

# Engenharia de Software Orientada a Serviços - SOA

**UNIP - Araraquara**

**Curso:** Ciências da Computação

**Disciplina:** Qualidade de Software

**Profº:** João Paulo Moreira dos Santos

# ARQUITETURA ORIENTADA A SERVIÇOS

- Na década de 1990, o desenvolvimento Web revolucionou a troca de informações organizacionais.
- Clientes acessavam informações em servidores remotos fora de suas próprias organizações.
- Porém o acesso era possível somente por *browser* de Web.
  - Acesso às informações por outros programas não era possível.
- Surgiu a ideia de um *web service*.



# ARQUITETURA ORIENTADA A SERVIÇOS

- Usando um *web service*, as organizações que desejam disponibilizar suas informações para outros programas podem fazê-lo definindo e publicando uma interface de *web service*.
- Essa interface define os dados disponíveis e como eles podem ser acessados.
- *Web service* é um tipo de serviço, isto é, uma instância de uma ideia mais geral de serviço.
  - Um ato ou desempenho oferecido de uma parte para outra.



# ARQUITETURA ORIENTADA A SERVIÇOS

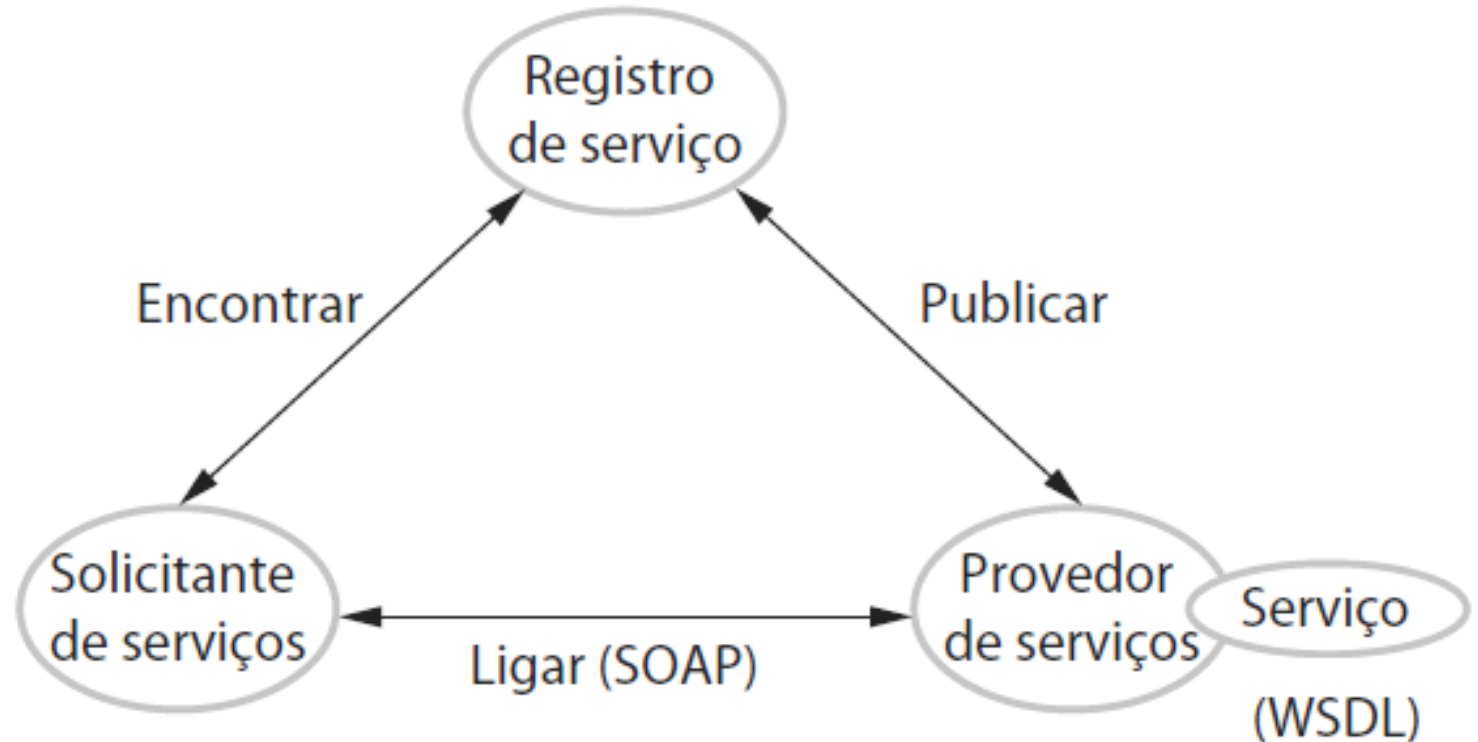
- Os provedores de serviços podem desenvolver serviços especializados e oferecê-los para uma variedade de usuários de serviço de diferentes organizações.
- As arquiteturas orientadas a serviços (**SOA**, *service-oriented architectures*) são uma forma de desenvolvimento de sistemas distribuídos.
  - Os componentes de sistema são serviços autônomos, executando em computadores geograficamente distribuídos.

