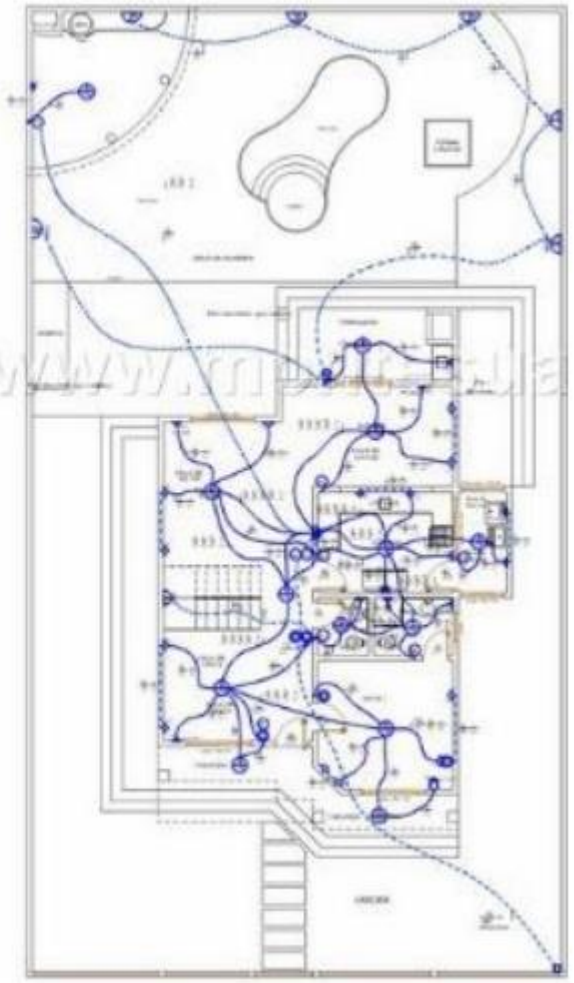
CMP1550

Prof. Fabricio

Projeto arquitetural e desenho de software



Desenho - SW

- Espaço do problema → Espaço da solução.

- Interpretação não literal.
  - Orientação à escrita.
  - Orientação à diagramação.

O que abrange Desenho de Software ?

- Desenho conceitual:
  - "**o que o sistema faz**".
  - Voltado para o usuário.
  - Documento de requisitos, descrevendo as funcionalidades.
- Desenho técnico:
  - "**como o sistema faz**".
  - Voltado para os desenvolvedores.
  - Fluxos de dados, descrevendo os algoritmos

Foco em qualidade

- Independência
  - Modularização.
  - Inputs e outputs de componentes.
  - Medição de independência:
    - Coesão intra-componente.
      - Uso total dos dados internos.
      - Encapsulamento respeitado.
    - Ligação extra-componente.
      - Fortemente ligados.
      - Fracamente ligados.
      - Desligados.

Foco em qualidade – tipos de ligações

Independência

- Ligação de conteúdo.
  - Um componente conhece a estrutura interna/conteúdo de outro.
- Ligação de partilha.
  - Conjunto de dados compartilhados. Ex.: Variável global.
- Ligação de controle.
  - Um componente passa parâmetros que controlam a lógica de outro(s).
- Ligação de estrutura.
  - Uma estrutura de dados fornece a interface entre componentes, gerando dependência.
- Ligação de dados.
  - Tipos de dados simples são passados entre componentes.

Foco em qualidade

- Adaptabilidade.
  - Evolução.
  - Redesenho.
    - Ligações fracas.
    - Abstrações estáveis.
    - Coesão (*self-contained*)

Foco em qualidade

- Inteligibilidade (entendimento).
  - Coesão e ligação dos componentes.
  - Nomenclatura dos componentes.
  - Documentação.
    - Correlação entre o espaço do problema e da solução.

