



## Desenvolvimento um aplicativo completo usando JSF, EJB e padrões

Rodrigo Cândido e Vinny Nunes  
Instrutores Globalcode / VOffice

Globalcode - OpenEducation

## Agenda

- Objetivo / Motivação
- Domínio, Ferramentas e Opções de Arquitetura
- OA1
  - Java Persistence API (JPA)
  - JavaServer Faces (JSF)
- OA2
  - Enterprise Java Beans (EJB)
- OA3
  - Injeção de Dependência (DI)
- Perguntas e Respostas

Globalcode - OpenEducation

## Objetivo

- Apresentar 3 opções modernas de arquitetura para aplicações JEE

Globalcode - OpenEducation

## Palestrantes

- Rodrigo Cândido
  - Entusiasta de tecnologias EE
  - SCJP, SCEA, SCWCD
  - AJ, AW, CJ, CW
  
- Vinny Nunes
  - Pós OO, Ciências da Computação e Técnico em Processamento
  - Java para web desde 1999
  - Entusiasta de tecnologias de apresentação
  - SCJP, SCJA, SCWCD
  - AJ, AW, CJ, CW

Globalcode - OpenEducation

## Motivação

- Simplificar o desenvolvimento:
  - Usando menos padrões de projetos
  - Usando os recursos do Java 5, em especial anotações e genéricos
  - Usando as especificações JCP nas suas versões mais recentes
  - Usando novos padrões como IoC e DI
- Encorajar o uso das tecnologias EE
  - JSF
  - EJB

Globalcode – OpenEducation

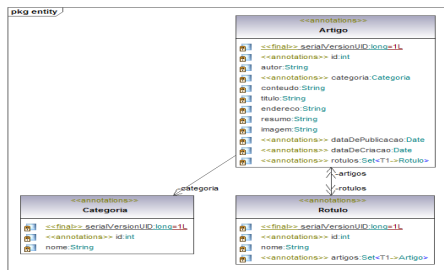
## Enquete: tecnologias usadas em produção?

- Apresentação
  - Quem utiliza JSF em produção?
  - Quem pretende usar?
  - Quem utiliza Facelets em produção?
  - Richfaces? IceFaces? Outro? Qual?
  - Struts? Flex? JavaFX? SilverLight? Outro? Qual?
- Negócio e Integração
  - EJB 2 ? EJB 3 ? Spring ?
  - JPA? Com Hibernate? Com TopLink? Outro? Qual?
  - JBoss? Glassfish? Tomcat? Jetty? WebSphere? Outros?

Globalcode – OpenEducation

## Domínio: Quick Post

- Artigo (id, título, autor, conteúdo, publicação, criação)
- Categoria (id, nome)
- Rótulo (id, nome)



Generated by UModel

www.altova.com

Globalcode – OpenEducation

## Ferramentas utilizadas



- eclipse 3.x – CPL, EPL (<http://www.eclipse.org/>)
- + jboss tools 3.x (<http://www.jboss.org/tools>)



- netbeans 6.x – CDDL, GPL (<http://www.netbeans.org/>)
- + Facelets support (<https://nfaceletssupport.dev.java.net/>)



- mysql 5.x - GPL (<http://www.mysql.com>)
- + mysql connector/j (<http://dev.mysql.com/downloads/#connector-j>)
- + mysql tools (<http://dev.mysql.com/downloads/qui-tools/5.0.html>)

Globalcode – OpenEducation

## Opções de Arquitetura JEE

#	JSF++	JPA	Padrões	EJB	DI/IoC	Containers
OA1	. 1.2 . Facelets . Richfaces	. 1.0 . Hibernate	. DAO . Factory . Service Locator . Business Object	Não	Não	Tomcat Jetty ...
OA2	. 1.2 . Facelets . Richfaces	. 1.0 . Hibernate	. Service Locator	3.0	Não	JBoss Glassfish WebSphere ...
OA3	. 1.2 . Facelets . Richfaces	. 1.0 . Hibernate	-	3.0	Sim	JBoss Glassfish WebSphere ...