# ARQUITETURA ORIENTADA A SERVIÇOS

- Protocolos-padrão baseados em XML foram projetados para oferecer suporte à comunicação de serviço e à troca de informações.
- Os serviços são plataforma e implementação independentes de linguagem.
- Sistemas de software podem ser construídos pela composição de serviços locais e serviços externos de provedores diferentes, com interação perfeita entre os serviços no sistema.

# ARQUITETURA ORIENTADA A SERVIÇOS

Arquitetura orientada a serviço





# PRINCIPAIS PADRÕES

- SOAP

- padrão de trocas de mensagens que oferece suporte à comunicação entre os serviços.

- WSDL

- Este padrão permite que uma interface de serviços 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.



# ARQUITETURA ORIENTADA A SERVIÇOS

- A construção de aplicações baseadas em serviços permite que empresas e outras organizações cooperem e façam uso das funções de negócios umas das outras.
- As SOA são arquiteturas menos rígidas, nas quais as ligações de serviços podem mudar durante a execução.
- Alguns sistemas serão construídos exclusivamente com o uso de *web services*, e outros misturarão *web services* com componentes desenvolvidos localmente.

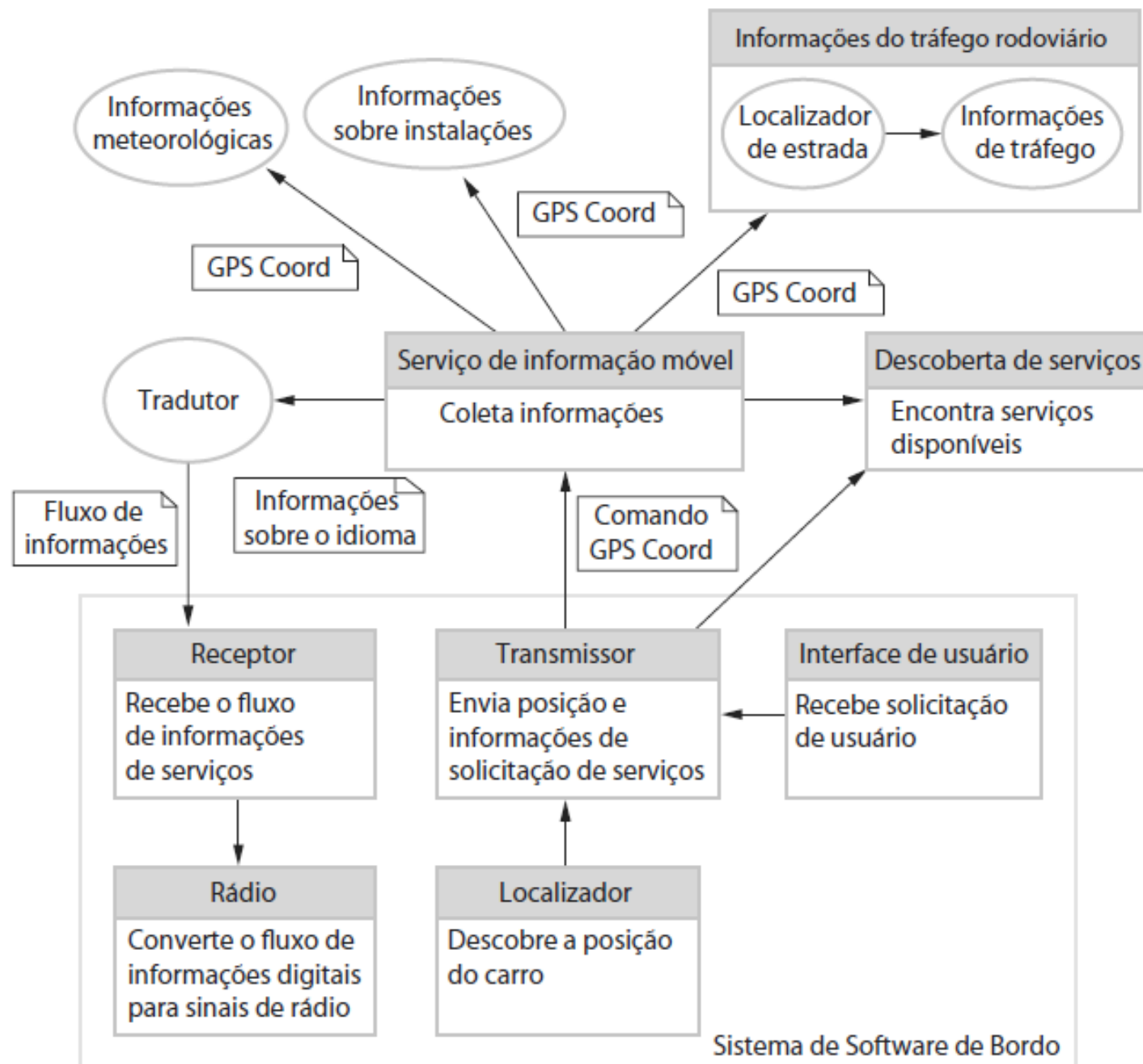




# CENÁRIO

- Exemplo de aplicações que usam uma mistura de componentes e serviços.
