

# Análise e Melhorias do Cache em Memória Personalizado

## 1. Estrutura Atual

O cache atual utiliza um Map nativo do JavaScript com funcionalidades como:

- TTL (tempo de expiração) por chave
- Limite de chaves
- Limpeza periódica
- Singleton para instância única

## 2. Problemas Detectados

- Concorrência em múltiplas threads/processos: o Map não é thread-safe.
- Condição de corrida: chamadas concorrentes a set/get podem interferir.
- Limpeza síncrona: cleanup() bloqueia o event loop se o cache for grande.
- Política de remoção fraca: FIFO simples, sem análise de uso real.

## 3. Melhorias Sugeridas

- Usar Mutex (ex: async-mutex) para proteger chamadas concorrentes.
- Implementar LRU com Map + DoublyLinkedList para remoção eficiente.
- Fazer cleanup assíncrono com setInterval ou worker thread.
- Substituir TTL global por setTimeout individual por chave.
- Modularizar: separar CacheStore (dados) de CacheController (regras).

## 4. Recursos Futuramente Interessantes

- Cache distribuído com Redis para clusters.
- Suporte a stale-while-revalidate para revalidação assíncrona.

## **Análise e Melhorias do Cache em Memória Personalizado**

- Métricas e contadores de cache hits/misses.
- Persistência leve com journal de expiração.