

# INTRODUÇÃO À ARQUITETURA DE SOFTWARE

**Prof. Sérgio Ricardo Vieira**  
**[sergio.ricardo@fmu.br](mailto:sergio.ricardo@fmu.br)**

# Conteúdo

- ☐ Introdução
- ☐ Processo Arquitetural
- ☐ Análise
- ☐ Síntese
- ☐ Avaliação
- ☐ Arquitetura de Referência
- ☐ Estudo de Caso
- ☐ Exercício

# Introdução

- O que é Arquitetura?

Disposição das partes ou elementos de um edifício ou espaço urbano. [Aurélio]

A estrutura e organização lógica de funcionamento de um sistema computacional. [Aurélio]

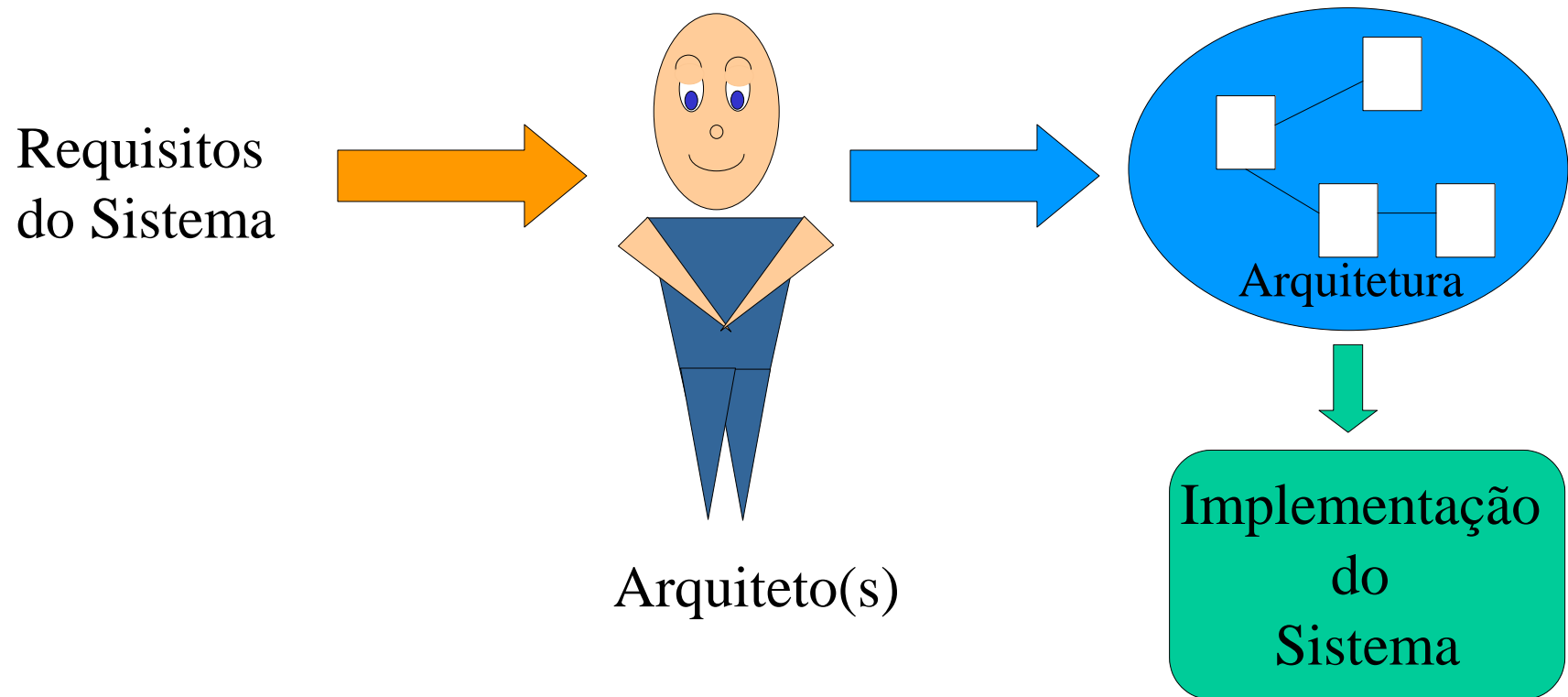
# Introdução

- O que é Arquitetura?

*Além dos algoritmos e das estruturas de dados da computação; o design e a especificação da estrutura geral do sistema emergem como um novo tipo de problema. As questões estruturais incluem organização total e estrutura de controle global; protocolos de comunicação, sincronização e acesso a dados; designação de funcionalidade a elementos de design; distribuição física; composição de elementos de design; escalação e desempenho; e seleção entre as alternativas de design.*

*An Introduction to Software Architecture, David Garlan e Mary Shaw*

# Introdução



# Introdução

Antes de começar efetivamente o projeto de um software, é necessário definir qual será a estrutura e a organização do projeto. O resultado desse passo é chamado de arquitetura de software. Esse é o primeiro estágio de qualquer projeto que deseja obter sucesso (SOMMERVILLE, 2011). Nas metodologias tradicionais de desenvolvimento de software, como no caso do Cascata, antes de qualquer coisa, era definida toda a arquitetura do software. Isso envolvia desde a organização de pastas do código-fonte até quantos servidores, além de quais padrões de projeto, seriam necessários para concluir o sistema com êxito (ENGHOLM JUNIOR, 2017; SOMMERVILLE, 2011).

