

# Arquitetura de Software

Prof<sup>a</sup> Elisabete T. Hato

# Definição

- Arquitetura de Software pode ser definido como uma estrutura dos componentes de um programa/sistema, seus inter-relacionamentos, princípios e diretrizes guiando o projeto e evolução ao longo do tempo. (Garlan, 1995)
- É o estudo da organização global dos sistemas de software bem como do relacionamento entre subsistemas e componentes permitindo o reuso dos recursos dos projetos. (Mendes, 2002)

# Contexto

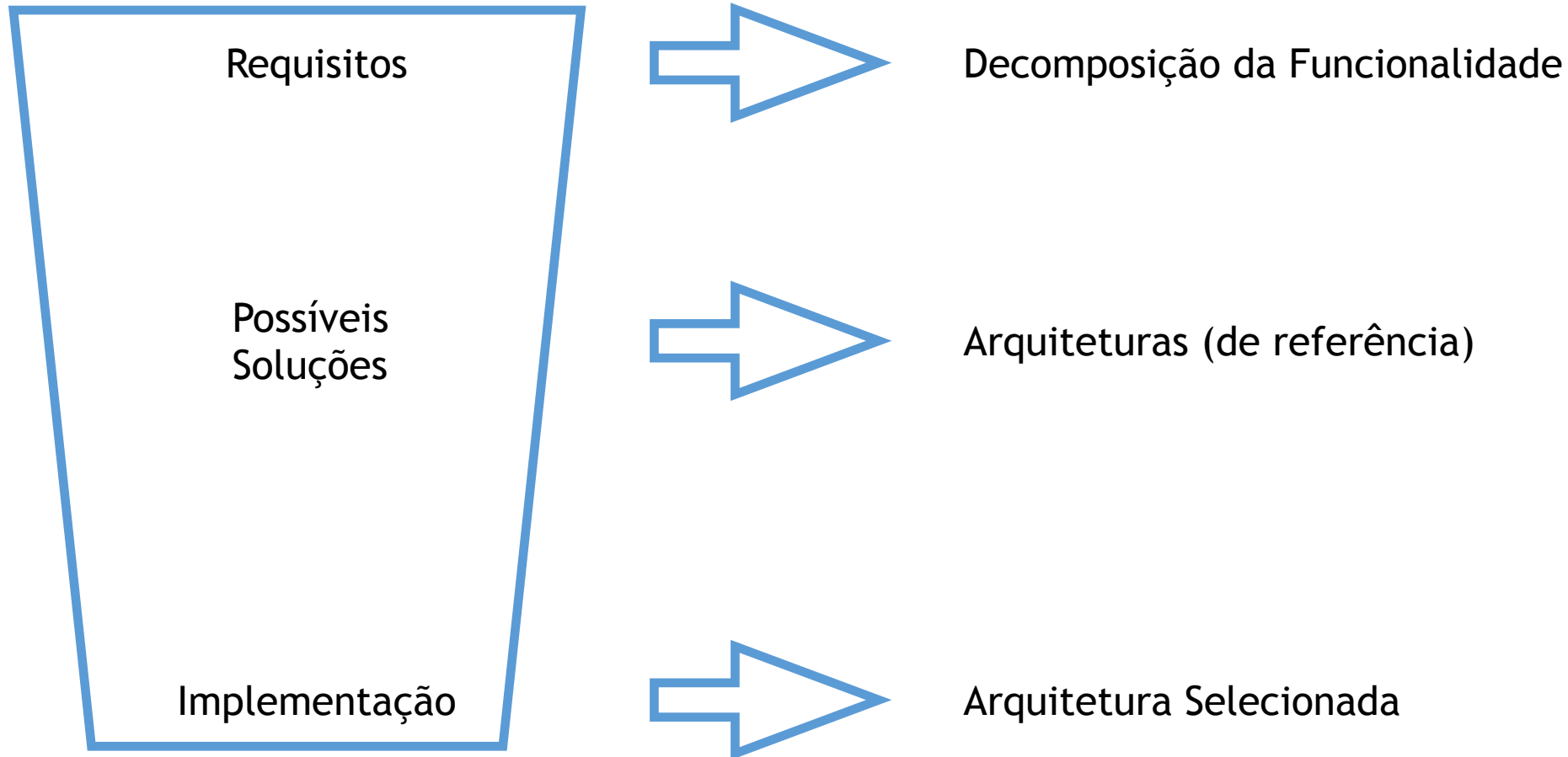
- Quanto maior o tamanho e complexidade de um sistema de software, maior a necessidade de se utilizar arquitetura de software.
- Questões arquiteturais englobam:
  - Organização e estrutura geral de controle
  - Protocolos de comunicação
  - Sincronização
  - Alocação de funcionalidade a componentes
  - Seleção de alternativas de projeto

