



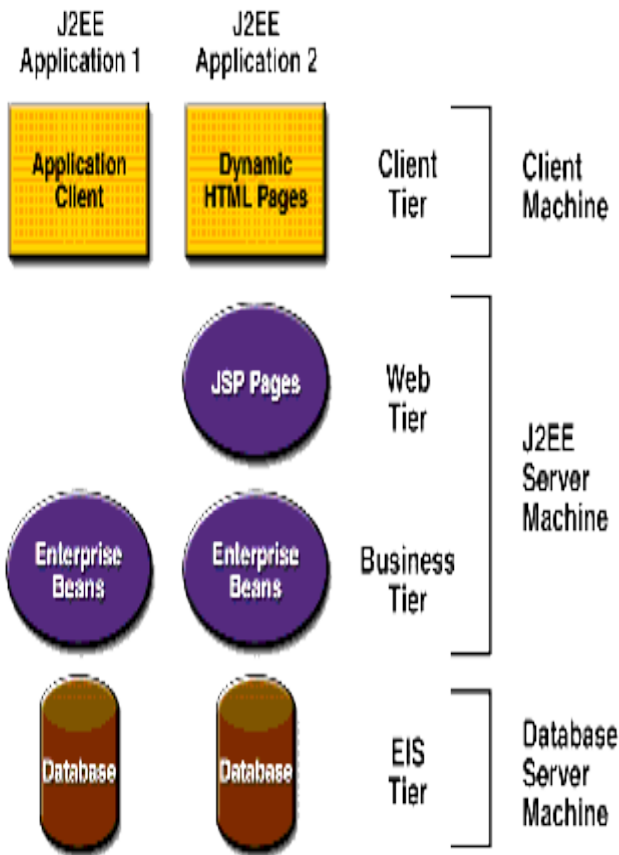
Plataforma J2EE 1.4

APIs, Melhorias e Diferenças da nova
versão do framework.

Por Rodrigo Urubatan Ferreira Jardim
rodrigo@usiinformatica.com.br

<http://www.rsjug.org>

Componentes da plataforma J2EE



- EJB 2.1
- JMX 1.0
- JNDI 1.2.1
- JTA 1.0.1
- Deployment API 1.1
- Java Mail 1.3
- Servlets 2.4
- JSP 2.0
- WebServices (JAXP, JXR, XML-RPC, SOAP-A)



Enterprise Java Beans 2.1



- A idéia por trás dos Enterprise Java Beans é facilitar o desenvolvimento de aplicações complexas, para isto o container onde os EJBs são executados oferece diversos serviços de “baixo nível” para que não seja necessário se preocupar com eles, estes serviços vão desde segurança à demarcação de transações declarativa.
- Para isto, temos 3 tipos de Enterprise Java Beans:
 - Message Driven Beans
 - Session Beans
 - Entity Beans (que se dividem em mais dois tipos)
 - Bean Managed Persistence
 - Container Managed Persistence



Message Driven Beans



- MDBs são a maneira J2EE de trabalhar com processamento assíncrono, já que a especificação proíbe a utilização de threads explicitamente.
- Mas MDBs são muito mais do que “threads” mesmo porque eles não rodam no mesmo “programa” que quem os chamou, eles funcionam com processamento de eventos, mas em vez de eventos, eles recebem mensagens JMS.
- Muitos ambientes, como por exemplo o SPB (Sistema de pagamentos brasileiro) utilizam MDBs para integração de sistemas distantes e integração com ambientes mainframe legados.

