

# REDIS

## Aula 21: Redis e SQL

Banco de Dados 2 (BD2)

# Objetivo



## **Criar um sistema com cache usando Redis e PostgreSQL**

- Entender melhor o que é cache e por que usá-lo;
- Integrar Redis e PostgreSQL com Python;
- Construir um sistema que:
  - Consulta dados do PostgreSQL
  - Armazena esses dados em Redis como cache
  - Verifica o Redis antes de ir ao SQL
  - Usa TTL para expiração do cache

# Problema



Você está desenvolvendo uma aplicação que exibe informações de produtos de um banco de dados. Seu objetivo é:

- Melhorar a velocidade de acesso aos dados
- Evitar consultas repetidas ao banco SQL
- Usar Redis como cache

# Cache



Cache é uma cópia temporária de dados usados com frequência.

## **Objetivo:**

- Evitar buscar o mesmo dado várias vezes no sistema de origem
- Reduzir carga no banco de dados
- Melhorar desempenho da aplicação

## **Cache global (compartilhado por todos)**

- Quando os dados são iguais para qualquer usuário
- Ex: lista de produtos, nomes de estados, configurações do sistema, taxas fixas

## **Cache por usuário**

- Quando os dados variam de acordo com o usuário logado
- Ex: carrinho de compras, sessão, preferências, histórico, permissões

# Cache



## **Exemplo prático:**

- Você pesquisa o mesmo produto várias vezes. Sem cache, o sistema consulta o SQL a cada clique. Com cache, a informação vem direto da RAM.

## Redis é ideal:

- Leitura e escrita em milissegundos
- Suporte nativo a expiração de chaves (TTL)
- Mesmo modelo de um localStorage

# Mas e....?

Inicialmente, vamos ver como isso funciona no front-end (localStorage) e, depois, aplicar a mesma lógica no back-end com Python + SQL + Redis





# Banco SQL

Use o arquivo **produtos.txt** para popular seu banco de dados.

Pode criar o banco com o nome: **redisCache**

id	nome	preco
1	Teclado Mecânico RGB	249.90
2	Mouse Gamer	129.50
3	Monitor 27"	1049.00
4	Notebook i7 16GB	3850.00
5	Headset USB	199.99
6	Cadeira Gamer	899.90
7	Webcam Full HD	299.00

# Sistema



Fluxo:

1. O usuário informa o ID do produto;
2. O sistema verifica se a chave produto:<id> existe no Redis;
  - **Se sim:** lê do Redis e mostra na tela
  - **Se não:** consulta o banco, salva no Redis e exibe
3. Exiba a chave HASH com os campos e valores.

OBS: A chave no Redis expira em 60 segundos (TTL)

Mas primeiro?

1. Que tipo de dados usaremos no REDIS?
2. Qual a nomenclatura das chaves?



# Sistema



Fluxo:

1. O usuário informa o ID do produto;
2. O sistema verifica se a chave produto:<id> existe no Redis;
  - **Se sim:** lê do Redis e mostra na tela
  - **Se não:** consulta o banco, salva no Redis e exibe
3. Exiba a chave HASH com os campos e valores.

OBS: A chave no Redis expira em 60 segundos (TTL)

Mas primeiro?

1. Que tipo de dados usaremos no REDIS? **HASH**
2. Qual a nomenclatura das chaves? **produto:<id>**

# Fluxo



```
--- MENU DE CONSULTA DE PRODUTOS ---
```

```
1 - Buscar produto por ID
```

```
2 - Listar produtos em cache (Redis)
```

```
3 - Sair
```

```
Escolha uma opção: 1
```

```
Informe o ID do produto: 8
```

```
Produto não está no cache. Buscando no banco de dados...
```

```
Produto não encontrado no banco.
```

```
--- MENU DE CONSULTA DE PRODUTOS ---
```

```
1 - Buscar produto por ID
```

```
2 - Listar produtos em cache (Redis)
```

```
3 - Sair
```

```
Escolha uma opção: 1
```

```
Informe o ID do produto: 2
```

```
Produto não está no cache. Buscando no banco de dados...
```

```
Produto salvo no cache com TTL de 60 segundos.
```

```
Produto: Mouse Gamer - R$ 129.50
```

```
--- MENU DE CONSULTA DE PRODUTOS ---
```

```
1 - Buscar produto por ID
```

```
2 - Listar produtos em cache (Redis)
```

```
3 - Sair
```

```
Escolha uma opção: 2
```

```
Chaves de produtos presentes no Redis:
```

```
produto:2
```

```
  nome: Mouse Gamer
```

```
  preco: 129.50
```