

Sistemas Distribuídos – 1º semestre de 2021

Professor Mestre Fabio Pereira da Silva

Objetivos

- Explicar a noção de um serviço reusável, baseado em padrões de Web service, que fornece um mecanismo para computação interorganizacional
- Descrever o processo de engenharia de serviços que tem a intenção de produzir web services reusáveis
- Apresentar a composição de serviços como um meio de desenvolvimento de aplicações
- Mostrar como os modelos de processo de negócios podem ser usados como referência para os projetos de sistemas orientados a serviços

Tópicos cobertos

- Serviços como componentes reusáveis
- Engenharia de serviços
- Desenvolvimento de software como serviços

Arquiteturas orientadas a serviços

- É um meio de desenvolvimento de sistemas distribuídos onde os componentes são serviços dedicados.
- Os serviços podem ser executados em computadores diferentes a partir de provedores de serviços diferentes.
- Protocolos padronizados foram desenvolvidos para apoiar a comunicação de serviços e a troca de informações.

Arquiteturas orientadas a serviços

Figura 31.1

Arquitetura orientada a serviços.



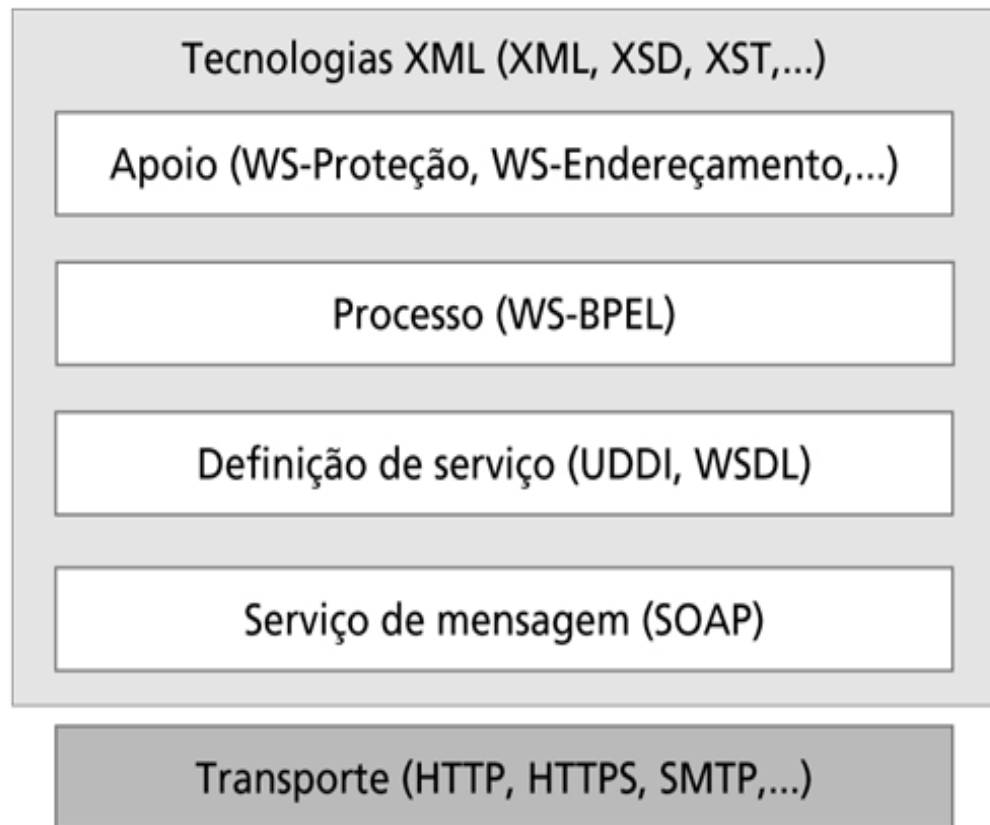
Benefícios do SOA

- Os serviços podem ser providos localmente ou terceirizados para provedores externos.
- Os serviços são independentes de linguagem.
- Investimentos em sistemas legados podem ser preservados.
- A computação interorganizacional é facilitada por meio da troca simplificada de informações.

Padrões de *Web services*

Figura 31.2

Padrões de serviços Web.



Padrões-chave

- SOAP
 - É um padrão de troca de mensagens que apóia a comunicação entre serviços.
- WSDL (Web Service Definition Language)
 - Este padrão permite que uma interface de serviço e suas ligações sejam definidas.
- UDDI
 - Define os componentes de uma especificação de serviços que pode ser usada para descobrir a existência de um serviço.
- WS-BPEL
 - É um padrão para linguagens de workflow usado para definir a composição de serviços.

Engenharia de software orientada a serviços

- As abordagens existentes para engenharia de software devem evoluir para refletir a abordagem orientada a serviços para o desenvolvimento de software
 - Engenharia de serviços é o desenvolvimento de serviços confiáveis e reusáveis
 - Desenvolvimento de software para reuso.
 - Desenvolvimento de software com serviços é o desenvolvimento de software confiável, onde serviços são componentes fundamentais
 - Desenvolvimento de software com reuso.

