



SOA

Service Oriented Architecture

Fabiano Oss

fabiano.oss@gmail.com



Roteiro

- SOA
- Serviços
- Tecnologias para o desenvolvimento de serviços
- Modelagem de Negócios

O que é SOA

- “É uma arquitetura de desenvolvimento cujo objetivo é criar módulos funcionais chamados de serviços, com baixo acoplamento e permitindo reutilização de código”

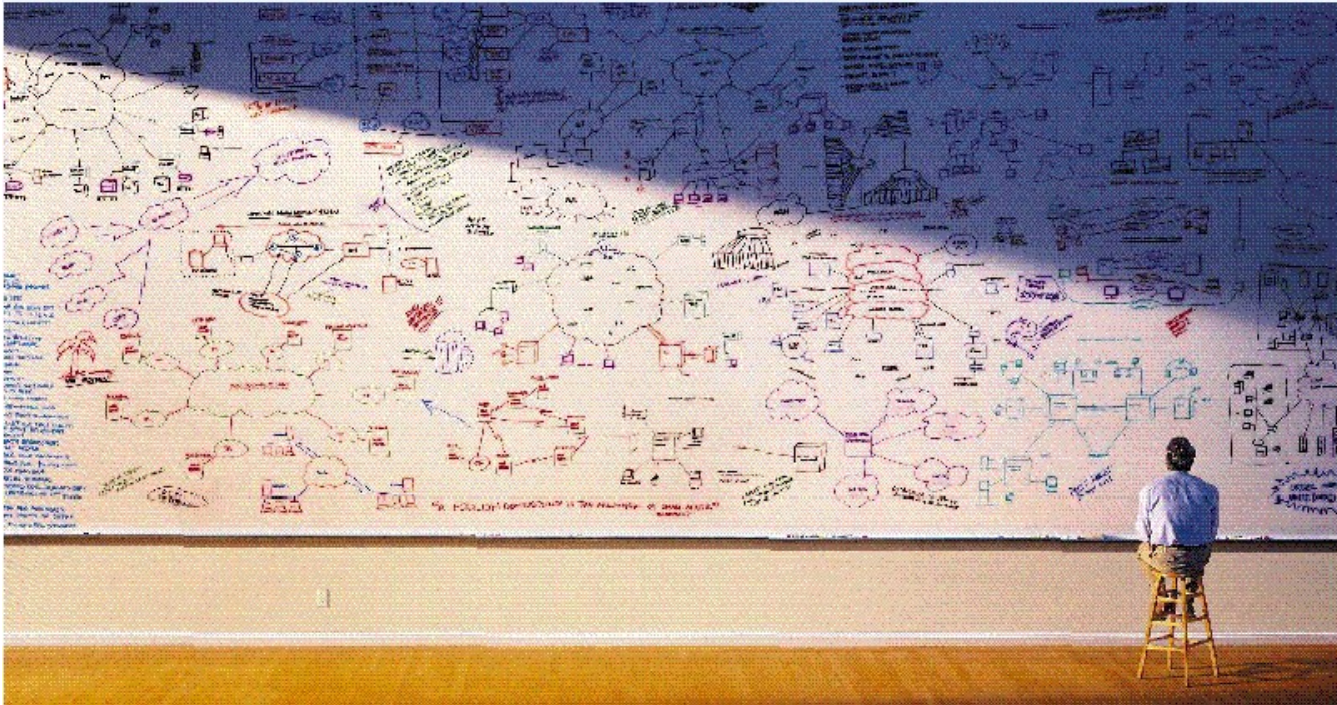
SAMPAIO, Cleuton. SOA e Web Services em Java, Rio de Janeiro, 2006

Objetivos

- Foco na melhoria e agilidade de negócios
- Reutilização de componentes em forma de serviços
- Alinhar o TI as estratégias de negócio
- Cria aplicações com base numa fundação de componentes
- Integrar aplicações

Integrar para quê?

- Não é melhor reescrever tudo?



O que é um Serviço?

- É uma função do sistema computacional construído de tal forma que possa ser facilmente vinculado a outros componentes de software
- Encapsula uma função do negócio reutilizável
- Serviços são independentes da implementação
- Serviços são fracamente interligados



Características

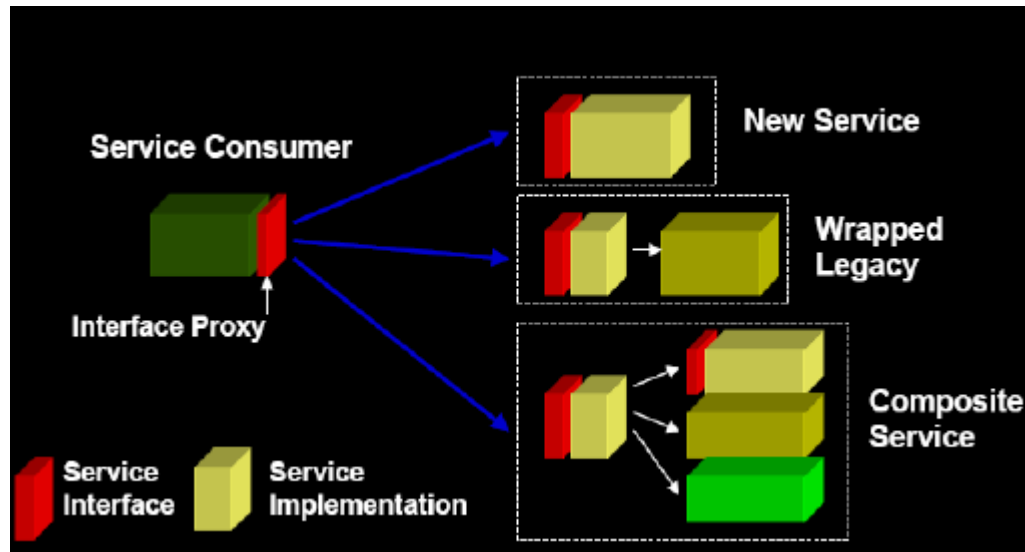
- Não existem herança ou dependência entre os serviços
- Separação clara entre fornecedor (provider) e consumidor (consumer)
- Foco na definição dos contratos

Características

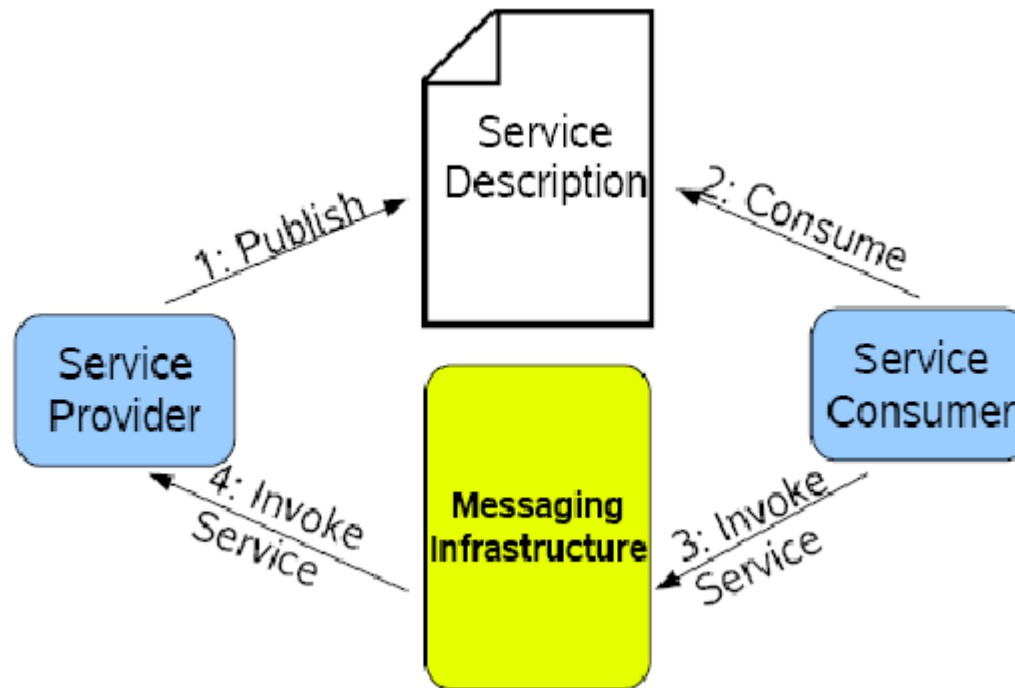
- Serviços podem ser compostos e orquestrados em processos de negócio;
- A arquitetura SOA não está ligada a uma tecnologia específica.
- SOA está associado a uma série de tecnologias: Web Services, BPEL, UDDI, XML, http, entre outras

Serviço

- Serviço = Contrato + Implementação



Características de implementação dos serviços



Propriedades

- Granularidade alta
- Fracamente interligados
- Ligação dinâmica
- Auto contidos e modulares
- Interfaces endereçáveis na rede
- Podem ser compostos
- Induzem a interoperação
- Permitem recuperação de falhas
- Serviço expressa a visão de negócio da aplicação



Acoplamento e coesão

- Acoplamento: é a medida de interdependência entre os módulos
- Coesão: é a medida de intensidade de associação funcional dos elementos

Graus de coesão

- Funcional: atividades internas estão relacionadas e servem para um propósito
- Seqüencial: a saída de um atividade é entrada de outra (linha montagem)
- Comunicacional: as atividades estão ligadas por um dado comum
- Procedural: as atividades podem ou não ter relação, mas devem ser executadas juntas
- Temporal: semelhante a procedural, só que em função do tempo;
- Lógica: as tarefas executadas dependendo de uma entrada um flag;
- Conincidental: as tarefas estão lá por que não tinha lugar melhor para colocá-las

Acoplamento

- Dados: comunicação entre módulos é feita através de parâmetros;
- Imagem: os módulos compartilham uma área comum;
- Controle: um módulo pode controlar a lógica de outro;
- Comum: semelhante ao de imagem, só ocorre quando todos os módulos compartilham uma área comum;
- Conteúdo: desvios incondicionais (goto)

Acoplamento

- “Acoplamento fraco pode ser aplicado a qualquer ligação livre de atrito, possibilitada por arquiteturas abertas. Serviços fracamente acoplados, mesmo usando tecnologias incompatíveis, podem ser unidos sob demanda para criar serviços compostos ou desunidos facilmente em seus componentes funcionais”

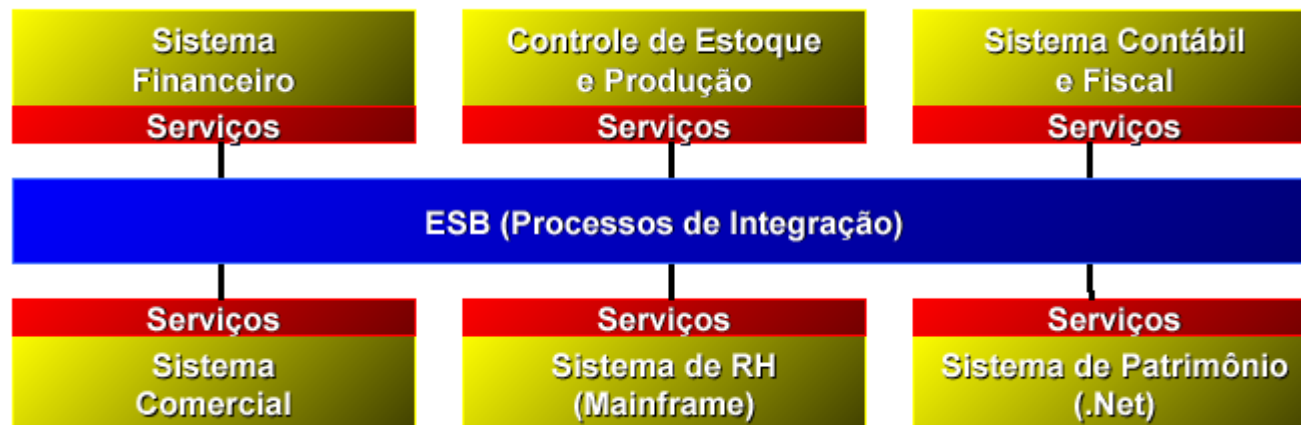
(en.wikipedia.org)

Vantagens

- Baixo acoplamento entre aplicações
- Alta interoperabilidade entre plataformas tecnológicas
- Alta reutilização de regras de negócio
- Resposta mais rápida nos processos de negócio
- Facilidade em executar testes nos serviços

SOI – Service Oriented Integration

- ESB



Orquestração

- É composto por um fluxo de etapas, com verificações de pré e pos condições e um coordenador;

BPEL

- Business Process Executaion Language
- Descreve workflows de um ESB
- Linguagem XML
- Ferramentas gráficas que auxiliam o desenho
 - Oracle BPEL Process Manager (<http://www.oracle.com/technology/products/ias/bpel/index.html>)
 - IBM WebSphere Business Integration Server Foundation (<http://www.ibm.com/software/integration/wbisf>)
 - IBM alphaWorks BPWS4J (<http://www.alphaworks.ibm.com/tech/bpws4j>)
 - OpenStorm Service Orchestrator (<http://www.openstorm.com>)
 - Vergil VCAB Server (http://www.vergiltech.com/products_VCAB.php)
 - Active Endpoints ActiveWebflow Server (<http://www.active-endpoints.com/products/index.html>)
 - ActiveBPEL engine (<http://www.activebpel.org/>)
 - Fivesight Process eXecution Engine (<http://www.fivesight.com/pxe.shtml>)

```

-<partnerLinks>
  <partnerLink myRole="LoanBrokerProvider" name="client"
    partnerLinkType="tns:LoanBroker"
    partnerRole="LoanBrokerRequester" />
  <partnerLink myRole="LoanServiceRequester" name="LoanService"
    partnerLinkType="services:LoanService"
    partnerRole="LoanServiceProvider" />
</partnerLinks>
-<variables>
  <variable name="input"
    messageType="tns:LoanBrokerRequestMessage" />
  <variable name="output"
    messageType="tns:LoanBrokerResultMessage" />
  <variable name="request"
    messageType="services:LoanServiceRequestMessage" />
  <variable name="response"
    messageType="services:LoanServiceResultMessage" />
</variables>
-<sequence>

  <!-- receive input from requestor -->
  <receive createInstance="yes" name="receiveInput"
    operation="initiate" partnerLink="client"
    portType="tns:LoanBroker" variable="input" />

  <!-- initialize the input of LoanService -->
  -<assign>
    -<copy>
      <from part="payload" variable="input" />
      <to part="payload" variable="request" />
    </copy>
  </assign>

  <!-- initiate the remote process -->
  <invoke inputVariable="request" name="invoke" operation="initiate"

```

PartnerLink

References to the services participating in the process flow and their role/port types

Variables

List of messages exchanged between the BPEL process and each of the participating Web Services.

Flow Logic

receive, invoke, assign, pick, wait, throw, terminate, flow, sequence, switch Scope, fault handlers, ...

[Fault Handler]

Catch and handle faults

[Compensation Handler]

Undo logic



O que serviços podem fazer

- transformar um dado
- rotiar mensagens
- acesso em banco de dados
- orquestrar conversações
- aplicar uma logica de negócio
- solicitar aprovações
- EDI

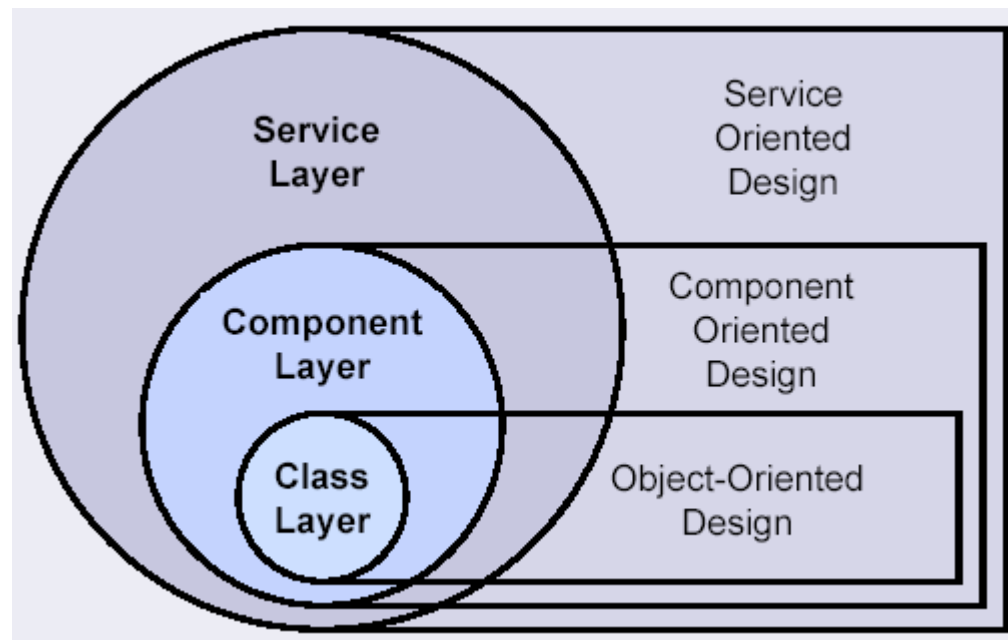


Exemplo de serviços

- Processamento de uma pagamento
- Gerenciamento de clientes
- Efetuar consultas de crédito
- Verificar a disponibilidade de um produto em estoque
- etc.

SOA = OOAD

- Granularidade
- Nível de abstração elevado





Unidade de desenvolvimento

- OO: Classe
- SOA: Serviço

Comparativo com OOAD

Conceito	OOAD	SOA
Unidade de desenvolvimento	Classes	Serviços
Foco principal	Funcionalidade	Processo de negócios
Acesso	Interno a empresa	Externo a empresa
Visão	Visão de aplicação (horizontal)	Visão trans-aplicações (vertical)
Acoplamento entre componentes	Forte	Fraco
Manutenção de estado	Stateful	Stateless
Ciclo de desenvolvimento	+ Longo	Incremental
Build	Produto rígido	Produto mais flexível



Componente X Serviço

- Granularidade
- Acoplamento
- Transação
- Estado

Quando não usar SOA

- Não podemos desenvolver um sistema só com SOA
- A empresa só tem 1 sistema crítico
- Maior complexidade
- Maior envolvimento entre gerente de negócios e área técnica
- Velocidade
- Segurança

Passos para implementação de SOA (proposta)

- Treinamento
- Definição dos papéis
- Identificação e criação de serviços
- Modelagem de processos
- Orquestração dos serviços
- Reuso dos serviços
- Revisão
- Monitoramento
- Fechamento



Tecnologias para o Desenvolvimento de Serviços

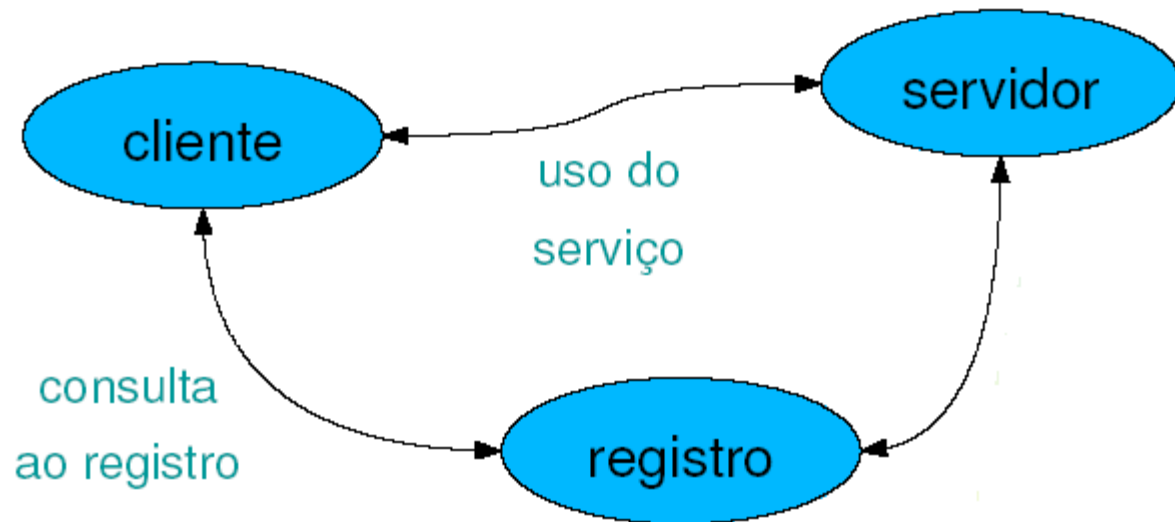
Tecnologias para programação distribuída

○ RMI

○ DCOM

○ CORBA

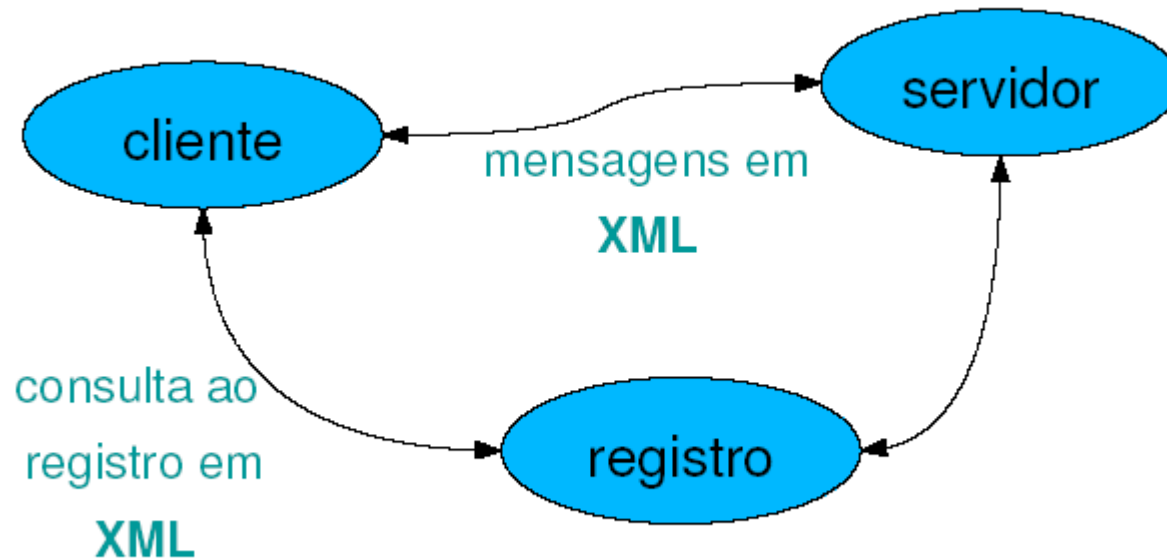
○ Web Service



Tecnologias

	Java RMI	CORBA	Web Services
Registro	RMI Registry	COS Naming	UDDI
Descrição do Serviço	Java	IDL	WSDL
Transporte	Java RMI	IIOP	SOAP

Web Services



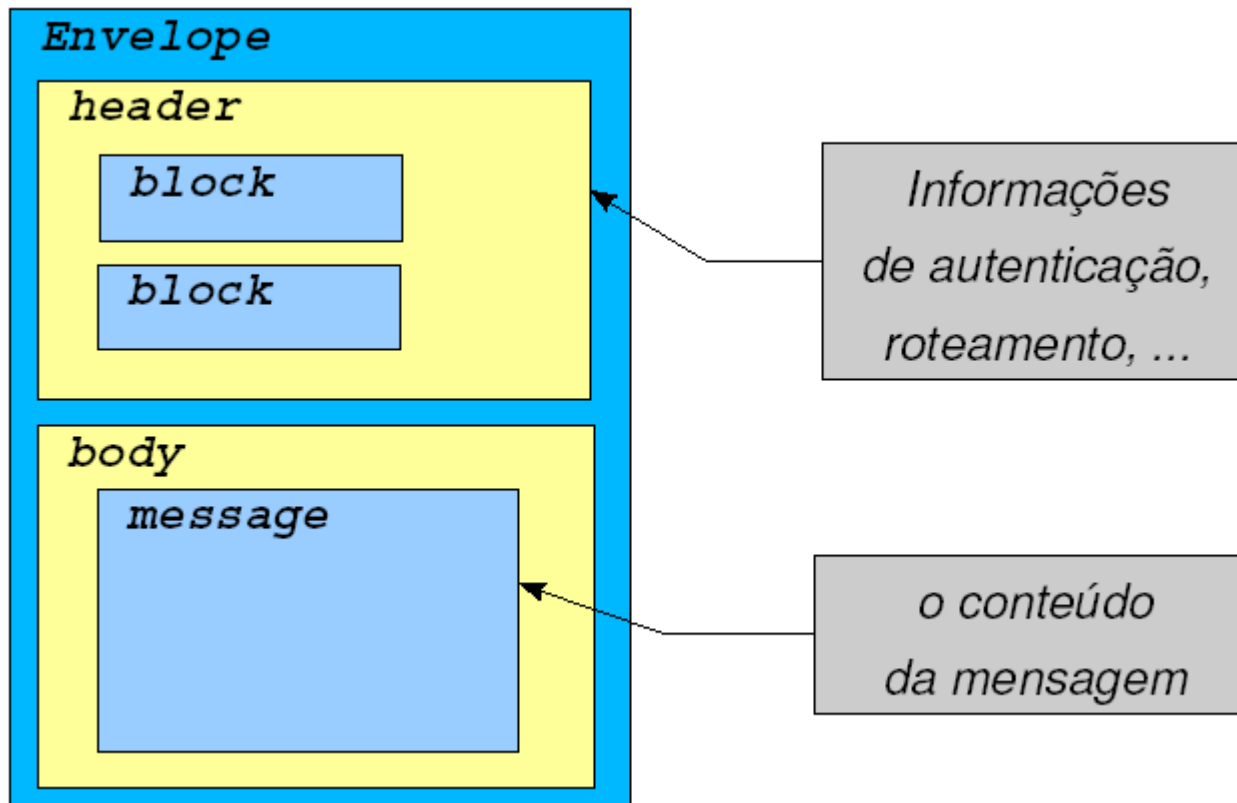
Só precisa XML

- Padrões
- Descrição dos serviços (WSDL)
- Comunicação (SOAP)
- Consulta de Serviços (UDDI)

SOAP

- Simple Object Access Protocol
- Protocolo de transporte

Mensagem SOAP



Exemplo de SOAP

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://www.w3c.org/2003/05/soap=envelope"
  soap:encodingStyle="http://www.w3c.org/2003/05/soap-
    encoding">

  <soap:Header>
  </soap:Header>

  <soap:Body>
  </soap:Body>

</soap:Envelope>
```



SOAP Header

- Informações de controle da mensagem XML
- Segurança

Exemplo

- <?xml version="1.0" encoding="UTF-8"?>
- <soap:Envelope
- xmlns:soap="http://www.w3c.org/2003/05/soap=envelope"
- soap:encodingStyle="http://www.w3c.org/2003/05/soap-encoding">

- <soap:Header>
- <x:idAgente
- xmlns:m="http://xpto.org/teste"
- soap:mustUnderstand="true">
- <x:codigo>
- 10
- </x:codigo>
- </x:idAgente>
- </soap:Header>

- <soap:Body>
- </soap:Body>

- </soap:Envelope>



SOAP Body

- Corpo da mensagem
- Request
- Response
- Fault

Exemplo SOAP Body (Request)

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://www.w3c.org/2003/05/soap=envelope"
  soap:encodingStyle="http://www.w3c.org/2003/05/soap-encoding">

  <soap:Header>
  ...
  </soap:Header>

  <soap:Body>

    <c:transferir
      xmlns:c="http://www.oss.com.br">
      <c:contaOrigem>123</c:contaOrigem>
      <c:contaDestino>456</c:contaDestino>
      <c:valor>250.00</c:valor>
    </c:transferir>

  </soap:Body>

</soap:Envelope>
```


Exemplo SOAP Body (Response)

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://www.w3c.org/2003/05/soap=envelope"
  soap:encodingStyle="http://www.w3c.org/2003/05/soap-encoding">

  <soap:Header>
  ...
  </soap:Header>

  <soap:Body>

    <c:transferirResponse
      xmlns:c="http://www.oss.com.br">
      <c:sucesso>>true</c:sucesso>
      <c:dataDisponibilidade>2007-08-08</c:dataDisponibilidade>
      <c:contaOrigem>123</c:contaOrigem>
      <c:contaDestino>456</c:contaDestino>
      <c:valor>250.00</c:valor>
    </c:transferirResponse>

  </soap:Body>

</soap:Envelope>
```

SOAP Fault

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://www.w3c.org/2003/05/soap=envelope"
  soap:encodingStyle="http://www.w3c.org/2003/05/soap-encoding">
  <soap:Header>
  ...
  </soap:Header>
  <soap:Body>
    <soap:Fault>
      <soap:Code>
        <soap:Value>soap:Sender</soap:Value>
        <soap:Subcode>
          <soap:Value>rpc:BadArguments</soap:Value>
        </soap:Subcode>
      </soap:Code>
      <soap:Reason>
        <soap:Text xml:lang="pt-BR">Sem saldo</soap:Text>
      </soap:Reason>
      <soap:Detail>
        <bc:erro
          xmlns:bc="http://oss.com.br/falha">
          <bc:mensagem>Sem saldo na conta origem</bc:mensagem>
        </bc:erro>
      </soap:Detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

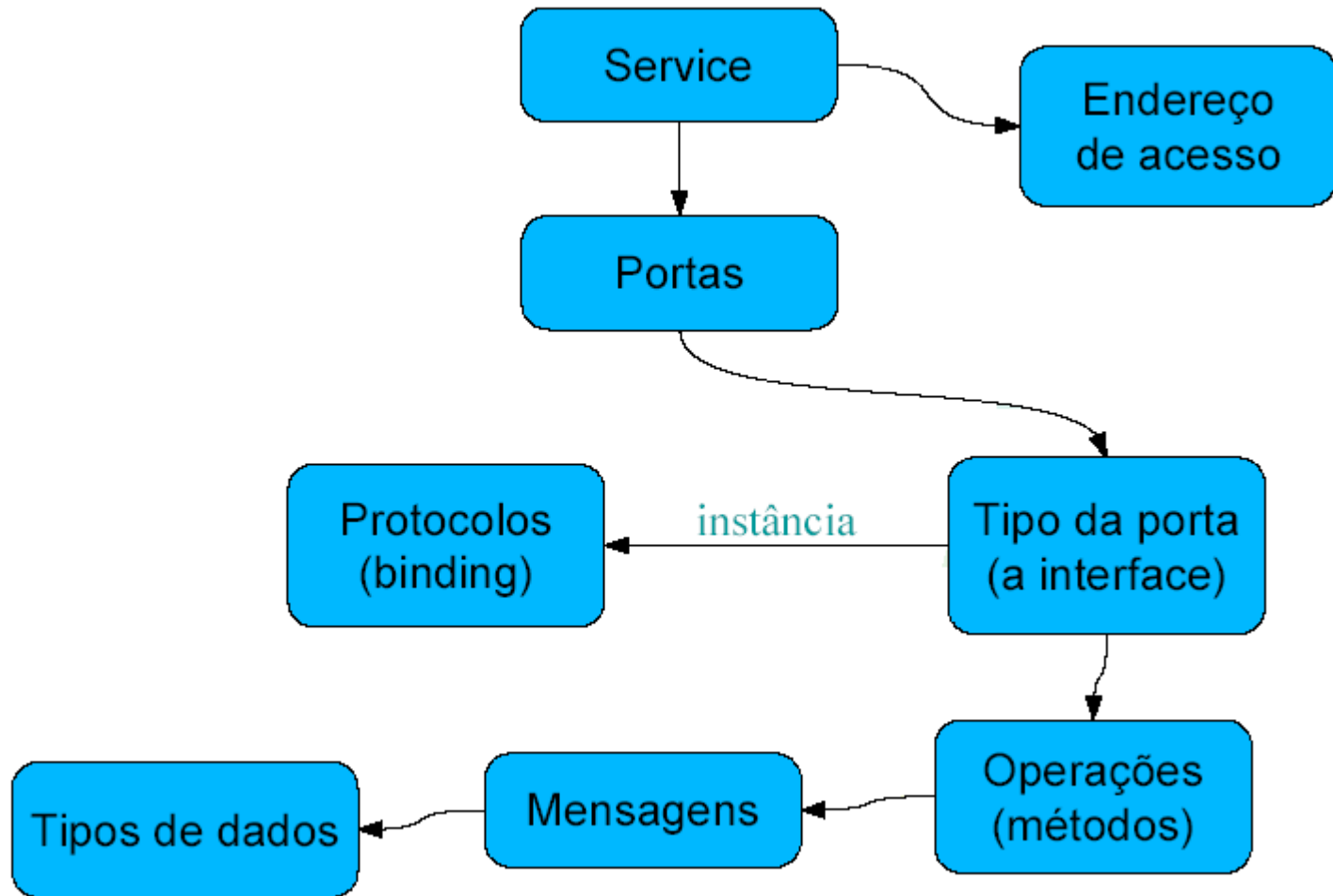
SOAP Bindings

- Variedade de protocolos de transporte
- HTTP: GET e Post
- GET / banco HTTP/1.1
- Host: www.xpto.com.br
- Accept: application/soap+xml

WSDL

- Web Service Description Language
- Documento XML que contém todas as informações necessárias para que um cliente possa utilizar um WS

Conteúdo de um WSDL



WSDL

- Tipo de dados (Types): definição de dados usado nas mensagens
- Mensagem (Message): definição abstrata dos dados sendo trocados
- Operação (Operation): definição abstrata de uma ação do WS;
- Tipo da Porta (Port Type): conjunto abstrato de operações suportados por uma ou mais portas
- Binding: uma especificação concreta de protocolo e formato de dados para um Port Type
- Porta (Port ou Endpoint): um único endpoint formado pela combinação de um Binding e um endereço de rede
- Service: coleção de portas;

Resumindo

- WSDL é
- uma coleção de endpoints (ports)
- que permite algumas operações (operations)
- e cada operação implica na troca de mensagens (messages)
- que são formadas por tipo (types)
- tudo isso definido num xml

Exemplo

- Dado o seguinte método

- String digaOi(String nome)

documentação