Abordagem	Foco	Padrões
Programação estruturada	Sistemas de pequeno porte	Estruturas de controle
Abstração e modularização	Sistemas de médio porte	Encapsulamento e ocultação de informações
Componentes e conectores	Sistemas de grande porte	Estilos arquiteturais

# Benefícios

- Atuar como uma estrutura a fim de atender aos requisitos de sistema.
- Ser utilizada como aspecto técnico para o projeto de sistema bem como suporte na estimação de custos e na gerencia do processo.
- Servir de base para análise da consistência e da dependência;
- Prover suporte ao reuso.

# Níveis de Descrição de um Sistema



# Arquiteto de Software

Habilidades desejadas	Tarefas atribuídas
Conhecimento do domínio e tecnologias relevantes	Modelagem
Conhecimento de questões técnicas para desenvolvimento de sistemas	Análise de compromisso e de viabilidade
Conhecimento de técnicas de levantamento de requisitos, e de métodos de modelagem e desenvolvimento de sistemas	Prototipagem, simulação e experimentação
Conhecimento das estratégias de negócios da empresa	Análise de tendências tecnológicas
Conhecimento de processos, estratégias e produtos de empresas concorrentes	Atuação como mentor de novos arquitetos

# Estilos Arquiteturais

- São classes de arquiteturas que possuem aspectos e componentes peculiares a elas, bem como formas de combiná-las.
- Características dos componentes e conectores do sistema, bem como a topologia da arquitetura, restrições semânticas e mecanismos de interação entre esses componentes ajudam a identificar o estilo que trata a arquitetura de software.
- Permite fazer um menor esforço para entender um novo projeto.

# Pipes e Filtros

- Considera a existência de uma rede pela qual flui dados de uma extremidade à outra.
- Tubulação de processamento:

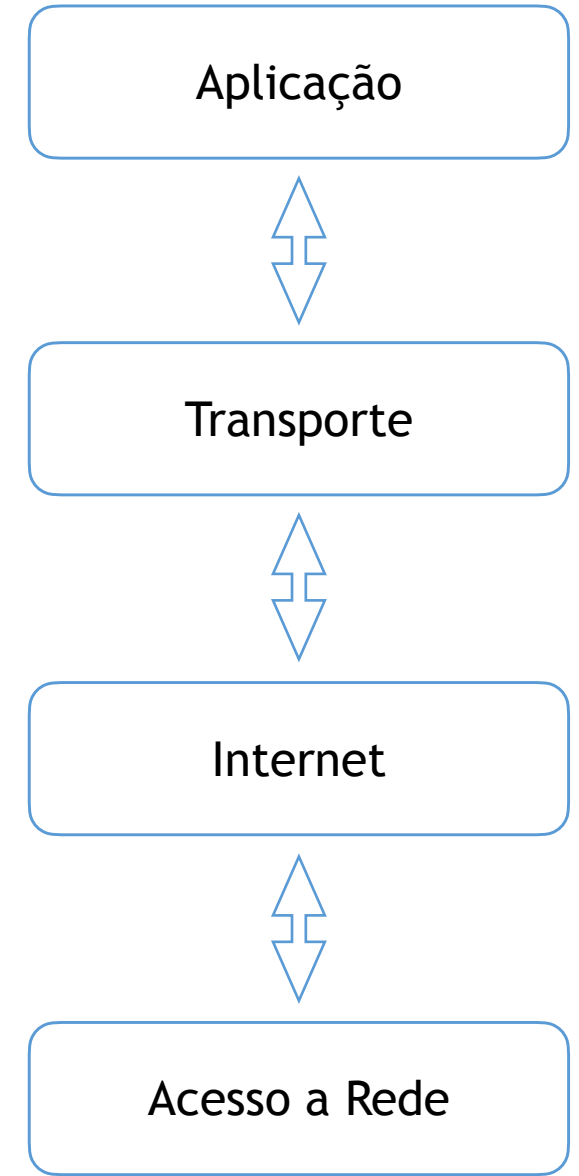


- Recomendado para sistemas que requer vários estágios de processamento.
- V: suporte a manutenibilidade devido a facilidade de reorganização.
- D: Mudanças frequentes em um componente impactam outros componentes.
- Exemplo: Compilador