Globalcode - OpenEducation

## Agenda

- Objetivo / Motivação
- Estudo de Caso
- **Java Persistence API (JPA)**
- JavaServer Faces (JSF)
- Enterprise Java Beans (EJB)
- Injeção de Dependência (DI)
- Perguntas e Respostas

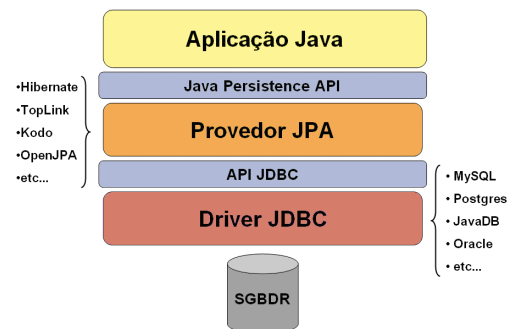
Globalcode - OpenEducation

## Java Persistence API

- Primeira especificação (padrão) de mapeamento de objetos Java para base de dados relacional
  - Integração com outros frameworks
    - Hibernate é uma implementação de JPA
    - TopLink Essenciais é uma implementação de JPA (RI)
- POJO (*Plain Old Java Objects*) → Entidades
  - Mapeamento via metadados (annotation Java 5)
- Pode ser utilizado fora de containeres EJBs (*lighweight container*)
- Uso de injeção de dependência ao invés de lookups JNDI

Globalcode - OpenEducation

## Java Persistence API



Globalcode - OpenEducation

## Java Persistence API

### Entidades JPA

```
@Entity
@Table(name = "artigo")
@NamedQuery({ @NamedQuery(name = "Artigo.findAll", quer
public class Artigo implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @ManyToOne(optional = false, fetch = FetchType.EAGER)
    @JoinColumn(name = "categoria")
    private Categoria categoria;

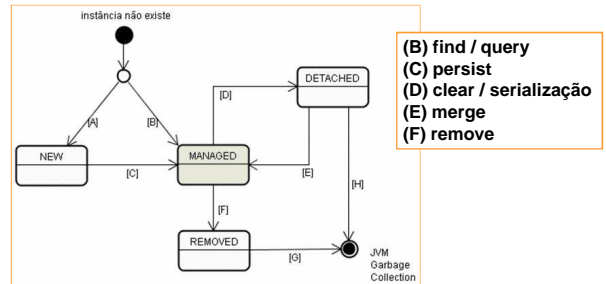
    private String autor;
    private String conteudo;
    private String titulo;
    private String endereco;
    private String resumo;
    private String imagem;
```

Globalcode - OpenEducation

## Java Persistence API

### EntityManager (javax.persistence)

- Gerencia o ciclo de vida das entidades



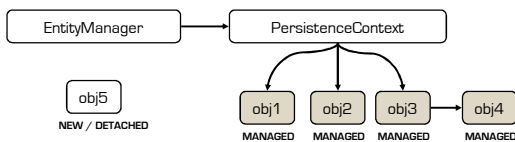
(B) find / query  
(C) persist  
(D) clear / serialização  
(E) merge  
(F) remove

Globalcode - OpenEducation

## Java Persistence API

### PersistenceContext

- Conjunto de objetos "managed" associados a um EntityManager



Globalcode - OpenEducation

## Java Persistence API

### Persistence Unit

- Uma unidade de persistência define um contexto de funcionamento para os EntityManagers

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0" xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.
<persistence-unit name="mc31p0" transaction-type="RESOURCE_LOCAL">
  <provider>org.hibernate.ejb.HibernatePersistence</provider>
  <class>business.entity.Rotulo</class>
  <class>business.entity.Categoria</class>
  <class>business.entity.Artigo</class>
  <exclude-unlisted-classes>true</exclude-unlisted-classes>
  <properties>
    <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect"/>
    <property name="hibernate.connection.driver_class" value="com.mysql.jdbc.Driver"/>
    <property name="hibernate.connection.uri" value="jdbc:mysql://localhost/mc31"/>
    <property name="hibernate.connection.username" value="root"/>
    <property name="hibernate.connection.password" value="root"/>
    <property name="hibernate.show_sql" value="true"/>
  </properties>
</persistence-unit>
</persistence>
```

Globalcode - OpenEducation

## Java Persistence API

### Exemplo de utilização de JPA

```
public void remove(final Serializable $serializable) {
    final EntityManager entityManager = this.getJpaHelper()
        .getEntityManager();
    final EntityTransaction transaction = entityManager.getTransaction();
    transaction.begin();
    entityManager.remove($serializable);
    transaction.commit();
    entityManager.close();
}

public void save(final Serializable $serializable) throws Exception {
    final EntityManager entityManager = this.getJpaHelper()
        .getEntityManager();
    final EntityTransaction transaction = entityManager.getTransaction();
    transaction.begin();
    entityManager.persist($serializable);
    transaction.commit();
    entityManager.close();
}

