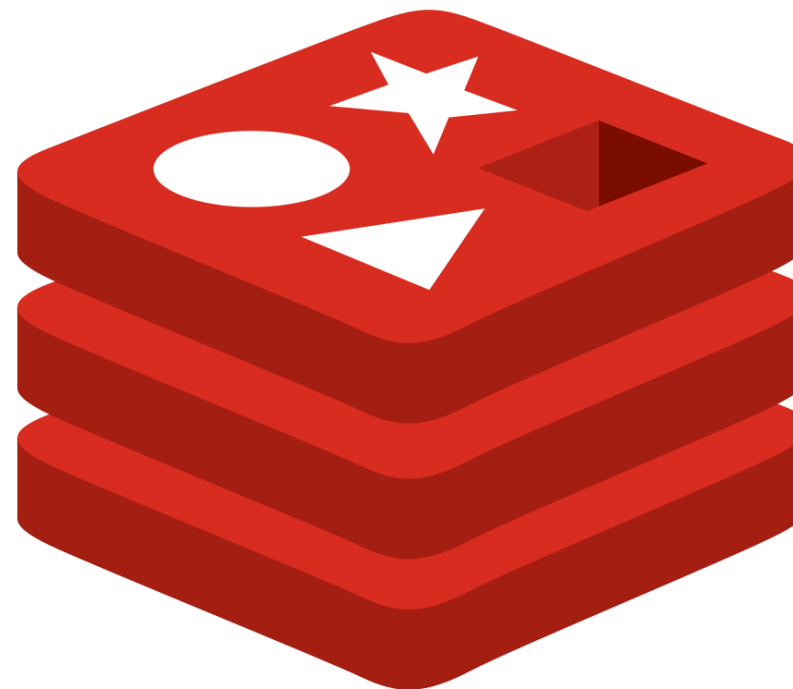


REDIS

Armazenamento de Estrutura
de Dados de Chave-valor na
Memória



INTEGRANTES

Davi Neves	191027383
Giovani Candido	191021601
Luis Morelli	181027097



SUMÁRIO

- Introdução;
- Revisão de Literatura;
- Tutorial;
- Metodologia;
- Considerações Finais.



INTRODUÇÃO

- O que são bancos relacionais?
- Tabela;
- Chave;
- **SQL!**

EMPLOYEE

E#	Name	City
E19	Stewart	Stow
E4	Bell	Kent
E1	Murphy	Kent
E7	Howard	Cleveland

Example query:
"Select the names of the employees living in Kent."

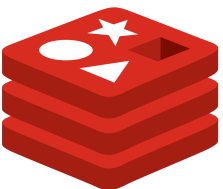
Formulation with SQL:

```
SELECT Name  
FROM EMPLOYEE  
WHERE City = 'Kent'
```

Results table:

Name
Bell
Murphy

Figura 1 – Exemplo de Consulta em SQL



INTRODUÇÃO

- Vantagens do Modelo Relacional:
 - ACID: Atomicidade, Consistência, Isolação e Durabilidade;
 - Facilidade em Projetar, Executar, Manter e Usar;
 - Padronização e Simplicidade.



INTRODUÇÃO

- Desvantagens do Modelo Relacional:
 - Problemas com *Big Data*;
 - Estrutura Predefinida dos Dados;
 - Sobre carga de *Hardware*.



INTRODUÇÃO

- O que são bancos **não** relacionais?
- **NoSQL: Não Apenas SQL!**
- Distribuição e *Big Data*!
- Quatro Tipos:
 - Chave-Valor;
 - Documentos;
 - Colunas;
 - Grafos.