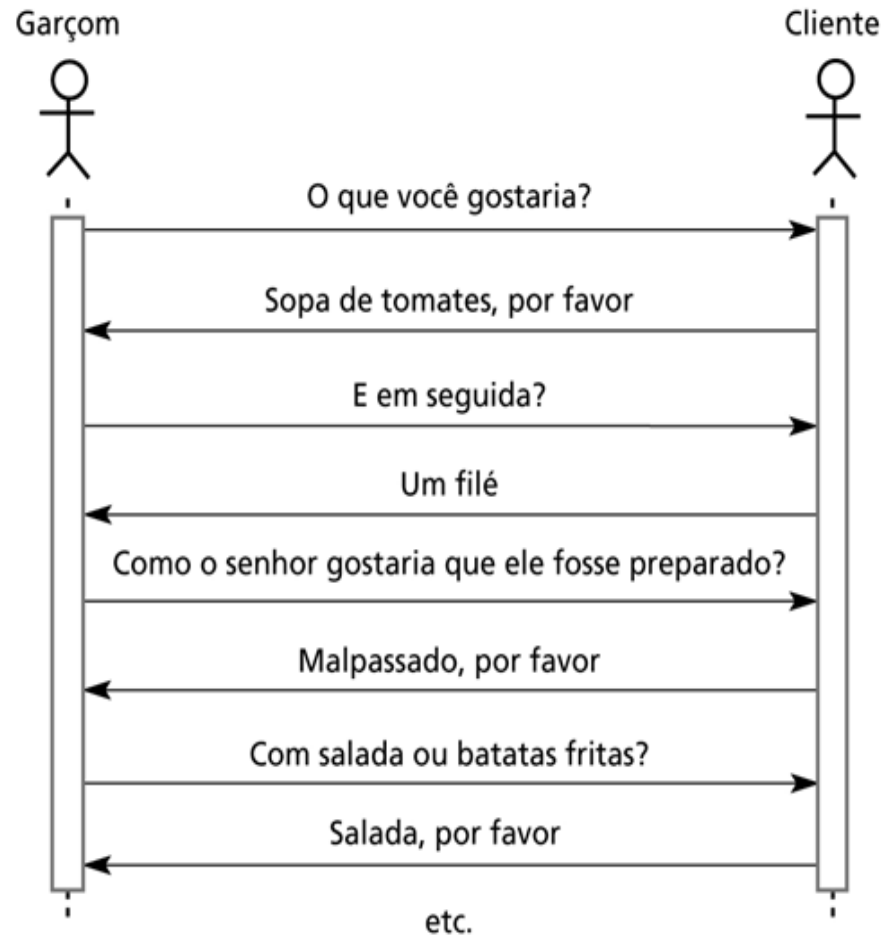
Serviços como componentes reusáveis

- Um serviço pode ser definido como:
 - *Um componente de software reusável, não firmemente acoplado, que engloba a funcionalidade discreta, que pode ser distribuído e acessado por meio de programa. Um Web service é um serviço acessado que usa protocolos-padrão da Internet e baseados em XML.*
- Uma crítica à distinção entre um serviço e um componente, conforme definido no CBSE, é que os serviços são independentes
 - Os serviços não tem uma interface ‘requires’.
 - Os serviços contam com comunicação baseada em mensagens com mensagens expressas em XML.

Interação síncrona

Figura 31.3

Interações síncronas no pedido de uma refeição.



Um pedido como uma mensagem XML

Figura 31.4

Pedido de restaurante expresso
como mensagem XML.

```
<starter>
  <dish name = "soup" type = "tomato" />
  <dish name = "soup" type = "fish" />
  <dish name = "pigeon salad" />
</starter>
<main course>
  <dish name = "steak" type = "sirloin" cooking = "medium" />
  <dish name = "steak" type = "fillet" cooking = "rare" />
  <dish name = "sea bass">
</main>
<accompaniment>
  <dish name = "french fries" portions = "2" />
  <dish name = "salad" portions = "1" />
</accompaniment>
```