# Camadas

- Estrutura um sistema num conjunto de camadas, onde cada uma delas agrupa um conjunto de tarefas num determinado nível de abstração.
- Recomendado para componentes complexos que exigirão decomposição do sistema.
- V: Definição de vários níveis de abstração, proporciona uma maior flexibilidade ao sistema.
- D: Comprometimento do desempenho do sistema, visto que é necessário passar por várias camadas.
- Exemplo: Protocolos de rede de computadores



Arquitetura Internet

# Objetos

- Conjunto de objetos independentes com estados próprios e operações associadas a esses estados, sendo que um objeto pode requisitar ou oferecer serviços a outro objeto.
- O Objeto é a combinação em uma única entidade dos dados e das funções que atuam sobre esses dados.
- Recomendado quando existe a necessidade da ocultação de informações.
- V: Melhor compreensão do sistema, devido ao mapeamento entre entidades reais e os objetos e reutilização dos objetos devido à independência dos mesmos.
- D: Desempenho do sistema devido a um maior uso de memória (processamento em cascata).



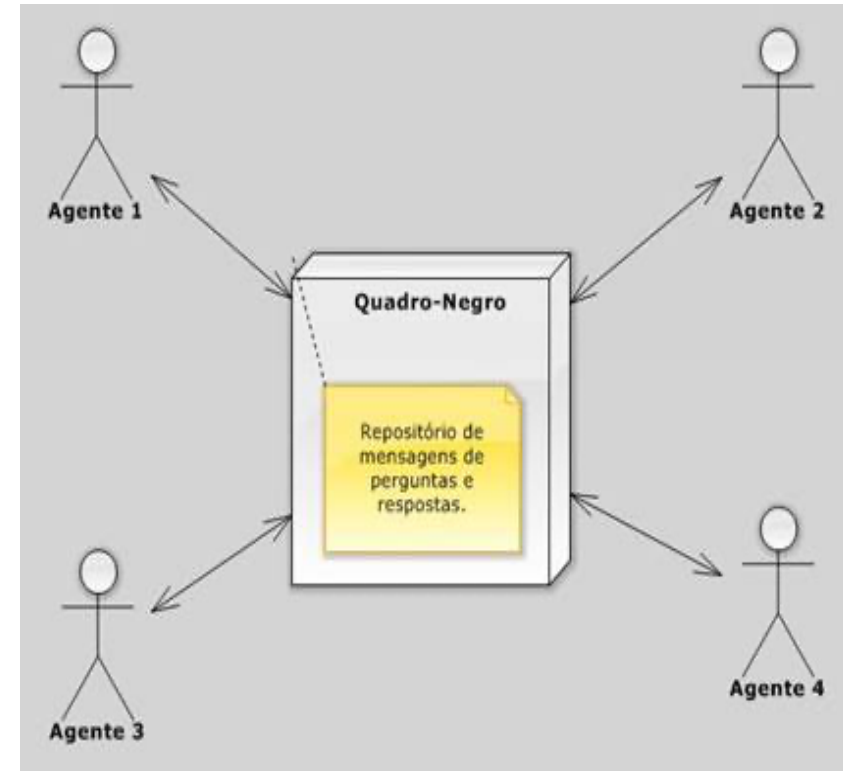
# Eventos (Invocação Implícita)

