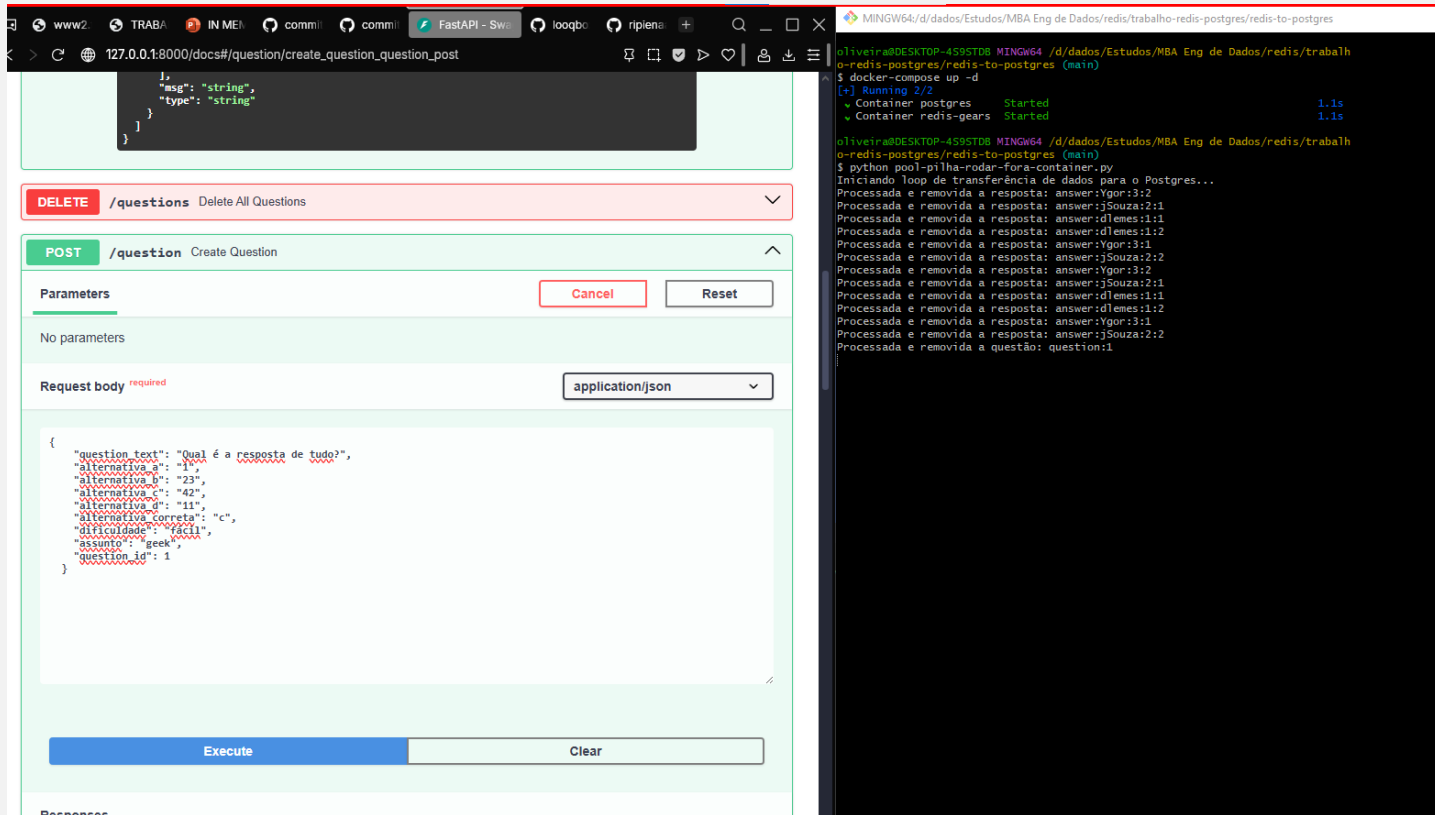
```
{
  "question_text": "Qual é a resposta de nada?",
  "alternativa_a": "1",
  "alternativa_b": "23",
  "alternativa_c": "42",
  "alternativa_d": "11",
  "alternativa_correta": "d",
  "dificuldade": "fácil",
  "assunto": "geek",
  "question_id": 2
}
```

```
1 HSET question:2
2   question_text "Qual é a resposta de nada?"
3   alternativa_a "1"
4   alternativa_b "23"
5   alternativa_c "42"
6   alternativa_d "11"
7   alternativa_correta "d"
8   dificuldade "fácil"
9   assunto "geek"
```

```
{
  "question_id": 1,
  "alternativa_escolhida": "c",
  "datahora": "19/05/2025 09:47",
  "usuario": "dlemes",
  "nro_tentativa": 1
},
```

```
1 HSET answer:dlemes:1:1
2
3   question_id 1
4   alternativa_escolhida "c"
5   datahora "19/05/2025 09:47"
6   usuario "dlemes"
7   nro_tentativa 1
```

### 3. Mecanismo de Ingestão



The image shows a Swagger UI interface on the left and a terminal window on the right. The Swagger UI is for a FastAPI application at 127.0.0.1:8000. It displays the 'POST /question' endpoint, which is used to create a question. The request body is a JSON object with the following structure:

```
{  "question_text": "Qual é a resposta de tudo?",  "alternativa_a": "1",  "alternativa_b": "23",  "alternativa_c": "42",  "alternativa_d": "11",  "alternativa_correta": "c",  "dificuldade": "fácil",  "assunto": "geek",  "question_id": 1}
```

The terminal window on the right shows the execution of a Python script that uses a Redis queue to transfer data to a PostgreSQL database. The script is running in a container named 'redis-to-postgres'.

```
o-redis-postgres/redis-to-postgres (main)
$ docker-compose up -d
[*] Running 2/2
  Container postgres Started 1.1s
  Container redis-gears Started 1.1s

o-redis-postgres/redis-to-postgres (main)
$ python pool-pilha-rodar-fora-container.py
Iniciando loop de transferência de dados para o Postgres...
Processada e removida a resposta: answer:Ygor:3:2
Processada e removida a resposta: answer:jsouza:2:1
Processada e removida a resposta: answer:dleues:1:1
Processada e removida a resposta: answer:dleues:1:2
Processada e removida a resposta: answer:Ygor:3:1
Processada e removida a resposta: answer:jsouza:2:2
Processada e removida a resposta: answer:Ygor:3:2
Processada e removida a resposta: answer:jsouza:2:1
Processada e removida a resposta: answer:dleues:1:1
Processada e removida a resposta: answer:dleues:1:2
Processada e removida a resposta: answer:Ygor:3:1
Processada e removida a resposta: answer:jsouza:2:2
Processada e removida a questão: question:1
```

Utilizamos o Swagger para inserir os dados no Redis.

Assim que entrou no Redis, em até 5 segundos, os dados foram consolidados no PostgreSQL.

Perguntas e respostas foram consolidadas!

### 3. Mecanismo de Ingestão

A-Z question_text	A-Z alternativa	A-Z	A-Z al	A-Z alternativa_d	A-Z	A-Z c	A-Z assur	criado_em	id	A-Z a	datahora	A-Z usuario	criado_em
Qual é o banco em memóri	apache ignite	refis	valkey	memcache	b	fácil	tech	2025-05-30 01:05:35.398	43	2	d	jSouza	2025-05-30 01:15:36.481
Qual é a resposta de nada?	1	23	42	11	d	fácil	geek	2025-05-30 01:05:35.382	44	1	c	dlmes	2025-05-30 01:16:01.512
Qual é a resposta de tudo?	1	23	42	11	c	fácil	geek	2025-05-30 01:05:35.404	45	3	d	Ygor	2025-05-30 01:16:01.519
									46	3	d	Ygor	2025-05-30 01:16:01.523
									47	2	a	jSouza	2025-05-30 01:16:01.528
												dlmes	2025-05-30 01:16:01.533
												jSouza	2025-05-30 01:16:06.538
												dlmes	2025-05-30 01:16:06.546
												Ygor	2025-05-30 01:16:06.551
									52	3	d	Ygor	2025-05-30 01:16:06.555
									53	2	a	jSouza	2025-05-30 01:16:06.559
									54	1	c	dlmes	2025-05-30 01:16:06.563
									55	3	d	Ygor	2025-06-03 23:15:28.288
									56	2	d	jSouza	2025-06-03 23:16:43.418
									57	1	c	dlmes	2025-06-03 23:16:43.425
									58	1	c	dlmes	2025-06-03 23:16:43.432
									59	3	d	Ygor	2025-06-03 23:16:43.439
									60	2	a	jSouza	2025-06-03 23:16:43.449
									61	3	d	Ygor	2025-06-03 23:16:48.458
									62	2	d	jSouza	2025-06-03 23:16:53.471
									63	1	c	dlmes	2025-06-03 23:16:53.475
									64	1	c	dlmes	2025-06-03 23:16:53.480
									65	3	d	Ygor	2025-06-03 23:16:53.485
									66	2	a	jSouza	2025-06-03 23:16:53.489

Aqui está o snapshot dos dados de perguntas e respostas que foram consolidados no PostgreSQL.

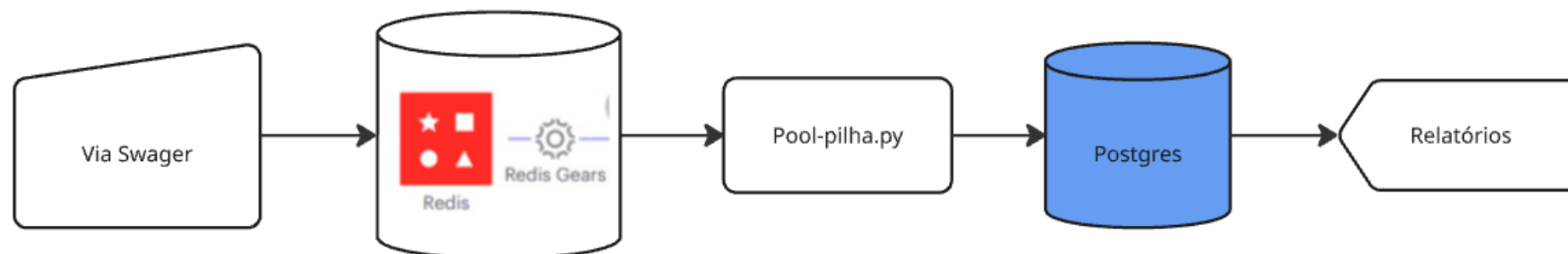
## 4. Modelagem no DW e Consultas

Os dados são inseridos manualmente por meio da interface **Swagger**.

Assim que esses dados são recebidos a aplicação utiliza o comando HSET para armazená-los como Hashes no Redis.

Uma vez que a nova chave é criada no **Redis**, o mecanismo **RedisGears** é acionado automaticamente, funcionando como um observador de eventos.

Esse gatilho notifica o script **pool-pilha-rodar-fora-container.py**, que é responsável por realizar a consolidação dos dados: ele coleta os registros recém-inseridos no Redis, processa e estrutura as informações conforme necessário e, por fim, realiza a inserção consolidada no **PostgreSQL**.



## 4. Modelagem no DW e Consultas

redis_answers	
Columns	
123 id (serial4)	
123 question_id (int4)	
A-Z alternativa_escolhida (varchar)	
datahora (timestamp)	
A-Z usuario (varchar)	
123 nro_tentativa (int4)	
criado_em (timestamp)	

redis_questions	
Columns	
123 id (serial4)	
123 question_id (int4)	
A-Z question_text (text)	
A-Z alternativa_a (text)	
A-Z alternativa_b (text)	
A-Z alternativa_c (text)	
A-Z alternativa_d (text)	
A-Z alternativa_correta (text)	
A-Z dificuldade (text)	
A-Z assunto (text)	
criado_em (timestamp)	

```
SELECT
    q.question_id,
    q.question_text,
    a.alternativa_escolhida,
    COUNT(*) as total_votos,
    ROUND(COUNT(*)::numeric / SUM(COUNT(*) OVER (PARTITION BY q.question_id) * 100, 2) as percentual_votos
FROM redis_questions q
JOIN redis_answers a ON q.question_id = a.question_id
GROUP BY q.question_id, q.question_text, a.alternativa_escolhida
ORDER BY q.question_id, total_votos DESC;
```

123 question_id	A-Z question_text	A-Z alternativa_escolhida	123 total_votos	123 percentual_votos
1	Qual é a resposta de tudo?	c	8	100
2	Qual é a resposta de nada?	a	4	50
2	Qual é a resposta de nada?	d	4	50
3	Qual é o banco em memória	d	8	100

```
SELECT
    q.question_id,
    q.question_text,
    q.dificuldade,
    q.assunto,
    COUNT(a.id) as total_respostas,
    COUNT(CASE WHEN a.alternativa_escolhida = q.alternativa_correta THEN 1 END) as acertos,
    ROUND(COUNT(CASE WHEN a.alternativa_escolhida = q.alternativa_correta THEN 1 END)::numeric / COUNT(a.id) * 100, 2) as taxa_acerto
FROM redis_questions q
LEFT JOIN redis_answers a ON q.question_id = a.question_id
GROUP BY q.question_id, q.question_text, q.dificuldade, q.assunto
HAVING COUNT(a.id) > 0
ORDER BY taxa_acerto DESC;
```

123 qu	A-Z question_text	A-Z dificuldade	A-Z assunto	123 tot	123 acertos	123 taxa_acerto
1	Qual é a resposta de tudo?	fácil	geek	8	8	100
2	Qual é a resposta de nada?	fácil	geek	8	4	50
3	Qual é o banco em memória mais difundido n	fácil	tech	8	0	0



---

## 5. Github

<https://github.com/iloonely/trab-redis-postgres/tree/main>

[apresentacao.mp4](#)

---