

Java Web

AULA 01 – INTRODUÇÃO

1

Objetivos e Conceitos

- Objetivos:
 - Apresentar o Java EE, histórico da evolução da especificação, servidores de aplicação
- Conceitos:
 - Java EE, Servidores de Aplicação, Aplicação Web

2

Tópicos

1. Plataforma Java
2. Arquitetura Java EE/Jakarta EE
3. Servidores de Aplicação
4. Aplicação Web
5. Ferramentas

3

1 Plataforma Java

4

Plataforma Java.

Plataforma criada pela Sun Microsystems, adquirida pela Oracle

Independente de Sistema Operacional

Trabalha com o conceito de Máquinas Virtuais

Formado por:

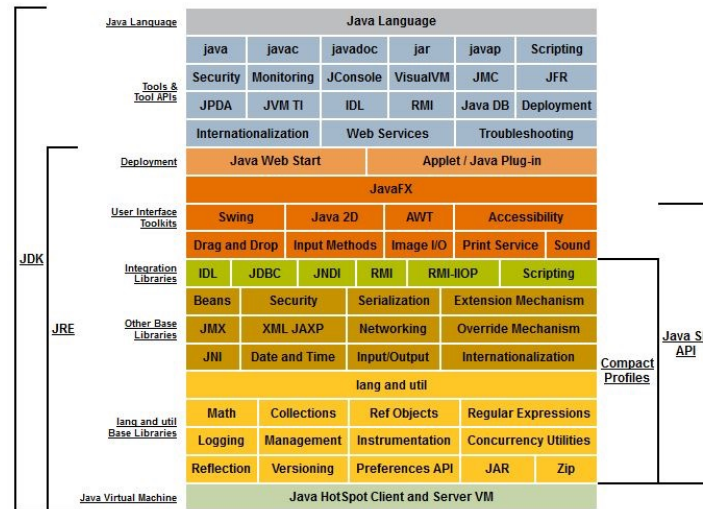
- **Java SE** : Linguagem Java, Ambiente de execução (Java Virtual Machine), Bibliotecas básicas
- **Java EE** : Plataforma para desenvolvimento de aplicações Web e Corporativas
- **Java ME** : Plataforma para desenvolvimento de aplicações embarcadas

Java SE

Basicamente 2 produtos

- **JRE : Java SE Runtime Environment**
 - JVM : Java Virtual Machine
 - Bibliotecas
 - APIs Padrão
 - I/O
 - Collections
 - JDBC
 - etc
 - Não contém ferramentas para desenvolvimento
- **JDK : Java Development Kit**
 - Contém o JRE
 - Contém ferramentas para desenvolvimento: Compilador, ferramentas e debuggers

Java SE



Fonte: <https://www.oracle.com/java/technologies/platform-glance.html>

Prof. Dr. Razer A N R Montaña

JAVA WEB

7

7

Java SE

JDK 1.0: (Oak) Jan/1996

JDK 1.1: Fev/1997

J2SE 1.2: (Playground) Dez/1998

J2SE 1.3: (Kestrel) Mai/2000

J2SE 1.4: (Merlin) Fev/2002

J2SE 5.0: (Tiger) Set/2004

Java SE 6: (Mustang) Dez/2006

Java SE 7: (Dolphin) Jul/2011

Java SE 8 - LTS: Mar/2014

Java SE 9: Set/2017

Java SE 10: Mar/2018

Java SE 11 - LTS: Set/2018

Java SE 12: Mar/2019

Java SE 13: Set/2019

Java SE 14: Mar/2020

Java SE 15: (esperado) Set/2020

Java SE 16: (esperado) Mar/2021

Java SE 17 - LTS: (esperado) Set/2021

Prof. Dr. Razer A N R Montaña

JAVA WEB

8

8

Java SE

Open JDK

- Open Source
- Licença GPL
- <http://jdk.java.net>
- Implementação de Referência – JDK 8: <http://jdk.java.net/java-se-ri/8-MR3>

Versão Comercial Oracle

- <https://www.oracle.com/java/technologies/javase-downloads.html>

Java SE..