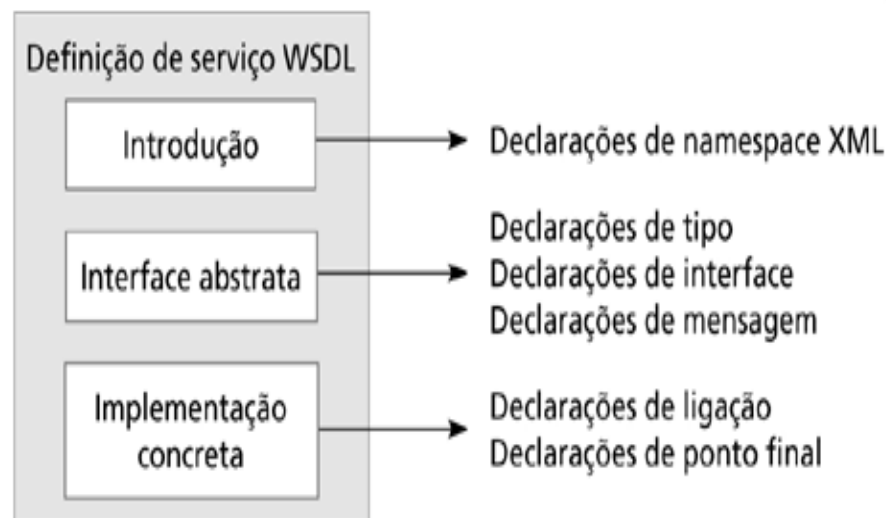
Linguagem de descrição de Web service

- A interface de serviço é definida em uma descrição de serviço expressa em WSDL. A especificação WSDL define:
 - Quais operações o serviço apóia e o formato das mensagens que são enviadas e recebidas pelo serviço;
 - Como o serviço é acessado – a ligação mapeia a interface abstrata em um conjunto concreto de protocolos;
 - Onde o serviço está localizado. Isto é geralmente expresso como uma URI (*Universal Resource Identifier*).

Estrutura de uma especificação WSDL

Figura 31.5

Organização de uma especificação WSDL.



Um fragmento de descrição WSDL

Figura 31.6

Parte de uma descrição WSDL para um serviço Web.

Define alguns dos tipos usados. Supõe que os prefixos namespace 'ws' referem-se ao namespace URI dos esquemas XML e o prefixo namespace associado com essa definição é weathns.

```
<types>
  <xs: schema targetNameSpace = "http://.../weathns"
    xmlns: weathns = "http://.../weathns" >
    <xs:element name = "PlaceAndDate" type = "pdrec" />
    <xs:element name = "MaxMinTemp" type = "mmtrec" />
    <xs: element name = "InDataFault" type = "errmess" />

    <xs: complexType name = "pdrec"
    <xs: sequence>
      <xs:element name = "town" type = "xs:string"/>
      <xs:element name = "country" type = "xs:string"/>
      <xs:element name = "day" type = "xs:date" />
    </xs:complexType>

    Definições de MaxMinType e InDataFault aqui

  </schema>
</types>
```

Definições de MaxMinType e InDataFault aqui

```
</schema>
</types>
```

Agora define a interface e suas operações. Nesse caso, há somente uma única operação para retornar as temperaturas máxima e mínima

```
<interface name = "weatherInfo">
  <operation name = "getMaxMinTemps" pattern = "wsdl:ns: in-out">
    <input messageLabel = "In" element = "weathns: PlaceAndDate" />
    <output messageLabel = "Out" element = "weathns:MaxMinTemp" />
    <outfault messageLabel = "Out" element = "weathns:InDataFault" />
  </operation>
</interface>
```

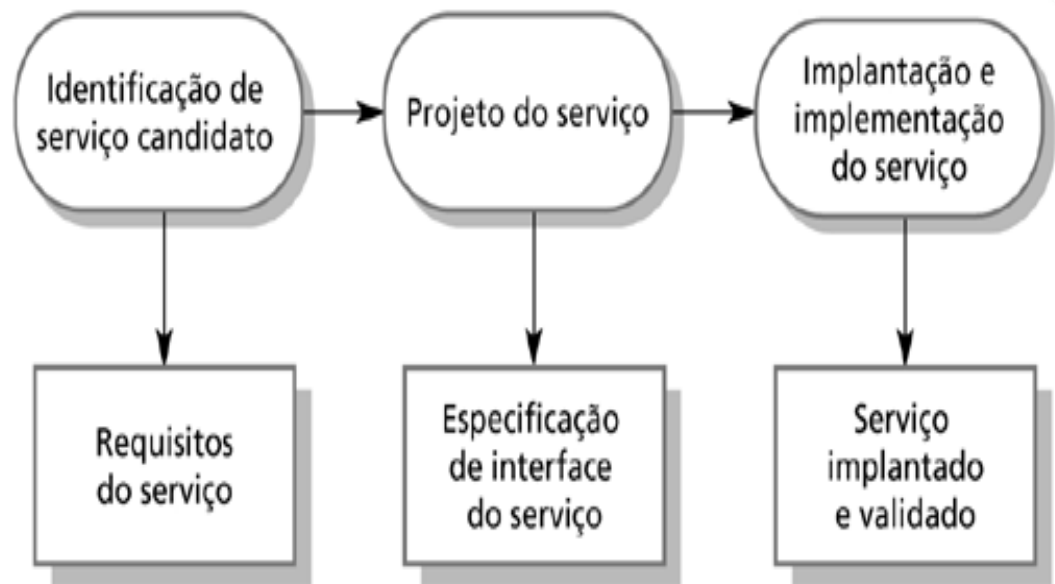
Engenharia de serviços

- É o processo de desenvolvimento de serviços para reuso em aplicações orientadas a serviços.
- O serviço deve ser projetado como uma abstração reusável, que pode ser usada em sistemas diferentes.
- Envolve:
 - Identificação do serviço candidato
 - Projeto do serviço
 - Implementação do serviço

O processo de engenharia de serviços

Figura 31.7

Processo de engenharia de serviços.