public void setJpaHelper(final JpaHelper $jpaHelper) {
    this.jpaHelper = $jpaHelper;
}
```

Globalcode - OpenEducation


## JPA: instalação e configuração

### bibliotecas

- persistence-api-xx.jar
- Bibliotecas do provider
  - Hibernate
  - TopLink
  - OpenJPA

Globalcode - OpenEducation

## Java Persistence API: demonstração

› eclipse 3.x 

› netbeans 6.x  *opcional*

Globalcode - OpenEducation

## Agenda

- › Objetivo / Motivação
- › Estudo de Caso
- › Java Persistence API (JPA)
- › **JavaServer Faces (JSF)**
- › Enterprise Java Beans (EJB)
- › Injeção de Dependência (DI)
- › Perguntas e Respostas

Globalcode - OpenEducation

## JSF: o que é?

- Java Server Faces
- Framework MVC proposto em 2001
- JSR 127 (1.0 e 1.1, 2001), JSR 252 (1.2, 2004), JSR 314 (2.0, 2007)
- Padronizou o desenvolvimento de interface de usuário web
- Definido por grupo de especialistas composto de empresas e pessoas físicas
- Suportado pelos containers homologados para JEE 5 ou superior
- Baseado na tecnologia Servlets
- Ed Burns, Roger Kitain - Sun

<http://java.sun.com/javaee/javaserverfaces/>

Globalcode - OpenEducation

## JSF: por que usar?

- Componentes ao invés de ações
- Renderizações além do HTML
- Conversores, validadores, eventos
- Ciclo de vida e árvore de componentes
- Binding e Expression Language
- Implementa os padrões pra você: Wrapper, Front Controller, MVC, DI, ...
- Resources, Ajax, MegaListeners, Anotações, Composite components, Bean validation, Bookmarking, new scopes, new tags, ... (2.x)

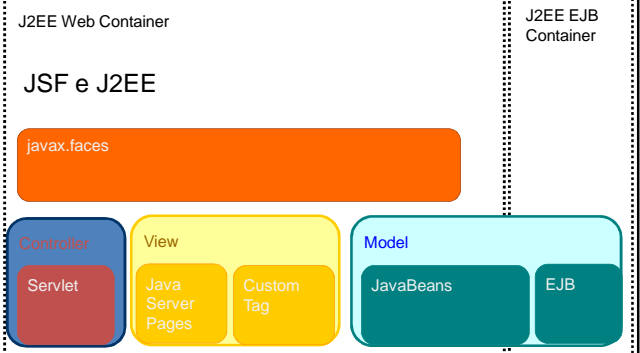
Globalcode - OpenEducation

## JSF: por que usar?

- Maduro: desde 2001! Já tem a 2.0 desde julho!
- Incorpora boas idéias de outros frameworks e de extensões JSF: Tapestry, Struts, WebWork, Richfaces, IceFaces
- Preparado para ser entendido e estendido
- Preparado para ferramentas de desenvolvimento

Globalcode - OpenEducation

## JSF: principais componentes



Globalcode - OpenEducation

## JSF: principais componentes

**Página JSF**

```
<f:view>
<h:form>
<h:outputText value="e-mail"/>
<h:inputText value="#{olaMundoMB.email}" />
<h:commandLink value="Chamar metodo do MB"
action="#{olaMundoMB.metodo}" />
</h:form>
</f:view>
```

**faces-config.xml**

```
<managed-bean>
<managed-bean-name>
olaMundoMB
</managed-bean-name>
<managed-bean-class>
..olaMundoMB
</managed-bean-class>
<managed-bean-scope>
request
</managed-bean-scope>
</managed-bean>
```

**Managed Bean**

```
public class OlaMundoMB {
private String email;
public void metodo() {
System.out.println("Chamada ao MB...");
System.out.println("Email lido:" + email);
}
public String getEmail() {
return email;
}
public void setEmail(String email) {
this.email = email;
}
}
```

## JSF: instalação e configuração

### ➤ bibliotecas

### JARs necessários:

- jsf-api.jar
- jsf-ri.jar
- jstl.jar
- standard.jar
- commons-beanutils.jar
- commons-digester.jar
- commons-collections.jar
- commons-logging.jar

## JSF: instalação e configuração

### ➤ arquivos de configuração

#### pom.xml:

```
...
<dependency>
<groupId>javax.faces</groupId>
<artifactId>jsf-api</artifactId>
<version>${jsf.version}</version>
</dependency>
<dependency>
<groupId>javax.faces</groupId>
<artifactId>jsf-impl</artifactId>
<version>${jsf.version}</version>
</dependency>
...
```

## JSF: instalação e configuração

### ➤ arquivos de configuração

#### web.xml:

```
<servlet>
<servlet-name>Faces Servlet</servlet-name>
<servlet-
class>javax.faces.webapp.FacesServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>Faces Servlet</servlet-name>
<url-pattern>*.jsf</url-pattern>
</servlet-mapping>
```

## JSF: instalação e configuração

### › arquivos de configuração

faces-config.xml:

```
<?xml version="1.0"?>
<faces-config version="1.2"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javae
e http://java.sun.com/xml/ns/javaee/web-
facesconfig_1_2.xsd">
</faces-config>
```

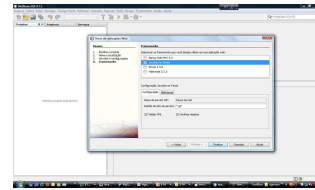
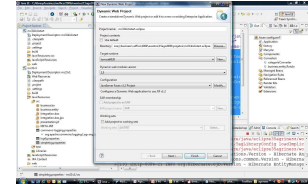
Globalcode - OpenEducation

## JSF: instalação e configuração

### › eclipse 3.x




### › netbeans 6.x



Globalcode - OpenEducation

## JSF: exemplos

### › Lista de artigos publicados

eclipse 3.x 

Globalcode - OpenEducation

## Facelets: o que é?

- › *JSF View Definition Framework*
- › Proposto em 2005
- › Tecnologia focada na criação de árvores de componente JSF
- › VDL alternativa
- › Subprojeto do JSF
- › Jacob Hookom - Sun

<https://facelets.dev.java.net/>

Globalcode - OpenEducation

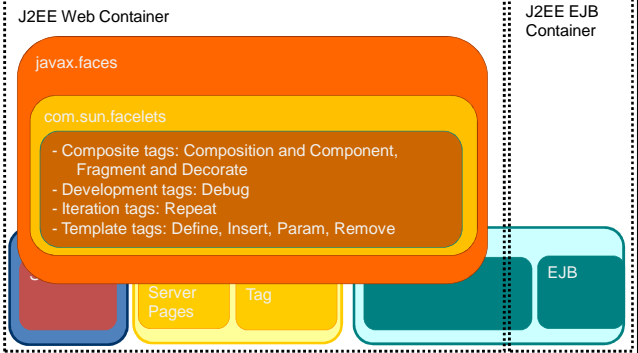


## Facelets: por que usar?

- › Templates
- › XHTML ao invés de JSP
- › Atributo jsfc
- › Rápido para renderizar
- › Melhor para identificar expressões EL erradas
- › Também é padrão (JSR 314, 2.x)
- › Turbinado pela biblioteca composite (2.x)

Globalcode - OpenEducation

## Facelets: principais componentes



Globalcode - OpenEducation

## Facelets: principais componentes

```
1<?xml version="1.0" encoding="ISO-8859-1"?>
2<html xmlns="http://www.w3.org/1999/xhtml"
3  xmlns:h="http://java.sun.com/jsf/html"
4  xmlns:f="http://java.sun.com/jsf/core"
5  xmlns:ui="http://java.sun.com/jsf/facelets"
6  xmlns:rich="http://richfaces.org/rich"
7  xmlns:a4j="http://richfaces.org/a4j">
8
9<ui:composition template="/resources/template/template.xhtml">
10  <ui:define name="body">
11    <h:form id="frm">
12      <rich:panel>
13        <f:facet name="header">
14          <h:outputText value="Novo artigo" />
15        </f:facet>
16        <h:panelGrid columns="3" styleClass="form">
17          <h:outputLabel value="Titulo:" for="itTitulo" />
18          <h:inputText id="itTitulo" value="#{artigosMB
19            size="30" required="true" label="Titulo"
20          <rich:message for="itTitulo"/></rich:message>
21
22          <h:outputLabel value="Autor:" for="itAutor" />
23          <h:inputText id="itAutor" value="#{artigosMB
24            size="30" required="true" label="Autor" />
25          <rich:message for="itAutor"/></rich:message>
26
```

cliente

Globalcode - OpenEducation

## Facelets: principais componentes

```
1<?xml version="1.0" encoding="ISO-8859-1"?>
2<ui:composition xmlns="http://www.w3.org/1999/xhtml"
3  xmlns:h="http://java.sun.com/jsf/html"
4  xmlns:f="http://java.sun.com/jsf/core"
5  xmlns:ui="http://java.sun.com/jsf/facelets"
6  xmlns:rich="http://richfaces.org/rich"
7  xmlns:a4j="http://richfaces.org/a4j">
8  <html>
9  <head>
10   <link rel="stylesheet" type="text/css" href="/resources/css/mc31.css" />
11   <script type="text/javascript" src="/resources/js/mc31.js"></script>
12 </head>
13 <body>
14   <ui:insert name="header">
15     <div class="divHeader"></div>
16   </ui:insert>
17   <div class="divBody"><rich:messages styleClass="msgs"
18     ajaxRendered="true">
19 </rich:messages> <ui:insert name="body">
20     corpo
21   </ui:insert></div>
22   <ui:insert name="footer">
23     <div class="divFooter"></div>
24   </ui:insert>
25 </body>
```

template

Globalcode - OpenEducation

## Facelets: instalação e configuração

- jsf-facelets.jar (1.x)
- jsf-api.jar
- jsf-ri.jar
- jstl.jar
- standard.jar
- commons-beansutils.jar
- commons-digester.jar
- commons-collections.jar
- commons-logging.jar

Globalcode - OpenEducation

## Facelets: instalação e configuração

### › arquivos de configuração

#### pom.xml:

```
...
<dependency>
  <groupId>com.sun.facelets</groupId>
  <artifactId>jsf-facelets</artifactId>
  <version>${facelets.version}</version>
</dependency>
...
```

Globalcode - OpenEducation

## Facelets: instalação e configuração

### › arquivos de configuração

#### web.xml:

```
...
<context-param>
  <param-name>javax.faces.DEFAULT_SUFFIX</param-name>
  <param-value>.xhtml</param-value>
</context-param>
<context-param>
  <param-name>facelets.REFRESH_PERIOD</param-name>
  <param-value>2</param-value>
</context-param>
...
```

Globalcode - OpenEducation

## Facelets: instalação e configuração


### › arquivos de configuração


#### faces-config.xml:

```
<?xml version="1.0"?>
<faces-config version="1.2"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-facesconfig_1_2.xsd">
  <application>
    <view-handler>com.sun.facelets.FaceletViewHandler</view-handler>
  </application>
</faces-config>
```

Globalcode - OpenEducation


## Facelets: instalação e configuração

➤ eclipse 3.x 

➤ netbeans 6.x 

Globalcode – OpenEducation

## Facelets: exemplos

eclipse 3.x 

➤ Lista de artigos publicados com html reaproveitado e template

Globalcode – OpenEducation

## Richfaces: o que é?

- Biblioteca de componentes
- Criada pela Exadel em 2006
- Mantida pela Jboss desde 2007
- Última versão estável: 3.3.1 em 18/04/2009
- Alexander Smirnov – Exadel
- Max Katz – Exadel

<http://www.jboss.org/richfaces>

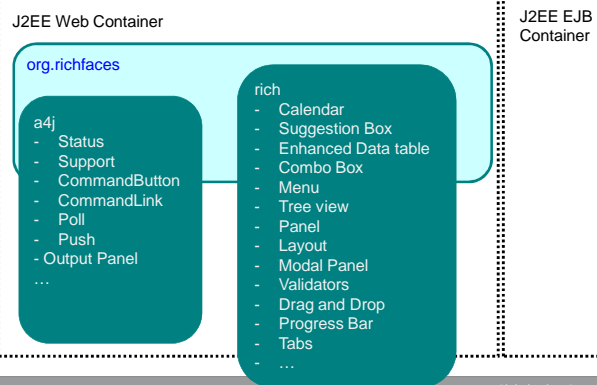
Globalcode – OpenEducation

## Richfaces: por que usar?

- Suporte a Ajax
- Aproximadamente 100 componentes prontos
- Component Development Kit
- Skins prontos e possibilidade de customização
- Resources
- Facilidades para testes
- Cross browser
- Comunidade ativa
- Integração com jQuery, Prototype e Scriptaculous

Globalcode – OpenEducation

## Richfaces: principais componentes



## Richfaces: principais componentes

```
<xmlns:a4j="http://richfaces.org/a4j">
<xmlns:rich="http://richfaces.org/rich">
```

```
<%@ taglib uri="http://richfaces.org/a4j" prefix="a4j"%>
<%@ taglib uri="http://richfaces.org/rich" prefix="rich"%>
```

Cliente

## Richfaces: instalação e configuração

### ► bibliotecas

- `richfaces-api-xx.GA.jar`
- `richfaces-impl-xx.GA.jar`
- `richfaces-ui-xx.GA.jar`
- `jsf-api.jar`
- `jsf-ri.jar`
- `jstl.jar`
- `standard.jar`
- `commons-beansutils.jar`
- `commons-digester.jar`
- `commons-collections.jar`
- `commons-logging.jar`

## Richfaces: instalação e configuração

### ► arquivos de configuração – pom.xml

```
...
<dependency>
<groupId>org.richfaces.framework</groupId>
<artifactId>richfaces-api</artifactId>
<version>${richfaces.version}</version>
</dependency>
<dependency>
<groupId>org.richfaces.framework</groupId>
<artifactId>richfaces-impl</artifactId>
<version>${richfaces.version}</version>
</dependency>
<dependency>
<groupId>org.richfaces.ui</groupId>
<artifactId>richfaces-ui</artifactId>
<version>${richfaces.version}</version>
</dependency>
```

## Richfaces: instalação e configuração

### › arquivos de configuração – web.xml

