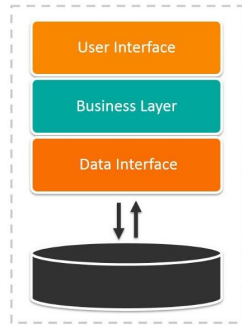


# Aula 6 - Arquitetura de Software

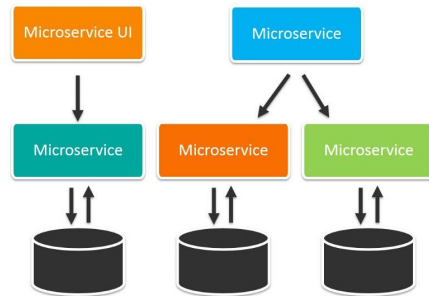
# Problema

**Cenário:** Uma empresa de e-commerce começou seu software de vendas de forma **monolítica**.

Monolithic Architecture



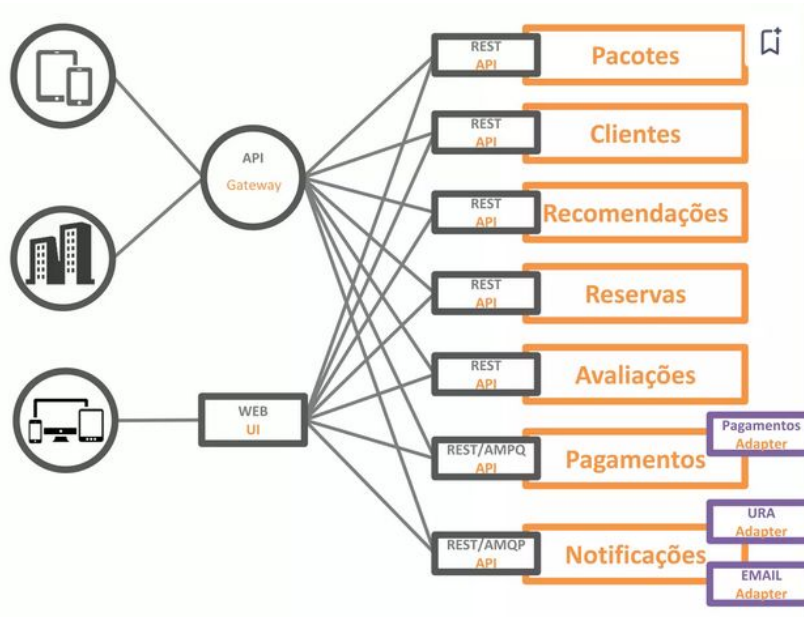
Microservices Architecture



## Problemas?

# Problema 2

**Cenário:** Uma empresa de fintech decidiu adotar uma arquitetura de **microservices** para desenvolver sua plataforma de serviços financeiros.



## Problemas?

“Arquitetura de software é a estrutura fundamental ou **partes** de um sistema de software, que define seus componentes, suas **relações** e seus **princípios de projeto** e evolução.”

**partes:** camadas, subserviços, containers...

**relações:** interações entre os componentes

**princípios de projeto:** padrões arquiteturais, setup e etc...

além de se preocupar com as responsabilidades do software: requisitos **funcionais** e **não-funcionais**.

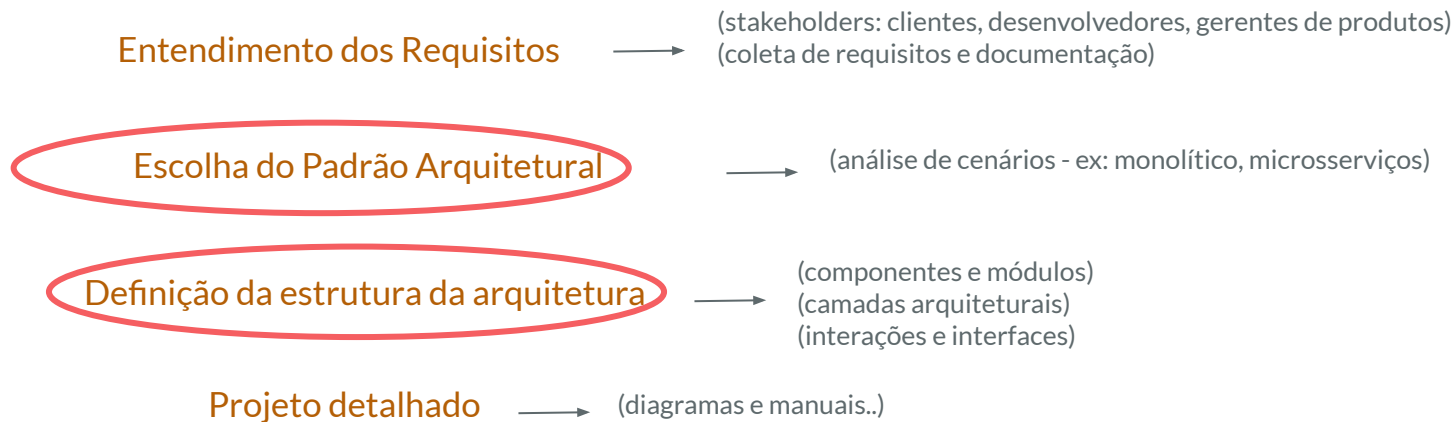
ISO/IEC/IEEE 42010:2022(en) Software, systems and enterprise —  
Architecture description