Session Beans

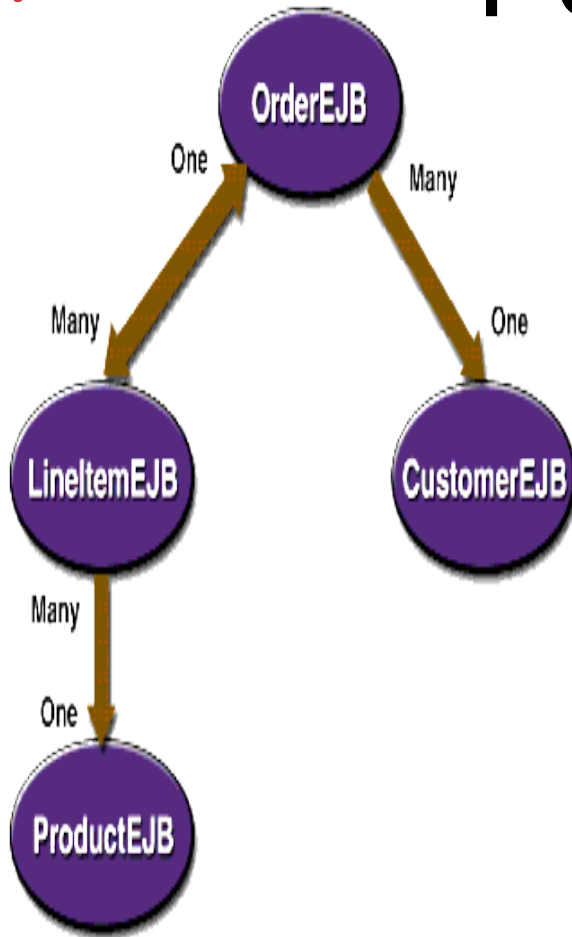
- Session Beans são o lugar certo para implementar as regras de negócio em um ambiente J2EE, eles podem ser Statefull ou Stateless, isto é, podem ou não armazenar o estado anterior entre chamadas de métodos
- Em um ambiente totalmente J2EE, os Session Beans são, normalmente, o ponto de junção entre a aplicação cliente, e o servidor onde ficam armazenadas as regras de negócio.
- Uma grande inovação nesta area na versão 1.4 da plataforma J2EE é a possibilidade de expor qualquer Stateless session bean como um WebService, mas veremos mais adiante todo o suporte a WebServices introduzido na plataforma J2EE.



Bean Managed Persistence

- Entity Beans deste tipo, são o lugar correto para se implementar persistência da maneira “antiga” dentro do ambiente J2EE.
- Este tipo de EJB representa uma entidade persistente, mas delega ao programador, persistir e recuperar estes dados.
- Normalmente esta abordagem é utilizada para interagir via banco de dados com sistemas legados.

Container Managed Persistency



- Entity beans deste tipo, são a maneira preferida de se persistir dados em um ambiente J2EE, estes beans desfrutam de diversos serviços adicionais que não estão disponíveis para os entity beans do tipo BMP, como por exemplo:
 - CMR – Container Managed Relationships, o container cuida automaticamente dos relacionamentos entre os entity beans
 - Abstract Schema – o deployment descriptor destes EJBs informa ao container todos os campos que devem ser persistidos e o relacionamento entre este e os demais Entity beans da aplicação