Identificação de serviço candidato

- Três tipos fundamentais de serviço
 - Serviços de utilidades que implementam funcionalidades gerais usada por diferentes processos de negócio.
 - Serviços de negócio que são associados à uma função específica de negócio, por exemplo, em uma universidade, registro de estudantes.
 - Serviços de coordenação que apóiam processos compostos, tais como pedidos.

Classificação de serviços

Tabela 31.1 Classificação de serviços

	Utilidade	Negócio	Coordenação
Tarefa	Conversor de moeda Localizador de funcionário	Validar formulário de solicitação Verificar avaliação de crédito	Processar solicitação de reembolso Pagar fornecedor externo
Entidade	Verificador de estilo de documento Conversor de formulário Web em XML	Formulário de gastos Formulário de solicitação do estudante	

Identificação de serviços

- O serviço é associado com uma única entidade lógica usada em diferentes processos de negócio?
- A tarefa é aquela que é cumprida por pessoas diferentes na organização?
- O serviço é independente?
- O serviço tem de manter o status? Uma base de dados é necessária?
- O serviço poderia ser usado por clientes de fora da organização?
- Diferentes usuários de serviços devem ter diferentes requisitos não funcionais?

Serviços de catálogo

- Criado por um fornecedor para mostrar qual bem pode ser solicitado por outras empresas
- Requisitos de serviços
 - Uma versão específica de catálogo deve ser criada para cada cliente.
 - O catálogo poderá ser baixado.
 - A especificação e os preços de até 6 itens podem ser comparados.
 - Recursos de browsing e pesquisa serão fornecidos.
 - Uma função que permita que a data de entrega seja prevista para itens pedidos será fornecida.
 - Pedidos virtuais serão apoiados que reservam produtos por 48 horas para permitir que um pedido real seja colocada serão fornecidos.

Catálogo – requisitos não-funcionais

- O acesso será restrito aos empregados das organizações credenciadas.
- Os preços e configurações oferecidos para cada organização serão confidenciais.
- O catálogo estará disponível de 0700 GMT a 1100 GMT.
- O catálogo será capaz de processar até 10 solicitações por segundo.

Operações de serviços de catálogo

Tabela 31.2 Descrições funcionais de operações de serviços de catálogo

Operação	Descrição
MakeCatalogue	Cria uma versão do catálogo feita sob medida para um cliente específico. Inclui um parâmetro opcional para criar uma versão do catálogo em PDF que possa ser baixada.
Compare	Fornece uma comparação de até seis características (por exemplo preço, dimensões, velocidade do processador etc.), de até quatro itens de catálogo por vez.
Lookup	Mostra todos os dados associados a um item específico do catálogo.
Search	Essa operação busca uma expressão lógica no catálogo. Ela mostra uma lista de todos os itens que possuem a expressão procurada.
CheckDelivery	Retorna a data de entrega prevista de um item pedido hoje.
MakeVirtualOrder	Reserva o número de itens a serem pedidos por um cliente e fornece informações do item para o sistema de compras dos clientes.

Projeto de interface de serviço

- Envolve pensar sobre as operações associadas ao serviço e às mensagens trocadas.
- O número de mensagens trocadas para completar uma solicitação de serviço deve, normalmente, ser minimizado.
- Informações de status de serviço podem ser incluídas nas mensagens.

Estágios de projeto de interface

- Projeto de interface lógica
 - Inicia com os requisitos de serviços e define os nomes das operações e parâmetros associados ao serviço. As exceções também devem ser definidas.
- Projeto de mensagem
 - Projetar a estrutura e a organização das mensagens de entrada e de saída. Notações, tais como a UML, são representações mais abstratas que XML.
- Descrição WSDL
 - A especificação lógica é traduzida para uma descrição WSDL.

