

SBS 2011  
SALVADOR - BAHIA - BRASIL  
23 a 25 de maio de 2011

NP2 Tec

MODULOSOFT

## Modelagem de serviços com SoaML

**Thaíssa Diirr, Leonardo Guerreiro Azevedo, Flávia Santoro, Fernanda Baião**  
(thaissa.medeiros, azevedo, flavia.santoro, fernanda.baião)@uniriotec.br

NP2Tec - Núcleo de Pesquisa e Prática em Tecnologia  
Departamento de Informática Aplicada (DIA)  
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)  
Av. Pasteur 458, Urca - Rio de Janeiro / RJ - CEP: 22290-240

Núcleo de Pesquisa e Prática em Tecnologia UNIRIO

---

---

---

---

---

---

---

---

NP2 Tec

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > Geração de código

## Agenda

- SOA (Service-Oriented Architecture)
- MDA (Model-Driven Architecture)
- Modelagem de serviços: UML e profiles
- Profile SoaML
- Ferramentas
- Modelagem SoaML com ferramenta Modelo
- Geração de código a partir de modelo

Núcleo de Pesquisa e Prática em Tecnologia UNIRIO

---

---

---

---

---

---

---

---

NP2 Tec

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > Geração de código

## SOA-Motivação

- Empresas modernas precisam responder de forma efetiva e rápida às oportunidades do mercado
  - Competitivo e globalizado - "lei de Darwin".
- Ao mesmo tempo, processos e sistemas estão cada vez mais complexos.



Núcleo de Pesquisa e Prática em Tecnologia UNIRIO

---

---

---

---

---

---

---

---




---

---

---

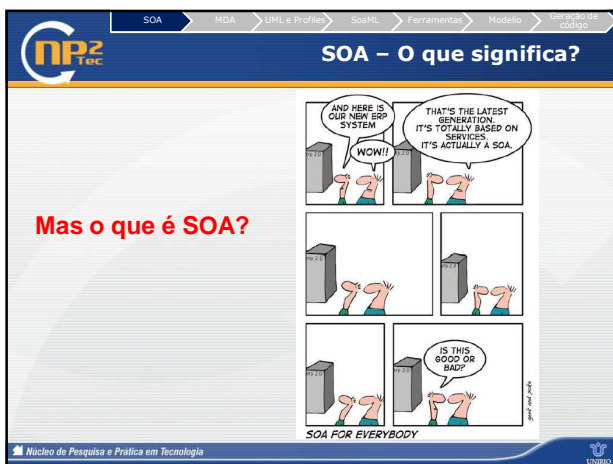
---

---

---

---

---




---

---

---

---

---

---

---

---

**SOA - Definição**

- SOA é um paradigma para a **realização** e **manutenção** de processos de negócio em um grande ambiente de **sistemas distribuídos** que são controlados por **diferentes proprietários**.  
*Nicolai M. Josuttis (2007)*
- SOA é uma **arquitetura conceitual** onde funcionalidade do negócio, ou lógica da aplicação, é disponibilizada para usuários SOA, ou **consumidores**, como **serviços compartilhados** e **reutilizáveis** em uma rede de TI.  
*Marks e Bell (2006)*

Footer: Núcleo de Pesquisa e Prática em Tecnologia | UNIBO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

**NP2 Tec** **Serviços - Definição**

- “Serviços” são pedaços de funcionalidades que possuem interfaces expostas, e que são invocados via mensagens. *(Marks e Bell, 2006)*
- Funcionalidades
  - Ligadas diretamente ao negócio
  - Podem fazer parte de um ou mais processos
- Exemplos
  - Simples: armazenar ou recuperar dados dos funcionários
  - Complexa: processo de negócio para tratar pedido do cliente.

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

**NP2 Tec** **Consumidor + Provedor = Contrato**

- Um serviço tem um tipo de relacionamento (contrato) entre um provedor e um consumidor.
- Provedor se compromete em realizar determinadas tarefas com resultados pré-estabelecidos.
- Consumidor se compromete em usar o serviço da forma contratada.

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

**NP2 Tec** **Perspectiva da TI**

- Serviço
  - Representação lógica de uma atividade do negócio que pode ser mapeada em:
    - Entrada - processamento - saída
  - Deve estar alinhado ao negócio
    - Atender necessidade representada em um processo da organização
  - Operacionalmente independente
    - Fornecer os mesmos resultados para uma mesma entrada
  - Permitir composição
  - Atômico
  - Garantir consistência das informações
  - Eventualmente, persistir resultados

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

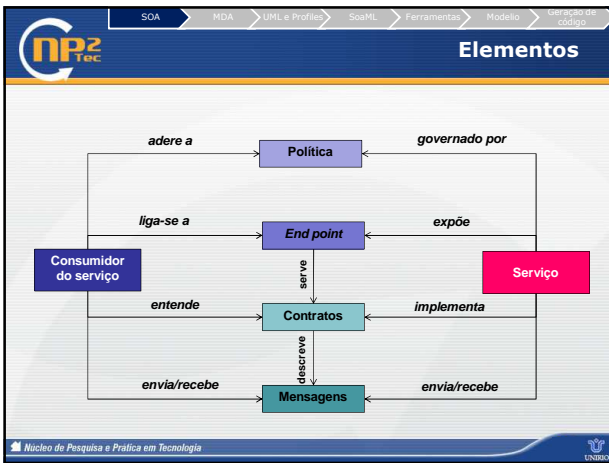
---

---

---

---

---




---

---

---

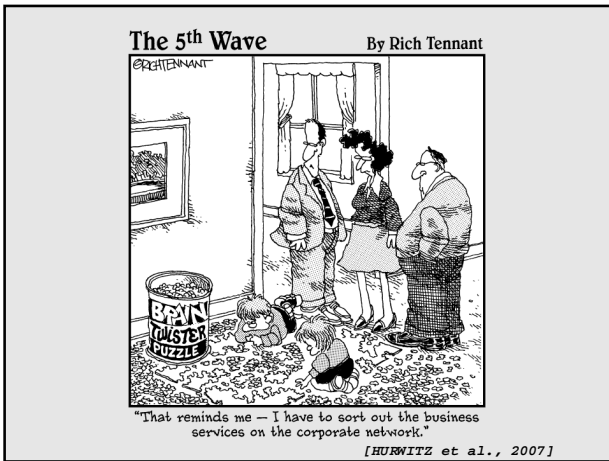
---

---

---

---

---




---

---

---

---

---

---

---

---

**Por que SOA agora?**

- Questões
  - Expansão geográfica dos negócios rapidamente
  - Suportar novos processos rapidamente
  - Utilização simultânea de funcionalidades de sistemas em diferentes plataformas
  - Implantação de novas tecnologias enquanto tecnologias mais antigas estão sendo subutilizadas
  - Alto custo de manutenção de sistemas legados, ao invés de investir em novos projetos estratégicos

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### NP2 Tec

## Por que SOA agora?

- SOA não é um conceito novo, mas agora é realizável:
  - Consenso sobre padrões por grandes empresas (por exemplo, Microsoft e IBM)
- Tecnologia atual permite implementar serviços baseados em padrões
  - Melhor forma de integrar aplicações e negócios.

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### NP2 Tec

## Origens e influências de SOA

Service-Oriented

Object-Oriented   Web Services   BPM   EAI   AOP   Others

Orientação a objetos: Princípios similares como reuso, abstração e composição

Web services: influenciou e promoveu abstração, baixo acoplamento e capacidade de composição de serviços

Business Process Management: essencial em SOA.

Enterprise Application Integration: Conceitos importantes como brokers, serviços sem estado, abstração, baixo acoplamento e capacidade de composição de serviços

Aspect Oriented Programming: Preocupação em construir soluções lógicas transversais, agnósticas a processos de negócio e altamente reutilizáveis

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### NP2 Tec

## Ciclo de Vida Orientado a Stakeholders

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA MDA UML e Profiles SoaML Ferramentas Modelo geração de código

**NP2 Tec** **Ciclo de vida de serviços**

- O ciclo de vida da engenharia de software tradicional não se aplica diretamente a SOA.
  - Novos papéis arquiteturais e novas tarefas de desenvolvimento
- Desafios
  - Como alinhar requisitos de negócio com serviços?
  - Como manipular serviços distribuídos além dos limites organizacionais de maneira segura?
  - Como tratar versões de serviços de acordo com mudanças nos processos de negócio?
- **Ponto de partida na maioria das metodologias propostas: Processos de Negócio**

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

---

---

SOA MDA UML e Profiles SoaML Ferramentas Modelo geração de código

**NP2 Tec** **Ciclo de vida (Gu e Lago)**

- Stakeholders (papéis arquiteturais)
  - Provedor de serviços
    - Desenvolve, publica e mantém serviços para serem consumidos
  - Broker de serviços
    - Provê a informação da localização do serviço (registro de serviços)
    - Registro de serviços: diretório para serviços publicados, como as listas amarelas para números de telefone.
      - Provedores de serviços: publicam seus serviços
      - Consumidores de serviços: localizam serviços.
  - Consumidor de serviços/provedor de aplicações
    - Provê a interface com o usuário para que serviços possam ser acessados
    - Consumidor de serviços: Quando integra os serviços em uma aplicação, que eventualmente atende aos requisitos do usuário final.