```
...
<filter>
<display-name>Ajax4jsf Filter</display-name>
<filter-name>ajax4jsf</filter-name>
<filter-class>org.ajax4jsf.Filter</filter-class>
</filter>
<filter-mapping>
<filter-name>ajax4jsf</filter-name>
<servlet-name>Faces Servlet</servlet-name>
<dispatcher>REQUEST</dispatcher>
<dispatcher>FORWARD</dispatcher>
<dispatcher>INCLUDE</dispatcher>
</filter-mapping>
...
```

Globalcode – OpenEducation

## Richfaces: instalação e configuração


### › arquivos de configuração – faces-config.xml

```
<?xml version="1.0"?>
<faces-config version="1.2"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-
  facesconfig_1_2.xsd">

</faces-config>
```

Globalcode – OpenEducation


## Richfaces: exemplos


eclipse 3.x 

- › Lista de artigos publicados com data e categoria. Filtro, ordenação, panel.

Globalcode – OpenEducation

## Richfaces: instalação e configuração


› eclipse 3.x 

› netbeans 6.x 

Globalcode – OpenEducation

## OA1

### › Demonstração

eclipse 3.x 

#	JSF++	JPA	Padrões	EJB	DI/IoC	Containers
OA1	. 1.2 . Facelets . Richfaces	. 1.0 . Hibernate	. DAO . Factory . Service Locator . Business Object	Não	Não	Tomcat Jetty ...
OA2	. 1.2 . Facelets . Richfaces	. 1.0 . Hibernate	. Service Locator	3.0	Não	JBoss Glassfish WebSphere ...
OA3	. 1.2 . Facelets . Richfaces	. 1.0 . Hibernate	-	3.0	Sim	JBoss Glassfish WebSphere ...

Globalcode – OpenEducation

## OA1

### › Vantagens

- › Basta um servidor EE com suporte a servlets
- › Mais rápido para codificar e para testar
- › Mais fácil para aprender

### › Desvantagens

- › Acesso de clientes desktop não é tão prático
- › Arquivo de expedição monolítico (.war)
- › Poucas facilidades de integração com outras tecnologias EE

Globalcode – OpenEducation

## Agenda

- › Objetivo / Motivação
- › Estudo de Caso
- › Java Persistence API (JPA)
- › JavaServer Faces (JSF)
- › **Enterprise Java Beans (EJB)**
- › Injeção de Dependência (DI)
- › Perguntas e Respostas

Globalcode – OpenEducation

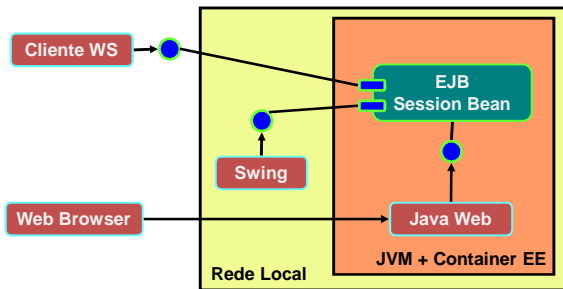
## Enterprise Java Beans (EJB)

- › Plataforma para criação de aplicações de negócio portáteis, robustas e reutilizáveis usando a linguagem de programação Java;
- › Componente para construir aplicações sem precisar reinventar uma série de recursos e serviços (transações, segurança, persistência);
- › Modelo de alto nível para objetos distribuídos (computação distribuída);
- › Pode ser utilizado em soluções Web e não Web;

Globalcode – OpenEducation

## Enterprise Java Beans (EJB)

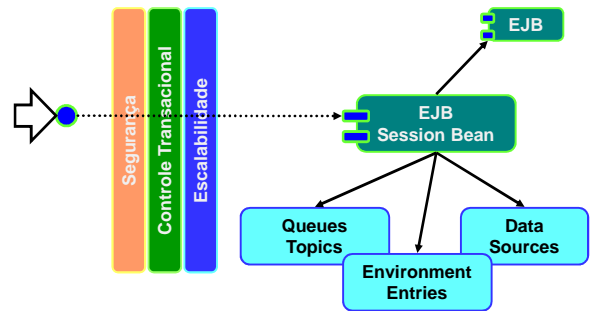
### Componentes Negócio Compartilhados



Globalcode - OpenEducation

## Enterprise Java Beans (EJB)

### Infra Estrutura do Container

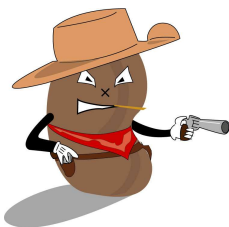


Globalcode - OpenEducation

## Enterprise Java Beans (EJB)

### Tipos de EJB:

- Session Bean
  - Representa um processo de negócio
- Message-Driven Bean
  - Trabalha de forma assíncrona
- Entity Bean
  - Representa um objeto persistente
  - Definido pela especificação JPA



Globalcode - OpenEducation

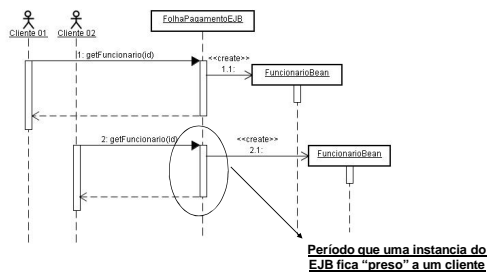
## Session Bean

- Representa um processo de negócio (regra de negócio);
- Pode ser do tipo *Stateless* ou *Stateful*;
- Stateless*
  - Não mantém estado de conversão com o cliente;
  - Não tem compromisso de manter uma sessão;
  - Não devemos utilizar atributos de negócio, somente atributos técnicos em cache;
  - É um EJB econômico;
- Stateful*
  - Mantém estado de conversão com o cliente;
  - Cada cliente terá uma instancia do EJB exclusiva, desde a criação até a remoção;
  - Podemos ter atributos de negócio com métodos getters e setters;

Globalcode - OpenEducation

## Session Bean

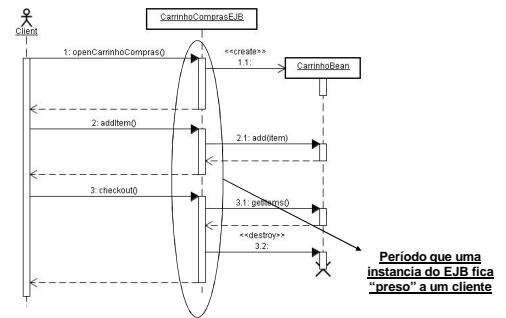
### › Stateless (sem estado)



Globalcode – OpenEducation

## Session Bean

### › Stateful (com estado)



Globalcode – OpenEducation

## Session Bean

### › Stateless (sem estado)

```
@Stateless
public class FolhaPagamentoEJB implements InterfaceName {

    public FuncionarioBean getFuncionario(Long id) {
        FuncionarioBean funcionario = new FuncionarioBean();
        // TODO Retrieve Data From the Database
        return funcionario;
    }
}
```

Globalcode – OpenEducation

## Session Bean

- › Clientes acessam o Session Bean através de uma interface Java simples (POJI);
- › Métodos (serviços) são publicados através desta interface;
- › A interface deve ser implementada pela classe do Bean;
- › Tipos de Interfaces
  - › Local (@Local)
    - › Somente acesso local (mesma JVM);
    - › Cliente pode ser um componente Web ou outro EJB;
    - › Localização não é transparente (não suporta clusterização).
  - › Remota (@Remote)
    - › Suporta acesso remoto (fora da JVM);
    - › Cliente pode ser um componente Web, outro EJB ou aplicação desktop;
    - › Transparência de localização (suporta clusterização).

Globalcode – OpenEducation



## Session Bean

### Local Interface

```
@Local
public interface InterfaceName {
    public void businessMethod();
}
```

### Remote Interface

```
@Remote
public interface InterfaceName {
    public void businessMethod();
}
```

Globalcode - OpenEducation

## OA2

### Demonstração

eclipse 3.x



#	JSF++	JPA	Padrões	EJB	DI/IoC	Containers
OA1	. 1.2 . Facelets . Richfaces	. 1.0 . Hibernate	. DAO . Factory . Service Locator . Business Object	Não	Não	Tomcat Jetty ...
OA2	. 1.2 . Facelets . Richfaces	. 1.0 . Hibernate	. Service Locator	3.0	Não	JBoss Glassfish WebSphere ...
OA3	. 1.2 . Facelets . Richfaces	. 1.0 . Hibernate	-	3.0	Sim	JBoss Glassfish WebSphere ...

Globalcode - OpenEducation

## OA2

### Vantagens

- Arquitetura mais robusta (melhor escalabilidade)
- Infra-estrutura do servidor EE
- Regras de negócio centralizadas
- Ampla diversidade de clientes (CORBA, RMI, etc)

### Desvantagens

- Maior custo operacional
- Maior complexidade na implementação
- Maior dificuldade de suporte e manutenção

Globalcode - OpenEducation

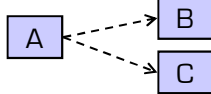
## Agenda

- Objetivo / Motivação
- Estudo de Caso
- Java Persistence API (JPA)
- JavaServer Faces (JSF)
- Enterprise Java Beans (EJB)
- Injeção de Dependência (DI)**
- Perguntas e Respostas

Globalcode - OpenEducation

## Injeção de Dependência

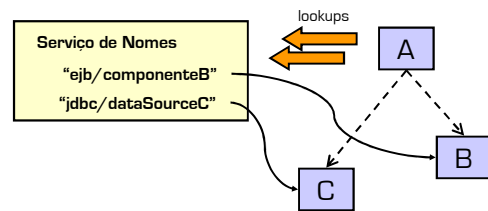
- Mecanismo de “injeção” de recursos necessários a um componente de forma automática
  - Sem a presença de instruções explícitas para acessar o recurso
- Exemplo:
  - O componente de negócios A depende do componente de negócios B e do pool de conexões C.



Globalcode – OpenEducation

## Injeção de Dependência

- Sem injeção de dependência:
  - Código fonte do componente A contém instruções para “buscar” o componente B e o recurso C.



Globalcode – OpenEducation

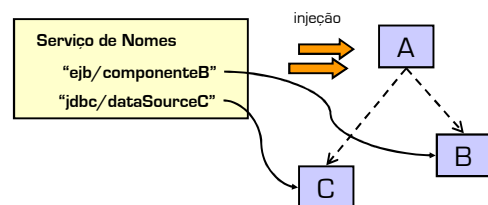
## Injeção de Dependência

- Com injeção de dependência
  - Código do componente A não busca recursos e outros componentes
  - Devemos declarar as dependências de componentes e recursos nos atributos ou métodos setters do componente A
  - Container (Java EE) interpreta as dependências declaradas no componente A e “injeta” os recursos e componentes necessários

Globalcode – OpenEducation

## Injeção de Dependência

```
class ComponenteA {  
    @EJB(name="ejb/componenteB") InterfaceB b;  
    @Resource(name="jdbc/dataSourceC") DataSource c;  
}
```



Globalcode – OpenEducation

## Injeção de Dependência

- › Quem pode receber injeção ?
  - › EJB, Interceptor
  - › Servlet, Filter, \*Listener, JSP Tag Handler
  - › JSF Managed Beans
  - › Endpoints JAX-WS e Handlers
- › O que pode ser injetado ?
  - › Referências a EJBs e a Web Services
  - › DataSource JDBC
  - › Destination e ConnectionFactory JMS
  - › UserTransaction
  - › EntityManager (JPA)

Globalcode - OpenEducation

## Injeção de Dependência

### › Exemplo de Utilização

```

@Stateless
public class SessionBean implements InterfaceA {

    @EJB
    private InterfaceB ejbB;

    @Resource(name="jdbc/Database")
    private DataSource dataSource;

    public void businessMethod() throws Exception {
        // Create New Database Connection
        Connection connection = dataSource.getConnection();
        Statement stmt = connection.createStatement();
        // Execute Another Method in ejbB
        ejbB.anotherMethod();
    }
}
    
```

Globalcode - OpenEducation

## OA3

### › Demonstração

eclipse 3.x



#	JSF++	JPA	Padrões	EJB	DI/loC	Containers
OA1	. 1.2 . Facelets . Richfaces	. 1.0 . Hibernate	. DAO . Factory . Service Locator . Business Object	Não	Não	Tomcat Jetty ...
OA2	. 1.2 . Facelets . Richfaces	. 1.0 . Hibernate	. Service Locator	3.0	Não	JBoss Glassfish WebSphere ...
OA3	. 1.2 . Facelets . Richfaces	. 1.0 . Hibernate	-	3.0	Sim	JBoss Glassfish WebSphere ...

Globalcode - OpenEducation

## OA3

### › Vantagens

- › Configuração facilitada
- › Diminuição na quantidade de padrões de projeto utilizados na aplicação
- › Maior produtividade
- › Facilidade de manutenção

### › Desvantagens

- › Suporte incompleto IoC (será corrigido no Java EE 6)
- › Incompatibilidade para escalabilidade
- › Somente para aplicações Java EE

Globalcode - OpenEducation

## Outras especificações usadas junto

- JSF
  - + Bean Validation
  - + Context and Dependency Injection
- EJB
  - + JMS
  - + JTA
  - + WS
  - + JNDI
  - + JAAS

Globalcode – OpenEducation

## Veja também ...

- JSF
  - + Scrum Toys no Netbeans 6.8
  - + SuperCRUD
  - + IceFaces
  - + MyFaces
  - + Exadel Flamingo
- JME
  - + Cosmo
  - + Floggy

Globalcode – OpenEducation

## Faça você mesmo!

- Baixar os 3 projetos demo
- Configurar no Eclipse ou no Netbeans
- Codificar CRUDS para
  - Autor
  - Rótulo
  - Categoria
- Codificar Login programático
- Usar componente de menu
- Usar componente para Drag and drop
- Usar componentes de calendário e numérico

Globalcode – OpenEducation

## Perguntas e Respostas



Globalcode – OpenEducation

Agradecidos pela presença!