Projeto de interface de catálogo

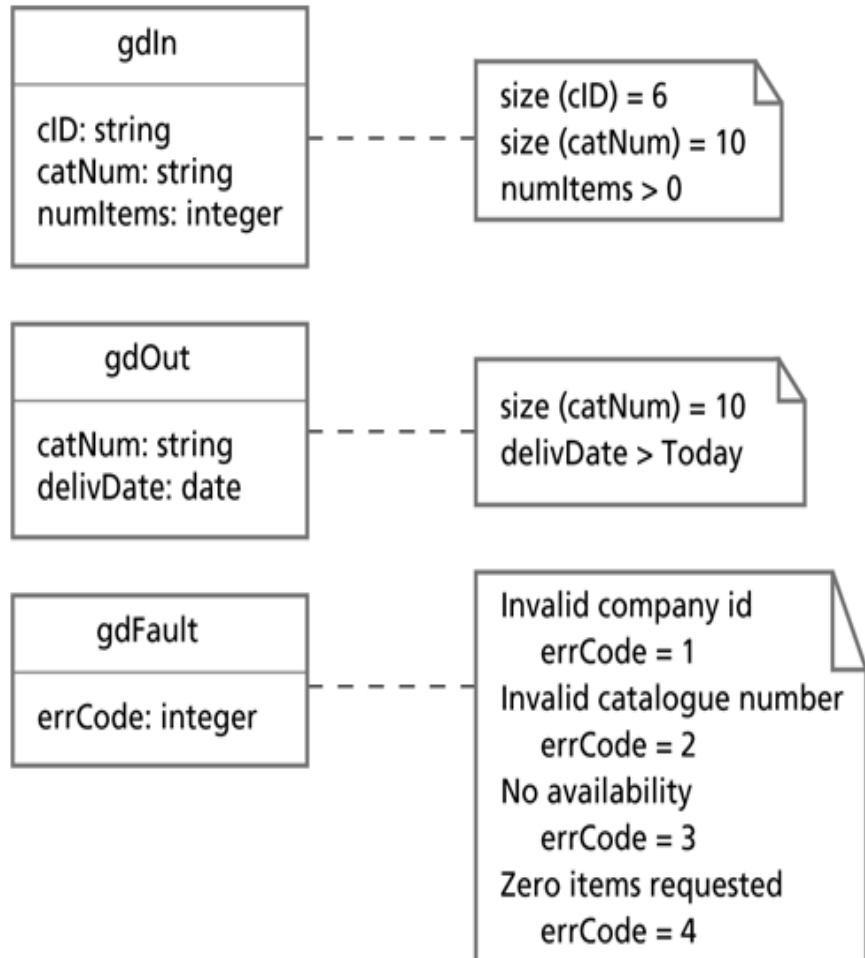
Tabela 31.3 Projeto de interface de catálogo

Operação	Entradas	Saídas	Exceções
MakeCatalogue	<i>mcIn</i> ID da empresa PDF-flag	<i>mcOut</i> URL do catálogo para essa empresa	<i>mcFault</i> ID da empresa inválido
Compare	<i>compIn</i> ID da empresa Atributo de entrada (até 6) Número de catálogo (até 4)	<i>compOut</i> URL da página que mostra a tabela de comparação	<i>compFault</i> ID da empresa inválido Número de catálogo inválido Atributo desconhecido
Lookup	<i>lookIn</i> ID da empresa Número de catálogo	<i>lookOut</i> URL da página com as informações do item	<i>lookFault</i> ID da empresa inválido Número de catálogo inválido
Search	<i>searchIn</i> Id da empresa String de busca	<i>searchOut</i> URL da página com resultados da busca	<i>searchFault</i> ID da empresa inválido String de busca mal-estruturado
CheckDelivery	<i>gdIn</i> ID da empresa Número de catálogo Número de itens solicitados	<i>gdOut</i> Número de catálogo Data de entrega prevista	<i>gdFault</i> ID da empresa inválido Número de catálogo inválido Sem disponibilidade
PlaceOrder	<i>poIn</i> ID da empresa Número de itens solicitados Número de catálogo	<i>poOut</i> Número de catálogo Número de itens solicitados Data de entrega prevista Estimativa de preço unitário Estimativa de preço total	<i>poFault</i> ID da empresa inválido Número de catálogo inválido Zero itens solicitados

Estrutura de entrada e saída de mensagem

Figura 31.8

Definição UML de mensagens de entrada e saída.



Implementação e implantação de serviço

- Pode envolver a programação de serviços usando uma linguagem padronizada de programação ou uma linguagem de workflow.
- Os serviços devem, então, ser testados, criando mensagens de entrada e verificando se as mensagens de saída produzidas são conforme esperadas.
- Implantação envolve a publicação do serviço usando UDDI e sua instalação em um servidor Web. Os servidores atuais fornecem apoio para a instalação de serviço.

Uma descrição UDDI

- Detalhes dos negócios que fornecem o serviço.
- Descrição informal da funcionalidade fornecida pelo serviço.
- Informações sobre onde encontrar a especificação WSDL do serviço.
- Informações de assinatura que permitem que os usuários se registrem para obter atualizações de serviços.