```
<definitions name="OiMundoWS"
```

namespace do WS

```
targetNamespace="urn:posWS/wsdl/Oi"
```

```
xmlns:tns="urn:posWS/wsdl/Oi"
```

namespace default

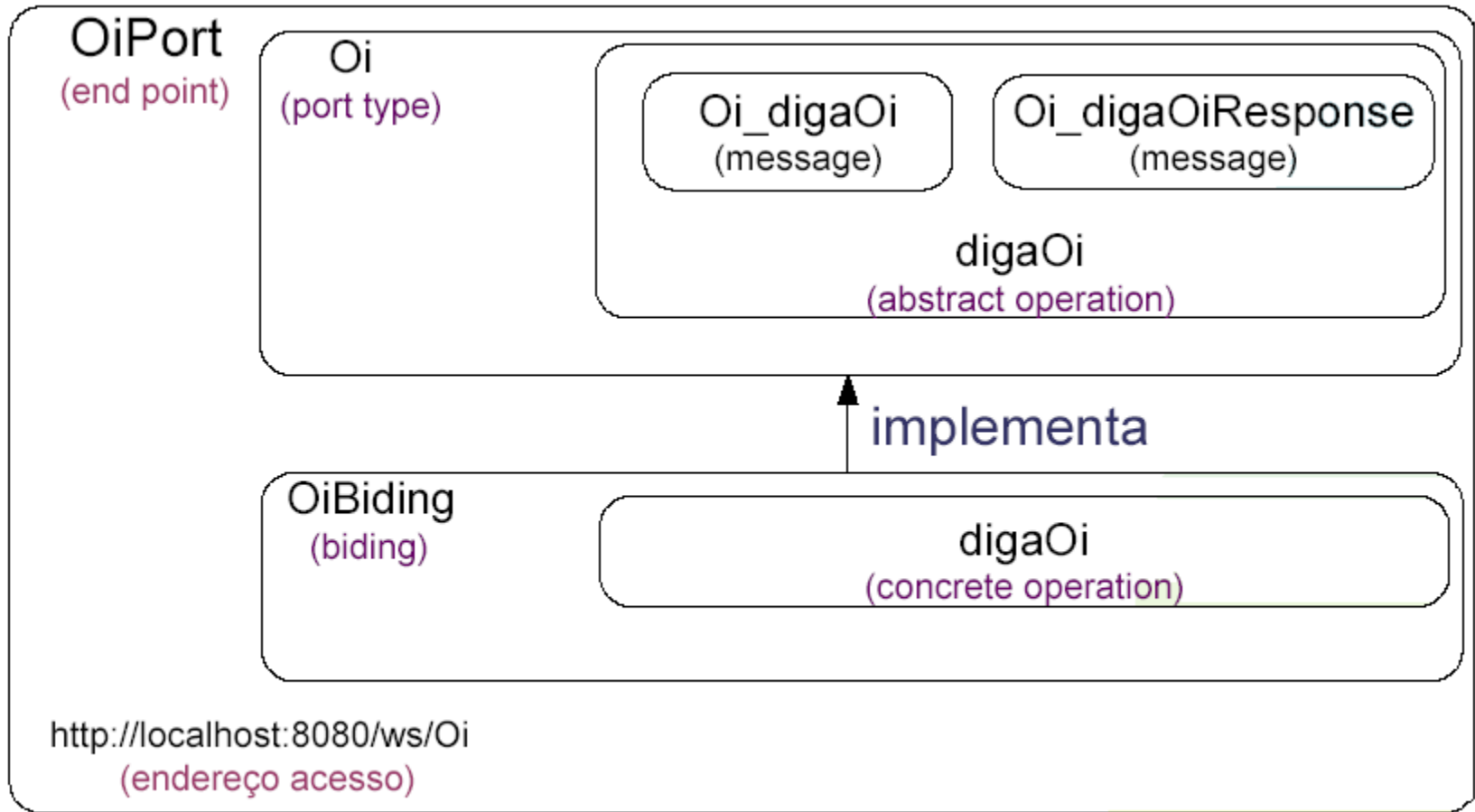
```
xmlns="http://schemas.xmlsoap.org/wsdl/"
```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

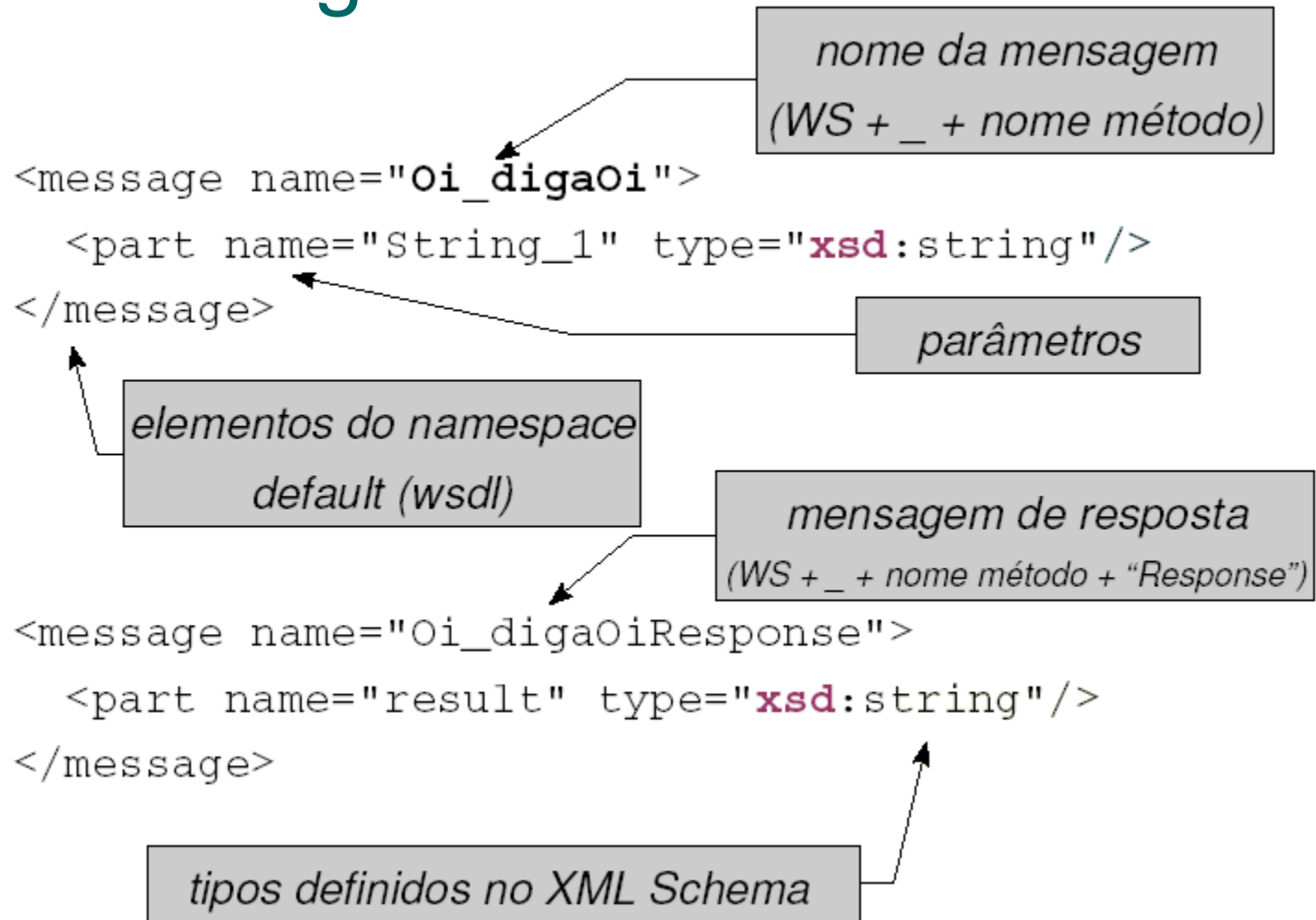
```
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
```

