

Arquitetura de Software

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

[Projeto Arquitetural]

- O projeto é uma atividade criativa
 - Cada arquiteto tem sua própria maneira de projetar o software
- Projeto arquitetural faz a ligação entre
 - Requisitos (domínio do problema)
 - Projeto detalhado (domínio da solução)



[Definindo a Solução]

- Primeiro passo do projeto de sistema
- É frequentemente conduzido em paralelo com atividades de especificação de requisitos
- O projeto arquitetural envolve
 - Identificação dos componentes principais do sistema
 - Definição das interfaces de comunicação entre os componentes



Vantagens da Arquitetura

- Comunicação entre *stakeholders*
 - A arquitetura pode ser usada como foco da discussão sobre o sistema
- Análise de sistema
 - Adequação aos requisitos não funcionais
- Reutilização em larga escala
 - Um componente da arquitetura pode ser reusado em outros sistemas

[Uma arquitetura pode afetar...]

- Desempenho
 - Incluir gargalos de comunicação
- Facilidade de distribuição
 - Sistemas podem executar em várias máquinas
- Facilidade de manutenção
 - Componentes devem ser substituíveis

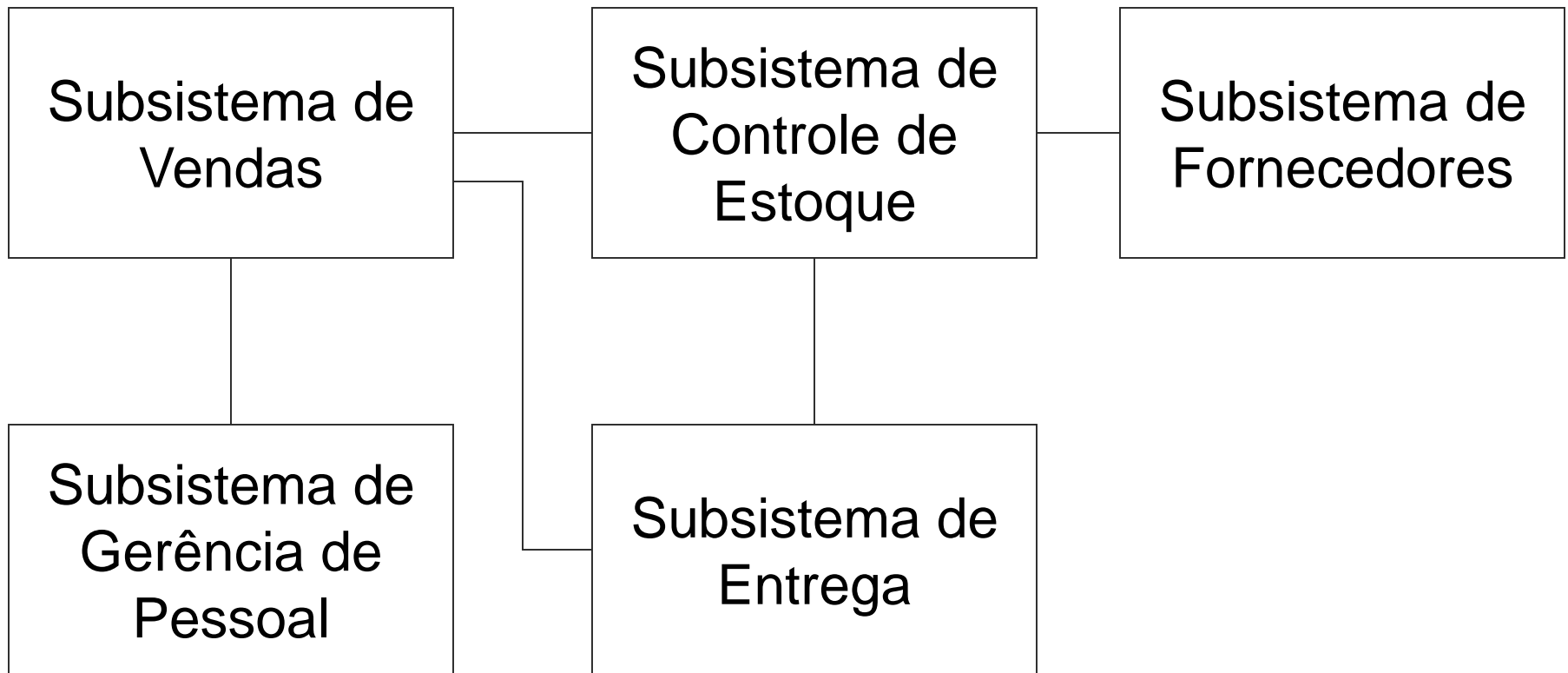




[Diagrama de Componentes]

- Identifica os principais subsistemas que serão desenvolvidos
- Notação simplificada
 - Caixas representam os subsistemas (componentes)
 - Linhas representam alguma comunicação entre os subsistemas
 - Caixas dentro de caixas indicam que um componente foi decomposto

[Exemplo de Arquitetura]



[Quando usar o diagrama?]

