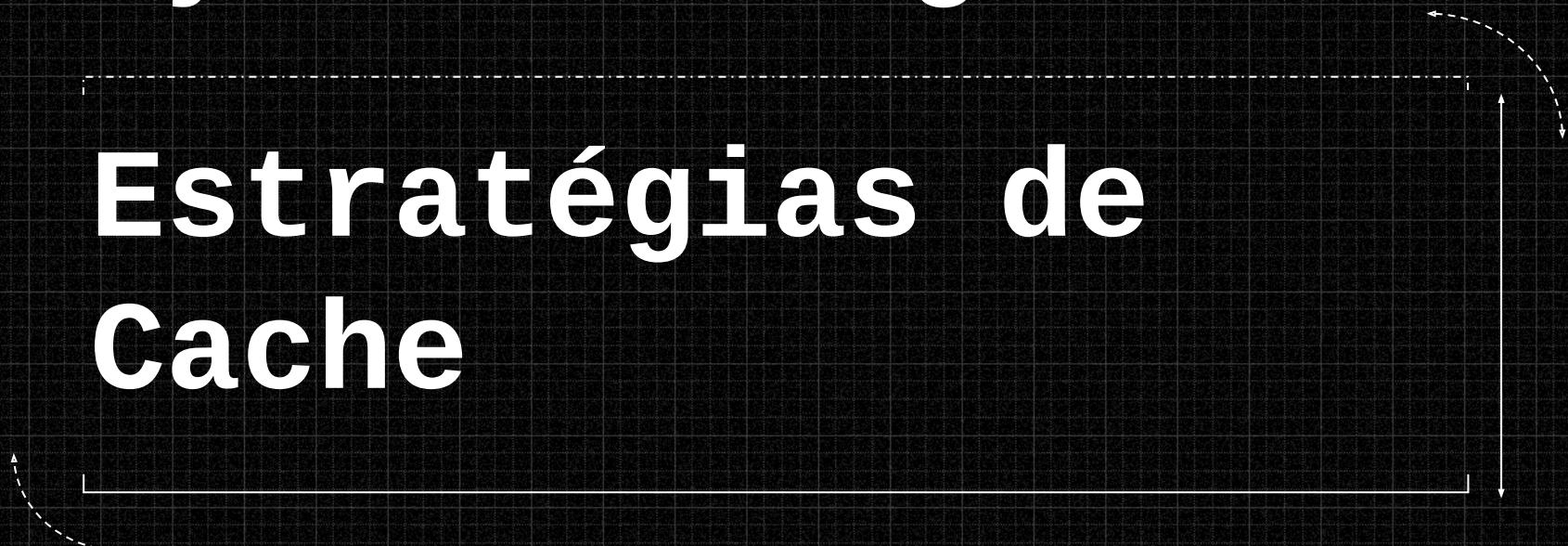


System Design

Estratégias de Cache



\$ whoami

Matheus Fidelis

Engenheiro de \$RANDOM

@fidelissauro

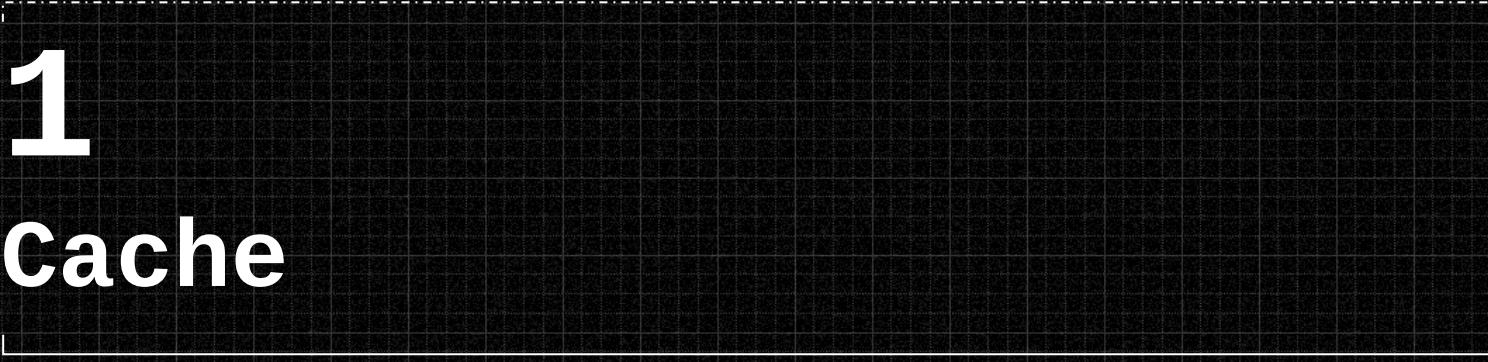
<https://fidelissauro.dev>

<https://linktr.ee/fidelissauro>



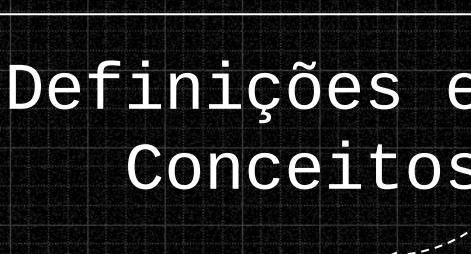
OBJETIVOS

-  Conceitos de cache
-  Cache em sistemas distribuídos
-  Estratégias de uso
-  Desafios e boas práticas



