



Diagrama de Pacotes

Universidade Federal do Maranhão - UFMA

Departamento de Informática

Processo de Desenvolvimento de Software

Prof^a.MSc Simara Rocha

simararocha@gmail.com/simara@deinf.ufma.br

Referências: Booch, G. et al. The Unified Modeling Language User Guide
Medeiros, E. Desenvolvendo Software com UML 2.0: Definitivo, Makron Books, 2006.



Definição

- A UML define o diagrama de pacotes como um modelo que descreve como os elementos são organizados dentro de pacotes e suas dependências.
- PACOTES
 - Como “quebrar” um grande sistema em sub-sistemas pequenos?
 - Pacote (package): agrupar classes em unidades de mais alto nível
 - A idéia de um pacote pode ser aplicada a qualquer elemento do modelo, não somente classes
 - Um “Diagrama de Pacotes” mostra pacotes de classes e as dependências entre eles
 - Relacionamento de Dependência



Definição

- Quando incluir?
 - Ao término da análise do subsistema de caso de uso
 - Ao término de um módulo
 - Para sistemas grandes, talvez grandes áreas, ou talvez você tenha optado por subdividir um grande módulo em outros pequenos.



Definição

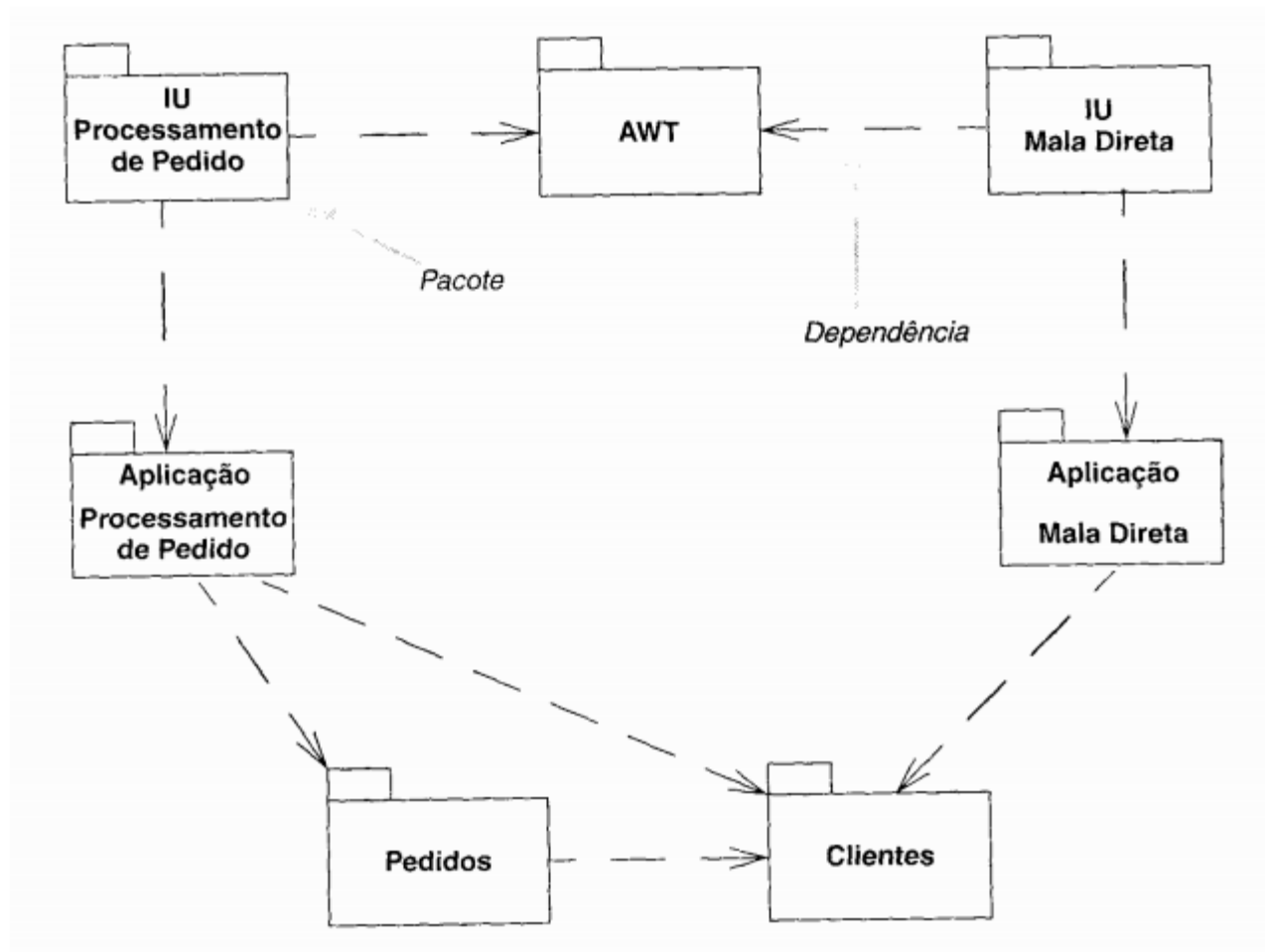
- O quê incluir
 - Classes que estejam em uma mesma árvore de herança
 - Classes que estejam em um mesmo jogo de agregação ou composição
 - Classes que estejam aparecendo em um mesmo diagrama de seqüência com muitas colaborações
 - Pacote de utilitários, contendo classes sem afinidade direta com o domínio do problema, porém são necessárias.
 - As classes estereotipadas, como interface gráfica.