- Um sistema de informações em um carro fornece aos motoristas informações sobre clima, condições de tráfego da estrada, informações locais, e assim por diante. Ele é ligado ao rádio do carro para que a informação seja entregue como um sinal em um canal de rádio específico. O carro é equipado com receptores GPS para descobrir sua posição, e, com base nessa posição, o sistema acessa uma gama de serviços de informação. Em seguida, as informações podem ser entregues na linguagem especificada pelo motorista.

## Um sistema de informações de bordo baseado em serviços



# ARQUITETURA ORIENTADA A SERVIÇOS

- Esse exemplo ilustra uma das principais vantagens da abordagem orientada a serviços.
  - Não é necessário decidir quando o sistema é programado ou implantado, qual provedor de serviço deve ser usado ou quais serviços específicos devem ser acessados.
  - Conforme o carro se move, o software de bordo usa o serviço de descoberta de serviços para encontrar o serviço de informações mais adequado a ele.
  - Por causa do uso de um serviço de tradução, ele pode mover-se além das fronteiras e, portanto, disponibilizar informações locais para as pessoas que não falam a língua local.



# ARQUITETURA ORIENTADA A SERVIÇOS

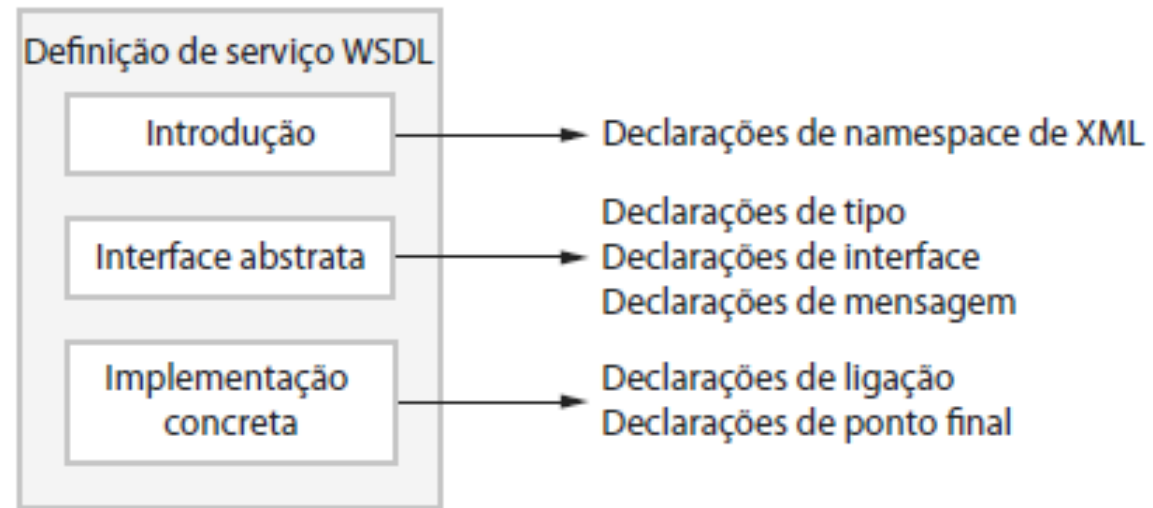
- As abordagens existentes para engenharia de software devem evoluir para refletir a abordagem orientada a serviços.
- Engenharia de serviços é o desenvolvimento de serviços confiáveis e reusáveis (desenvolvimento de software para reuso).
- Desenvolvimento de software com serviços é o desenvolvimento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