Licenciamento

- **JDK 8 – Versão LTS (*Long-Term Support*)**
 - Término das atualizações gratuitas
 - Comercial - Final de 2019
 - Pessoal – Final de 2020
 - Haverá suporte pago pelo menos até 2025
 - Ainda pode ser usado indefinidamente sem atualização
 - Para conseguir outras atualizações, baixar de OpenJDK ou outro fabricante alternativo: IBM, Red Hat, etc
- **JDK 11 – Versão LTS (*Long-Term Support*)**
 - Binários OpenJDK da Oracle sob licença GPL
 - Binários Oracle JDK sob licença comercial paga

2 Arquitetura Java EE/Jakarta EE

Prof. Dr. Razer A N R Montano

JAVA WEB

11

11

Arquitetura Java EE.

- Java EE – Java Enterprise Edition
 - Plataforma de desenvolvimento de aplicações Web e Corporativas
 - Conjunto de tecnologias que quando coordenadas reduzem o custo e a complexidade do desenvolvimento
 - Desenvolvida pela JCP (<http://www.jcp.org/>)
 - JSR : *Java Specification Requests*
 - Especialistas, comunidade, empresas, universidades
 - Criam as JSRs para definir as tecnologias Java
- Tecnologias mais difundidas
 - Servlets/JSP
 - JSF
 - JPA
 - EJB
 - Web Services

Prof. Dr. Razer A N R Montano

JAVA WEB

12

12

Histórico de Versões.

- **JPE Project** – Mai/1998
- **J2EE 1.2** – 12/dez/1999
- **J2EE1.3** – 24/set/2001
- **J2EE 1.4** – 11/Nov/2003
- **Java EE 5** – 11/Mai/2006
- **Java EE 6** – 10/Dez/2009
- **Java EE 7** – 5/Abr/2013
- **Java EE 8** – 31/Ago/2017
- *Set/2017 – Especificação do Java EE passa da Oracle para Eclipse Foundation*
- **Jakarta EE 8** – 10/Set/2019 – Totalmente compatível com Java EE 8
- **Jakarta EE 9** – (em desenvolvimento) - Set/2020
- **Jakarta EE 10** – Em discussão

13

Tecnologias

Tutorial Java EE 7:

<https://docs.oracle.com/javaee/7/tutorial/>

Tutorial Java EE 8:

<https://javaee.github.io/tutorial/>

Especificação Jakarta EE 8:

<https://jakarta.ee/specifications/platform/9/>

14

Tecnologias: Java EE 7

- Link:
<http://www.oracle.com/technetwork/java/javaee/tech/index-jsp-142185.html>
- Tecnologias Java EE 7 – JSR 342
 - Servlet 3.1 – JSR 340
 - JSP 2.3 – JSR 245
 - EL 3.0 – JSR 341
 - JSTL 1.2 – JSR 52
 - JSF 2.2 – JSR 344
 - JAX-RS 2.0 - JSR 339
 - JPA 2.1 - JSR 338
 - JTA 1.2 – JSR 907
 - CDI 1.1 – JSR 346
 - EJB 3.2 – JSR 345

Tecnologias: Java EE 8

- Link:
<http://www.oracle.com/technetwork/java/javaee/tech/index.html>
- Tecnologias Java EE 8 – JSR 366
 - <https://www.oracle.com/java/technologies/java-ee-8.html>
 - Servlet 4.0 – JSR 369
 - JSP 2.3 – JSR 245
 - EL 3.0 – JSR 341
 - JSTL 1.2 – JSR 52
 - JSF 2.3 – JSR 372
 - JAX-RS 2.1 - JSR 370
 - JPA 2.2 - JSR 338
 - JTA 1.2 – JSR 907
 - CDI 2.0 – JSR 365
 - EJB 3.2 – JSR 345

Tecnologias: Jakarta EE 9.

- Link:
<https://jakarta.ee/specifications/platform/9/>
- Tecnologias Jakarta EE 9
 - <https://jakarta.ee/specifications/platform/9/platform-spec-9-SNAPSHOT.html#a2161>
 - Servlet 5.0
 - JSP 3.0
 - EL 4.0
 - JSTL 2.0
 - JSF 3.0
 - JAX-RS 3.0
 - JPA 3.0
 - JTA 2.0
 - CDI 3.0
 - EJB 4.0

17

Servidores.

- Java EE/Jakarta EE um conjunto de especificações
 - Indicam como as tecnologias funcionam
 - Indicam como desenvolver um software que implementa cada uma das tecnologias
 - Indicam como desenvolver os SERVIDORES
- Essas especificações são implementadas em Servidores
 - Cada fabricante tem o seu
- Assim, sua aplicação necessita de um servidor
 - A aplicação deve ser instalada neste servidor (*deploy*)
 - A aplicação se torna disponível

18

XML e Anotações.

