



redis

REDIS

ARMAZENAMENTO CHAVES E VALOR

BANCO DE DADOS AVANÇADOS

VALÉRIA TIMES

Redis

- ▶ Utilizado
 - ▶ Session Cache:
 - ▶ Lojas online para carrinho de vendas
 - ▶ Cache de Página
 - ▶ Diversos plugin para lojas virtuais
 - ▶ Wordpress
 - ▶ Majento
 - ▶ Filas
 - ▶ Análise em tempo real de eventos
 - ▶ Estatísticas (Contagem/Scores)
 - ▶ Exemplos de projetos que utilizam redis
 - ▶ <https://redislabs.com/solutions>

Redis

- ▶ Basicamente um Dicionário chave-valor.
 - ▶ C#: **Dictionary<key,value>**
 - ▶ Java: **HashMap<key,value>**
- ▶ O Redis armazena em memória RAM os dados, e de tempos em tempos transfere o conteúdo da memória para o disco.

Redis

- ▶ Armazena dados simples
 - ▶ Strings
 - ▶ Números
- ▶ Dados complexos
 - ▶ Lists
 - ▶ Sets
 - ▶ Hashes
 - ▶ Sorted Sets

Redis

- ▶ Porque usar?
 - ▶ Insanamente rápido
 - ▶ Leve e pequeno
 - ▶ Dados armazenados em memória/disco
 - ▶ É OpenSource

Redis

- ▶ Vantagens
 - ▶ Acesso aos dados é realizado de forma ágil, sem acesso ao disco.
- ▶ Desvantagens
 - ▶ Não permite JOINS do SQL
 - ▶ Necessário ter o mesmo espaço de memória do mesmo tamanho do dado que deseja armazenar

Redis

- ▶ Tipos de dados
 - ▶ Strings
 - ▶ Integers
 - ▶ Hash
 - ▶ Lists
 - ▶ Sets
 - ▶ Sorted Sets

Redis

- ▶ Exemplo:
 - ▶ redis> set disciplina “Banco de Dados Avançados”
 - ▶ redis> get disciplina
 - ▶ “Banco de Dados Avançados”

Redis

- ▶ Operações
 - ▶ get/set
 - ▶ push/pop
 - ▶ add/remove
 - ▶ union/intersect/diffs
 - ▶ e muito mais, incluindo opções de ordenação.

Redis

- ▶ Download
 - ▶ <https://github.com/MSOpenTech/redis/releases>
 - ▶ <http://redis.io/commands>
- ▶ Avançar... Avançar...
 - ▶ Abrir redis-cli.exe
 - ▶ Ou
 - ▶ CMD = redis-cli

Redis

- ▶ Trabalhando com strings e inteiros
 - ▶ **set key** = define um valor a uma chave
 - ▶ **get key** = retorna o valor de uma chave
 - ▶ **append key** = concatena um novo valor ao valor atual de uma chave e altera o valor atual da chave
 - ▶ **mget key [key...]** = retorna o valor de várias chaves
 - ▶ **substr key args args** = retorna um intervalo do valor de uma chave
 - ▶ **Incr key** = incrementa valor 1 a uma chave do tipo inteiro
 - ▶ **incrby num arg** = incrementa valor especificado a uma chave do tipo inteiro
 - ▶ **decr num** = decrementa valor 1 a uma chave do tipo inteiro

Redis

- ▶ Trabalhando com strings e inteiros
 - ▶ set nome "Urbano"
 - ▶ get nome
 - ▶ "Urbano"
 - ▶ set disciplina "Banco de Dados "
 - ▶ append disciplina "Avançados"
 - ▶ mget nome disciplina
 - ▶ 1. "Urbano"
 - ▶ 2. "Banco de Dados Avançados"
 - ▶ substr nome 2 6
 - ▶ "bano"

Redis

- ▶ Trabalhando com strings e inteiros
 - ▶ set num 1
 - ▶ incr num
 - ▶ (integer) 2
 - ▶ incrby num 8
 - ▶ (integer) 10
 - ▶ decr num
 - ▶ (integer) 9
 - ▶ get num
 - ▶ "9"