# SERVIÇOS COMO COMPONENTES REUSÁVEIS

- A interface de serviço é definida em uma descrição de serviço expressa em WSDL. A especificação WSDL define:
  - Quais operações o serviço suporta 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

# SERVIÇOS COMO COMPONENTES REUSÁVEIS

A organização de uma especificação WSDL



# SERVIÇOS COMO COMPONENTES REUSÁVEIS

- Uma parte introdutória que geralmente define os *namespaces* de XML usados e que o serviço pode incluir uma seção de documentação, fornecendo informações adicionais sobre o serviço.
- Uma descrição opcional dos tipos usados em mensagens trocadas pelo serviço.
- Uma descrição da interface de serviço, ou seja, as operações que fornecem o serviço para outros serviços ou usuários.
- Uma descrição das mensagens de entrada e saída processadas pelo serviço.
- Uma descrição da ligação usada pelo serviço, ou seja, o protocolo de mensagens que será usado para enviar e receber mensagens (SOAP).
- Uma especificação de ponto final que é o local físico do serviço, expresso como um identificador de recurso uniforme (URI).



# ENGENHARIA DE SERVIÇOS

- É o processo de desenvolvimento de serviços para reuso em aplicações orientadas a serviços.
- Os engenheiros de serviço precisam garantir que o serviço seja projetado como uma abstração reusável, que pode ser usada em sistemas diferentes.
- Devem documentar o serviço para que possam ser descobertos e compreendidos pelos usuários.

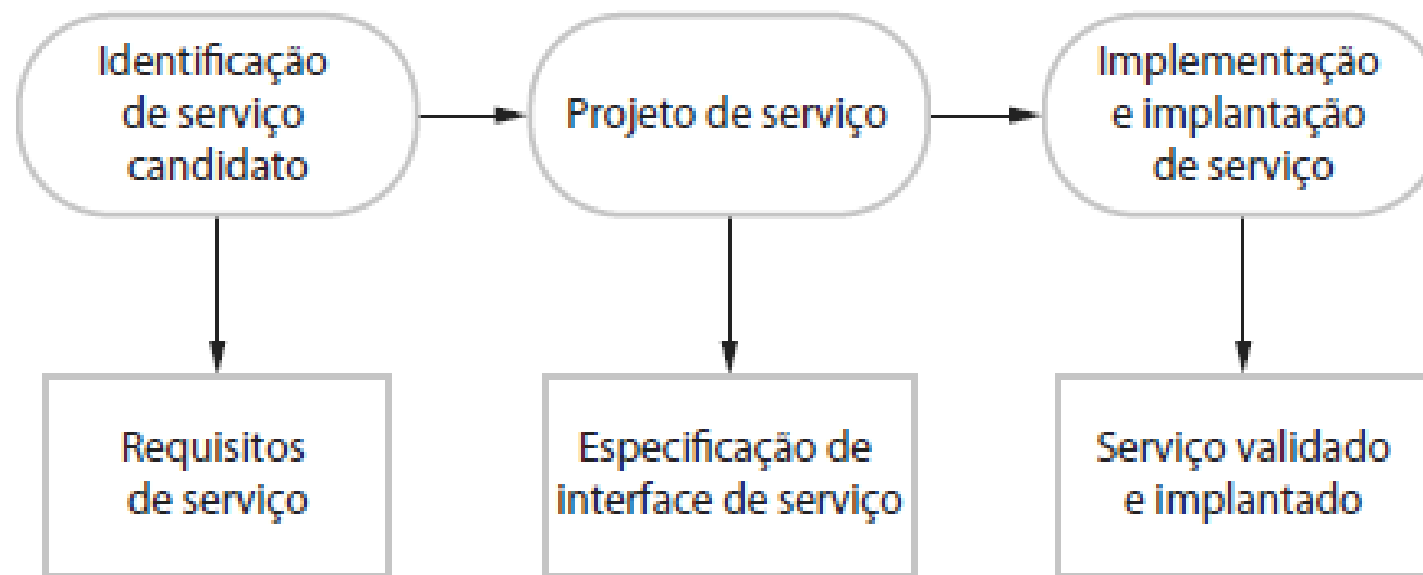


# ENGENHARIA DE SERVIÇOS

- Existem três estágios lógicos no processo de engenharia de serviços:
  - Identificação de serviço candidato, em que você identifica os possíveis serviços que podem ser implementados.
  - Projeto de serviço, em que projeta a lógica e as interfaces de serviço WSDL.
  - Implementação e implantação de serviço, em que você implementa e testa os serviços tornando-os disponíveis para uso.