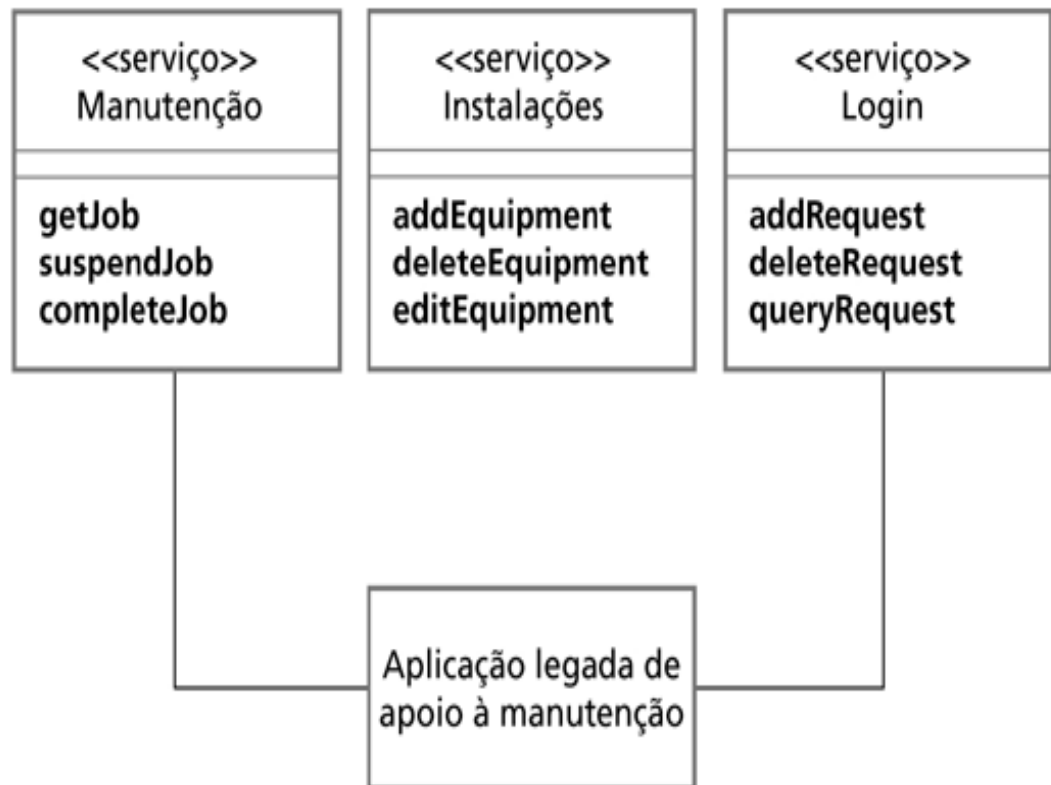
Serviços de sistemas legados

- Uma aplicação importante dos serviços é fornecer acesso à funcionalidade embutida em sistemas legados.
- Sistemas legados oferecem funcionalidade extensiva e isso pode reduzir o custo de implementação de serviço.
- Aplicações externas podem acessar essa funcionalidade por meio de interfaces de serviço.

Acesso a sistema legado

Figura 31.9

Serviços que fornecem acesso a um sistema legado.

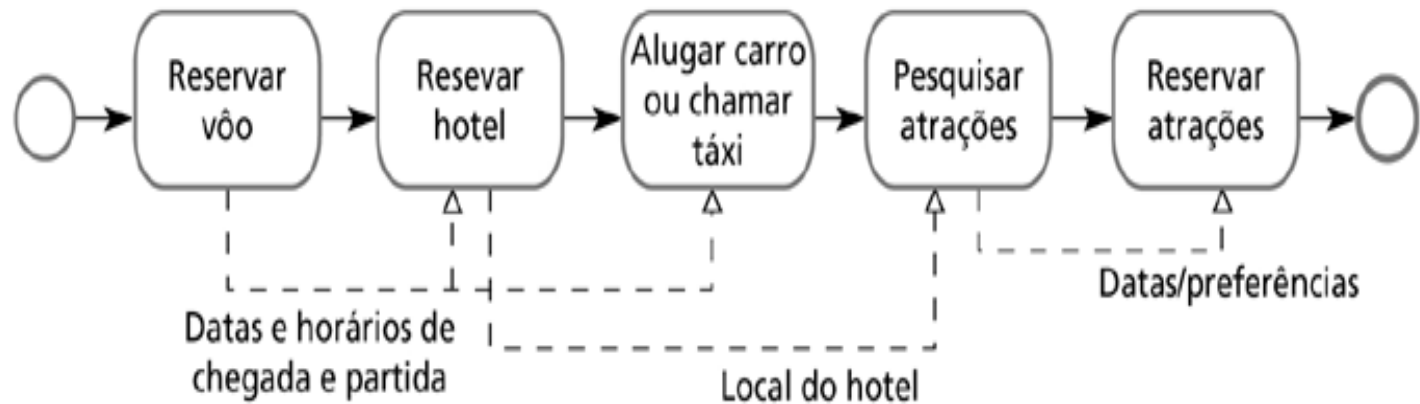


Desenvolvimento de software com serviços

- Os serviços existentes são compostos e configurados para criar novos serviços compostos e aplicações
- A base para composição do serviço é, freqüentemente, um workflow
 - Workflows são seqüências lógicas de atividades que, juntos, modelam um processo coerente de negócio;
 - Por exemplo, fornecer serviços de reserva de viagens que permitem que reservas de vôos, aluguel de carro e hotel sejam coordenados.

Workflow de pacote de férias

Figura 31.10 Workflow de pacote de férias.