Redis

- ▶ Trabalhando com Listas
 - ▶ **rpush key** = insere um item a direita da lista (final)
 - ▶ **lpush key** = insere um item a esquerda da lista (início)
 - ▶ **lrange key arg arg** = retorna parte da lista definido pelo intervalo
 - ▶ **lpop key** = retorna um elemento a esquerda da lista e o remove
 - ▶ **rpop key** = retorna um elemento a direita da lista e o remove

Redis

- ▶ Trabalhando com Listas
 - ▶ rpush lista 1
 - ▶ rpush lista 2
 - ▶ lpush lista 0
 - ▶ lrange lista 0 2
 - ▶ 1) "0"
 - ▶ 2) "1"
 - ▶ 3) "3"
 - ▶ lrange lista 0 1
 - ▶ 1) "0"
 - ▶ 2) "1"

Redis

- ▶ Trabalhando com Listas
 - ▶ lpop lista
 - ▶ “0”
 - ▶ lrange lista 0 -1
 - ▶ 1) “1”
 - ▶ 2) “2”
 - ▶ rpop lista
 - ▶ “2”
 - ▶ lrange lista 0 -1
 - ▶ 1) “1”

Redis

- ▶ Trabalhando com SET - Coleções de String – Não Ordenadas
 - ▶ **sadd key** = adiciona um item a uma coleção
 - ▶ **smembers key** = retorna os itens de uma coleção
 - ▶ **sunion key [key...]** = realiza a união de duas ou mais coleções
 - ▶ **sinter key [key...]** = retorna intercessão entre coleções
 - ▶ **sdiff key [key...]** = retorna a diferença entre coleções
 - ▶ **srem key arg** = remove um item específico de uma coleção

Redis

- ▶ Trabalhando com SET - Coleções de String – Não Ordenadas
 - ▶ sadd usuarios "amanda"
 - ▶ sadd usuarios "francisco"
 - ▶ sadd usuarios "fernando"
 - ▶ smembers usuarios
 - ▶ 1) "francisco"
 - ▶ 2) "amanda"
 - ▶ 3) "fernando"

Redis

- ▶ Trabalhando com SET - Coleções de String – Não Ordenadas
 - ▶ sadd log “amanda”
 - ▶ sinter usuarios log
 - ▶ 1) “amanda”
 - ▶ sdiff usuarios log
 - ▶ 1) “francisco”
 - ▶ 2) “fernando”
 - ▶ sunion usuarios log
 - ▶ srem usuarios “fernando”
 - ▶ smembers usuarios
 - ▶ 1) “francisco”
 - ▶ 2) “amanda”

Redis

- ▶ Trabalhando com SORTED SET - Coleções de String – Ordenadas
 - ▶ zadd nomes 1 "Fulano"
 - ▶ zadd nomes 2 "Cicrano"
 - ▶ zrange nomes 0 -1
 - ▶ 1) "Fulano"
 - ▶ 2) "Cicrano"
 - ▶ zrevrange nomes 0 -1
 - ▶ 1) "Cicrano"
 - ▶ 2) "Fulano"
 - ▶ zrem nomes 1

Redis

- ▶ Trabalhando com HASH
 - ▶ `hset pessoa nome "Urbano"`
 - ▶ `hmset pessoa email "mus at cin.ufpe.br" senha "123"`
 - ▶ `hkeys pessoa`
 - ▶ 1) "nome"
 - ▶ 2) "email"
 - ▶ 3) "senha"
 - ▶ `hvals pessoa`
 - ▶ 1) "Urbano"
 - ▶ 2) "mus at cin.ufpe.br"
 - ▶ 3) "123"

Redis

- ▶ Trabalhando com HASH
 - ▶ hgetall pessoa
 - ▶ 1) "nome"
 - ▶ 2) "email"
 - ▶ 3) "senha"
 - ▶ 4) "Urbano"
 - ▶ 5) "mus at cin.ufpe.br"
 - ▶ 6) "123"
 - ▶ hdel pessoa senha
 - ▶ hmget pessoa nome email
 - ▶ 1) "Urbano"
 - ▶ 2) "mus at cin.ufpe.br"

Redis

- ▶ Documentação
 - ▶ <http://redis.io/documentation>
- ▶ Comandos
 - ▶ <http://redis.io/commands>