Porém, com o uso das metodologias ágeis na construção de projetos de softwares, esse passo tornou-se incremental. Agora acontece sempre que existe a necessidade de entregar uma nova funcionalidade ou componente do sistema que não seja contemplado pela arquitetura atual. Dessa maneira, é importante determinar uma arquitetura que seja flexível e escalável dentro do projeto. O custo da refatoração de uma arquitetura de software durante a execução do projeto é bem caro (SOMMERVILLE, 2011)

# Introdução

Sendo assim, quando se fala em arquitetura é preciso analisar o problema da melhor maneira possível e imaginar qual é a melhor estrutura para atender à implementação do projeto. Isso guia um conjunto de decisões difíceis que devem ser tomadas. Definir uma arquitetura não tem uma receita pronta. É necessária certa experiência e estudos para entender quais componentes serão utilizados. Serão serviços de cloud computing? Qual a vantagem dessa decisão para o projeto? Os servidores serão ou não virtualizados? As aplicações serão distribuídas? Até que ponto utilizaremos essa estratégia? (SILVEIRA et al., 2011).

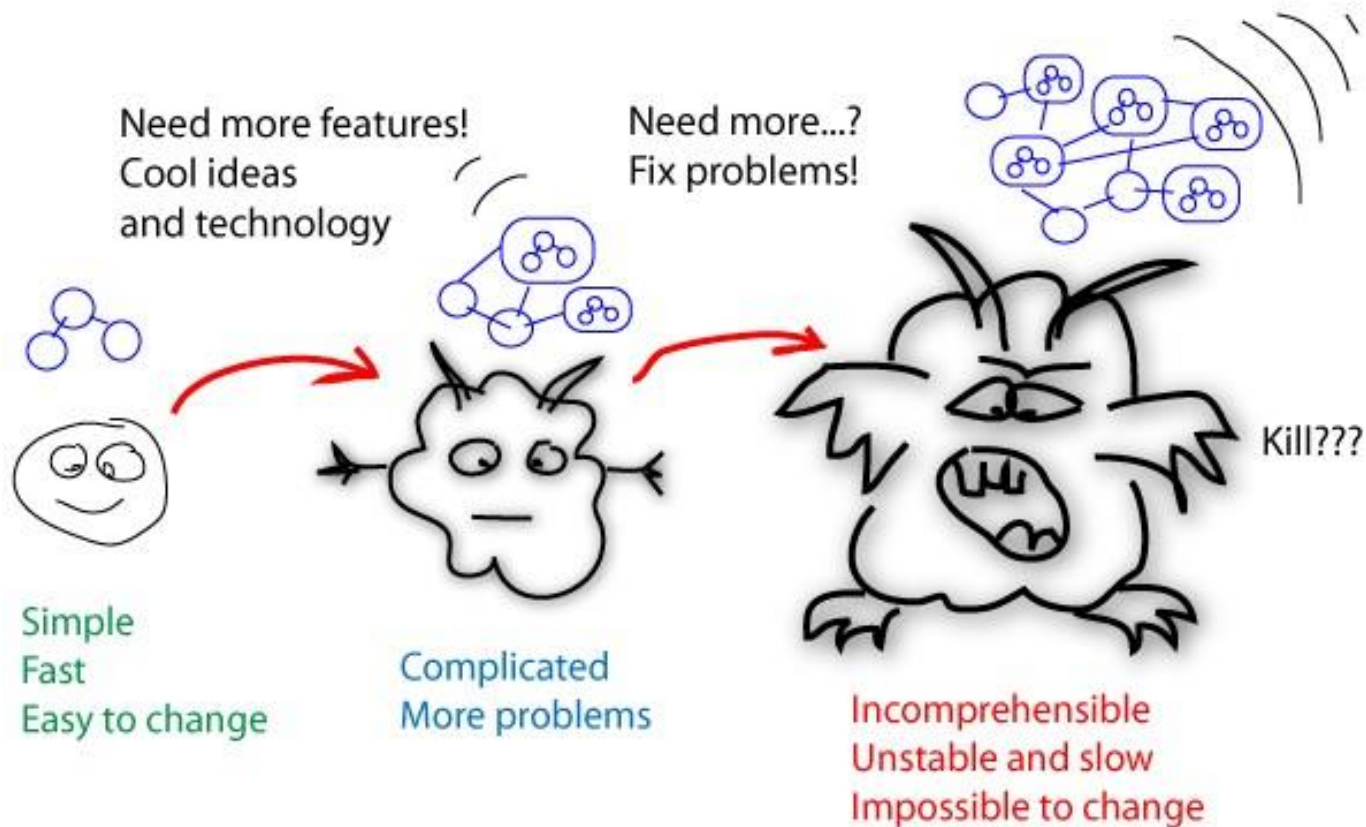
Escolher os frameworks que irão compor o arcabouço de soluções do projeto; usar ferramentas de mapeamento de objetos relacionais (ORM) ou stored procedures; optar por projetos baseados em componentes ou baseados em ações. Todas essas são questões que descrevem o problema da estrutura do projeto de software, pois são todas questões arquiteturais que podem levar o projeto ao sucesso completo ou à ruína (SILVEIRA et al., 2011). A arquitetura deve identificar quais são os principais componentes estruturais do sistema e como será feito o relacionamento entre cada um. Assim, é considerada o elo crítico entre o projeto e a engenharia de requisitos (ENGHOLM JUNIOR, 2017; SOMMERVILLE, 2011). Pode-se dizer que existem dois níveis de abstração da arquitetura de software (SOMMERVILLE, 2011)

# Introdução

- Considerando essas definições, como desenvolver sistemas mais complexos, sem pensar em sua arquitetura?
- Como fica a qualidade desses sistemas?