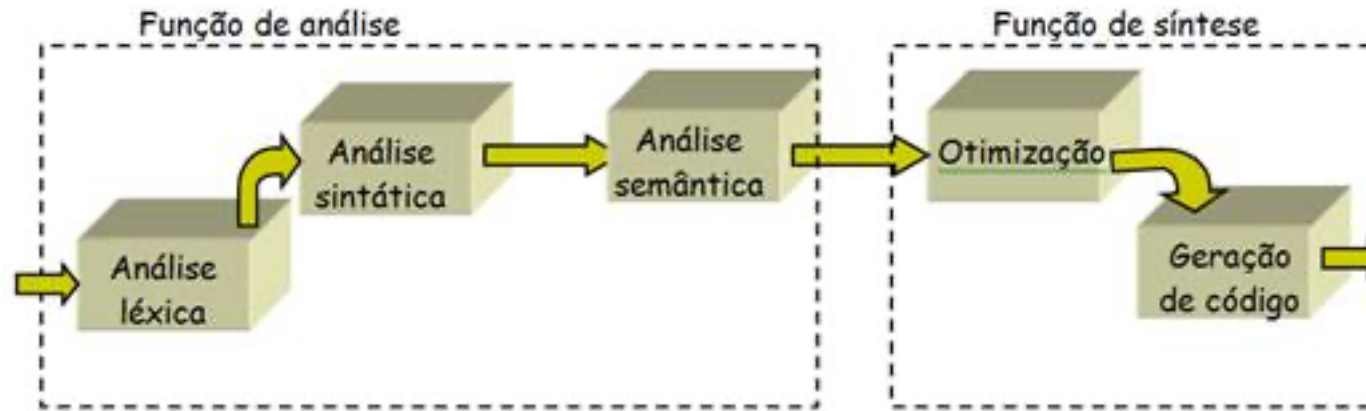
- Requer que os componentes interessados em um evento registrem-se a fim de receber ou divulgar o evento, sendo necessário um mecanismo para tratamento deste. Esse mecanismo desconhece a identidade dos componentes consumidor de eventos.
- Recomendado quando a quantidade de modificações e aperfeiçoamentos aumenta de modo que os componentes do sistema sejam alterados independentes e incrementalmente.
- Exemplo: SGBD
- V: Permite fácil inserção e remoção de componentes.
- D: Os componentes não detêm o controle sobre a computação realizada no sistema.



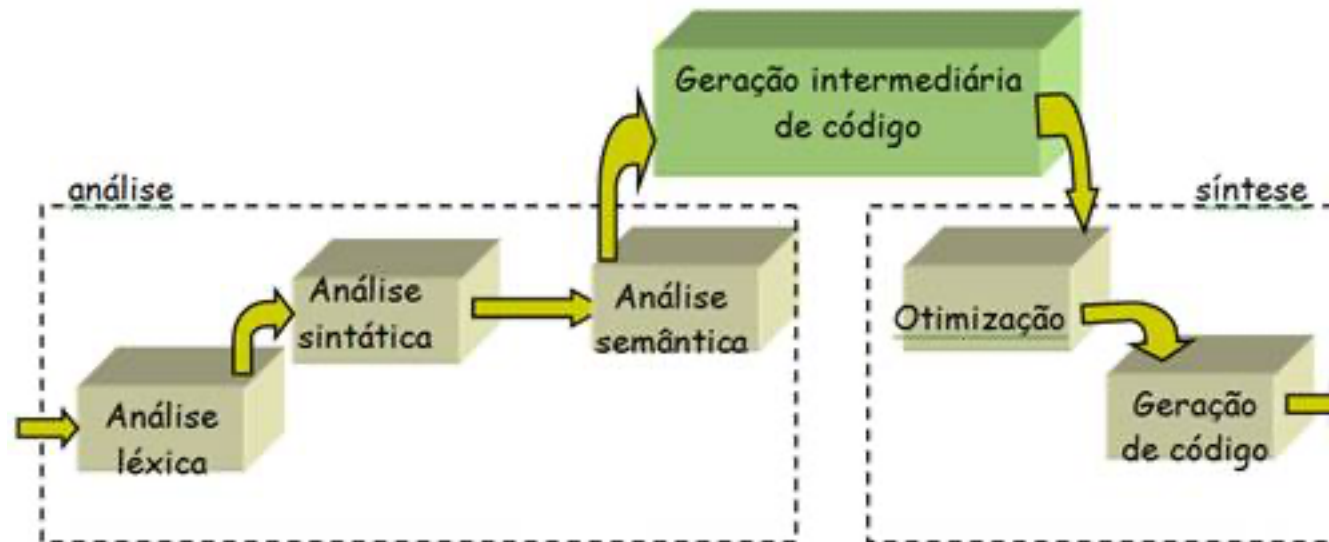
# Quadro-Negro (Blackboard)

- Considera a existência de um repositório central de dados (quadro-negro) circundado por um conjunto de componentes (células de conhecimento).
- Recomendado para aplicações onde diversos tipos de conhecimento devem ser considerados a fim de dar suporte à interpretação de um conjunto de dados iniciais.
- V: Componentes podem ser adicionados e removidos sem acarretar modificações no sistema.
- D: Redundância devido a múltiplos componentes processarem dados de um mesmo problema.