- Antigamente – “XML Hell”
 - Muitos arquivos XML para serem configurados (*deployment descriptors*)
- Hoje
 - XML é opcional na maioria dos casos
 - Usa-se anotações (@) para configurar os componentes, configurando-os em tempo de deploy e em tempo de execução
- Vantagem:
 - A configuração está próxima do componente a que se refere

Servlets.

- Classes Java que dão como saída um texto em HTML
- São invocadas pelo Servidor (Tomcat, JBoss, Glassfish, etc)
 - Parâmetros são passados na requisição (*request*)
- O servidor invoca métodos específicos destas classes
 - `init()`
 - `service()`
 - `destroy()`
- Toda a saída gerada em HTML é repassado ao cliente que requisitou a página
 - Resultado é retornado na resposta (*response*)
- São implementadas de uma forma padronizada:
 - Estendem de `HttpServlet`
 - Anotadas com `@WebServlet`
 - Implementam os métodos `doGet()`, `doPost()`

JSP.

- Criação de páginas dinâmicas
- Baseado na tecnologia Java
- Vasta gama de bibliotecas
- Independente de plataforma
- Processamento no servidor (server-side)
- Seu código é convertido em Servlet na primeira execução
- Tags especiais inseridas na página HTML

21

EJB

- EJB – Enterprise Java Bean
- É um componente de software da plataforma Java EE
- Executa em um CONTEINER ou SERVIDOR DE APLICAÇÃO
- Desenvolvimento rápido e componentizado
- Possui características para distribuição
 - Componentes sendo invocados em servidores diferentes

22

EJB: 4 Tipos de Componentes.

- Session Beans
 - Lógica de negócio – com guarda de estado e sem guarda de estado
 - Stateless Session Beans: `@Stateless`
 - Stateful Session Beans: `@Stateful`
 - Singleton Session Beans: `@Singleton`
- Message-driven Beans
 - Lógica de negócio baseada em mensagens (assíncrona)
 - Message-Driven Beans: `@MessageDriven`

23

JSF..

Especificação Java para construção de aplicações Web baseadas em componentes

Funciona via Requisição e Resposta

- O processamento de um requisição se dá através de um ciclo de vida bem definido
- Resolve o problema de não manutenção de estado de componentes

Duas tecnologias fundamentais

- Facelets:
 - Tecnologia de template para telas JSF (Não usa JSP)
 - Integração com EL - Expression Language (`#{bean.propriedade}`)
 - Arquivos .xhtml
- Managed Beans:
 - Classes Java - POJO
 - Beans gerenciados (ciclo de vida) pelo Framework (CDI)
 - Componentes que armazenam informações e executam ações

Admite componentes ricos: ex, primefaces

24

3 Servidores de Aplicação

Prof. Dr. Razer A N R Montano

JAVA WEB

25

25

Servidores

- Implementam a especificação Jakarta EE
 - <https://jakarta.ee/compatibility/>
- Java EE/Jakarta EE possui dois *profiles*:
 - Coleção de tecnologias que devem ser implementadas
 - *Full Platform Profile* e *Web Profile*

Prof. Dr. Razer A N R Montano

JAVA WEB

26

26

Servidores.

- *Full Platform Profile*
 - Implementa toda a API do Java EE/Jakarta EE
 - Ex.: Glassfish Full Platform, Wildfly Java EE Full & Web Distribution
- *Web Profile*
 - Subconjunto do Java EE/Jakarta EE
 - Elementos mais usados: Servlets/JSP, EL/JSTL, JSF, JAX-RS, JTA, JPA, CDI, Etc
 - Ex.: Apache TomEE, Eclipse Jetty + add-ons, Wildfly Java EE Full & Web Distribution
- Servlet-Only / Servlet Container / Web Container
 - Não é oficialmente um "*profile*", mas é muito comum
 - Alternativa leve ao *Web Profile*
 - Container Servlet: HTTP web server, Servlets, JSP, EL, Web Socket, JNDI
 - Ex.: Apache Tomcat, Eclipse Jetty, Wildfly Servlet-Only Distribution