# Processo de desenvolvimento da arquitetura do software

Os **elementos arquiteturais** são modelados de forma que atenda às necessidades do cliente.

- requisitos funcionais ou não funcionais.
  - alto desempenho
  - segurança
  - processamento paralelo
  - processamento em tempo real e etc

Fornecer um roadmap e práticas recomendadas a serem seguidos durante esse desenvolvimento para que você tenha uma aplicação bem-estruturada.



entre outros..

# Como meu modelo de negócios influencia diretamente a minha arquitetura?

**Definição do cenário:** "A nossa empresa de e-commerce, que atende milhões de usuários globalmente, tem como objetivo estratégico oferecer uma experiência de compra personalizada e de alta performance."

**Desafios e Requisitos:** "Para atingir esses objetivos, enfrentamos desafios como a necessidade de escalar rapidamente durante eventos de pico, garantir a personalização em tempo real das recomendações de produtos e manter a alta disponibilidade do sistema."

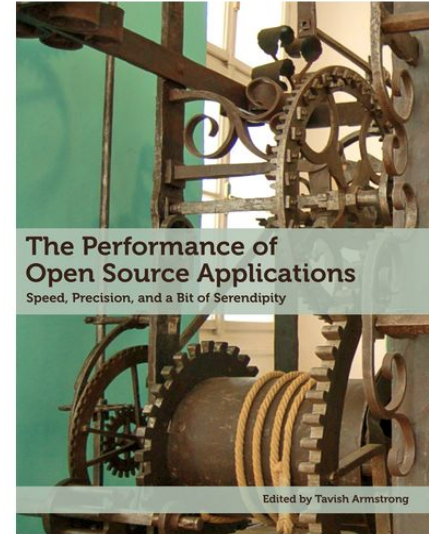
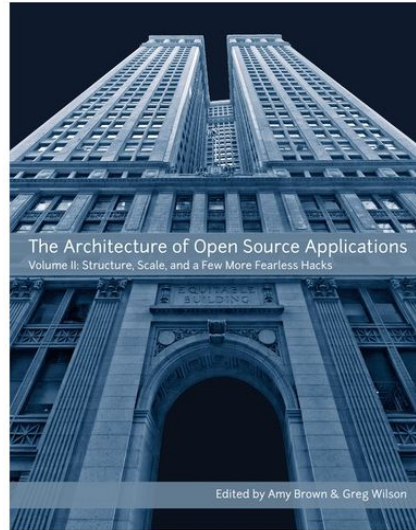
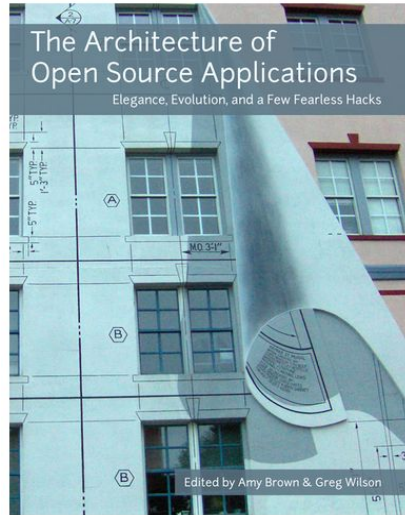
**Escolha da Arquitetura:** "Optamos por uma arquitetura de microservices porque permite escalar independentemente os diferentes componentes do sistema, como catálogo de produtos, carrinho de compras e processamento de pagamentos. Isso é crucial para lidar com o aumento do tráfego durante promoções sazonais."

**Benefícios da Arquitetura:** "Com microservices, podemos atualizar e implantar novos recursos sem afetar o sistema inteiro, permitindo uma inovação mais rápida. Além disso, a separação de responsabilidades melhora o desempenho e a resiliência, já que falhas em um serviço não comprometem os demais."

**Exemplos Práticos:** "Por exemplo, durante a última Black Friday, a arquitetura de microservices permitiu que escalássemos o serviço de pagamento independentemente do resto do sistema, reduzindo os tempos de transação em 30% e aumentando a taxa de conversão."

# Um pouco mais...

<https://aosabook.org/en/>





# Quais tipos de arquitetura mais discutidos?

1. Arquitetura monolítica
2. Arquitetura em camadas
3. Arquitetura de microsserviços



# Arquitetura monolítica

Ele é fortemente acoplado, tanto na interação entre os serviços como na forma que é desenvolvido e fornecido. Todas as funções estão em **um mesmo pacote** a ser distribuído para o cliente e toda a modularização utilizada vai trabalhar em uma mesma máquina. Dessa maneira, **todos os módulos vão compartilhar recursos de processamento, memória, banco de dados e arquivos.**



Basecamp, wordpress e etc..

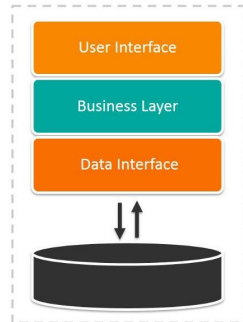
Realizar uma alteração = lançar ela novamente inteira

1x/2x ao ano..

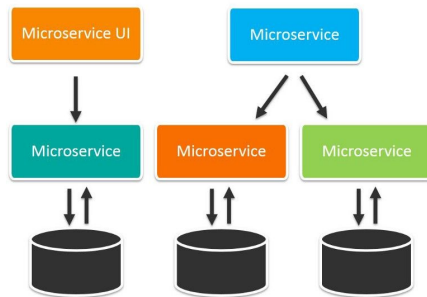
# Problema

**Cenário:** Uma empresa de e-commerce começou seu software de vendas de forma **monolítica**.

Monolithic Architecture



Microservices Architecture



## Problemas:

- Dificuldades em escalar
- Manutenção complexa
- Acoplamento complexo em um único serviço

## Consequências

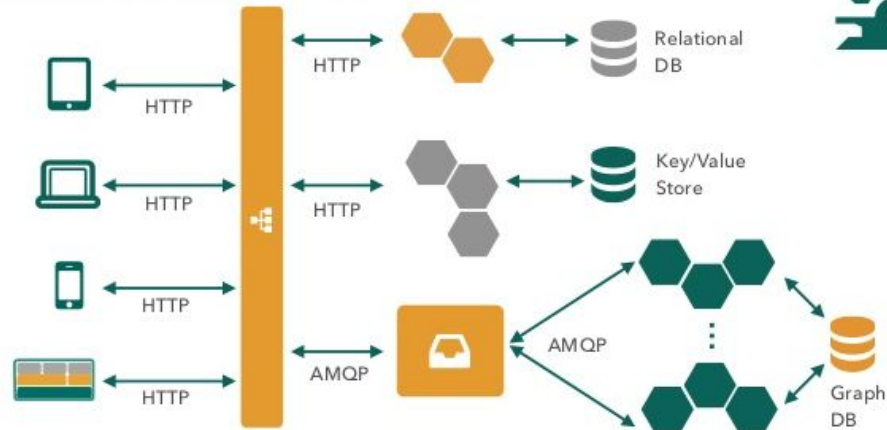
- Experiência do usuário prejudicada
- Aumento de custos
- Redução de competitividade

# Arquitetura de microsserviços

Aplicações desmembradas em componentes menores e independentes.

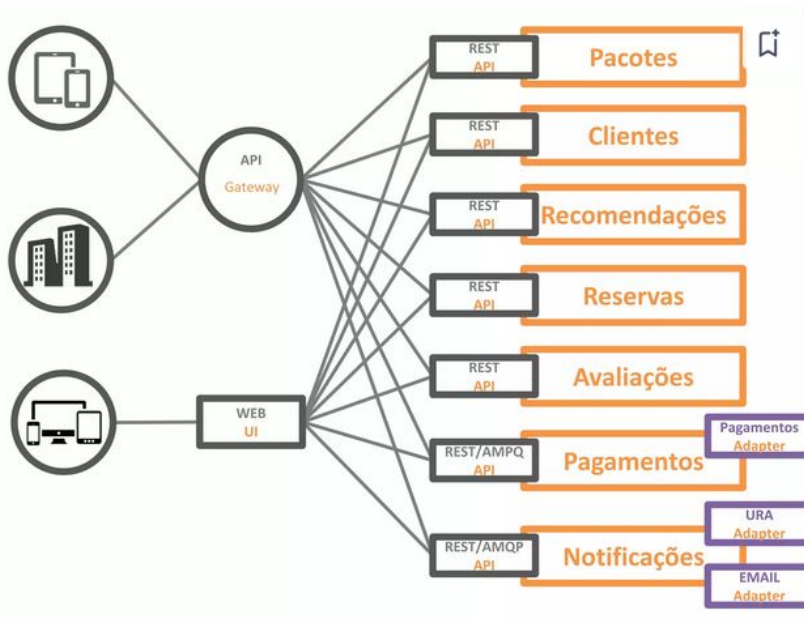
Como os serviços são implantados de maneira independente, não é necessário recriar ou reimplantar a aplicação inteira quando alterações são feitas.

## Microservice Architecture



# Problema 2

**Cenário:** Uma empresa de fintech decidiu adotar uma arquitetura de **microservices** para desenvolver sua plataforma de serviços financeiros.



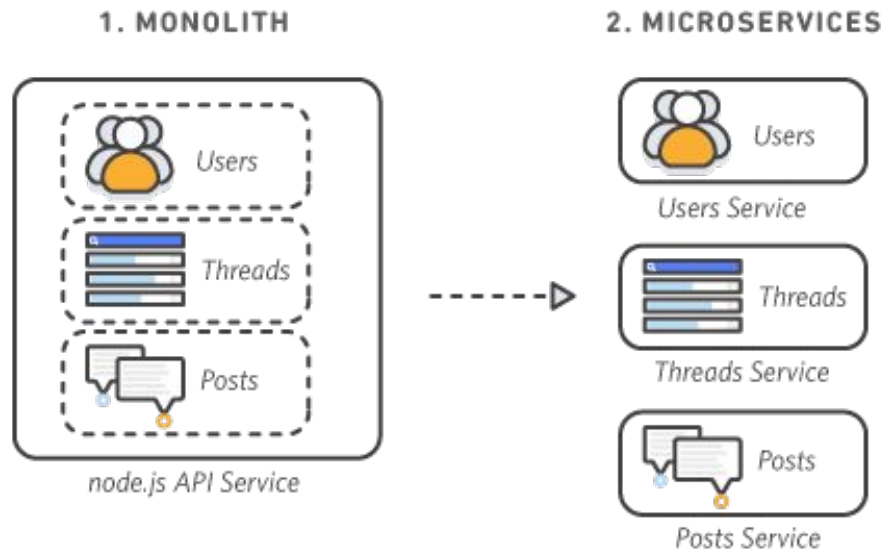
## Problemas:

- Complexidade de gerenciamento
- Latência e desempenho
- Sobrecarga de rede
- Dificuldade de depuração
- Gestão de configurações

## Consequências

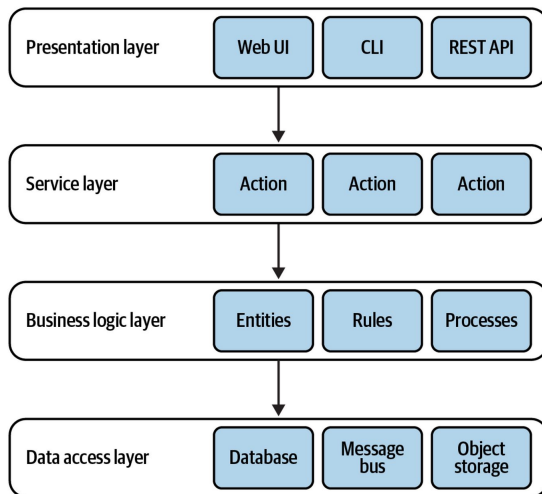
- Desempenho degradado
- Custos operacionais elevados
- Menor agilidade

# Arquitetura monolítica x Microserviços



# Arquitetura em N camadas

Uma arquitetura em camadas particiona a complexidade envolvida no desenvolvimento de um sistema em componentes menores (as camadas). As camadas são dispostas de forma hierárquica, assim uma camada só pode usar serviços da camada imediatamente inferior.



# Arquitetura em N camadas

- Camada de Apresentação: Interação com o usuário final
  - tecnologias de front
- Camada de Aplicação
  - Modelos, lógica de negócios e operações da aplicação.
  - Controllers;
- Camada de Negócios
  - Regras de negócios, validação e processamento de dados.
  - Serviços;
- Camada de Acesso aos dados
  - Acesso e manipulação de dados armazenados.
- Camada de Dados
  - Armazenamento e recuperação de dados.



# Arquitetura em N camadas (exemplo)

- **Camada de Apresentação:** Interface do usuário que exibe produtos, carrinho de compras e permite que o usuário faça pedidos.
- **Camada de Aplicação:** Serviços que gerenciam o fluxo do pedido, incluindo a adição de produtos ao carrinho e a finalização da compra.
- **Camada de Negócio:** Regras de negócio que calculam descontos, impostos e validam o pedido.
- **Camada de Acesso a Dados:** Repositórios que recuperam e armazenam informações sobre produtos, pedidos e clientes.
- **Camada de Dados:** Banco de dados que armazena informações persistentes sobre produtos, pedidos e clientes.

# Arquitetura em N camadas

## Vantagens:

- Separação de responsabilidades;
- Reusabilidade de código;
- Escalabilidade;
- Manutenibilidade;
- Segurança;

## Desvantagens:

- Complexidade
- Desempenho
- Dificuldade de depuração

# Padrões arquiteturais

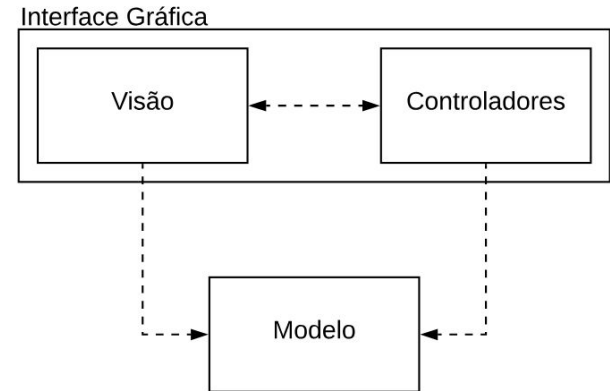
Arquitetura de software também possui padrões direcionados para cada modelo de negócio.

Padrões de arquitetura de software são soluções comprovadas e reutilizáveis para problemas recorrentes em design de software. Eles servem como guias para a estruturação de sistemas de forma eficiente e eficaz.

(próx aula)

# Padrão arquitetural MVC

- **Visão:** classes responsáveis pela apresentação da interface gráfica do sistema.
- **Controladoras:** classes que tratam e interpretam eventos gerados por dispositivos de entrada, como mouse e teclado. Como resultado de tais eventos, Controladoras podem solicitar uma alteração no estado do Modelo ou da Visão.
- **Modelo:** classes que armazenam os dados manipulados pela aplicação e que têm a ver com o domínio do sistema em construção.



# Padrão arquitetural MVC

“O coração e a parte mais preciosa de MVC está na separação entre código de interface com o usuário (a Visão, também chamada de apresentação) e a lógica do domínio (o Modelo). As classes de apresentação implementam apenas a lógica necessária para lidar com a interface do usuário. Por outro lado, objetos de domínio não incluem código visual, mas apenas lógica de negócios. Isso separa duas partes complexas de sistemas de software em partes que são mais fáceis de se modificar. Também permite várias apresentações da mesma lógica de negócio. (Fowler e Beck)”

# Teste final da aula

[https://docs.google.com/forms/d/e/1FAIpQLSdj1Z4obj-rAbOiTbwuryO88E7ozXJ3\\_iXJ8jPKWdZ2S39gyw/viewform?usp=preview](https://docs.google.com/forms/d/e/1FAIpQLSdj1Z4obj-rAbOiTbwuryO88E7ozXJ3_iXJ8jPKWdZ2S39gyw/viewform?usp=preview)

# Referências

<https://www.iso.org/obp/ui/en/#iso:std:iso-iec-ieee:42010:ed-2:v1:en>

<https://medium.com/@gwenilorac/compreendendo-as-arquiteturas-de-aplica%C3%A7%C3%B5es-web-uma-vis%C3%A3o-abrangente-dos-modelos-e-componentes-de-a30224c934dc>

<https://www.alura.com.br/artigos/padroes-arquiteturais-arquitetura-software-descomplicada>

<https://lgertel.medium.com/padr%C3%B5es-de-arquitetura-web-monol%C3%ADtica-ou-micro-servi%C3%A7os-7b3f0c9394fe>

<https://www.devmedia.com.br/principais-anomalias-arquiteturais-de-software/39807>