

8 Glossário

API	Application Programming Interface
B2B	Business to Business
BPEL4WS	Business Process Execution Language for Web Services
<i>BPM</i>	Business Process Module
BSC	Business-Service-Computing
CMP	Container Managed Persistence
DAO	Data Access Objects
EAI	Enterprise Application Integration
EJB	Enterprise JavaBeans
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
J2EE	Java 2 Enterprise Edition
JSP	JavaServer Pages
MVC	Model-View-Controller
RPC	Remote Procedure Call
SGML	Standard Generalized Markup Language
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SWC	Semantic Web Challenge
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
URI	Universal Resource Indicator
URL	Universal Resource Locator
W3C	World Wide Web Consortium
WS	Web Service
WS-BPEL	Business Process Execution Language for Web Services
WSD	Web Service Description
WSDL	Web Services Description Language
WSFL	Web Services Flow Language
XLANG	Web Services for Business Process Design
XML	Extensible Markup Language
XSD	XML Schema
XSLT	Extensible Stylesheet Language Transformations

9 Bibliografia

- [1] OASIS Web Services Business Process Execution Language (WS-BPEL) Technical Committee. 2007. Web Services Business Process Execution Language Version 2.0. OASIS. [Online] 31 de janeiro de 2007. Acesso em 5 de abril de 2008. <http://docs.oasis-open.org/wsbpel/2.0/CS01/wsbpel-v2.0-CS01.html>
- [2] Khalaf, R., Mukhi, N. e Weerawarana, S. 2003. Service–Oriented Composition in BPEL4WS. WWW2003, Budapeste, Hungria. [Online] 20-24 de maio de 2003. Acesso em 21 de janeiro de 2008. <http://www.unizh.ch/home/mazzo/reports/www2003conf/papers/P768/p768-khalaf.pdf>.
- [3] Aoyama, M. 2002. A Business-Driven Web Service Creation Methodology. IEEE Symposium on Applications and The Internet (Saint) Workshops, [Online] 2002. Acesso em 6 de abril de 2008. <http://www.seto.nanzan-u.ac.jp/~amikio/NISE/jp/papers/2002-01-WebSEServiceCreation>.
- [4] Srivastava, B. e Koehler, J. 2003. Web Service Composition - Current Solutions and Open Problems. ICAPS 2003 Workshop on Planning for Web Services. pp. 28 - 35. Itália.
- [5] W3C. 2003. Extensible Markup Language (XML). W3C Architecture Domain. [Online] 2003. Acesso em 6 de fevereiro de 2008. <http://www.w3.org/XML/>.
- [6] W3C Working Group. 2004. W3C Working Group Note 11 February 2004. Web Services Achitecture. [Online] 2004. Acesso em 6 de fevereiro de 2008. <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211>.
- [7] SOA Definition team of the SOA Working Group. 2006. Definition of SOA. The Open Group. [Online] 2006. Acesso em 6 de fevereiro de 2008. <http://opengroup.org/projects/soa/doc.tpl?gdid=10632>.
- [8] SAP AG. 2008. SAP Netweaver Developer's Guide. [Online] http://help.sap.com/saphelp_nw70/helpdata/en/8b/0b674240449c60e1000000a1550b0/frameset.htm
- [9] SAP AG. 2008. SAP Netweaver Exchange Infrastructure Knowledge Center. [Online] <https://www.sdn.sap.com/irj/sdn/soa-modeling?rid=/webcontent/uuid/a36de75d-0501-0010-1cad-a7b7650f2d11>
- [10] Karastoyanova, D., Buchmann A.. 2004. Extending Web Service Flow Models to Provide for Adaptability. Proc. *BPMSOA Workshop, OOPSLA '04*, October 2004. [Online]. Acesso em 16 de junho de 2008. <http://www.dvs1.informatik.tu-darmstadt.de/publications/pdf/WS-flow-Adaptability-OOPSLA04.pdf>

- [11] Lins, F. 2007. Composição Adaptativa de Web Services. Dissertação de Mestrado. UFPE. Orientador: Nelson Souto Rosa.
- [12] Martins, R. 2007. Composição Dinâmica de Web Services. Dissertação de Mestrado. Universidade do Vale do Rio dos Sinos. Orientador: Prof. Dr. Sérgio Crespo Coelho da Silva Pinto.
- [13] Zaremba, M., Vitvar, T., Moran, M., Haselwanter, T. 2006. WSMX Discovery for the SWS Challenge. In Proceedings of the Semantic Web Services Challenge - Phase III Workshop. Workshop at 5th International Semantic Web Conference (ISWC 2006), Athens, Georgia, USA. [Online] <http://see.deri.ie/maciej/papers/Workshop/SWS-Challenge2006-phase3-Discovery.pdf>
- [14] Nüßer, W. 2006. Web services or RFCs - choosing the right technology for your SAP integration challenges. SAP Professional Journal, Wellesley Informations Services - Maio/Junho 2006. pp. 66-96. www.sappro.com
- [15] Staab, S. et al. 2003. Web services: Been there, done that? IEEE Intelligent Systems, Jan-Feb issue. pp. 72-85.
- [16] Cardeli, L. Davies, R. 1999. Service Combinators for Web Computing. IEEE Transactions of Software Engineering, v. 25(3), pp. 309-316.
- [17] F. Curbera, R. Khalaf, F. 2004. Implementing BPEL4WS: The Architecture of a BPEL4WS Implementation. Presented at Proceedings of the Grid Workflow Workshop at GGF-10, Berlin, Germany.
- [18] Benatallah. B., Dumas. M., Fauvei. M., Rabhi, F. 2002. Towards Patterns of Web Services Composition. Em: Patters and Skeletons for Parallel and Distributed Computing. Springer Verlag. UK.
- [19] Medjahed, B., Bouguettaya, A., Elmagarmid, A. 2003. Composing Web services on the Semantic Web. The VLDB Journal (2003). Published online: September 23, 2003. Springer-Verlag. Acesso em 9 de junho de 08. http://www.cs.purdue.edu/homes/ake/pub/composition_VLDBJ.pdf.
- [20] Allampallam, R. 2005. Bridging the Sync-Async bridge with Fork (XI). [Online] Posted on Feb. 17, 2005 in SAP Exchange Infrastructure (XI). <https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/1295>

10 Apêndice A – Casos de Teste

Este projeto adotou testes unitários e testes funcionais para avaliar a qualidade do produto final. Os testes unitários contemplaram inclusive os métodos para os quais é necessário tratamento de exceção, verificando se os mesmos apresentavam o comportamento esperado em situações de erro. Os testes funcionais utilizaram checagem de conteúdo automática para validar as mensagens apresentadas ao usuário.

10.1. Testes Unitários

Para os testes unitários da classe *Helper* e EJBs, adotamos o *framework* JUnitEE⁵ versão 1.11, uma versão estendida do JUnit⁶ que suporta testes automatizados em J2EE. O teste dos *servlets* foi realizado com uma aplicação comercial em versão de experimentação, o DevTest4J 9.2.0 da AppPerfect⁷.

No *framework* JUnit todas as unidades testadas estendem a classe *junit.framework.TestCase*. Os casos de teste são identificados automaticamente através do nome padrão da classe de teste criada pelo JUnit tendo seu prefixo "Test".

10.1.1. Testes da classe Helper e EJBs

O JUnitEE disponibiliza um *servlet* por meio do qual é possível chamar métodos no ambiente do WebAS. Associamos um módulo Web correspondente ao JUnitEE à aplicação do fornecedor, e desta forma integramos os testes unitários ao sistema.

Cada subclasse dos testes possui dois métodos que são usados para inicialização (*setUp*) e finalização (*tearDown*). O método de inicialização foi utilizado para criar uma instância do EJB pedido e obter o código pedidold

⁵ JUnitEE – <http://www.junitee.org/>

⁶ JUnit – <http://www.junit.org/>

⁷ DevTest4J – <http://www.appperfect.com/products/devtest.html>

necessário a alguns testes. Separamos os testes de EJB em duas classes, BeanTest.java e BeanTest2.java, sendo que a primeira testa os métodos que dependem da existência de um pedido, e a segunda testa os demais. Realizamos testes de verificação de correção dos retornos (*assertTrue*) e também testes de tratamento de exceção, realizando chamadas com parâmetros inválidos. Indicamos o erro esperado no nome do método.

Apresentamos e comentamos a seguir o resultado testes unitários automáticos acionados pelo URL:

<http://athlon64x2:50100/JUnitEEFramework/servlet/org.junitee.servlet.JUnitEEServlet?all=true>

JUnit Test Results		
Summary of test results		
✓	nom.marcos.HelperTest	0,000 sec
✓	nom.marcos.BeanTest	0,062 sec
✓	nom.marcos.BeanTest2	0,609 sec
List of executed tests		
nom.marcos.HelperTest		
✓	testLeCatalogo(nom.marcos.HelperTest)	0,000 sec
nom.marcos.BeanTest		
✓	testListarItensPedido(nom.marcos.BeanTest)	0,046 sec
✓	testCancelarPedido(nom.marcos.BeanTest)	0,000 sec
✓	testGravarItens(nom.marcos.BeanTest)	0,016 sec
✓	testInserirItens(nom.marcos.BeanTest)	0,000 sec
nom.marcos.BeanTest2		
✓	testCatalogo(nom.marcos.BeanTest2)	0,000 sec
✓	testListarPedidos(nom.marcos.BeanTest2)	0,359 sec
✓	testNovoPedido_solicitanteInvalido(nom.marcos.BeanTest2)	0,188 sec
✓	testCancelarPedido_pedidoInvalido(nom.marcos.BeanTest2)	0,015 sec
✓	testGravarItens_dadosInvalidos(nom.marcos.BeanTest2)	0,016 sec
✓	testPedidos_semSolicitante(nom.marcos.BeanTest2)	0,000 sec
✓	testListarItensPedido_semPedido(nom.marcos.BeanTest2)	0,000 sec
✓	testInserirItens_itemInvalido(nom.marcos.BeanTest2)	0,031 sec

10.1.2. HelperTest

O teste da classe *Helper* consistiu em verificar o retorno do método que lê o arquivo com o catálogo de produtos, que não pode ser nulo ou gerar uma exceção.

10.1.2.1. testLeCatalogo

Verificação se o array *Helper.CATALOGO* contém itens

Resultado esperado: *Helper.CATALOGO* possui 1 ou mais itens

10.1.3. BeanTest

Este primeiro conjunto de testes visa comprovar o funcionamento de operações sobre um pedido. Conforme explicamos, utilizamos o método *setUp* para gerar este pedido, e o código do pedido gerado foi passado para o método sendo testado. O solicitante neste caso foi definido como “TEST”, por meio de uma constante.

10.1.3.1. testListarItensPedido

Verificação do comportamento do método *ListarItensPedido* para um pedido que não tem itens.

Resultado esperado: exceção, pois o pedido não tem itens.

10.1.3.2. testCancelarPedido

Verificação do comportamento do método *CancelarPedido* para um pedido aberto.

Resultado esperado: mensagem de cancelamento.

10.1.3.3. testGravarItens

Verificação do comportamento do método *GravarItens* para um item válido.

Resultado esperado: mensagem de sucesso.

10.1.3.4. testInserirItens

Verificação do comportamento do método *InserirItens* para um item válido.

Resultado esperado: mensagem de sucesso.

10.1.4. BeanTest2

Este segundo conjunto de testes visa comprovar o funcionamento de operações que não dependem da existência de um pedido.

10.1.4.1. testCatalogo

Verificação se o método *Catalogo* retorna itens.

Resultado esperado: o retorno do método possui 1 ou mais itens

**10.1.4.2.
testListarPedidos**

Verificação se o método ListarPedidos retorna itens.

Resultado esperado: o retorno do método possui 1 ou mais pedidos

**10.1.4.3.
testListarPedidos**

Verificação se o método ListarPedidos retorna itens.

Resultado esperado: o retorno do método possui 1 ou mais pedidos

**10.1.4.4.
testNovoPedido_solicitanteInvalido**

Verificação do comportamento do método NovoPedido com um solicitante inválido.

Resultado esperado: exceção.

**10.1.4.5.
testCancelarPedido_pedidoInvalido**

Verificação do comportamento do método CancelarPedido para um pedido inexistente.

Resultado esperado: exceção.

**10.1.4.6.
testGravarItens_dadosInvalidos**

Verificação do comportamento do método GravarItens para um item e solicitante inválidos.

Resultado esperado: exceção.

**10.1.4.7.
testInserirItens_itemInvalido**

Verificação do comportamento do método InserirItens para um item inválido.

Resultado esperado: exceção.

**10.1.4.8.
testPedidos_semSolicitante**

Verificação do comportamento do método Pedidos para um solicitante nulo (“”).

Resultado esperado: exceção.

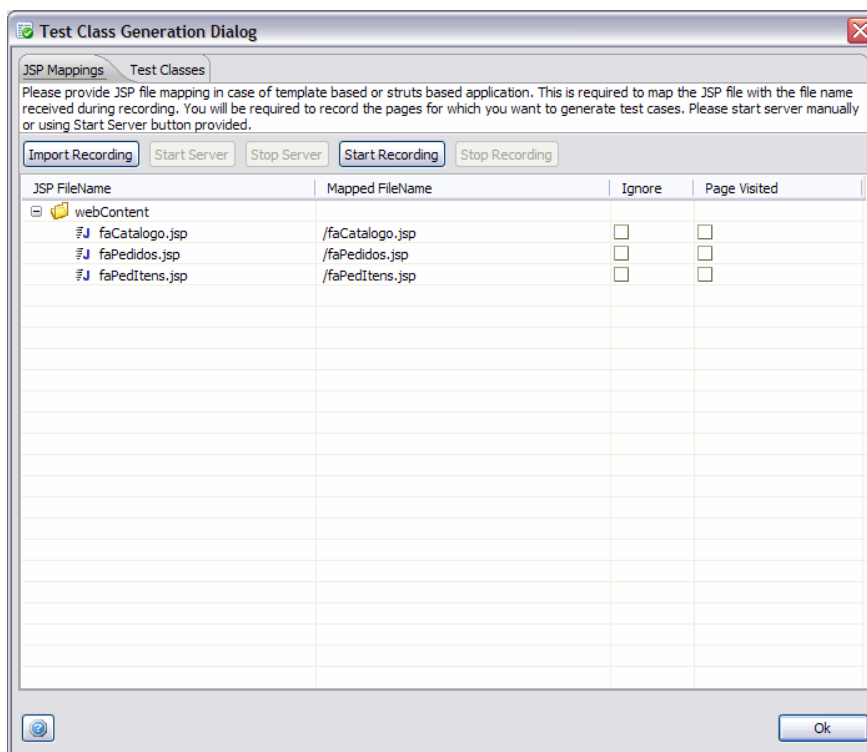
10.1.4.9. testListarItensPedido_semPedido

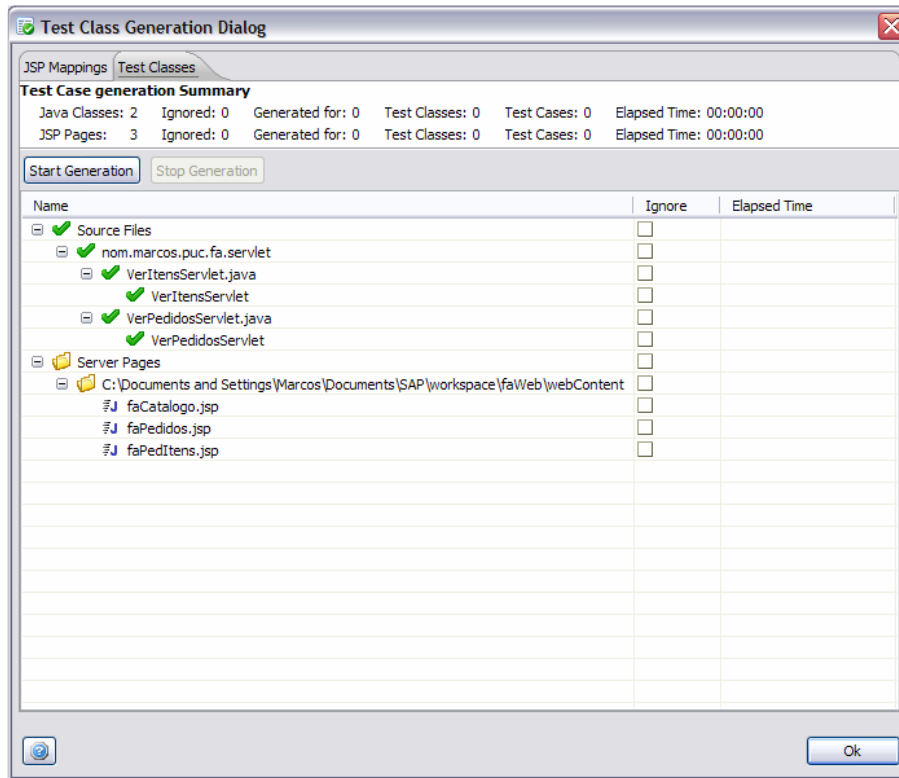
Verificação do comportamento do método ListarItensPedido para um pedido nulo.

Resultado esperado: exceção.

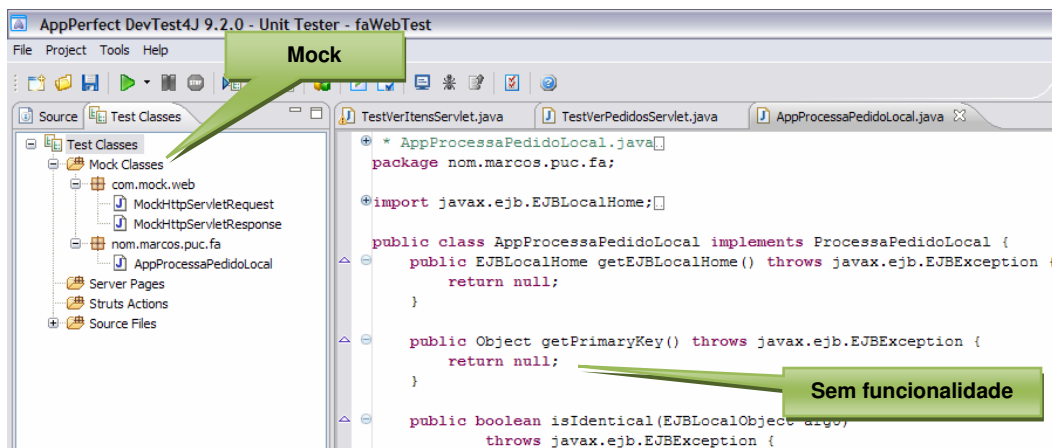
10.1.5. Testes dos Servlets

Para testes unitários dos *servlets* utilizamos o DevTest4J 9.2.0, que gera as classes de teste HttpUnit automaticamente a partir de uma gravação da execução do cenário:



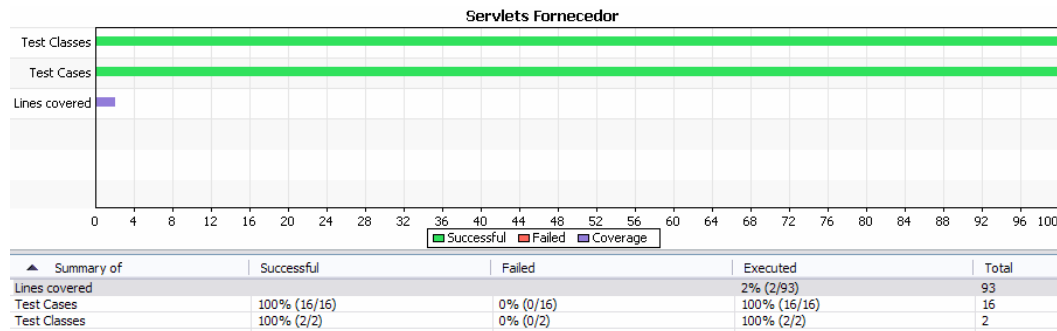


A ferramenta é útil porque o código gerado baseia-se em *Mock Classes*, que atuam como duplês das classes originais, mas sem suas funcionalidades. Desta forma conseguimos o nível de isolamento necessário para testar os métodos dos servlets que dependem basicamente do contexto do servidor de aplicações web.



Apenas para registro, esta técnica de *mock classes* é empregada em outros *frameworks* de teste unitário, como os derivados do JUnit. O mérito da ferramenta é gerar estas classes automaticamente a partir de um *proxy* que é configurado no navegador. Conforme a documentação do produto, é possível inclusive gerar classes que simulam um banco de dados relacional.

Os resultados foram os seguintes:



Este quadro mostra o sucesso na execução de 16 casos de teste nos dois *servlets*. A parte referente à cobertura dos testes em função das linhas de código deve ser ignorada, trata-se de uma funcionalidade do programa que não foi devidamente configurada.

Os testes referiram-se à instanciação das classes e chamada de seus métodos *doGet*, *doPost*, *doWork*, e *handleRequest*, sem contudo efetuar operação alguma no WebAS.

10.2. Testes Funcionais

Os testes funcionais avaliaram o sistema a partir da camada de *interface* JSP (módulo do fornecedor) e *Web Dynpro* (módulo do solicitante), e foram realizados com a ferramenta *QuickTest Professional 9.2* da HP Software⁸ (antiga Mercury Interactive).

10.2.1. Teste do Módulo do Fornecedor

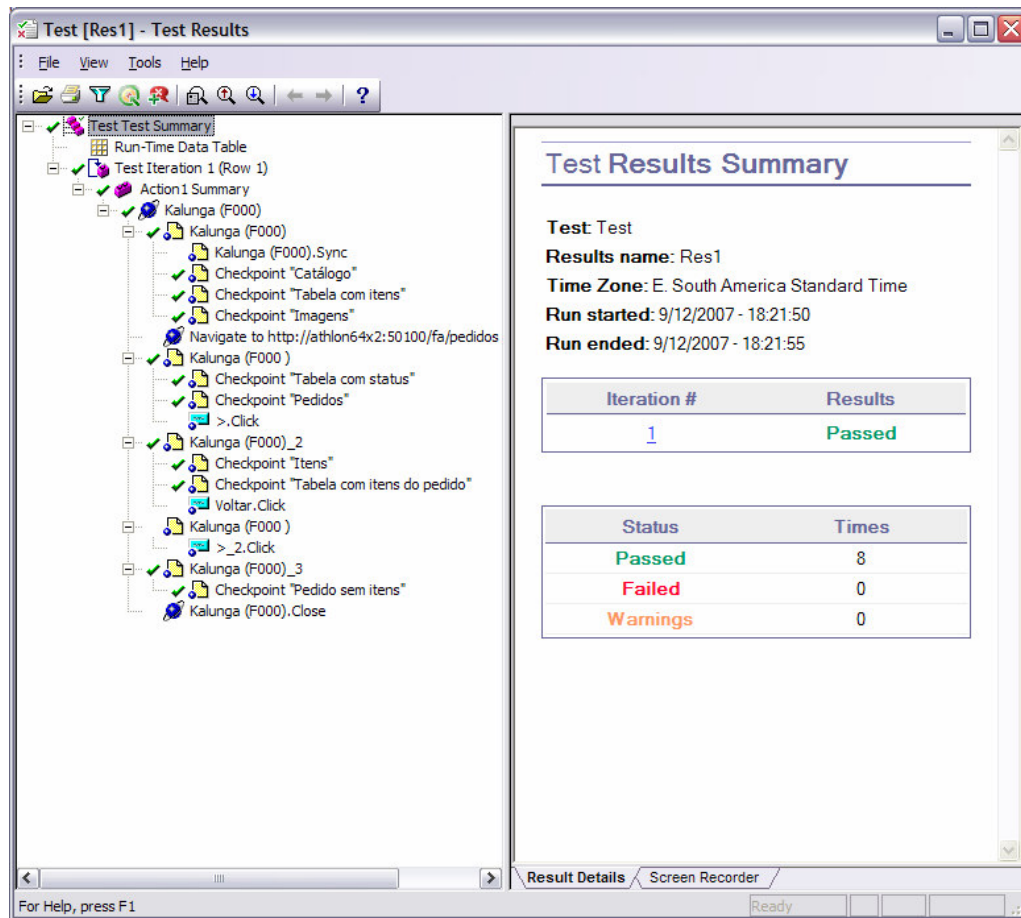
O teste do módulo do fornecedor (JSP) compreendeu 8 verificações:

1. Abrir página com o catálogo, verificar texto do cabeçalho.
2. Na mesma tela, verificar a presença de uma tabela com a coluna itens.
3. Na mesma tela, verificar se 6 imagens foram carregadas.
4. Abrir página com os pedidos, verificar texto do cabeçalho.
5. Na mesma tela, verificar a presença de uma tabela com a coluna status.
6. Clicar em um pedido com itens. Verificar texto do cabeçalho.

⁸ QuickTest -

https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-127-24^1352_4000_100__

7. Na mesma tela, verificar a presença de uma tabela com a coluna itens.
8. Clicar em voltar, então clicar em um pedido sem itens. Verificar a mensagem “sem itens”

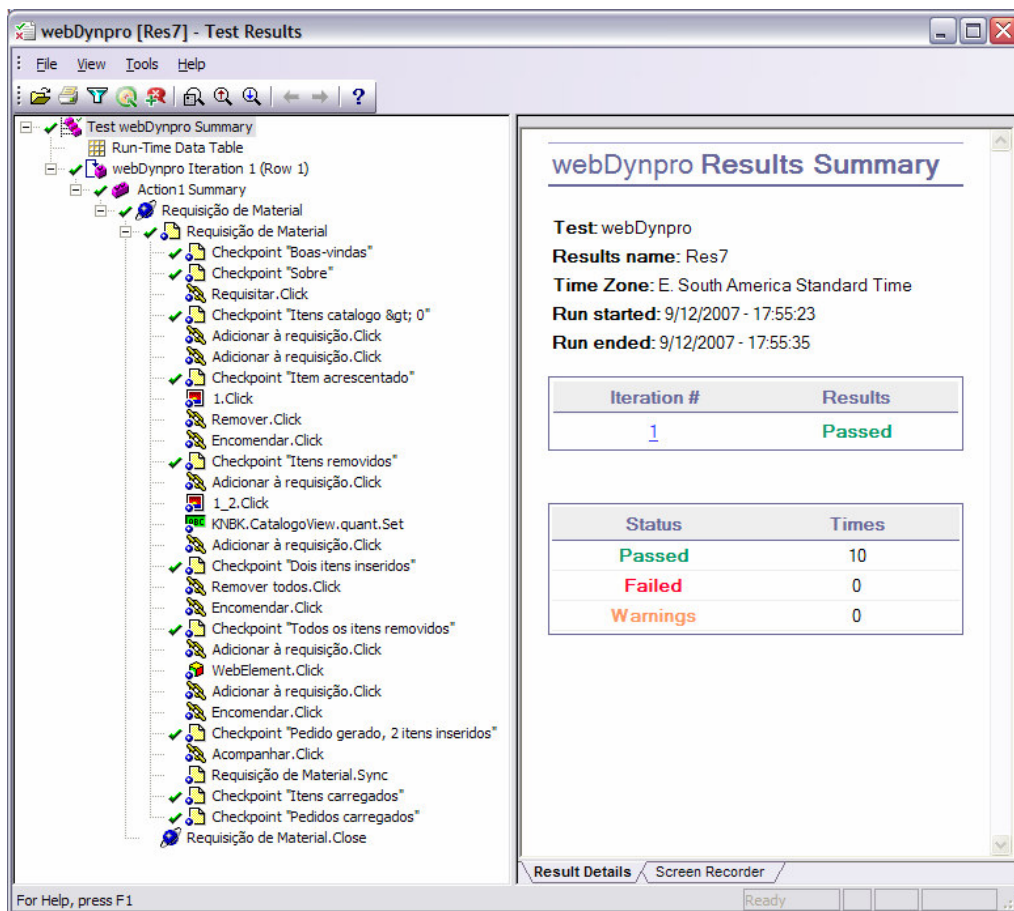


10.2.2. Teste do Módulo do Solicitante

O teste do módulo dos solicitantes (Web Dynpro) compreendeu 10 verificações na seguinte execução de operações:

1. Abrir página inicial, verificar a existência da mensagem de boas-vindas.
2. Na mesma tela, verificar a existência da informação do nome do programa
3. Abrir a página de requisição. Verificar se catálogo foi carregado.
4. Acrescentar o primeiro item 2 vezes. Verificar se há 1 item com quantidade 2.
5. Selecionar o item e removê-lo. Pressionar Encomendar. Verificar mensagem de erro indicando requisição vazia.

6. Adicionar 2 itens diferentes à requisição, um deles com quantidade maior que 1. Verificar se o total de itens na requisição é 2.
7. Pressionar Remover Todos. Pressionar Encomendar. Verificar mensagem de erro indicando requisição vazia.
8. Adicionar 2 itens diferentes à requisição. Encomendar. Verificar mensagem indicando a geração de pedido com 2 itens.
9. Abrir tela de acompanhamento. Verificar se 1 ou mais pedidos foram carregados.
10. Na mesma tela, verificar se 1 ou mais itens foram carregados.



11

Apêndice B – Códigos-fonte WS-BPEL

11.1.FlexComp_PAR.bpel

```

<?xml version="1.0" encoding="UTF-8"?>
<process name="FlexComp_PAR" targetNamespace="http://marcos.nom/zebpxi"
xsi:schemalocation="BPEL4WS.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:tns="http://marcos.nom/zebpxi" xmlns:sapxi="http://www.sap.com/xi/">

<!--
+-----+
| Definições |
+-----+
-->

    <partnerLinks>
        <partnerLink name="Route_Link" partnerLinkType="sapxi:Route_LT"
partnerRole="Route_Role" />
        <partnerLink name="Transform_Link" partnerLinkType="sapxi:Transform_LT"
partnerRole="Transform_Role" />
        <partnerLink name="CatalogoXI_AbstractSync_Link"
partnerLinkType="tns:CatalogoXI_AbstractSync_LT"
myRole="CatalogoXI_AbstractSync_Role_myRole" />
        <partnerLink name="Catalogo_RespAbstractAsync_rec_299_Link"
partnerLinkType="tns:Catalogo_RespAbstractAsync_LT"
myRole="Catalogo_RespAbstractAsync_Role_myRole" />
        <partnerLink name="CatalogoXI_ReqAbstractAsync_send_154_Link"
partnerLinkType="tns:CatalogoXI_ReqAbstractAsync_LT"
myRole="CatalogoXI_ReqAbstractAsync_Role_partnerRole" />
    </partnerLinks>
    <correlationSets>
<!-- Correlacao: relaciona a mensagem assincrona enviada aos BPMs dos fornecedores
com as respostas -->
        <correlationSet name="Correlacao"
properties="tns:Correlacao|ID|8c354dd0f3c611dca695000b6a102e86 " />
    </correlationSets>
    <variables>
<!-- ColResp: armazena as respostas dos fornecedores (multilinha) -->
        <variable name="ColResp" messageType="http://marcos.nom/zebpxi:Fornecedor"
sap-extn:multiline="true" xmlns:sap-extn="http://www.sap.com/xi/extensions" />
        <variable name="ColResp_headerline"
messageType="http://marcos.nom/zebpxi:Fornecedor" />

```

```

    <variable name="ColResp_operand1"
messageType="http://marcos.nom/zebpxi:Fornecedor" />
<!-- ColResp2: armazena as respostas válidas dos fornecedores (multilinha) -->
    <variable name="ColResp2" messageType="http://marcos.nom/zebpxi:Fornecedor"
sap-extn:multiline="true" xmlns:sap-extn="http://www.sap.com/xi/extensions" />
    <variable name="ColResp2_headerline"
messageType="http://marcos.nom/zebpxi:Fornecedor" />
<!-- Count: conta as respostas válidas dos fornecedores, é usado também como
índice -->
    <variable name="Count" type="xsd:integer" />
<!-- NumResp: conta as respostas recebidas dos fornecedores -->
    <variable name="NumResp" type="xsd:integer" />
<!-- Receivers: armazena o resultado da etapa de determinação de destinatários
realizada pelo XI -->
    <variable name="Receivers" type="sapxi:receiver" sap-extn:multiline="true"
xmlns:sap-extn="http://www.sap.com/xi/extensions" />
    <variable name="Receivers_headerline" type="sapxi:receiver" />
<!-- ReqX: requisição síncrona. O X indica que a mesma está no formato do
mediador, não do fornecedor -->
    <variable name="ReqX" messageType="http://marcos.nom/zebpxi:FornecedorXI" />
<!-- Resp: resposta síncrona -->
    <variable name="Resp" messageType="http://marcos.nom/zebpxi:Fornecedor" />
<!-- As variáveis abaixo são para a fase de determinaçãod de destinatários. Foram
geradas pelo XI -->
    <variable name="Route_packedInMsg" messageType="sapxi:inRouteMsg" />
    <variable name="Route_packedOutMsg" messageType="sapxi:outRouteMsg" />
    <variable name="Transform_packedInMsg" messageType="sapxi:inTransformMsg" />
    <variable name="Transform_packedOutMsg" messageType="sapxi:outTransformMsg" />
</variables>

<!--
+-----+
| Início do processo |
+-----+
-->

    <sequence>
<!-- Recebe a requisição síncrona, em formato do mediador. Armazena em ReqX -->
    <receive name="Recebe Req Sinc" sap-extn:mode="bridge" variable="ReqX"
createInstance="yes" sap-extn:messageExchange="Recebe Req Sinc"
portType="http://marcos.nom/zebpxi:CatalogoXI_AbstractSync"
operation="CatalogoXI_AbstractSync" sap-
extn:syncInfNamespace="http://marcos.nom/zebpxi"
partnerLink="CatalogoXI_AbstractSync_Link" xmlns:sap-
extn="http://www.sap.com/xi/extensions" />
<!-- Etapa de determinação de destinatários. Armazena em Receivers -->
    <sequence sap-extn:stepType="ROUTE" sap-extn:stepName="Consulta Fornecedores"
xmlns:sap-extn="http://www.sap.com/xi/extensions">
    <assign name="packSendMsg">
    <copy>

```

```

        <from expression="sap-
extn:packSendMsgInRouteParam(&apos;Route_packedInMsg&apos;;, &apos;sendMsg&apos;; ,
&apos;ReqX&apos;;, &apos;&apos;;, &apos;FornecedorXI&apos;)" />
        <to variable="Route_packedInMsg" />
    </copy>
</assign>
<assign name="packSendContext">
    <copy>
        <from expression="sap-
extn:packSendContextInRouteParam(&apos;Route_packedInMsg&apos;;,
&apos;sendcontext&apos;; , &apos;&apos;)" />
        <to variable="Route_packedInMsg" />
    </copy>
</assign>
<invoke partnerLink="Route_Link" portType="sapxi:Route_PT"
operation="RouteOp" inputVariable="Route_packedInMsg"
outputVariable="Route_packedOutMsg" />
<assign name="unPackReceivers">
    <sap-extn:append xmlns:sap-
extn="http://schemas.xmlsoap.org/ws/2003/03/business-process/">
        <sap-extn:from sap-extn:expression="sap-
extn:unpackReceiversOutOfRouteParam( &apos;Route_packedOutMsg&apos;)" xmlns:sap-
extn="http://www.sap.com/xi/extensions" />
        <sap-extn:to sap-extn:variable="Receivers" xmlns:sap-
extn="http://www.sap.com/xi/extensions" />
    </sap-extn:append>
</assign>
</sequence>
<!-- Envia a mensagem assíncrona para os BPMs dos destinatários. Ativa a
correlação -->
    <invoke name="Envia Fornecedores" sap-extn:receiver="Receivers" sap-
extn:mode="async" inputVariable="ReqX"
portType="http://marcos.nom/zebpxi:CatalogoXI_ReqAbstractAsync"
operation="CatalogoXI_ReqAbstractAsync"
partnerLink="CatalogoXI_ReqAbstractAsync_send_154_Link" xmlns:sap-
extn="http://www.sap.com/xi/extensions">
        <correlations>
            <correlation set="Correlacao" initiate="yes" />
        </correlations>
    </invoke>
<!-- Zera o contador NumResp -->
    <assign name="Inicializa NumResp">
        <copy>
            <from>&apos;0&apos;</from>
            <to variable="NumResp" />
        </copy>
    </assign>

<!--
+-----+
| Coleta de Respostas|

```

```

+-----+
-->

<!-- Bloco de processamento paralelo. Equivalente SAP ao forEach mode="parallel"
-->
    <sap-extn:parforeach name="Recebe Fornecedores" select="Receivers"
variable="Receiver" xmlns:sap-extn="http://www.sap.com/xi/extensions">
<!-- Condição de término: o número de respostas recebidas for igual a num, ou
todos os bpmns listados em Receivers retornarem -->
    <sap-extn:completionCondition condition="(NumResp =
ReqX./p1:CatalogoXI/p1:num)" />
    <variables>
<!-- Receiver: o destinatário em questão -->
        <variable name="Receiver" type="sapxi:receiver" />
<!-- RespF: mensagem recebida do destinatário em questão -->
        <variable name="RespF" messageType="http://marcos.nom/zebpxi:Fornecedor"
/>
    </variables>
    <sequence>
<!-- Recebe retorno do bpm, faz correlação com a mensagem enviada-->
        <receive name="Recebe Forn" sap-extn:mode="async" variable="RespF"
partnerLink="Catalogo_RespAbstractAsync_rec_299_Link"
portType="http://marcos.nom/zebpxi:Catalogo_RespAbstractAsync"
operation="Catalogo_RespAbstractAsync">
            <correlations>
                <correlation set="Correlacao" initiate="no" />
            </correlations>
        </receive>
<!-- Verifica se a mensagem é válida, inspecionando a existência do campo prazo-->
        <switch name="Verifica Retorno">
            <case
condition="(RespF./p1:CatalogoResponse/p1:Response/p2:ItemModel/p2:prazo EX )">
<!-- Caso positivo, coleta-->
                <sequence>
                    <assign name="Coleta Resp Forn">
                        <copy>
                            <from variable="RespF"
part="http://marcos.nom/zebpxi:Fornecedor" />
                            <to variable="ColResp_headerline"
part="http://marcos.nom/zebpxi:Fornecedor" />
                        </copy>
                    </assign>
<!-- Armazena no elemento multilinha-->
                    <assign sap-extn:linkTo="Coleta Resp Forn">
                        <sap-extn:append>
                            <sap-extn:from sap-extn:variable="ColResp_headerline" />
                            <sap-extn:to sap-extn:variable="ColResp" />
                        </sap-extn:append>
                    </assign>
<!-- Incrementa o contador-->
                    <assign name="Incr NumResp">

```



```

        <copy>
            <from expression=" NumResp + &apos;1&apos;" />
            <to variable="NumResp" />
        </copy>
    </assign>
</sequence>
</case>
<!-- Caso negativo, ignora-->
    <otherwise>
        <sequence>
            <empty />
        </sequence>
    </otherwise>
</switch>
</sequence>
</sap-extn:parforeach>

<!--
+-----+
| Agregação de Respostas|
+-----+
-->

    <switch name="Verifica se ha resp">
<!-- Verifica se foram recebidas respostas válidas -->
        <case condition="(NumResp = 0)">
            <sequence>
<!-- Inicializa índice -->
                <assign name="Inicializa Count">
                    <copy>
                        <from>&apos;0&apos;</from>
                        <to variable="Count" />
                    </copy>
                </assign>
<!-- Enquanto índice diferente do número de respostas (não há operação menor que)
-->
                    <while name="Processa RespF Coletados" condition="(Count ≠
ReqX./p1:CatalogoXI/p1:num AND ColResp[Count] EX ) OR (Count = 0 AND ColResp[1] EX
)">
                        <sequence>
<!-- Incrementa o índice -->
                            <assign name="Incrementa Count">
                                <copy>
                                    <from expression=" Count + &apos;1&apos;" />
                                    <to variable="Count" />
                                </copy>
                            </assign>
<!-- Copia elementos válidos de ColResp para ColResp2 -->
                                <assign sap-extn:linkTo="Transfere ColResp" xmlns:sap-
extn="http://www.sap.com/xi/extensions">
                                    <sap-extn:copy>

```

```

        <sap-extn:from sap-extn:variable="ColResp" sap-
extn:index="bpws:getVariableData (&apos;Count&apos;)" />
        <sap-extn:to sap-extn:variable="ColResp_operand1" />
    </sap-extn:copy>
</assign>
<assign name="Transfere ColResp">
    <copy>
        <from variable="ColResp_operand1"
part="http://marcos.nom/zebpxi:Fornecedor" />
        <to variable="ColResp2_headerline"
part="http://marcos.nom/zebpxi:Fornecedor" />
    </copy>
</assign>
    <assign sap-extn:linkTo="Transfere ColResp" xmlns:sap-
extn="http://www.sap.com/xi/extensions">
        <sap-extn:append>
            <sap-extn:from sap-extn:variable="ColResp2_headerline" />
            <sap-extn:to sap-extn:variable="ColResp2" />
        </sap-extn:append>
    </assign>
</sequence>
</while>
<!-- Transforma ColResp2 em Resp por meio de agregação -->
    <sequence sap-extn:stepType="TRANSFORM" sap-extn:stepName="Gera Resp"
sap-extn:mapProgram="http://marcos.nom/zebpxi:Catalogo_CollectMerge" xmlns:sap-
extn="http://www.sap.com/xi/extensions">
        <assign>
            <sap-extn:copy>
                <sap-extn:from sap-extn:variable="ColResp2" />
                <sap-extn:to sap-extn:variable="ColResp2_headerline" />
            </sap-extn:copy>
        </assign>
        <assign name="packMsg#1">
            <copy>
                <from expression="sap-
extn:packMsgInTransformParam (&apos;Transform_packedInMsg&apos;;, &apos;inMsg&apos;;
, &apos;ColResp2_headerline&apos;;, &apos;Catalogo_RespAbstractAsync&apos;;,
&apos;http://marcos.nom/zebpxi&apos;;, &apos;&apos;;, &apos;Fornecedor&apos;)" />
                <to variable="Transform_packedInMsg" />
            </copy>
        </assign>
    <!-- Chama a operação de transformação -->
        <invoke partnerLink="Transform_Link" portType="sapxi:Transform_PT"
operation="TransformOp" inputVariable="Transform_packedInMsg"
outputVariable="Transform_packedOutMsg" />
        <assign name="unpackMsg#1">
            <copy>
                <from expression="sap-
extn:unpackMsgOutofTransformParam (&apos;Transform_packedOutMsg&apos;;,
&apos;outMsg&apos;;, &apos;Catalogo_RespAbstractAsync&apos;;,
&apos;http://marcos.nom/zebpxi&apos;;, &apos;Fornecedor&apos;)" />

```

```

<!-- Grava o resultado em Resp -->
    <to variable="Resp" />
    </copy>
  </assign>
</sequence>
</sequence>
</case>
<!-- Não há respostas válidas, gera resposta vazia -->
  <otherwise>
    <sequence>
      <sequence sap-extn:stepType="TRANSFORM" sap-extn:stepName="Gera Resp
Vazia" sap-extn:mapProgram="http://marcos.nom/zebpxi:ReqX_RespVazia" xmlns:sap-
extn="http://www.sap.com/xi/extensions">
        <assign name="packMsg#1">
          <copy>
            <from expression="sap-
extn:packMsgInTransformParam(&apos;Transform_packedInMsg&apos;;, &apos;inMsg&apos;;
, &apos;ReqX&apos;;, &apos;CatalogoXI_ReqAbstractAsync&apos;;,
&apos;http://marcos.nom/zebpxi&apos;;, &apos;&apos;;, &apos;FornecedorXI&apos;)" />
            <to variable="Transform_packedInMsg" />
          </copy>
        </assign>
      <!-- Chama a operação de transformação que irá gerar a resposta vazia -->
      <invoke partnerLink="Transform_Link" portType="sapxi:Transform_PT"
operation="TransformOp" inputVariable="Transform_packedInMsg"
outputVariable="Transform_packedOutMsg" />
      <assign name="unpackMsg#1">
        <copy>
          <from expression="sap-
extn:unpackMsgOutofTransformParam(&apos;Transform_packedOutMsg&apos;;,
&apos;outMsg&apos;;, &apos;Catalogo_RespAbstractAsync&apos;;,
&apos;http://marcos.nom/zebpxi&apos;;, &apos;Fornecedor&apos;)" />
        </copy>
      </assign>
    </sequence>
  </otherwise>
</switch>

<!--
+-----+
| Envio do Resultado |
+-----+
-->

<!-- Envia Resp, fechando a ponte sync/async -->
  <reply name="Envia Resp Sinc" sap-extn:mode="bridge" sap-
extn:messageExchange="Recebe Req Sinc" variable="Resp"
portType="http://marcos.nom/zebpxi:CatalogoXI_AbstractSync"

```

```

operation="CatalogoXI_AbstractSync" sap-
extn:syncInfNamespace="http://marcos.nom/zebpxi"
partnerLink="CatalogoXI_AbstractSync_Link" xmlns:sap-
extn="http://www.sap.com/xi/extensions" />
  </sequence>
</process>

```

11.2.Forn_BPM.bpel

```

<?xml version="1.0" encoding="UTF-8"?>
<process name="Forn_BPM" targetNamespace="http://marcos.nom/zebpxi"
xsi:schemalocation="BPEL4WS.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:tns="http://marcos.nom/zebpxi" xmlns:sapxi="http://www.sap.com/xi/">

<!--
+-----+
| Definições |
+-----+
-->

  <partnerLinks>
    <partnerLink name="Transform_Link" partnerLinkType="sapxi:Transform_LT"
partnerRole="Transform_Role" />
    <partnerLink name="Catalogo_ReqAbstractAsync_rec_3_Link"
partnerLinkType="tns:Catalogo_ReqAbstractAsync_LT"
myRole="Catalogo_ReqAbstractAsync_Role_myRole" />
    <partnerLink name="Catalogo_AbstractSync_send_5_Link"
partnerLinkType="tns:Catalogo_AbstractSync_LT"
myRole="Catalogo_AbstractSync_Role_partnerRole" />
    <partnerLink name="Catalogo_RespAbstractAsync_send_7_Link"
partnerLinkType="tns:Catalogo_RespAbstractAsync_LT"
myRole="Catalogo_RespAbstractAsync_Role_partnerRole" />
  </partnerLinks>
  <variables>
<!-- Req: Requisição a ser enviada ao fornecedor -->
    <variable name="Req" messageType="http://marcos.nom/zebpxi:Fornecedor" />
<!-- Resp: Resposta a ser enviada ao bpm principal -->
    <variable name="Resp" messageType="http://marcos.nom/zebpxi:Fornecedor" />
<!-- As variáveis abaixo são relativas ao bloco de transformação. Foram geradas
pelo XI -->
    <variable name="Transform_packedInMsg" messageType="sapxi:inTransformMsg" />
    <variable name="Transform_packedOutMsg" messageType="sapxi:outTransformMsg" />
  </variables>

<!--
+-----+
| Início do processo |
+-----+
-->

```

```

<sequence>
<!-- Recebe a requisição assíncrona, já em formato do fornecedor. Armazena em Req
-->
    <receive name="Recebe do BPM" sap-extn:mode="async" variable="Req"
createInstance="yes" partnerLink="Catalogo_ReqAbstractAsync_rec_3_Link"
portType="http://marcos.nom/zebpxi:Catalogo_ReqAbstractAsync"
operation="Catalogo_ReqAbstractAsync" xmlns:sap-
extn="http://www.sap.com/xi/extensions" />
    <scope name="Bloco de Envio Forn">
        <faultHandlers>
<!-- Define um bloco para o caso em que o fornecedor não responda dentro do prazo
limite. -->
            <catch faultName="erroEnvio">
                <sequence>
<!-- Gera resposta, tranformando a requisição em resposta vazia -->
                    <sequence sap-extn:stepType="TRANSFORM" sap-extn:stepName="Gera Resp
Vazia" sap-extn:mapProgram="http://marcos.nom/zebpxi:Req_RespVazia" xmlns:sap-
extn="http://www.sap.com/xi/extensions">
                        <assign name="packMsg#1">
                            <copy>
                                <from expression="sap-
extn:packMsgInTransformParam(&apos;Transform_packedInMsg&apos;;, &apos;inMsg&apos;;
, &apos;Req&apos;;, &apos;Catalogo_ReqAbstractAsync&apos;;,
&apos;http://marcos.nom/zebpxi&apos;;, &apos;&apos;;, &apos;Fornecedor&apos;)" />
                                    <to variable="Transform_packedInMsg" />
                                </copy>
                            </assign>
                            <invoke partnerLink="Transform_Link" portType="sapxi:Transform_PT"
operation="TransformOp" inputVariable="Transform_packedInMsg"
outputVariable="Transform_packedOutMsg" />
                            <assign name="unpackMsg#1">
                                <copy>
                                    <from expression="sap-
extn:unpackMsgOutofTransformParam(&apos;Transform_packedOutMsg&apos;;,
&apos;outMsg&apos;;, &apos;Catalogo_RespAbstractAsync&apos;;,
&apos;http://marcos.nom/zebpxi&apos;;, &apos;Fornecedor&apos;)" />
                                </copy>
                            </assign>
                        </sequence>
                    </sequence>
                </catch>
            </faultHandlers>

<!--
+-----+
| Bloco de envio síncrono |
+-----+
-->

```

```

    <sequence>
<!-- Envia requisição síncrona para o fornecedor, grava o resultado em Resp -->
    <invoke name="Envia para Forn" sap-extn:context="" sap-extn:mode="sync"
inputVariable="Req" outputVariable="Resp"
portType="http://marcos.nom/zebpixi:Catalogo_AbstractSync"
operation="Catalogo_AbstractSync" sap-
extn:syncInfNamespace="http://marcos.nom/zebpixi"
partnerLink="Catalogo_AbstractSync_send_5_Link" xmlns:sap-
extn="http://www.sap.com/xi/extensions">
    <catch faultName="sapxi:system_exception">
        <throw name="system_exception" faultName="erroEnvio" />
    </catch>
    <catch faultName="http://marcos.nom/zebpixi:Fornecedor">
        <throw name="Fornecedor" faultName="erroEnvio" />
    </catch>
    </invoke>
</sequence>
</scope>

<!--
+-----+
| Envia Resp |
+-----+
-->

<!-- Envia Resp para o bpm principal -->
    <invoke name="Envia para o BPM" sap-extn:replyContainer="Req" sap-
extn:mode="async" inputVariable="Resp"
portType="http://marcos.nom/zebpixi:Catalogo_RespAbstractAsync"
operation="Catalogo_RespAbstractAsync"
partnerLink="Catalogo_RespAbstractAsync_send_7_Link" xmlns:sap-
extn="http://www.sap.com/xi/extensions" />
    </sequence>
</process>

```