

# Arquitetura de Software

- O termo arquitetura de software é usado para designar processo e produto.
  - Produto – representação da estrutura de software
  - Área da engenharia de software que trata de produzir as estruturas de software, visando a reduzir complexidade.

# O Processo de Arquitetura de Software

- Elaboração do modelo de negócio – envolve analisar custo, tempo de desenvolvimento, restrições de mercado, interfaces com outros sistemas, etc
- Entendimento dos requisitos: levantamento de requisitos e modelo do domínio
- Criação ou seleção de uma arquitetura: identificação dos componentes e suas interações, das dependências de construção e tecnologias que apoiam a implementação.

# O Processo de Arquitetura de Software

- Representação da arquitetura e divulgação: para permitir aos desenvolvedores e testadores o entendimento da arquitetura
- Implementação da arquitetura, seguindo seus protocolos e estruturas.
- Análise e avaliação: verificar a adequação da arquitetura, registrando impactos, riscos e dificuldades, o que servirá para evolução da arquitetura.

# O Processo de Arquitetura de Software - Stakeholders

## Participantes (interessados)

- Analista de requisitos – identifica os requisitos
- Arquiteto de software – cria a arquitetura - pode ser um time com um arquiteto líder.
- Projetista ou Desenvolvedor – implementa os componentes

Outros: cliente/usuário, testador, gerente de projeto, programador, secretários, etc.

# O Arquiteto de Software

- ser capaz de reconhecer estruturas comuns em sistemas já desenvolvidos
- usar o conhecimento sobre arquiteturas existentes para tomar decisões de projeto em novos sistemas
- ser capaz de realizar uma descrição formal da arquitetura de um sistema a fim de analisar as propriedades do sistema
- apresentar a arquitetura para outras pessoas

# O Arquiteto de Software

## Habilidades:

- compreender profundamente o domínio e as tecnologias pertinentes
- dominar técnicas de modelagem e metodologias de desenvolvimento
- entender as estratégias de negócios da instituição onde atua
- conhecer produtos, processos e estratégias de concorrentes

# O Arquiteto de Software

Tarefas:

- Especificação da arquitetura do software e das bases para o sistema de acordo com os requisitos do cliente.
- modelagem
- análise de trade-offs e viabilidade
- prototipação, simulação e realização de experimentos
- análise de tendências tecnológicas
- atuação como mentor de arquitetos novatos

# Importância da Arquitetura

## Como construir uma casa sem a planta?

- Atua como uma estrutura a fim de checar o atendimento aos requisitos do sistema
- Suporte na estimação de custos e gerência da complexidade do sistema
- Suporte ao reúso
- Reduz o intervalo entre especificação e implementação
- Permite considerar alternativas arquitetônicas em estágios iniciais do desenvolvimento
- Reduz riscos associados à construção do software



# Importância da Arquitetura

- A arquitetura – abstração que serve como base para criar um entendimento mútuo, para comunicação entre os participantes.
- Sua representação serve como guia para o projeto de sua implementação, teste e implantação do sistema.

# A Definição da Arquitetura

- Para o arquiteto, a fase de engenharia de requisitos é subsídio para a definição da arquitetura da qual se obtém:
  - processo de negócio modelado
  - planejamento estratégico das versões
  - requisitos de cada versão, etc.

# A Definição da Arquitetura

- *Deve facilitar reúso em diferentes níveis.*

É necessário que o sistema possa sofrer alterações de forma localizada, sem afetar outras partes.

É necessário que novas funcionalidades sejam adicionadas sem causar impacto nas já existentes.

A vida útil do sistema depende de uma boa arquitetura que facilite modificações, e permita sua evolução.

# A Definição da Arquitetura

A definição está baseada na escolha de alternativas mais adequadas ao domínio da aplicação.

É importante reutilizar e adotar estratégias previamente validadas.

Utilizam-se frameworks, estilos, padrões e linguagens de descrição de componentes, previamente definidos.

Arquitetura: frameworks, padrões  
e estilos arquiteturais.

