EBOOK

Guideline

MICROSSERVIÇOS

Os Guardiões da Galáxia do Desenvolvimento de Software

Criado por Tiago Peniche

O QUE É ARQUITETURA DE MICROSSERVIÇOS?

CODE AND SHINE

Imagine um sistema de RPG de mesa. Cada jogador representa um personagem com habilidades e responsabilidades específicas.

Da mesma forma, em uma arquitetura de microsserviços, cada serviço é como um personagem em seu sistema de software.

Em resumo, eles são independentes, têm funções específicas e trabalham juntos para atingir um objetivo comum.

ANALOGIA NERD:
OS GUARDIÕES DA GALÁXIA E OS MICROSSERVIÇOS

- Star-Lord: O líder da equipe, coordenando e gerenciando os serviços.
- Gamora: A assassina altamente treinada, responsável pela segurança do sistema.
- Drax: O brutamontes forte, lidando com tarefas pesadas de processamento.
- Groot: O gigante protetor, fornecendo armazenamento de dados.
- Rocket: O gênio técnico, criando e mantendo os serviços.

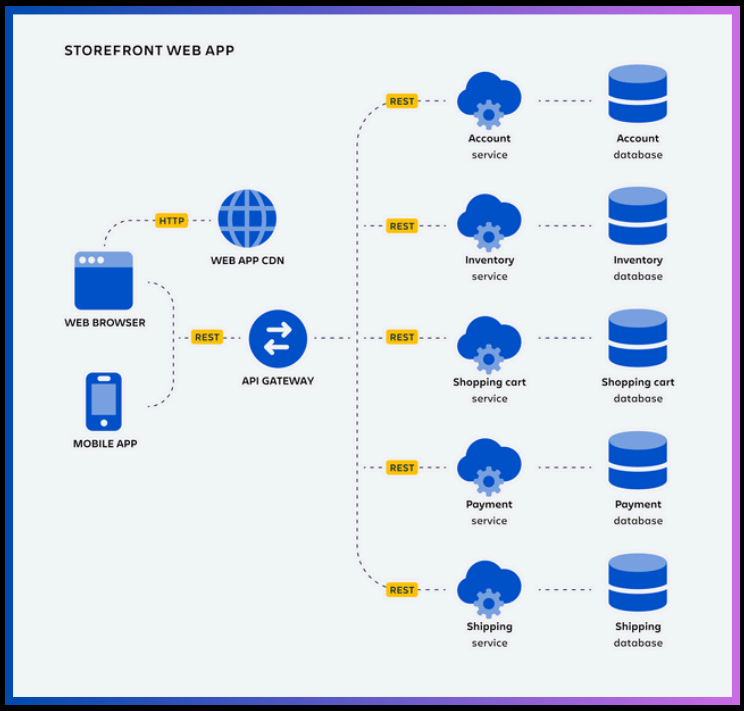
COMO OS MICROSSERVIÇOS FUNCIONAM

Os microsserviços se comunicam entre si através de APIs. Pense neles como mensagens de texto que os personagens usam para se coordenar no RPG de mesa. Essas APIs definem como os serviços podem interagir e trocar dados.

Os microsserviços funcionam como uma equipe de especialistas trabalhando juntos para atingir um objetivo comum. Cada microsserviço é responsável por uma tarefa específica e se comunica com os outros serviços por meio de APIs.

API'S

As APIs são como mensagens de texto que os microsserviços usam para se comunicar. Elas definem como os serviços podem interagir e trocar dados.



Por exemplo, um microsserviço de "pedidos" pode ter uma API que permite que outros serviços criem, atualizem ou excluam pedidos.

COMUNICAÇÃO ENTRE MICROSSERVIÇOS

Os microsserviços podem se comunicar entre si de várias maneiras, incluindo:

- **HTTP:** O protocolo da web padrão, usado para comunicação entre serviços na Internet.

- **Mensageria:** Sistemas de mensagens como Kafka ou RabbitMQ, usados para comunicação assíncrona.

- ***Barramento de Eventos***: Um mecanismo que permite que os serviços publiquem e assinem eventos, facilitando a comunicação em tempo real.

EXEMPLO DE FUNCIONAMENTO

1. Um usuário acessa um site de comércio eletrônico e adiciona um item ao carrinho.

2. O microsserviço de "carrinho de compras" recebe a solicitação e atualiza o carrinho do usuário.

3. O microsserviço de "carrinho de compras" chama o microsserviço de "catálogo" para obter informações sobre o item adicionado.