# Introdução



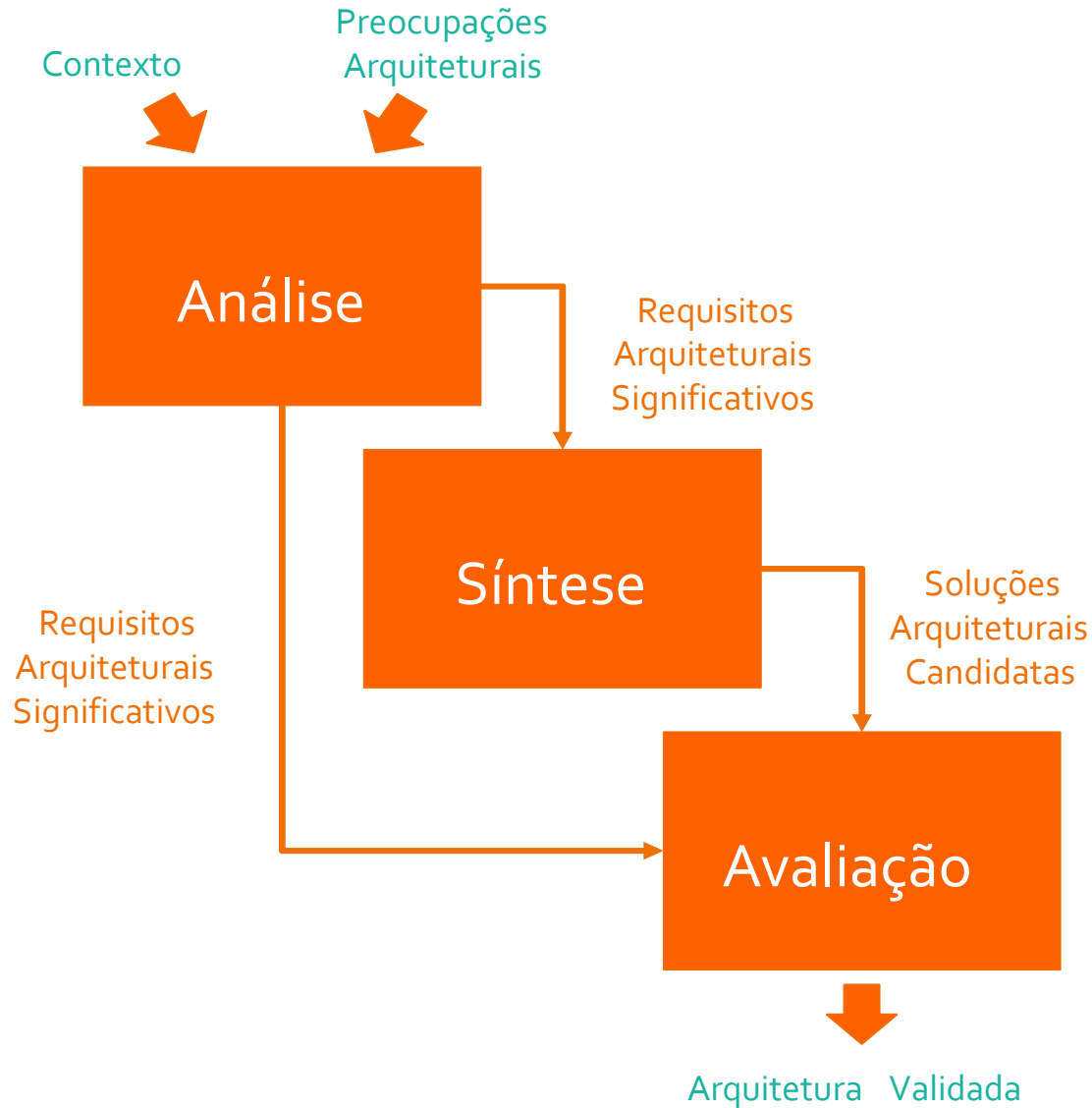
# Introdução

## Desafios

- ☐ Complexidade
- ☐ Diversidade
- ☐ Escopo
- ☐ Tamanho
- ☐ Robustez



# Processo Arquitetural



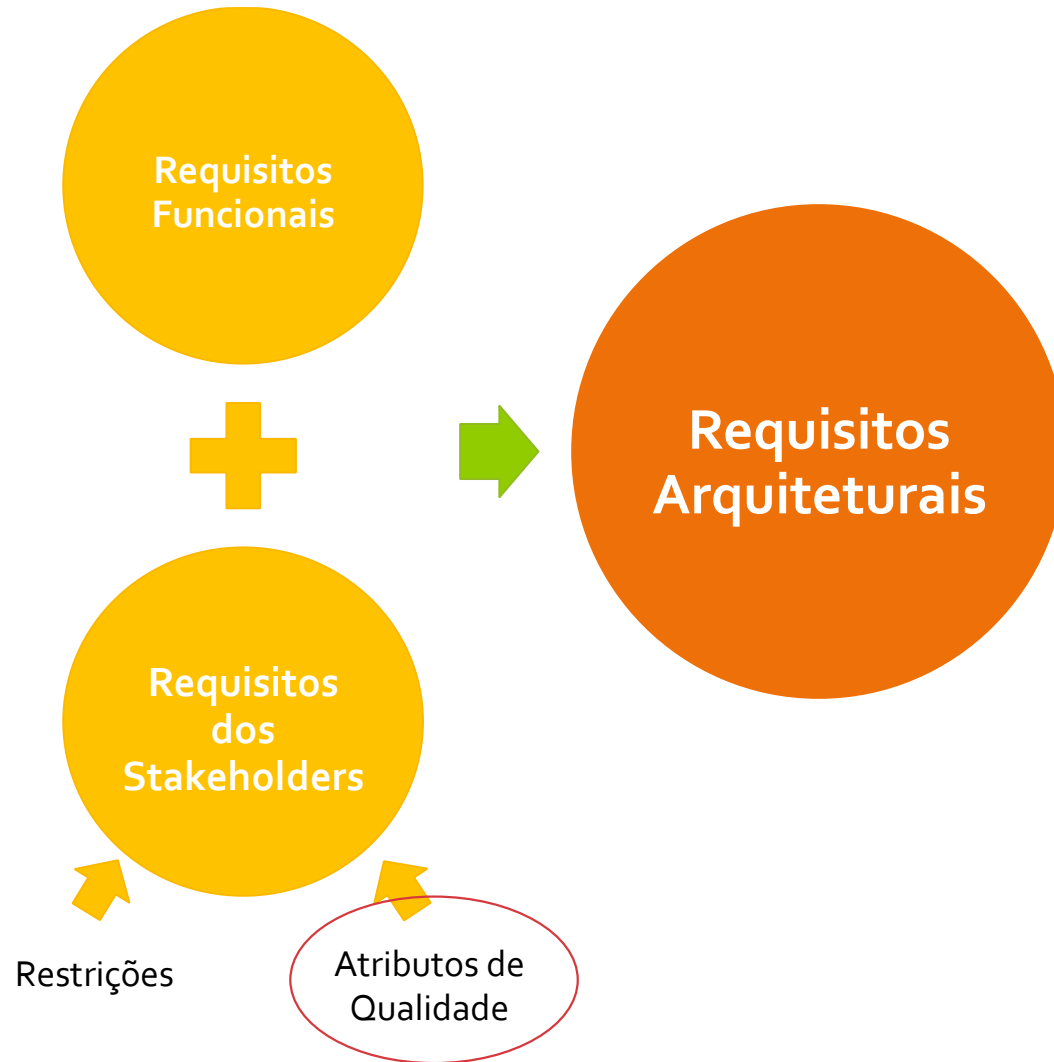
