



Introduciéndose en el mundo de Java EE

Martín Pérez Mariñán

I Jornada de Tecnologías Java, Alicante, 12/05/2006

Sobre la compañía



IT ■ Deusto

La **segunda compañía** española
independiente de consultoría y
servicios de **IT**

Sobre el autor

- Martín Pérez Mariñán es Ingeniero de sistemas por la Universidad de A Coruña.
- Actualmente trabaja como desarrollador J2EE para IT Deusto dentro del Complejo Hospitalario Universitario Juan Canalejo de A Coruña desarrollando aplicaciones de gestión de costes sobre IBM WebSphere e IBM Informix.
- Es Sun Certified Java Programmer, Sun Certified Java Developer y Sun Certified Business Component Developer
- Martín es el creador de jLibrary (<http://jlibrary.sourceforge.net>), una herramienta de gestión de documentación Open Source desarrollada sobre Eclipse RCP y el Apache Jackrabbit (JSR-170). Este proyecto le ha supuesto nominaciones para eventos internacionales como EclipseCon 2006 o los JAX Innovation Awards.

Sobre esta presentación

- La universidad es una excelente herramienta de modelado de mentes, pero en cierto modo te esconde del mundo exterior. Todos nos hemos preguntado al estar ahí
 - ¿Qué voy a hacer al salir?
 - ¿Qué hacen las empresas fuera?
 - ¿Qué perfiles piden?
 - ¿Cómo se hace un proyecto del mundo real?
 - ¿Cuánto pagan?
 - ¿Vale la pena escoger esta tecnología?
- Ninguna de estas preguntas tiene una respuesta fija o directa. Esta presentación pretende ser una pequeña guía para que tengáis algo más claras las posibles respuestas cuando hablamos de Java EE (J2EE).

Índice



1. Las posibilidades
2. Estado de salud de J2EE
3. Lo último: Java EE 5
4. Modelos y tendencias de desarrollo
5. Conclusiones

Las posibilidades



Las posibilidades

- Los entornos de trabajo en los días actuales están bastante claros:
 - J2EE
 - Microsoft .NET
- A menor escala pero muy común en sistemas Open Source:
 - LAMP (Linux + Apache + MySQL + Perl o PHP o Python)
- Lo más de moda:
 - Ruby
 - Ruby on Rails
- Sistemas de gestión:
 - Siebel, SAP, ...
- Otros frameworks propios de compañías y lenguajes menores

Las posibilidades: Java EE

- Java EE o J2EE o JEE
- Es probablemente la solución más completa en cuanto a arquitectura, escalabilidad, patrones, metodologías, etc.
- Enorme soporte de fabricantes: Sun, IBM, Oracle, BEA, RedHat, SAP, Siebel, Compuware, Sybase, ...
- Gobernada por un sistema de especificaciones y un organismo coordinador, el JCP.
- Las especificaciones van evolucionando. Diferentes empresas participan en su evolución bajo un sistema democrático de votaciones. No hay vetos.

Las posibilidades: Java EE

[file:///C:/Users/Deusto/Documents/Proyectos/Deusto/2018/06/20096/Charlas/Overview-DeeArchitecture.pdf](#)

Estado de Salud de Java EE



Las posibilidades: Java EE

- El estado de salud en España es excelente
- No sólo ha copado las grandes empresas y administraciones públicas sino también poco a poco las medianas empresas
 - Influencia de universidades
 - Mayor rendimiento en los contenedores
 - Muchísimas más posibilidades
 - Despliegues desde 0€
 - Rendimiento cada vez mejor en el escritorio
- Bendecido por la administración pública
- Hace tres o cuatro años se premonizaba el fin con .NET...
 - ... sin embargo el uso de J2EE sigue y sigue aumentando

Estado de Salud de Java EE

- Una de las medidas más subjetivas, inválidas desde el punto teórico, pero muy efectivas desde el punto de vista práctico es ver el impacto en buscadores de empleo.
- Una búsqueda muy rápida sobre Infojobs.net (1 mes de antigüedad) refleja lo siguiente:
 - Java: 1622 ofertas
 - J2EE: 781 ofertas
 - Microsoft: 1167 ofertas
 - ASP: 1012 ofertas
 - JSP: 332 ofertas
- <http://www.itjobswatch.co.uk/> Proporciona cifras analíticas más interesantes sobre la salud de tecnologías en UK.

Estado de salud de Java EE

Results 1 - 20 of 143

Description	Rank 3 Months to 6 May 2006	Rank Change on Same Period Last Year	Average Salary 3 Months to 6 May 2006	Average Salary % Change on Same Period Last Year	Matching Job Ads Last 3 Months (% of All Permanent Job Ads Sampled)
JRockit	877	☉ -	£105,000	-	7 (0.003 %)
E4JMS	883	☉ -	£90,000	-	1 (0.000 %)
JINI	878	▼ -85	£80,000	+43.34 %	6 (0.003 %)
JavaSpaces	879	▼ -56	£75,000	+8.55 %	5 (0.002 %)
SAP MI	858	▼ -50	£69,000	+39.10 %	26 (0.012 %)
XML Messaging	630	▲ +80	£67,042	+33.22 %	271 (0.13 %)
SAML	855	▼ -46	£62,500	+12.10 %	29 (0.014 %)
JFC	879	▼ -64	£62,500	+78.57 %	5 (0.002 %)
JTS	872	▼ -51	£60,750	-26.04 %	12 (0.006 %)
FpML	637	▲ +170	£60,702	+6.09 %	262 (0.12 %)
JMI	857	▼ -26	£60,278	-59.81 %	27 (0.013 %)
PicoContainer	871	☉ -	£58,409	-	13 (0.006 %)
SAP XI	558	▲ +56	£58,186	+5.27 %	377 (0.18 %)
AspectJ	875	☉ -	£56,111	-	9 (0.004 %)
FIXML	880	☉ -	£55,625	-	4 (0.002 %)
Middleware	151	▼ -15	£54,998	+6.77 %	2499 (1.19 %)
JAAS	839	▼ -63	£53,609	+16.56 %	45 (0.021 %)
JRules	735	▲ +69	£52,980	+54.78 %	149 (0.071 %)
RMI	580	▲ +103	£51,120	+6.15 %	342 (0.16 %)
JNI	833	▼ -88	£50,539	+10.95 %	51 (0.024 %)