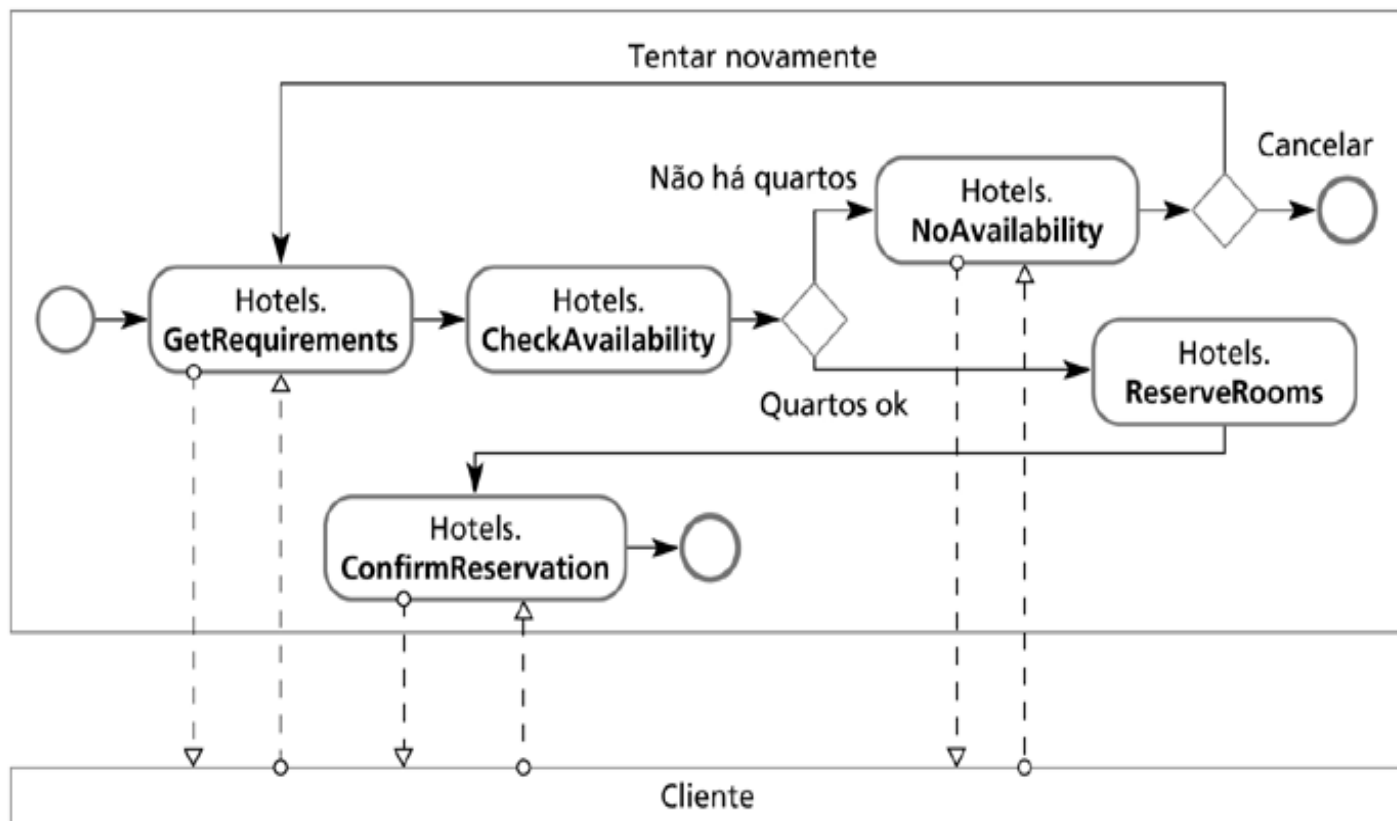
Construção por composição

Figura 31.11 Construção de serviços por composição.



Workflow de reserva de hotel

Figura 31.12 Workflow de reserva de hotel.