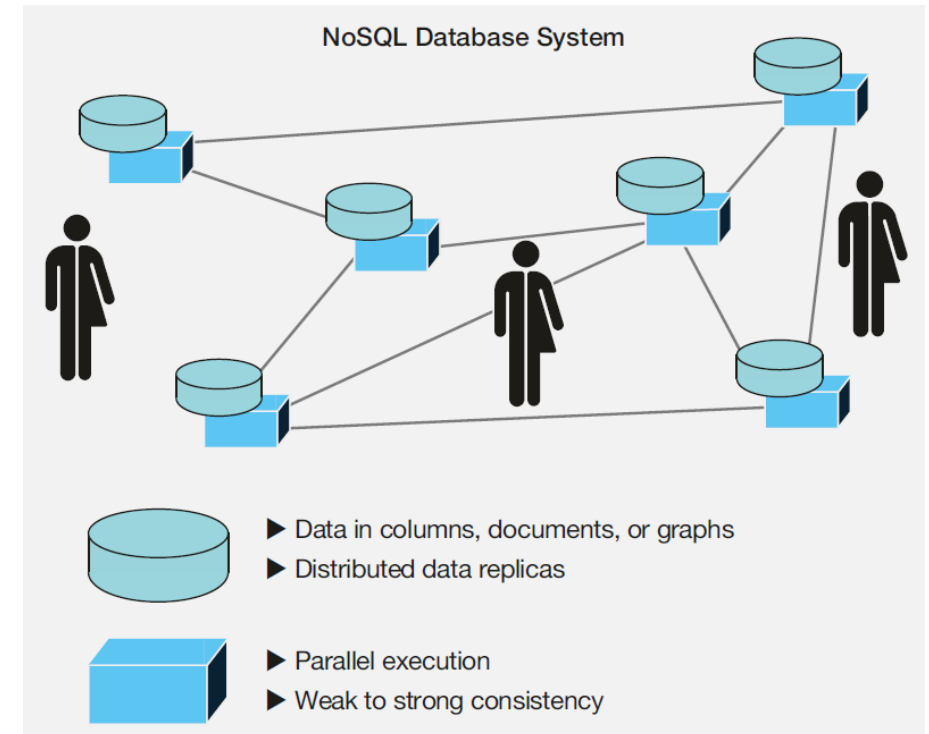
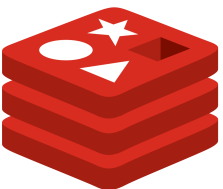


Figura 2 – Estrutura Básica de um NoSQL



INTRODUÇÃO

- Vantagens do Modelo Não Relacional:
 - Alta Performance e Baixa Complexidade;
 - Escala Horizontal: Distribuição de Dados;
 - Sem Restrições ou Modelos de Dados;
 - Ausência de Administradores (DBMA).



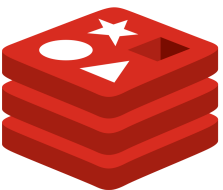
INTRODUÇÃO

- Desvantagens do Modelo Não Relacional:
 - Imaturidade: mais recente que SQL;
 - Dificuldade de Manutenção;
 - Ausência de Interface Específica.



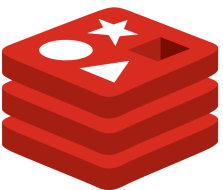
JUSTIFICATIVA

- O presente trabalho se justifica pela diversidade de informações acerca do tema e, desta maneira, **há uma dificuldade** dos desenvolvedores em **filtrar** os livros e artigos que tragam **os dados mais importantes** acerca do Redis.



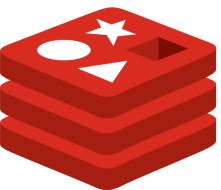
OBJETIVOS

- O objetivo principal deste trabalho, portanto, consiste em **reunir** e **apresentar** as informações mais importantes a respeito do Redis, um banco de dados não relacional. Especificamente, devem ser apresentados **dois exemplos práticos**, a fim de demonstrar a **usabilidade** da ferramenta para as aplicações.



