

MODELO DE DESENVOLVIMENTO DE APLICAÇÕES CORPORATIVAS COM *JAVA ENTERPRISE EDITION 5*

**Gilbert AZEVEDO da Silva (1); Fellipe Araújo ALEIXO (2); Rosemary Pessoa BORGES (3);
Wanderson CÂMARA dos Santos (4); Bruno Gomes de ARAÚJO (5)**

(1) CEFET-RN, Av. Sen. Salgado Filho, 1559, Tirol, Natal-RN – CEP: 59015-000, telefone: (84) 4005 2637,

e-mail: gilbert@cefetrn.br

(2) CEFET-RN, e-mail: fellipe@cefetrn.br

(3) CEFET-RN, e-mail: maryborges1@yahoo.com.br

(4) CEFET-RN, e-mail: wan007@gmail.com

(5) CEFET-RN, e-mail: bga3000@yahoo.com.br

RESUMO

Ao passo que as tecnologias relacionadas com o desenvolvimento de sistemas evoluem, aumenta a complexidade dos sistemas que se deseja desenvolver. O aumento da complexidade dos sistemas a serem desenvolvidos, e o fato do desenvolvimento necessitar acontecer em menos tempo (por uma questão de competitividade), tem levado ao desenvolvimento de mecanismos de *middleware*. Um *middleware* visa oferecer uma série de serviços prontos para serem utilizados pelos desenvolvedores de aplicações. O nome de aplicações corporativas foi associado a tais aplicações que por sua complexidade, fazem uso de serviços de *middleware* para acelerar o seu desenvolvimento, concentrando esforços na lógica do negócio. Uma das plataformas para o desenvolvimento de sistemas corporativos foi desenvolvida pela Sun Microsystems, e é chamada de *Java Enterprise Edition*, atualmente na sua versão 5, o JEE 5. O objetivo desse trabalho é a definição de um modelo de desenvolvimento de aplicações corporativas baseado na plataforma JEE 5. O desenvolvimento deste modelo baseia-se em um estudo de caso, onde a plataforma JEE 5 é utilizada para o desenvolvimento. São mostradas as fases desse desenvolvimento, bem como as implicações da escolha da plataforma na arquitetura da aplicação a ser desenvolvida. A seguinte metodologia será utilizada para o desenvolvimento do trabalho: realizar uma revisão bibliográfica das tecnologias envolvidas no desenvolvimento; levantar os requisitos e especificar o sistema a ser desenvolvido; definir a arquitetura do sistema a ser desenvolvido com base na plataforma escolhida para o desenvolvimento; e apresentar aspectos relativos a fase de implementação e testes. Será adotado para o desenvolvimento do sistema proposto uma customização do processo unificado.

Palavras-chave: *Java Enterprise Edition*, Arquitetura de Software, Aplicações Corporativas.

1. INTRODUÇÃO

Com o acelerado crescimento do número de aplicações, desenvolvedores estão buscando alternativas que se adequem a atual realidade competitiva dos desenvolvimentos de sistemas. Segundo JENDROCK, BALL, CARSON, EVANS, FORDIN e HAASE (2006), aplicações necessitam ser rápidas, seguras e confiáveis, ao custo de uma redução no tempo e recursos no desenvolvimento.

A plataforma Java EE 5 sugere um modelo de programação simplificada, que atende as atuais necessidades de desenvolvimento. Surgiu com o objetivo de auxiliar na implementação de aplicações corporativas, e oferece diversas facilidades e vantagens sobre as demais plataformas existentes.

A plataforma oferece toda a infra-estrutura necessária para o desenvolvimento de aplicações, permitindo aos desenvolvedores focarem na lógica de negócio. Isto proporciona uma maior segurança, escalabilidade e uma maior reusabilidade futura do sistema, ganhando um grande destaque na escolha pelos desenvolvedores para a implementação de sistemas com nível de complexidade alta.

Como estudo de caso, a plataforma foi escolhida para o desenvolvimento de um sistema corporativo que necessita atender alguma das necessidades citadas anteriormente, como segurança e escalabilidade principalmente. O sistema é focado na camada de persistência e visualização de dados. Ao final, serão mostrados todas as facilidades e vantagens proporcionadas com a utilização das tecnologias que a plataforma oferece.

