Arquitetura da Informação e Microsserviços

14.11.2020

Cleber Costa da Fonseca
clebercf@gmail.com

Objetivo

- Definir
  - Arquitetura de Software
  - Princípios arquiteturais LuizaLabs (2018-2019)
  - Arquitetura monolítica
  - Microsserviços
  - API
  - Arquitetura utilizando microsserviços

Atividade #1

Complete as frases

- “Eu acho que arquitetura de software é…”
- “Eu acho que a principal missão de um arquiteto de software é...”

Arquitetura de Software (da web - Wikipedia)

- *“Interagir com usuários, patrocinadores e qualquer stakeholder (interessado) com o intuito de determinar as necessidades que precisam ser realizadas pelo software”*
- *“Gerar a visão mais macro de requisitos do software, baseado nas necessidades do usuário e outras restrições como custo e prazo”*
- *“Garantir que essa visão macro é consistente, completa, correta e operacionalmente definida”*
- *“Fazer análise de custo/benefício para determinar os melhores métodos para atender os requisitos de software: maximizando o uso de produtos e componentes prontos”*

http://en.wikipedia.org/wiki/Software_architect

Arquitetura de Software (definição formal)

- *“A arquitetura de software de um programa ou sistema computacional trata-se da estrutura ou estruturas do sistema, que é composta dos elementos, as propriedades visíveis externamente destes elementos e o relacionamento entre eles”* - Software Architecture in Practice, segunda edição

Palavras chave da definição:

– Estrutura, elementos, interfaces (propriedades visíveis externamente) e relacionamento.

Arquitetura de Software (definição prática)

- São todas as definições técnicas relevantes que definem como o sistema deve ser estruturado, construído e implantado para atender de modo produtivo e padronizado os requisitos funcionais e não-funcionais de um projeto de software
- Planejar os pilares do Software para não ter surpresas ao longo do projeto

Arquitetura de Software (tenta evitar...)

- “Não entendo porque essa integração não está funcionando. Eu tinha fé que iria funcionar de primeira. Acho que vamos perder o prazo...”
- “O sistema está lento? Quem foi que falou que o sistema deveria ser rápido? Aliás, para mim ele está ótimo!”
- “Nossa, dar manutenção nesse sistema me tira do sério. Você mexe em qualquer coisinha e já aparece um monte de defeito no restante do sistema.”
- “Os implementadores não estão conseguindo ser produtivos? Esses implementadores...”
- “Se é fácil dar manutenção nos relatórios do sistema? Depende de qual relatório. Cada um foi construído de uma maneira diferente.”

Princípios arquiteturais LuizaLabs (2018-2019)

- Segurança
- Disponibilidade e performance
- Cloud First
- Exposição por meio de APIs
- Operar na velocidade dos negócios
- Automatizar tudo
- Desenvolver vantagem competitiva, tratar dados como ativos estratégicos
- Projetar para usabilidade

Atividade #2

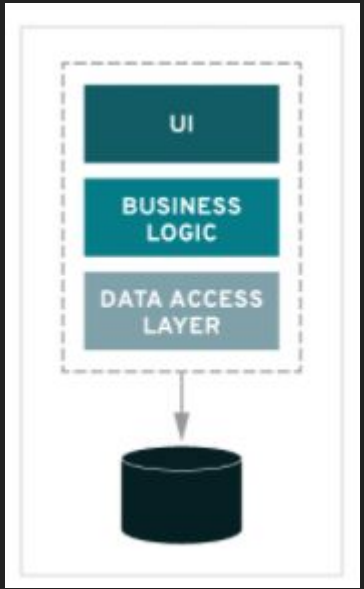
- Reflexão
  - Faz sentido continuar falando de arquitetura e arquiteto de software num projeto que segue metodologias Agile? Por quê?
  - Modelos tradicionais, se define e documenta boa parte das questões no início projeto, como ficaria em modelos ágeis?

Atividade #2

- Os desafios técnicos continuam existindo, independente do modelo de processos adotados
- Tipicamente nos primeiros sprints avalia-se a arquitetura e depois esquece?
  - Não, deve sempre ser avaliada a arquitetura ao longo dos sprints

Arquitetura monolítica

- Há uma dependência entre os serviços de uma mesma aplicação
  - forte acoplamento entre os serviços
- Manutenção de um único serviço pode mexer com o funcionamento de outras features dependentes
- É usada por uma maior fatia do mercado de desenvolvimento



https://www.redhat.com/pt-br/topics/microservices/what-are-microservices

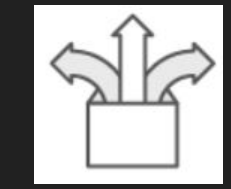
Microsserviços

- Abordagem arquitetônica e organizacional do desenvolvimento de software na qual o software consiste em pequenos serviços independentes que se comunicam usando APIs bem definidas. Esses serviços pertencem a pequenas equipes autossuficientes.