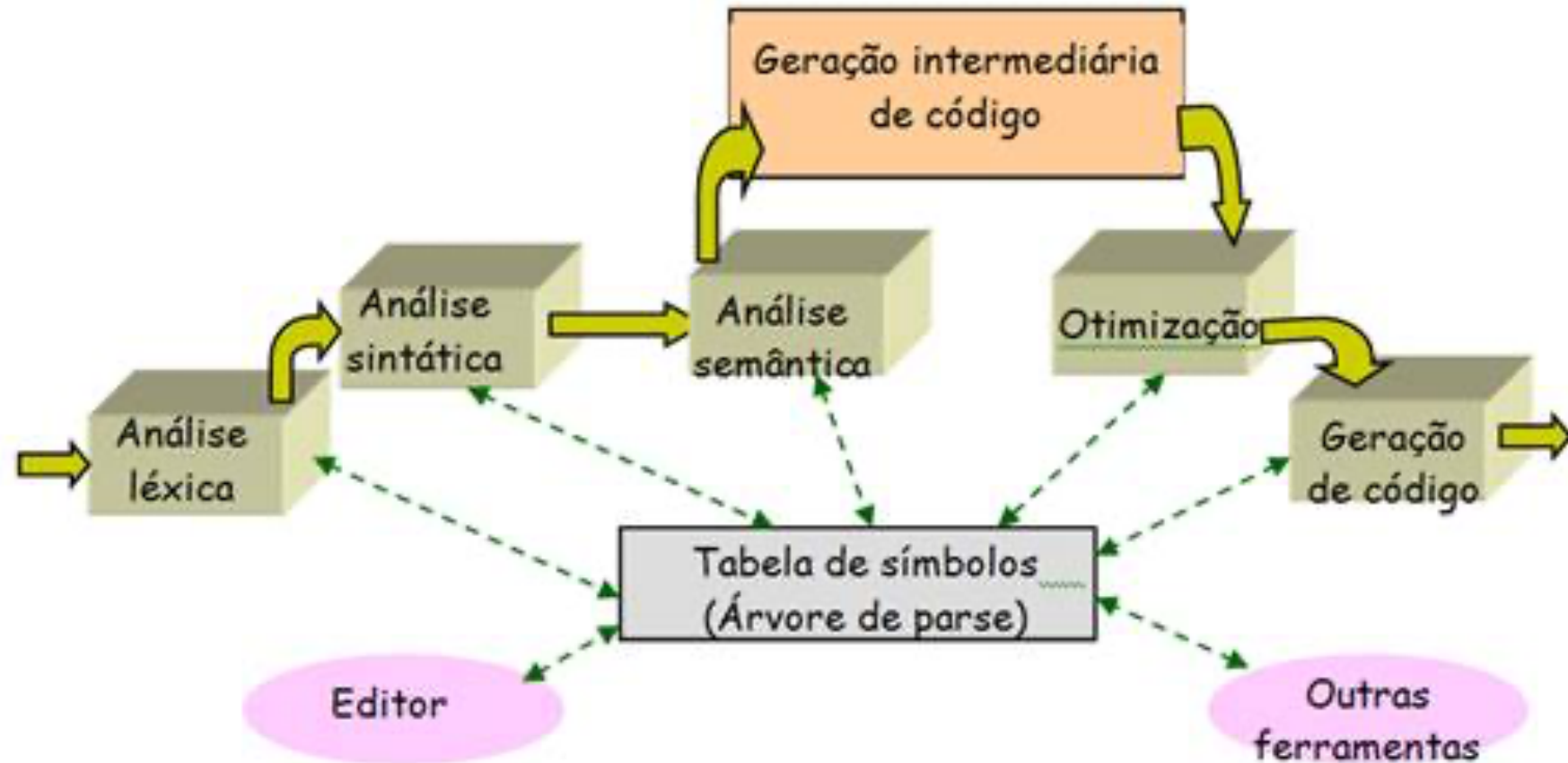




Década de 70 e 80, ocorreu a evolução do hardware. Surgiu a necessidade dos compiladores oferecerem suporte a diferentes plataformas.



Com o avanço tecnológico, o compilador deixou de ser uma ferramenta isolada e foi integrado com outras ferramentas: Editor e Depurador. Devido a esse novo requisito de portabilidade, a arquitetura pipe e filtros não atende mais a especificação do sistema. Qual arquitetura se assemelha a estrutura abaixo?



# Trabalho Individual

- Monte um relatório contendo os seguintes tópicos abaixo para os estilos arquiteturais: Aplicações Distribuídas e SOA:
  - Definição
  - Característica
  - Funcionamento
  - Dois Exemplos