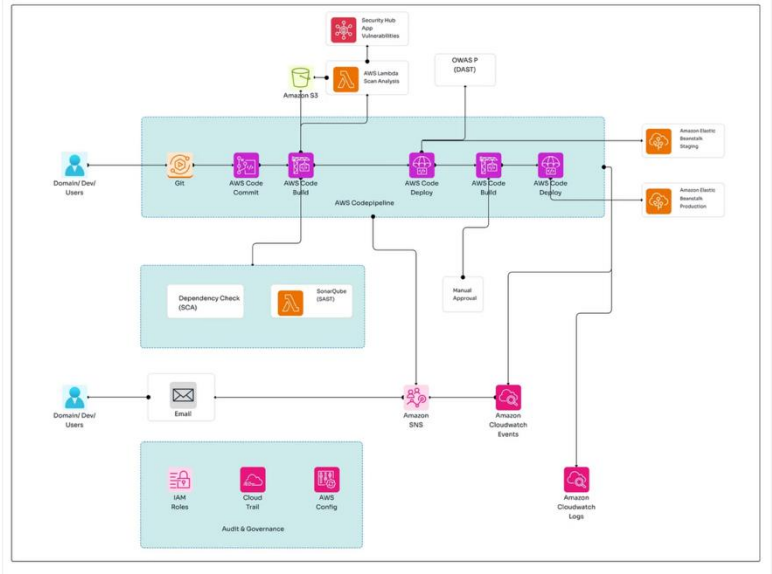
Introdução à Arquitetura de Software

O que é Arquitetura de Software?

Estrutura fundamental de um sistema de software

Conjunto de decisões importantes sobre a organização do sistema

Define componentes, suas relações, propriedades e princípios que guiam sua evolução



Papel do Arquiteto de Software

Tomar decisões técnicas de alto nível

Equilibrar requisitos funcionais e não-funcionais

Comunicar a visão técnica para stakeholders

Guiar a equipe de desenvolvimento

Diferença entre Design e Arquitetura

Arquitetura: **decisões estruturais de alto nível**

Componentes de uma Arquitetura

Estrutura

Módulos: Unidades lógicas de código

Componentes: Elementos executáveis ou implantáveis

Conectores: Mecanismos de comunicação entre componentes

Comportamento

Interações: Como os componentes se comunicam

Fluxos: Sequência de operações e dados

Estados: Condições do sistema em diferentes momentos

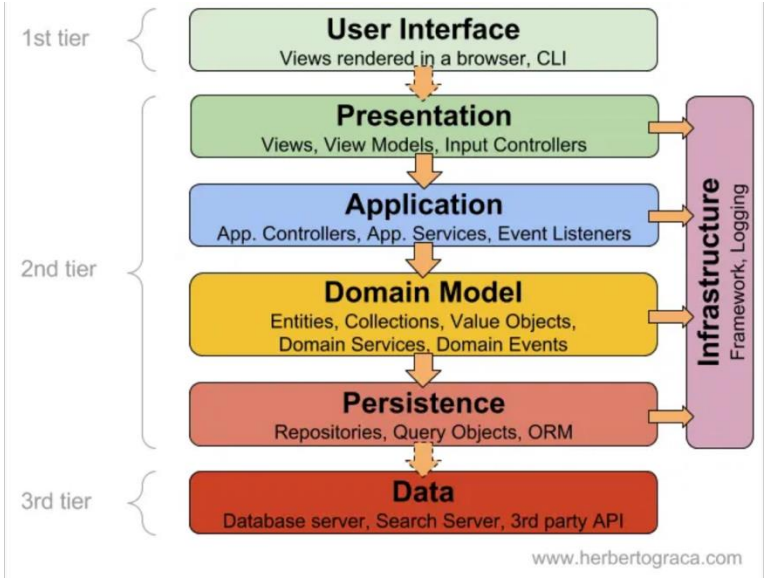
Requisitos Não-Funcionais

Desempenho: Tempo de resposta, throughput

Segurança: Proteção contra ameaças

Escalabilidade: Capacidade de crescimento

Manutenibilidade: Facilidade de manutenção



Estilos Arquiteturais: Visão Geral

O que são Estilos Arquiteturais?

Padrões recorrentes de organização e interação entre componentes que resolvem problemas específicos de design.

Principais Estilos

Arquitetura em Camadas : Organização hierárquica de componentes

Cliente-Servidor : Distribuição de responsabilidades entre provedores e consumidores

Microsserviços : Decomposição em serviços pequenos e independentes

Orientada a Eventos : Comunicação assíncrona baseada em eventos

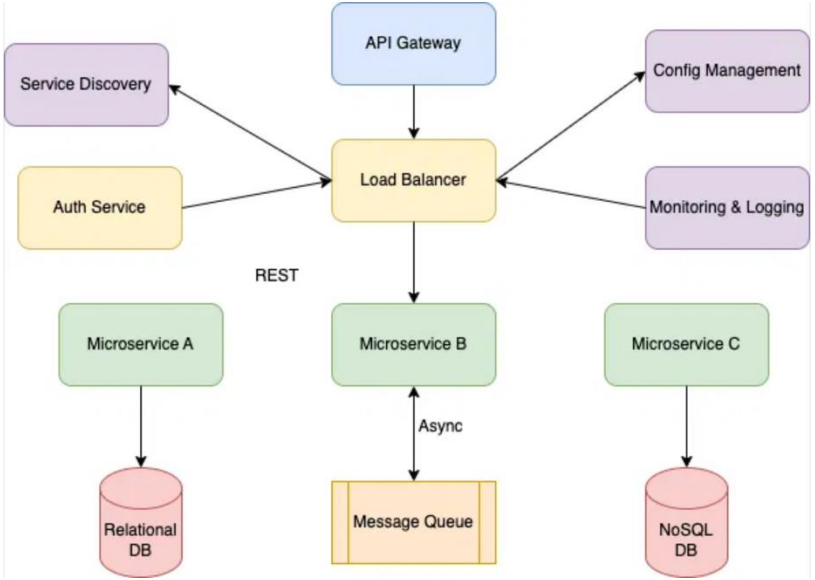
Critérios de Seleção

Requisitos não-funcionais prioritários

Natureza do problema a ser resolvido

Experiência da equipe

Restrições tecnológicas e de negócio

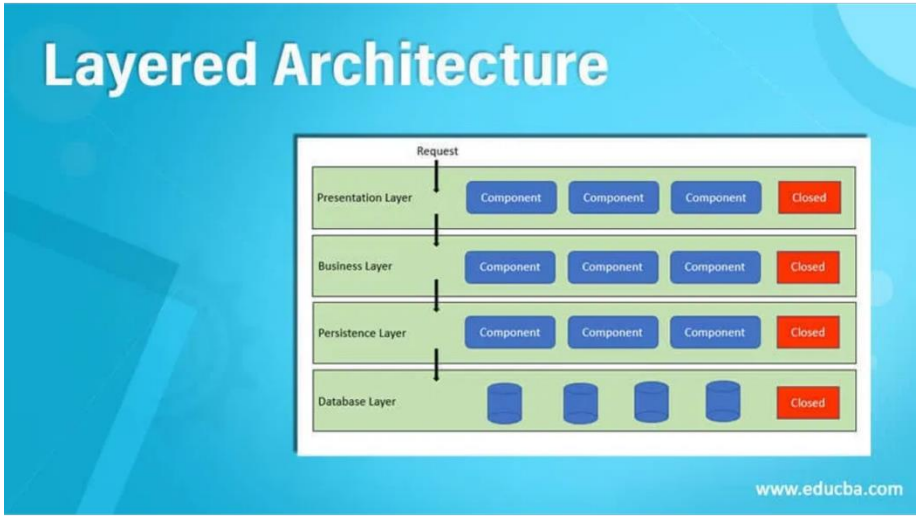


Arquitetura em Camadas

Descrição

Organiza o sistema em camadas horizontais, onde cada camada:

- Fornece serviços para a camada acima
- Consome serviços da camada abaixo
- Possui responsabilidades bem definidas



Exemplo: Aplicação Web Tradicional

- **Camada de Apresentação** : Interface com usuário
- **Camada de Aplicação** : Lógica de negócio
- **Camada de Persistência** : Acesso a dados
- **Camada de Dados** : Armazenamento

Vantagens

- Separação de responsabilidades
- Facilidade de manutenção
- Reutilização de camadas

Desvantagens

- Overhead de comunicação
- Pode gerar acoplamento
- Escalabilidade limitada

Arquitetura Cliente-Servidor

Descrição

Divide o sistema em dois componentes principais:

Cliente : Solicita serviços, geralmente interface com usuário

Servidor : Fornece serviços, processa solicitações

Exemplos Reais

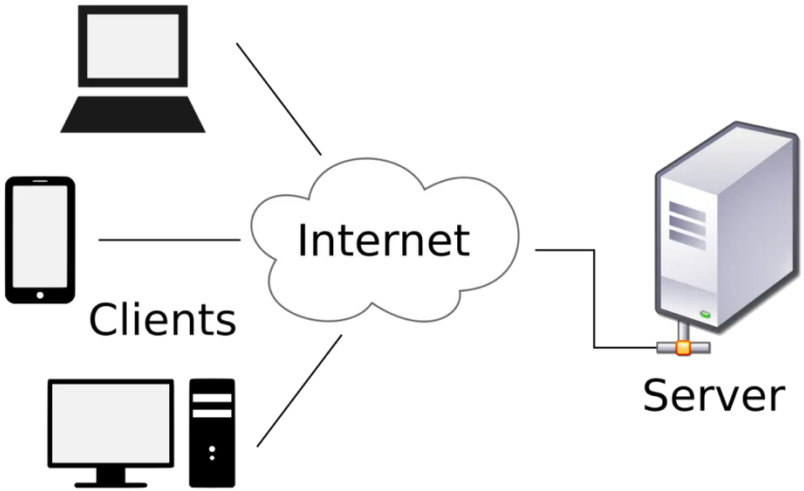
- Navegadores web e servidores HTTP
- Aplicativos de e-mail
- Jogos online
- Aplicações bancárias

Vantagens

- Centralização de recursos
- Manutenção simplificada
- Segurança centralizada

Desvantagens

- Ponto único de falha
- Congestionamento de rede
- Escalabilidade limitada



Arquitetura de Microsserviços

Descrição

Decompõe o sistema em serviços pequenos, independentes e com propósito único:

- Cada serviço implementa uma capacidade de negócio específica
- Comunicação via APIs bem definidas (geralmente REST ou mensageria)
- Implantação e escalabilidade independentes

Exemplos Reais

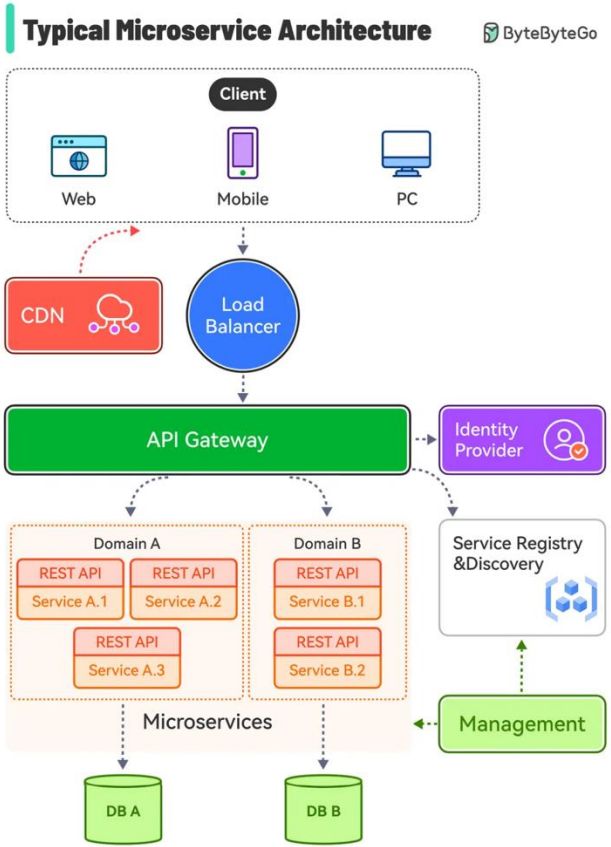
- **Netflix** : +700 microsserviços para streaming, recomendações, etc.
- **Amazon** : Decomposição do monolito em serviços independentes
- **Spotify** : Serviços para playlists, busca, recomendações

Vantagens

- Alta escalabilidade
- Implantação independente
- Resiliência

Desvantagens

- Complexidade distribuída
- Overhead de rede
- Desafios de consistência



Arquitetura Orientada a Eventos

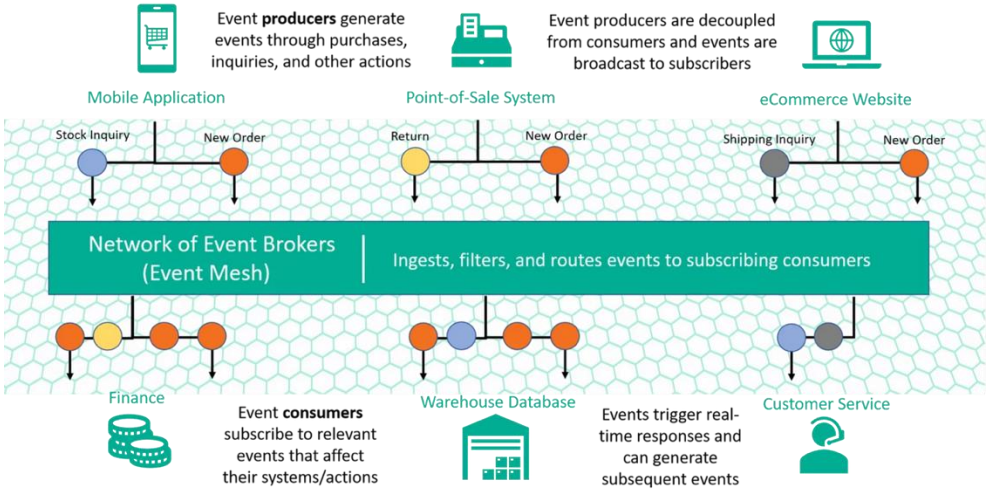
Descrição

Baseada na produção, detecção e consumo de eventos:

- **Produtores** : Geram eventos quando algo significativo ocorre
- **Broker** : Intermediário que roteia eventos
- **Consumidores** : Reagem aos eventos recebidos

Exemplos Reais

- **E-commerce** : Eventos de compra, pagamento, envio
- **IoT**: Sensores gerando eventos de leitura
- **Aplicações financeiras** : Transações bancárias
- **Redes sociais** : Notificações e atualizações



Vantagens

- Baixo acoplamento
- Alta escalabilidade
- Processamento assíncrono

Desvantagens

- Complexidade de depuração
- Garantia de entrega
- Consistência eventual

Qualidades Arquiteturais

Atributos de Qualidade (ATRs)

Disponibilidade : Tempo em que o sistema está operacional

Desempenho : Tempo de resposta, throughput, utilização de recursos

Segurança : Proteção contra acessos não autorizados e ataques

Escalabilidade : Capacidade de lidar com aumento de carga

Manutenibilidade : Facilidade de modificação e correção

Testabilidade : Facilidade de testar o sistema

Trade-offs Arquiteturais

Raramente é possível otimizar todas as qualidades simultaneamente:

Desempenho vs. Segurança : Criptografia melhora segurança mas reduz desempenho

Disponibilidade vs. Consistência : Teorema CAP - escolha dois entre consistência, disponibilidade e tolerância a partição

Escalabilidade vs. Simplicidade : Sistemas distribuídos são mais escaláveis mas mais complexos

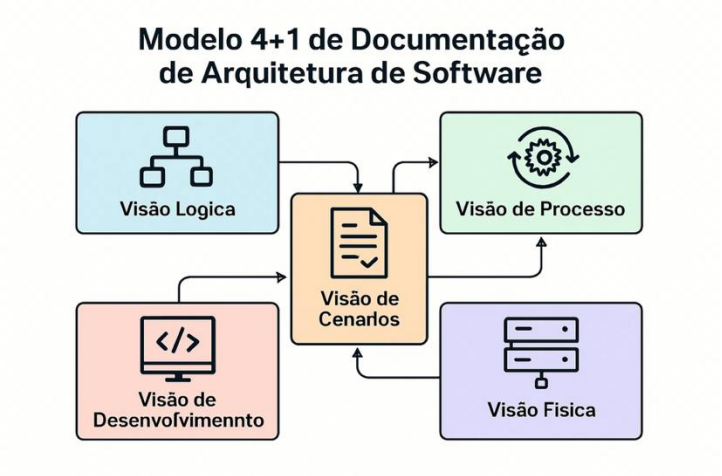
Documentação da Arquitetura

Por que Documentar?

- Comunicação entre stakeholders
- Registro de decisões arquiteturais
- Guia para implementação e manutenção

Modelo 4+1 de Visões

- **Visão Lógica** : Funcionalidades do sistema
- **Visão de Processo** : Aspectos dinâmicos
- **Visão de Desenvolvimento** : Organização do código
- **Visão Física** : Implantação
- **Visão de Cenários** : Casos de uso integradores



Ferramentas

UML, C4 Model, Architecture Decision Records (ADRs)

Atividades Práticas

Propostas de Atividades

Estudo de Caso

Para um sistema de gerenciamento de biblioteca online, qual estilo arquitetural seria mais adequado? Justifique sua escolha e aponte os principais desafios.

Análise de Trade-offs

Escolha um dos estilos arquiteturais discutidos e identifique quais qualidades arquiteturais ele tende a otimizar e quais podem ser desafiadoras.

Projeto Final (em grupo)

Proponha uma arquitetura para um novo sistema (ex: plataforma de telemedicina). Apresente o estilo arquitetural, as qualidades priorizadas, e um esboço da documentação.