- As arquiteturas de microsserviços facilitam a escalabilidade e agilizam o desenvolvimento de aplicativos, habilitando a inovação e acelerando o tempo de introdução de novos recursos no mercado.

https://aws.amazon.com/pt/microservices/

Microsserviços (Características)

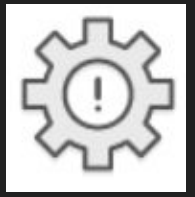


Autônomos:

- Cada serviço do componente de uma arquitetura de microsserviços pode ser desenvolvido, implantado, operado e escalado sem afetar o funcionamento de outros serviços. Os serviços não precisam compartilhar nenhum código ou implementação com os outros serviços. Todas as comunicações entre componentes individuais ocorrem por meio de APIs bem definidas.

https://aws.amazon.com/pt/microservices/

Microsserviços (Características)



Especializados:

- Cada serviço é projetado para ter um conjunto de recursos e é dedicado à solução de um problema específico. Se os desenvolvedores acrescentarem mais código a um serviço ao longo do tempo, aumentando sua complexidade, ele poderá ser dividido em serviços menores.

https://aws.amazon.com/pt/microservices/

Microsserviços (Benefícios)

Agilidade:

- Os microsserviços promovem uma organização de equipes pequenas e independentes que são proprietárias de seus serviços.
- O time fica especializado num determinado assunto e assim mais produtivo.

https://aws.amazon.com/pt/microservices/

Microsserviços (Benefícios)

Fácil implantação (deploy):

- Os microsserviços permitem a integração e a entrega contínuas, o que facilita o teste de novas ideias e sua reversão caso algo não funcione corretamente. O baixo custo de falha permite a experimentação, facilita a atualização do código e acelera o tempo de introdução de novos recursos no mercado.

https://aws.amazon.com/pt/microservices/

Microsserviços (Benefícios)

Liberdade tecnológica:

- As arquiteturas de microsserviços não seguem uma abordagem generalista. As equipes são livres para escolher a melhor ferramenta para resolver problemas específicos. O resultado é que as equipes que criam microsserviços podem optar pela melhor ferramenta para cada tarefa.

https://aws.amazon.com/pt/microservices/

Microsserviços (Benefícios)

Código reutilizável:

- A divisão do software em módulos pequenos e bem definidos permite que as equipes usem funções para várias finalidades. Um serviço criado para uma determinada função pode ser usado como componente básico para outro recurso. Isso permite que os aplicativos sejam reutilizados, pois os desenvolvedores podem criar recursos sem precisar escrever código.

https://aws.amazon.com/pt/microservices/

Microsserviços (Benefícios)

Resiliência:

- A independência do serviço aumenta a resistência a falhas do aplicativo. Em uma arquitetura monolítica, a falha de um único componente poderá causar a falha de todo o aplicativo. Com os microsserviços, os aplicativos lidam com a falha total do serviço degradando a funcionalidade, sem interromper todo o aplicativo.

https://aws.amazon.com/pt/microservices/

Microsserviços

Pontos de atenção:

- Complexidade
  - um sistema formado por partes autônomas e especializadas forma um todo bem complexo, distribuído
- Governança
  - Com vários componentes completamente distintos trabalhando juntos em um único aplicativo, a governança pode deixar a desejar.
- Rede
  - gerar uma cadeia de comunicação extremamente prolixa e interdependente