# ENGENHARIA DE SERVIÇOS

O processo de engenharia de serviço



# IDENTIFICAÇÃO DE SERVIÇO CANDIDATO

- A noção básica da computação orientada a serviços é que os serviços deveriam apoiar os processos de negócios.
- Como cada organização tem uma ampla gama de processos, existem muitos serviços possíveis, que podem ser implementados.
- Portanto a identificação de serviços candidato envolve a compreensão e análise dos processos de negócios da organização para decidir quais serviços reusáveis serem implementados.

# IDENTIFICAÇÃO DE SERVIÇO CANDIDATO

- Três tipos fundamentais de serviço que podem ser identificados:
  - Serviços utilitários: que implementam funcionalidades gerais usada por diferentes processos de negócio, por exemplo, conversão de moedas.
  - Serviços de negócio: que são associados à uma função específica de negócio, por exemplo, registro de alunos de um curso em uma universidade.
  - Serviços de coordenação ou de processo: que apoiam processos compostos, por exemplo, serviço de pedidos, onde realiza o pedido, aceita o produto e o pagamento é efetuado.

# IDENTIFICAÇÃO DE SERVIÇO CANDIDATO

- Serviços orientados a tarefas:
  - São associados com alguma atividade.
- Serviços orientados a entidades:
  - São como objetos.

# IDENTIFICAÇÃO DE SERVIÇO CANDIDATO

A classificação de serviços

	Utilitário	Negócios	Coordenação
<b>Tarefa</b>	Conversor de moeda Localizador de funcionário	Validar formulário de reclamação Avaliar classificação de crédito	Processar despesas de reclamações Pagar fornecedor externo
<b>Entidade</b>	Verificador de estilo de documento Conversor de formulário Web para XML	Formulário de despesas Formulário de solicitação de estudante	

# IDENTIFICAÇÃO DE SERVIÇO CANDIDATO

- O objetivo na identificação de serviço é identificar serviços que são logicamente coerentes, independentes e reusáveis.
- Identificar candidatos a serviços pode ser difícil porque é necessário considerar como os serviços serão usados.
- É preciso pensar em possíveis candidatos e analisar uma série de questões sobre eles para verem se tendem a ser serviços úteis.

# IDENTIFICAÇÃO DE SERVIÇO CANDIDATO

- 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?



# IDENTIFICAÇÃO DE SERVIÇO CANDIDATO

- As respostas a essas questões ajudam a selecionar e refinar abstrações que podem ser implementadas como serviços.
- No entanto, não existe uma forma única e regular para decidir quais são os melhores serviços.
- A identificação é um processo baseado em experiência e habilidade.

# PROJETO DE INTERFACE DE SERVIÇO

- Após selecionar serviços candidatos, o próximo estágio do processo de engenharia de serviço é projetar as interfaces de serviço.
- Envolve a definição das operações associadas com o serviço e seus parâmetros.
- Deve-se pensar cuidadosamente sobre o projeto de operações e mensagens de serviço.

# PROJETO DE INTERFACE DE SERVIÇO

- Seu objetivo deve ser minimizar o número de trocas de mensagens que deve acontecer para se completar a solicitação de serviço.
- É necessário garantir que tanta informação quanto possível seja passada para o serviço por uma mensagem, em vez de interações síncronas de serviço.
- Deve se lembrar que serviços não têm estado.
  - Gerenciar estado é responsabilidade do usuário do serviço

# PROJETO DE INTERFACE DE SERVIÇO

- Existem três estágios de projeto de interface de serviço:
  - **Projeto de interface lógica:** em que você identifica as operações associadas ao serviço, suas entradas e saídas e as exceções associadas a essas operações.
  - **Projeto de mensagem:** em que você cria a estrutura das mensagens que são enviadas e recebidas pelo serviço.
  - **Desenvolvimento WSDL:** em que você traduz o projeto lógico e de mensagem para uma descrição da interface abstrata escrita em WSDL.