REVISÃO DE LITERATURA

- Explicação do conceito de bancos de dados;
- Modelos relacionais e aos não relacionais;
- Motivo da emergência e os diferentes tipos dos não relacionais;
- Prós e contras do relacional e do não relacional;



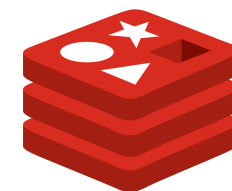
REVISÃO DE LITERATURA

- Banco de dados Redis:
 - Introdução;
 - História;
 - Operação;
 - Casos de uso.



BANCOS DE DADOS

- Coleção coerente de dados relacionados;
- Que possuem um significado inerente;
- Precisam ser armazenados por um propósito específico;



BANCOS DE DADOS

- Dizem respeito a um universo de discurso;
- Como exemplo, temos uma agenda telefônica;
- Organizações precisam lidar com dados.



BANCOS DE DADOS RELACIONAIS

- Composto por relações;
- Cada relação sendo formada por tuplas;
- Onde uma tupla contém um conj. igual de campos;



BANCOS DE DADOS RELACIONAIS

- Cada campo sendo identificado por um nome;
- Com os campos homônimos formando uma coluna;
- Uma relação representa uma entidade ou um relacionamento;



BANCOS DE DADOS RELACIONAIS

- Cada tupla de uma relação tem uma chave;
- A chave é composta por uma combinação única;
- A chave também serve para estabelecer relacionamentos;



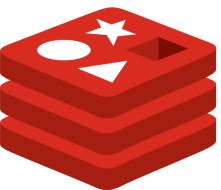
BANCOS DE DADOS RELACIONAIS

- Cada campo tem um domínio ou tipo;
- Os campos devem ser atômicos e mono-valorados;
- SQL: linguagem base para definição e manipulação de dados;
- Cada banco tem suas peculiaridades em relação a SQL;



BANCOS DE DADOS RELACIONAIS

- Bancos relacionais proprietários:
 - Oracle;
 - DB2;
 - SQL Server;
 - SQL Access.
- Alternativas livres:
 - MySQL;
 - PostgreSQL;
 - SQLite.



MODELOS NÃO RELACIONAIS EMERGIRAM

- Incompatibilidade de impedância;
- As aplicações de banco de dados;
- Uso de clusters por grandes empresas.



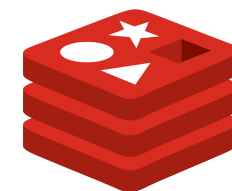
INCOMPATIBILIDADE DE IMPEDÂNCIA

- Diferença entre o modelo e as estruturas de dados dos programas;
- Modelo relacional: relações, tuplas e álgebra linear;
- Não é possível representar estruturas mais complexas;
- Estruturas precisam ser convertidas para o modelo relacional;
- Alguns bancos já haviam tentado superar essa barreira.



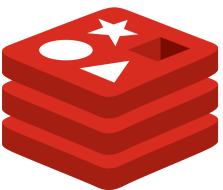
APLICAÇÕES DE BANCO DE DADOS

- Bancos relacionais servem como um mecanismo de integração;
- Algumas base precisam ser acessada por diversas aplicações;
- Os bancos relacionais têm interface conhecida por todos os times;
- Quando o banco é específico para um app, o time tem flexibilidade.



USO DE *CLUSTERS* NA INTERNET

- O modelo relacional não se deu bem com a evolução da internet;
- Havia uma demanda por mais velocidade e capacidade de processamento;
- Bancos relacionais eram licenciados por máquina servidora;
- Os bancos relacionais não foram projetados para *cluster*.



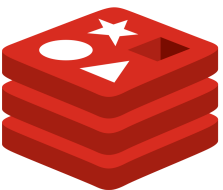
PERSISTÊNCIA POLIGLOTA

- Google criou o BigTable e a Amazon o Dynamo;
- Outras empresas passaram pela mesma dificuldade;
- Muito incentivo para projetos NoSQL de código aberto;
- Múltiplos bancos para múltiplas situações distintas.