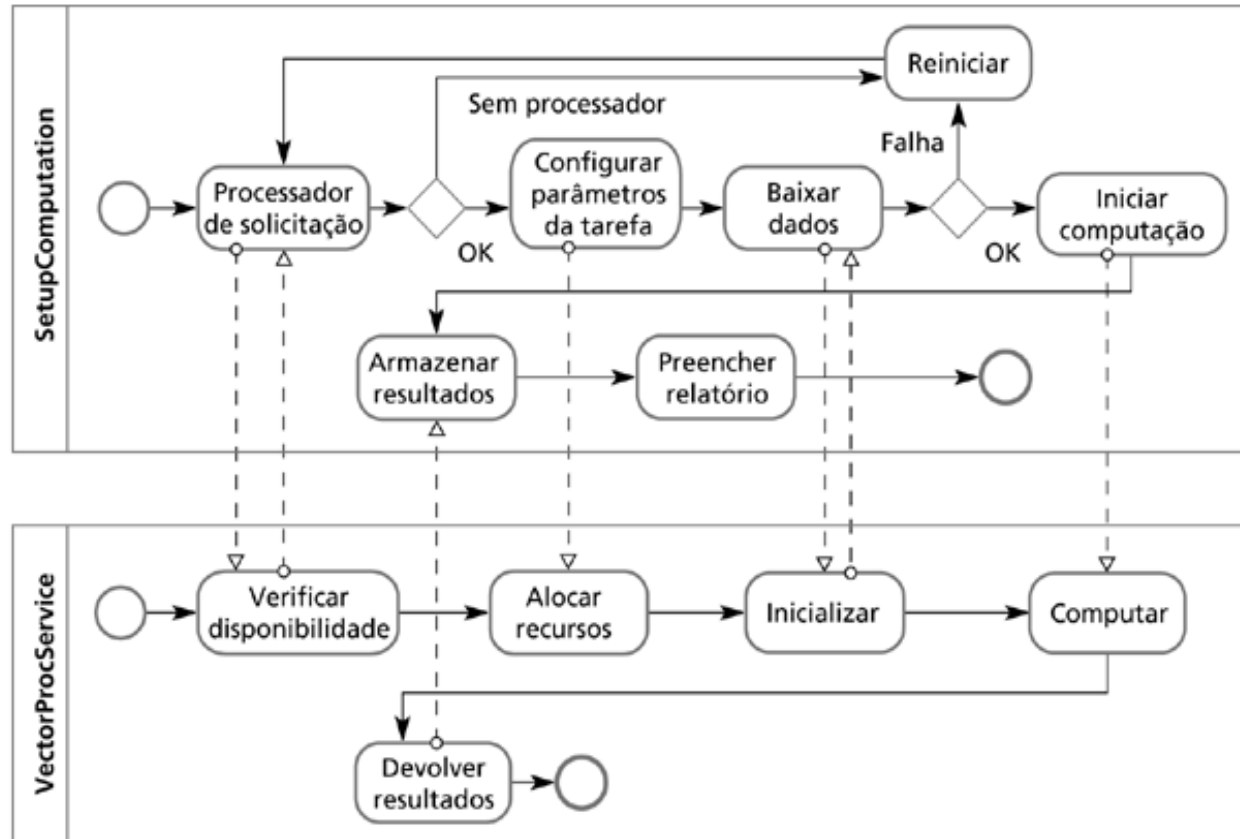
Projeto e implementação de workflow

- WS-BPEL é um padrão XML para especificação de workflows. Contudo, as descrições WS-BPEL são longas e ilegíveis.
- Notações gráficas de workflow, tal como BPMN, são mais legíveis, e as WS-BPEL podem ser geradas a partir delas.
- Em sistemas interorganizacionais, workflows separados são criados para cada organização e ligados por meio de troca de mensagem.

Interação entre *workflows*

Figura 31.13

Interação de workflows.



Teste de serviços

- O teste dedica-se a encontrar defeitos e a demonstrar que um sistema atende aos requisitos funcionais e não-funcionais.
- O teste de serviço é difícil, visto que os serviços (externos) são 'caixas-pretas'. Técnicas de teste baseadas em código de programa fonte não podem ser usadas.

Problemas de teste de serviços

- Serviços externos podem ser modificados pelo provedor de serviço, portanto, invalidam os testes que foram realizados.
- Ligação dinâmica significa que o serviço usado em uma aplicação pode variar – os testes de aplicação não são, portanto, confiáveis.
- O comportamento não funcional do serviço é imprevisível porque é dependente da carga.
- Se os serviços devem ser pagos quando forem usados, testar um serviço pode ser oneroso.
- Pode ser difícil invocar ações de compensação em serviços externos, visto que estes podem se basear em falhas de outros serviços que não podem ser simulados.

Pontos-chave

- Engenharia de software orientada a serviços é baseada na noção de que programas podem ser construídos pela composição de serviços independentes que englobam funcionalidade reusável.
- Interfaces de serviços são definidas em WSDL. Uma especificação WSDL inclui a definição de tipos de interface e operações, o protocolo de ligação usado pelo serviço e a localização do serviço.
- Os serviços podem ser classificados como serviços de utilidades, serviços de negócios ou serviços de coordenação.
- O processo de engenharia de serviço envolve a identificação de serviços candidatos para implementação, definição de interface de serviço e implementação, teste e implantação do serviço.

Pontos-chave

- Interfaces de serviço podem ser definidas para sistemas de software legados que podem, então, ser reusados em outras aplicações.
- O desenvolvimento de software usando serviços envolve a criação de programas pela composição e configuração de serviços para criar novos serviços compostos.
- Modelos de processo de negócio definem as atividades e troca de informações em processos de negócio. As atividades em processo de negócio podem ser implementadas por serviços e, desse modo, o modelo de processo de negócio representa uma composição de serviços.
- Técnicas de teste de software baseadas em análise de código fonte não podem ser usadas em sistemas orientadas a serviços que contam com serviços fornecidos externamente.

Contatos

- Email: fabio.silva321@fatec.sp.gov.br
- LinkedIn: <https://br.linkedin.com/in/b41a5269>
- Facebook: <https://www.facebook.com/fabio.silva.56211>

Bibliografia

- O conteúdo da aula foi elaborado utilizando o seguinte material de apoio:
- Engenharia de software - Ian Sommerville - 9 edição