Definição

- Em um diagrama de pacotes, estes são ligados por setas pontilhadas.
- Um pacote pode ter qualquer diagrama da UML, porém são mais comuns em:
 - Diagramas de Casos de Uso, para ajudar a abstração do domínio do problema.
 - Diagramas de Classes, para ajudar na organização das classes construídas em sistemas médios e grandes.

Exemplo



Exemplo

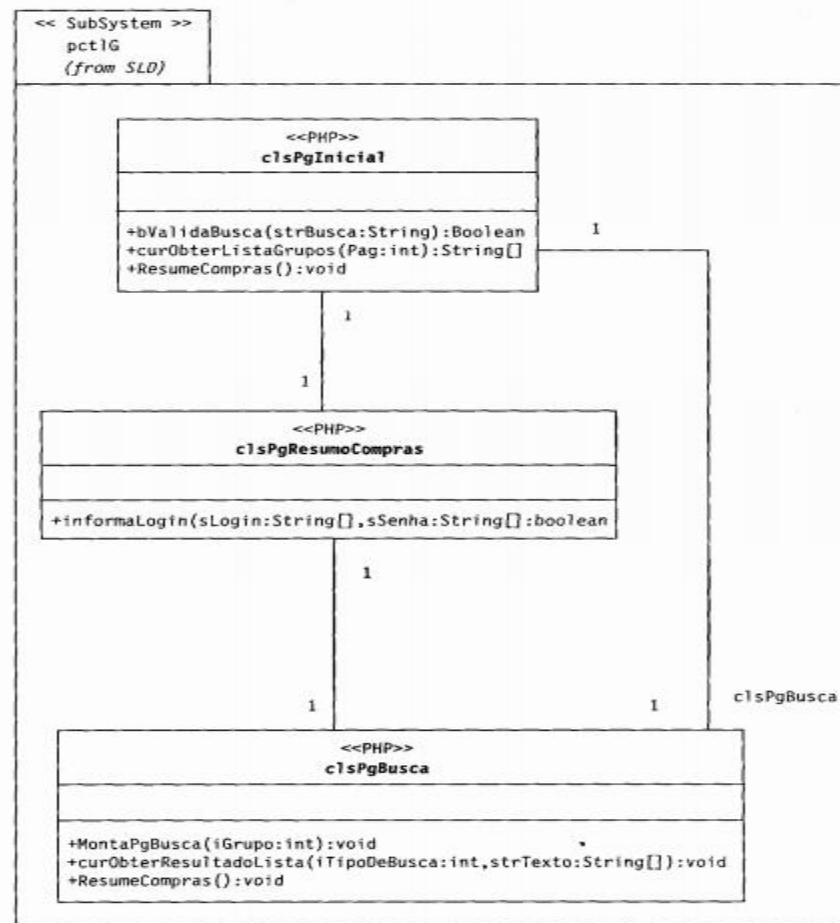




Diagrama de Componentes

Universidade Federal do Maranhão - UFMA

Departamento de Informática

Processo de Desenvolvimento de Software

Prof^a.MSc Simara Rocha

simararocha@gmail.com/simara@deinf.ufma.br

Referências: Booch, G. et al. The Unified Modeling Language User Guide
Medeiros, E. Desenvolvendo Software com UML 2.0: Definitivo, Makron Books, 2006.



Definição

- Diagrama de Componentes: mostra vários componentes em um sistema e suas dependências
- Diagrama de Implantação (Utilização): mostras as relações físicas entre componentes de software e hardware no sistema implementado
- Podem ser criados separadamente ou combinados (quais os componentes funcionam em que nós)



Diagrama de Componentes

- Apresenta uma visão estática de como o sistema está implementado e quais os seus módulos de software: componentes
- Muito associado a linguagem de programação
- Procurar associar módulos, bibliotecas, formulários, arquivos, tabelas ...



Objetivos

- Modelar os componentes do código-fonte e executável
- Bando de dados físicos
- Destacar a função de cada módulo para facilitar a reutilização
- Pode ser utilizado no processo de engenharia reversa por meio da organização dos módulos do sistema e seus relacionamentos



Diagrama de Componente

- Qualquer parte de seu sistema pode ser representado em um diagrama de componente
- São usados para explicar a lógica os artefatos que são usados para implementar as expressões lógicas de projeto do Caso de uso e Diagrama de Classes;
 - Artefatos: qualquer tipo de código que implementa o componente. Se agrupam normalmente em três níveis:
 - Componentes de Instalação: Ex. Máquina Java, DB...;
 - Componente de Funcionamento do produto: Jar, DLL...
 - Componentes de Execução: HTML, Servlets...

Diagrama de Componentes

■ Notação

- Componente:.. O componente pode ser uma página HTML, um arquivo txt, dll, jar e etc.
- Um componente expõe suas interfaces (métodos públicos) para o mundo externo. Para representar isso é possível utilizar a notação de uma interface e estereotipá-la como um componente.
- É possível representar as interface públicas de um componente.