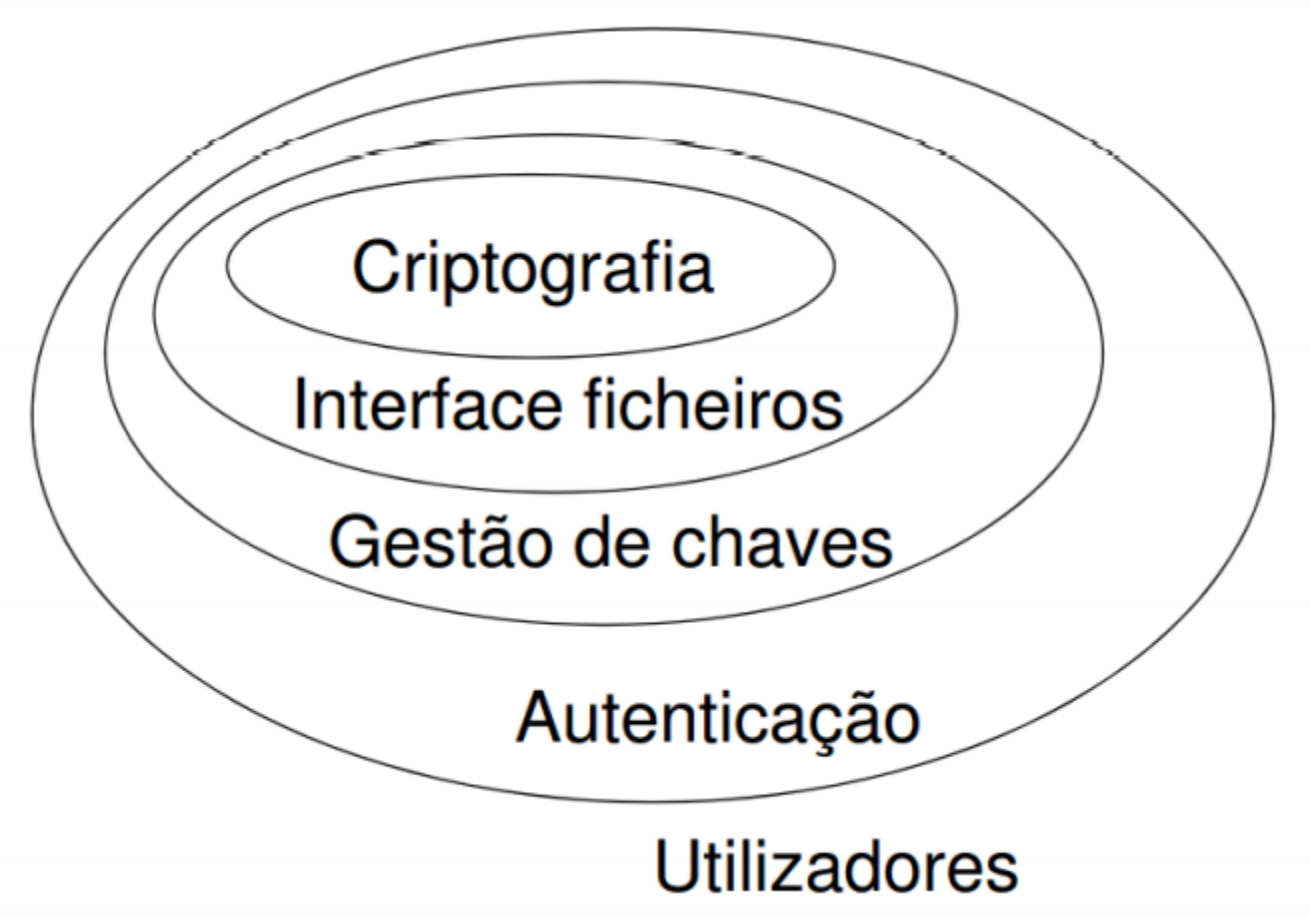
ESTILOS ARQUITETURAIS (desenho)

- Camadas.
- Repositórios.
- Pipes e filtros.
- Desenho funcional.
- Desenho com objetos.

Estilos arquiteturais

- Para a arquitetura de software:
  - Componentes do mesmo.
  - Interação entre estes componentes.
- Estilo arquitetural define o estilo do padrão a ser utilizado no desenho.
  - Descreve os tipos dos componentes.
  - Descreve a complexidade dos componentes.
  - Descreve a orientação da linguagem (escrita, diagramas, etc).
  - Descreve os conectores.
  - Descreve as possibilidades de combinação destes conectores e componentes.

Camadas (estilos do desenho)



Estilos desenho (em camadas)