API

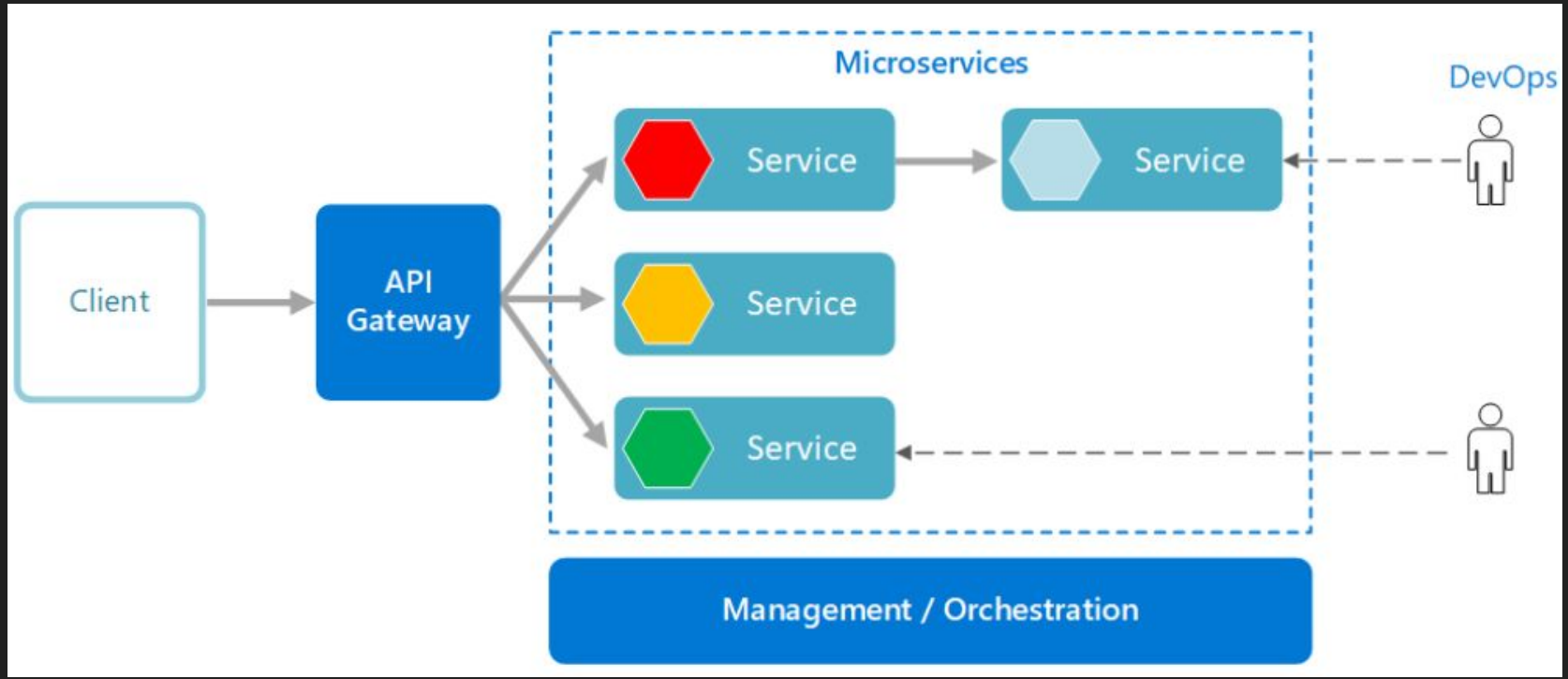
- Artefato fundamental na arquitetura de microsserviços
- DEMO
  - https://github.com/clebercf/bankslipi-api

https://aws.amazon.com/pt/microservices/

Atividade #3

- Pesquisa
  - Liste uma vantagem e outra desvantagem deste tipo de arquitetura
- Discussão

Arquitetura utilizando microsserviços



https://docs.microsoft.com/pt-br/azure/architecture/guide/architecture-styles/microservices#what-are-microservices

Arquitetura utilizando microsserviços

- API Gateway
  - Em vez de chamar serviços diretamente, os clientes chamam o gateway de API, que encaminha a chamada para os serviços adequados no back-end.
  - O Gateway de API pode executar outras funções abrangentes, como autenticação, registro em log, terminação SSL e balanceamento de carga
  - Exemplo: Apigee (gateway)