Results 21 - 40 of 143

Description	Rank 3 Months to 6 May 2006	Rank Change on Same Period Last Year	Average Salary 3 Months to 6 May 2006	Average Salary % Change on Same Period Last Year	Matching Job Ads Last 3 Months (% of All Permanent Job Ads Sampled)
JMS	216	▲ +18	£50,108	-3.14 %	1721 (0.82 %)
XMPP	880	▼ -54	£50,000	-	4 (0.002 %)
XMLBeans	883	▼ -51	£50,000	0	1 (0.000 %)
AOM	883	● -	£50,000	-	1 (0.000 %)
SAP Basis	349	▼ -17	£49,809	+6.46 %	908 (0.43 %)
J2SE	376	▲ +158	£49,632	+11.37 %	802 (0.38 %)
STL	280	▼ -25	£49,310	-12.79 %	1230 (0.59 %)
ADSI	854	▼ -80	£48,214	-13.19 %	30 (0.014 %)
Jabber	878	● -	£47,929	-	6 (0.003 %)
Swing	260	▼ -57	£47,790	-4.68 %	1358 (0.65 %)
CORBA	399	▼ -97	£47,405	-11.44 %	709 (0.34 %)
WinForms	261	▲ +141	£47,380	+7.64 %	1352 (0.64 %)
XSD	618	▲ +41	£47,222	+10.83 %	292 (0.14 %)
J2EE	20	▼ -1	£47,169	-0.77 %	14041 (6.69 %)
Spring Framework	408	▲ +253	£47,119	+16.85 %	685 (0.33 %)
WSDL	612	▲ +68	£47,049	+7.96 %	300 (0.14 %)
TopLink	853	▼ -51	£46,700	+23.57 %	31 (0.015 %)
POSIX	801	▼ -15	£46,522	+15.96 %	83 (0.040 %)
BPEL	793	▲ +32	£46,054	-3.04 %	91 (0.043 %)
LDAP	345	▲ +68	£45,334	-4.65 %	920 (0.44 %)

Description	Rank 3 Months to 6 May 2006	Rank Change on Same Period Last Year	Average Salary 3 Months to 6 May 2006	Average Salary % Change on Same Period Last Year	Matching Job Ads Last 3 Months (% of All Permanent Job Ads Sampled)
JNDI	797	▼ -88	£45,079	+4.96 %	87 (0.041 %)
Web Dynpro	855	▼ -60	£45,063	-10.19 %	29 (0.014 %)
 JTA	883	● -	£45,000	-	1 (0.000 %)
JTA	835	▼ -55	£44,988	-26.02 %	49 (0.023 %)
SAP BSP	872	▼ -70	£44,833	-7.47 %	12 (0.006 %)
MHP	806	▼ -8	£44,699	+13.20 %	78 (0.037 %)
MAPI	818	▼ -6	£44,487	+28.70 %	66 (0.031 %)
XForms	874	▼ -44	£44,444	+22.60 %	10 (0.005 %)
 SOAP	156	▲ +23	£44,191	+0.84 %	2445 (1.16 %)
EJB	159	▼ -47	£44,015	-3.81 %	2395 (1.14 %)
CGI	505	↔ 0	£43,977	+19.33 %	467 (0.22 %)
 WDK	794	▲ +6	£43,819	-1.55 %	90 (0.043 %)
 Hibernate	274	▲ +218	£43,764	+7.81 %	1287 (0.61 %)
XML Schema	670	▼ -136	£43,674	-1.89 %	221 (0.11 %)
WSH	844	▼ -61	£43,617	+23.69 %	40 (0.019 %)
 JAX-RPC	822	▲ +9	£43,526	-56.47 %	62 (0.030 %)
WapTV	881	▼ -53	£43,333	-3.70 %	3 (0.001 %)
XQuery	873	▼ -63	£43,045	+12.90 %	11 (0.005 %)
Lucene	878	▼ -71	£43,000	+17.91 %	6 (0.003 %)
SAP ALE	768	▼ -126	£42,896	-4.64 %	116 (0.055 %)

Results 61 - 80 of 143

Description	Rank 3 Months to 6 May 2006	Rank Change on Same Period Last Year	Average Salary 3 Months to 6 May 2006	Average Salary % Change on Same Period Last Year	Matching Job Ads Last 3 Months (% of All Permanent Job Ads Sampled)
ATL	368	▼ -15	£42,841	-11.85 %	828 (0.39 %)
JMX	752	▼ -25	£42,772	-21.31 %	132 (0.063 %)
Struts	152	▼ -9	£42,488	-1.57 %	2483 (1.18 %)
Xerces	858	▼ -33	£42,367	-2.23 %	26 (0.012 %)
JCE	861	▼ -41	£42,341	-0.16 %	23 (0.011 %)
Xalan	881	▼ -60	£42,333	-5.92 %	3 (0.001 %)
DCOM	470	▼ -84	£42,064	-12.56 %	527 (0.25 %)
SAX	831	▼ -61	£41,900	+10.26 %	53 (0.025 %)
COM+	314	▼ -62	£41,732	-3.45 %	1033 (0.49 %)
JAXB	873	▼ -45	£41,700	+2.12 %	11 (0.005 %)
UDDI	839	▼ -117	£41,549	+0.05 %	45 (0.021 %)
MIDP	822	▼ -36	£41,517	+25.46 %	62 (0.030 %)
BAPI	794	▼ -112	£41,462	-8.81 %	90 (0.043 %)
XML-RPC	825	▼ -35	£41,459	+16.88 %	59 (0.028 %)
JDO	833	▼ -27	£41,395	+5.22 %	51 (0.024 %)
SAP Smart Forms	827	▼ -52	£41,389	+9.43 %	57 (0.027 %)
SyncML	815	▼ -7	£41,282	+10.14 %	69 (0.033 %)
 JSF	589	▲ +208	£41,146	+5.39 %	328 (0.16 %)
Servlets	140	▼ -12	£41,108	-6.86 %	2715 (1.29 %)
 WebServices	78	▲ +18	£40,975	-3.75 %	4865 (2.32 %)

Results 81 - 100 of 143

Description	Rank 3 Months to 6 May 2006	Rank Change on Same Period Last Year	Average Salary 3 Months to 6 May 2006	Average Salary % Change on Same Period Last Year	Matching Job Ads Last 3 Months (% of All Permanent Job Ads Sampled)
 MFC	197	 +28	£40,970	+4.96 %	1871 (0.89 %)
J2ME	526	 -16	£40,935	+2.19 %	426 (0.20 %)
JDBC	206	 -23	£40,935	-9.88 %	1784 (0.85 %)
COM	512	 -269	£40,820	-2.50 %	457 (0.22 %)
JSP	70	 -4	£40,425	-5.79 %	5437 (2.59 %)
XML	17	 -1	£40,142	-1.29 %	19739 (9.40 %)
 XAML	883	 -	£40,000	-	1 (0.000 %)
JavaBeans	735	 -74	£39,963	+3.25 %	149 (0.071 %)
DOM	693	 -115	£39,123	-16.22 %	195 (0.093 %)
 VML	883	 -	£39,000	-	1 (0.000 %)
EDI	620	 -86	£38,801	+5.31 %	286 (0.14 %)
 .NET	10	 +3	£38,797	-3.69 %	28314 (13.48 %)
OLE DB	858	 -77	£38,553	+18.77 %	26 (0.012 %)
 XSLT	122	 +11	£38,328	-6.38 %	3180 (1.51 %)
 WAP	458	 +68	£38,041	-9.42 %	546 (0.26 %)
VXML	856	 -40	£37,911	+2.80 %	28 (0.013 %)
OCX	863	 -58	£37,794	+27.96 %	21 (0.010 %)
 UIQ	800	 +2	£37,644	+1.28 %	84 (0.040 %)
OLE	864	 -101	£37,589	-2.53 %	20 (0.010 %)
XSL	246	 -41	£37,414	-6.69 %	1464 (0.70 %)

Results 1 - 20 of 143

Description	Rank 3 Months to 6 May 2006 	Rank Change on Same Period Last Year	Average Salary 3 Months to 6 May 2006	Average Salary % Change on Same Period Last Year	Matching Job Ads Last 3 Months (% of All Permanent Job Ads Sampled)
 .NET	10	 +3	£38,797	-3.69 %	28314 (13.48 %)
XML	17	 -1	£40,142	-1.29 %	19739 (9.40 %)
J2EE	20	 -1	£47,169	-0.77 %	14041 (6.69 %)
 HTML	21	 +4	£33,903	-0.29 %	12769 (6.08 %)
 ASP.NET	23	 +19	£35,454	-0.83 %	12079 (5.75 %)
JSP	70	 -4	£40,425	-5.79 %	5437 (2.59 %)
 CSS	73	 +57	£31,745	-1.95 %	5289 (2.52 %)
 WebServices	78	 +18	£40,975	-3.75 %	4865 (2.32 %)
ASP	80	 -17	£33,149	-1.03 %	4775 (2.27 %)
 XSLT	122	 +11	£38,328	-6.38 %	3180 (1.51 %)
Servlets	140	 -12	£41,108	-6.86 %	2715 (1.29 %)
Middleware	151	 -15	£54,998	+6.77 %	2499 (1.19 %)
Struts	152	 -9	£42,488	-1.57 %	2483 (1.18 %)
 SOAP	156	 +23	£44,191	+0.84 %	2445 (1.16 %)
EJB	159	 -47	£44,015	-3.81 %	2395 (1.14 %)
 MFC	197	 +28	£40,970	+4.96 %	1871 (0.89 %)
JDBC	206	 -23	£40,935	-9.88 %	1784 (0.85 %)
 XHTML	208	 +126	£31,292	-7.75 %	1770 (0.84 %)
 JMS	216	 +18	£50,108	-3.14 %	1721 (0.82 %)
XSL	246	 -41	£37,414	-6.69 %	1464 (0.70 %)

Estado de Salud de Java EE

- Algunas conclusiones:
 - En UK se cobra más :-)
 - Las tecnologías concretas se pagan mejor que tecnologías más comunes.
 - En general el mundo Java y J2EE está mucho mejor pagado que el mundo .NET
 - Sin embargo, el entorno Microsoft presenta muchas más ofertas.
 - El mercado de XML y Servicios Web también es interesante.
 - Las tecnologías web (asp, jsp) parecen estar peor pagadas que las de escritorio (swing).

Lo último: Java EE 5.0

Java EE 5.0: Historia

<p>Java EE 5 Mayo, 2006</p>	<p>La mayor revisión desde J2EE 1.3. Cambio de nombre. Cambio completo en el modelo de EJBs. Más fáciles, nuevo modelo de persistencia. Introducción de anotaciones en todos los componentes, inyección de dependencias y conceptos de AOP. A parece Java Server Faces. Streaming XML.</p> <p>JSP 2.1, Servlets 2.5, EJB 3.0, JCA 1.5, JAX-WS 2.0, JAX-RPC 1.1, JAXB 2.0, StAX, Java Server Faces 1.0</p>
<p>J2EE 1.4 Abril, 2004</p>	<p>Revisión menor. Mejoras menores en EJB como timers, mejoras en EJB-QL, etc. Soporte de servicios web y endpoints. Soporte del WS-I Basic Profile. Nuevo lenguaje de expresiones para JSP. Extensiones de gestión JMX.</p> <p>Es el estándar actual en las últimas versiones de servidores de aplicaciones: jboss 4, WebSphere 6.0, WebLogic 9.1, Apache Geronimo 1.0, ...</p> <p>JSP 2.0, Servlets 2.4, EJB 2.1, JCA 1.5, JMS 1.1, JDBC 3.0, Javamail 1.3, JMX 1.0, JAXP 1.2, JAXR 1.0, JAX-RPC 1.1, SAAJ 1.1</p>
<p>J2EE 1.3 Octubre, 2001</p>	<p>Revisión importante. Renovado el modelo de componentes de EJB. CMP 2.0. Aumento enorme de rendimiento de EJBs. Interfaces locales. Beans de mensajería. Simplificación de conceptos. Filtros de Servlet, mejoras en el soporte de custom tags en JSP. Soporte básico de XML.</p> <p>Es todavía una especificación muy utilizada y tuvo un gran éxito.</p> <p>JSP 1.2, Servlets 2.3, EJB 2.0, JCA 1.0, JMS 1.0, JDBC 3.0, Javamail 1.2, JAXP, ...</p>
<p>J2EE 1.2 Diciembre, 1999</p>	<p>Primera versión seria. Muchos problemas de rendimiento y persistencia con los EJB. Prácticamente sólo usables los componentes web.</p> <p>JSP 1.1, Servlets 2.2, EJB 1.1, JCA 1.0, JDBC 3.0, Javamail 1.1, ...</p>

Java EE 5.0

- Java EE 5.0 es la última versión de lo que siempre se ha conocido como J2EE.
- El principal objetivo es la facilidad de desarrollo:
 - Siempre ha sido complicado hacer aplicaciones en J2EE. Era necesaria una solución.
 - Menos código que escribir.
 - Valores por defecto.
 - Uso extensivo de anotaciones en todos los componentes.
 - Inyección de dependencias
- La principal novedad es EJB 3.0
 - Adopción del modelo de POJOs.
 - Nueva API de persistencia.
 - Utilizable fuera de servidores de aplicaciones.

Java EE 5.0

- Construcción más sencilla de aplicaciones web
 - Soporte de Java Server Faces.
 - Mejor integración con herramientas de desarrollo.
 - Mejoras en JSTL.
 - Soporte de Web 2.0 y AJAX
- Soporte mejorado de servicios web
 - Despliegue más sencillo
 - Añade las últimas APIs
- ¿Puede empezar?
 - Todavía no hay servidores que soporten esta especificación ya que se aprobó hace unos días.
 - Hay versiones beta de varios servidores y servidores que ofrecen parte de las funcionalidades.

Java EE 5.0

■ Tabla de servidores:

<i>Servidores de aplicaciones Open Source</i>	
<i>Servidor de aplicaciones</i>	<i>Versión de J2EE soportada</i>
Apache Geronimo 1.0	J2EE 1.4
jboss 4.0.4	J2EE 1.4, disponible plug-in EJB 3.0
JonAS 4.7.1	J2EE 1.4, disponible plug-in EJB 3.0
Glassfish Milestone 6	Java EE 5
<i>Servidores de aplicaciones no Open Source</i>	
<i>Servidor de aplicaciones</i>	<i>Versión de J2EE soportada</i>
Oracle Application Server 10g	J2EE 1.4, disponible plug-in EJB 3.0
IBM WebSphere	J2EE 1.4
WebSphere Application Server Community Edition 1.0 (basado en Apache Geronimo)	J2EE 1.4
BEA WebLogic 9.1	J2EE 1.4

Java EE 5.0: Java Server Faces

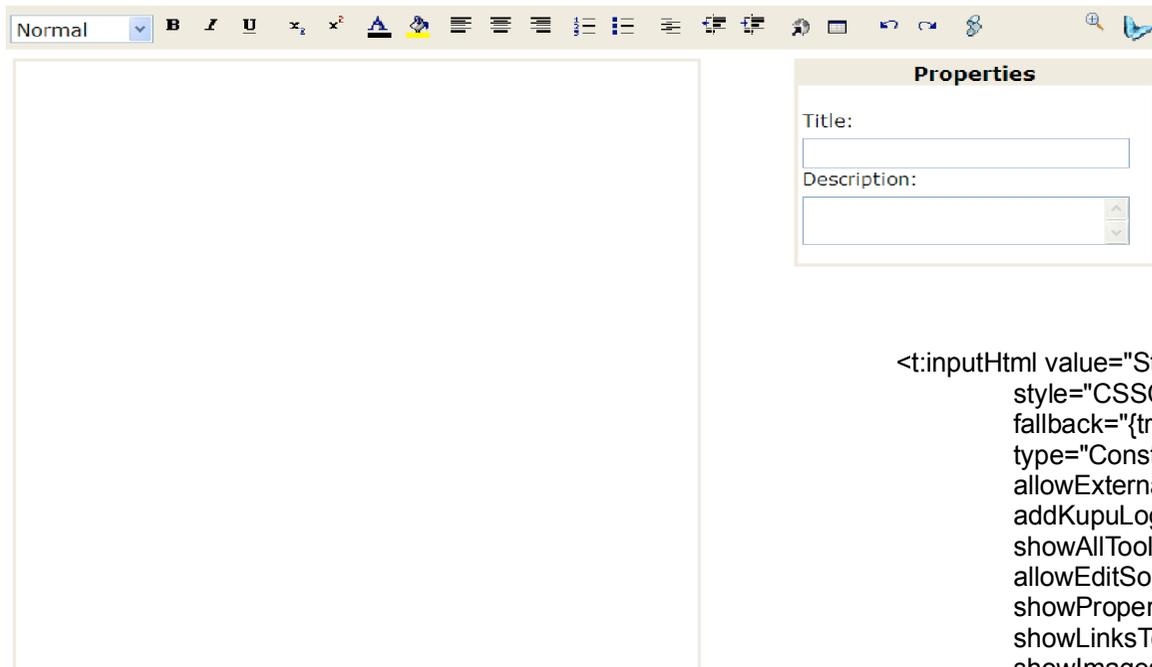
- Especificación que define un framework web para la creación de aplicaciones.
 - Gestión de componentes gráficos.
 - Manejo de eventos en el servidor.
 - Validación de entradas.
 - Asociación entre el modelo de datos y el interfaz de usuario.
 - Gestión del flujo entre páginas.
 - Sistema de renderización independiente de la web.
- Alternativa a Struts y otros frameworks.
- El colíder de la especificación es Craig McClanahan, autor de Struts.
- Ventajas frente a otros frameworks
 - Estándar
 - Gran soporte por parte de la industria y fabricantes. Es difícil que te den soporte de Spring o Struts pero te lo darán de JSF.
 - Muchas herramientas de diseño gráfico y desarrollo disponibles.

Java EE 5.0: Java Server Faces

- IDEs: MyEclipse, Sun Java Studio, Borland Jbuilder, Oracle Jdeveloper, IBM Rational Application Developer, ...
- Respuesta al desarrollo rápido de aplicaciones en ASP .NET
- Shale, la evolución de Struts, se fundamente sobre JSF
- La tendencia ahora mismo es crear componentes JSF con soporte de AJAX.
- El problema de la especificación es que define sólo los componentes más básicos. Lo más complejo es una tabla.
- Se hace necesario buscar frameworks más completos. Ej: MyFaces, Oracle ADF Faces, ...
- Surge también un mercado de componentes. Ej: AjaxFaces.

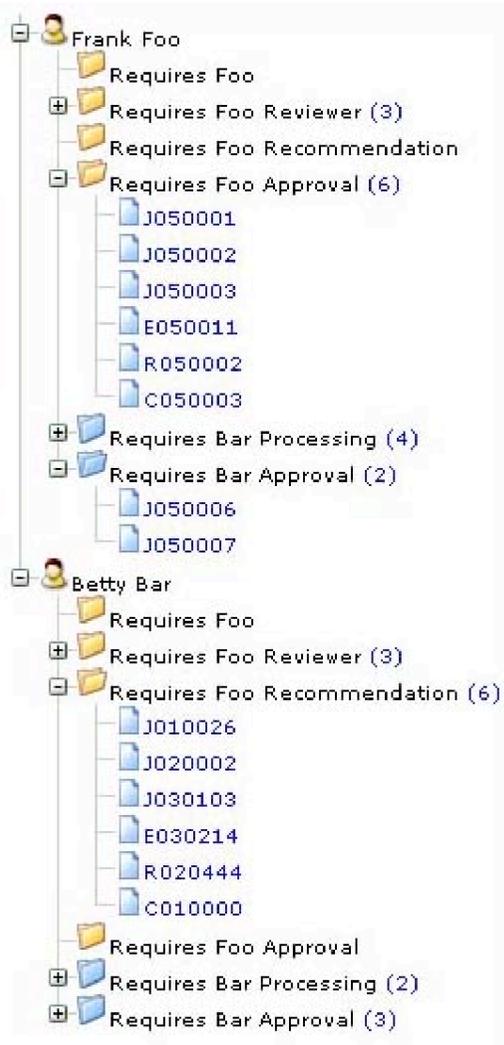
Java EE 5.0: Java Server Faces

Editor de texto en MyFaces



```
<t:inputHtml value="String"
  style="CSSClass"
  fallback="{true|false}"
  type="Constant"
  allowExternalLinks="{true|false}"
  addKupuLogo="{true|false}"
  showAllToolBoxes="{true|false}"
  allowEditSource="{true|false}"
  showPropertiesToolBox="{true|false}"
  showLinksToolBox="{true|false}"
  showImagesToolBox="{true|false}"
  showTablesToolBox="{true|false}"
  showDebugToolBox="{true|false}"
  showCleanupExpressionsToolBox="{true|false}"/>
```

Java EE 5.0: Java Server Faces

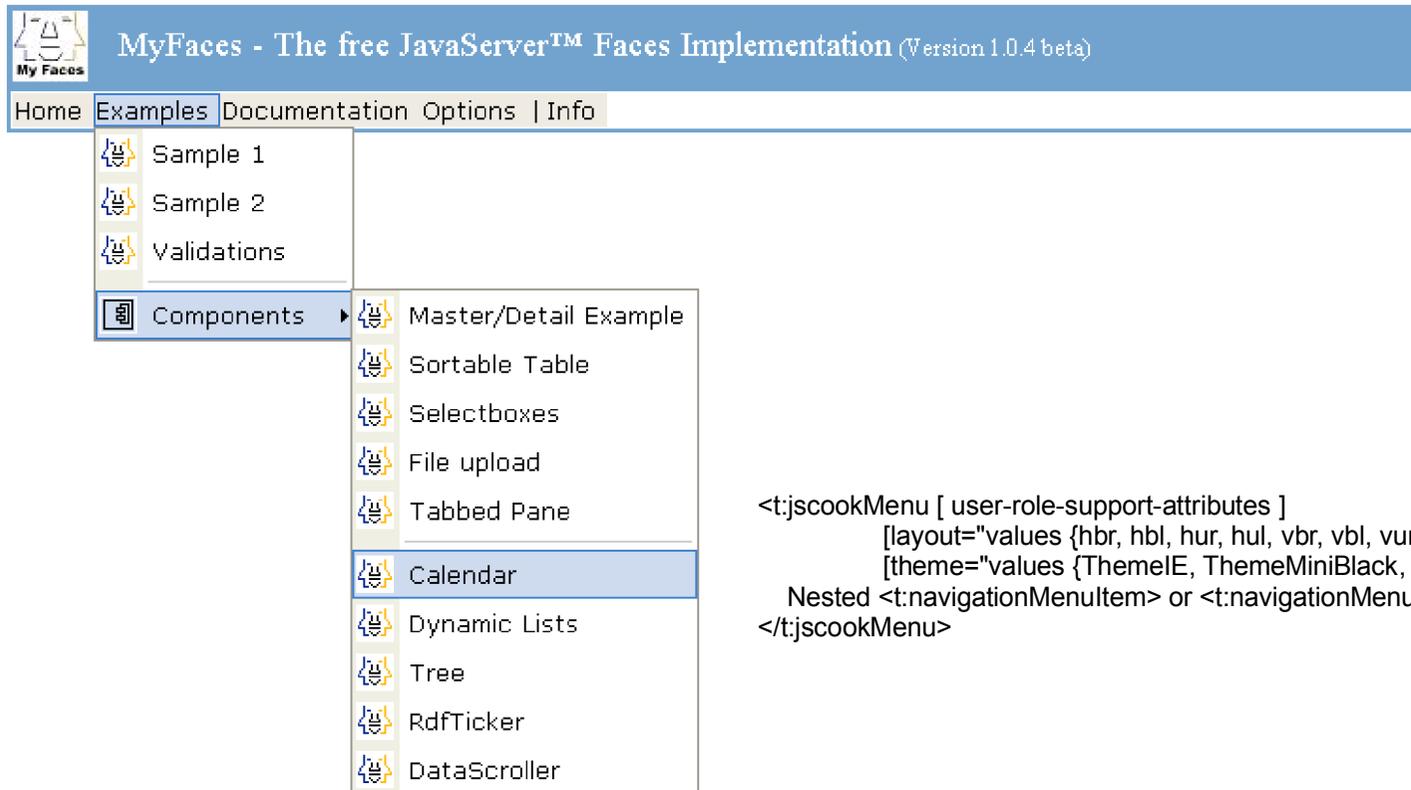


Árbol de MyFaces

```
<t:tree2 value="TreeModel"
         var="String"
         varNodeToggle="String">
  ... (anyComponent) ...
</t:tree2>
```

Java EE 5.0: Java Server Faces

Menús en MyFaces



MyFaces - The free JavaServer™ Faces Implementation (Version 1.0.4 beta)

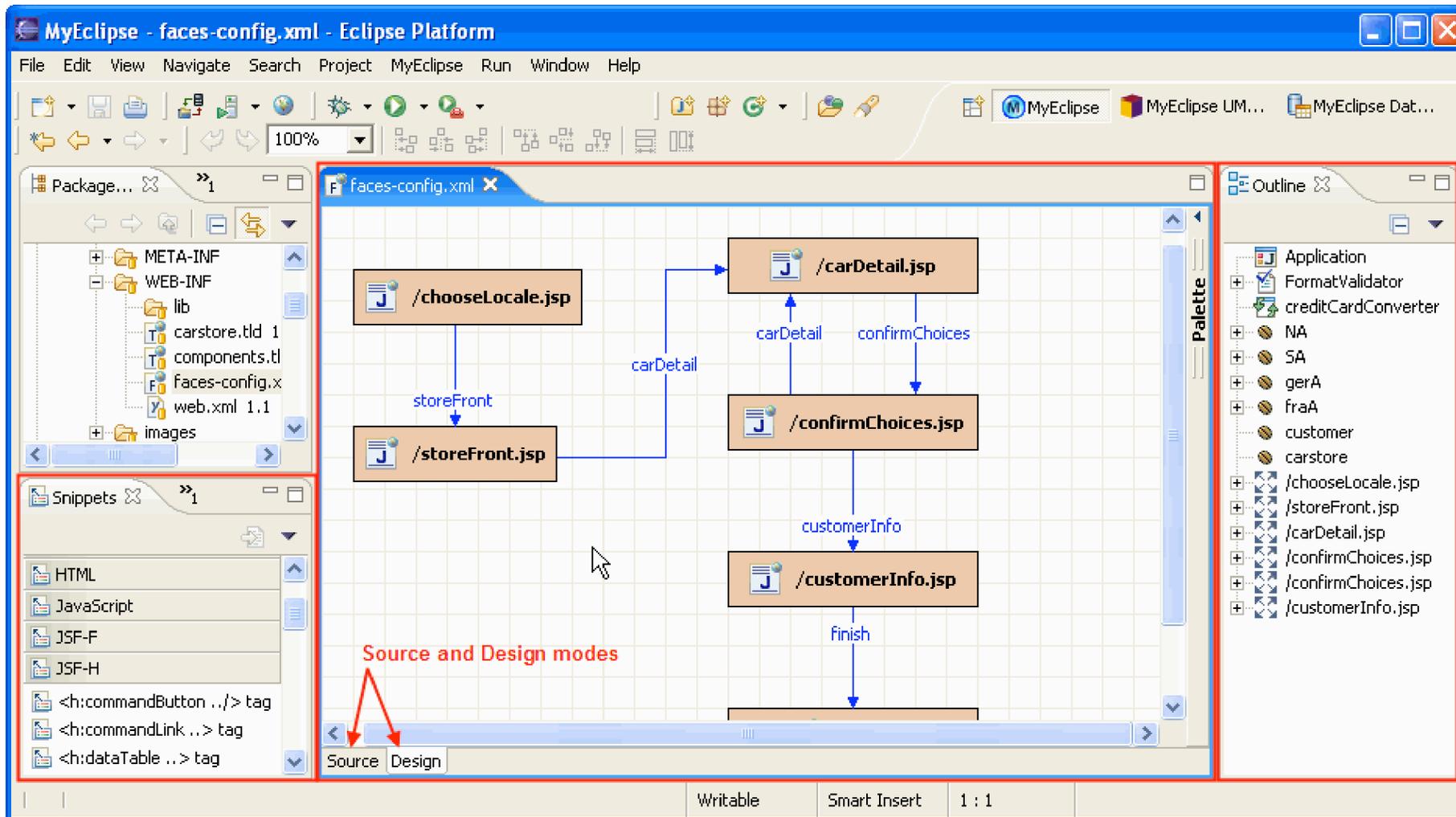
Home Examples Documentation Options | Info

- Sample 1
- Sample 2
- Validations
- Components
 - Master/Detail Example
 - Sortable Table
 - Selectboxes
 - File upload
 - Tabbed Pane
 - Calendar**
 - Dynamic Lists
 - Tree
 - RdfTicker
 - DataScroller

```
<t:jscookMenu [ user-role-support-attributes ]  
    [layout="values {hbr, hbl, hur, hul, vbr, vbl, vur, vul}"]  
    [theme="values {ThemeIE, ThemeMiniBlack, ThemeOffice, ThemePanel}"]  
    Nested <t:navigationMenuItem> or <t:navigationMenuItems> tags (menu items)  
</t:jscookMenu>
```

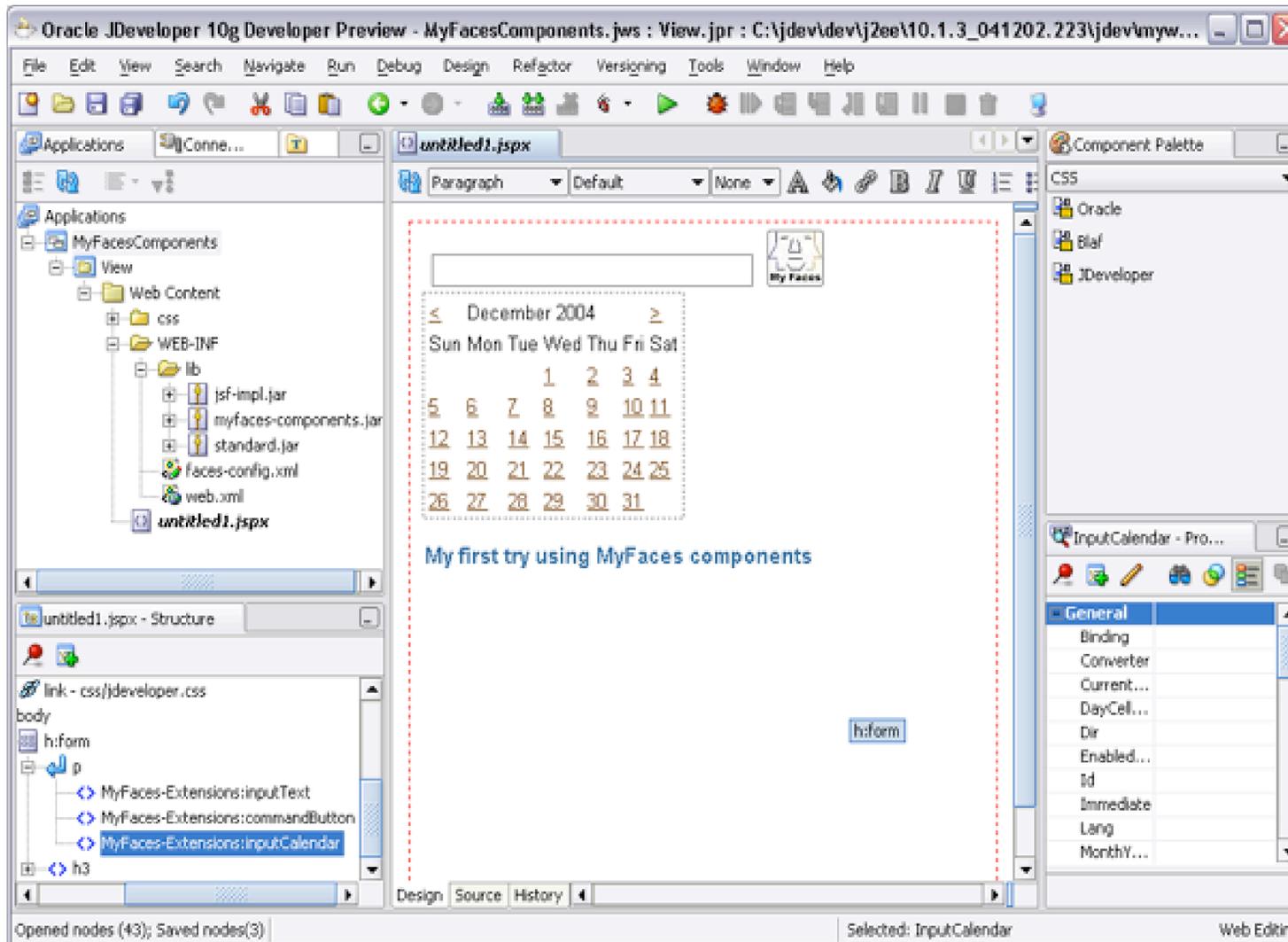
Java EE 5.0: Java Server Faces

MyEclipse y JSF



Java EE 5.0: Java Server Faces

Oracle JDeveloper y JSF



Java EE 5.0: EJB 3.0

- Nueva API de persistencia
 - Se separa la persistencia de los componentes de negocio
 - Modelo de persistencia basado en POJOs y POJIs
 - CMP Simplificada
 - Es utilizable tanto en J2SE como en J2EE
 - Influencias de Hibernate, TopLink y JDO
- Anotaciones
 - Simplificación en el desarrollo
 - Mejor integración con herramientas de desarrollo
 - Influencias de Xdoclet

Java EE 5.0: EJB 3.0

@Entity

```
public class Empleado {  
    private Cubiculo cubiculoAsignado;
```

@OneToOne

```
    public Cubiculo getCubiculoAsignado() {  
        return cubiculoAsignado;  
    }  
    public setCubiculoAsignado(Cubiculo cubiculo) {  
        this.cubiculoAsignado = cubiculo;  
    }  
}
```

*Ejemplo de
anotaciones:*

Relaciones CMP

Java EE 5.0: EJB 3.0

```
@Entity
```

```
@Table (name="empleados")
```

```
@Inheritance (strategy=JOINED)
```

```
public abstract class Empleado {
```

```
    @Id protected Integer idEmpleado;
```

```
    @Version protected Integer version;
```

```
    @ManyToOne protected Direccion direccion;
```

```
}
```

```
    @Entity
```

```
    @Table(name="empleados_fijos")
```

```
    @DiscriminatorValue("FI")
```

```
    @PrimaryKeyJoinColumn(name="FI_idemp")
```

```
public class EmpleadoFijo extends Empleado {
```

```
    protected Integer salario;
```

```
    public Integer getSalario() { return salario; }
```

```
}
```

Anotaciones

Mejora en el modelo de persistencia: Herencia

Java EE 5.0: EJB 3.0

Uso del Entity Manager

```
@Stateless public class FacturasBean {  
  
    @PersistenceContext EntityManager manager;  
  
    public void introducirFactura(int idClient, Factura factura) {  
  
        Client cliente = manager.find(Cliente.class, idCliente);  
        cliente.getFacturas().add(factura);  
        factura.setCliente(cliente);  
    }  
}
```

Muy similar a las Session de Hibernate

Java EE 5.0

- Inyección de dependencias
 - También conocido como IoC (Inversión de control)
 - No es el usuario el que busca los recursos.
 - El contenedor se los inyecta automáticamente.
 - Ya no es necesario realizar las farragosas llamadas JNDI.
 - El código se simplifica y es menos propenso a errores.
 - Influencias de contenedores lightweight como Spring.

Java EE 5.0: EJB 3.0

```
<ejb-local-ref>  
  <ejb-ref-name>ejb/HolaAlicante</ejb-ref-name>  
  <local>org.jtech.HolaAlicanteBean</local>  
</ejb-local-ref>
```

Antes

```
Context context = new InitialContext();  
HolaAlicante bean (HolaAlicante)  
context.lookup("java:comp/env/ejb/HolaAlicanteBean");
```

Ahora.

```
@EJB(name="HolaAlicante")  
private HolaAlicante myEJB;
```

La dependencia se inyecta. No hay que buscar el EJB.

■ Intercepción (Interceptors)

- Permiten al desarrollador un mayor control sobre el flujo de sus programas.
- Se llaman antes y después de la invocación a un método.
- Se aplican a todos los métodos de negocio de un EJB.
- Permiten modificar valores de entrada a métodos.
- Permiten modificar valores de retorno.
- Gestionar las excepciones.
- Interrumpir llamadas completamente.
- Realizar análisis y optimización de métodos.
- Influencias de AOP.
- etc.

```
@Stateless
@WebService
@Interceptors( { Interceptor2.class } )
public class HelloWorldBean {
```

```
    @Interceptors( { Interceptor1.class } )
    @WebMethod
    public String sayHello() {
        return "Hello";
    }
```

```
    @WebMethod
    public String sayHi() {
        return "Hi";
    }
```

```
    @AroundInvoke
    public Object log(InvocationContext invocationContext) throws Exception {
        System.err.println(invocationContext.getMethod().getName() + " called from interceptor 3");
        return invocationContext.proceed();
    }
```

```
}
```

Uso de los interceptores. Uno a nivel de EJB y otro a nivel de método

```
public class Interceptor1 {  
  
    @AroundInvoke  
    public Object log(InvocationContext invocationContext) throws Exception {  
        System.err.println(invocationContext.getMethod().getName() + " called from  
            interceptor 1");  
        return invocationContext.proceed();  
    }  
}
```

Declaración de los interceptores

```
public class Interceptor2 {  
  
    @AroundInvoke  
    public Object log(InvocationContext invocationContext) throws Exception {  
        System.err.println(invocationContext.getMethod().getName() + " called from  
            interceptor 2");  
        return invocationContext.proceed();  
    }  
}
```

Java EE 5.0

■ Simplificación

- API de persistencia (ya visto)
- Metadatos. (ya visto)
- Inyección de dependencias (ya visto)
- Eliminación de las interfaces de componente e interfaces home. Todo se autogenera a partir de las anotaciones.
- Valores por defecto en la configuración.
- Reducción del uso de excepciones chequeadas.

Modelos y tendencias de desarrollo



Java EE 5.0

- Modelos ágiles de desarrollo con Java EE
 - Antes complicado. Infraestructura compleja. Difícil testabilidad.
 - Ahora se abren nuevas perspectivas:
 - APIs más sencillas
 - Contenedores ligeros
 - Herramientas de pruebas más evolucionadas: bases de datos, interfaces de usuario, componentes web, etc.
- Aceptación del cambio
 - Refactorización
 - El cliente puede modificar los requisitos, renegociación
- Buscar la simplicidad
- TDD: Test Driven Development

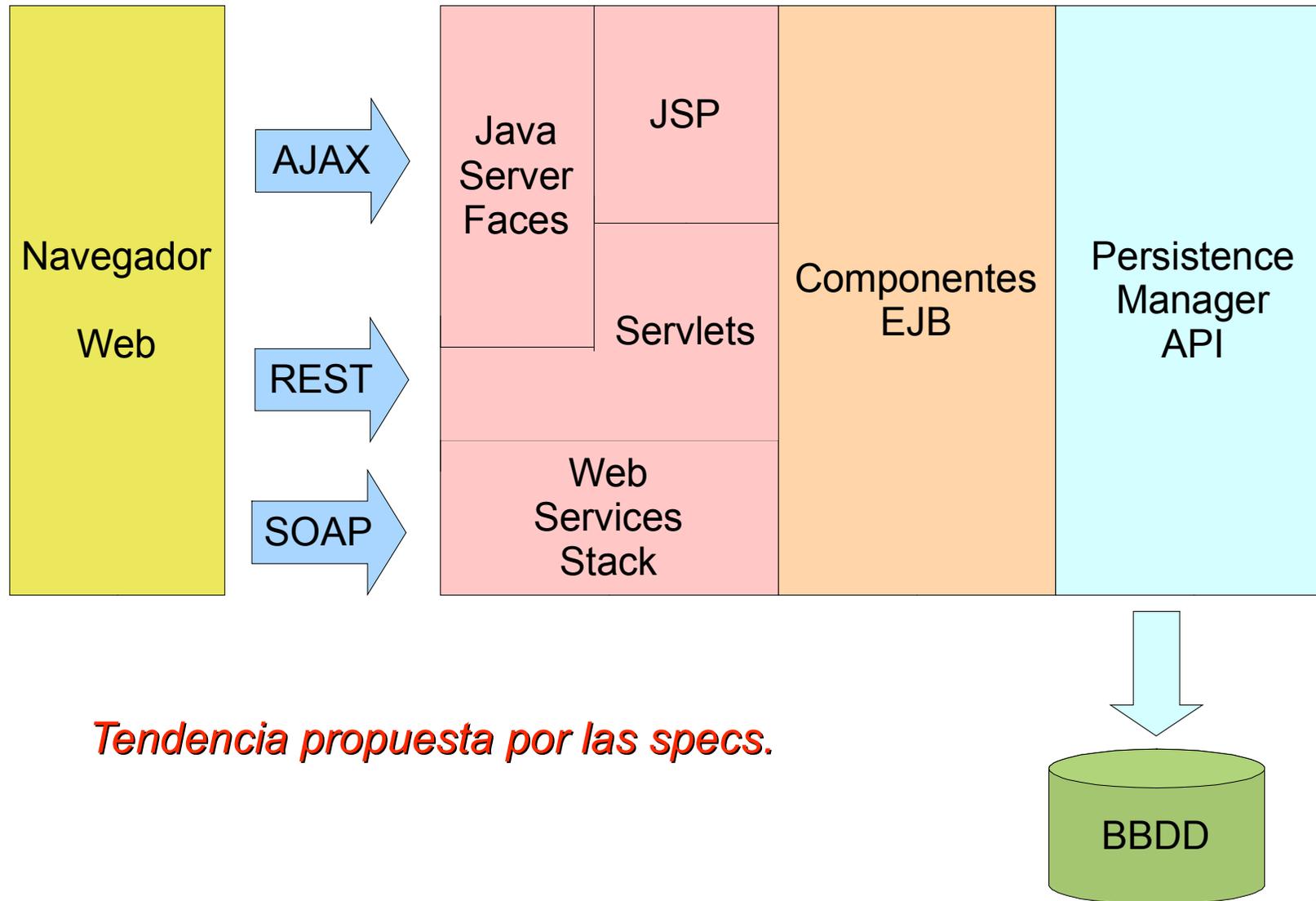
Modelos y tendencias de desarrollo

- Con el paso de los años se ha impuesto un modelo de desarrollo orientado a los frameworks.
 - Un framework te ofrece una base de desarrollo y servicios listos para utilizar.
 - No hay necesidad de invertir tiempo, dinero y recursos en reinventar la rueda.
 - Las compañías prefieren estándares oficiales (Java Server Faces) o estándares de facto (Struts, Spring, Hibernate).
 - Los desarrolladores pueden centrarse en construir funcionalidades sobre los frameworks.
- Los frameworks-de-casa están en declive
 - Demasiado trabajo para hacer siempre lo mismo
 - Es difícil ofrecer servicios sobre algo no estándar. [jLibrary]
 - Es difícil encontrar gente que extienda los sistemas. [javahispano]
- Los frameworks con éxito han influenciado muchísimo en J2EE.
 - Ej: Hibernate, Struts, ...

Modelos y tendencias de desarrollo

- La tendencia es a construir stacks completos sobre Open Source
- Los frameworks propietarios prácticamente han desaparecido
 - ¿Quién va a utilizar frameworks de pago con lo que hay actualmente?
- Las grandes empresas están liberando sus frameworks
 - Ej: Oracle ADF (Java Server Faces)
 - Ej: BEA, XMLBeans, KODO, ...
- Existen frameworks excelentes cubriendo todas las partes del desarrollo empresarial
- Hubo gente que predijo el caos debido a la variedad de frameworks
- El mundo del Open Source se ha mostrado muy efectivo. Todos han ganado.
- Las empresas ya no venden frameworks, venden soporte, formación y servicios. Ej: Spring (Apache License) e Interface21.

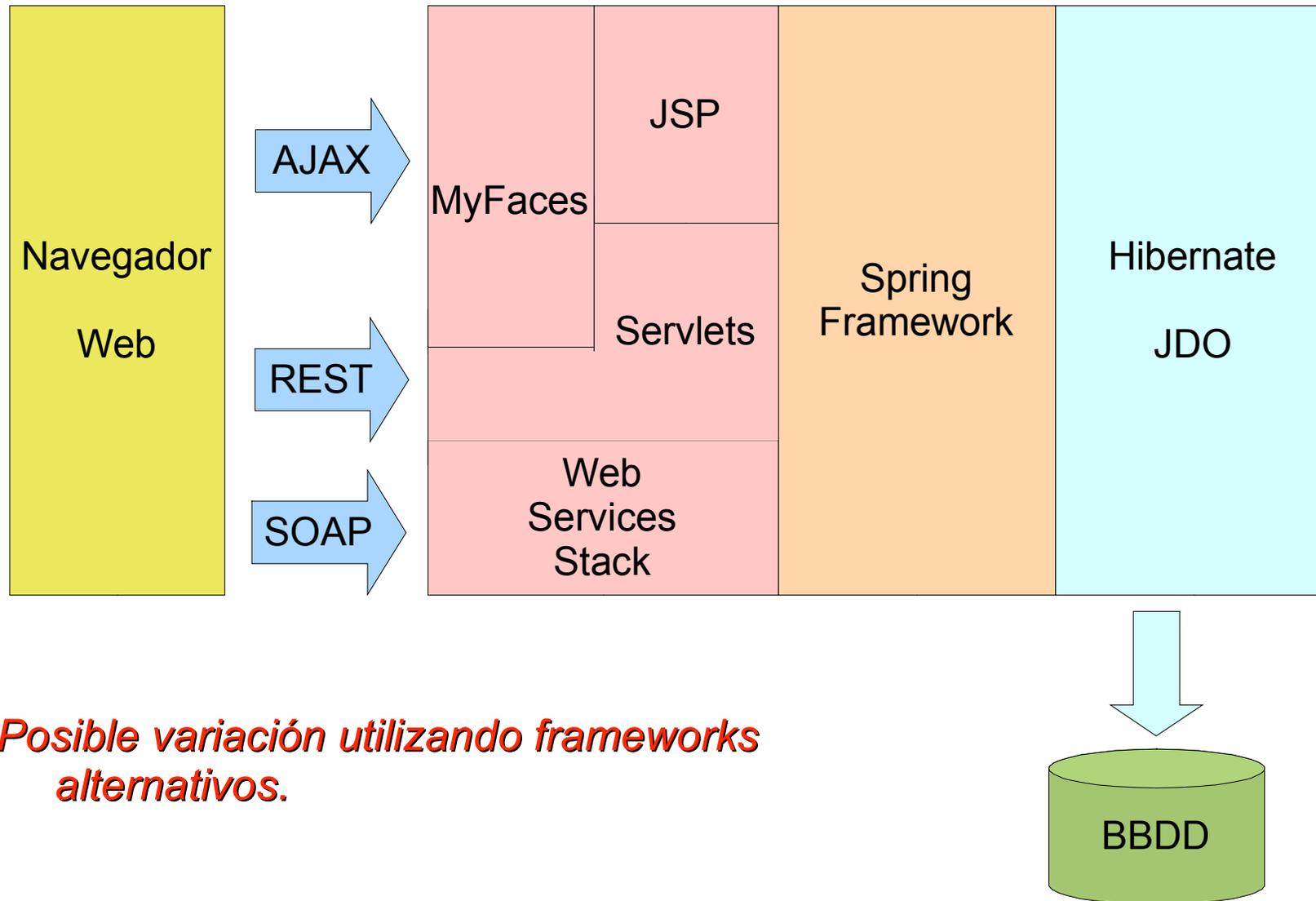
Modelos y tendencias de desarrollo



Modelos y tendencias de desarrollo

- Pero también hay críticas a las especificaciones:
 - Casi todas provienen de las comunidades de frameworks más afectados por la especificación.
 - Sobre todo Spring.
 - Críticas a la inyección de dependencias e interceptores.
 - AOP no aparece realmente como un actor de primer nivel. El soporte es muy light.
 - La comunidad JDO también clama su superioridad.
- La realidad es que los frameworks siempre van a ser más específicos que el estándar que tiene que mantenerse más neutral.
- El estándar se muestra fantástico para situaciones habituales.
- Hay hueco para todo:
 - Siempre es una buena elección seguir el estándar...
 - ... pero habrá ocasiones en que será necesaria más flexibilidad.

Modelos y tendencias de desarrollo



Possible variation using alternative frameworks.

Java EE 5.0

■ Ventajas

- La especificación Java Server Faces ofrece pocos componentes. MyFaces ofrece muchos más, pero no estándar.
- Spring se presenta como un framework de gran potencia.
- Permite aprovechar al máximo el contenedor de IoC y la programación orientada a aspectos.
- Ofrece un contenedor ligero para la ejecución de aplicaciones sin la carga de un stack completo Java EE.
- Hibernate está disponible y tiene una gran comunidad. No hay que esperar a que los servidores lo soporten
- JDO ofrece funcionalidades no presentes en EJB

Java EE 5.0

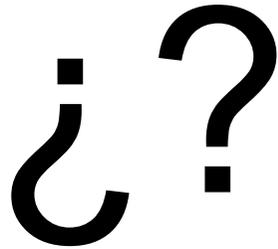
■ Desventajas

- Muchos componentes no estándar: Spring, Hibernate, etc.
- Alta probabilidad de quedarse atado a un framework o producto.
- Se pierde toda la potencia extra de Java EE: timers, mensajería, soporte implícito de servicios web, etc.
- A más número de frameworks más necesidades de formación extra.
- A veces surgen complejidades para la integración de diferentes frameworks. En un stack estándar todo viene ya preparado.
- El nuevo API de persistencia cubre la mayor parte de necesidades de un proyecto.
- JDO tiene mayor complejidad.

Java EE 5.0



- ¿Cuál escojo entonces?



Modelos y tendencias de desarrollo



Conclusiones

■ Conclusiones

- J2EE es una plataforma muy saludable. Una buena salida profesional en todos los sentidos: salario, aprendizaje, experiencia, ...
- Java EE 5 promete traer muchos cambios al desarrollo con J2EE.
- Java EE 5 busca simplificar el desarrollo en J2EE que tradicionalmente ha sido muy complejo.
- El desarrollo con EJBs ha evolucionado. Más fácil, más flexible, más potente.
- Los modelos ágiles de desarrollos e han enraizado en J2EE.
- Java EE es una especificación realmente fuerte que amenaza la supervivencia de muchos frameworks. Competencia. Beneficio para todos.
- La tendencia es el desarrollo contra frameworks.
- La tendencia es el desarrollo con componentes Open Source.

**Gracias
y
mucho suerte**