<<componente>>

PedirMaterial

PedirMaterial

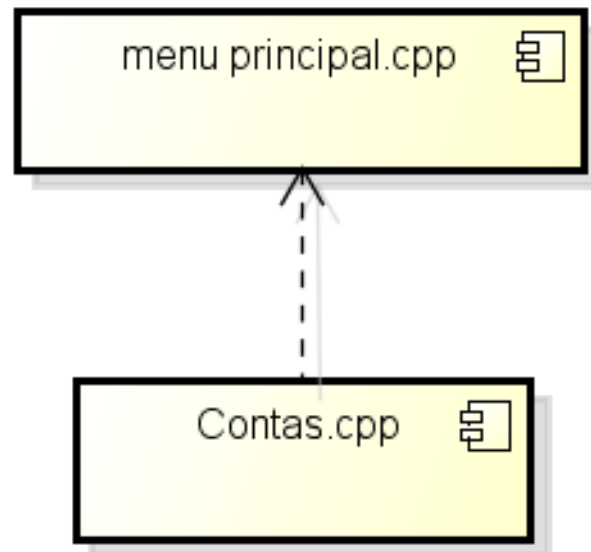


PedirMaterial



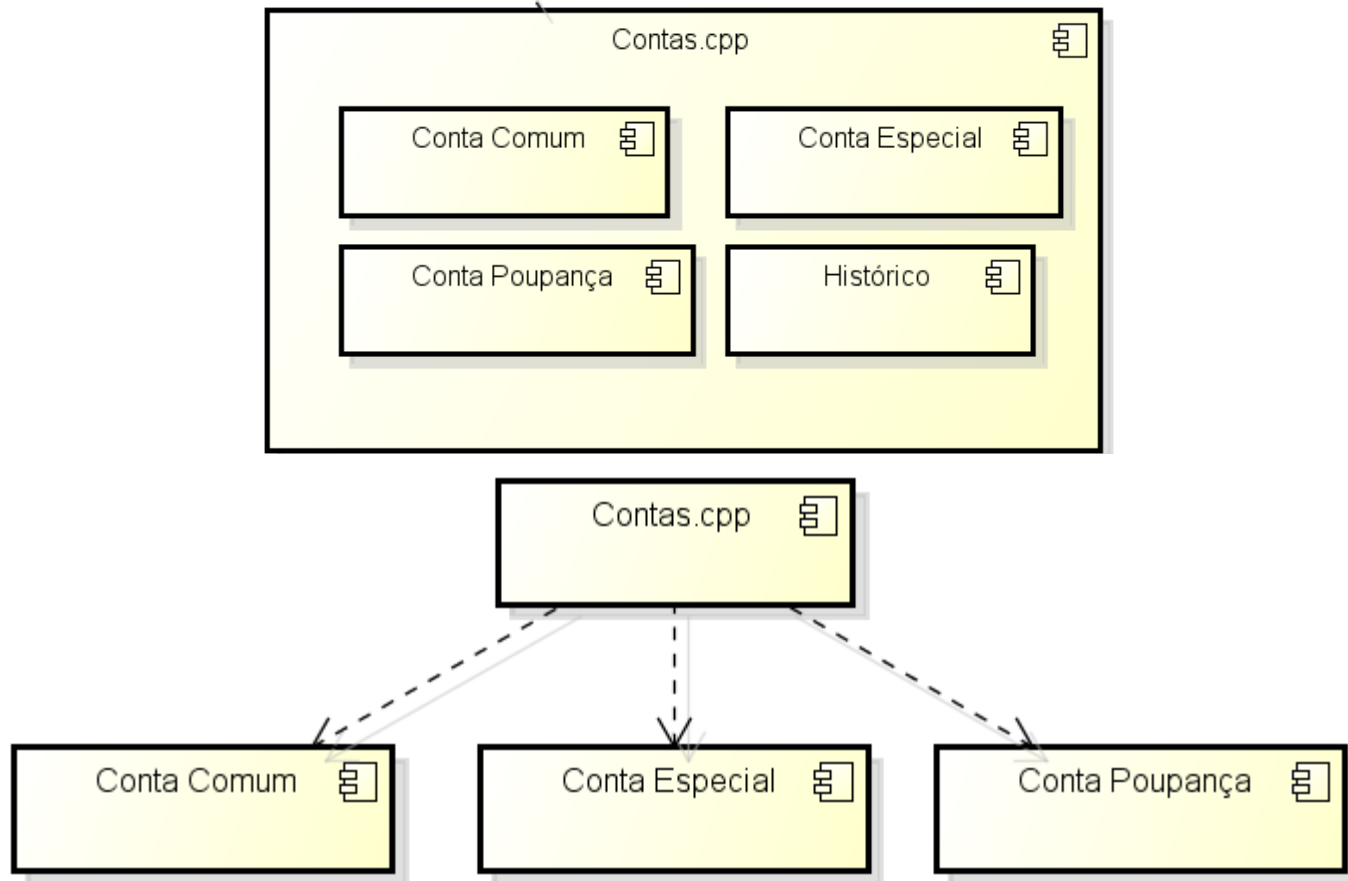
Dependências

- Um componente pode utilizar serviços ou depender de alguma outra forma de outros componentes do sistema