1

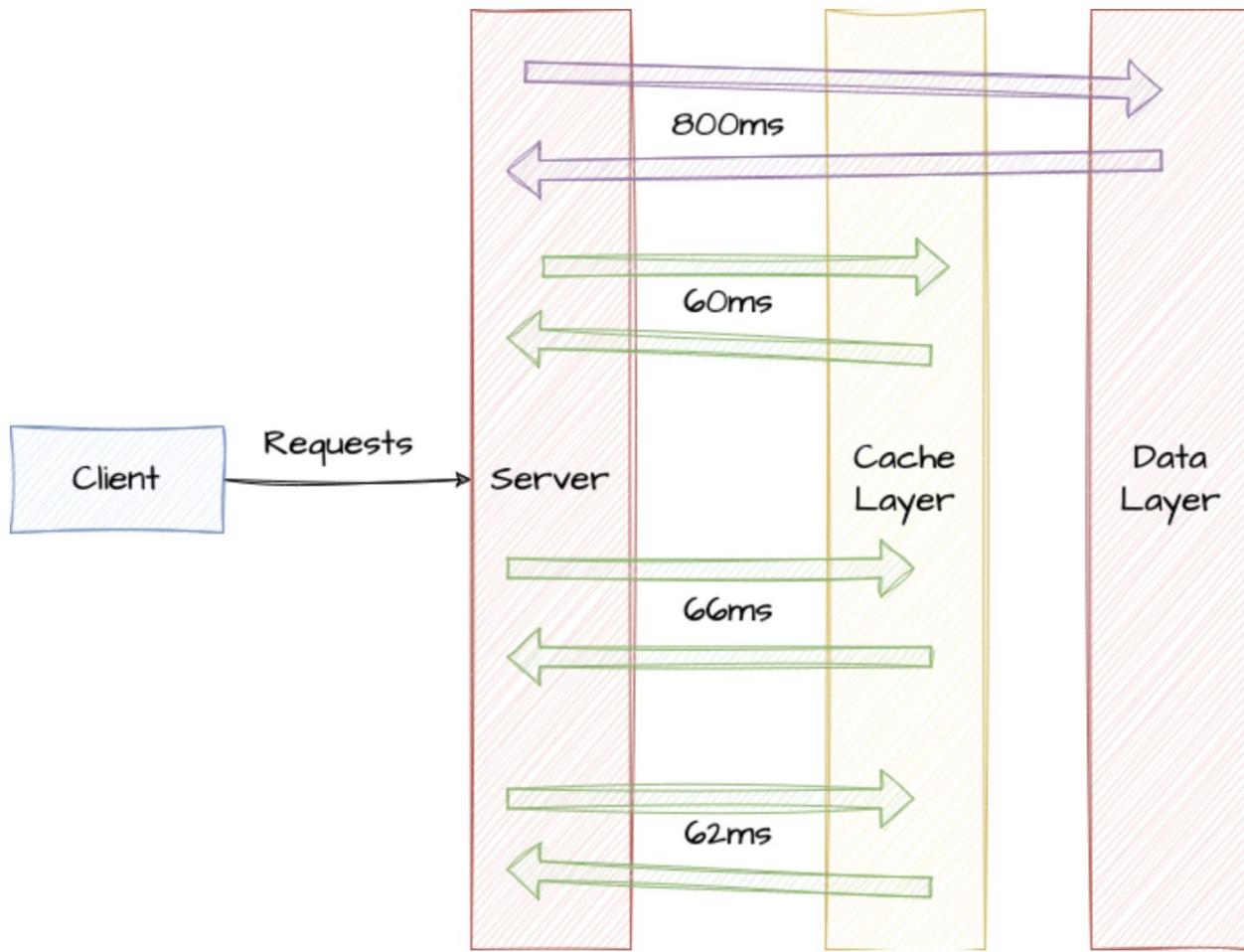
Cache



Definições e
Conceitos

Definindo Cache

- Camada intermediária entre cliente e fonte de dados
- Armazena resultados de operações custosas
- Aproxima os dados do consumidor
- Reduz latência e carga em dependências
- "Camada Barata"



2

Conceitos de Cache

Definições e
Conceitos

Consistência de Dados

- Consistência Fraca
- Consistência Eventual
- Camada Extra da Golden Source
- Garantir alinhamento entre cache e fonte de dados
- Escritas precisam atualizar ou invalidar o cache
- “Quantas vezes os dados cadastrais de um usuário podem mudar?”

Time to Live (TTL)

- Soft State (BASE)
- Define a vida útil de um item em cache
- Evita dados desatualizados circulando no sistema
- Garante reciclagem periódica de informações
- Essencial em sistemas de larga escala

Políticas de Evicção

- Definem quais itens remover quando cache lota
- LRU: remove o menos usado recentemente
- LFU: remove o menos usado frequentemente
- FIFO: remove o mais antigo
- RR: remove aleatoriamente

Invalidez de Cache

- Processo de remover ou marcar itens como inválidos
- Pode ser manual, lógica ou via TTL
- Consistência e atualização de dados
- Um dos maiores desafios de caching

Cache Hit, Cache Miss e Hit Rate

- Métricas de Caching
- Cache Hit: dado já no cache, resposta imediata
- Cache Miss: dado não encontrado, busca na origem
- $\text{Hit Rate} = \text{hits} \div \text{total de requisições}$
- Alta taxa de hits = sistema eficiente
- Baixa taxa de hits = sistema ineficiente



3 Implementações de Cache

Definições e
Conceitos

Implementações de Cache

- Cache em Memória (In-Memory) (Hashmap)
- Cache Distribuído (Redis, Memcached...)
- Cache em Bancos de Dados e Camadas de Dados
- Cache-Aside (Lazy Loading)
- Write-Through (Escrita Dupla)
- Write-Behind (Lazy Writing)
- Cache de Conteúdo Distribuído (CDN Cache)

Cache em Memória (In-Memory)

- Cache em Memória Local
- Excelente performance em acesso
- Estruturas de Dados Comuns
- Chave-Valor (HashMap)
- Escopo limitado a processo/thread
- Requer invalidação manual para evitar leaks