Container Managed Persistency - EJBQL



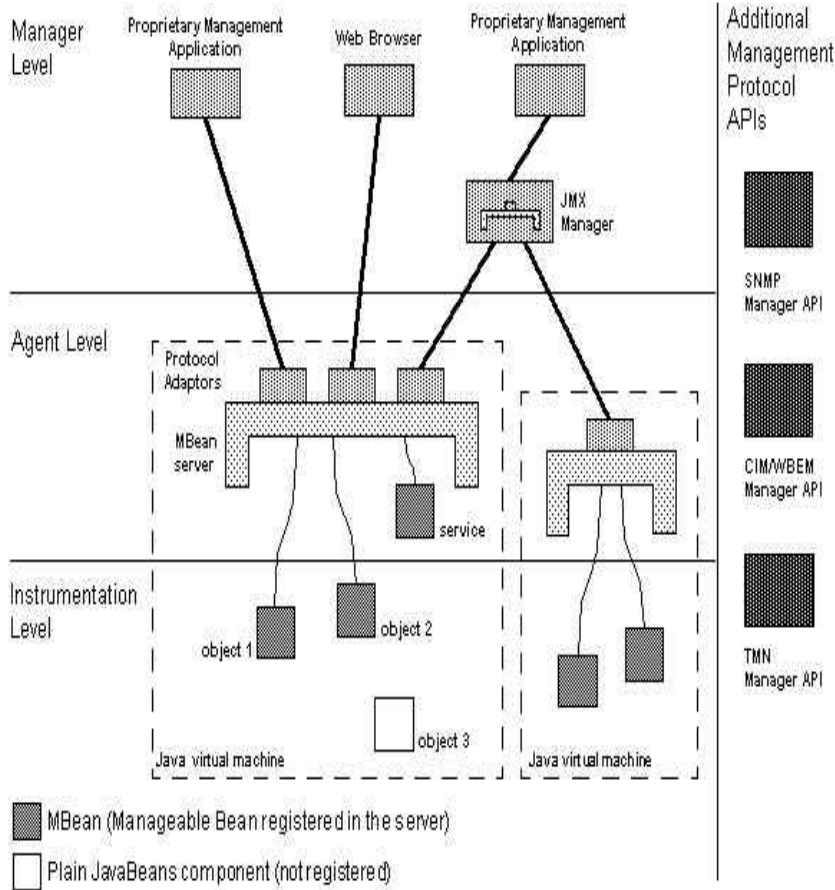
- A EJBQL é a linguagem de consulta para os Entity Beans, formada por um subset do próprio SQL com algumas alterações para possibilitar a consulta de objetos.
- Nesta linguagem estão as inovações mais esperadas da especificação, as funções de agregação para consultas.
- Foram adicionadas as seguintes funções de agregação: AVG, MAX, MIN, SUM, COUNT e a opção DISTINCT.
- Desta maneira flexibilizando muito mais as consultas, e tornando a EJBQL finalmente utilizável.

Timer Service



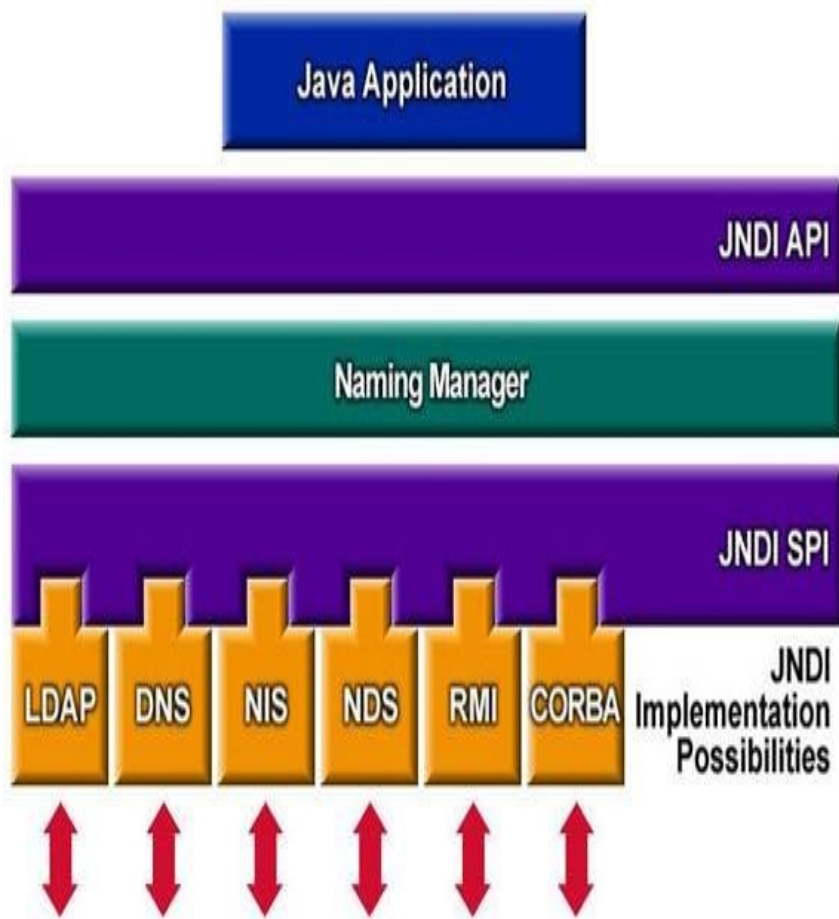
- A grande novidade introduzida na especificação de EJBs 2.1 é o serviço de timer a tanto tempo esperado, este serviço possibilita o agendamento da execução de métodos de qualquer tipo de EJB em um determinado momento, por exemplo, pode ser agendado o cálculo da folha de uma empresa para o dia primeiro do mês, ou melhor ainda, para todo o dia primeiro, coisa que sempre precisou ser feita por um processo externo ao ambiente J2EE até a introdução deste serviço.
- Um timer criado para este serviço, está sempre atrelado a um EJB específico, no caso de Statefull session beans ou de entity beans, que possuem um identificador único, o timer é atrelado a uma instância do EJB, tendo a vantagem de um estado previamente setado durante a execução do intervalo do timer, no caso de MDBs ou de Stateless session beans, o timer fica atrelado ao tipo/classe do EJB e não a uma instância específica.
- O serviço de timer, como todos os serviços providos por um Container J2EE, pode utilizar o controle de transações e todos os outros serviços providos.