# Conceito de Padrão

- Um template (formulário) de solução para um problema recorrente que seja comprovadamente útil em um determinado contexto.
- Um padrão de software é instanciado através da vinculação de valores a seus parâmetros.
- Os padrões podem existir em várias escalas e níveis de abstração; por exemplo, como *padrões de arquitetura*, *padrões de análise*, *padrões de projeto*, *padrões de teste* e *idiomas* ou *padrões de implementação*.

# Definição

- “um padrão expressa uma solução reutilizável descrita através de três partes: um **contexto**, um **problema** e uma **solução**”. (GAMMA et al., 1995).
- **Contexto**: estende o problema a ser solucionado, apresentando situações de ocorrência desses problemas.
- **Problema**: determinado por um sistema de forças, onde estas forças estabelecem os aspectos do problema que devem ser considerados.
- **Solução**: mostra como resolver o problema recorrente e como balancear as forças associadas a ele.

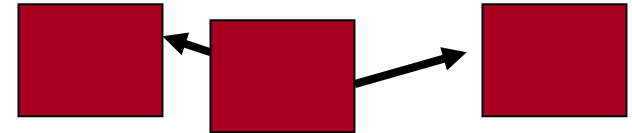
# Padrões em ES

- padrões em ES permitem que desenvolvedores possam **recorrer a soluções já existentes** para solucionar problemas que normalmente ocorrem em desenvolvimento de software;
- **Surgimento**: início dos anos 90;
- Padrões capturam experiência existente e comprovada em desenvolvimento de software, ajudando a promover boa prática de projeto.

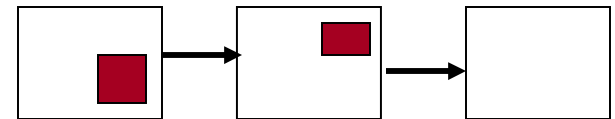


# Categorias de Padrões

- **Padrões Arquiteturais**: expressam um esquema de organização estrutural fundamental para sistemas de software. (BUSCHMANN et al., 1996)



- **Padrões de Projeto**: disponibilizam um esquema para refinamento de subsistemas ou componentes de um sistema de software (GAMMA et al., 1995)



- **Idiomas**: descrevem como implementar aspectos particulares de componentes ou de relacionamentos entre eles, usando as características de uma dada linguagem

```
public void runServer()
{
    ServerSocket server;
    Socket connection;
    try { connection =
        server = new ServerSocket(7000,
        100);
    .....
    server.accept();
    .....
```

# Padrões de arquiteturas (ou arquiteturais)

- São templates prontos que solucionam problemas arquiteturais recorrentes
- Expressam um esquema fundamental de organização estrutural para sistemas de software.
- Fornecem um conjunto de subsistemas pré-definidos, especificando suas responsabilidades e incluindo regras e diretrizes para organizar as relações entre eles.

# Padrões de arquiteturas

Os templates contêm as seguintes informações:

- Nome do padrão
- Contexto
- Problema
  - Impõe a descrição de vários aspectos problemáticos que devem ser considerados
- Solução
  - Fundamentos
  - Contexto resultante
  - Exemplos

# **Nome do Padrão Broker**

## **Contexto**

Ambiente distribuido

## **Problema**

Como os componentes do sistema devem se comunicar entre si.

## **Solução**

Crie um intermediári entre um componente-cliente e um componente servidor, o broker. Um cilente envia uma mensaem parao Broker contendo todas as informações apropiads para que a comunicação seja efetuada. O Broker é responsável por completar a conexão

# Vantagens e Benefícios

- Padrões reduzem a complexidade da solução
- Padrões promovem o reuso
- Padrões facilitam a geração de alternativas
- Padrões facilitam a comunicação

# Estilos de Arquitetura

- Uma arquitetura de software, ou somente uma visão de arquitetura, pode ter um atributo chamado **estilo de arquitetura**, que reduz o conjunto de formulários que podem ser escolhidos e impõe um determinado grau de uniformidade à arquitetura.
- O estilo pode ser definido por um conjunto de padrões, ou pela escolha de componentes ou conectores específicos que funcionarão como os tijolos básicos da construção.

# Estilos de Arquitetura

- Expressam esquemas de organização estrutural de sistemas, fornecendo um conjunto de componentes do sistema, suas responsabilidades e a forma de interação entre eles.
- Cada estilo de arquitetura lida com diferentes tipos de atributos de qualidade. Para obter a definição de uma arquitetura a partir dos estilos existentes, basta saber quais os atributos mais relevantes para a solução e confrontá-los com os atributos que o estilo atende.

# Estilos Arquiteturais

- A arquitetura de um sistema pode aderir a um ou mais **estilos arquiteturais**.
  - Um estilo define os tipos de **elementos** que podem aparecer em uma arquitetura e as regras que regem a **interconexão** entre estes elementos.
- Esses estilos podem simplificar o problema de definição de arquiteturas de sistema.
- A maioria dos sistemas de grande porte adere a vários estilos
- Estilos arquiteturais = “**modelos arquiteturais**”



# Exemplos de Estilos Arquiteturais

- Cliente-Servidor
- Camadas
- Filtros e dutos (*pipes and filters*)
- *Repositório*
- *Orientado a eventos (publisher/subscriber)*
- *Objetos distribuídos, etc*

# Estilos Arquiteturais e Escolhas de Projeto

- Um estilo arquitetural representa um conjunto de **escolhas de projeto**
  - Conjunto de características comuns a diversos sistemas nos quais as mesmas escolhas foram feitas
    - **Padrões arquiteturais**
  - Um sistema aderente a determinado estilo “ganha” as características a ele inerentes
- Estilos podem ser usados para descrever uma determinada arquitetura
  - Foco nas **soluções de projeto** e não em sua **documentação**

# Organização de sistema

- Reflete a estratégia básica que é usada para estruturar um sistema.
- Exemplos:
  - O estilo de repositório de dados compartilhados
  - Estilo de serviços e servidores compartilhados
  - Estilo de máquina virtual ou em camadas
  - Orientado a objetos (ou Objetos Distribuídos)
  - *Pipes and Filters* ou *Pipelining*

# Modelo de referência da Arquitetura

- Consiste na decomposição padronizada do problema em partes conhecidas que cooperam entre si em prol de uma solução.
- Geralmente, estes problemas são de domínio bastante amadurecido e trazem a experiência de analistas de negócio em conjunto com desenvolvedores [Bass98].
- O modelo de referência de um determinado domínio surge durante o processo de amadurecimento da solução em função da necessidade de representações mais abstratas que caracterizam o domínio.