2. APLICAÇÕES CORPORATIVAS

A partir do momento em que as empresas perceberam o poder de divulgação da Internet houve um interesse, uma necessidade de usá-la com objetivo de mostrar-se diante o mundo, conquistar clientes e colocar-se à frente de seus concorrentes. Neste momento muitas empresas usavam a Internet somente como uma vitrine para seus produtos. Porém o cliente, com mais informações, tornou-se exigente e a concorrência aumentava no mundo digital, percebeu-se a necessidade de uma diferenciação não só na qualidade de produtos e serviços, mas também de atendimento. A tendência é de aceitar que cada cliente é único e necessita resolver suas pendências independente de sua localização geográfica.

Diante deste contexto, as Aplicações desenvolvidas devem ir muito além das simples vitrines, devem, por exemplo, contemplar atendimento ao cliente, disponibilizar serviços referente sua área de atuação (possíveis de ser implementados através deste meio) e além de tudo devem ser seguras e confiáveis. Com objetivo de suprir estas necessidades surgiram as Aplicações Corporativas.

Para garantir o sucesso deste verdadeiro empreendimento, que é o desenvolvimento de uma Aplicação Corporativa, é necessária uma linguagem confiável que ofereça alta performance e que seja customizável as necessidades da organização, pensando nestes requisitos a linguagem Java foi a escolha mais sensata.

2.1. *Middleware*

Com um sistema tão complexo não é interessante e nem um pouco produtivo pensar por exemplo, em implementar serviços de interconexão entre cliente e servidor. Para isto utiliza-se de *Middlewares*, que proporciona aos desenvolvedores manter o foco no desenvolvimento da lógica de negócio. Tarefas básicas que são necessárias a qualquer Aplicação Corporativa como segurança, gerenciamento de ciclo de vida, transação, persistência entre outras, passam a ser de responsabilidade do *Middleware*. Entre os mais conhecidos merecem destaque o JBoss (Red Hat), WebSphere (IBM), Geronimo (Apache) e WebLogic (BEA).

3. JEE 5

A tecnologia Java criada pela Sun Microsystems permite que Aplicações rodem em diferentes ambientes desde que possuam a JVM (*Java Virtual Machine*), esta por sua vez, possui um tipo de tradutor para cada dispositivo específico.

A tecnologia Java é subdividida em (LEWIS, 2007):

- JME (*Java Micro Edition*) – desenvolvida para dispositivos que possuem baixa capacidade de memória e processamento;

- JSE (Java *Standard Edition*) – rica plataforma direcionada predominantemente a aplicações desktop, sendo a base para o JEE;
- JEE (Java *Enterprise Edition*) – especificação destinada a Aplicações Corporativas.

O crescente aumento do número de aplicações corporativas, e a necessidade de interconexão das mesmas, fez com que um grupo de desenvolvedores da Sun se empenhasse em definir uma especificação com o objetivo de padronizar uma solução completa para desenvolvimento de sistemas em Java. Como resultado desse empenho, surgiu o Java *Enterprise Edition* – JEE. Essa especificação encontra-se atualmente na versão 5, sendo denominada de JEE 5 (SAMPAIO, 2007).

O JEE5 compreende um vasto conjunto de tecnologias que abrangem uma série de especificações voltadas principalmente para aplicações corporativas, contendo quase todos os pontos vitais para o desenvolvimento de aplicações de alta disponibilidade. Dentre as tecnologias reunidas pelo JEE5, três, em especial, merecem destaque: *Servlet*, JSP (Java *Server Pages*) e EJB (*Enterprise JavaBeans*).

- *Servlet* – São classes pré-compiladas que residem em um servidor e processam requisições e devolvem respostas, baseado no paradigma cliente/servidor. Essa tecnologia é bastante utilizada na criação de conteúdos dinâmicos e em controles de aplicações JSP (GONÇALVES, 2007).
- JSP – É utilizada para a criação de páginas *web* dinâmicas. São páginas HTML com código Java (tags) embutido que operam como *Servlets* (ANSELMO, 2005).
- EJB – Formam o núcleo de uma aplicação distribuída, são os objetos que devem tratar a regra de negócio de domínio do problema. Os EJBs são gerenciados por um contêiner e possuem acesso à serviços providos pelo mesmo (SULLINS, 2004).