■ Pontos positivos



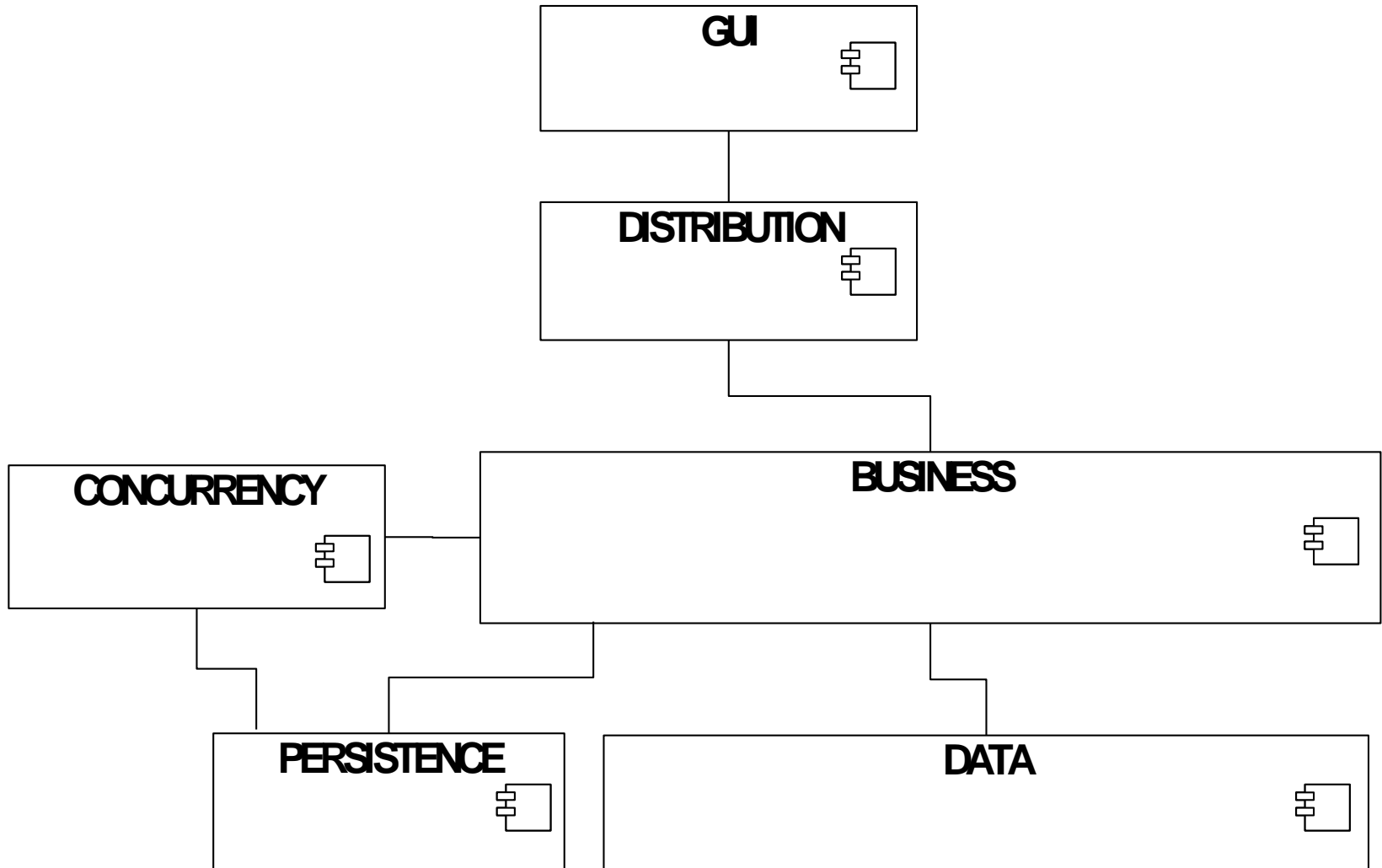
- Facilita comunicação com *stakeholders*
- Úteis para planejamento de projeto
- Facilita o desenvolvimento em paralelo

