

Tech Emporium

Plataformas, Microserviços, APIs e DevOps

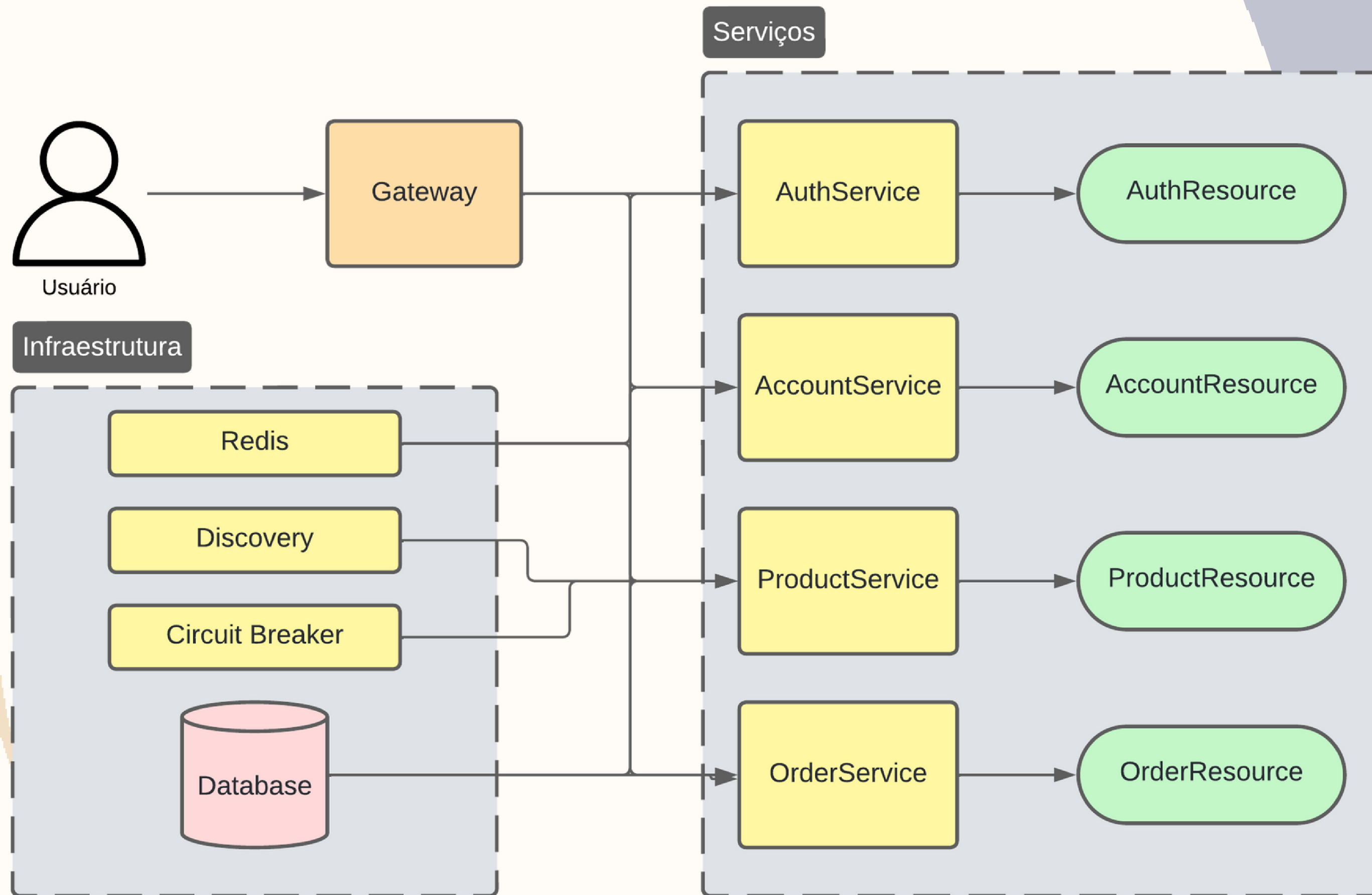
Lucca Hiratsuca Costa

Thomas Chiari Ciocchetti de Souza

Overview

- Loja de produtos eletrônicos
- Cadastro de usuários para sistema da loja
- Usuários autenticados podem realizar operações CRUD com produtos, pedidos e clientes

Fluxograma



Serviços

Gateway

- Redirecionamento de requisições

Product Service

- Gerenciamento de Produtos
 - Criação, Leitura, Atualização e Deleção (CRUD)

Order Service

- Gerenciamento de Pedidos (CRUD)

Account Service

- Gerenciamento de contas de usuário (CRUD)

Auth Service

- Gerenciamento de autenticação de usuários (CRUD)

Circuit Breaker

- Aumentar a resiliência e estabilidade do sistema
- Proteção contra falhas em chamadas
- Impede falhas em cascata em cada serviço

Como funciona?

- Interrompe chamadas externas ou problemáticas
- Prevenir erros sequenciais, que podem gerar condições de falha

1. **Closed State:** todas as chamadas são permitidas. Se há um número definido de falhas, passam para o Open State.
2. **Open State:** chamadas são redirecionadas para o método de fallback.
3. **Half-Open State:** apenas uma quantidade limitada de chamadas é permitida se o problema for resolvido.

Circuit Breaker

- Aumentar a resiliência e estabilidade do sistema
- Proteção contra falhas em chamadas
- Impede falhas em cascata em cada serviço

Como funciona?

- Interrompe chamadas externas ou problemáticas
- Prevenir erros sequenciais, que podem gerar condições de falha

1. **Closed State:** todas as chamadas são permitidas. Se há um número definido de falhas, passam para o Open State.
2. **Open State:** chamadas são redirecionadas para o método de fallback.
3. **Half-Open State:** apenas uma quantidade limitada de chamadas é permitida se o problema for resolvido.

Redis

- Armazenamento em Cache
- Melhorar performance e escalabilidade de cada serviço
- Armazenamento de estruturas em dados de memória

Como funciona?

- Cachear métodos de criação, leitura, atualização, para cada um dos serviços
- Dados acessados muito mais rapidamente

1. **Armazenamento em Cache:** armazenar em dados de memória torna o acesso muito mais rápido que em bancos de dados.
2. **Redução de Latência:** reduz as cargas sobre o banco de dados principal; requisições são atendidas a partir do cache.
3. **Suporte a Estruturas:** o Redis consegue suportar uma maior variedade de estruturas de dados.

Redis

- Armazenamento em Cache
- Melhorar performance e escalabilidade de cada serviço
- Armazenamento de estruturas em dados de memória

Como funciona?

- Cachear métodos de criação, leitura, atualização, para cada um dos serviços
- Dados acessados muito mais rapidamente

1. **Armazenamento em Cache:** armazenar em dados de memória torna o acesso muito mais rápido que em bancos de dados.
2. **Redução de Latência:** reduz as cargas sobre o banco de dados principal; requisições são atendidas a partir do cache.
3. **Suporte a Estruturas:** o Redis consegue suportar uma maior variedade de estruturas de dados.

Obrigado!

Acesse o nosso projeto