```
>
```

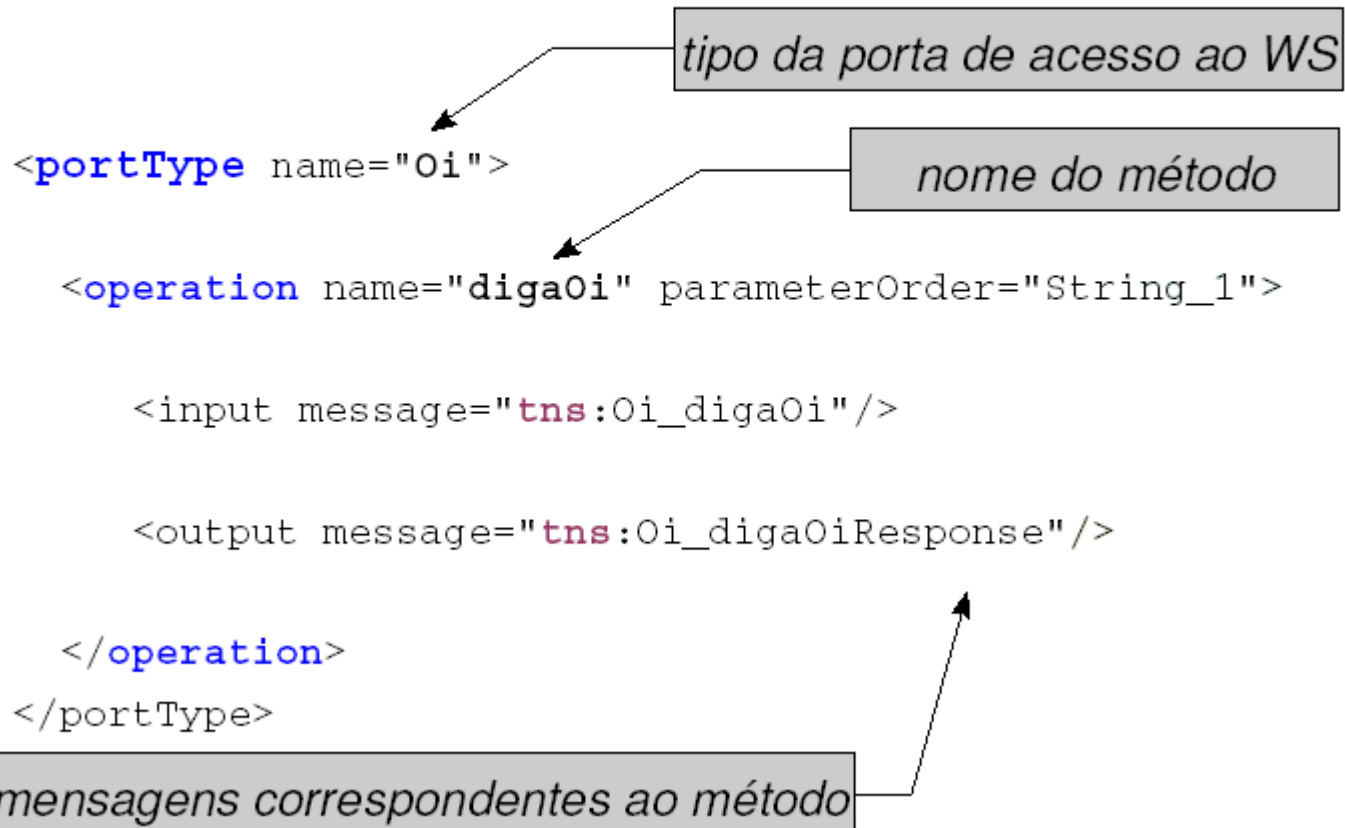

Exemplo



Mensagens



Tipo da Porta e Operação



Binding

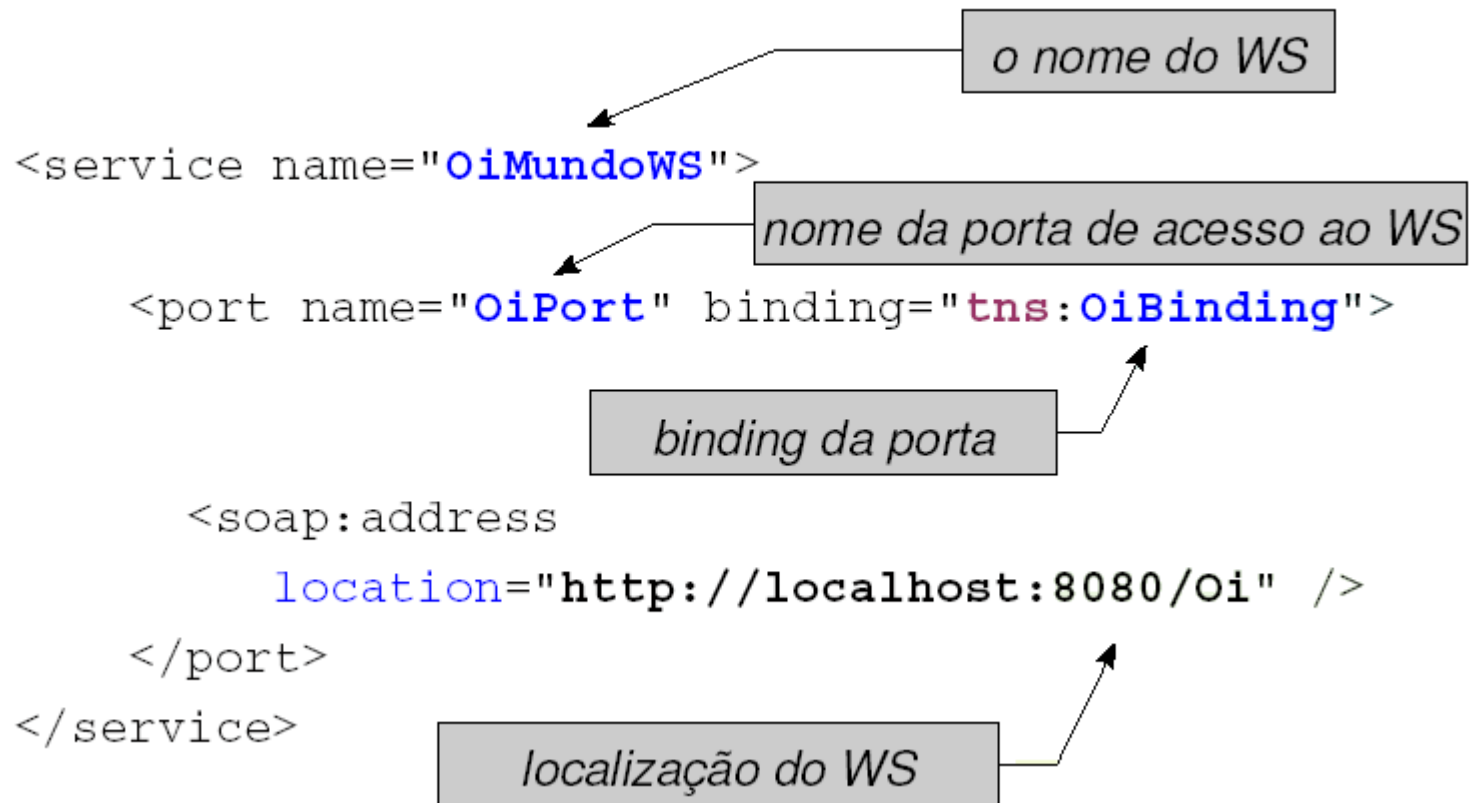
```
<binding name="OiBinding" type="tns:Oi">
  <soap:binding
    transport="http://schemas.xmlsoap.org/soap/http"
    style="rpc"/>
  <operation name="digaOi">
    <soap:operation soapAction="" />
    <input>
      <soap:body
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        use="encoded" namespace="urn:posWS/wsdl/Oi" />
    </input>
    <output>
      <soap:body
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        use="encoded" namespace="urn:posWS/wsdl/Oi" />
    </output>
  </operation>
</binding>
```

tipo da porta

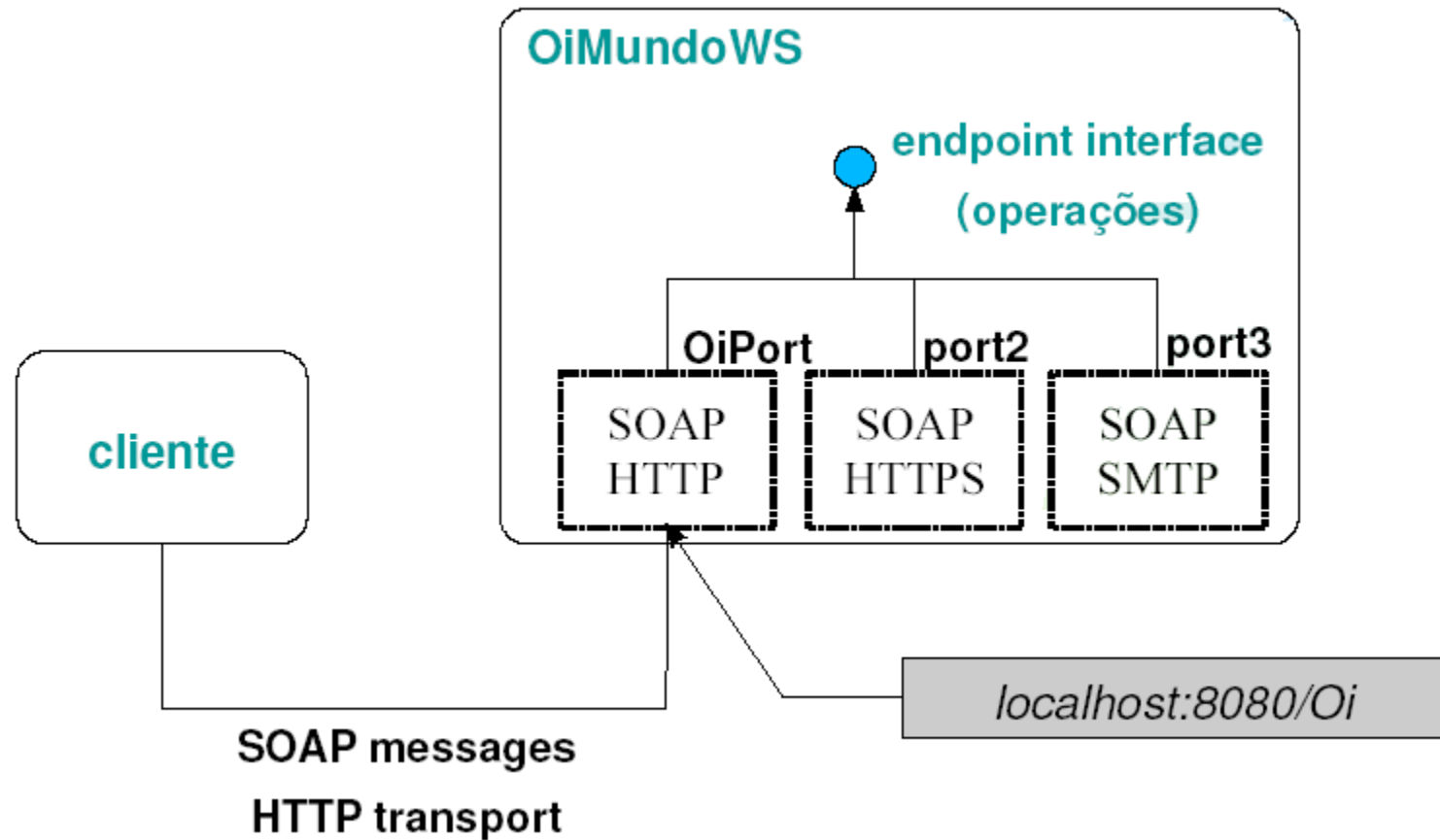
protocolo SOAP sobre HTTP - RPC

*como os dados são codificados
(segundo o padrão SOAP)*

Service



Visão geral de funcionamento



Java e Web Services

- JAX-RPC
 - Extensão do RMI
- JAX-WS
 - Especificação JEE 5.0

JAX-WS

- Java API for XML Web Services
- Utiliza anotações para gerar a WSDL
- Faz parte do JDK 6.0

Anotações (JRS 181)

- @WebService: Indica que a classe é um Web Service
- @WebMethod: Indica que o método anotado é uma função
- @WebServiceRef: Permite o cliente referenciar um serviço descrito no WSDL

Exemplo de Web Service

- **package** service;
- **import** javax.jws.WebMethod;
- **import** javax.jws.WebParam;
- **import** javax.jws.WebService;
- @WebService
- **public class** Teste {
- @WebMethod
- **public** String getPrimeiro() {
- **return** "xxx";
- }
- @WebMethod
- **public** String getSegundo(@WebParam(name="s1") String s1) {
- **return** "segundo " + s1;
- }
- @WebMethod
- **public double** getSoma(**double** d1, **double** d2){
- **return** d1 + d2;
- }
- }

XFire

- SOAP Framework
- Suporte a SOAP, WSDL, WS-Security, etc.
- Projeto irá chamar-se Apache CXF



Modelagem de Negócio

Modelo de negócio

- **representação de uma organização** real que servem como um referência comum para todos os seus membros, sejam eles pessoas, sistemas ou recursos
- tipo específico de modelo formado por um conjunto de **modelos que procuram representar as diferentes visões da empresa.**

Modelo de negócio

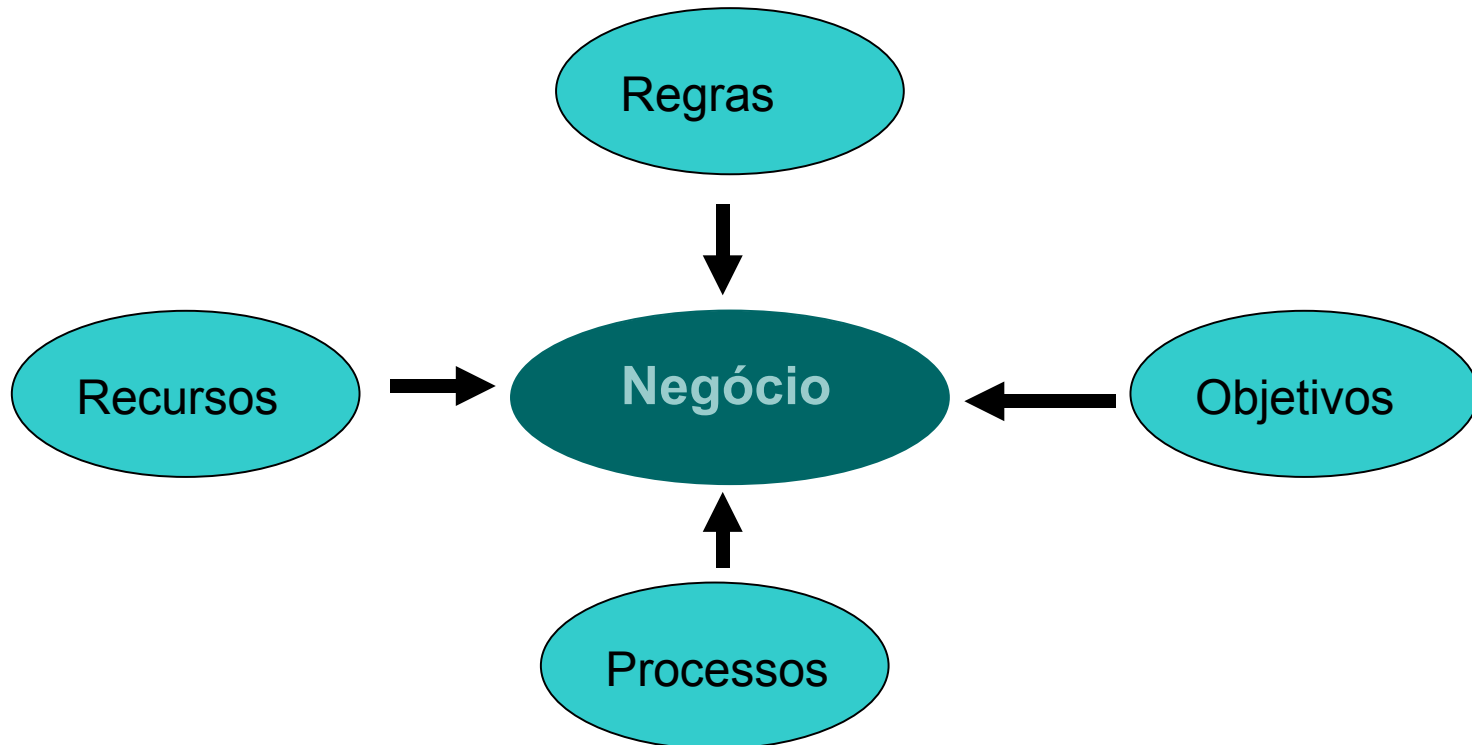
- Mostrar qual é o **ambiente da organização** e como a organização **age em relação a este ambiente.**

Processos

- Conjunto de elementos que agregam valor aos insumos usados para produzir o bem ou serviço e têm por finalidade atender às expectativas do cliente.

Negócio

- Conjunto de atividades, visando um objetivo



Metodologias e Técnicas

- Rup
- Erikson e Penker
- Técnica do i*
- EKD (Enterprise Knowledge Development)
- GP da IBM
- Mac Knigth
- Sharp – Mc Dermott
- BPMn

BPMN

- Business Process Modeling Notation
- Modelo de negócio que pode ser mapeado para um formato de execução, como BPEL4WS
- Assemelha-se aos diagramas de atividades da UML
- Pode ter granularidades variadas



Simbologia do BPMN




- Um dos objetivos da BPMN é criar um mecanismo simples para o desenvolvimento dos modelos processos de negócio, e ao mesmo tempo poder garantir a complexidade inerente aos processos.

Simbologia do BPMN




Quatro categorias básicas de elementos:

- Objetos de Fluxo
- Objetos de Conexão
- *Swimlanes*
- Artefatos

Objetos de Fluxo


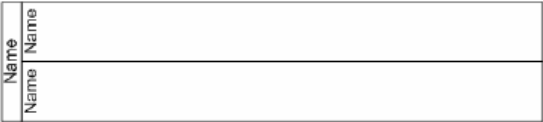
Objeto	Descrição	Figura
Evento	É algo que acontece durante um processo do negócio. Estes eventos afetam o fluxo do processo e têm geralmente uma causa (trigger) ou um impacto (result). Há três tipos de eventos, baseados sobre quando afetam o fluxo: Start, Intermediate, e End.	
Atividade	É um termo genérico para um trabalho executado. Os tipos de atividades são: Tarefas e sub-processos. O sub-processo é distinguido por uma pequena cruz no centro inferior da figura.	
Gateway	É usado para controlar a divergência e a convergência da seqüência de um fluxo. Assim, determinará decisões tradicionais, como juntar ou dividir trajetos.	

Objetos de Conexão

Objeto	Descrição	Figura
Fluxo de seqüência	É usado para mostrar a ordem (seqüência) com que as atividades serão executadas em um processo.	
Fluxo de mensagem	É usado para mostrar o fluxo das mensagens entre dois participantes diferentes que os emitem e recebem.	
Associação	É usada para associar dados, texto, e outros artefatos com os objetos de fluxo. As associações são usadas para mostrar as entradas e as saídas das atividades.	

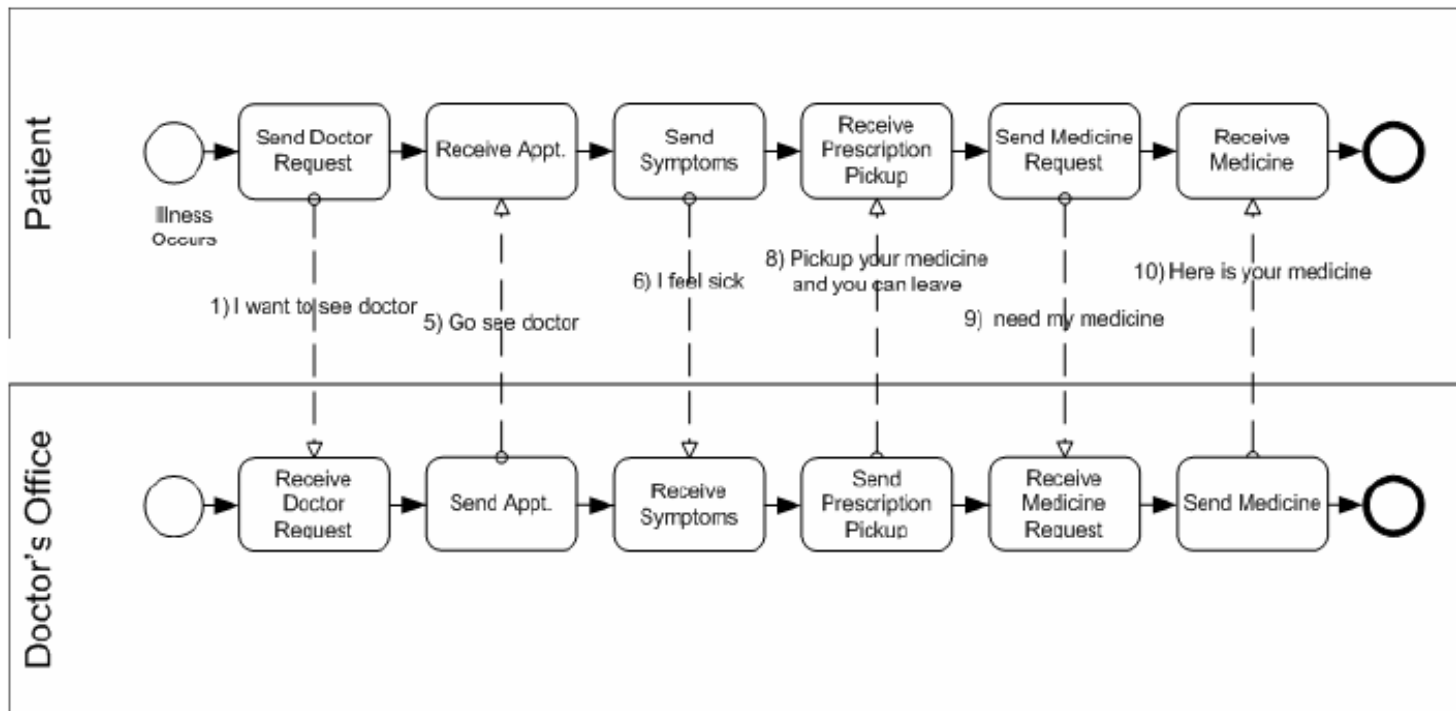
Swimlanes

Funcionam como um mecanismo de organização das atividades em categorias visuais separadas.

Objeto	Descrição	Figura
Pool	Um <i>pool</i> representa um participante em um processo. Ele atua como um container gráfico para dividir um conjunto de atividades de outros <i>pools</i> , geralmente no contexto de situações de B2B.	
Lane	Uma <i>lane</i> é uma subdivisão dentro de um <i>pool</i> usado para organizar e categorizar as atividades.	

Swimlanes - pools

- *Pools* são utilizados quando o diagrama envolve duas entidades de negócio ou participantes que estão separados fisicamente no diagrama.
- Especifica o "que faz o que" colocando os eventos e os processos em áreas protegidas, chamados de

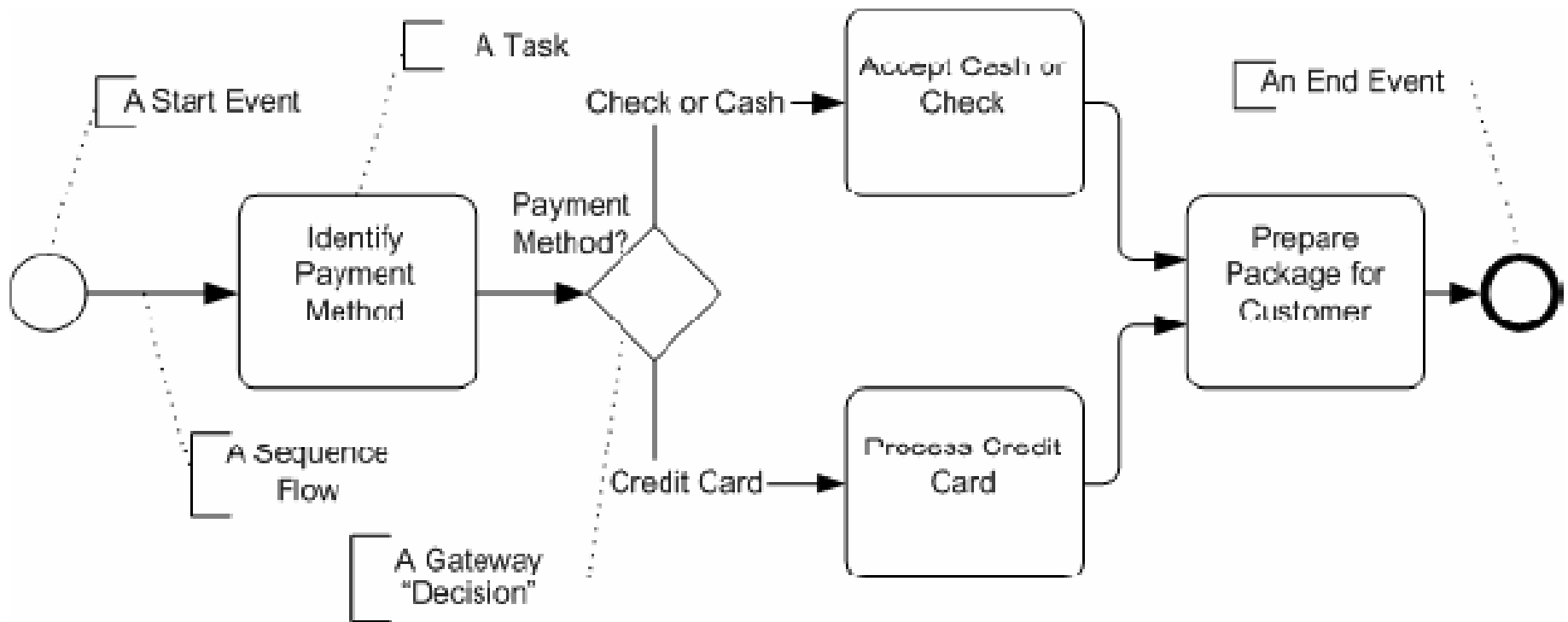


Swimlanes - *lanes*

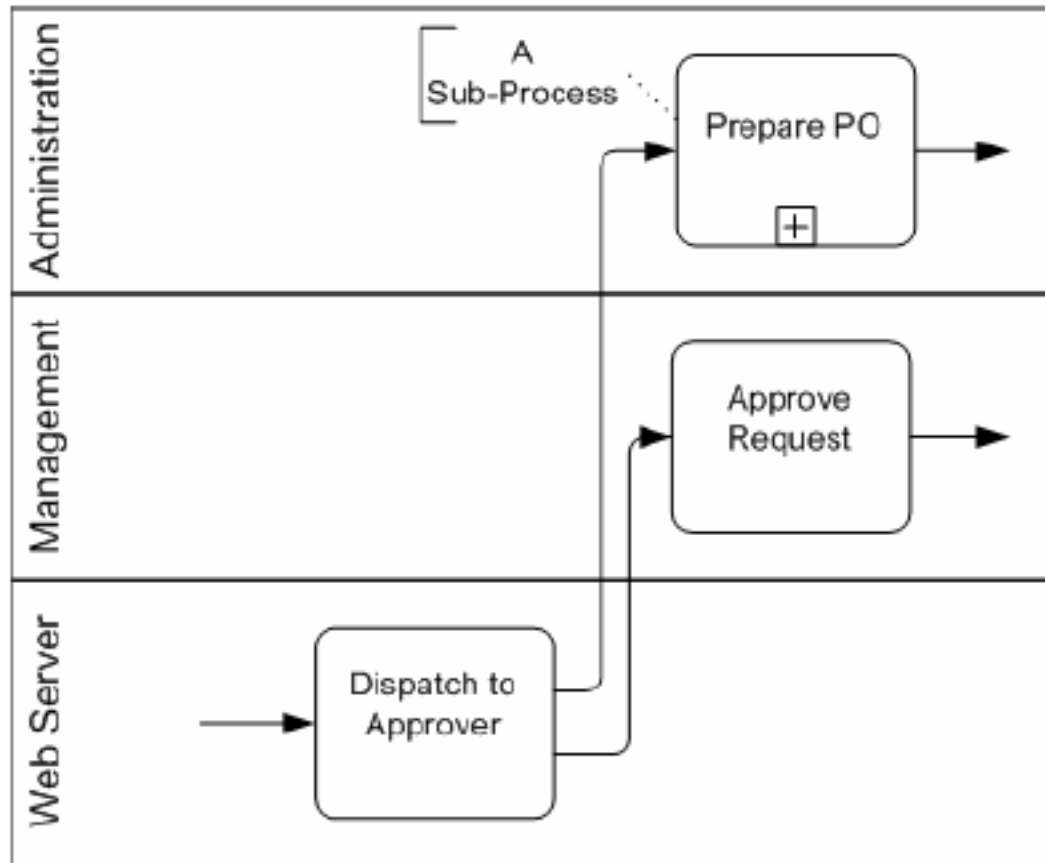
- Os objetos do tipo *lanes* são utilizados para separar as atividades associadas para uma função ou papel específico
- Um pool representa uma organização e uma *lane* representa tipicamente um departamento dentro dessa organização.

Exemplo de um processo

A simbologia da BPMN permite criar modelos de processos para finalidades de documentação e comunicação



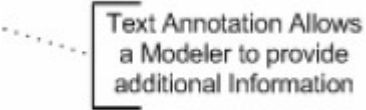


Exemplo de um segmento de processo utilizando *lanes*

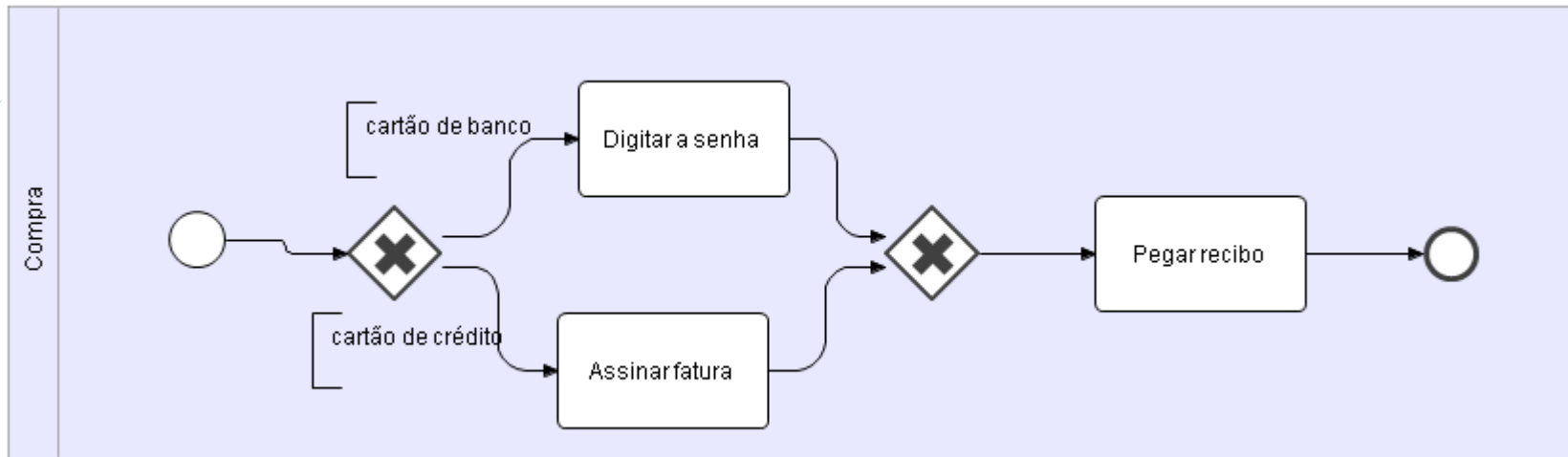


Artefatos (Artefacts)

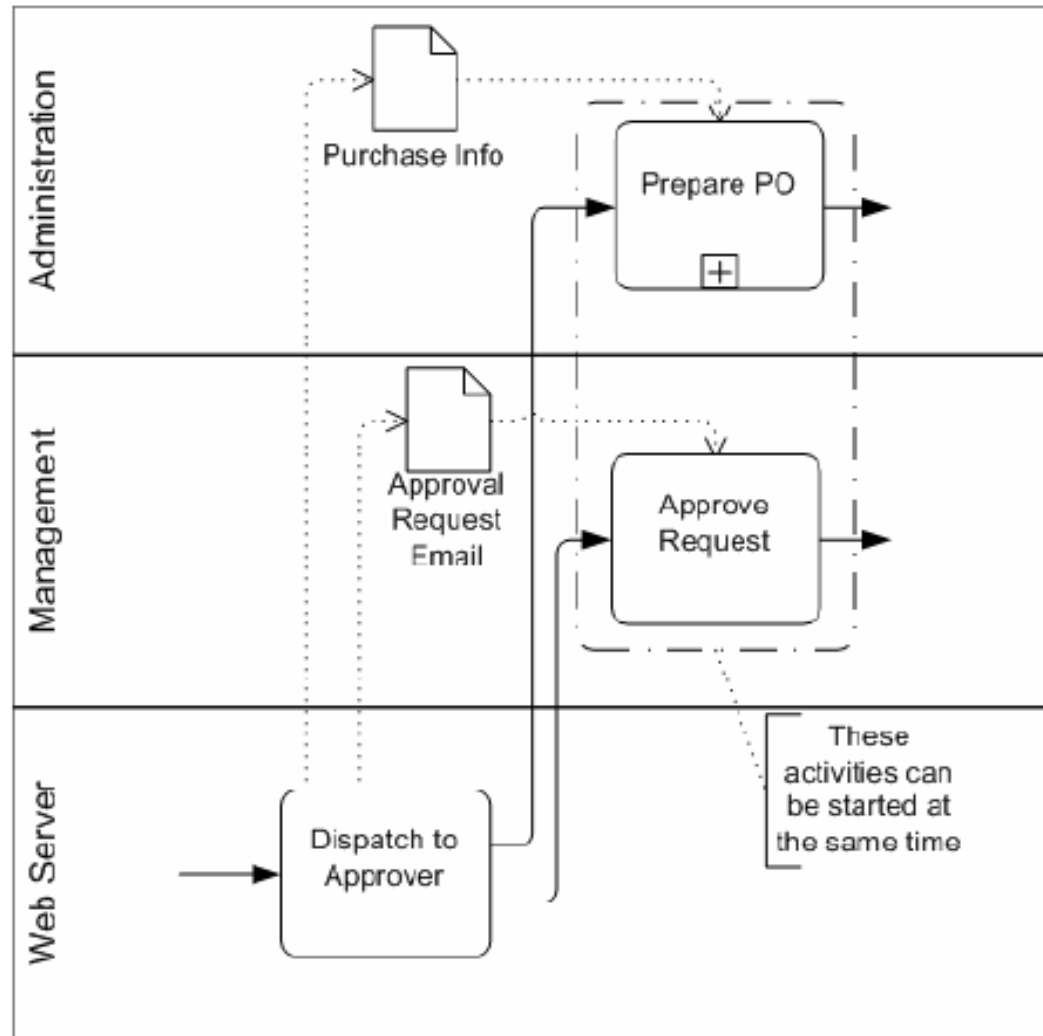
Ilustram as entradas e as saídas das atividades no processo

Objeto	Descrição	Figura
Objetos de dados	O objeto de dado é um mecanismo para mostrar como os dados são requeridos ou produzidos por atividades. São conectados às atividades com as associações.	
Grupo	Um grupo é representado por um retângulo e pode ser usado para finalidades de documentação ou de análise.	
Anotações	As anotações são mecanismos para fornecer informações adicionais para o leitor de um diagrama BPMN.	

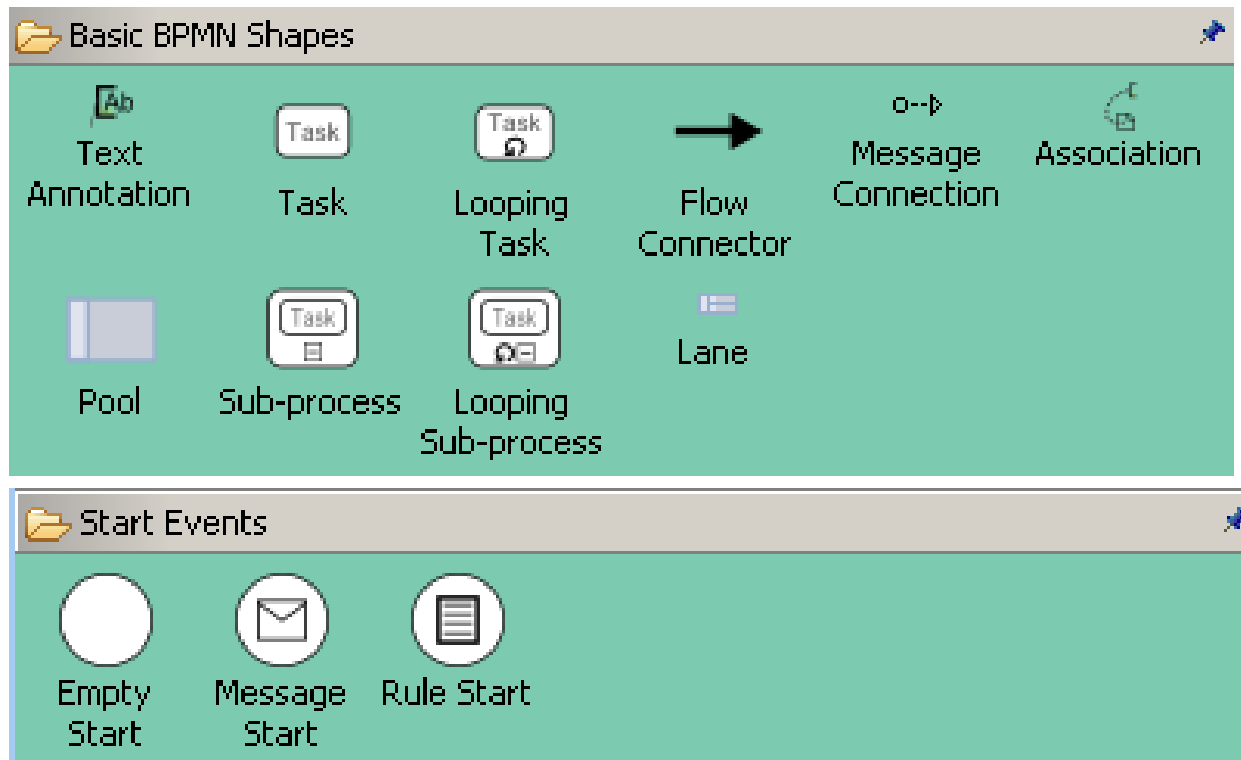
Exemplos Diagrama



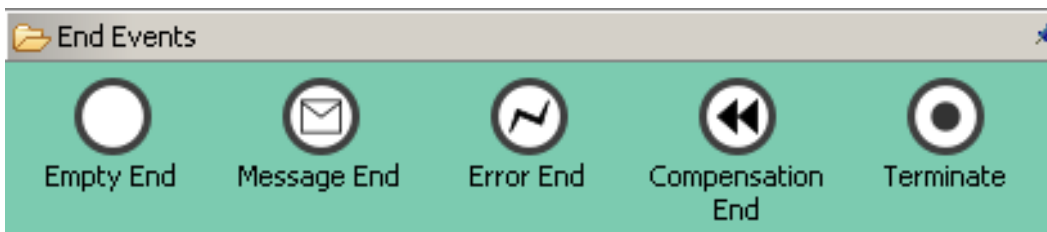
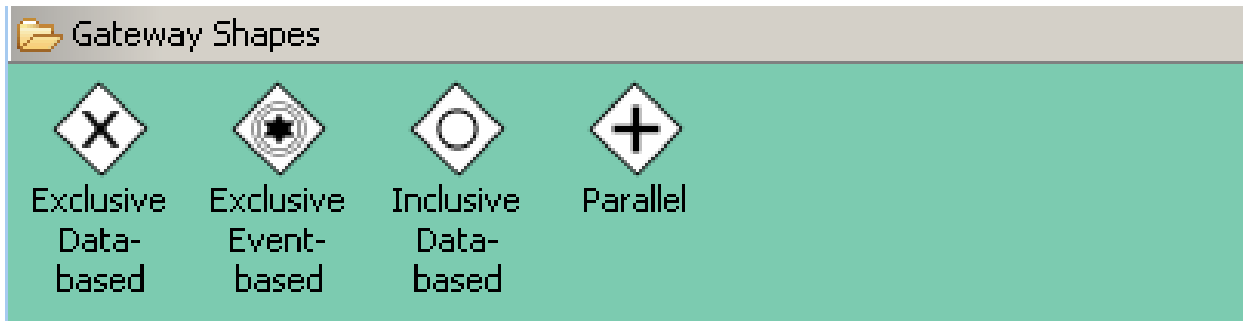
Exemplo de um segmento de processo utilizando artefatos



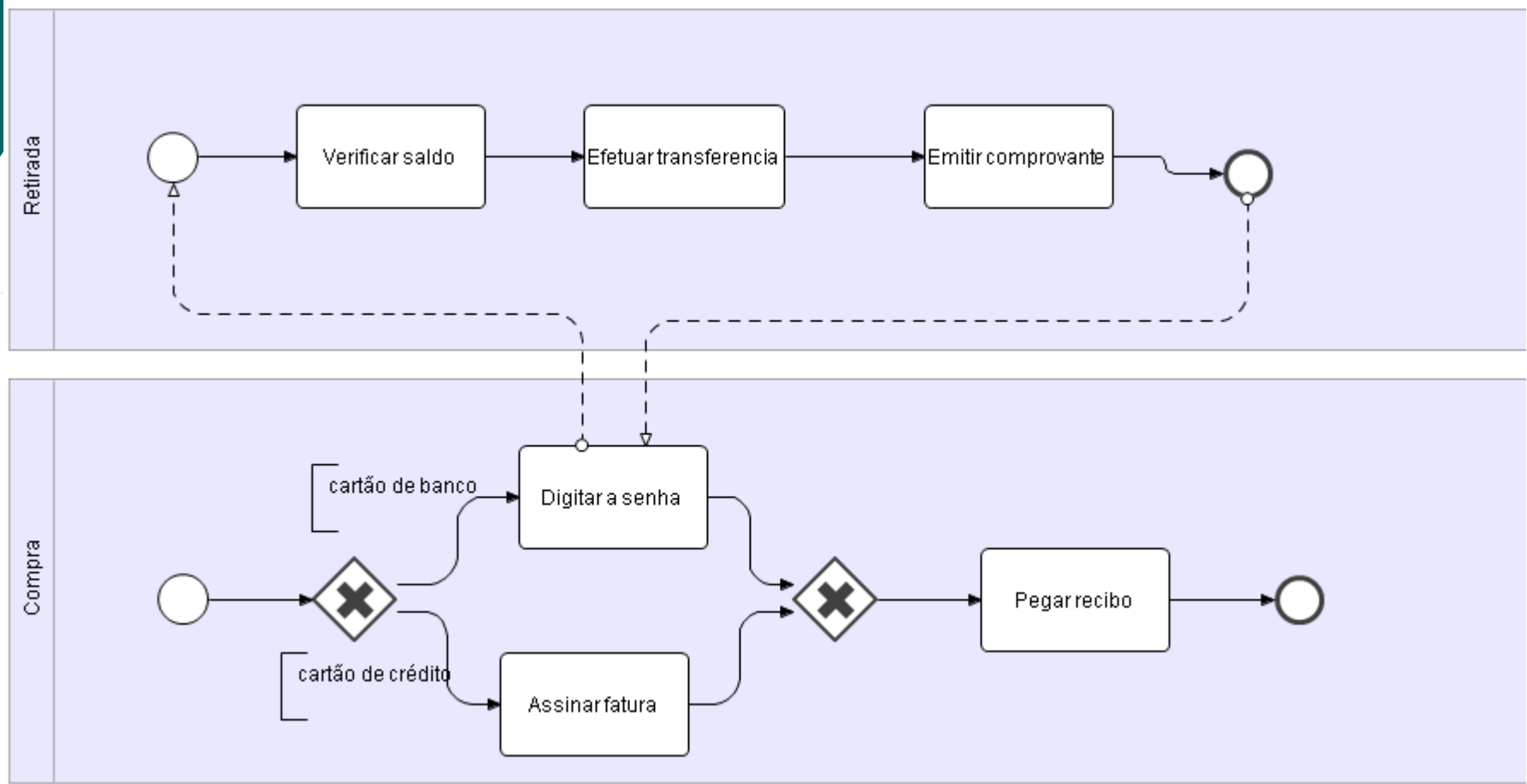
Intalio Designer



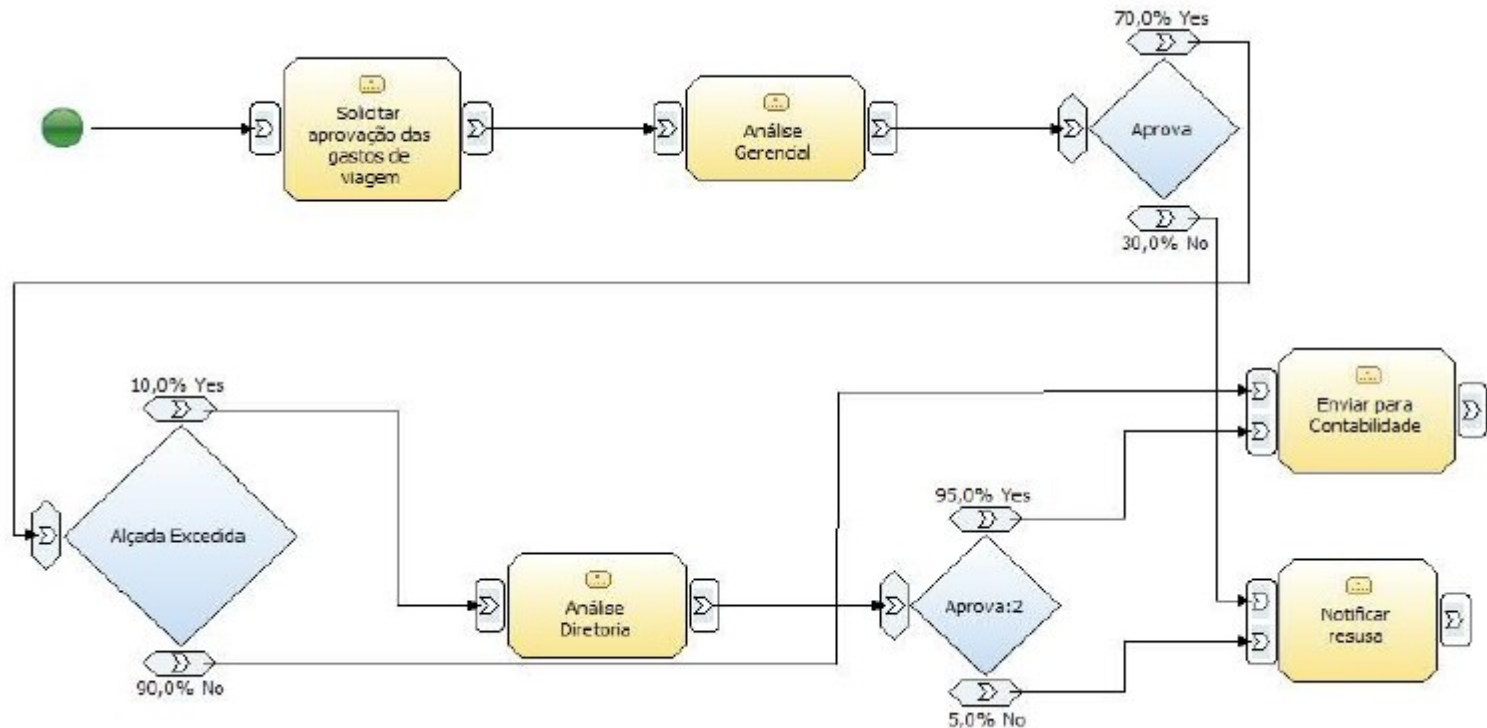
Intalio Designer



Exemplos de diagrama



Aprovação de gastos de viagem



Bibliografia

- BUSSINES PROCESS MANAGEMENT INSTITUTE. <http://bpmi.org>, acesso em agosto/2007.
- GHALIMI, Ismael Chang. **BPM 2.0**. IT-Reduz, Porto Alegre, 2007.
- HURWITZ, J et al. **Service Oriented Architecture for Dummies**, EUA: Willey 2007.
- INTALIO. **Open Source BPMS** <http://www.intalio.com/>, acessado em agosto/2007
- OASIS. **Reference Model for Service Oriented Architecture 1.0** <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html>, acessado em Julho/2007.
- ORACLE, **Getting started with BPEL**. <http://otn.oracle.com/bpel>, acesso em agosto/2007
- SAMPAIO, Cleuton. **SOA e Web Services em Java**. Editora Brasport. Rio de Janeiro, 2006.