



Nombre del Proyecto

Consultoría de Migración de Aplicaciones

Cliente

Servicio Nacional
de Aduanas



Fecha

11/03/16

Versión

1.1

Tipo de documento

Documento de análisis de brecha

34_DipsViajeros

1.Objetivo.....	4
2.Matriz de cumplimiento de documentación.....	5
3.Incompatibilidad detectadas.....	6
3.1.Librerías de BEA Systems.....	6
3.2.Clases Dependientes de BEA Systems.....	6
3.3.Frameworks / Tecnologías.....	8
4.Conclusiones.....	10
4.1.1.Cumplimiento de la documentación requerida.....	10
4.1.2.Windup.....	10
4.1.3.Complejidad de la Migración.....	10
5.Anexos.....	11
5.1.JBoss Windup.....	11
5.2.Informe de Levantamiento.....	11

CONTROL DE DIFUSIÓN

Nombre y Apellidos
Antonio Gabriel González Casado
Rafael Vázquez Ballesteros
Pedro Lisana
Gerardo Olmedo Nova
Eduardo Godoy Llanca

1. Objetivo

El objetivo presente documento es realizar el análisis de brecha para migrar el sistema 34_DipsViajeros de Weblogic 8.1 a la plataforma Jboss 6.4 EAP y de Java 1.4 a 1.7

2. Matriz de cumplimiento de documentación

Documento	Documentación existente *
Documento de análisis del sistema	✓
Documento de diseño del sistema	✓
Documento de instalación	X
Documento de integración	X
Documentación de Pruebas (planes, ejecuciones, etc)	✓
Manual de usuario	✓

Nomenclatura:

- ✓ : Existe
- X : No Existe

* Toda la documentación encontrada está especificada en el documento de Levantamiento. No obstante:

La ruta base de SVN donde se encuentra la documentación es:

- <http://172.40.100.50/svn/sistemas/workshop/DIPSVI>

3. Incompatibilidad detectadas

3.1. Librerías de BEA Systems

Se han identificado librerías pertenecientes a BEA Systems:

1. netui-pageflow.jar
2. netui-scoping.jar
3. netui-tags-databinding.jar
4. netui-tags-html.jar
5. netui-tags-template.jar
6. netui-util.jar
7. netui-wl.jar

3.2. Clases Dependientes de BEA Systems

Se hace una búsqueda de las clases dependientes de BEA Systems dentro del reporte Tattletale y se encuentran dependencias directas:

1. com.bea.jws.WebService
2. com.bea.wlw.runtime.core.bean.ContainerAdaptor
3. com.bea.wlw.runtime.core.bean.ContainerAdaptor\$Factory
4. com.bea.wlw.runtime.core.bean.SLSBContainerBean
5. com.bea.wlw.runtime.core.bean.SyncDispatcher
6. com.bea.wlw.runtime.core.bean.SyncDispatcherBean
7. com.bea.wlw.runtime.core.bean.SyncDispatcherHome
8. com.bea.wlw.runtime.core.container.Invocable
9. com.bea.wlw.runtime.core.debug.DebugContext
10. com.bea.wlw.runtime.core.dispatcher.AppManager
11. com.bea.wlw.runtime.core.dispatcher.DispBean
12. com.bea.wlw.runtime.core.dispatcher.DispControl
13. com.bea.wlw.runtime.core.dispatcher.DispFile
14. com.bea.wlw.runtime.core.dispatcher.DispMethod
15. com.bea.wlw.runtime.core.dispatcher.InvokeResult
16. com.bea.wlw.runtime.core.dispatcher.ServiceException
17. com.bea.wlw.runtime.core.dispatcher.ServiceHandleImpl
18. com.bea.wlw.runtime.core.request.Request
19. com.bea.wlw.runtime.core.request.Response
20. com.bea.xml.XmlObject

21. weblogic.ejb20.interfaces.BaseEJBLocalObjectIntf
22. weblogic.ejb20.interfaces.BaseEJBObjectIntf
23. weblogic.ejb20.interfaces.RemoteHome
24. weblogic.ejb20.interfaces.WLEnterpriseBean
25. weblogic.ejb20.internal.DummyContextHandler
26. weblogic.ejb20.internal.EJBContextHandler
27. weblogic.ejb20.internal.InvocationWrapper
28. weblogic.ejb20.internal.MethodDescriptor
29. weblogic.ejb20.internal.NeedRealContextHandlerError
30. weblogic.ejb20.internal.StatelessEJBHome
31. weblogic.ejb20.internal.StatelessEJBLocalHome
32. weblogic.ejb20.internal.StatelessEJBLocalObject
33. weblogic.ejb20.internal.StatelessEJBObject
34. weblogic.logging.NonCatalogLogger
35. weblogic.rmi.extensions.RemoteRuntimeException
36. weblogic.rmi.extensions.server.RemoteReference
37. weblogic.rmi.extensions.server.RuntimeMethodDescriptor
38. weblogic.rmi.extensions.server.Stub
39. weblogic.rmi.internal.BasicServerRef
40. weblogic.rmi.internal.MethodDescriptor
41. weblogic.rmi.internal.Skeleton
42. weblogic.rmi.internal.StubInfo
43. weblogic.rmi.internal.StubInfoIntf
44. weblogic.rmi.spi.InboundRequest
45. weblogic.rmi.spi.MsgInput
46. weblogic.rmi.spi.MsgOutput
47. weblogic.rmi.spi.OutboundResponse
48. weblogic.rmi.utils.Utilities
49. weblogic.security.service.ContextHandler
50. weblogic.servlet.internal.WebAppServletContext
51. weblogic.servlet.jsp.ByteWriter
52. weblogic.servlet.jsp.JspBase
53. weblogic.servlet.jsp.StaleChecker

54. weblogic.servlet.jsp.StaleIndicator

55. weblogic.utils.PlatformConstants

56. weblogic.utils.StackTraceUtils

Además, se encuentran clases dependientes de BEA Systems en librerías utilizadas por el sistema:

1. Schemas.jar

Se hace una búsqueda de las clases que el sistema provee de BEA Systems dentro del reporte Tattletale y se encuentran las siguientes:

1. com.bea.wlw.runtime.core.bean.SyncDispatcher_k1mrl8_EOImpl
2. com.bea.wlw.runtime.core.bean.SyncDispatcher_k1mrl8_EOImpl_814_WLStub
3. com.bea.wlw.runtime.core.bean.SyncDispatcher_k1mrl8_EOImpl_816_WLStub
4. com.bea.wlw.runtime.core.bean.SyncDispatcher_k1mrl8_EOImpl_WLSkel
5. com.bea.wlw.runtime.core.bean.SyncDispatcher_k1mrl8_HomeImpl
6. com.bea.wlw.runtime.core.bean.SyncDispatcher_k1mrl8_HomeImpl_814_WLStub
7. com.bea.wlw.runtime.core.bean.SyncDispatcher_k1mrl8_HomeImpl_816_WLStub
8. com.bea.wlw.runtime.core.bean.SyncDispatcher_k1mrl8_HomeImpl_WLSkel
9. com.bea.wlw.runtime.core.bean.SyncDispatcher_k1mrl8_Impl
10. com.bea.wlw.runtime.core.bean.SyncDispatcher_k1mrl8_Intf
11. com.bea.wlwgen.GenericStatelessSLSBContAdpt
12. com.bea.wlwgen.GenericStatelessSLSBContHome
13. com.bea.wlwgen.GenericStatelessSLSBContImpl
14. com.bea.wlwgen.GenericStatelessSLSBContIntf
15. com.bea.wlwgen.StatelessContainer_ly05hg_ELOImpl
16. com.bea.wlwgen.StatelessContainer_ly05hg_Impl
17. com.bea.wlwgen.StatelessContainer_ly05hg_Intf
18. com.bea.wlwgen.StatelessContainer_ly05hg_LocalHomeImpl

3.3. Frameworks / Tecnologías

Framework / Tecnología	Capa	Riesgo	Observaciones
NetUI Controls y Struts 1.1	Presentación	X	Riesgo de no ejecutarse en la arquitectura destino.
Vista en Flash mediante Flex	Presentación		Tecnología compatible con la arquitectura destino
Javax Mail	Negocio		Tecnología compatible con la arquitectura destino
iText 1.3.1	Negocio		Tecnología compatible con la

Framework / Tecnología	Capa	Riesgo	Observaciones
			arquitectura destino
HyperSQL DataBase	Negocio		Tecnología compatible con la arquitectura destino
Jasper Reports 1.3.4	Negocio		Tecnología compatible con la arquitectura destino
OPTA2000	Negocio		Tecnología compatible con la arquitectura destino
Apache POI 2.0-final	Negocio		Tecnología compatible con la arquitectura destino
Spring Frameworks 2.0.3	Negocio		Tecnología compatible con la arquitectura destino
Apache Avalon Framework 4.2.0	Negocio		Tecnología compatible con la arquitectura destino
Web Services vía Simple Object Access Protocol (SOAP)	Negocio	X	Riesgo de no ejecutarse en la arquitectura destino.
Apache™ Batik SVG Toolkit	Negocio		Tecnología compatible con la arquitectura destino
Java Database Connectivity (JDBC)	Persistencia		Tecnología compatible con la arquitectura destino
Flex Messaging	Integración		Tecnología compatible con la arquitectura destino

4. Conclusiones

4.1.1. Cumplimiento de la documentación requerida

Escala de 1-10 según del cumplimiento de documentación requerida y sugerida (1 poca documentación, 10 documentación referente a todos los grupos de documentos).

Ponderación: 7 (4 de 6 tipos de documentos)

4.1.2. Windup

Se ha analizado el EAR de la aplicación mediante la herramienta Windup y estima un esfuerzo de **126 puntos**:

Problema	Puntos	Observación
Call of JNDI lookup	6	Se deberán renombrar las llamadas a los JNDI con la nomenclatura aceptada por JBoss.
Dynamic instantiation of class	105	Se deberá habilitar "Dynamic Loaded Class" en JBoss.
Reference to javax.management.ObjectName	13	
Proprietary InitialContext initialization	2	
Web XML	0	
EJB XML	0	
JBoss Web XML	0	
JBoss Legacy EJB XML	0	
EJB Specification Extension	0	

4.1.3. Complejidad de la Migración

Escala de 1-5 de complejidad de migración (1 muy complejo, 5 poco complejo).

Ponderación: 3

Observaciones:

1. Las dependencias que requieren de clases BEA Systems deberán ser sustituidas por otras libres de dependencias BEA o eliminar estas dependencias BEA de ellas cuando sean librerías generadas por SNA.
2. El uso de controles netui es una fuerte vinculación con BEA.
3. Al usar clientes SOAP hay que considerar que, si los Servicios consumidos son migrados a JAX-WS, los clientes deberán ser regenerados para adaptarse a esta tecnología.
4. El uso de Flex y Flex Messaging (al ser una tecnología fuera de tendencia) puede hacer que la migración sea más compleja aunque no se considera una tecnología incompatible con la arquitectura.
5. Flex utiliza la API Java Message Service (JMS) para el envío de mensajes entre dos clientes, esto no se considera un riesgo ya que es compatible con la plataforma de destino.

5. Anexos

5.1. JBoss Windup

Informe Windup del proyecto disponible en el **34_DipsViajeros.zip** anexo a este informe.

5.2. Informe de Levantamiento

Informe del levantamiento de la situación actual, **34_DipsViajeros**, se encuentra en el documento **CMAA_levantamiento_34_DipsViajeros.pdf** anexo a este informe.