Java Management Extensions 1.0



- A inclusão do JMX na especificação J2EE foi um grande avanço, utilizando esta especificação, é possível desenvolver ferramentas unificadas para o gerenciamento total de todos os containers J2EE de uma maneira unificada, facilitando assim a diversificação do ambiente, e a centralização do controle sobre o ambiente.
- A especificação JMX inclui interfaces para configuração e monitoramento via SNMP e EJBs (MEJB), possibilitando a integração de diversos tipos de ferramentas para monitoração e configuração de serviços.
- Desta maneira, pode-se gerenciar ou monitorar cada serviço disponibilizado por um container, através das mesmas ferramentas.

Java Naming and Directory Interface 1.2.1



- A API JNDI é o ponto de encontro de todos os tipos de gerenciamento de nomes, incluindo DNS, LDAP, NDS, RMI, CORBA e até mesmo diretórios locais.
- O JNDI é a maneira padrão J2EE de buscar referencias de objetos, e de mapear um nome para um serviço, desta maneira, unificando o acesso e a distribuição de objetos na rede.
- Atravez de um registro unificado JNDI para todo o cluster de servidores de aplicação, todos os componentes e clientes da aplicação, tem acesso a todos os serviços disponibilizados pelo container.



Java Transaction API 1.0.1

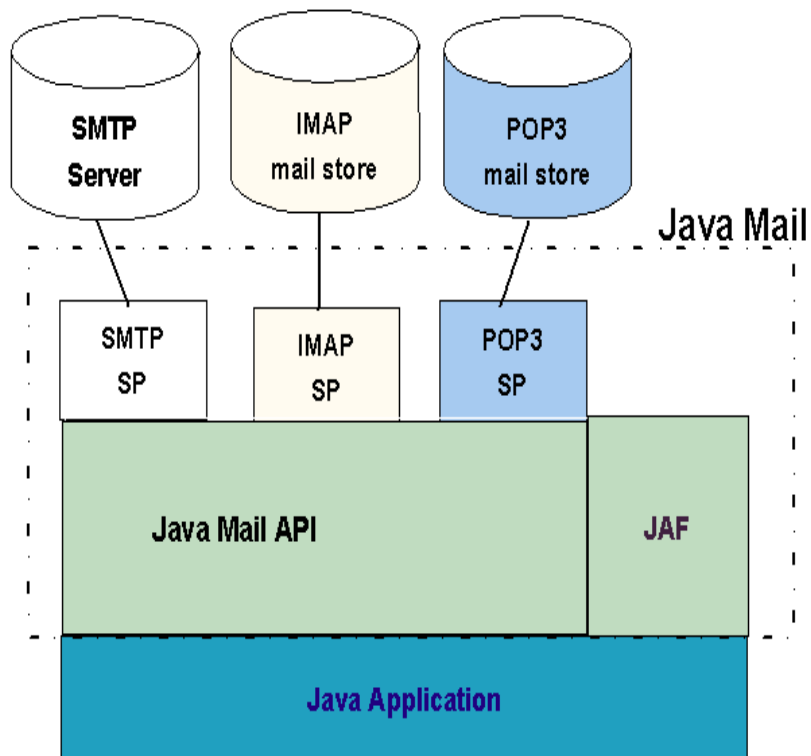


- A API de transações J2EE é a maneira utilizada para unificar todos os tipos de gerenciamento de transações intrínsecos às regras de negócio, podendo fazer parte da mesma transação, desde o acesso a recursos de bancos de dados, ao acesso a arquivos em disco e gerenciamento de documentos, bastando para isto que o container disponibilize um resource manager para o recurso em questão.
- A utilização desta API deixa totalmente transparente para o cliente o tipo de recurso que deve ser controlado, entrando dentro da mesma transação já controlada pelo container, todo e qualquer recurso que possua um resource manager disponível

Deployment API 1.1

- Outra grande adição à plataforma J2EE foi a deployment API que veio para colocar ordem no caos que era disponibilizar uma mesma aplicação em diversos containers diferentes.
- Com a utilização desta API, tornou-se possível, integrar o deploy de uma aplicação ou de apenas uma parte da aplicação em todas as ferramentas de desenvolvimento utilizadas, ou até mesmo em scripts ANT por exemplo, e incluir o deploy da aplicação em mais de um servidor de aplicações dentro do ciclo de integração contínua do desenvolvimento de um sistema J2EE.

Java Mail 1.3



- A API JavaMail é a maneira default de se enviar e receber e-mails em um ambiente J2EE.
- Utilizando esta API pode-se de maneira fácil enviar e receber e-mails utilizando todos os protocolos e formatos padrão de e-mail utilizados na Internet, incluindo as mensagens multi parte (MIME) codificadas de diversas maneiras diferentes.
- As maiores adições nesta API foram o suporte a novos formatos de e-mail e o melhor suporte à autenticação.



Servlets 2.4

- A especificação de Servlets define toda a camada WEB de uma aplicação J2EE, mesmo as JSPs são compiladas para servlets na sua primeira execução.
- A especificação de servlets inclui a API para servlets, filtros, gerenciamento de sessão de usuários e diversos tipos de listeners para diversas situações diferentes em uma aplicação WEB.
- As principais inovações na nova versão desta API estão no maior detalhamento de configurações no deployment descriptor (web.xml) e em diversos novos níveis de configuração, como por exemplo, agora é possível especificar em que situações um filtro vai ser ativado, e não apenas um pattern de URL ou servlet para ele.
- Foram adicionadas também diversas novas configurações de segurança e integração com JSPs, como por exemplo, especificar quando toda a aplicação pode ou não utilizar Scriptlets ou EL em suas JSPs.



Java Server Pages 2.0



- Uma das maiores inovações na plataforma J2EE 1.4 foi a nova especificação de JSPs, esta nova versão inclui suporte obrigatório no container para a JSTL, inclui uma nova linguagem de scripting para as JSPs (a EL, Expression Language) e diversas possibilidades de configurações.
- É possível especificar por página ou para toda a aplicação quando a EL e os scriptlets estão habilitados, foram adicionados TLVs padrão para facilitar as configurações (Tag Library Validators) que são utilizados para especificar quais as TagLibraries que podem ou não ser utilizadas, e até para especificar qual a sintaxe que é válida para a EL em uma página ou na aplicação.
- Foram adicionadas também dois novos modos de componentização para as páginas do sistema, as TagFiles e as Simple Tags.

Componentização de JSPs

- Tag Files – são pedaços de JSP reutilizáveis, que são compilados em tempo de execução para Simple Tags
- Simple Tags – é uma simplificação da API para taglibraries, onde se precisa simplesmente implementar um método, ao contrário da abordagem anterior (que continua válida) onde era necessário implementar diversos métodos para criar uma tag
- E por fim, foi adicionada a possibilidade de tags terem um número variável de parâmetros, simplesmente implementando uma interface a mais, aumentando muito a flexibilidade desta tecnologia



JSTL 1.1

- Na nova especificação de JSPs foi incluída o suporte a versão 1.1 da JSTL (Java Standard Tag Library)
- Esta versão da JSTL inclui o suporte à execução de funções (tags de funções, em vez de geração de HTML) e aumenta bastante o poder dos TLVs
- As tags da JSTL continuam as mesmas com a adição das function tags, e o suporte à Expression Language 2.0 em vez da 1.0



EL 2.0

- A nova versão da Expression language, além de poder ser utilizada diretamente em uma JSP, também ficou muito mais poderosa, facilitando e simplificando o desenvolvimento WEB, mas continua como na proposta original, sem suporte à alteração de dados.
- Foi adicionado o suporte à chamada de funções, desde que estas sejam definidas como Function tags, que são basicamente uma nova versão das antigas TagLibs, e foi bastante estendida a sintaxe, possibilitando a eliminação de diversos “IFs” de uma página, e foi também adicionado o suporte à expressões, como o operador ternário do java.

Web Services

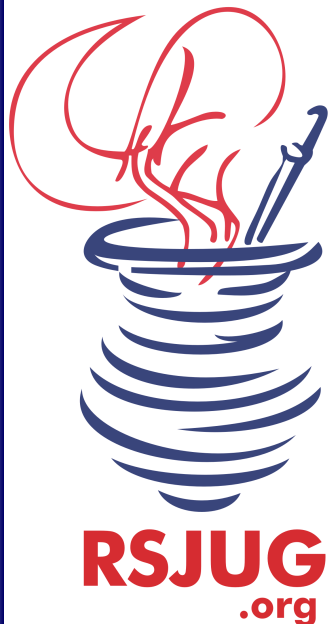
- Na nova versão da plataforma J2EE foi adicionado um suporte bom (mas não perfeito) a WebServices, foram adicionadas APIs de suporte para esta tecnologia, como JAXP, JXR e JXRPC, e principalmente foi adicionado o suporte nativo em todos os containers EJB para expor qualquer Stateless Session Bean como um WebService, bastando para isto especificar no deployment descriptor esta opção.
- A parte cliente dos WebServices também está bastante facilitada, tendo sido incluída a biblioteca JXRPC, ou então o cliente pode ser implementado em qualquer outra linguagem, como Delphi, C++ ou na plataforma .NET, sendo uma ótima opção para a integração de plataformas.





Referências

- **Java 2 Enterprise Edition, Home**
<http://java.sun.com/j2ee/index.jsp>
- **The J2EE 1.4 Tutorial**
<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>
- **JCP, J2EE Specifications**
<http://www.jcp.org/en/jsr/tech?listBy=3&listByType=p>



Rodrigo Urubatan Ferreira Jardim
Consultor/Desenvolvedor J2EE



Sun Certified Web Component Developer for J2EE 1.4
Sun Certified Programmer for the Java 2 Platform 1.4

rodrigo@usiinformatica.com.br
<http://www.usiinformatica.com.br>

PATROCÍNIO



APOIO ESPECIAL



APOIO



APOIO DIVULGAÇÃO



ORGANIZAÇÃO