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

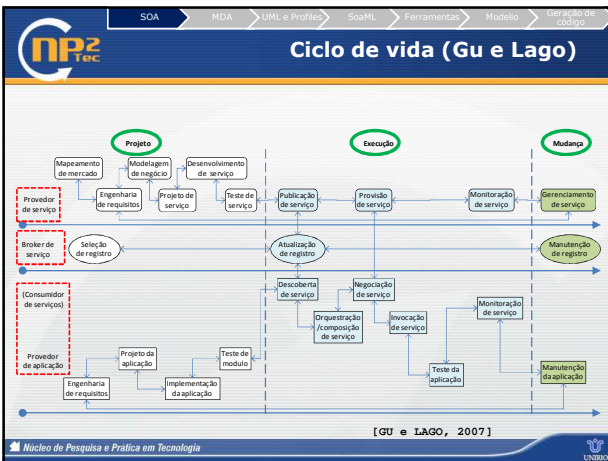
---

---

---

---

---




---

---

---

---

---

---

---

---


---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código



## Web Services

Padrões de 1ª geração  
Padrões de 2ª geração (WS\*)

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---


---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código



## WS: Padrões (1ª geração)

- Web Services
  - Principal tecnologia para desenvolvimento de serviços
  - Padrões de primeira geração
    - Extensible Markup Language (XML)
    - XML Schema Definition Language (XSD)
    - Extensible Stylesheet Language Transformations (XSLT)
    - XML Path Language (XPath)
    - XML Query Language (XQuery)
    - Web Services Description Language (WSDL)
    - Simple Object Access Protocol (SOAP)
    - Universal Description, Discovery, and Integration (UDDI)

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---


---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código



## WS\*: Padrões (2ª geração)

- Algumas das especificações da segunda geração:
 

<ul style="list-style-type: none"> <li>WS-Coordination</li> <li>WS-Transaction</li> <li>WS-BPEL</li> <li>WS-ReliableMessaging</li> <li>WS-Addressing</li> <li>WS-Policy</li> <li>WS-PolicyAssertions</li> <li>WS-PolicyAttachments</li> <li>WS-Attachments</li> <li>SOAP with Attachments (SwA)</li> </ul>	<p><i>Especificações para segurança:</i></p> <ul style="list-style-type: none"> <li>• WS-Security Framework</li> <li>• Extensible Access Control Markup Language (XACML)</li> <li>• Extensible Rights Markup Language (XrML)</li> <li>• Security Assertion Markup Language (SAML)</li> <li>• .NET Passport</li> <li>• XML-Encryption</li> <li>• Secure Sockets Layer (SSL)</li> <li>• XML-Signature</li> <li>• WS-Trust</li> <li>• WS-Privacy</li> <li>• WS-SecurityConversation</li> <li>• WS-Federation</li> <li>• WS-Authorization</li> </ul>
--	--

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

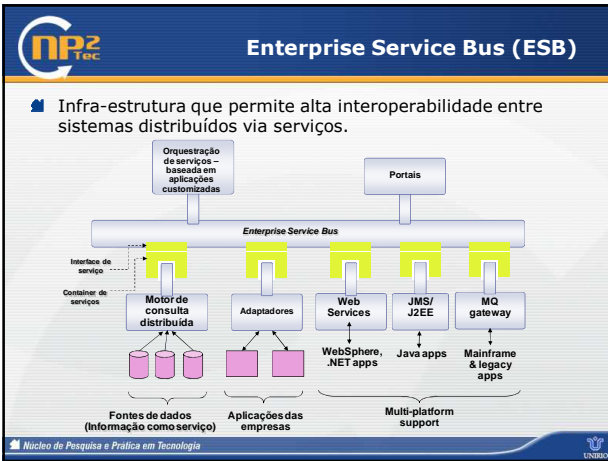
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

- ### Responsabilidades do ESB
- Prover conectividade
  - Baixo acoplamento entre aplicações
  - Lógica de integração é dividida em pedaços de fácil gestão
  - Mapeamento e transformação de dados
  - Roteamento (inteligente)
  - Tratar segurança
  - Tratar confiabilidade
  - Provê facilidades para execução de processos de negócio
  - Gerenciamento de serviços
  - Monitoramento e log de execução
- Núcleo de Pesquisa e Prática em Tecnologia 23 UNIBRIO

---

---

---

---

---

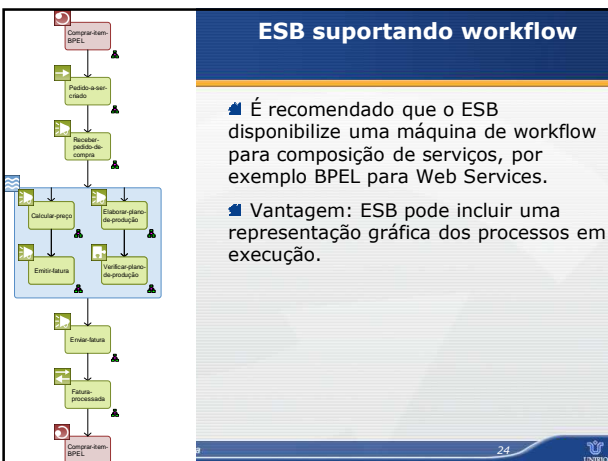
---

---

---

---

---



- É recomendado que o ESB disponibilize uma máquina de workflow para composição de serviços, por exemplo BPEL para Web Services.
- Vantagem: ESB pode incluir uma representação gráfica dos processos em execução.

---

---

---

---

---

---

---

---

---

---



**NP2 Tec** **ESBs disponíveis**

- ESB comerciais
  - IBM WebSphere ESB (<http://www-01.ibm.com/software/integration/wsesb/>)
  - Microsoft BizTalk Server (<http://www.microsoft.com/biztalk/en/us/default.aspx>)
  - Sonic ESB ([http://www.sonicsoftware.com/products/sonic\\_esb\\_family/index.ssp](http://www.sonicsoftware.com/products/sonic_esb_family/index.ssp))
  - TIBCO BusinessWorks (<http://www.tibco.com/software/soa/default.jsp>)
  - Oracle SOA Suite (<http://www.oracle.com/technologies/soa/soa-suite.html>)
- ESBs de código aberto
  - OpenESB da Sun (<http://www.sun.com>)
  - Mule ESB da MuleSource (<http://www.mulesoft.org>)
  - Apache ServiceMix ([www.servicemix.org](http://www.servicemix.org))
  - Fuse ESB (<http://open.iona.com>)

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

---

---

**NP2 Tec** **Regras de negócio**

- Regras de negócio
  - Importantes ativos da organização
  - Devem ser gerenciadas e externalizadas
- Gestão de regras de negócio
  - Guia a definição, categorização, governança e implantação das regras de negócio
  - Geralmente, apoiada por uma tecnologia (BRMS)
- Disponibilização de regras de negócio
  - Baixo acoplamento e interoperabilidade
  - Características da abordagem orientada a serviços (SOA)

Núcleo de Pesquisa e Prática em Tecnologia 26 UNIBO

---

---

---

---

---

---

---

---

---

---

---

---

**NP2 Tec** **Regras de Negócio e SOA**

- Ferramenta BRMS
  - Deve possibilitar que regras, rulesets ou projetos de regras sejam disponibilizados como serviços
- Vantagens
  - Maior facilidade para reuso de regras
  - Invocação de regras utilizando padrões
  - Independência de tecnologia
  - Ferramental para controle da execução das regras

Núcleo de Pesquisa e Prática em Tecnologia 27 UNIBO

---

---

---

---

---

---

---

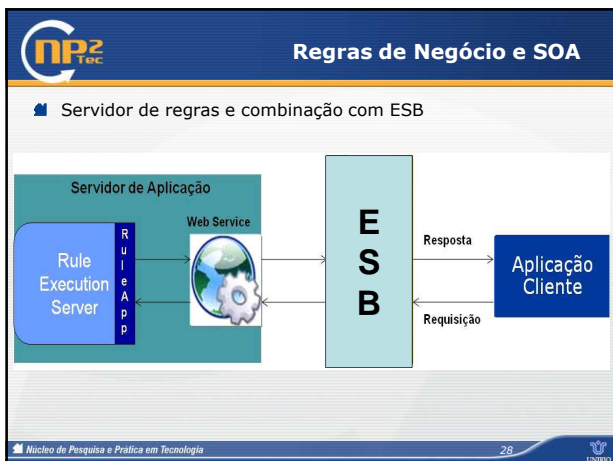
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

**Exemplo: Tipos de dados**

Properties for class Cliente

Name	Type	Direction	Default Value	Verbalize
clienteNome	cliente.Cliente	IN		clienteNome
clienteID	cliente.Cliente	IN_OUT		clienteID

---

---

---

---

---

---

---

---

---

---

**Exemplo: Especificação**

Business Rule: checkClienteNovo

```

code
if the obj of clienteID is null
then
set the estado cadastro of clienteID to "Cliente Novo";
para: "O Cliente precisa ser cadastrado!";
else
set the estado cadastro of clienteID to "Cliente Atualizado";
    
```

Flowchart: Start -> if (clienteID is null) -> set estado cadastro to "Cliente Novo" -> para: "O Cliente precisa ser cadastrado!" -> else -> set estado cadastro to "Cliente Atualizado" -> End.

---

---

---

---

---

---

---

---

---

---

**Ferramentas**

- Corticon BRMS (<http://www.corticon.com>)
- Fico Blaze Advisor (<http://www.fico.com>)
- WebSphere Ilog JRules BRMS (<http://www-01.ibm.com/software/websphere/ilog/>)
- JBoss Enterprise BRMS (<http://www.jboss.com>)
- Sapiens eMerge (<http://www.sapiens.com>)
- Visual Rules (<http://www.visual-rules.com>)
- Versata BRMS (<http://www.versata.com>)
- PegaRules (<http://www.pegacom>)
- InRule (<http://www.inrule.com/products/InRule.aspx>)
- Oracle Business Rules (<http://www.oracle.com>)

Núcleo de Pesquisa e Prática em Tecnologia 31 UNIBO

---

---

---

---

---

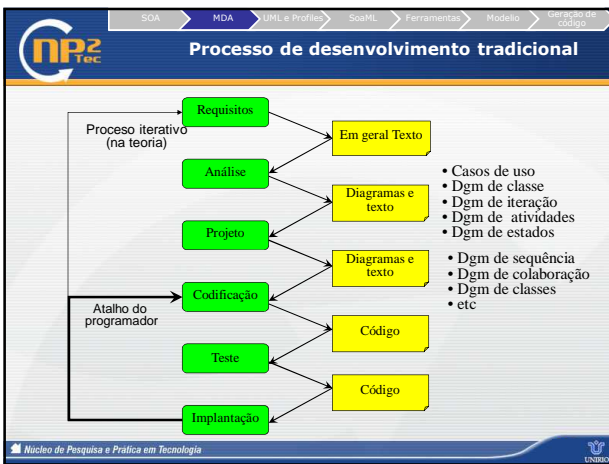
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

**Problemas**

- Os documentos e diagramas criados perdem o valor rapidamente quando se inicia a codificação
  - Perde-se a precisão da especificação
  - Distância aumenta com o passar do tempo
  - Geralmente não há tempo disponível para atualização
- Conhecimento do sistema está na equipe
  - Se mudar a equipe?
- Evoluir para uma nova tecnologia
  - Reescrever todo o código?

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---


---

---

SOA MDA UML e Profiles SoaML Ferramentas Modelo **geração de código**

**NP2 Tec** **Motivação**

- Em engenharia de Software é consenso o ganho com uso de modelos.
- Modelo: Conjunto de elementos que descreve alguma realidade física, abstrata ou hipotética.
  - Descrições de uma solução com diferentes níveis de abstração
    - Ignorar informações que não sejam de interesse no contexto
  - Meios de comunicação
  - Mais baratos de construir e testar



Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA MDA UML e Profiles SoaML Ferramentas Modelo **geração de código**

**NP2 Tec** **MDD – Model Driven Development**

- Desenvolvimento Orientado ao Modelo
  - Todo o desenvolvimento a partir de modelos
  - Programas são gerados automaticamente a partir dos modelos
    - Geração automática de código
    - Verificação dos modelos
  - Modelos são menos sensíveis à tecnologia e às suas evoluções
  - Ampla utilização por várias organizações

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

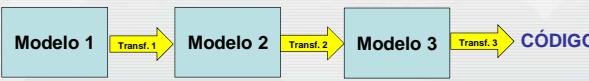
---

---

SOA MDA UML e Profiles SoaML Ferramentas Modelo **geração de código**

**NP2 Tec** **MDD – Model Driven Development**

- Evolução do modelo através de transformações
- Processo de construção de uma aplicação
  - Conjunto de transformações que levam ao sistema final



Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

## Model Driven Architecture (MDA)

- Proposta MDD da OMG (Object Management Group)
  - Lançada em 2001
  - Atualizada em 2003
- Define conjunto de padrões que suportam MDD
  - Unified Modeling Language (UML)
  - Meta-Object Facility (MOF)
  - XML Metadata Interchange (XMI)

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

## Model Driven Architecture (MDA)

- Objetivos
  - Facilitar a construção de sistemas
  - Especificar
    - Conceitos e mecanismos para o desenvolvimento de software considerando cenário atual e futuro
  - Motivação
    - Portabilidade
    - Interoperabilidade
    - Reusabilidade
    - Produtividade
    - Qualidade
    - Facilidade de manutenção da documentação

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

## Processo de desenvolvimento com MDA

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

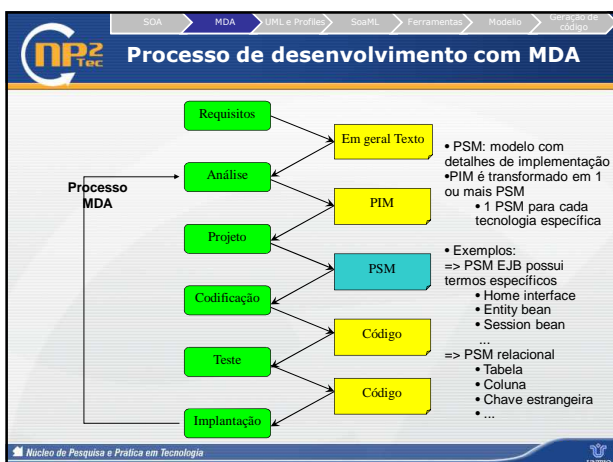
---

---

---

---

---




---

---

---

---

---

---

---

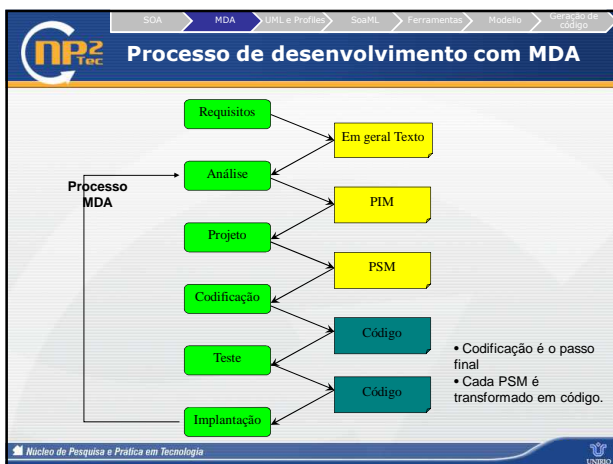
---

---

---

---

---




---

---

---

---

---

---

---

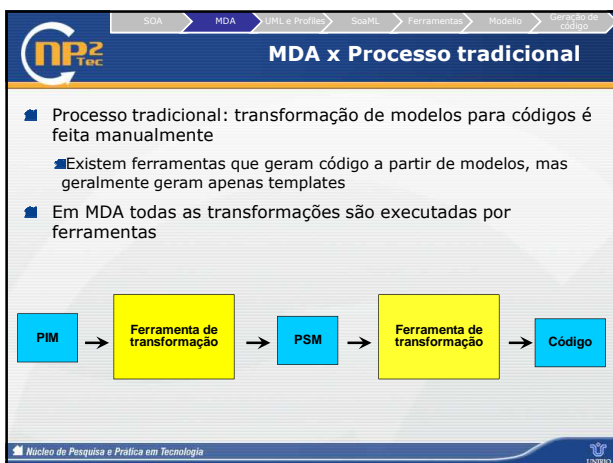
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

**NP2 Tec** **Benefícios do MDA (Produtividade)**

- **Produtividade**
  - O foco do desenvolvedor está no PIM
  - Apesar de ser necessário definir transformações, esta só é feita uma vez
  - Menos código para ser escrito
    - Grande parte do código é gerada
    - Código é gerado de acordo com padrões

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

**NP2 Tec** **Benefícios do MDA (Portabilidade)**

- **Portabilidade**
  - É alcançada com o foco sendo no PIM, que é independente de plataforma.
  - Se precisar redefinir outro ambiente de implementação para a aplicação
    - Definir novas regras de transformação
    - Selecionar os modelos para o novo ambiente
    - Regerar os modelos
  - Custo mais baixo
  - Modelos têm maior longevidade do que o código

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

**NP2 Tec** **Exemplo**

- Transformar um modelo PIM em UML para PSM e o modelo PSM para código Java
- Transformação de atributos
  - PIM:
    - Todos os atributos são públicos
    - Significado:
      - Objeto tem uma propriedade específica
      - Propriedade muda de valor ao longo do tempo.
  - PSM
    - Encapsulamento de atributos
    - Todos os atributos são privados
    - Todo acesso a atributos é controlado por operações bem definidas (get e set)

**Cliente**

+titulo : String  
+nome : String  
+dataDeNascimento : Date

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### Exemplo: PIM para PSM

■ Regras de transformação:

- Para cada classe nomeada *className* no PIM criar classe nomeada *className* no PSM

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### Exemplo: PIM para PSM

■ Regras de transformação:

- Para cada atributo **attributeName: Type** da classe *className* do PIM criar na classe *className* do PSM
  - Atributo privado: **attributeName : Type**

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### Exemplo: PIM para PSM

■ Regras de transformação:

- Para cada atributo **attributeName: Type** da classe *className* do PIM criar na classe *className* do PSM
  - Atributo privado: **attributeName : Type**
  - Operação pública "get": **getAttributeName() : Type**
  - Operação pública "set": **setAttributeName( att : Type)** sem valor de retorno

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---



SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### Exemplo: PSM para código

```

classDiagram
    class Cliente {
        -titulo : String
        -nome : String
        -dataDeNascimento : Date
        +getTitulo() : String
        +setTitulo(titulo : String)
        +getNome() : String
        +setNome(nome : String)
        +getDataDeNascimento() : Date
        +setDataDeNascimento(data : Date)
    }
    
```

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### Exemplo: PSM para código

```

package cliente;

import java.util.Date;

public class Cliente {
    private String titulo;
    private String nome;
    private Date dataDeNascimento;

    public String getTitulo() {
        return titulo;
    }
    public void setTitulo(String titulo) {
        this.titulo = titulo;
    }

    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }

    public Date getDataDeNascimento() {
        return dataDeNascimento;
    }
    public void setDataDeNascimento(Date dataDeNascimento) {
        this.dataDeNascimento = dataDeNascimento;
    }
}
    
```

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### Marcações

- Funções de transformação sozinhas nem sempre são suficientes para transformar completamente um modelo de origem.
  - Necessidade de entradas adicionais
- Especificar marcas em um modelo de marcação.
- Modelo de marcação define:
  - Nomes, tipos e valores default

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### Marcações

- Exemplo 1: Classes no modelo de origem podem ter operações que acessam dados locais ou dados remotamente
  - Gerar todas as operações como acesso remoto (default) pode produzir implementação muito lenta.
  - Criar marca AccessorType com os possíveis valores:
    - isRemote
    - isLocal

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### Marcações

- Exemplo 2: Função de transformação para produzir um esquema de banco de dados relacional a partir de um modelo UML
  - Marca pode ser usada para:
    - Especificar nome da tabela de destino
    - Nome da coluna de destino
  - Função de transformação copiará o valor da marca para a definição do esquema do banco de dados.

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### Mecanismos de extensão da UML

- Quando precisamos de uma sintaxe/semântica específica e contextualizada?
- Pode haver casos em que os construtores da linguagem sejam genéricos demais para representar particularidades de um domínio ou plataforma.
- Solução: mecanismos de extensão da UML.

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

## Perfil UML (profiles)

- Mecanismo de especialização definido como parte da UML
- Mecanismo para adaptar um metamodelo existente a construções que são específicas a um modelo particular, plataforma ou metodologia.
- Não altera o metamodelo
  - Preserva a UML
  - Especializa alguns de seus conceitos
- Suporte ferramental existente.
- Perfil: combinação de estereótipos e OCL (Object Constraint Language)

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

## Estereótipo

- Estende o vocabulário básico da UML.
  - Definir estereótipo é semelhante a criar uma subclasse de um tipo UML existente.
- Podem ser anexados a um elemento de modelo para diferenciá-los.
- Exemplo: Descrever elementos como persistentes
  - Estereótipo <<persistent>> atribuído a elementos do tipo Classe.

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

## OCL

- OCL (Object Constraint Language)
  - Linguagem de expressão na qual é possível escrever expressões sobre modelos
    - Exemplos:
      - Regras de derivação
      - Corpo de operações
      - Pré-condições
      - Pós-condições
      - Invariantes
      - Especificações de inicializações
  - Atende a necessidades não cobertas pela UML
  - O modelo UML de um sistema torna-se mais preciso e mais completo aplicado-se OCL.

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

**NP2 Tec** **OCL – Exemplo 1**

- Exemplo 1) Uma transformação mapeia um ou mais elementos de um modelo origem a um ou mais elementos de um modelo destino
  - Consultas OCL pode ser usada para
    - Especificar elementos do modelo de origem
    - Especificar elementos do modelo de destino

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

**NP2 Tec** **OCL – Exemplo 2**

- Exemplo 2) Transformações só podem ser aplicadas em determinadas condições.
  - Especificar
    - Condição OCL sobre os elementos origem
    - Condição OCL sobre os elementos de destino
  - Todas as expressões OCL usadas na definição da transformação são especificadas no metamodelo das linguagens de origem e de destino.

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

**NP2 Tec** **Modelagem de serviços com UML**

- Existem propostas para modelagem SOA com o uso de UML padrão
  - Rahmani *et al* [2006] sugerem o uso de diagramas de classes
    - São especificados objetos de valor do domínio (dados)
    - Interfaces que manipulam estes objetos
    - Interface do serviço
      - Pode ser composta de uma ou mais interfaces manipuladoras de objetos do domínio
  - Sanders *et al* [2005] sugerem o uso de diagramas de colaboração
    - Serviços são definidos como uma colaboração
    - Estrutura da colaboração identifica papéis que colaboram para proporcionar a funcionalidade para o consumidor

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > Geração de código

## Modelagem de serviços com UML

- Porém, a UML padrão não suporta a modelagem de características específicas de SOA
  - Necessário o uso de um *profile*
    - Permitir a representação de características de serviços que a UML padrão não suporta, como:
      - Provedores e consumidores de serviço
      - Mensagens trocadas entre eles
      - Interfaces dos serviços
      - Orquestração e coreografia de serviços
      - Contratos e políticas entre provedores e consumidores
      - Arquiteturas de serviços
    - Gerar modelos em um nível de abstração que permita gerar artefatos da implementação dos serviços a partir de modelos

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > Geração de código

## Modelagem de serviços com UML

- Propostas de *profile* para SOA na literatura
  - Kenzi et al [2009]; Lopez-Sanza et al. [2008]; Emig et al [2008]; Mayer et al. [2008a, 2008b]; Ermagen e Krüger [2007]; Johnston [2005]; Amir e Zeid [2004]
  - SoaML [OMG, 2008]
    - Service Oriented Architecture Modeling Language
    - Proposto pela OMG (Object Management Group) em 2008 para se tornar o padrão de modelagem de serviços em uma SOA
    - Descreve um *profile* e metamodelo que estendem a UML 2.0
    - Suportado por ferramentas disponíveis no mercado que permitem:
      - Produção de modelos com este *profile*
      - Transformação dos modelos em artefatos de plataformas específicas como WSDL, XSD, BPEL e Java

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > Geração de código

## O profile SoaML

- SoaML suporta
  - Definição de serviços candidatos
  - Especificação de serviços
    - Capacidades funcionais providas
    - Capacidades que os consumidores devem possuir
    - Fluxo de interação para o serviço
    - Contratos
  - Especificação de consumidores e provedores
    - Serviços consumidos
    - Serviços providos
    - Como eles estão conectados

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

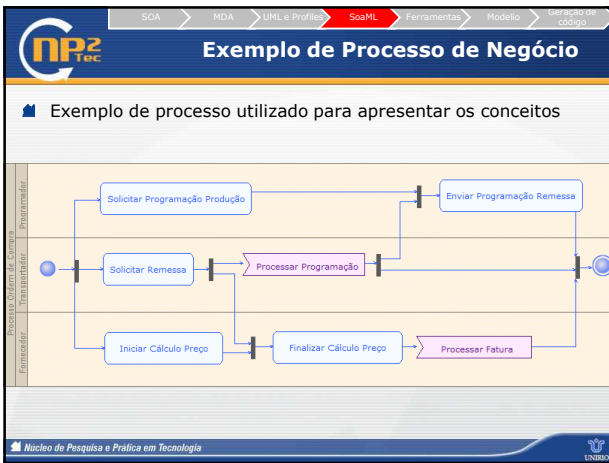
---

---

---

---

---




---

---

---

---

---

---

---

---

**Conceitos introduzidos à UML**

**Capacidade (<<Capability>>)**

- Habilidade de agir e produzir resultados que atingem objetivos
- Identificam um conjunto de funções ou recursos que um serviço pode oferecer
- Podem ser vistas de duas perspectivas
  - Capacidades que um participante possui e pode ser exposta como serviço
    - Exemplo: Calcular preço do pedido.
  - Capacidades que uma organização necessita e podem ser utilizadas para identificar serviços candidatos
    - Exemplo: Processar ordem de compra.

---

---

---

---

---

---

---

---

**Conceitos introduzidos à UML**

**Capacidade (<<Capability>>)**

- Uma capacidade pode depender de outras para ser fornecida
- Capacidades ajudam a:
  - Identificar serviços necessários
  - Organizar serviços segundo necessidades de uma área
  - Analisar como estão relacionados
  - Analisar se podem ser combinados em capacidades maiores ou divididos em capacidades menores
- Na especificação de capacidades
  - Não há preocupação em como os serviços devem ser implementados e oferecidos aos consumidores

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > Geração de código

### Conceitos introduzidos à UML

#### Capacidade (<<Capability>>)

Processos são ótimas fontes para identificação de capacidades, mas outras fontes podem ser utilizadas.

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > Geração de código

### Conceitos introduzidos à UML

#### Dados e mensagens

- Dois estilos para definição de parâmetros de operações de serviços
  - Centrado a documento (ou mensagem) (*document*)
    - Tipos de mensagens como parâmetros
    - Cada operação só pode ter um parâmetro de entrada, um de saída e um de exceção
    - +processarOrdemCompra (ordemCompra: OrdemCompraMensagem)
  - RPC (*Remote Procedure Call*)
    - Parâmetros restritos a tipos primitivos ou tipos de dados
    - Não existe um número máximo para quantidade de parâmetros
    - +iniciarCalculoPreco (cliente: Cliente, ordemCompra: OrdemCompra)

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > Geração de código

### Conceitos introduzidos à UML

#### Dados e mensagens

- O modelo de dados do domínio é especificado como um diagramas de classe da UML padrão
- Podem ser utilizados neste diagrama:
  - Tipos de dados («DataType») definidos pelo usuário
    - Com atributos que são tipos simples da UML padrão
    - Ou esses novos tipos de dados definidos

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

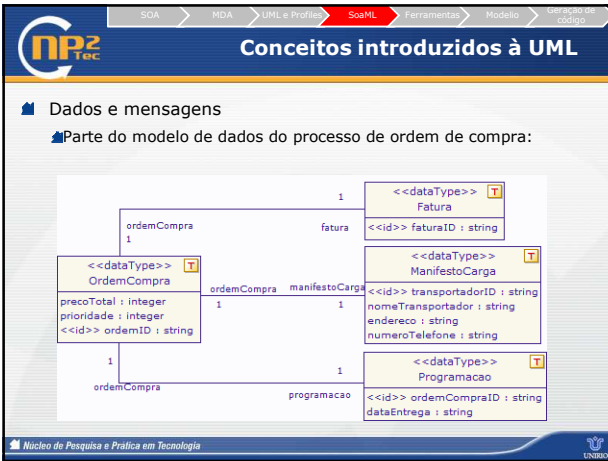
---

---

---

---

---



- Dados e mensagens
- Parte do modelo de dados do processo de ordem de compra:

---

---

---

---

---

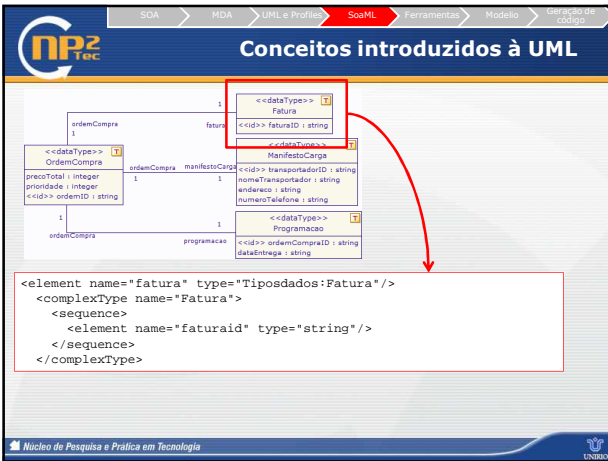
---

---

---

---

---




---

---

---

---

---

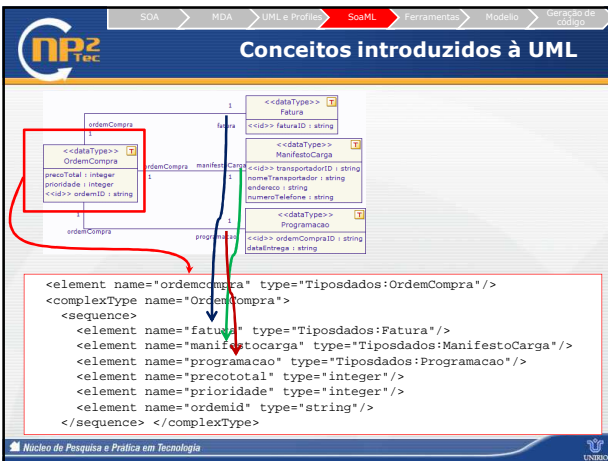
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---



SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

**NP2 Tec** **Conceitos introduzidos à UML**

- Dados e mensagens
  - Tipos de mensagens («*MessageType*») também podem ser definidos
  - Tipos de mensagens
    - Tipos de dados que representam a informação trocada entre requisições de consumidores e os serviços de provedores
    - Dados passados para uma operação de serviço e/ou retornados a partir de sua invocação
      - Entradas, saídas e exceções das operações de serviços

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

**NP2 Tec** **Conceitos introduzidos à UML**

- Dados e mensagens
  - Tipos de mensagens podem possuir anexos («*Attachment*») além de atributos
    - Não são parte direta da mensagem em si
    - Usados para indicar parte dos dados do serviço que podem ser acessados separadamente
      - Reduz os dados enviados entre consumidores e fornecedores, exceto quando necessário

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

**NP2 Tec** **Conceitos introduzidos à UML**

- Dados e mensagens
  - Mensagens do processo de ordem de compra:
    - ```

                    classDiagram
                        class FaturaMensagem {
                            <<Attachment>> conteudo : string
                            <<id>> faturaID : string
                        }
                        class OrdemCompraMensagem {
                            <<Attachment>> modelo : string
                        }
                        class Cliente {
                            clienteID : string
                            nome : Nome
                            endereco : string
                            pontuacaoCredito : integer
                        }
                        class OrdemCompra {
                            precoTotal : integer
                            prioridade : integer
                            ordemID : string
                        }
                        Cliente "1" -- "1..*" OrdemCompraMensagem : cliente
                        OrdemCompraMensagem "1" -- "1" OrdemCompra : pedidoCompra
                    
```

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

## Conceitos introduzidos à UML

**Dados e mensagens**

- Parte do XSD das mensagens

```

<element name="faturamensagem" type="Mensagens:FaturaMensagem"/>
<complexType name="FaturaMensagem">
  <sequence>
    <element name="conteudo" type="string"/>
    <element name="faturaaid" type="string"/>
  </sequence>
</complexType>
    
```

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

## Conceitos introduzidos à UML

- Consumidores e provedores**
  - Consumidor («Consumer»)
    - Solicita um serviço do provedor
  - Provedor («Provider»)
    - Usa suas capacidades para atender à solicitação para o serviço
  - Interação entre o provedor e o consumidor
    - Governada por um contrato de serviço
    - Em geral, consumidor inicia a interação de serviço
    - Em geral, provedor responde a interação de serviço
  - Estereótipos são aplicados às interfaces dos serviços

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

## Conceitos introduzidos à UML

- Interfaces de um serviço**
  - Definem operações disponibilizadas pelos provedores
  - Assinatura de um serviço
    - Descreve
      - Operações
      - Parâmetros de entrada
      - Parâmetros de saída
      - Exceções
  - Em SoaML, as interfaces podem ser:
    - Interfaces simples da UML padrão
    - Interfaces do tipo ServiceInterface adicionadas pelo profile

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

**NP2 Tec** **Conceitos introduzidos à UML**

**■ Interfaces simples**

- Consumidor apenas invoca as operações definidas na interface
  - É unidirecional
  - O provedor não invoca o consumidor em resposta à execução do serviço
- Não é requerido uma interação para a comunicação entre as partes consumidora e provedora
- Pode ser usada por diferentes consumidores
  - Provedor não faz nenhuma suposição sobre eles

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

**NP2 Tec** **Conceitos introduzidos à UML**

**■ Interfaces simples**

```

<<interface, Provider>>
Programacao
solicitarProgramacaoRemessa(in cliente : Cliente, in ordemCompra : OrdemCompra)
enviarProgramacaoProducao(in programacao : Programacao)
    
```

- Atividades do diagrama de atividades do processo que geraram a capacidade são ações do tipo action
- Possui o estereótipo de provedor («Provider»)
- Interfaces em SoaML são definidas do ponto de vista dos provedores

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

**NP2 Tec** **Conceitos introduzidos à UML**

**■ ServiceInterfaces**

- Definem: operações providas e necessidades dos consumidores para invocar estas operações
- Especificam um serviço bi-direcional
  - Provedor e consumidor: enviam e recebem mensagens
  - Existem callbacks do provedor ao consumidor como parte da comunicação

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

### NP2 Tec

## Conceitos introduzidos à UML

#### ServiceInterfaces

- Definidas da perspectiva do provedor do serviço
  - Especificando
    - Interface que o provedor oferece
    - Interface que ele espera do consumidor
  - As interfaces providas e requeridas por uma ServiceInterface são interfaces simples da UML padrão
    - São, respectivamente, realizadas ou utilizadas por ela
  - As interfaces realizadas especificam o que é provido pelo provedor
  - As interfaces utilizadas definem o que é requerido do consumidor

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

### NP2 Tec

## Conceitos introduzidos à UML

#### ServiceInterfaces

- Relacionamento de exposição («Expose»)
  - Pode ser definido entre ServiceInterface e uma capacidade
  - Interface provê as operações de forma consistente com a capacidade exposta

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

### NP2 Tec

## Conceitos introduzidos à UML

#### ServiceInterfaces

- Interface conjugada
  - Iniciadas por "~" (til)
  - Representam a perspectiva do consumidor do serviço

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---



---

---

---

---

---

---

---

---

---

---

**ServiceInterfaces**

- No diagrama de atividades do processo, as operações que os serviços precisam invocar dos consumidores são do tipo:
  - `Accept event action` (para chamadas assíncronas)
  - `Accept call action` (para chamadas síncronas)
- No processo de ordem de compra, são as atividades "ProcessarFatura" e "ProcessarProgramacao"

---

---

---

---

---

---

---

---

---

---

**Contratos de serviço («ServiceContract»)**

- Especifica como consumidores e provedores trabalham juntos para atingir algum valor
- Acordo que define como um serviço é provido e consumido
- São especificados:
  - Papéis que cada participante assume no serviço
  - Interfaces que eles implementam para assumir estes papéis
    - Papel
      - Define a função básica (ou conjunto de funções) que uma entidade executa em um contexto

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > Geração de código

**NP2 Tec** **Conceitos introduzidos à UML**

- Contratos de serviço («ServiceContract»)
  - Especificação de um contrato
    - Não se preocupa com sua realização
    - Descrevem como as partes concordam para prover ou usar o serviço
    - Não considera como estas partes implementam seu papel no serviço através de seus processos internos
  - Serviço e contratos
    - Serviço pode ser construído para atender a um contrato já estabelecido
    - Ou, contrato pode ser definido após a construção do serviço correspondente

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---



---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > Geração de código

**NP2 Tec** **Conceitos introduzidos à UML**

- Contratos de serviço («ServiceContract»)
  - Sua base é uma colaboração da UML
    - Com interações necessárias à provisão do serviço
  - Cada papel envolvido possui um nome e um tipo
    - Tipos são interfaces provedoras ou consumidoras
      - Indicando papéis provedores  e consumidores 
    - Para assumir papel de um contrato
      - Participante deve fornecer a interface definida como tipo deste papel

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > Geração de código

**NP2 Tec** **Conceitos introduzidos à UML**

- Contratos de serviço («ServiceContract»)
  - Na maioria dos casos, um contrato de serviços especifica dois papéis
    - Um consumidor e um provedor
  - Existem casos onde mais papéis podem ser especificados
    - Contratos multi-parte

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### NP2 Tec

## Conceitos introduzidos à UML

- Contratos de serviço para interface simples
  - Não requerem fluxo de interação entre consumidor e provedor em resposta à execução do serviço
    - Consumidores podem ser desconhecidos
  - No contrato, o papel consumidor não possui tipo definido
  - Papel provedor possui o tipo definido normalmente e indicação de provedor

UML diagram: <<ServiceContract>> ProgramacaoContrato. Roles: cliente:, programador:Programacao (with green circle icon).

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### NP2 Tec

## Conceitos introduzidos à UML

- Contratos de serviço para interface *ServiceInterface*
  - Requerem fluxo de informação entre consumidor e provedor
    - Consumidores devem ser conhecidos, pois existem *callbacks*
  - Papéis do provedor e do consumidor devem ter o tipo definido
  - Provedor ou consumidor devem ser indicados

UML diagram: <<ServiceContract>> FaturaContrato. Roles: fornecedor:Fatura (with green circle icon), solicitante:ProcessamentoFatura (with blue arrow icon).

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### NP2 Tec

## Conceitos introduzidos à UML

- Contratos de serviço multi-parte
  - Menos comum, ocorre entre mais de dois participantes
  - No contexto do processo de ordem de compra não há este tipo de contrato

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

### Conceitos introduzidos à UML

- Contratos de serviço multi-parte
  - Exemplo: Serviço de compras onde há um intermediário
    - Há interação do comprador com o vendedor e com o intermediário

Núcleo de Pesquisa e Prática em Tecnologia

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

### Conceitos introduzidos à UML

- Fluxos de interação de contratos
  - Protocolo da comunicação entre provedor e consumidor
  - Representa as interações válidas entre eles
  - Pode ser modelado como
    - Diagrama de interação ou um diagrama de atividade
  - Define como participantes que assumem os papéis do contrato (sem especificar quem são) devem proceder
  - Não são definidos para serviços cuja interface é simples
    - Não há tipo de interação entre provedor e consumidor

Núcleo de Pesquisa e Prática em Tecnologia

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > **SoaML** > Ferramentas > Modelo > geração de código

### Conceitos introduzidos à UML

- Fluxos de interação de contratos

Núcleo de Pesquisa e Prática em Tecnologia

---

---

---

---

---

---

---

---



SOA MDA UML e Profiles **SoaML** Ferramentas Modelo geração de código

## Conceitos introduzidos à UML

- Participantes («Participant»)
  - Pessoas, organizações, unidades organizacionais ou sistemas de informação que provêem e/ou utilizam serviços
  - Um participante
    - É um provedor se ele provê um serviço
    - É um consumidor se ele utiliza um serviço
  - Não há limite para quantidade de serviços que um participante pode prover e/ou consumir
  - Um participante não pode realizar ou utilizar interfaces diretamente
    - São necessárias portas de serviço ou de requisição

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

---

---

SOA MDA UML e Profiles **SoaML** Ferramentas Modelo geração de código

## Conceitos introduzidos à UML

- Portas
  - Ponto de conexão em um participante
  - Onde um serviço é provido ou consumido
    - Solicitações do consumidor são enviadas nas portas dos provedores
  - Podem ser
    - Portas de serviço
    - ou
    - Portas de requisição

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

---

---

SOA MDA UML e Profiles **SoaML** Ferramentas Modelo geração de código

## Conceitos introduzidos à UML

- Portas de serviço («Service»)
  - Representam serviços providos por participantes
  - Para um serviço ser provido através de uma porta, seu contrato deve ser cumprido
    - Interfaces providas, interfaces requeridas e protocolo de comunicação
  - Seu tipo pode ser interface simples ou *ServiceInterface*
  - Exemplo (*ServiceInterface*)

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### Conceitos introduzidos à UML

- Portas de serviço («Service»)
  - Representam serviços providos por participantes
  - Para um serviço ser provido através de uma porta, seu contrato deve ser cumprido
    - Interfaces providas, interfaces requeridas e protocolo de comunicação
  - Seu tipo pode ser interface simples ou *ServiceInterface*
  - Exemplo (Interface simples)

Provisão de serviço com interface simples

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### Conceitos introduzidos à UML

- Portas de requisição («Request»)
  - Representam o consumo de um serviço
  - Para um serviço ser consumido através de uma porta, seu contrato deve ser cumprido
    - Interfaces providas, interfaces requeridas e protocolo de comunicação
  - Seu tipo pode ser interface simples ou *ServiceInterface*
  - Operações providas através da porta de serviço são as definidas na interface conjugada correspondente

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

### Conceitos introduzidos à UML

- Portas de requisição («Request»)
  - Porta de requisição como *ServiceInterface*
    - Interface provida através da porta é a realizada pela interface conjugada
    - Interface requerida é aquela usada pela interface conjugada

Consumo de serviço com *ServiceInterface*

Consumo de serviço com *ServiceInterface*

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

---

---

NP2 Tec SQA MDA UML e Profiles SoaML Ferramentas Modelo geração de código

### Conceitos introduzidos à UML

- Portas de requisição («Request»)
  - Porta como interface simples
    - Interface requerida será esta própria interface
    - Não haverá interfaces providas
    - Não há protocolos de comunicação a serem cumpridos

Consumo de serviço com interface simples

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

NP2 Tec SQA MDA UML e Profiles SoaML Ferramentas Modelo geração de código

### Conceitos introduzidos à UML

- Arquitetura de serviços («ServiceArchitecture»)
  - Considera um conjunto de serviços em um contexto
  - Apresenta como os participantes trabalham juntos
    - para apoiar uma comunidade, processo, conjunto de sistemas ou organização
    - Assumindo papéis de provedores e/ou consumidores de serviço
  - Expressa o uso dos serviços
    - Algum conhecimento sobre as dependências entre os participantes no contexto que define

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

NP2 Tec SQA MDA UML e Profiles SoaML Ferramentas Modelo geração de código

### Conceitos introduzidos à UML

- Arquitetura de serviços («ServiceArchitecture»)

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

## Conceitos introduzidos à UML

- **Arquitetura de serviços («ServiceArchitecture»)**
  - Os mesmos serviços e participantes podem ser usados em várias arquiteturas diferentes
  - Pode representar o domínio do processo de negócio usado na identificação dos serviços
  - Pode ser modelada em dois níveis de granularidade
    - De domínio
      - Participantes independentes trabalhando juntos
    - De um participante
      - Partes de um participante trabalhando juntas
      - Ex.: Departamentos de uma organização

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

## Ferramentas que suportam SoaML

| Ferramenta                                                        | Fabricante                                  | Permite geração de código a partir de modelos? | Gratuita? |
|-------------------------------------------------------------------|---------------------------------------------|------------------------------------------------|-----------|
| Modelio Designer (Free) Edition                                   | Modeliosoft                                 | Não                                            | Sim       |
| Modelio Enterprise Edition                                        | Modeliosoft                                 | Sim                                            | Não       |
| Magic Draw (com CameoSOA)                                         | No Magic                                    | Não                                            | Não       |
| Magic Draw com Model Pro Gold (com CameoSOA, ModelPro e profiles) | Respectivamente, No Magic e ModelDriven.org | Sim                                            | Não       |
| Rational Software Architect                                       | IBM                                         | Sim                                            | Não       |
| Rational Software Modeler                                         | IBM                                         | Sim                                            | Não       |
| Sparx Systems' Enterprise Architect                               | Sparx System's                              | Sim                                            | Não       |

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

---

---

SQA > MDA > UML e Profiles > SoaML > Ferramentas > Modelo > geração de código

## Ferramentas que suportam SoaML

- A ferramenta *Modelio Enterprise Edition* foi escolhida para utilização no minicurso
  - É de fácil utilização
  - Possui uma versão livre para modelagem SoaML
    - Modelio Designer Edition
  - Permite geração de códigos a partir de modelos na sua versão fechada
  - Ferramenta foi disponibilizada para utilização no minicurso

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelio > geração de código

## Ferramentas que suportam SoaML

Algumas características do *Modelio Enterprise Edition*

- Suporta:
  - UML2
  - BPMN
  - SoaML
  - Profiles UML (modelagem e criação)
  - Linguagem de script Jython
- Possui módulos de geração C#, Java, C++, SQL, BPEL, XSD, WSDL e Hibernate
  - A partir de modelos SoaML pode-se gerar código WSDL, Java e XSD

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelio > geração de código

## Ferramentas que suportam SoaML

Algumas características do *Modelio Enterprise Edition*

- Permite:
  - Importação e exportação de modelos XMI entre ferramentas
  - Verificação de modelos
  - Gerenciamento de rastreabilidade
  - Sincronização de modelo com código para Java, C++ e C#
  - Definição de diagramas de objetivos e requisitos
  - Compartilhamento de projeto e gerenciamento de equipe
  - Configuração e gerenciamento de versão com o Subversion

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

---

---

---

---

---

SOA > MDA > UML e Profiles > SoaML > Ferramentas > Modelio > geração de código

## SoaML na prática

Apresentação prática do uso de SoaML com a ferramenta Modelio.

- Criação do projeto:

Núcleo de Pesquisa e Prática em Tecnologia UNIBO

---

---

---

---

---

---

---

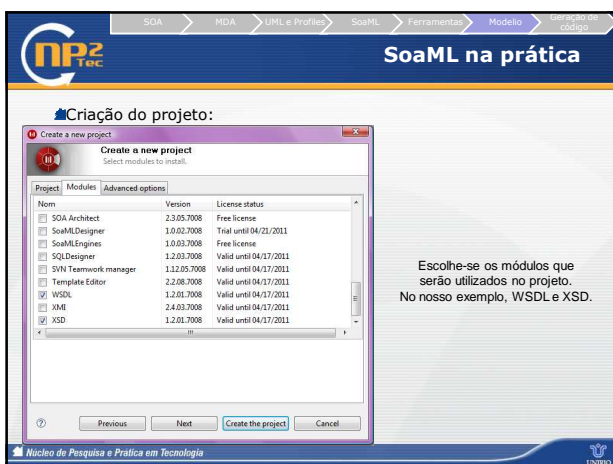
---

---

---

---

---




---

---

---

---

---

---

---

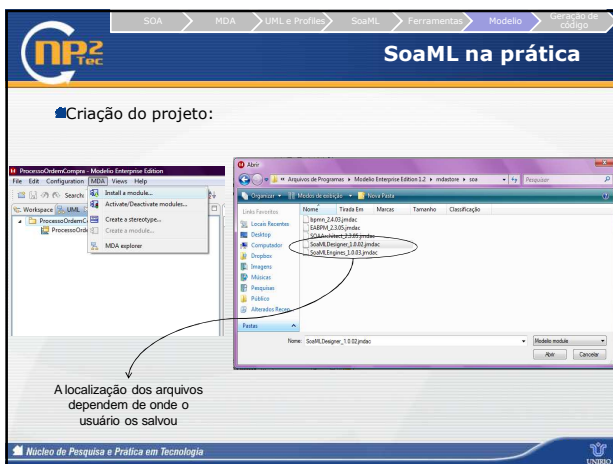
---

---

---

---

---




---

---

---

---

---

---

---

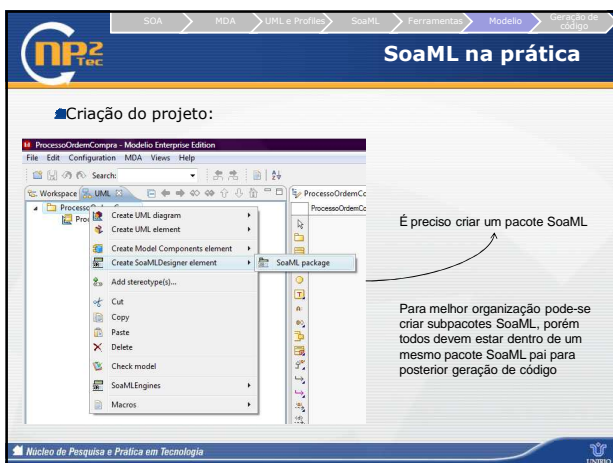
---

---

---

---

---




---

---

---

---

---

---

---

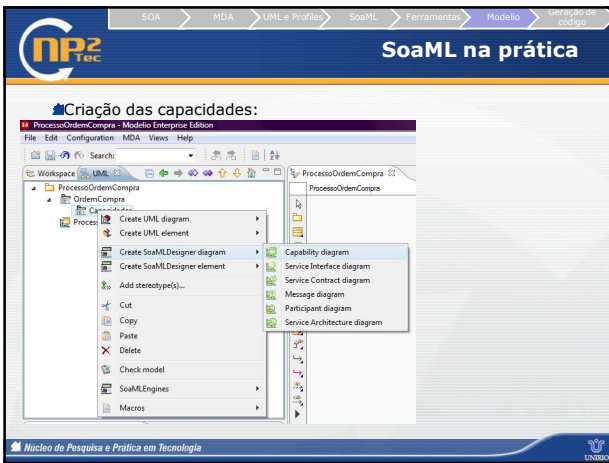
---

---

---

---

---



---

---

---

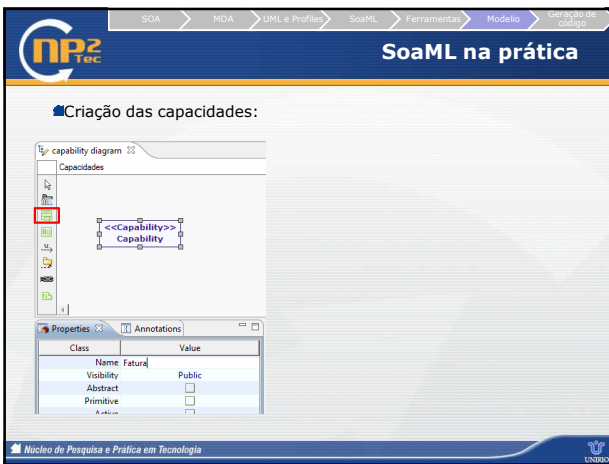
---

---

---

---

---



---

---

---

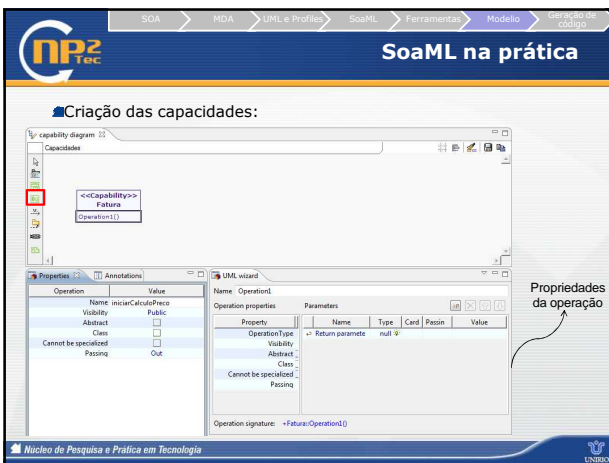
---

---

---

---

---



---

---

---

---

---

---

---

---

SoaML na prática

■ Criação das capacidades:

Diagram showing the creation of capabilities: <<Capability>> OrdemCompra, <<Capability>> Fatura, <<Capability>> Programação, and <<Capability>> Remessa.

Núcleo de Pesquisa e Prática em Tecnologia

---

---

---

---

---

---

---

---

SoaML na prática

■ Criação do modelo de dados:

Diagram showing the creation of the data model: Create UML diagram, Create UML element, Create SoaML Designer diagram, Create SoaML Designer element, Add stereotype(s)..., Cut, Copy, Paste, Delete, Check model, SoaML Engines, Macros.

Núcleo de Pesquisa e Prática em Tecnologia

---

---

---

---

---

---

---

---

SoaML na prática

■ Criação do modelo de dados:

Diagram showing the creation of the data model: <<data type>> OrdemCompra, <<data type>> OrdemCompra, <<data type>> OrdemCompra.

Properties window showing attributes: Name, Visibility, Minimum multiplicity.

Núcleo de Pesquisa e Prática em Tecnologia

---

---

---

---

---

---

---

---



SoaML na prática

**criação do modelo de dados:**

Propriedades da associação

Núcleo de Pesquisa e Prática em Tecnologia

---



---



---



---



---



---



---



---



---

SoaML na prática

**criação do modelo de mensagens:**

A criação de atributos e associações para mensagens é feita da mesma maneira que foi feito nos tipos de dados

Núcleo de Pesquisa e Prática em Tecnologia

---



---



---



---



---



---



---



---



---

SoaML na prática

**criação das interfaces de serviço:**

Núcleo de Pesquisa e Prática em Tecnologia

---



---



---



---



---



---



---



---



---

NP2 Tec SoaML na prática

criação de código

Modelo

Ferramentas

SoaML

UML e Profiles

MDA

SOA

criação das interfaces de serviço simples:

services interface diagram

Interfaces

<<interface: Provider>>  
Programacao

Properties

| Interface  | Value       |
|------------|-------------|
| Name       | Programacao |
| Visibility | Public      |

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

NP2 Tec SoaML na prática

criação de código

Modelo

Ferramentas

SoaML

UML e Profiles

MDA

SOA

criação das interfaces de serviço simples:

services interface diagram

Interfaces

<<interface: Provider>>  
Programacao

solicitarProgramacaoProducao()

UML wizard

Name: solicitarProgramacaoProducao

Operation properties

| Property       | Name             | Type | Card | Passin | V |
|----------------|------------------|------|------|--------|---|
| Operation type | Return parameter | null | V    |        |   |

Operation signature: +Programacao:solicitarProgramacaoProducao()

Propriedades da operação

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

---

---

---

---

---

NP2 Tec SoaML na prática

criação de código

Modelo

Ferramentas

SoaML

UML e Profiles

MDA

SOA

criação das interfaces de serviço ServiceInterface:

services interface diagram

Interfaces

<<ServiceInterface>>  
FaturaService

Properties

| Class      | Value         |
|------------|---------------|
| Name       | FaturaService |
| Visibility | Abstract      |

services interface diagram

Interfaces

Fatura

iniciarCalculoPreco()  
finalizarCalculoPreco()

<<Expose>>

<<ServiceInterface>>  
FaturaService

Capacidade Fatura (do pacote de capacidades) é "arrastada" para o diagrama de interface do serviço

Núcleo de Pesquisa e Prática em Tecnologia UNBIO

---

---

---

---

---

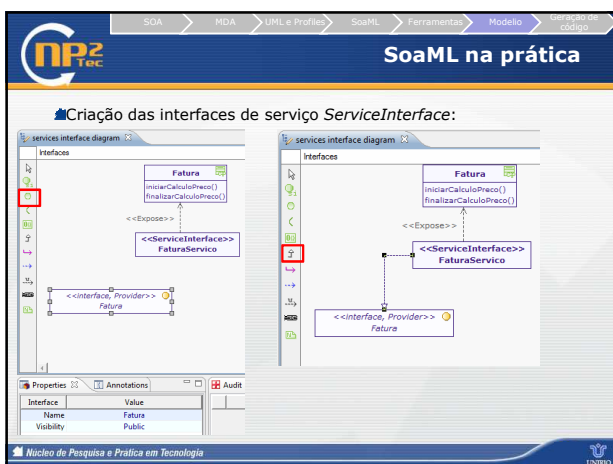
---

---

---

---

---




---

---

---

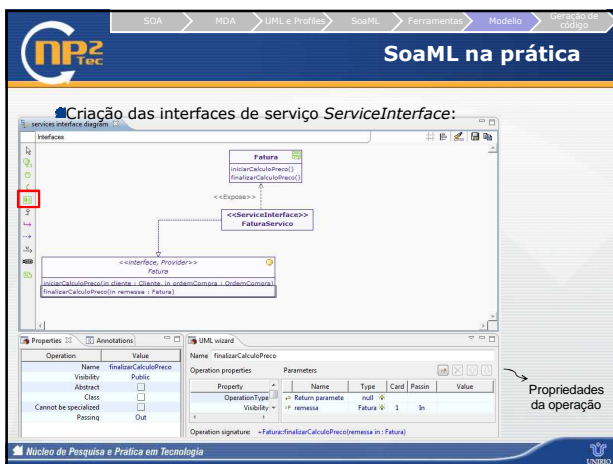
---

---

---

---

---




---

---

---

---

---

---

---

---




---

---

---

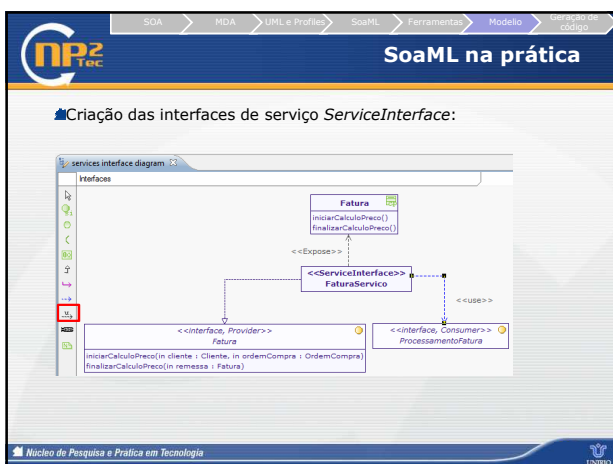
---

---

---

---

---



---

---

---

---

---

---

---

---

---

---



---

---

---

---

---

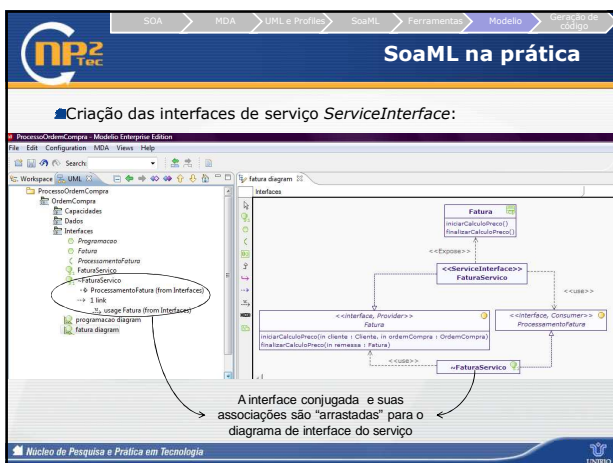
---

---

---

---

---



---

---

---

---

---

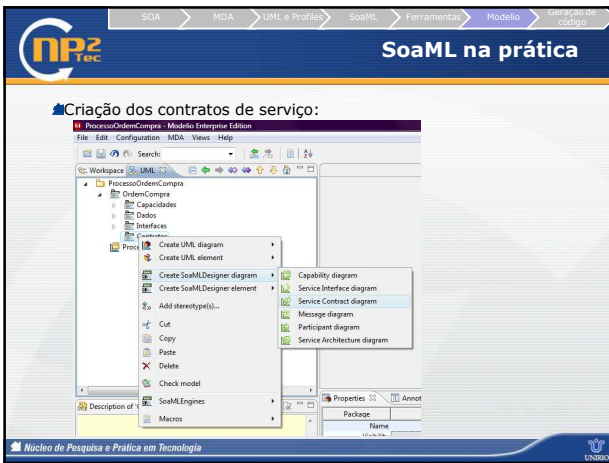
---

---

---

---

---




---

---

---

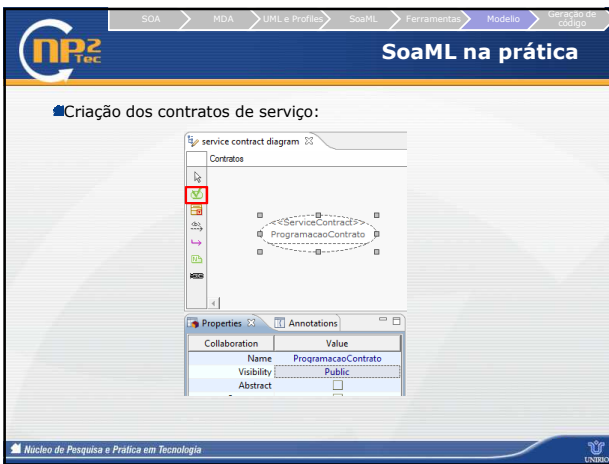
---

---

---

---

---




---

---

---

---

---

---

---

---




---

---

---

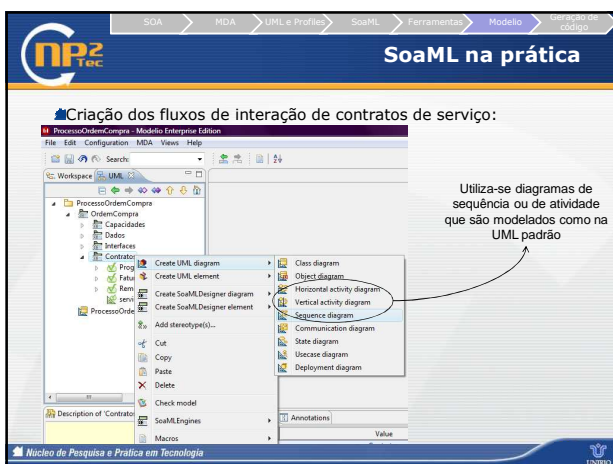
---

---

---

---

---




---

---

---

---

---

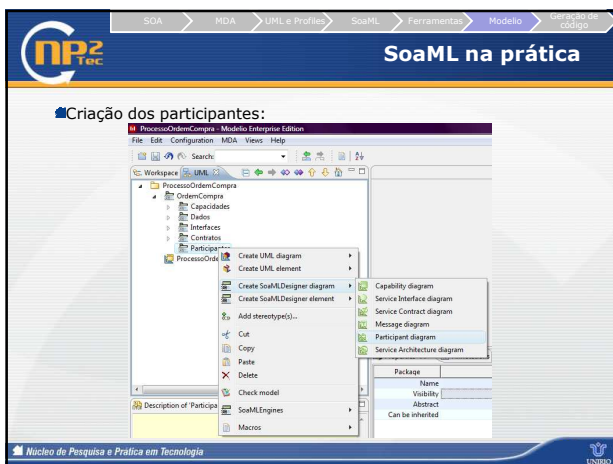
---

---

---

---

---




---

---

---

---

---

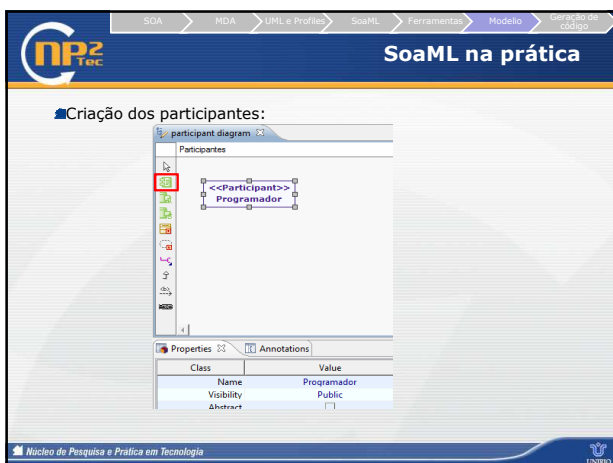
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

SoaML na prática

■ Criação dos participantes (provisão de serviço com interface simples):

The screenshot shows two views of a participant diagram. The left view shows the participant 'Programador' with a provided service interface 'programacao:Programacao'. The right view shows the participant 'Programador' with a provided service interface 'programacao:Programacao' and a provided interface 'programacao (from Interface)'. The Properties window at the bottom shows the 'Provided interfaces' property set to 'programacao (from Interface)'.

Núcleo de Pesquisa e Prática em Tecnologia

---

---

---

---

---

---

---

---

---

---

SoaML na prática

■ Criação dos participantes (provisão de serviço com ServiceInterface):

The screenshot shows three views of a participant diagram. The first view shows the participant 'Fornecedor' with a provided service interface 'fatura:FaturaService'. The second view shows the participant 'Fornecedor' with a provided service interface 'fatura:FaturaService' and a provided interface 'fatura (from Interface)'. The third view shows the participant 'Fornecedor' with a provided service interface 'fatura:FaturaService' and a provided interface 'fatura (from Interface)'. The Properties window at the bottom shows the 'Provided interfaces' property set to 'fatura (from Interface)'.

Núcleo de Pesquisa e Prática em Tecnologia

---

---

---

---

---

---

---

---

---

---

SoaML na prática

■ Criação dos participantes (requisição de serviço):

The screenshot shows a participant diagram for 'ProcessadorOrdem' with a required service interface 'fatura:FaturaService'. The Properties window at the bottom shows the 'Required interface' property set to 'fatura (from Interface)'.

Para consumo de serviços, utiliza-se ponto de requisição.

A definição de interfaces providas e requeridas, segue o mesmo procedimento da provisão de serviços.

Porém, no caso de ServiceInterface é preciso atentar para o tipo da porta de requisição que deve ser uma interface conjugada.

Núcleo de Pesquisa e Prática em Tecnologia

---

---

---

---

---

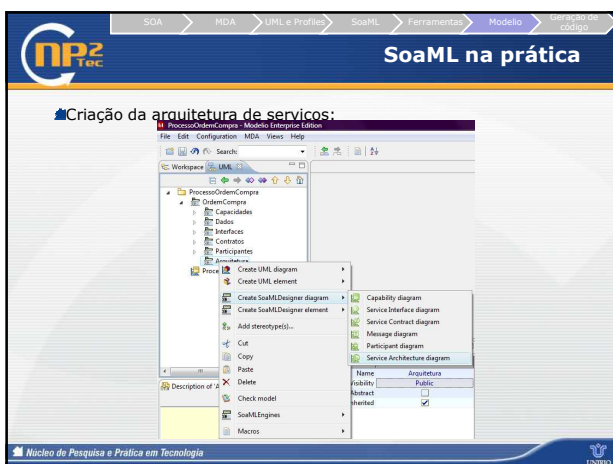
---

---

---

---

---



---

---

---

---

---

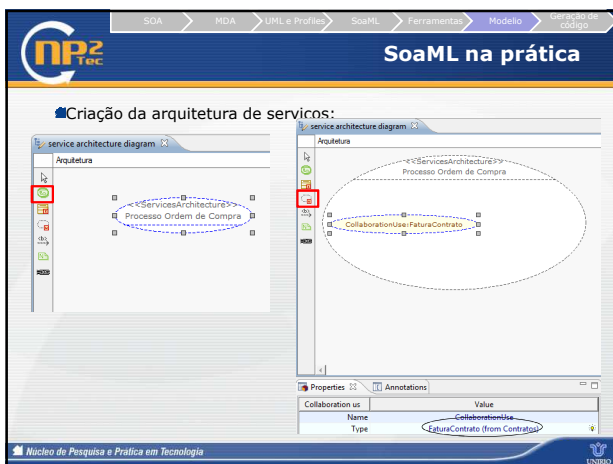
---

---

---

---

---



---

---

---

---

---

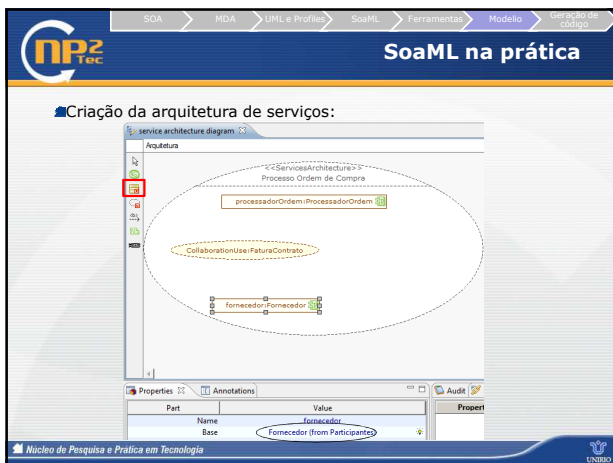
---

---

---

---

---



---

---

---

---

---

---

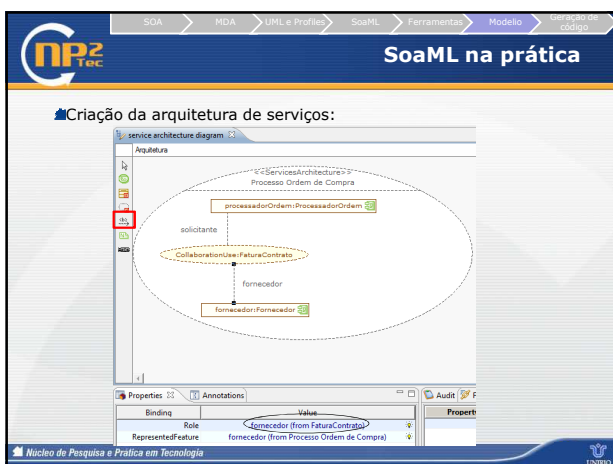
---

---

---

---





---

---

---

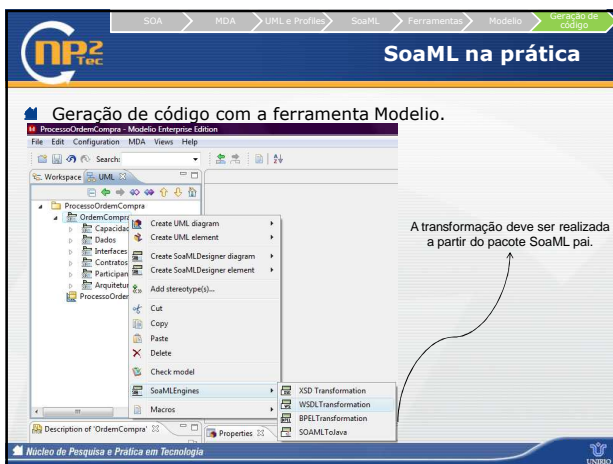
---

---

---

---

---



---

---

---

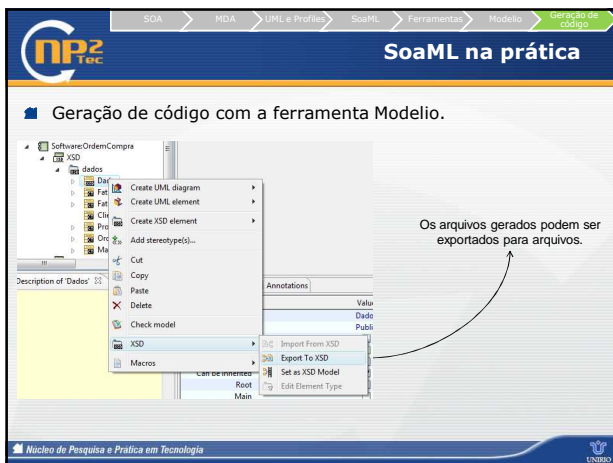
---

---

---

---

---



---

---

---

---

---

---

---

---

**NP2**  
Tec

**Agradecimentos**



**MODELIO**SOFT www.modeliosoft.com

Núcleo de Pesquisa e Prática em Tecnologia



---

---

---

---

---

---

---


---

**NP2**  
Tec

**Perguntas?**

**OBRIGADO!**

Núcleo de Pesquisa e Prática em Tecnologia



---

---

---

---

---

---

---

---