# IMPLEMENTAÇÃO E IMPLANTAÇÃO DE SERVIÇO

- O estágio final do processo de engenharia de serviço é a implementação 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.

# IMPLEMENTAÇÃO E IMPLANTAÇÃO 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.

# DESENVOLVIMENTO DE SOFTWARE COM SERVIÇOS

- O desenvolvimento de software usando serviços baseia-se na ideia de compor e configurar serviços para criar novos serviços compostos.
- Podem ser integrados com uma interface de usuário implementada em um *browser* para criar uma aplicação web.
- Podem ser usados como componentes em alguma outra composição de serviço.

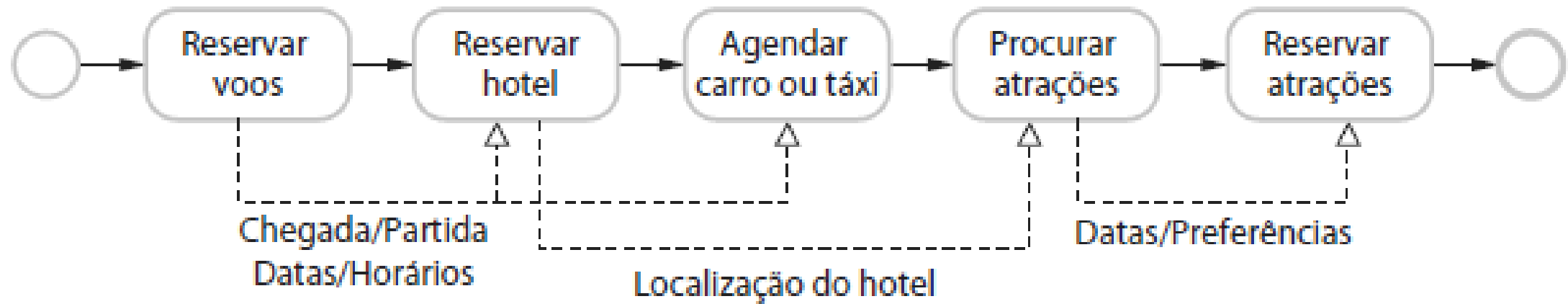


# DESENVOLVIMENTO DE SOFTWARE COM SERVIÇOS

- A composição de serviços pode ser usada para integrar os processos de negócios separados para fornecer um processo integrado oferecendo funcionalidade mais ampla.
  - Por exemplo, fornecer serviços de reserva de viagens que permitem que reservas de voos, aluguel de carro e hotel sejam coordenados.
- A base para composição do serviço é, frequentemente, um *workflow*.
  - *Workflows* são sequências lógicas de atividades que, juntos, modelam um processo coerente de negócio.