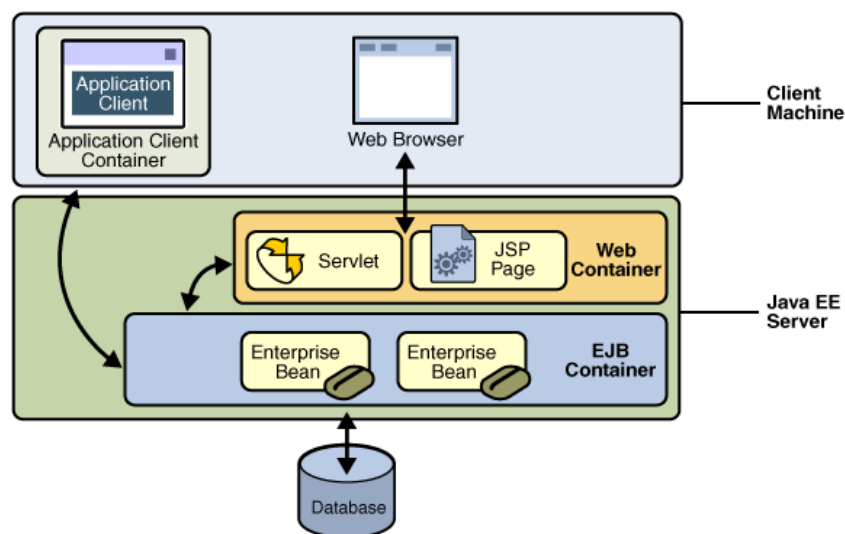


Figura 1 - Java EE: Cliente e Servidor. No servidor o Container Web (Servlets e JSPs) e o Container EJB.

JEE5 ainda compreende muitas outras tecnologias além das citadas a cima, como Transações (JTA), conectividade a bancos de dados (JDBC), tratamento de mensagens (JMS), serviço de diretórios (JNDI), acesso a sistemas legados (JCA) dentre outras. Essa quantidade de APIs fornece a base de infra-estrutura para que os serviços de *middleware* possam ser fornecidos.

4. ESTUDO DE CASO

Para dar resposta a necessidade de acesso às informações das várias unidades acadêmicas que oferecem educação profissional no país, foi proposto um sistema que centralize tais informações, denominado de SIEP Gerencial. O SIEP Gerencial atua desde a extração dos dados a partir dos sistemas de informação de cada unidade acadêmica, até a disponibilização dessas informações para os vários sistemas no âmbito do Ministério da Educação.

O SIEP Gerencial foi estruturado em cinco módulos independentes e complementares. O fraco acoplamento proporcionado pela arquitetura escolhida irá facilitar a manutenção e as futuras alterações. Os nomes dos módulos definidos foram: Extrator, Atualizador, Seletor, Relator e Sincronizador.

Atualmente, as unidades acadêmicas de Educação Profissional e Tecnológica – EPT – do país possuem seus próprios sistemas, cada sistema organiza os dados de uma forma particular. O Módulo Extrator tem por finalidade principal a extração dos dados necessários ao SIEP Gerencial a partir das várias bases de dados utilizadas pelas unidades acadêmicas de EPT. Para a coleta das informações extraídas, o Módulo Extrator disponibiliza um serviço web para receber a solicitação de coleta (HANSEN, 2007).

O Módulo Extrator ainda permite que o gestor da unidade acadêmica de EPT visualize os dados extraídos, com a opção de realizar alterações nos mesmos e por fim autorizar a coleta desses dados, esquematicamente demonstrada na Figura 2. Todas as alterações realizadas pelo gestor serão registradas pelo sistema. Após a autorização de coleta por parte do gestor os dados são assinados digitalmente e não mais passíveis de alteração até que a coleta dos mesmos tenha sido processada.



Figura 2 – Fração do Diagrama de Casos de Uso do Módulo Extrator.

Para a visualização dos dados extraídos foi utilizada a tecnologia JSF, disponível no JEE5, por incorporar características de um *framework* MVC (*Model-View-Controller*) para *web*. No JSF, o *servlet FacesServlet* é o responsável pelo controle, ou seja, não é necessário que o desenvolvedor implemente um *servlet* para esta função, o próprio JSF disponibiliza. Para a visualização são utilizadas páginas JSP's que usam *TagLibs* do JSF, para que os dados possam ser acessados através das páginas JSP's o controlador (*FacesServlet*) faz a ligação entre a visualização (JSP) e o modelo que é composto por um simples *bean* (EJB).

O *bean* é responsável por disponibilizar as entidades, que atualmente são: instituição, campus, curso e oferta. Estas entidades estão persistidas no banco de dados, caso haja adição de uma nova entidade não será necessário alterar o *bean*, já que ele recebe todas as entidades existentes no banco. A JSP através das *TagLibs* do JSF acessará o *bean* e exibirá as entidades disponíveis.

O Gestor poderá escolher a entidade desejada através de seu browser. O *bean* será informado sobre a escolha, e auxiliado pelo pacote *java.lang.reflect* criará uma nova instância deste objeto (entidade), selecionará os atributos (que serão utilizados posteriormente) e fará uma requisição dos dados desta entidade ao responsável pela comunicação com o banco de dados. Para este intermediário será repassado somente a entidade escolhida, auxiliado por outra tecnologia do JEE5 chamada Java *Persistence API* (JPA), que disponibiliza inúmeras vantagens às quais não serão discutidas aqui, este intermediário através de uma simples linha de código selecionará do banco os dados referentes aquela entidade.

Os dados serão exibidos na tela organizados em tabela, de forma que, cada coluna é intitulada com o nome do atributo referente ao dado exibido, os dados serão dispostos de acordo com o atributo.

A escolha da tecnologia JSF foi a mais adequada neste caso por facilitar o desenvolvimento de tal funcionalidade do sistema, de forma que permitiu a comunicação entre visão e modelo sem que fosse necessário o programador fazer esta configuração. O uso do pacote *java.lang.reflect* também foi importante por permitir em tempo de execução, ao ser identificado o tipo de objeto, a seleção dos atributos que identificam os dados de uma determinada entidade. O JPA demonstra grande simplificação no acesso aos dados do banco de dados, facilitando ao programador realizar ações relacionadas à persistência.

5. CONCLUSÃO

A utilização da especificação Java *Enterprise Edition 5* mostrou-se adequada à aplicação proposta. Permitindo, através de suas tecnologias, a dinamicidade da aplicação de forma que as informações não são

previamente determinadas tudo ocorrerá em tempo de execução, esta característica será muito importante, pois novas entidades, provavelmente, serão adicionadas ao sistema. No futuro, a aplicação fornecerá suporte à alteração dos dados extraídos, o que também não afetará o sistema devido à escalabilidade obtida através do uso das tecnologias descritas.

REFERÊNCIAS

BURKE, Bill; MONSON-HAEFEL, Richard. **Enterprise JavaBeans 3.0**. 5. ed. Sebastopol, Estados Unidos: O'Reilly, 2006.

JENDROCK, Eric; BALL, Jennifer; CARSON, Debbie; EVANS, Ian; FORDIN, Scott; HAASE, Kim; **The Java™ EE 5 Tutorial, Third Edition: For Sun Java System Application Server Platform Edition 9.3** ed.: Santa Clara, Califórnia: O'Reilly, 2006.

Sun Microsystem. **update An Introduction to the Java EE 5 Platform**. Disponível em: <http://java.sun.com/developer/technicalArticles/J2EE/intro_ee5/> Acesso em: 03 set 2007.

DEITEL, Harvey M. **Java – Como Programar**. Prentice Hall Brasil, 2007.

LEWIS John; CHASE, Joe; DEPASQUALE, Peter. **Java Foundations**. Addison Wesley, 2007.

SAVITCH, Walter; MOCK, Kenrick. **Absolute Java**. Addison Wesley, 2007.

SAMPAIO, Cleuton. **Guia do Java Enterprise Edition 5**. Brasport, 2007.

MCLAUGHLIN, Brett. **Building Java Enterprise Applications**. O'Reilly & Associates, 2002.

CRAWFORD, William; FARLEY, Jim. **Java Enterprise in a Nutshell**. O'Reilly & Associates, 2005.

ECKSTEIN, Robert. **Java Enterprise Best Practices**. O'Reilly & Associates, 2002.

MARK, Grand. **Java Enterprise Design Patterns**. John Wiley Consumer, 2001.

SULLINS, Benjamin G.; WHIPPLE, Mark B. **EJB – Livro de Receitas**. Ciência Moderna, 2004.

GONÇALVES, Edson; **Desenvolvendo Aplicações Web com JSP e Servlets**. Ciência Moderna, 2007.

ANSELMO, Fernando. **Tudo Sobre a JSP**. Visual Books, 2005.

SCHALK, Chris. **Java Server Faces – The Complete Reference**. MCGRAW-HILL Professional, 2006.

KEITH, Michael; HALEY, Jason; SCHINCARIOL, Merrick. **PRO EJB 3 – Java Persistence API**. Apress, 2006.

HANSEN, Mark. D. **SOA Using Java Web Services**. Prentice Hall, 2007.