# Processo Arquitetural-Análise

A análise arquitetural articula os requisitos significativos para a arquitetura com base nos problemas que essa arquitetura deverá solucionar e o contexto.

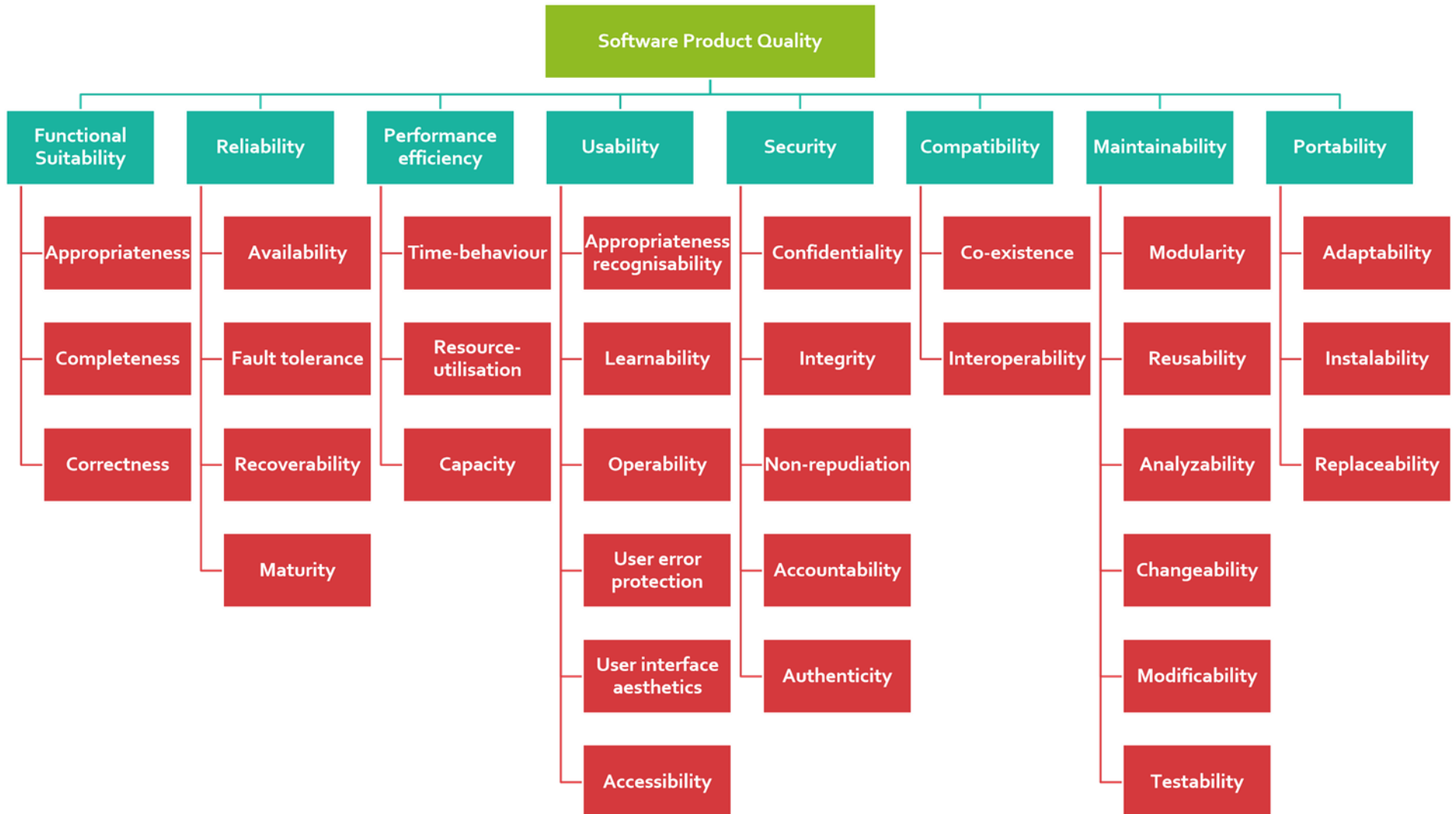
Exemplode requisito:

- “a taxa de erros do Sistema não pode ser maior que 2%” (Desempenho);
- “o tempo de resposta do Sistema deve ser de, no máximo, 2 segundos” (Performance)
- “a perda do pacote de dados deve ser inferior a 0,1% (Confiabilidade);
- “o Sistema deve compartilhar informações com Facebook, Twitter e Instagram) (Compatibilidade).

# Processo Arquitetural-Análise



# Processo Arquitetural



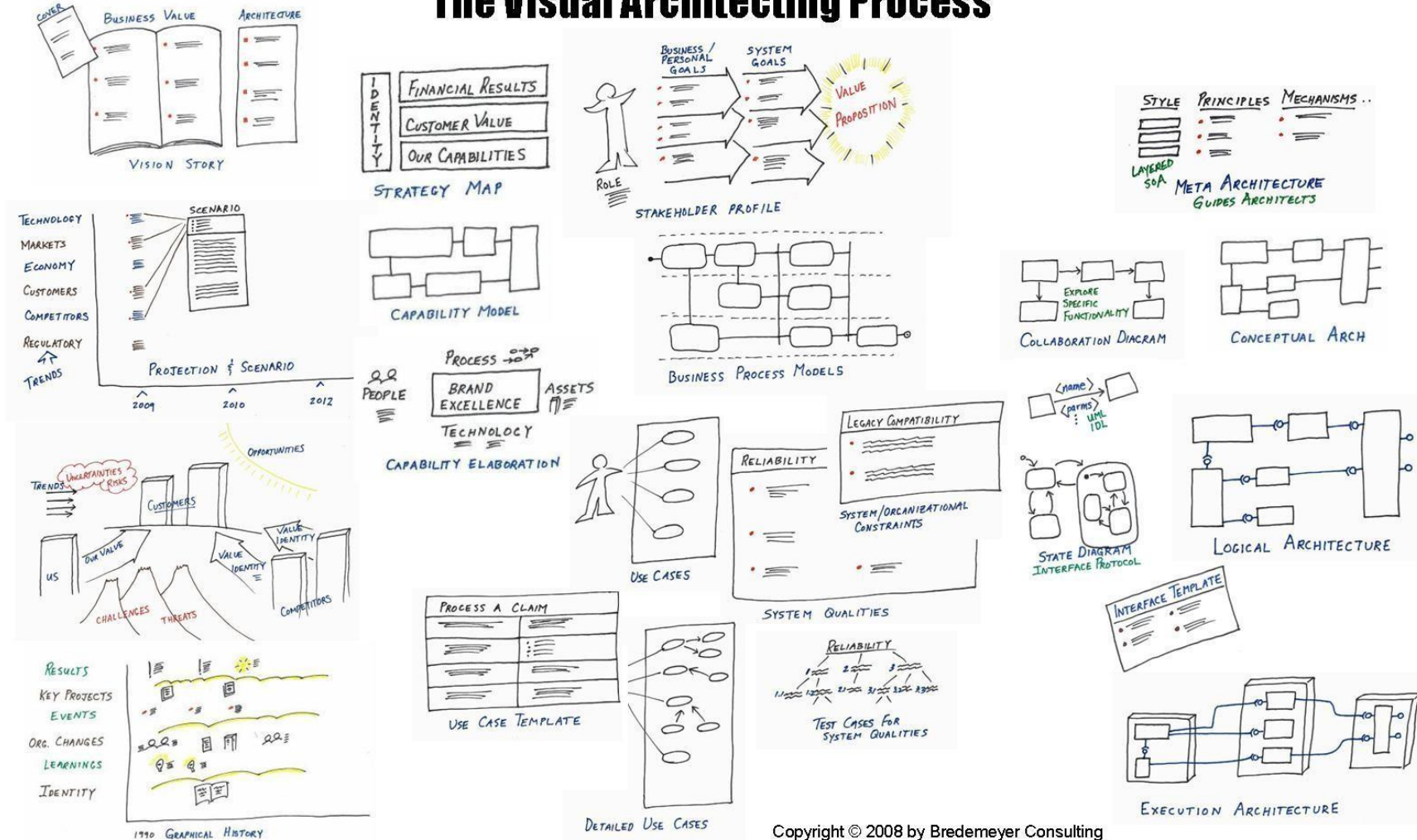
# Processo Arquitetural-Síntese

A síntese arquitetural é a tarefa de encontrar o projeto arquitetônico que atenda às exigências arquitetônicas (uma solução arquitetural candidata).