Dependências

- Classes manipuladas por um componente

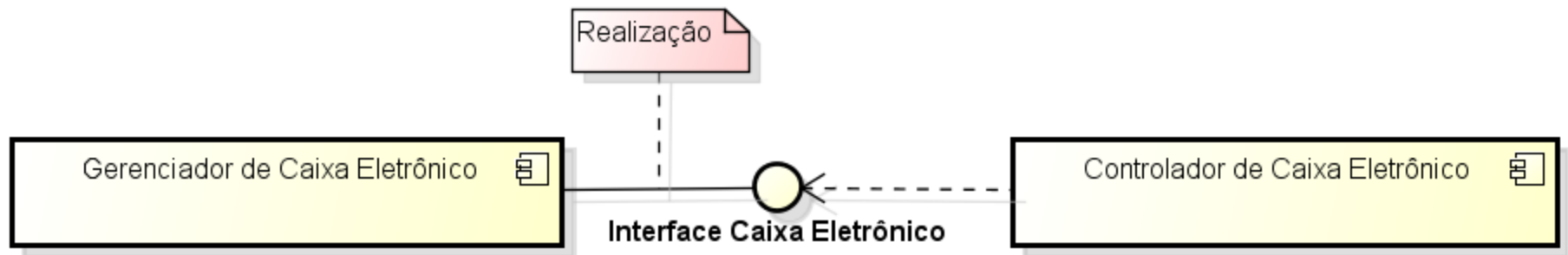




Interface

- Representa um serviço realizado por uma classe ou componente
- As interfaces não possuem implementação ou qualquer especificação interna
- Se um componente implementa uma interface, este relaciona-se com ela através de uma realização
- Se um componente utiliza a interface, relacionamento de dependência

Interface

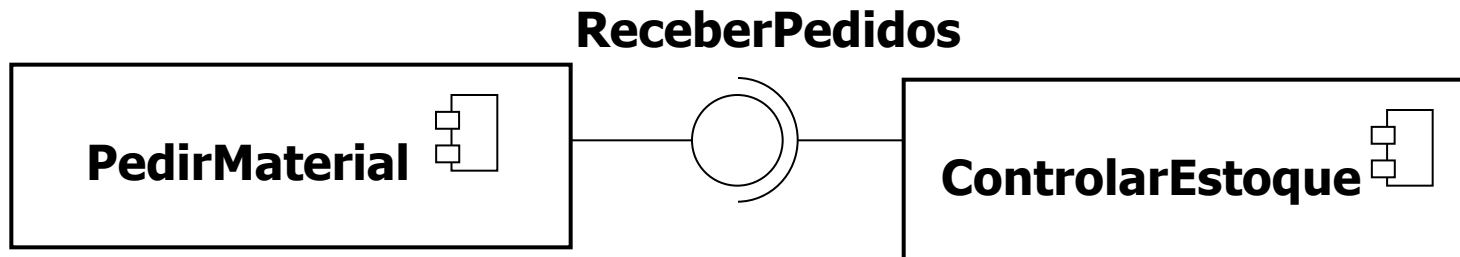


Notação

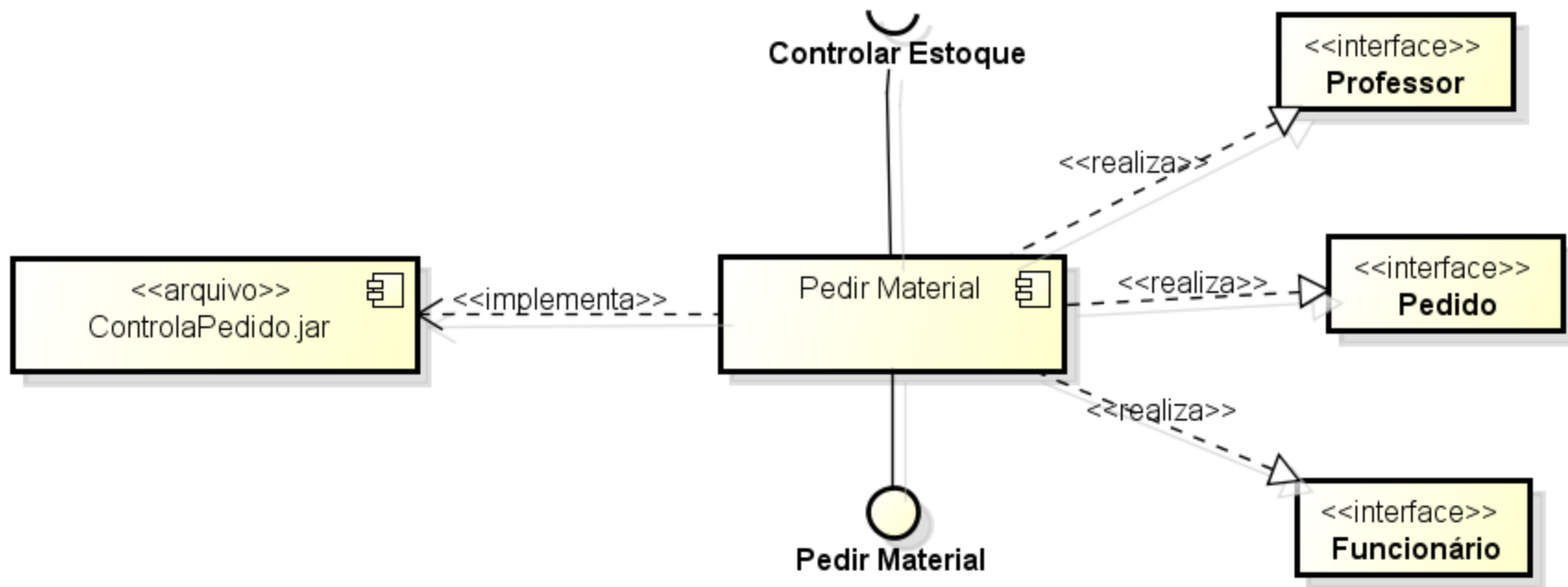
- A forma de representar uma interface esperada



- Interfaces esperadas se encontrando com interfaces fornecidas



Interface





Interface

PedirMaterial



<<interfaces fornecidas>>

ReceberPedidos

<<interfaces requeridas>>

ControlarEstoque

<<realiza>>

Professor

Pedido

Funcionario

<<artefato>>

ControlePedido.JAR

Exemplo

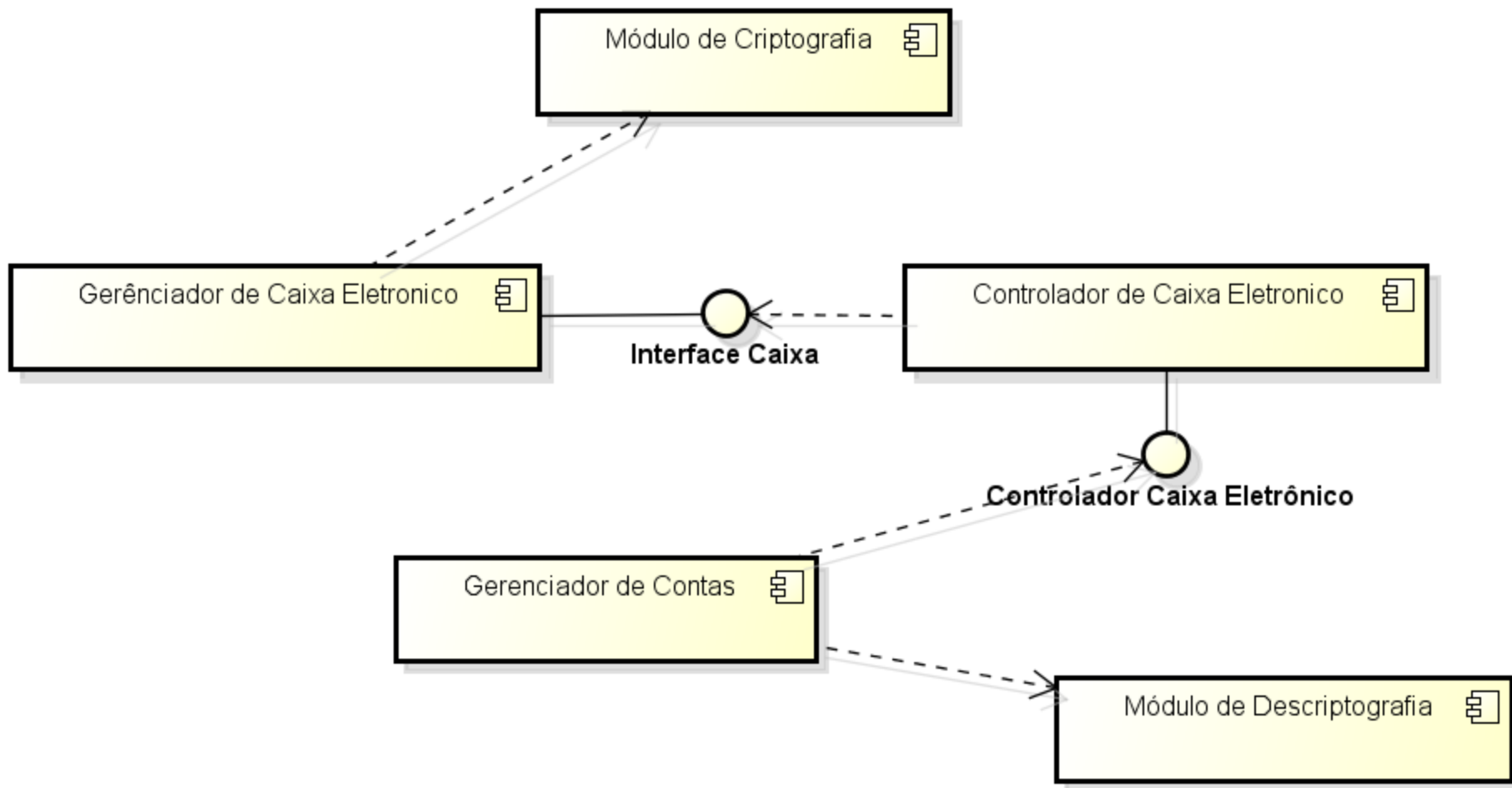




Diagrama de Implantação

Universidade Federal do Maranhão - UFMA

Departamento de Informática

Processo de Desenvolvimento de Software

Prof^a.MSc Simara Rocha

simararocha@gmail.com/simara@deinf.ufma.br

Referências: Booch, G. et al. The Unified Modeling Language User Guide
Medeiros, E. Desenvolvendo Software com UML 2.0: Definitivo, Makron Books, 2006.

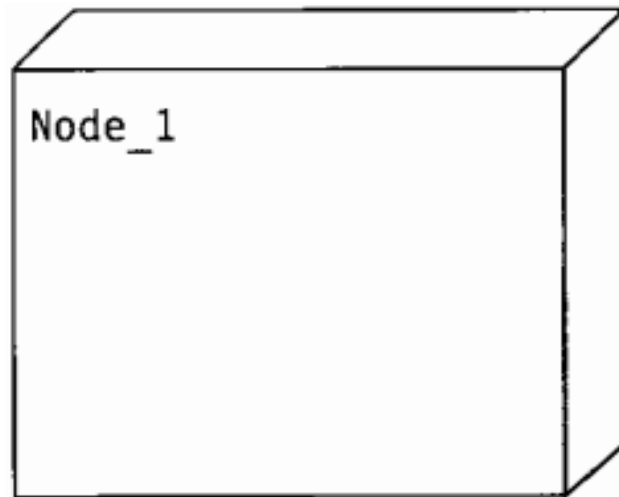


Diagrama de Implantação

- O diagrama de implantação representa como é realizada a distribuição do sistema através de nós de hardware, componentes e dependências de software e as suas devidas relações de comunicação.
- Um diagrama de implantação modela o inter-relacionamento entre recursos de infraestrutura, de rede ou artefatos de sistemas. Normalmente representamos servidores neste diagrama. Estes recursos são chamados de nodes ou nós.

Diagrama de Implantação

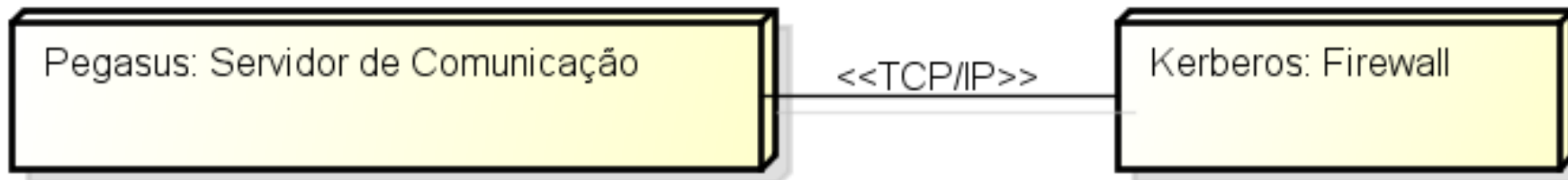
- Cada nó é uma máquina física que encerra um ou vários componentes. Outros dispositivos podem ser representados com o estereótipo de <<dispositivos>> ou <<device>>





Associações

- Os nós podem possuir ligações entre si de forma que possam se comunicar e trocar informações





Nós com Componentes

- Comum identificar os componentes que são executados por um nó

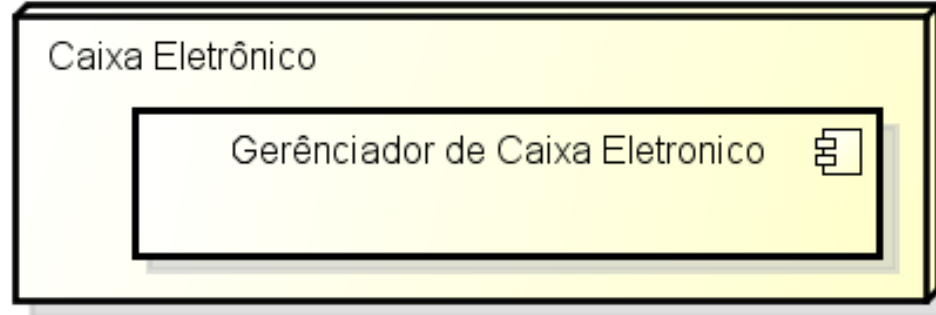


Diagrama de Implantação

- Exemplo:

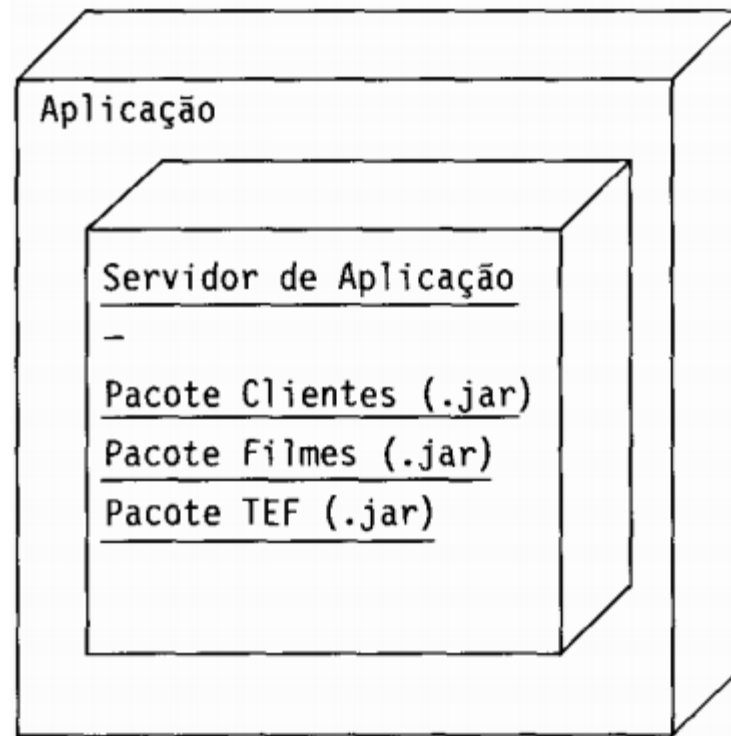


Diagrama de Implantação

- Exemplo:

