Melhores práticas no desenvolvimento Java

Oziel Moreira Neto
Consultor

Novas tecnologias, velhos problemas.

Novas Tecnologias, Velhos Problemas

- Processo de construção
- Gerenciamento
- Monitoramento
- Performance
- Alta disponibilidade
- Evolutividade
- Prazo, Custo e Qualidade

Novas Tecnologias, Velhos Problemas

Processo de construção

- Rational Unififed Process (RUP)
- eXtreme Programming (XP)
- SunTone®
- Oriented Object Analisys and Design

Quer ter sucesso?
Escolha um e USE!

Novas Tecnologias, Velhos Problemas

Gerenciamento

- Inicio do projeto
- Durante a construção
- Durante os testes
- Durante a implantação
- Durante a manutenção

Gerenciar SEMPRE!

Novas Tecnologias, Velhos Problemas

Monitoramento

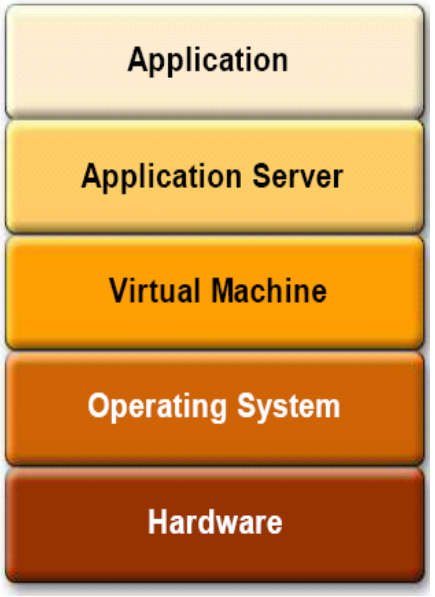
- Geração e coleta de dados:
  - da aplicação (esforço nos testes);
  - da infra-estrutura (esforço na implantação);
  - em ambos (esforço na produção);
- Execução de projeções
  - plano de capacidade (usuários X consumo);

Monitorar SEMPRE!

Novas Tecnologias, Velhos Problemas

Performance

- Irrelevante para os programadores;
- Lembrado pelos analistas;
- Esquecido pela implantação;
- Sofrido pela produção;



Medir a performance SEMPRE!

Novas Tecnologias, Velhos Problemas

Alta disponibilidade

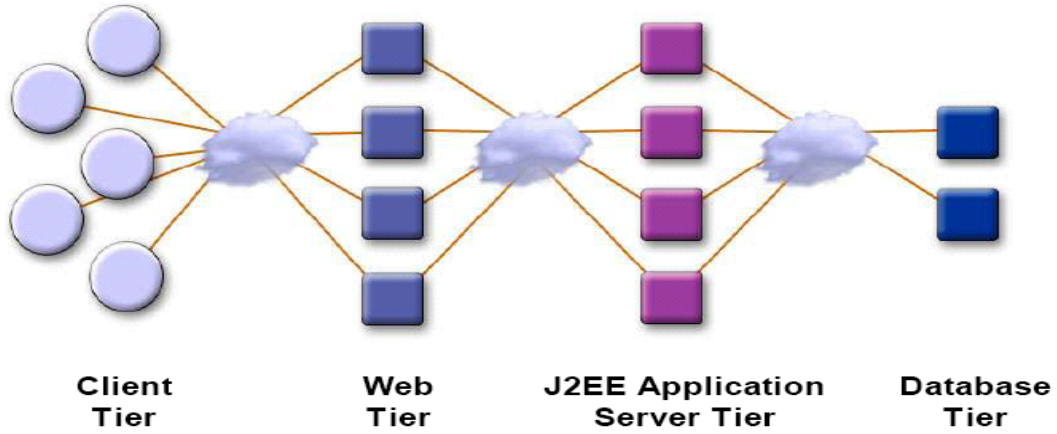
- Analistas não se preocupam;
- Soluções inadequadas ou inviáveis;
- Alta complexidade;
- Preocupação dos Arquitetos;

Arquitetura e modelo computacional!

Novas Tecnologias, Velhos Problemas

Alta disponibilidade

- Preocupação dos Arquitetos de Software;



Arquitetura e Padrões

Novas Tecnologias, Velhos Problemas

Evolutividade

- Manutenção dos sistemas:
  - "*como não sou eu que vou ter de arrumar isso, vou deixar assim mesmo...*" (pensamento dos programadores)
- Preocupação de arquitetos e analistas;

Componentizar para EVOLUIR!

Novas Tecnologias, Velhos Problemas

Prazo, Custo e Qualidade

- Preocupação de todos;
- Lei Universal:
  - *"não dá para fazer um filho com nove mães em um mês"* (desconhecido)
- Desafio: fazer com qualidade, dentro do prazo e sem custos excessivos.

Como equilibrar?

Melhores práticas.

Melhores práticas

- Java Blueprints
- Java Design Patterns
- Modelagem UML
- Codificação
- Especificações JavaSoft®
- Arquitetura
- Testes

Melhores práticas

Java Blueprints

- Estudar os Sun Java Blueprints;
  - http://java.sun.com/reference/blueprints/index.html
- Fazer provas de conceitos;
- Não se escreve código sem DOMINAR as tecnologias J2EE;

Acumular conhecimento é essencial

Melhores práticas

Java Design Patterns

- Entender e saber aplicar;
  - http://java.sun.com/blueprints/patterns/index.html
- Consenso entre arquitetos e analistas;
- Resolvem TODOS os problemas sistêmicos (aplicação e infra);
  - http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html

USAR Design Patterns

Melhores práticas

Modelagem UML

- Documentação formal e eficaz;
  - http://java.sun.com/developer/technicalArticles/J2EE/patterns/References.html
- Linguagem comum para analistas, arquitetos, programadores e gerentes;
- Facilitador em todos os processos;

MODELE antes de construir!

Melhores práticas

Codificação Java

- Seguir o padrão de codificação JavaSoft
  http://java.sun.com/docs/codeconv/
- Seguir a plataforma J2SE/J2EE adequada:
  - J2EE 1.3 – J2SE 1.4
  - J2EE 1.4 – J2SE 1.5
  - JEE 5.0 – Java SE 6.0
- Não usar métodos ***deprecated***;
- Não usar Legacy Collections (Vector e Hashtable);

Melhores práticas

Codificação Java

- Documentar o código usando JavaDoc; http://java.sun.com/j2se/javadoc/
- Logar SEMPRE, e adequadamente todas as exceções capturadas;
- Manter dados no HttpSession somente se necessário! *(quando?)*
- Escolher as Collections de acordo com seu uso! *(Ref. Core Java Vol. II)*

Melhores práticas

Codificação Java

- Evitar acessar o JNDI a cada transação:
  - Usar sempre um **ServiceLocator** com cache para referências remotas de objetos EJB ou recursos Java EE.

Quer ter performance?
USE o ServiceLocator!

Melhores práticas

Codificação Java

- Envio de e-mail:
  - síncrono: usar o connector de JavaMail da J2EE!
    - javax.mail.Session
  - assíncrono: combinar JMS com JavaMail (J2EE 1.3>)
  - JAMAIS CONECTAR DIRETAMENTE COM O SMTP;

Quer ter performance?
USE o Connector JavaMail!

Melhores práticas

Codificação Java

- JDBC: Sempre fechar:
  - Statement e ResulSet, ao fim das transações SQL;
  - Connection ao fim da transação Web/EJB;
  - Usar adequadamente Commit e Rollback
- Durante uma transação Web ou EJB, DEVEMOS usar somente uma Connection;

USE o DAO Pattern!

Melhores práticas

Codificação Java

- JDBC: Evitar executar comandos SQL com Statement;
  - executar comandos SQL via PreparedStatement;
    - http://java.sun.com/j2se/1.3/docs/api/java/sql/PreparedStatement.html
  - executar StoredProcedures via CallableStatement;
    - http://java.sun.com/j2se/1.3/docs/api/java/sql/CallableStatement.html
  - em ambos os casos usar Bind de Parâmetros usando os setters;

Quer ter performance?
USE Bind de Parâmetros!

Melhores práticas

Codificação Java

- JDBC: Evitar ler ResulSets aninhados;
  - substitua por SQL Queries usando JOIN;

```
SQL> SELECT c.course_name, c.period, e.student_name
  2  FROM course c,
  3  JOIN enrollment e ON ( c.course_name =
e.course_name AND c.period = e.period )
  4  WHERE c.course_number = '10';
```

Economize recursos no acesso a dados usando JOINs!

Melhores práticas

Especificações

- Seguir a RISCA as especificações de codificação da JavaSoft:
  - *"funciona no JBOSS mas não funciona no Sun ONE"* (pensamento do desenvolvedor)
  - J2EE 1.3 - http://java.sun.com/j2ee/1.3/docs/index.html
  - J2EE 1.4 - http://java.sun.com/j2ee/1.4/docs/index.html
  - Java EE 5.0 - http://java.sun.com/javaee/

OBEDEÇA as especificações!

Melhores práticas

Especificações

- Validar os frameworks escolhidos com a especificação e produtos usados:
  - Struts, Hibernate, Avalon, Xalan, FOP, etc.

Verifique se o framework é suportado!

Melhores práticas

Especificações

- Sun Application Server:
  - 7.0 > J2SE 1.4, J2EE 1.3;
  - 8.0 > J2SE 1.5, J2EE 1.4;
  - 9.0 > Java SE 6.0, Java EE 5.0
- AVK – Java Application Verification Kit
  - Verifica a porcentagem de aderência da aplicação ás especificações da JavaSoft
  - Indica as falhas permitindo as correções necessárias
    - http://java.sun.com/j2ee/avk/

Melhores práticas

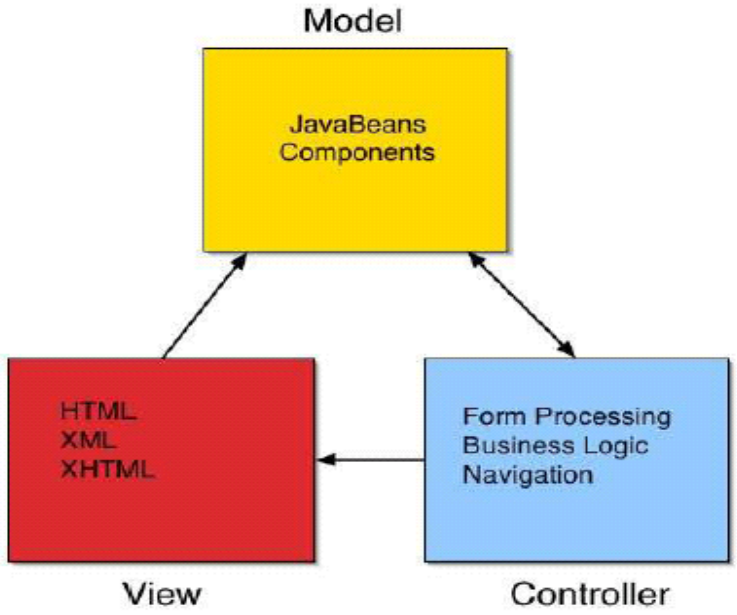
Arquitetura

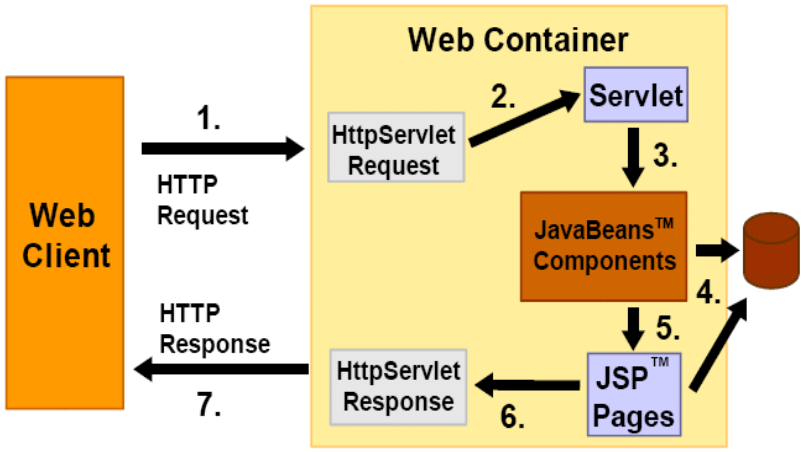
- 80% dos sistemas falham em:
  - prazo, custo e qualidade;
  - performance, segurança e gerenciamento
- A arquitetura da aplicação deve garantir as qualidades sistêmicas:
  - performance, segurança e gerenciamento;
- A habilidade da equipe deve garantir:
  - prazo, custo e qualidade;

Melhores práticas

Arquitetura

- Usando e aplicando o MVC

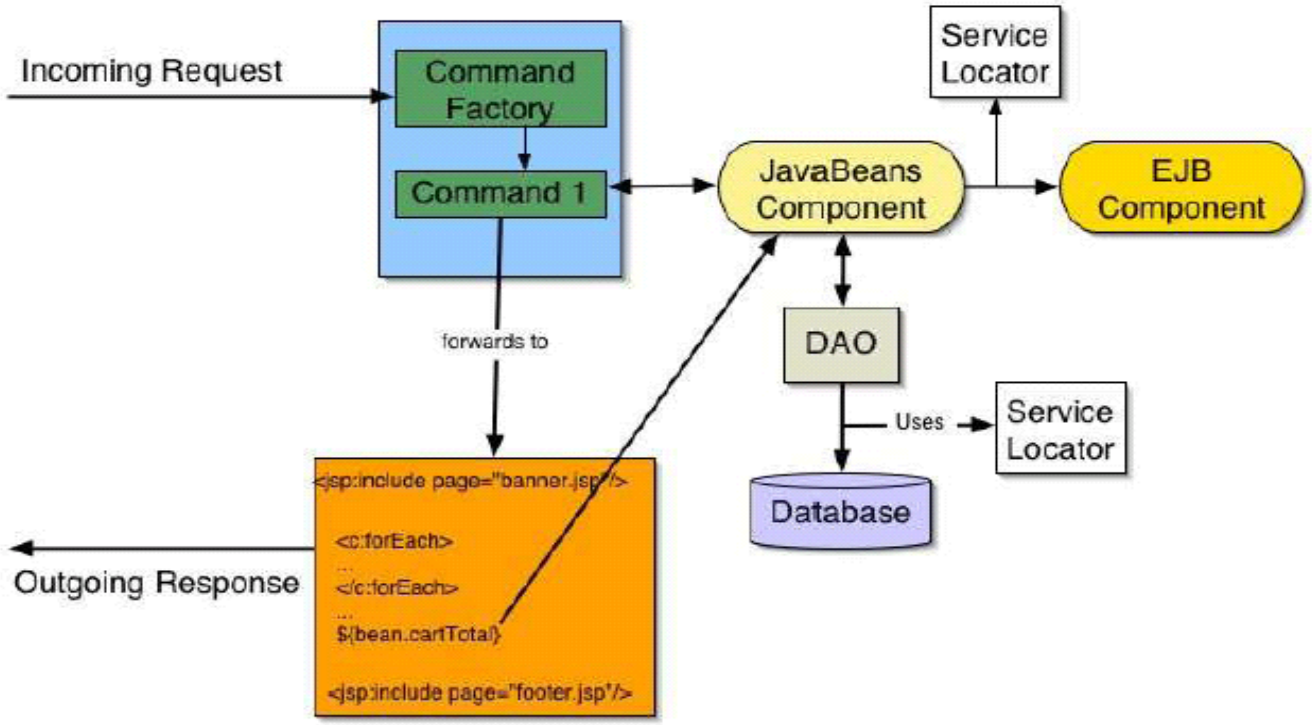




Melhores práticas

Arquitetura

- Modelo básico de aplicações J2EE



Melhores práticas

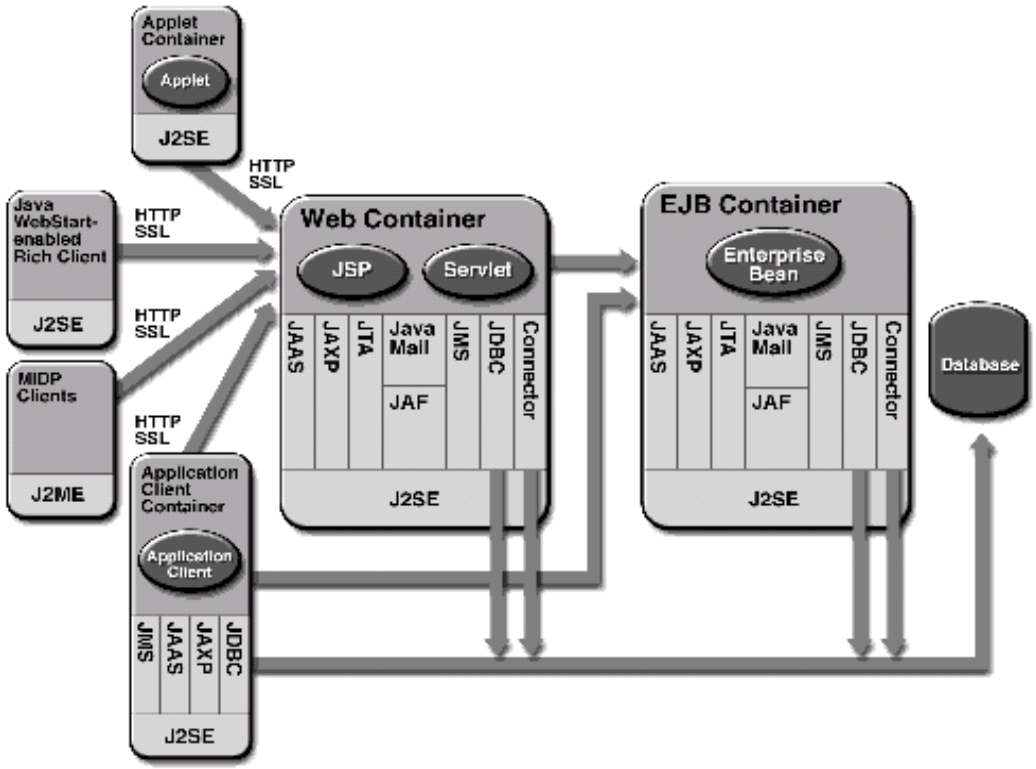
Arquitetura

- Arquitetos de Software são vitais no processo de desenvolvimento, pois:
  - sistemas distribuídos são complexos;
  - muitas tecnologias envolvidas e integradas;
  - modelos orientados á objetos;
  - falta de domínio do todo pelos envolvidos;
  - disseminador de conhecimento;
  - validador do uso das boas práticas;

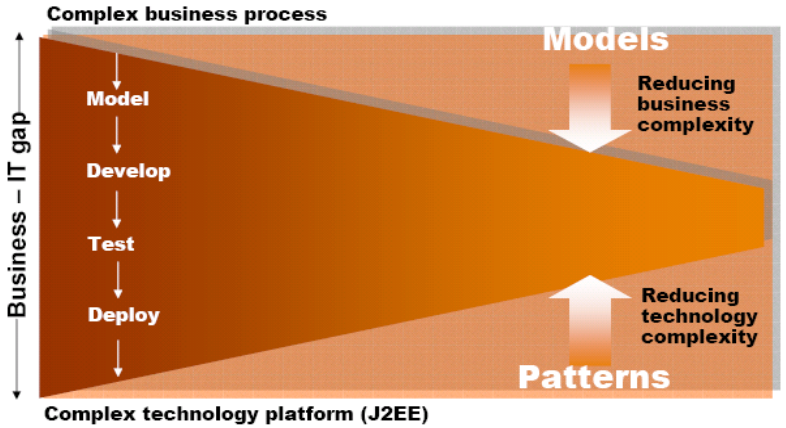
Melhores práticas

Arquitetura

- Plataforma Java:
  - muitas tecnologias envolvidas e integradas;



Melhores práticas



Testes

- Cada desenvolvedor:
  - testes unitários por componente;

    validar o código;
- Cada equipe:
  - testes integrados por módulos;

    validar a integração;
- Homologador:
  - testes integrados funcionais completos;

    validar o negócio;

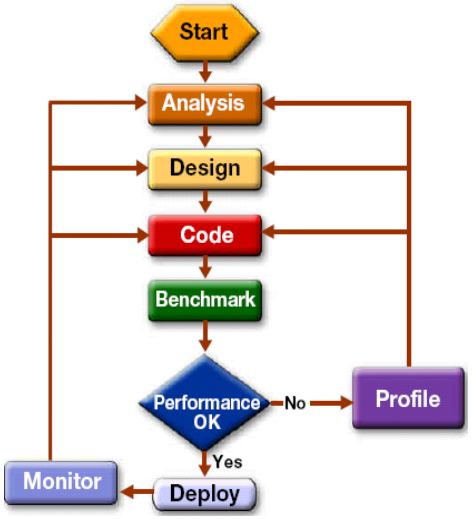
Melhores práticas

Testes

- Execução de testes:
  - de carga para identificar gargalos;
  - de stress para identificar os níveis de qualidade;
- Ferramentas:
  - JUNIT e CACTUS, Compuware, JProbe, etc.

TESTE tudo e sempre!

Melhores práticas

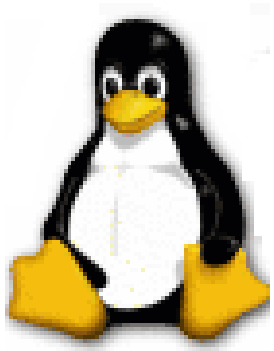


Testes

- Controle de BUGs:
  - usar uma ferramenta de Bug Parade;
  - determinar baselines de entrega;
- Ferramentas:
  - FindBugs, JLint, PMD, CheckStyle, etc.

TESTE tudo e sempre!

Serviços Profissionais



Consultoria e Serviços?

Melhores práticas no desenvolvimento Java

Oziel Moreira Neto
blog.oziel.com.br
oziel@oziel.com.br