

# 5 Projeto

## ■ Entrega 2

# Entrega 2

Introdução

Objetivos e  
Desafios

Da pedra ao  
smartphone

Tipos e  
Arquiteturas

Projeto

Entrega 1

Entrega 2

Em época de Big Data, um banco de dados com apenas um servidor é uma nulidade. Como armazenar meus milhões e milhões de registros? É necessário espalhar os dados por múltiplos servidores, e para tanto usaremos como modelo o funcionamento do Chord.

# Roteamento

Introdução

Objetivos e  
Desafios

Da pedra ao  
smartphone

Tipos e  
Arquiteturas

Projeto

Entrega 1  
Entrega 2

De acordo com a especificação da entrega anterior, cada requisição é colocada em uma fila F1, de onde é re-enfileirada nas filas F2 e F3.

- Para esta entrega, antes de re-enfileirar a mensagem, o servidor deverá analisar se é realmente responsabilidade deste servidor.
- Caso o seja, a requisição é re-enfileirada em F2 e F3. Caso contrário, será colocada em uma fila F4.
- Um thread retira de F4 e invoca, consultando uma tabela de roteamento, o nó responsável pela requisição para que a processe ou que pelo menos esteja mais próximo que o mesmo.
- O servidor primeiro contactado pelo cliente é o responsável por enviar a resposta para o cliente.

# Particionamento

Introdução

Objetivos e  
Desafios

Da pedra ao  
smartphone

Tipos e  
Arquiteturas

Projeto

Entrega 1

Entrega 2

O particionamento da responsabilidade sobre os dados seguirá o esquema de anel lógico definido pelo Chord.

- Seja  $n$  o número de nós a serem colocados no sistema na execução de testes.
- Cada servidor é identificado por um número de  $m$  bits.
- O primeiro nó a entrar no sistema recebe necessariamente o identificador  $2^m - 1$
- O nó seguinte recebe identificador menor  $2^m/n$  que o anterior.
- Seja uma sequência de nós com identificadores  $X < Y < Z$ . O nó  $Y$  é responsável pelos dados com chaves na faixa  $(X, Y]$ .

# Comunicação

Introdução

Objetivos e  
Desafios

Da pedra ao  
smartphone

Tipos e  
Arquiteturas

Projeto

Entrega 1

Entrega 2

- Toda comunicação deve ser agora feita usando gRPC.
- Cada operação é realizada via uma função diferente (i.e., há uma função para C, outra para R, ...).
- Servidores redirecionam requisições também usando gRPC, usando a mesma interface usada por clientes.
- Toda requisição é executada assincronamente do ponto de vista de quem invoca a requisição.
- Uma requisição é redirecionada para o nó seguinte ou anterior, dependendo de qual o caminho mais curto até o nó responsável pelos dados.
- Múltiplos saltos podem ser necessários até que a requisição seja respondida.

# Tratamento de falhas

Introdução

Objetivos e  
Desafios

Da pedra ao  
smartphone

Tipos e  
Arquiteturas

Projeto

Entrega 1

Entrega 2

- Assuma que não haverão falhas permanentes ou envio de requisições enquanto algum nó estiver falho.
- Nós podem ser reiniciados e, como na primeira entrega, devem ter seu estado recuperado pelo uso do log de operações e de snapshot do banco de dados.

# Log + Snapshot

Introdução

Objetivos e  
Desafios

Da pedra ao  
smartphone

Tipos e  
Arquiteturas

Projeto

Entrega 1

Entrega 2

- Para evitar que o log se torne grande demais, frequentemente serão feitos snapshots do estado atual do banco de dados.
- Um snapshot do banco captura o estado atual do mesmo, em arquivo, e portanto torna desnecessário o arquivo de logs contendo as operações anteriores ao snapshot.

# Snapshotting

Introdução

Objetivos e  
Desafios

Da pedra ao  
smartphone

Tipos e  
Arquiteturas

Projeto

Entrega 1

Entrega 2

- A cada  $U$  segundos, o estado atual do banco será gravado em um arquivo nomeado `snap.X`, onde  $X$  é um contador de logs. Isto é, o primeiro snapshot será gravado como `snap.1`, o segundo como `snap.2` e assim por diante.
- As operações executadas antes de um snapshot  $X$  serão gravadas e um arquivo de `log.(X-1)`. O primeiro arquivo de logs será então o `log.0`.
- Uma vez inicializado o snapshot que cria `snap.X`, nenhuma nova operação será escrita em `log.(X-1)`. Novas operações são escritas em `log.X`.
- Serão mantidos pelo sistema os últimos 3 arquivos de log e de snapshot. Isto é, se o último snapshot executado foi o décimo, então há no sistema os logs `log.8`, `log.9`, `log.10` (sendo escrito), e os snapshots `snap.8`, `snap.9`, e `snap.10`.