4. O microsserviço de "catálogo" retorna as informações do item, que são exibidas no carrinho do usuário.

BENEFÍCIOS DOS MICROSSERVIÇOS EM API'S

- Flexibilidade: Permite que os serviços sejam desenvolvidos e implantados independentemente.

- Escalabilidade: Permite que os serviços sejam dimensionados individualmente com base na demanda.

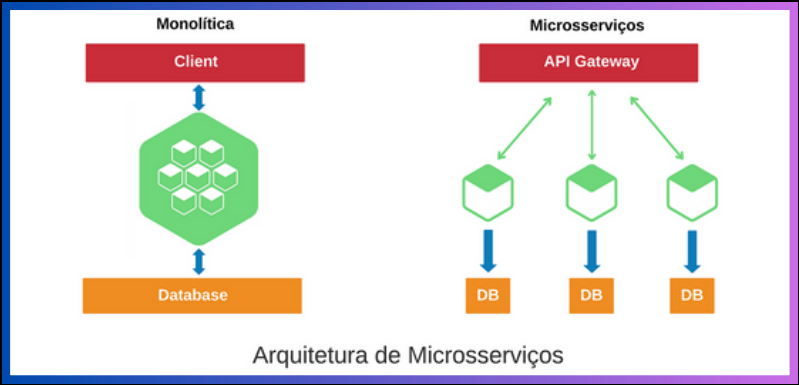
- Reutilização: As APIs podem ser reutilizadas por vários serviços, reduzindo a duplicação de código.

- Confiabilidade: Se um microsserviço falhar, os outros serviços continuarão funcionando, minimizando o tempo de inatividade.

MONOLITOS VS. MICROSSERVIÇOS

Antes da era dos microsserviços, os sistemas de software eram frequentemente construídos como monolitos.

Um monolito é como uma fortaleza gigante, onde todo o código do sistema é armazenado em um único pacote, as diferenças entre eles são mostradas na figura abaixo:



Arquitetura de Microsserviços

DESVANTAGENS DOS MONOLITOS

- Rígidos: É difícil fazer alterações ou atualizações em um monolito sem afetar todo o sistema.

- Difíceis de escalar: Dimensionar um monolito pode ser desafiador, pois requer escalar todo o sistema.

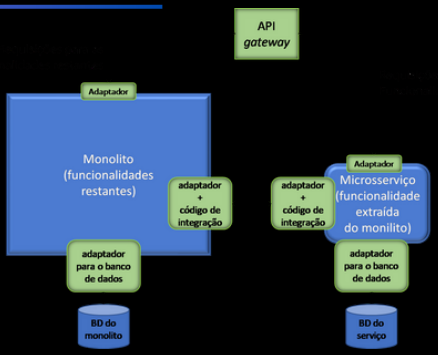
- Menos confiáveis: Se uma parte do monolito falhar, todo o sistema pode ficar inativo.

MIGRANDO DE MONOLITOS PARA MICROSSERVIÇOS

Migrar um monolito para uma arquitetura de microsserviços pode ser como desmontar uma fortaleza e reconstruí-la peça por peça. Aqui estão algumas estratégias:

- Identifique os limites: Divida o monolito em módulos ou componentes que podem ser separados em serviços independentes.

- Crie APIs: Defina APIs para que os serviços possam se comunicar e trocar dados.



- Refatore gradualmente: Reconstrua os módulos um de cada vez, convertendo-os em microsserviços.

- Teste e implante: Teste cada microsserviço individualmente e implante-os gradualmente no novo sistema.

BENEFÍCIOS DA MIGRAÇÃO

- Maior flexibilidade: Você pode fazer alterações ou atualizações em serviços individuais sem afetar o sistema inteiro.

- Escalabilidade aprimorada: Você pode dimensionar serviços específicos com base na demanda.

- Maior confiabilidade: Se um microsserviço falhar, os outros serviços continuarão funcionando, minimizando o tempo de inatividade.

OBRIGADO POR LER ATÉ AQUI

Este Ebook foi criado baseado em um artigo homônimo usando IA para coleta de informações e revisado e diagramado por mim

O passo a passo de como foi realizado esse projeto encontra-se em um repositório no meu Github abaixo