BANCOS DE DADOS NÃO RELACIONAIS

- Alternativas flexíveis e gratuitas do modelo relacional;
- Recursos específicos para certos tipos de uso;
- Suporte apropriado aos *clusters*;



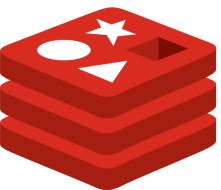
BANCOS DE DADOS NÃO RELACIONAIS

- Tipos de modelos não relacionais (NoSQL):
 - Bancos de dados de chave-valor;
 - Bancos de dados orientados a documentos;
 - Bancos de dados em colunas;
 - Bancos de dados baseados em grafos.



BANCOS DE DADOS DE CHAVE-VALOR

- Estrutura de armazenamento de chave-valor;
- Semelhante a ideia de uma tabela de dispersão;
- O acesso a um valor é feito por uma chave única;
- Como uma relação com apenas dois campos;



BANCOS DE DADOS DE CHAVE-VALOR

- Bancos de chave-valor livres:
 - Riak;
 - Redis;
 - Memcached;
 - Berkeley DB;
 - HamsterDB;
 - Project Voldemort.
- Banco proprietário: Amazon DynamoDB.



BANCOS ORIENTADOS A DOCUMENTOS

- Também empregam uma estrutura chave-valor;
- O valor é um documento XML, JSON, BSON entre outros;
- O projetista é que impõe a estrutura do documento;



BANCOS ORIENTADOS A DOCUMENTOS

- Bancos orientados a documentos:
 - MongoDB;
 - CouchDB;
 - Terrastore;
 - OrientDB;
 - RavenDB.



BANCOS DE DADOS EM COLUNAS

- Emprega uma estrutura chave-valor;
- Cada chave endereça uma família de colunas;
- Trata-se de valores geralmente acessados ao mesmo tempo;



BANCOS DE DADOS EM COLUNAS

- Bancos livres e em colunas:
 - Apache HBase;
 - Hypertable.
- Bancos proprietários:
 - Apache Cassandra;
 - Amazon DynamoDB.



BANCOS BASEADOS EM GRAFOS

- Permite guardar entidades e estabelecer relacionamentos;
- Nó representa uma entidade e seus atributos;
- Cada aresta representa um relacionamento entre duas entidades;
- A aresta pode ter atributos e a direção importa;
- Os dados podem ser facilmente interpretados;



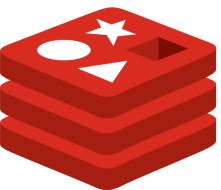
BANCOS BASEADOS EM GRAFOS

- Banco livres e baseado em grafos:
 - OrientDB;
 - FlockDB.
- Bancos proprietários:
 - Infinite Graph;
 - Neo4J.



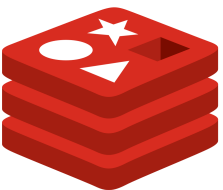
VANTAGENS DOS BANCOS NÃO RELACIONAIS

- Disponibilizam uma série de modelos de dados distintos;
- Facilmente escaláveis;
- Não requerem um administrador de banco de dados;



VANTAGENS DOS BANCOS NÃO RELACIONAIS

- Lidam com falhas de hardware automaticamente;
- Flexíveis em relação a tipos de dados;
- Eficientes e rápidos.



DESVANTAGENS DOS BANCOS NÃO RELACIONAIS

- Falta de maturidade;
- Falta de uma linguagem padrão para definição e consulta dos dados;
- Não tem conformidade com as propriedades ACID;
- Dificuldade de manutenção.



VANTAGENS DOS BANCOS RELACIONAIS

- Simples de empregar, projetar, operar e manter;
- Interfaces comuns aos desenvolvedores;
- Garantem integridade dos dados armazenados;
- Linguagem padrão de consulta de dados.