# DESENVOLVIMENTO DE SOFTWARE COM SERVIÇOS

*Workflow do pacote de férias*

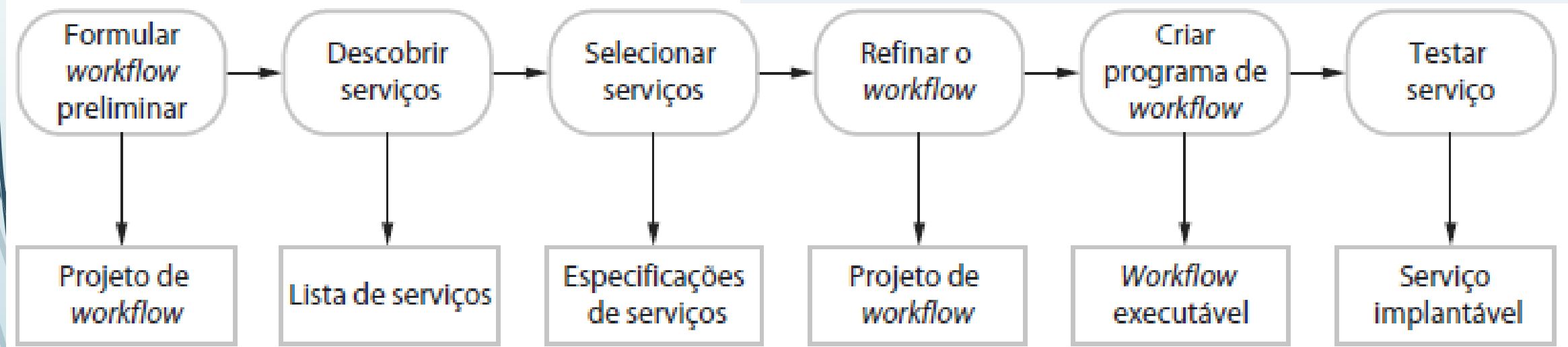


# DESENVOLVIMENTO DE SOFTWARE COM SERVIÇOS

- Um *workflow* é uma ideia simples, e o cenário de fazer reservas para férias parece ser simples.
- Na prática, a composição de serviço é muito mais complexa do que esse modelo simples representa.
- Por exemplo, deve considerar a possibilidade de falha de serviço e incluir mecanismos para lidar com essas falhas.

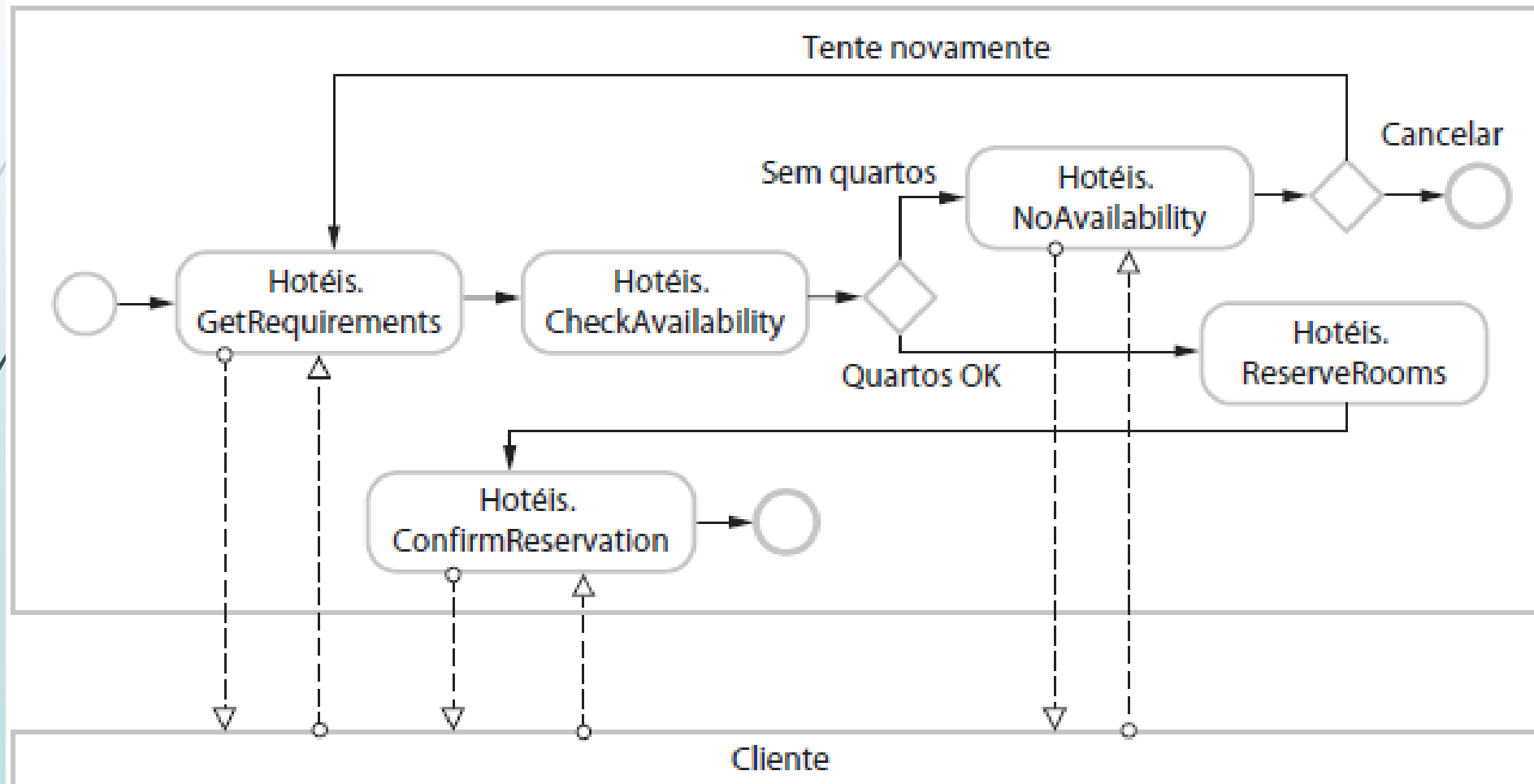
# DESENVOLVIMENTO DE SOFTWARE COM SERVIÇOS