Cache Em Sistemas Distribuído

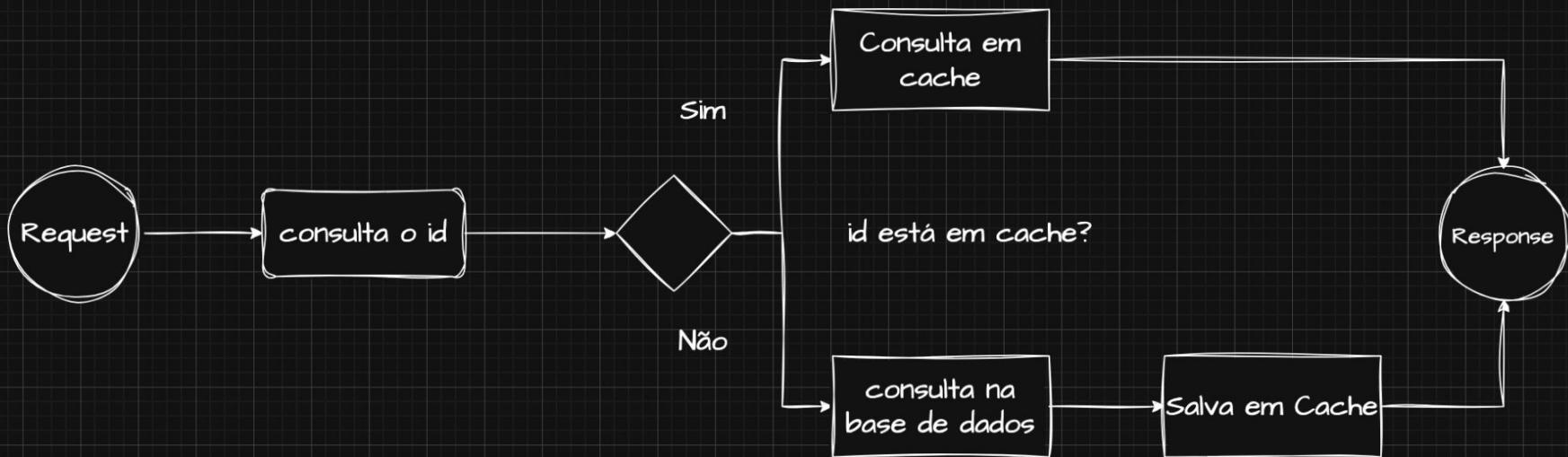
- Performance, redução da latência e escalabilidade
- Servir conteúdo dinâmico ou estático
- Compartilhado entre várias replicas
- Escalabilidade horizontal via clusterização
- Replicação e Alta Disponibilidade
- Suporte a políticas de expiração e evicção nativas
- Tecnologias: Redis, Memcached

Cache em Camadas de Dados

- Camada Adicional do Dado
- Dado mais barato computacionalmente
- Alivio de Gargalos de Componentes Críticos
- Poupar recursos computacionais dos databases
- Cachear itens com pouca modificação
- Armazena resultados de queries e registros quentes

Cache-Aside

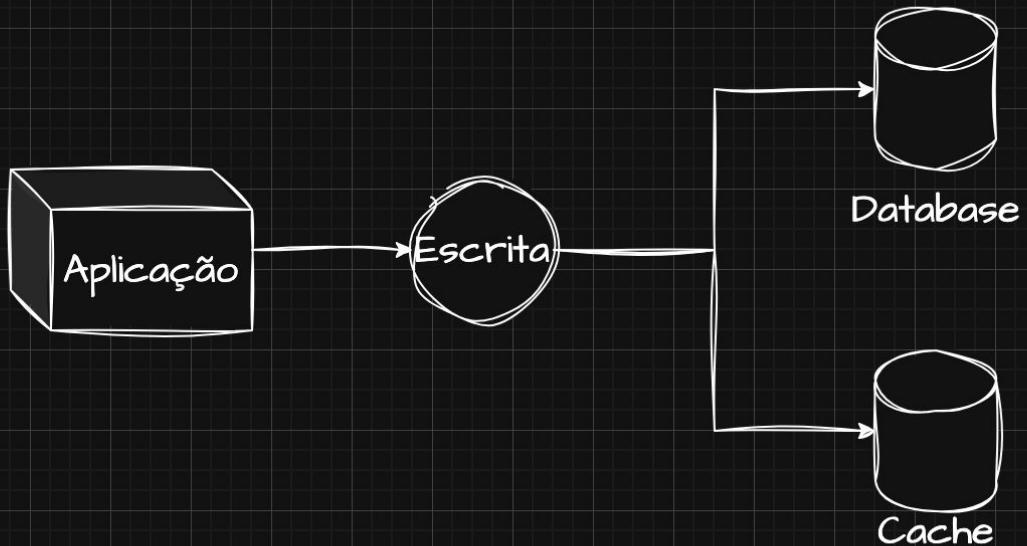
- Lazy Loading
- Só cria o cache quando a aplicação precisa
- Estratégia mais comum em camadas de dados
- Se não encontrar, busca no database
- Popula o Cache e Retorna
- Desafio de Consistência do banco dade Dados