- ❑ ISO/IEC 42010 – ampla definição conceitual de visões arquiteturais, viewpoints e modelos
- ❑ Decisões arquiteturais– “por que seguimos por este caminho?”
- ❑ PadrõesArquiteturais– ditam uma determinada decomposição modular do Sistema que ajuda a satisfazer os requisitos essenciais
- ❑ Uso de uma Linguagem de Descrição Arquitetural (ADL) – formal ou semi-formal

# Processo Arquitetural-Síntese

## The Visual Architecting Process





# Processo Arquitetural-Avaliação

- ☐ A solução irá atender aos requisitos de qualidade?
- ☐ Temos recursos suficientes para desenvolver a solução?
- ☐ Tomamos as decisões arquiteturais certas? Assegura que as decisões arquitetônicas tomadas sejam as certas e atendem aos requisitos levantados;
- ☐ Avaliações são feitas para identificar potenciais problemas na arquitetura;
- ☐ As arquiteturas de software não são boas ou ruins. São apenas adequadas ou não a um determinado conjunto de metas.

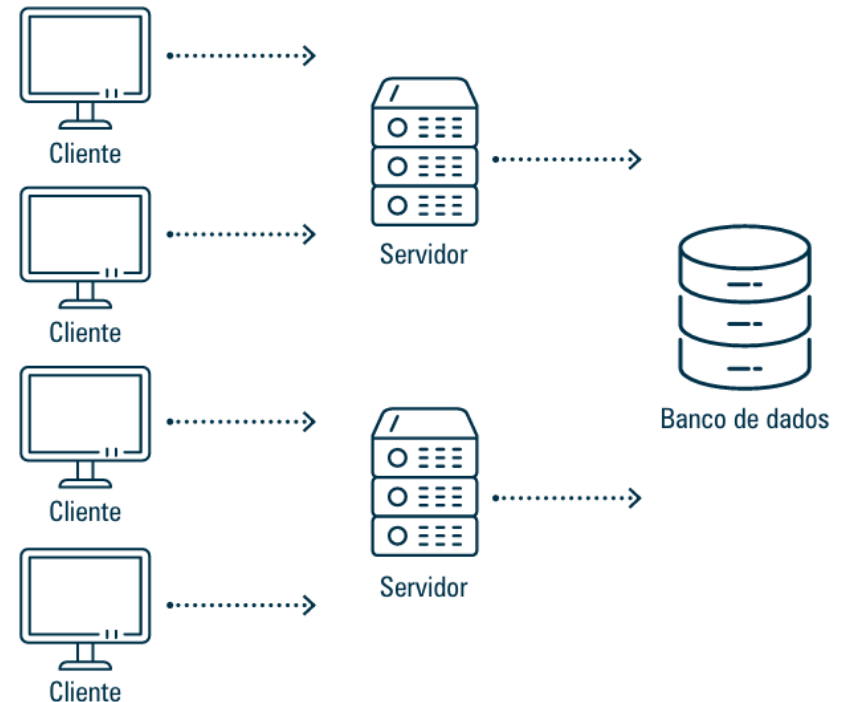
# Arquitetura de Referência

- ❑ Uma arquitetura de referência se diferencia da arquitetura de software por ser mais ampla, ou seja, ela possui um nível alto de abstração e contempla mais atributos de qualidade. Além disso, ela pode envolver, também, a definição e pré-configuração das tecnologias que serão utilizadas nos projetos.
- ❑ Define tipos de elementos e interações permitidas em um domínio particular de aplicações. Ou seja, define como a funcionalidade de um domínio é mapeada em elementos arquiteturais.

# Tipos de Arquitetura de Software

## 1. Arquitetura em Camadas:

- A arquitetura em camadas divide o sistema em camadas com responsabilidades específicas.
- Cada camada é responsável por um conjunto de funcionalidades.
- Proporciona uma separação clara entre as responsabilidades do sistema.
- Facilita a manutenção e a evolução do software.
- Permite a reutilização de código.



## 2. Arquitetura Cliente-Servidor:

- A arquitetura cliente-servidor divide o sistema em dois componentes principais: o cliente e o servidor.
- O cliente é responsável pela interface com o usuário, enquanto o servidor lida com as regras de negócio e o armazenamento de dados.
- Permite uma distribuição eficiente das tarefas entre cliente e servidor.
- Facilita a escalabilidade do sistema.

# Tipos de Arquitetura de Software

## 3. Arquitetura Orientada a Serviços (SOA):

A arquitetura orientada a serviços é baseada na implementação de serviços independentes que podem ser compostos de forma flexível para atender às necessidades do sistema.

Os serviços são componentes autônomos que possuem interfaces bem definidas e podem ser reutilizados por outros sistemas.

Proporciona uma maior modularidade e flexibilidade no desenvolvimento de sistemas.

Permite a integração de sistemas heterogêneos.

É comumente utilizado em sistemas empresariais e integração de aplicações.

## 4. Arquitetura Microservices:

A arquitetura de microservices divide o sistema em pequenos serviços independentes que se comunicam entre si através de APIs.

Cada serviço é responsável por uma funcionalidade específica.

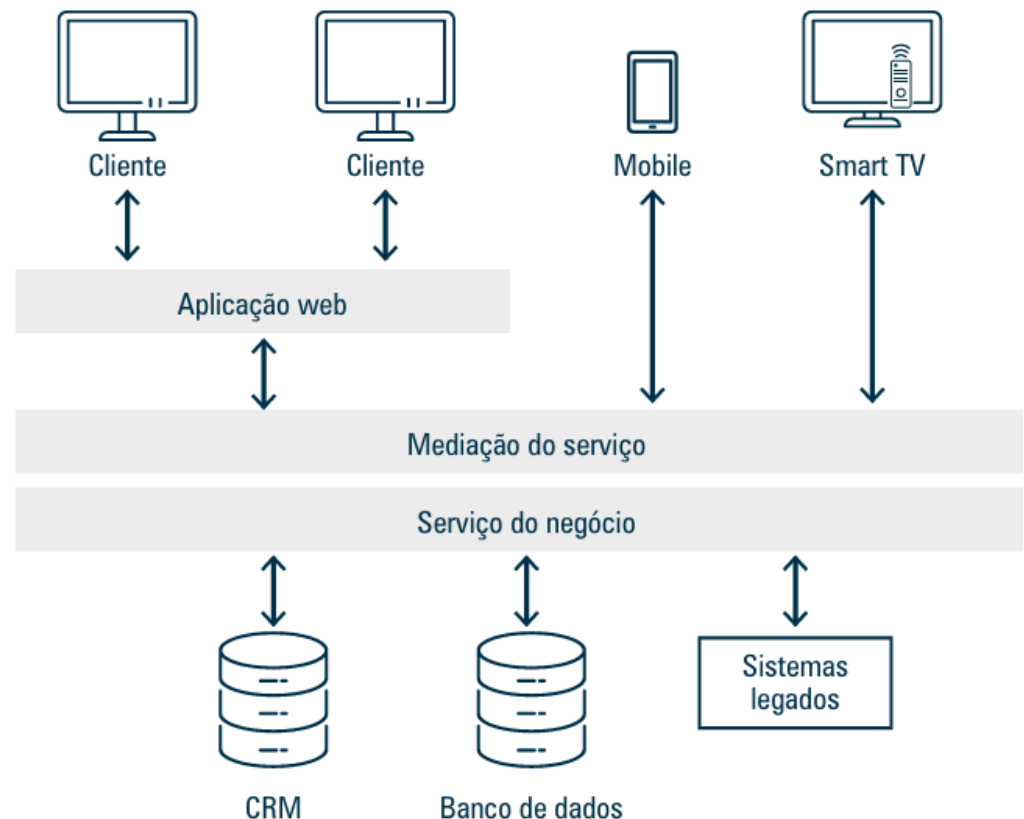
Permite uma maior escalabilidade e disponibilidade do sistema.

Facilita a implantação contínua e a atualização de cada serviço de forma isolada.

É amplamente utilizado em aplicações distribuídas e em nuvem.

A SOA tem como objetivo facilitar a integração entre sistemas, orientando a criação e a disponibilização de soluções modulares e fracamente acopladas, tendo como base o conceito de serviços. Alguns dos benefícios desse tipo de arquitetura são: facilidade de manutenção, re-úso e controle de componentes, flexibilidade, qualidade e menor custo. Na figura 4, temos um exemplo da arquitetura orientada a serviços.

Figura 4 – Exemplo de uma arquitetura de software orientada a serviços

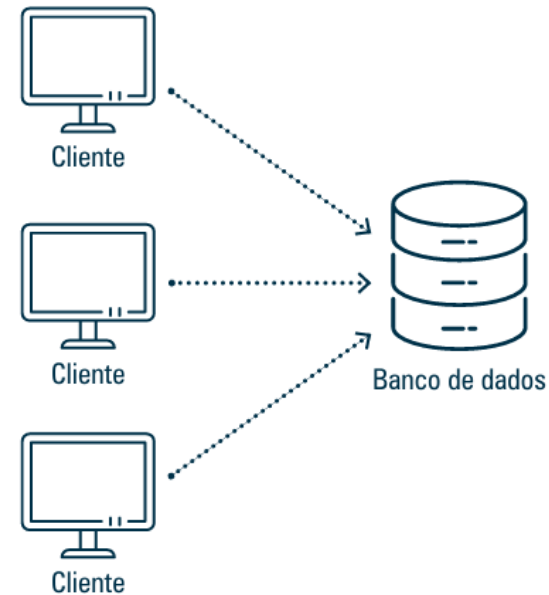


# •Tipos de Arquitetura de Software

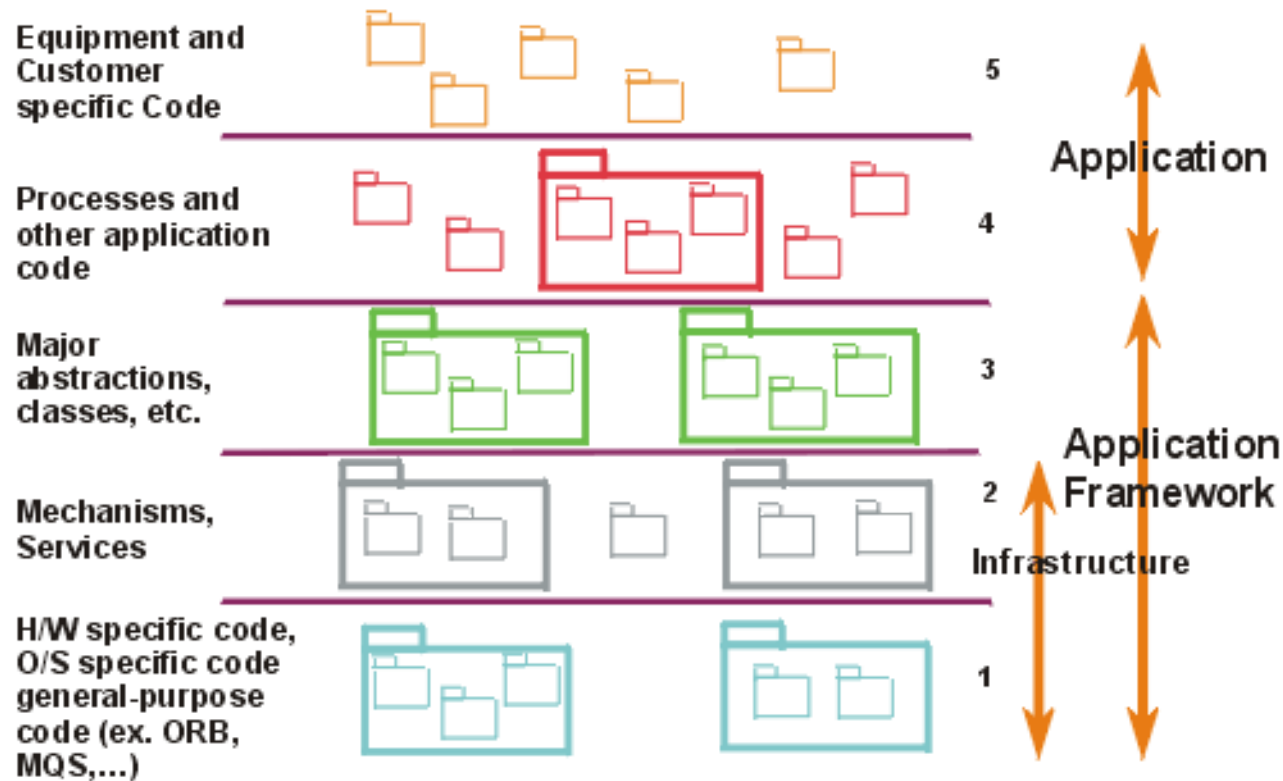
## 5. Arquitetura Monolítica:

- A arquitetura monolítica é caracterizada por ter todas as funcionalidades do sistema em um único código-fonte e em um único processo de execução.
- Facilita o desenvolvimento inicial do sistema.
- Pode ser mais simples de testar e implantar em ambientes de desenvolvimento.
- Pode enfrentar dificuldades de escalabilidade e manutenção em sistemas complexos e em crescimento.
- É comumente utilizado em projetos menores e com equipes pequenas.

· Arquitetura monolítica criada para que os clientes acessem um banco de dados centralizado



# Exemplo de Camadas



Uma arquitetura estritamente em camadas estabelece que os elementos de design (classes, pacotes, subsistemas) utilizem apenas os serviços da camada abaixo deles. Os serviços podem incluir identificação de erros, acesso ao banco de dados e outros. Ela contém mecanismos mais palpáveis, em oposição às chamadas brutas em nível de sistema operacional documentadas na camada inferior.

# Tipos de Arquitetura de Software

## **Arquitetura em Camadas:**

### •Vantagens:

- Separação clara de responsabilidades.
- Facilidade de manutenção e evolução do sistema.
- Reutilização de código.

### •Desvantagens:

- Comunicação entre as camadas pode ser complexa.
- Dificuldade em lidar com requisitos que não se encaixam bem na separação em camadas.

## **Arquitetura Cliente-Servidor:**

### •Vantagens:

- Distribuição eficiente de tarefas entre cliente e servidor.
- Facilidade de escalabilidade do sistema.
- Utilização de tecnologias consolidadas.

### •Desvantagens:

- Complexidade na implementação da comunicação entre cliente e servidor.
- Dependência da disponibilidade e desempenho do servidor.

## **Arquitetura Orientada a Serviços (SOA):**

### •Vantagens:

- Maior modularidade e flexibilidade no desenvolvimento de sistemas.
- Possibilidade de reutilização de serviços.
- Integração de sistemas heterogêneos.

### •Desvantagens:

- Complexidade na implementação e gerenciamento dos serviços.
- Necessidade de definir e manter contratos bem definidos entre os serviços.

# Tipos de Arquitetura de Software

## **Arquitetura Microservices:**

### •Vantagens:

- Escalabilidade e disponibilidade do sistema.
- Implantação contínua e atualização de serviços de forma isolada.
- Flexibilidade e autonomia no desenvolvimento de cada serviço.

### •Desvantagens:

- Complexidade na comunicação e coordenação dos microservices.
- Dificuldade em testar e garantir a consistência das transações distribuídas.

## **Arquitetura Monolítica:**

### •Vantagens:

- Desenvolvimento inicial mais simples e rápido.
- Facilidade de testes e implantação em ambientes de desenvolvimento.

### •Desvantagens:

- Dificuldade em escalar e manter sistemas grandes e complexos.
- Restrição no uso de diferentes tecnologias e linguagens de programação.



# **Exercícios-Sala de Breakout-Timebox: 20 à 30 min**

**Dado uma aplicação e-commerce a ser criada do zero, avalie os seguintes requisitos pedidos:**

- **Arquitetura Multiplataforma;**
- **Carga do servidor próxima a 10 mil conexões simultâneas;**
- **Possibilidade de crescimento limitada;**
- **Suscetível a falhas e sobrecargas;**
- **Tempo de resposta compatível, com os demais aplicativos análogos;**
- **Garantia de estabilidade em acessos simultâneos.**

**Pede-se:**

1. **Quais os requisitos de Qualidade importantes, que devem ser considerados, de acordo com o escopo acima?**
2. **Qual melhor tipo de Arquitetura a ser pensado? Justifique.**

**Ao final cada sala de breakout, voltará para a sala principal e o grupo terá 5-10 minutos para apresentar sua proposta**