■ Pontos negativos

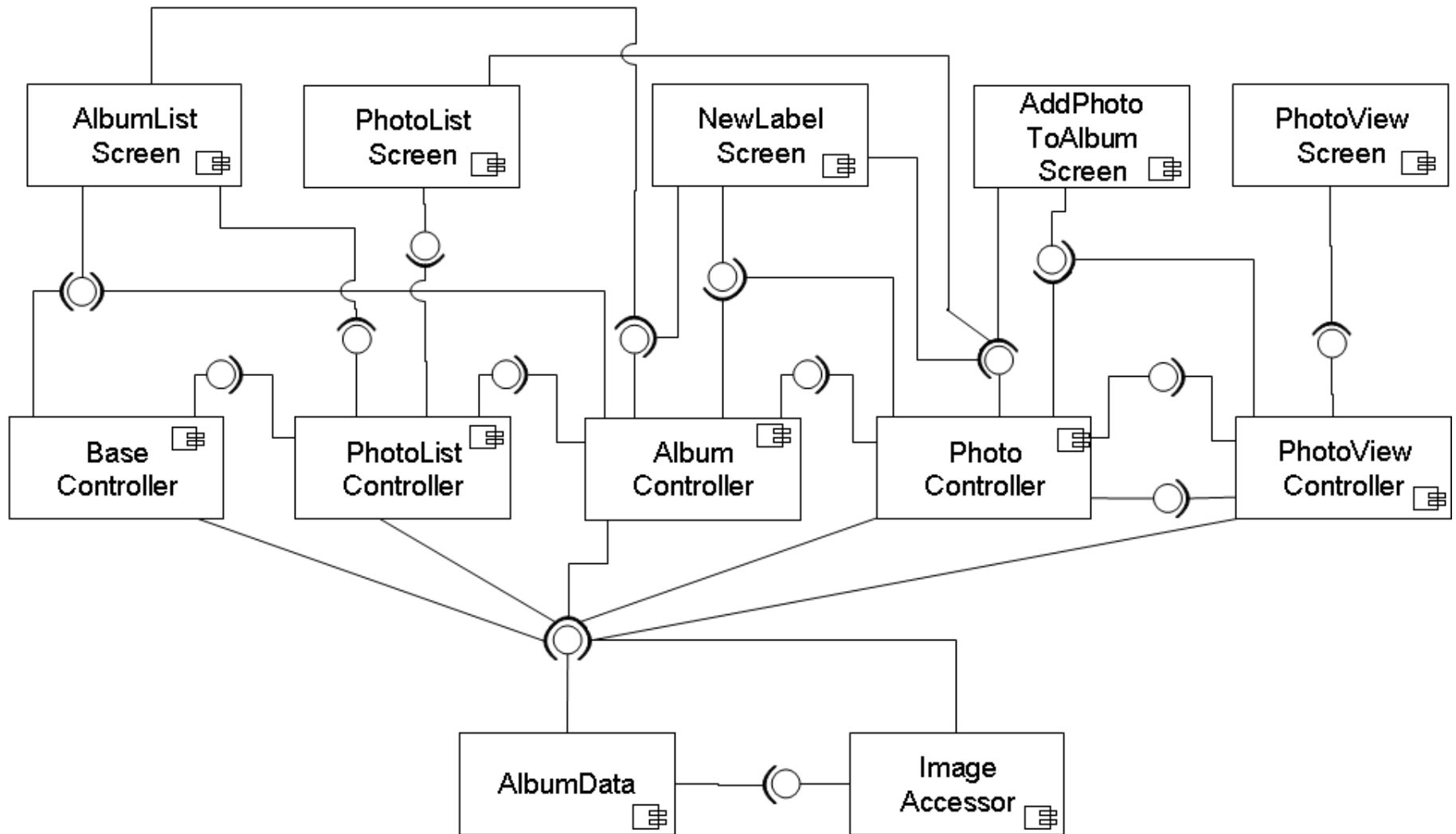


- Não mostra a natureza dos relacionamentos entre componentes
- Não indica as propriedades internas dos subsistemas