## Construção de serviço por composição



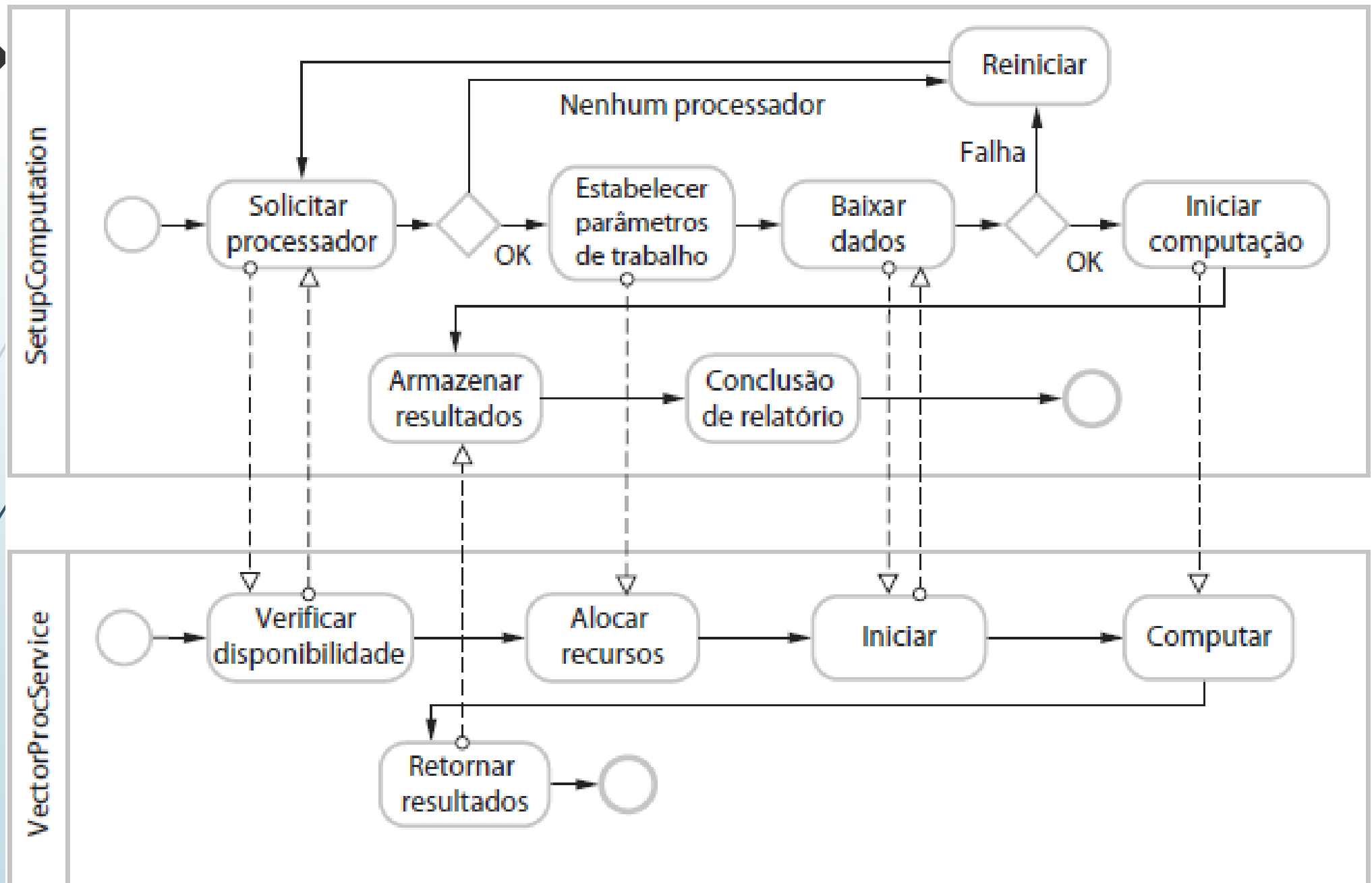
# DESENVOLVIMENTO DE SOFTWARE COM SERVIÇOS

Um fragmento de um *workflow* de reserva de hotel



# DESENVOLVIMENTO DE SOFTWARE COM SERVIÇOS

- 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.



# 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.



# 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.
- 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.