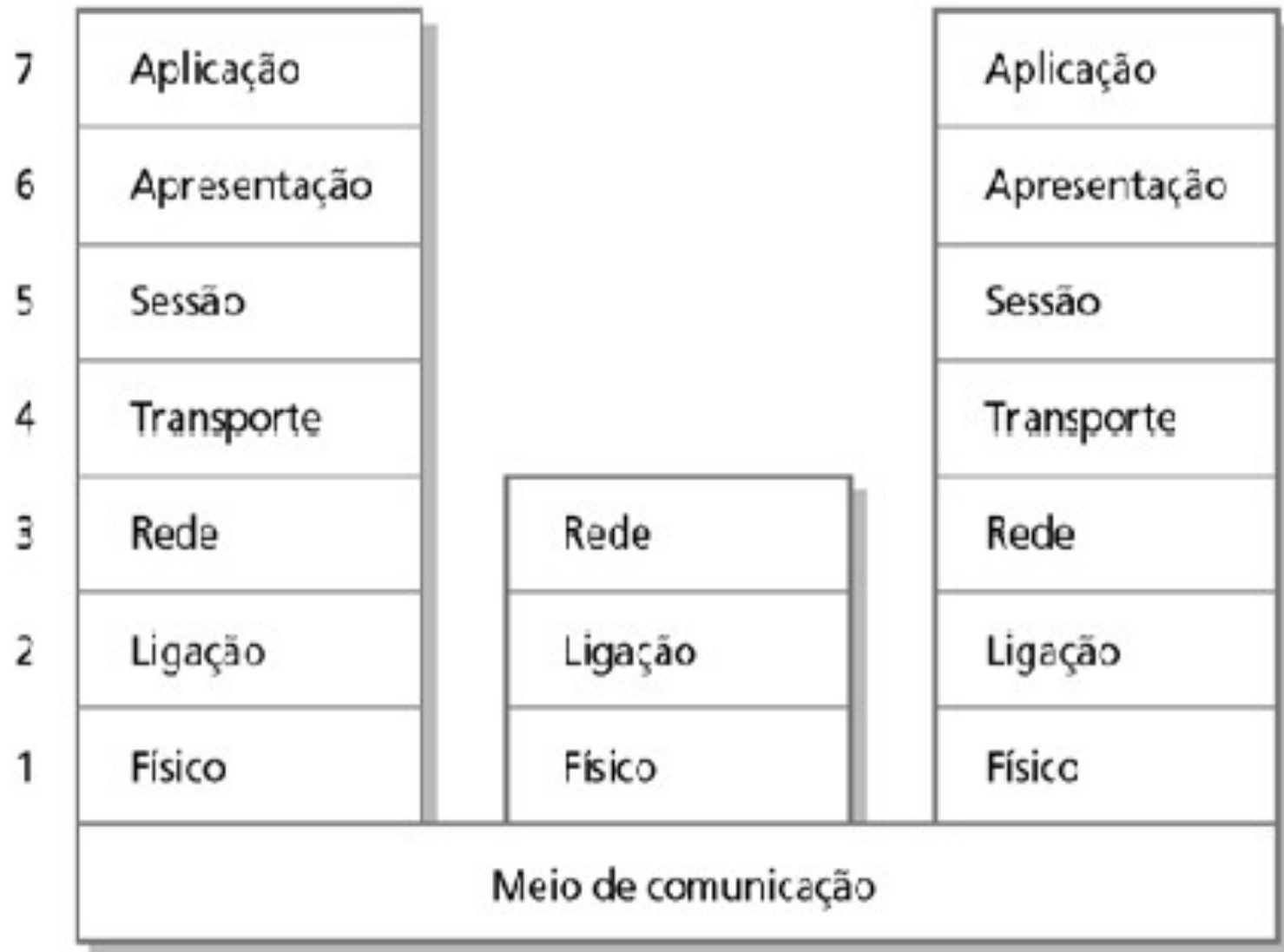
# Arquitetura de Referência

- Consiste em componentes de software e nos relacionamentos entre eles que implementam funcionalidades relativas às partes definidas no modelo de referência.
- Cada uma destas partes pode ser implementada em apenas um ou vários componentes de software, ou seja, o mapeamento das funcionalidades do modelo de referência em componentes da arquitetura de referência nem sempre é um para um.
- As arquiteturas de referência são aplicáveis a um domínio particular.

# Arquiteturas de Referência

- Derivadas de um estudo de domínio de aplicação, ao invés de sistemas existentes.
- Podem ser usadas como base para a implementação de sistemas ou comparação de sistemas diferentes.
  - Atuam como um padrão com relação ao qual os sistemas podem ser avaliados.
- Exs.
  - Modelo OSI para sistemas de comunicação
  - Organização tradicional de compiladores em vanguarda e retaguarda (e seus elementos internos)

# Modelo de referência OSI



# Frameworks de Arquitetura

- Um **framework de arquitetura** ou uma **infra-estrutura de arquitetura** é um conjunto de componentes com os quais pode-se criar um determinado tipo de arquitetura.
- Várias das maiores dificuldades arquiteturais devem ser resolvidas no framework ou na infra-estrutura, geralmente, direcionadas a um domínio específico: comando e controle, sistema de controle, etc.



# Relacionamento entre eles

