



**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS**

**FUNDAMENTOS E TÉCNICAS DE VERIFICAÇÃO DE SOFTWARE**

**Relatório Projeto Redis**

Isabella Maria Tressino Bruno.....RA: 23013050  
Izabelle de Oliveira Silva.....RA: 23001444  
Natália Naomi Sumida.....RA:23008255  
Jéssica Silva Kushida.....RA:23012069

Campinas

2024

## **1. Solução Implementada**

### **1.1. Sincronização inicial do cache**

Para garantir que o Redis sempre contenha os dados mais recentes ao iniciar o servidor, uma rota de sincronização “/syncProducts” é chamada automaticamente. A rota busca e recupera os produtos cadastrados no MySQL e os produtos armazenados no cache.

Após isso, na função “compareArrays”, é gerado um hash para cada produto obtido no banco de dados e as duas listas (de produtos no banco e produtos no cache) são comparadas:

- Se um produto do banco não está presente no cache, isso indica que é um novo item;
- Se o produto existe no cache, mas com o hash diferente, significa que o produto foi alterado;

Ambos são adicionados à uma lista e retornados, para que sejam inseridos no Redis, garantindo a sincronização dos dados.

### **1.2. Consultas**

#### **1.2.1. Consultar todos os produtos**

Para obter todos os produtos, a consulta é feita diretamente no cache, partindo do pressuposto de que o Redis está atualizado pela sincronização inicial.

#### **1.2.2. Consultar produto pelo ID**

Ao buscar um produto específico pelo ID, primeiro a consulta é feita diretamente no cache. Se não estiver no cache, é buscado no banco de dados, se encontrado é adicionado ao cache.

### **1.3. Inserir Produto**

Ao cadastrar um novo produto ele é salvo tanto no banco de dados quanto no cache com seu respectivo hash, gerado com base nos valores do produto.

## **1.4. Atualizar Produto**

Quando um produto é atualizado, tanto o banco de dados quanto o cache são atualizados.

## **1.5. Deletar Produto**

Ao excluir um produto, o sistema remove o item do banco de dados e do cache.

## **2. Problemas que não resolve**

Após a implementação do cache com Redis ao sistema, apresenta algumas limitações, onde a implementação atual não resolve(trata) os problemas como:

### **1. Sincronização entre cache e o Banco de Dados**

- O cache atual pode ficar desatualizado se os dados forem atualizados sem uma atualização certa no Reids

### **2. Atualização fora do Sistema**

- Worker para monitorar e atualizar cache pode ser configurado para periodicamente checar o banco de dados por mudanças fora do sistema .Quando detectar alterações, o Worker atualiza automaticamente o cache, sincronizando os dados.

## **3. Resultado Observados**

Após testarmos a solução implementada, observamos que a realização de consultas diretamente pelo cache diminuiu o tempo de resposta, devido a redução no número de acessos ao banco de dados.

Além disso, a solução demonstrou consistência nos dados atualizados. Os produtos que foram inseridos, atualizados ou excluídos foram corretamente armazenados tanto no banco de dados quanto no cache, garantindo a sincronização entre os dois.