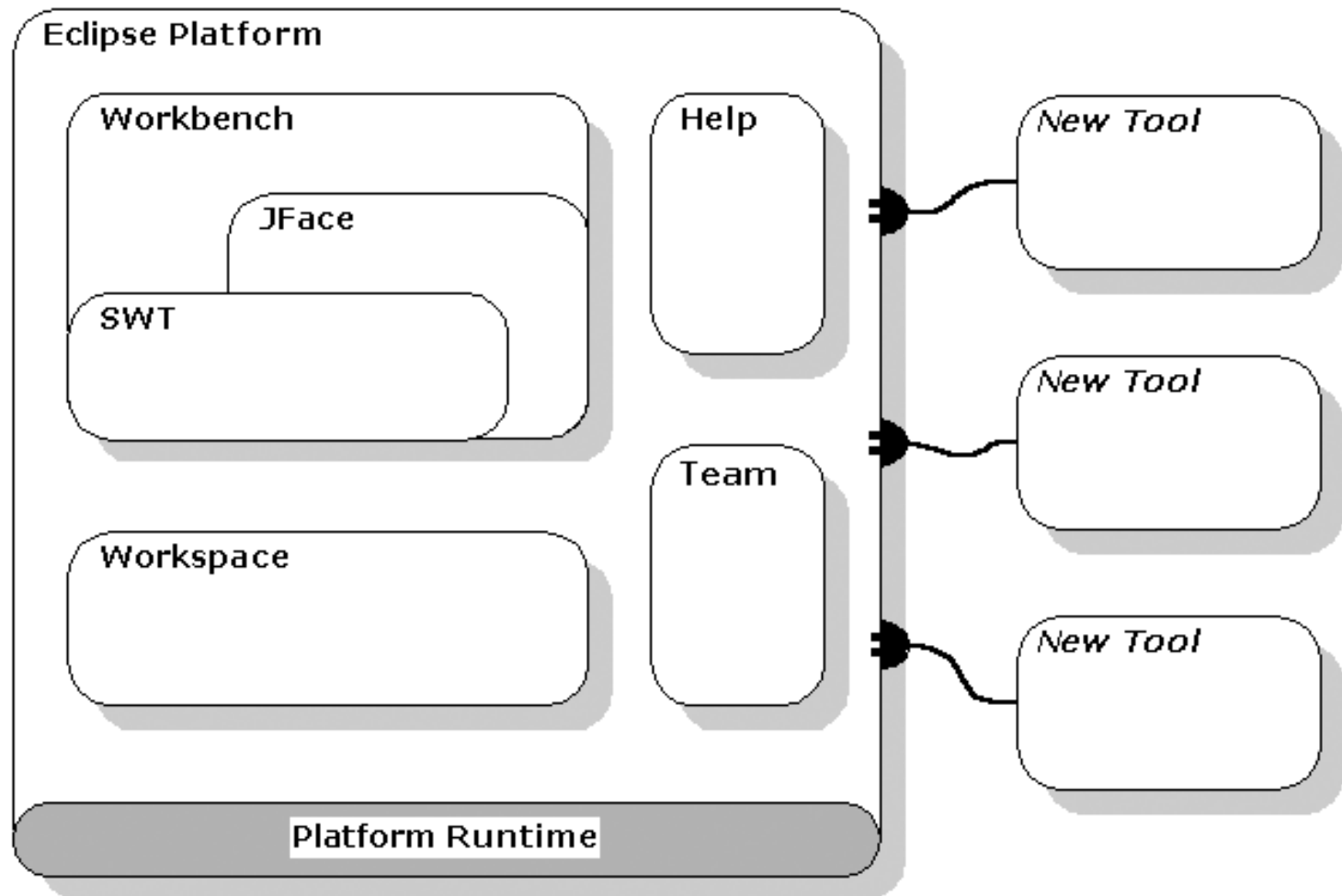
[Exemplo de Arquitetura (2)]



[Exemplo de Arquitetura (3)]



[Exemplo de Arquitetura (4)]



[Decisões de projeto arquitetural]

- Questões precisam ser respondidas
 - Como o sistema será distribuído em diferentes máquinas?
 - Como as funcionalidades serão decompostas em componentes?
 - Como avaliar se a arquitetura atende aos requisitos não funcionais?
 - Existe uma arquitetura genérica que possa ser usada?



[Requisitos Não Funcionais]

- A arquitetura deve considerar requisitos não funcionais
 - **Desempenho:** evitar comunicação excessiva entre componentes distribuídos
 - **Segurança:** ocultar características críticas de segurança em camadas mais internas
 - **Disponibilidade:** incluir componentes redundantes e tolerância à falhas, etc.



[Bibliografia]

- Ian Sommerville. **Engenharia de Software**, 9a. Edição. 2011.
 - Cap. 6 até a Seção 6.2