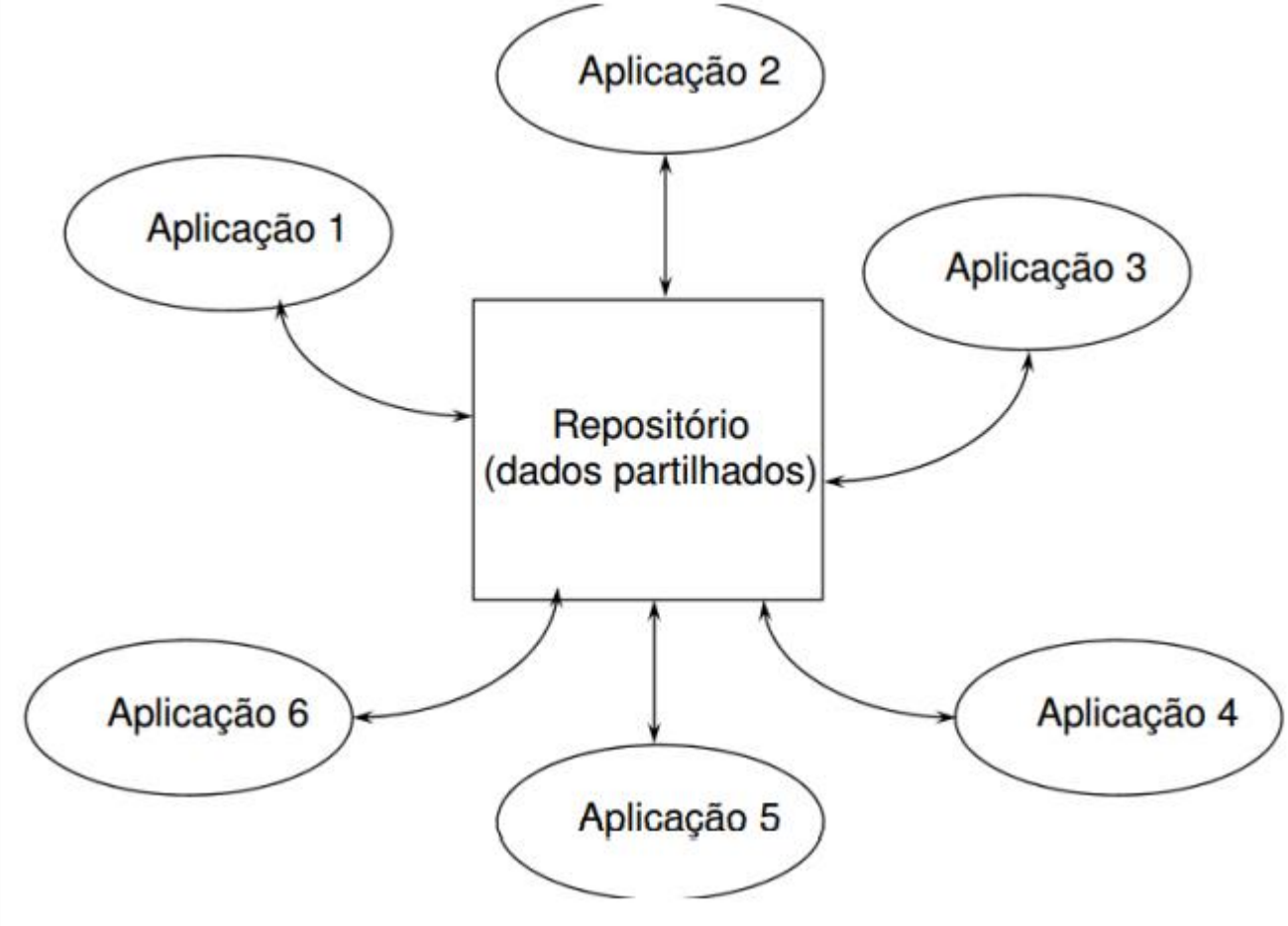
- Hierarquia entre as camadas.
  - ↓ cliente, ↑ fornecedor
- Vantagens:
  - Desenvolvimento incremental.
  - Evolução facilitada.
  - Reutilização.
- Desvantagens
  - Estruturação
  - Desempenho

Estilo arquitetural - repositórios

- Necessidade de troca de dados com frequência.
- Banco de dados central e acessados por todos os subsistemas
- subsistema mantém seu próprio banco de dados
  - abstração de repositório centralizado
  - Implementação distribuída

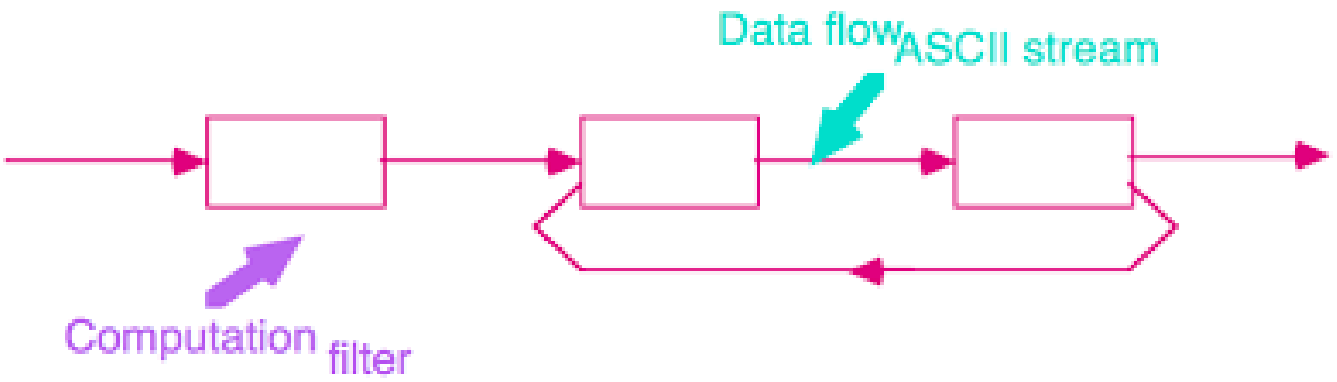
Vantagem: Arquitetura aberta.
Desvantagem: Dependência dos dados

Estilo arquitetural - repositórios



ESTILOS DE DESENHO - PIPES E FILTROS

- Fluxo de dados = pipes.
- Transformação de dados = filtros.
- Vantagem: Controle de inputs e outputs.
- Desvantagem: Detalhamento interno.



Pipe e filtro

Figura 11.6 Modelo de pipeline de um sistema de processamento de faturas.

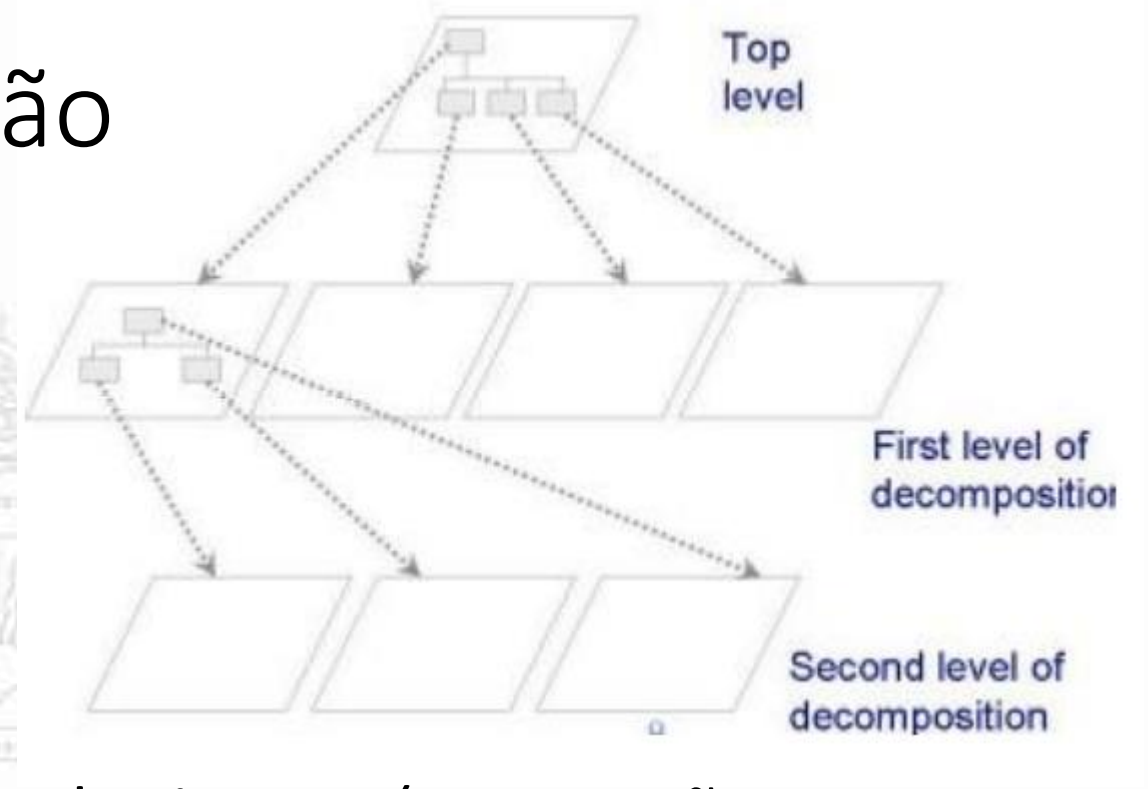
Estilos de Desenho - DESENHO FUNCIONAL

- Decomposição do sistema em um conjunto de funções.
- Estado global partilhado
- Estado local dura enquanto a função estiver em execução.
- Detalhes dos algoritmos nas funções, mas o estado não.
- Alterações em cadeia (estado).

DESENHO COM OBJETOS

- Encapsulamento.
- Abstração.
- Modularidade.
- Primitivas da Orientação à Objetos.
  - Classes, objetos, mensagens, herança, etc.
- Reutilização.
- Extensibilidade.
  - Alterações.

Decomposição



- Descreve os dados de sistema (e como são transitados).
- Descreve funcionalidades (em alto nível).
- Detalha e agrupa informações (hierarquicamente)

Dúvidas

- Analisar (fazendo engenharia reversa) os estilos arquiteturais encontrados.