https://aws.amazon.com/pt/microservices/

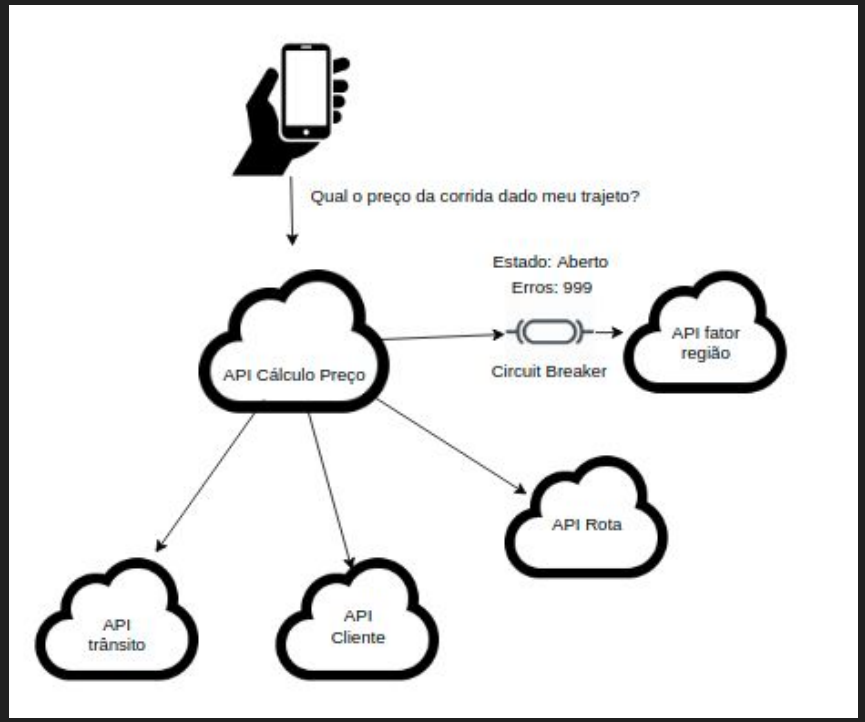
Atividade #4

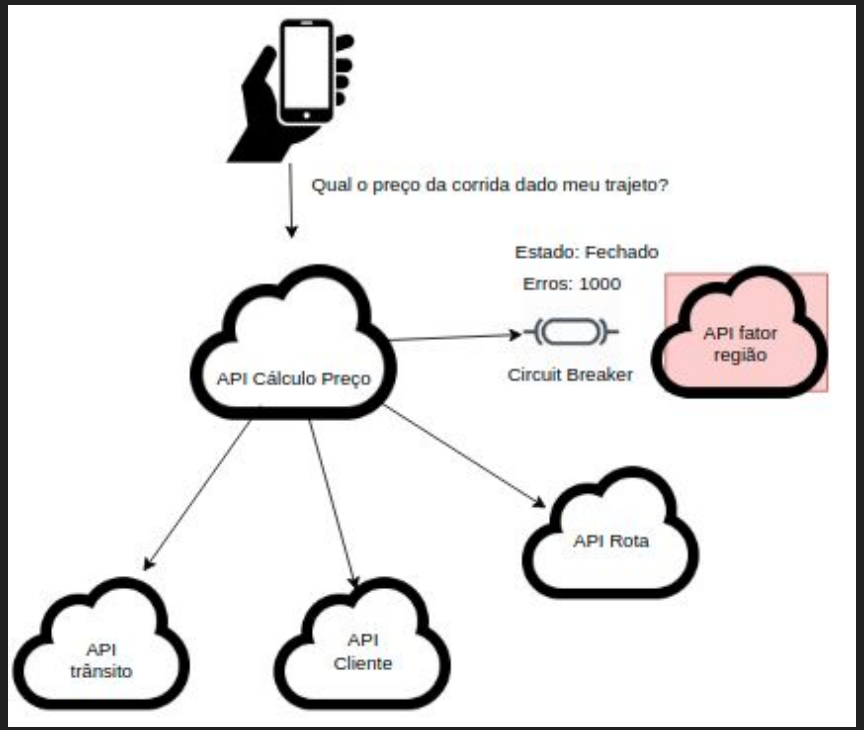
- Pesquisa
  - Quais técnicas são utilizadas para evitar falhas neste tipo de arquitetura? (ser tolerável a falha)
  - Em deploys de uma determinada api, como podemos evitar o dowtime?
  - Defina Kubernetes
  - Em aplicações mobiles, devemos invokar múltiplas apis e a questão da banda?

Atividade #4

- Pesquisa
  - Quais técnicas são utilizadas para evitar falhas neste tipo de arquitetura? (ser tolerável a falha)
    - Ex.: Circuit breaker
  - Em deploys de uma determinada api, como podemos evitar o dowtime?
    - Ex.: Deploy green/blue
  - Defina Kubernetes
    - orquestração de contêineres open-source que automatiza a implantação, o dimensionamento e a gestão de aplicações em contêineres.
  - Em aplicações mobiles, devemos invokar múltiplas apis e a questão da banda?
    - Serviços são criados para compactar numa única requisição

Atividade #4


Qual o preço da corrida dado meu trajeto?
Estado: Aberto
Erros: 999
API Cálculo Preço
Circuit Breaker
API fator região
API Rota
API trânsito
API Cliente


Qual o preço da corrida dado meu trajeto?
Estado: Fechado
Erros: 1000
API Cálculo Preço
Circuit Breaker
API fator região
API Rota
API trânsito
API Cliente

Atividade #4