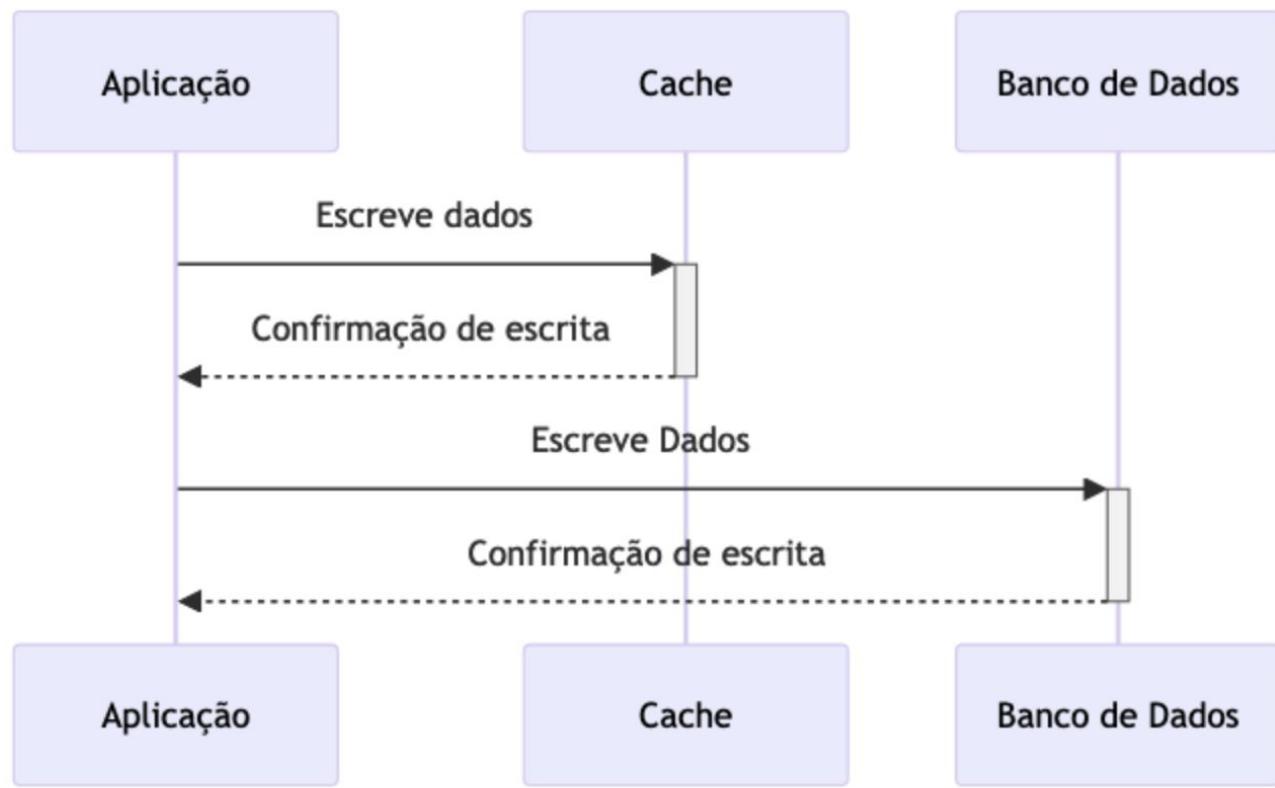
Write-Through (Escrita Dupla)

- Escrita efetuada simultaneamente
- Database e Cache
- Ideal quando as leituras precisam ser rápidas desde o inicio
- Desafios de Consistência
- Combinado com Cache-Aside (Fallback)

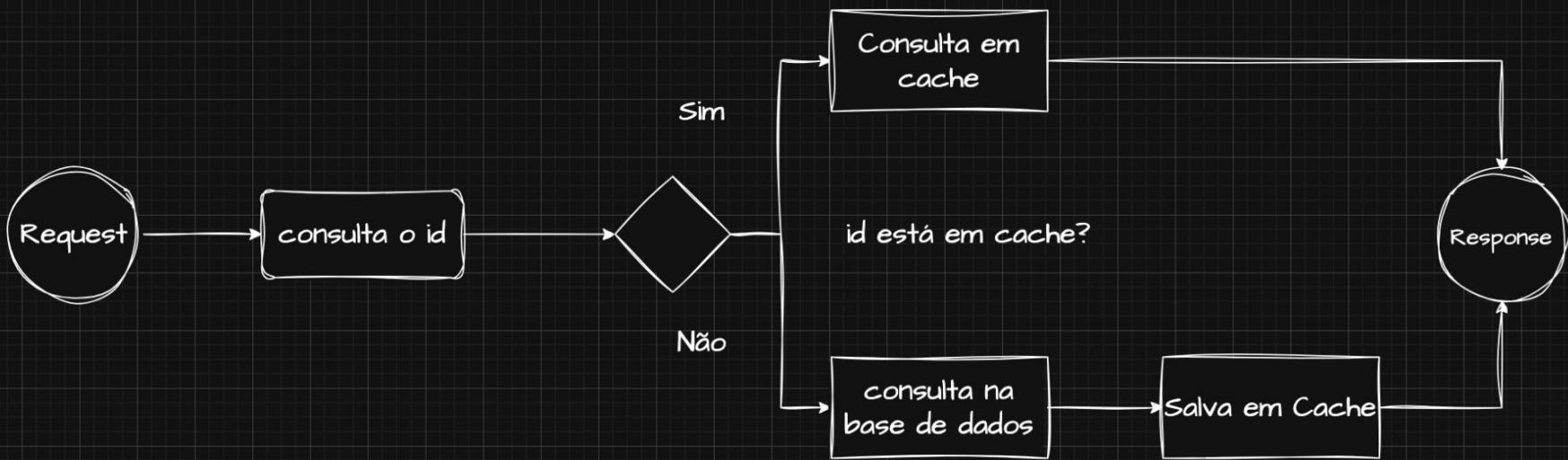
Write-Through (Escrita Dupla)



Write-Through (Escrita Dupla)



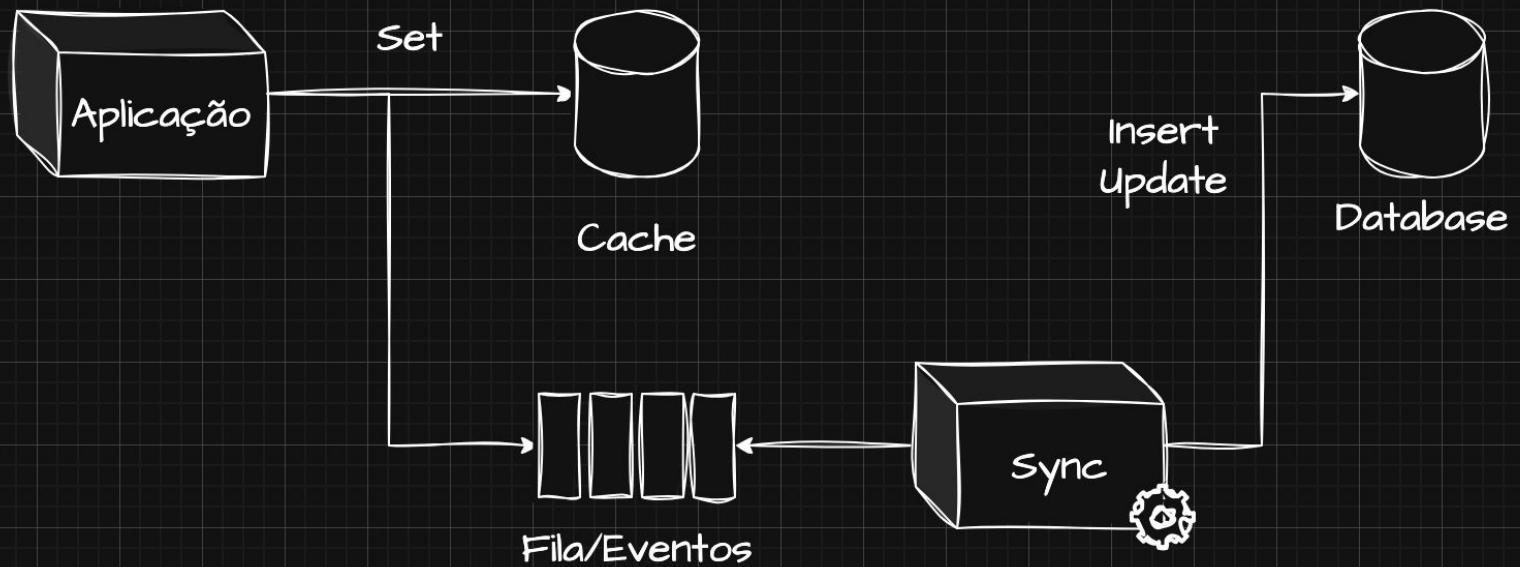
Combinação com Cache-Aside (Recap)



Write-Behind (Lazy Writing)

- Escrita ocorre primeiro no cache
- Prioriza a performance da aplicação
- Atualização do DB é realizada de forma assíncrona
- Baixa latência de escrita para a aplicação
- Risco de perda do dado antes do "flush"
- Usado com Filas, Eventos e serviços intermediários

Write-Behind (Lazy Writing)

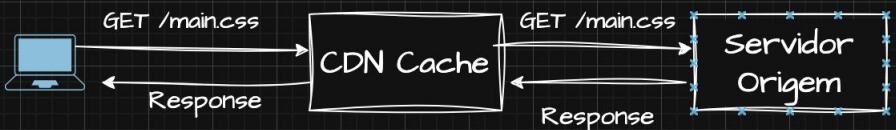


Cache de Conteúdo Distribuído (CDN Cache)

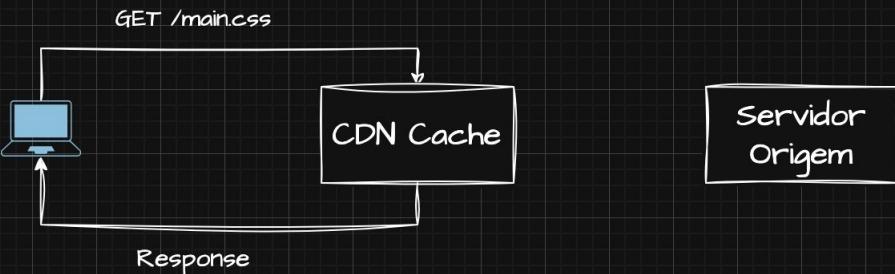
- Rede de servidores distribuídos geograficamente
- Edge Location (Proximo do Usuário)
- Armazena conteúdo estático
- Imagens, Vídeos, CSS, JS
- Estratégia de Cache Aside (miss->origem, hit->edge)
- Replicação Geográfica
- Invalidação via expiração ou manual

Cache de Conteúdo Distribuído (CDN Cache)

Primeiro Request

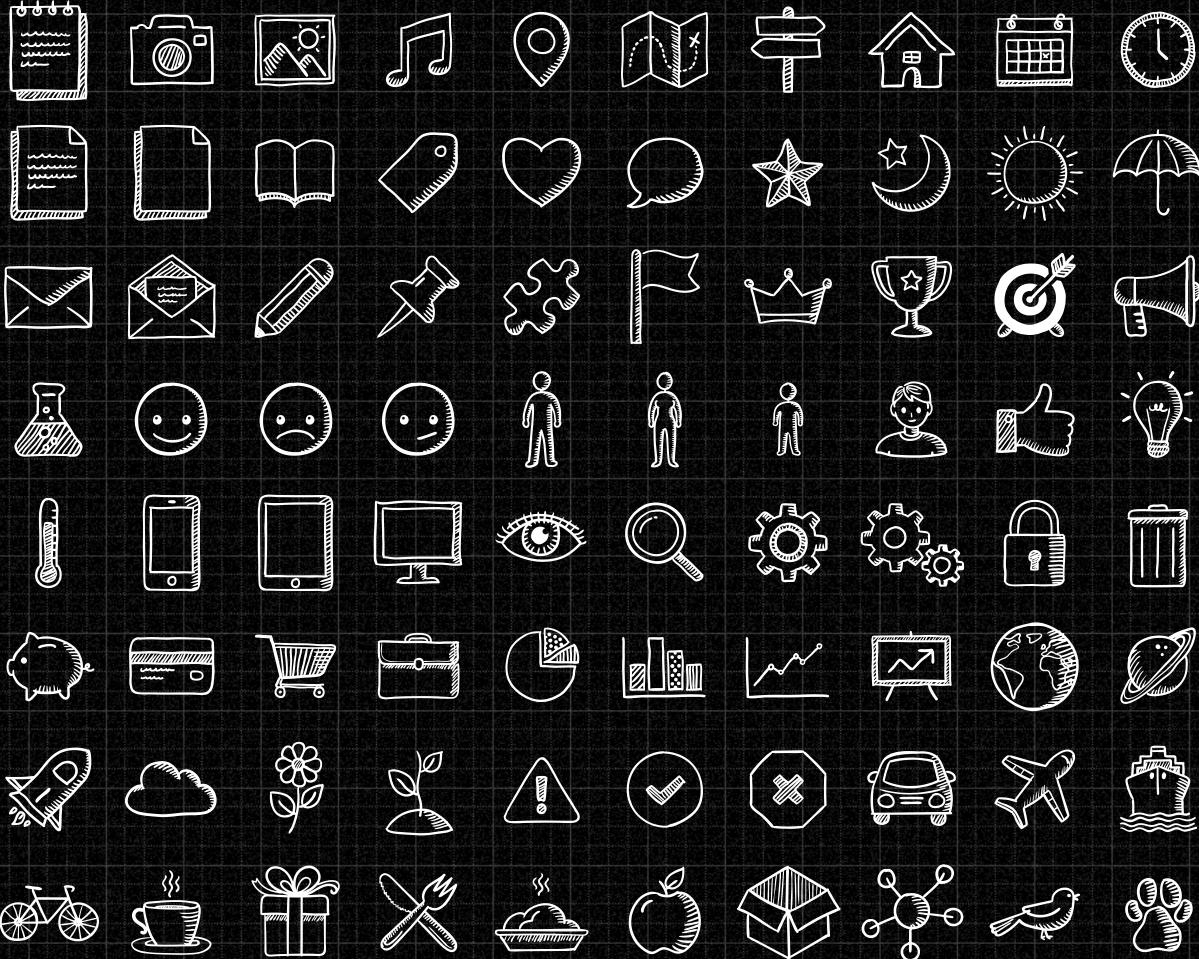


Demais Requests



Cache de Conteúdo Distribuído (CDN Cache)

- Reduz Latência Geográfica
- Reduz carga da origem
- Suporta picos de tráfego
- Inclui segurança (DDoS, Firewall, Threat Detection)
- Invalidação Assincrona e Não Performática
- Custo de Operação em Larga Escala



SlidesCarnival icons are **editable shapes**.

This means that you can:

- Resize them without losing quality.
- Change fill color and opacity.

Isn't that nice? :)

Examples:

