

The top-left corner features three overlapping circles in shades of gray and dark gray. The top-right corner features a single dark gray circle.

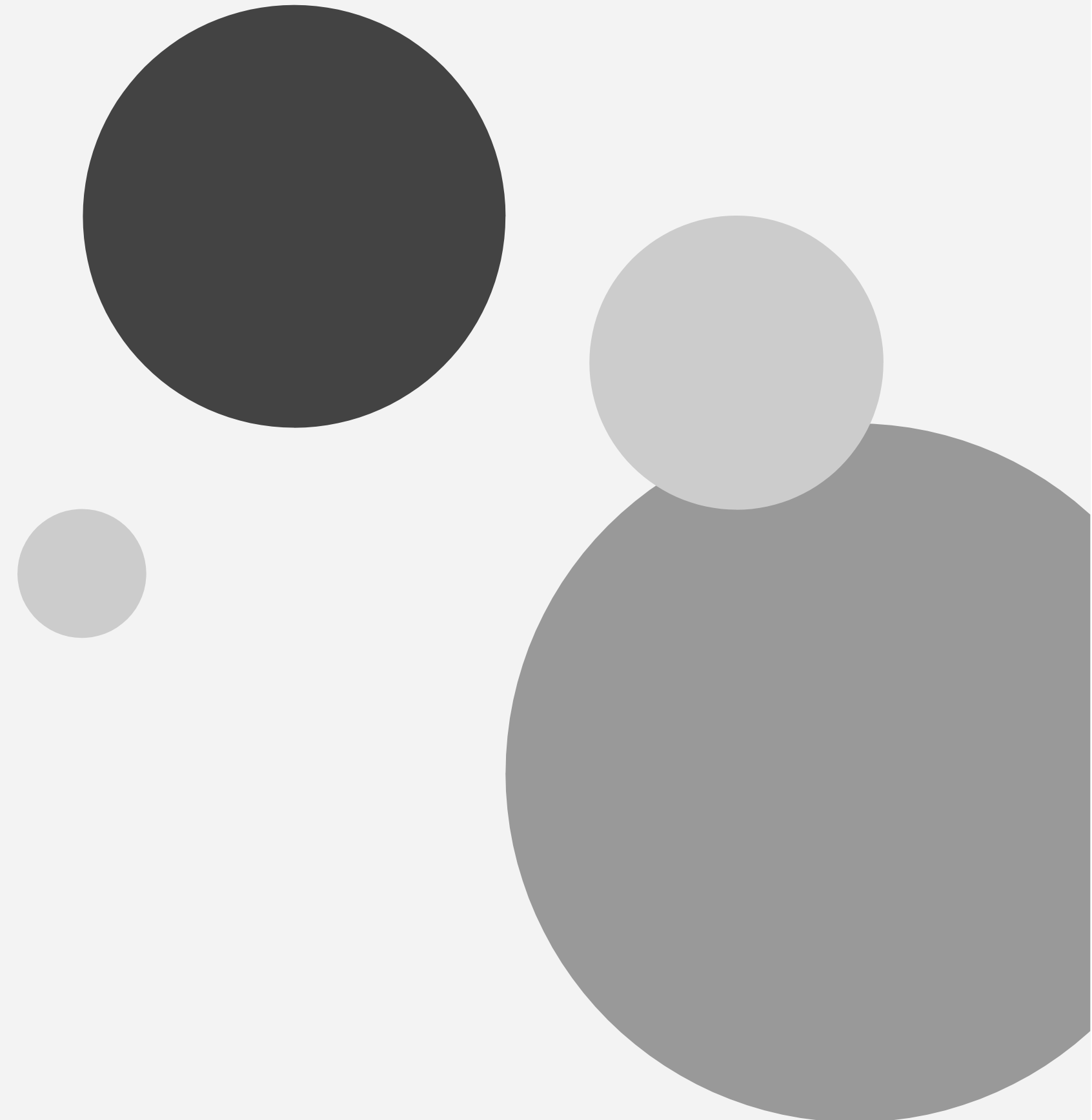
Engenharia de Software Baseada em Componentes

Por Débora Alves e Isabelly Remígio

The bottom-left corner features two overlapping circles, one light gray and one dark gray. The bottom-right corner features three overlapping circles in shades of gray and dark gray.

Sumário

- 01 Desenvolvimento baseado em componentes
- 02 Abordagens da CBSE
- 03 Características dos componentes
- 04 Interface dos componentes
- 05 Modelo dos componentes
- 06 Processo do CBSE
- 07 Tipos de composição de componentes
- 08 Referências



Desenvolvimento baseado em componentes

A Engenharia de Software Baseada em Componentes (CBSE) surgiu no final dos anos 90 para promover o reúso de componentes de software, superando limitações do desenvolvimento orientado a objetos, que falhou em alcançar reúso extensivo devido à especificidade excessiva das classes individuais.

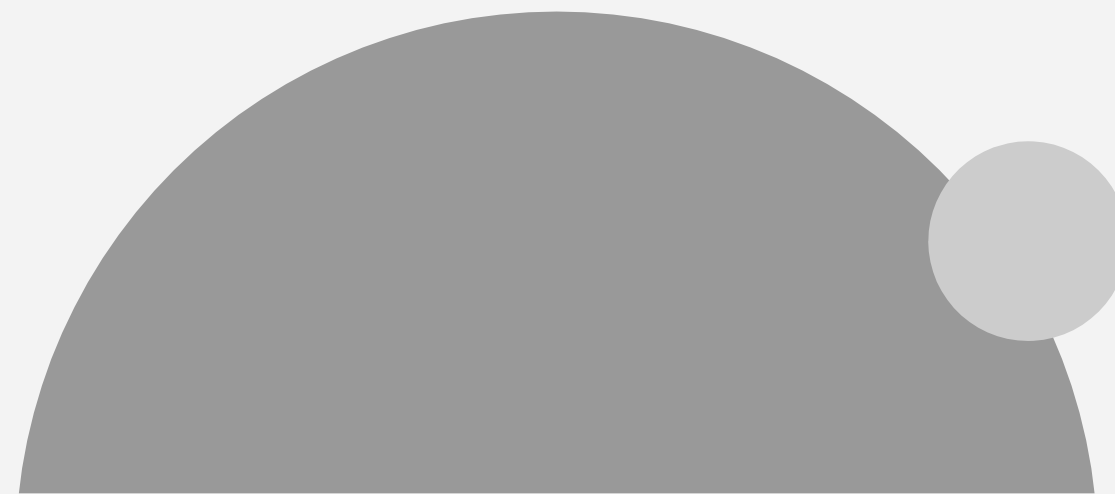
A CBSE tornou-se essencial para o desenvolvimento de sistemas corporativos de larga escala, atendendo a requisitos rigorosos de desempenho e segurança. Para suprir a demanda por software **seguro e confiável, fornecido e implantado com mais rapidez**, a solução é criar software usando componentes existentes.



Abordagens da CBSE

Desenvolvimento para reuso: Desenvolvimento de componentes (ou serviços) a serem reusados em outras aplicações. Envolve geralmente generalizar os componentes existentes.

Desenvolvimento com reuso: Desenvolvimento de novas aplicações usando componentes (ou serviços) existentes.



Características dos componentes



Padronizado

Um componente usado em CBSE deve seguir um modelo padrão que define interfaces, metadados, documentação, composição e implantação.



Independente

Um componente deve ser composto e implantado de forma independente, sem depender de componentes específicos. Serviços externos necessários devem ser explicitamente especificados em uma interface do tipo "requires".



Documentado

Devem ser completamente documentados para que os usuários avaliem sua adequação às necessidades. Todas as interfaces devem ter a sintaxe especificada e, idealmente, também a semântica.



Passível de composição

Para permitir composição, todas as interações externas de um componente devem ocorrer por interfaces públicas definidas. O componente também deve fornecer acesso externo a informações como seus métodos e atributos.

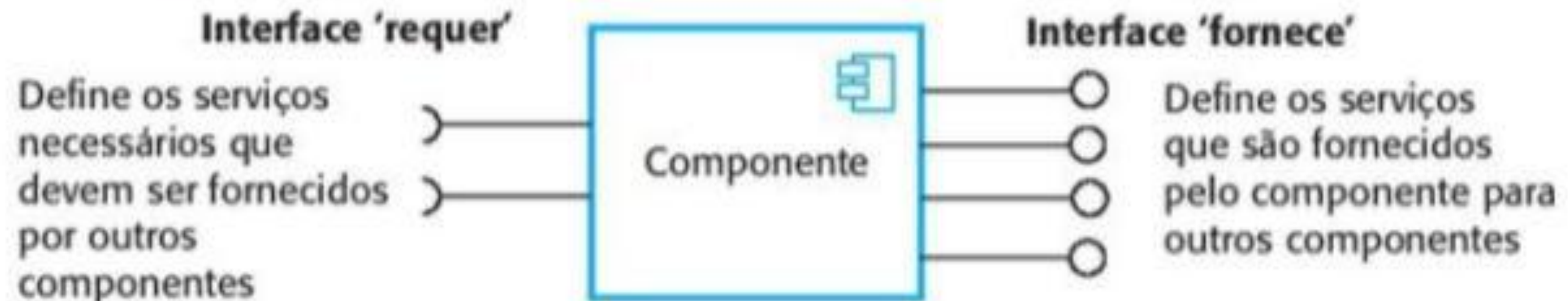
Características dos componentes



Implantável

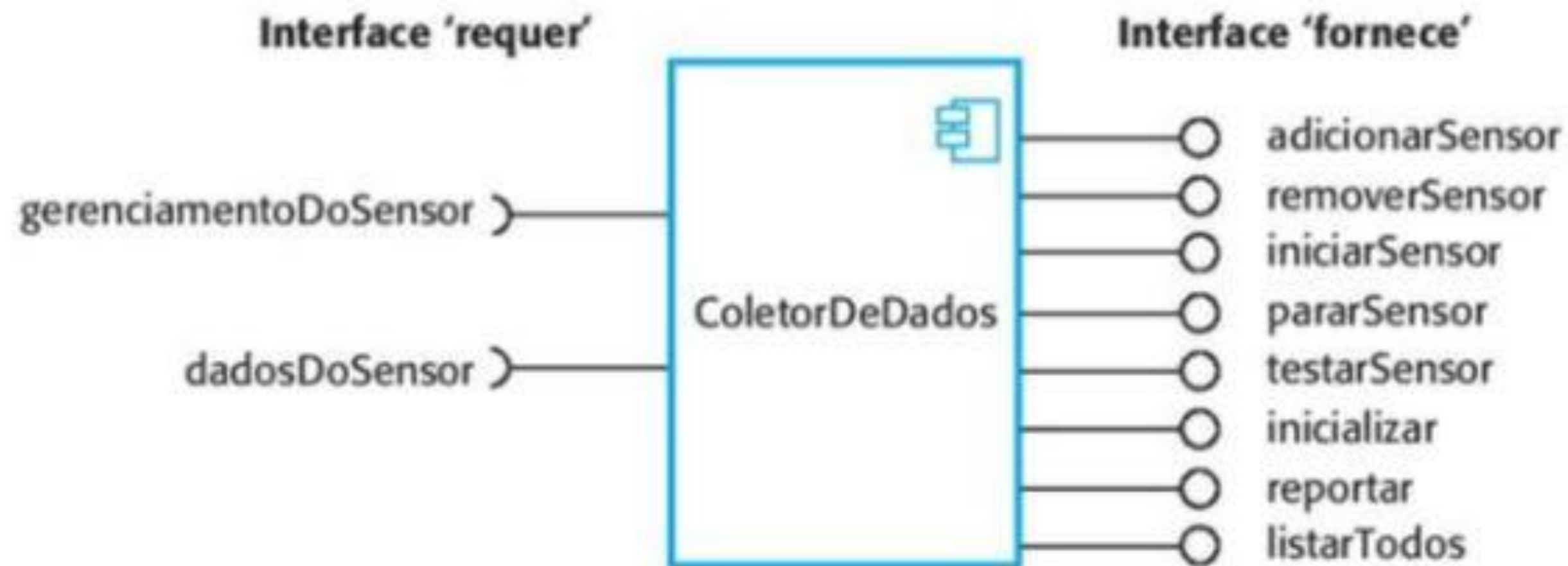
Para ser implantado, um componente deve ser autocontido e capaz de operar como uma entidade independente em uma plataforma com suporte ao modelo de componentes. Geralmente, isso significa que o componente é um código binário, dispensando compilação antes da implantação. Se implementado como um serviço, sua implantação é feita pelo prestador de serviço, e não pelo usuário.

FIGURA 16.2 Interfaces de componentes.



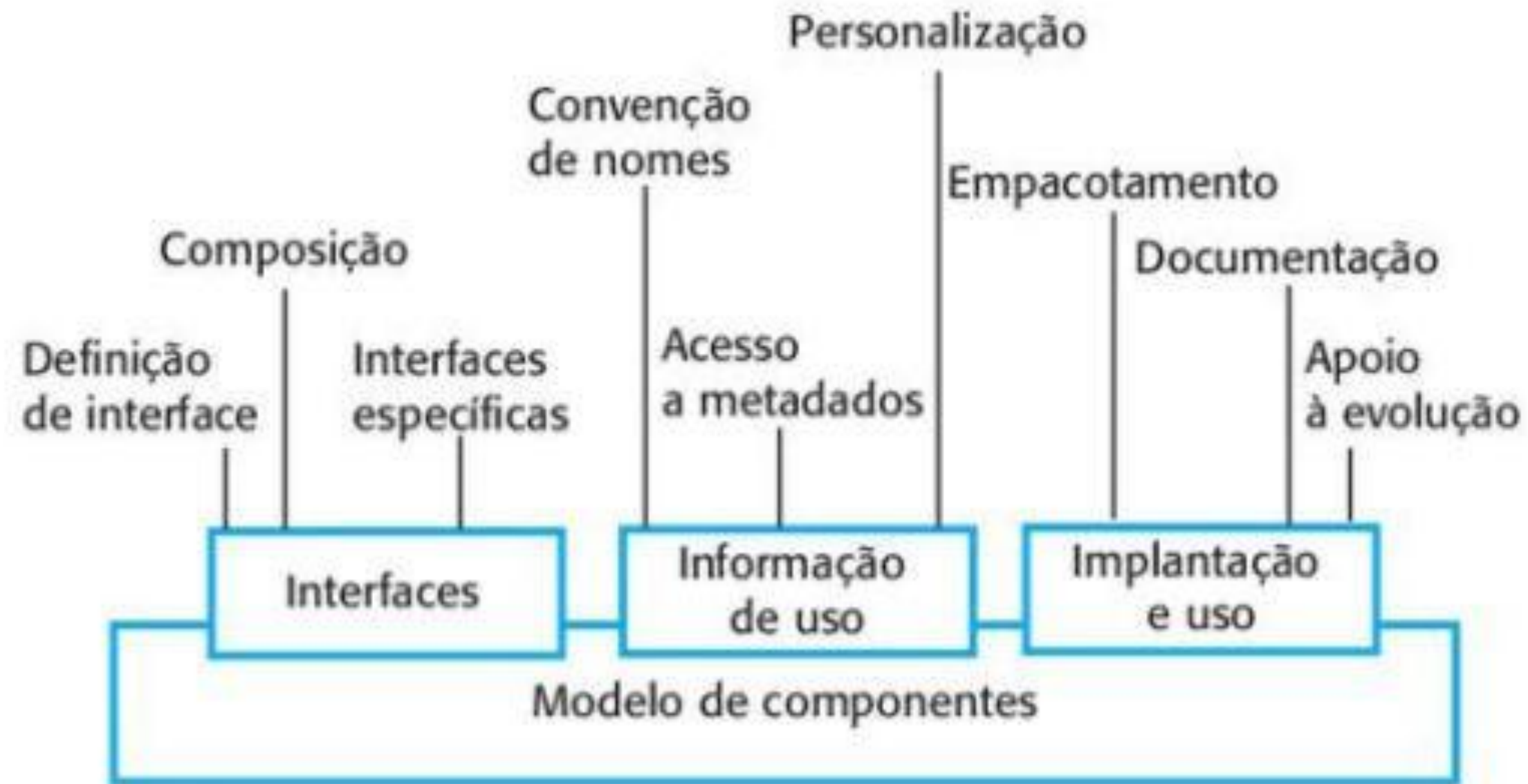
Interface dos componentes

FIGURA 16.3 Modelo de um componente coletor de dados.



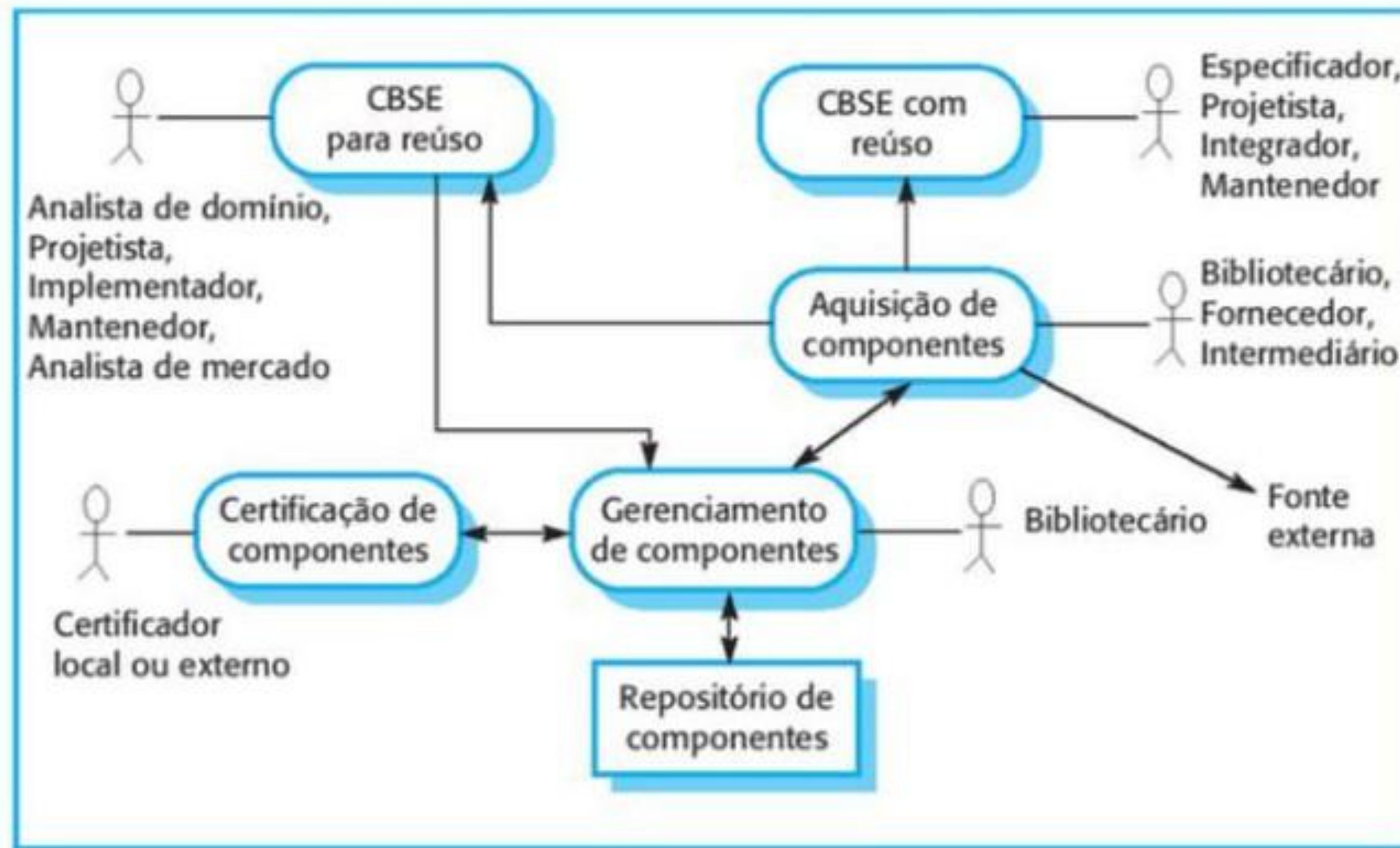
Exemplo de Interface dos componentes

FIGURA 16.4 Elementos básicos de um modelo de componentes.



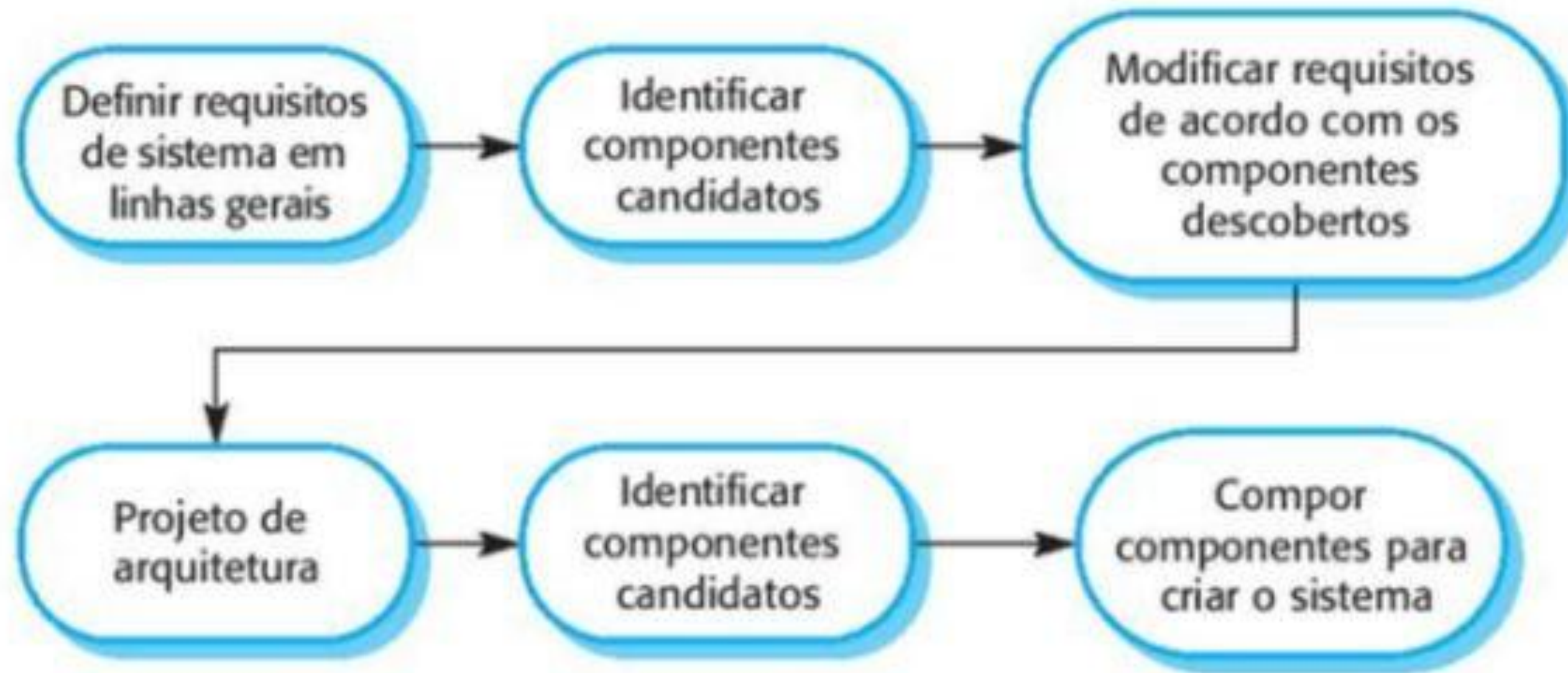
Modelo de componentes

FIGURA 16.6 Processos de CBSE.



Processos do CBSE

FIGURA 16.7 CBSE com reúso.



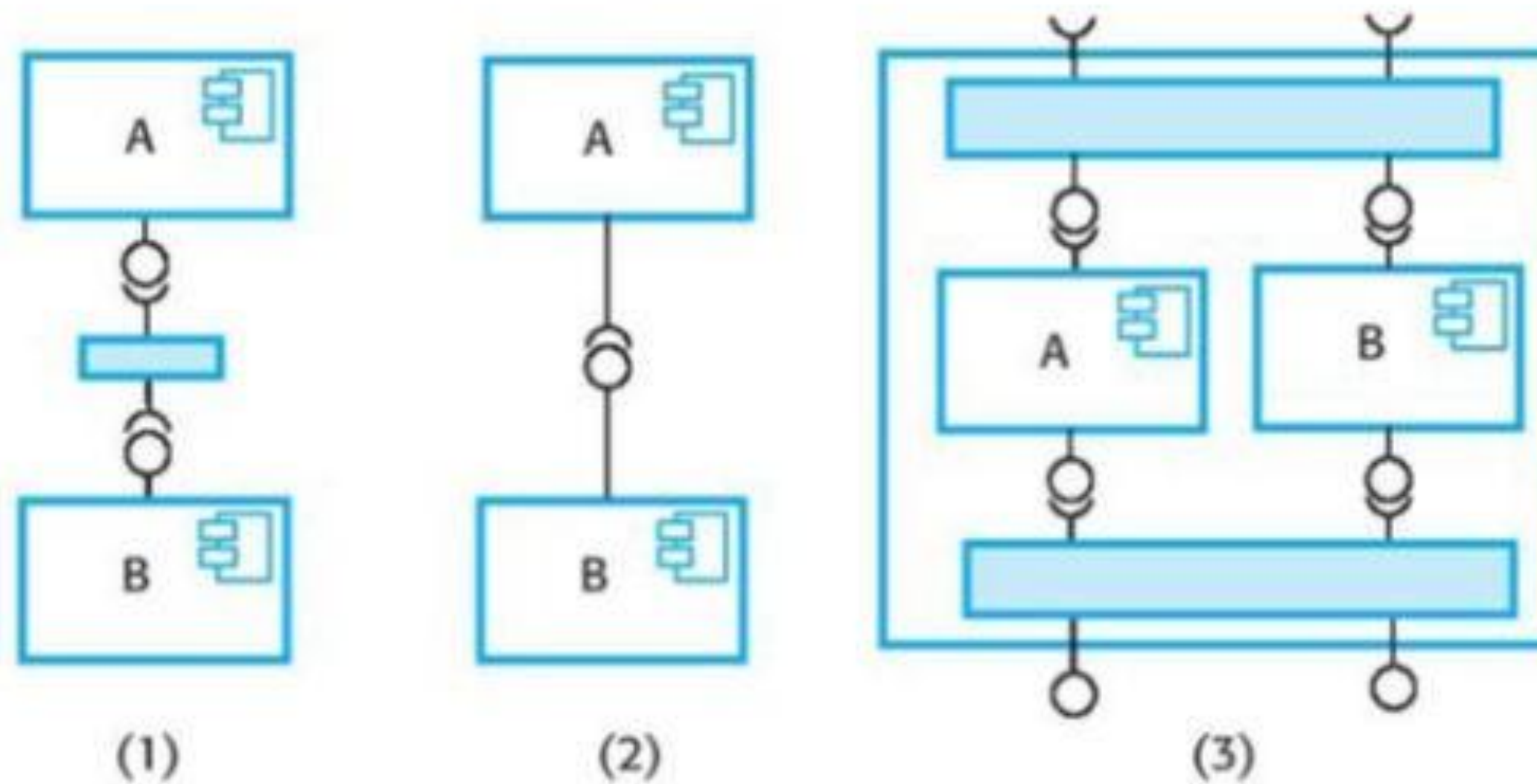
Processos do CBSE com reúso

FIGURA 16.8 Processo de identificação de componentes.



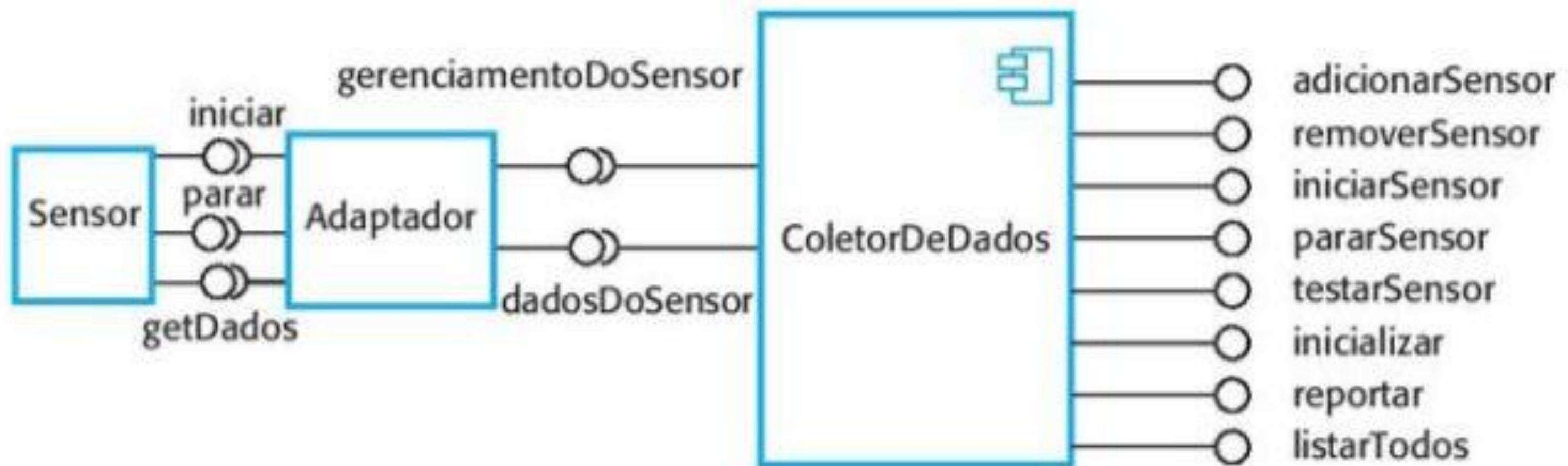
Identificação de componentes

FIGURA 16.10 Tipos de composição de componente.



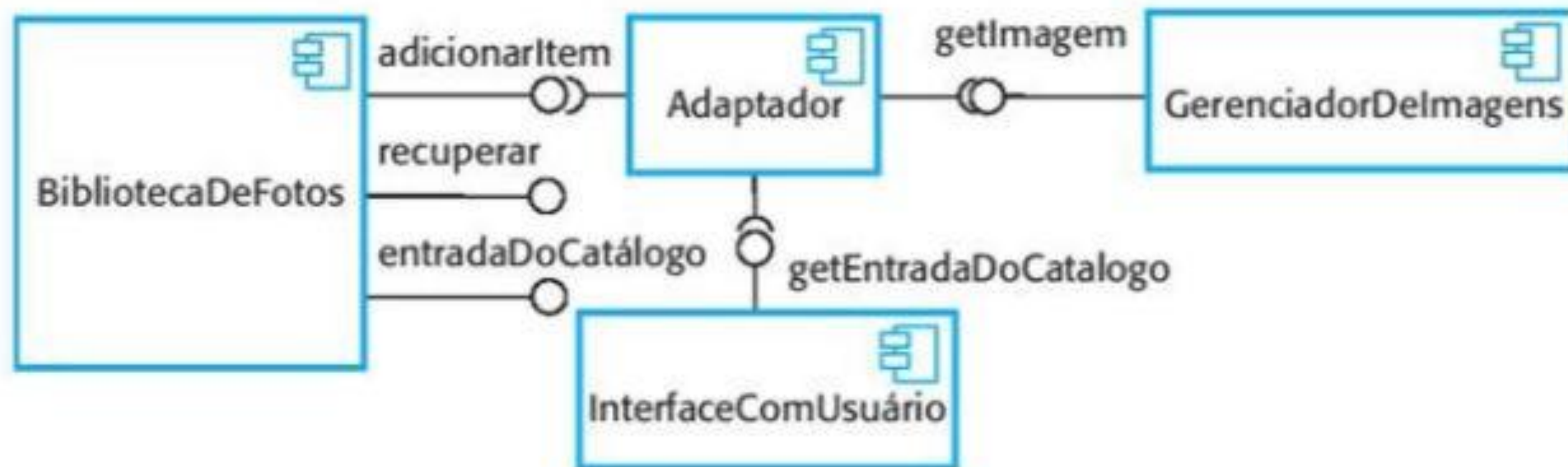
Tipos de composição de componentes

FIGURA 16.12 Adaptador ligando um coletor de dados e um sensor.



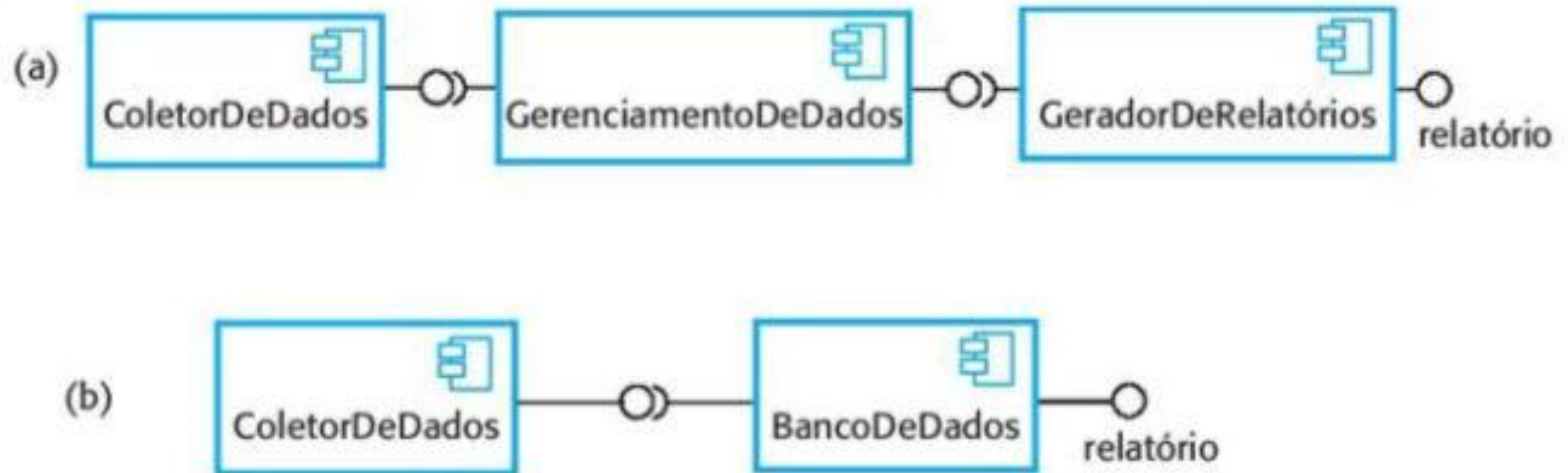
Exemplo de composição de componentes

FIGURA 16.13 Composição da biblioteca de fotos.



Exemplo de composição de componentes

FIGURA 16.15 Componentes de coletar dados e gerar relatórios.



Exemplo de composição de componentes

Referências

UNIVESP. *Engenharia de Software - Aula 23 - Desenvolvimento baseado em componentes*. 2017. Disponível em: https://www.youtube.com/watch?v=_IBEYvZMJfl. Acesso em: 9 jan. 2025.

SOMMERVILLE, Ian. *Engenharia desoftware*. 10. ed. Tradução de Luiz Cláudio Queiroz. Revisão de Fábio Levy Siqueira. São Paulo: Pearson, 2018.