DESVANTAGENS DOS BANCOS RELACIONAIS

- Alto custo de licenciamento;
- Sobrecarga do hardware;
- Limitações estruturais das relações;
- Dados perdidos são dificilmente recuperados.



BANCO DE DADOS REDIS

- **Remote Dictionary Server**;
- Armazenamento de Estrutura de Dados na Memória;
- Clientes para mais de 30 linguagens de programação;
- Código fonte disponível no [GitHub](#);
- Documentação oficial também [disponível](#).

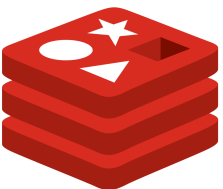


HISTÓRIA

- Salvatore Sanfillipo, em 2009;
- Otimizar performance de seu produto, LLOOGG;
- Que relizava Web Analytics em tempo real;



Figura 3 – Salvatore Sanfillipo



HISTÓRIA

- Tratamento de grandes quantias de dados;
- Necessitava de velocidade e eficiência;
- Projeto evoluiu e ganhou uma comunidade;
- Houve a criação da Redis Labs.

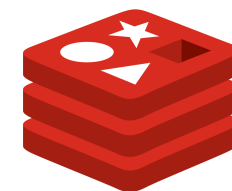


Figura 3 – Salvatore Sanfilippo



OPERAÇÃO

- Arquitetura Cliente-Servidor;
- Lógica Chave-valor;
- Armazenamento na memória;
- Mecanismo semelhante a paginação;
- Persistência e Replicação de Dados;
- Comandos de Manipulação e Consulta de Dados.



CASOS DE USO

- Armazenamento Cache;
- Chat, sistemas de mensagens e filas;
- Placares de Jogos;
- Streaming de Mídia;
- Aprendizado de Máquina;
- Análises em Tempo Real;
- Dados Geoespaciais.



TUTORIAL: USABILIDADE DO REDIS

- Guia de Instalação e Configuração do Redis;
- Exemplo Prático #1: Linha de Comando;
- Exemplo Prático #2: Aplicação em Python;

Todo o processo é demonstrado em vídeo!



METODOLOGIA

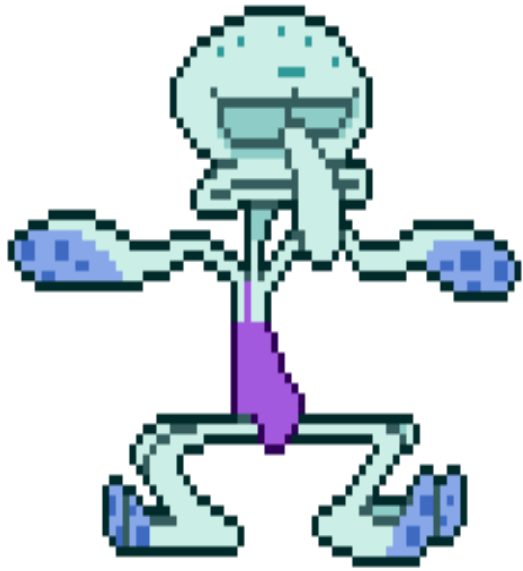
- Revisão bibliográfica;
- Levantamento de informações;
- Coleta de Dados:
 - Artigos Científicos;
 - Documentação Oficial;
 - Livros sobre Redis.
- Análise dos Dados:
 - Filtragem;
 - Concatenação do conteúdo.



CONSIDERAÇÕES FINAIS

- Objetivos: **reunir** e **apresentar** as informações mais relevantes sobre o Redis;
- Metodologia:
 - Procura de artigos e livros extraoficiais;
 - Filtragem das informações;
 - Criação e apresentação de tutorial.
- Informações relevantes trazidas de maneira **direta**: cumprimento com os objetivos!
- Espera-se surgimento de mais desenvolvedores na área que utilizem o Redis!





Sobrevivemos até aqui e
é um misto de emoções.
Obrigado, pessoal!