Prof. Dr. Razer A N R Montano

JAVA WEB

27

27

Eclipse Glassfish

- Servidor de aplicação
 - Open Source
 - Implementação de Referência do Jakarta EE
- <https://glassfish.java.net/>
- Implementa TODA a especificação Jakarta EE
 - Possui versões com Web Profile e Imagem Docker
- História
 - 2005 v1: Iniciado pela SUN MICROSYSTEMS, Suporte Java EE 5
 - 2007 v2
 - 2009 v2.1
 - 2009 v3: Implementação de Referência do Java EE 6
 - 2011 v3.1: Oracle
 - 2013 v4: Suporte a Java EE 7
 - 2017 v5: Suporte a Java EE 8
 - 2019 v5.1: Igual ao da Oracle v5.1, mas com o código todo transferido para Eclipse Foundation
 - 2020 v6: Eclipse Glassfish – Suporte ao Jakarta EE 9

Prof. Dr. Razer A N R Montano

JAVA WEB

28

28

Eclipse Glassfish

- Servidor de aplicação
 - Open Source
 - Implementação de Referência

<https://glassfish.java.net/>

- Implementa TODA a especificação Java EE
- Para iniciá-lo em linha de comando
 - Ir até: <Diretório do GLASSFISH>/bin
 - Executar:
`./asadmin start-domain --verbose domain1`
- Estará disponível em
`http://localhost:8080`

Eclipse Glassfish

- Listar os domínios

`./asadmin list-domains`

```
hilbert:bin razer$ ./asadmin list-domains
domain1 not running
domain111 not running
Command list-domains executed successfully.
hilbert:bin razer$
```

- Criar um domínio

`./asadmin create-domain --adminport 4848 dominioteste`

Eclipse Glassfish

- Criar um domínio

```
./asadmin create-domain --adminport 4848 dominioteste
```

```
hilbert:bin razer$ ./asadmin create-domain --adminport 4848 dominioteste
Enter admin user name [Enter to accept default "admin" / no password]>
Using port 4848 for Admin.
Using default port 8080 for HTTP Instance.
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using default port 8181 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Using default port 6666 for OSGI_SHELL.
Using default port 9009 for JAVA_DEBUGGER.
Distinguished Name of the self-signed X.509 Server Certificate is:
[CN=hilbert.local,OU=GlassFish,O=Oracle Corporation,L=Santa Clara,ST=California,C=US]
Distinguished Name of the self-signed X.509 Server Certificate is:
[CN=hilbert.local-instance,OU=GlassFish,O=Oracle Corporation,L=Santa Clara,ST=California,C=US]
Domain dominioteste created.
Domain dominioteste admin port is 4848.
Domain dominioteste allows admin login as user "admin" with no password.
Command create-domain executed successfully.
hilbert:bin razer$
```

Prof. Dr. Razer A N R Montaña

JAVA WEB

31

31

Eclipse Glassfish

- Para iniciar um domínio

```
./asadmin start-domain --verbose dominioteste
```

```
GlassFish Server Open Source Edition 5.1.0 (default-private) startup time : Felix (2.491ms), startup services(591ms), total(3.082ms)l#]
```

- HTTP Estará disponível em
`http://localhost:8080`
- ADMIN Estará disponível em
`http://localhost:4848`

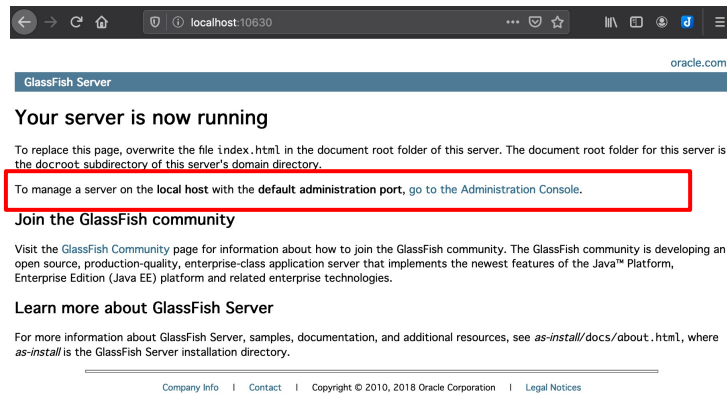
Prof. Dr. Razer A N R Montaña

JAVA WEB

32

32

Eclipse Glassfish

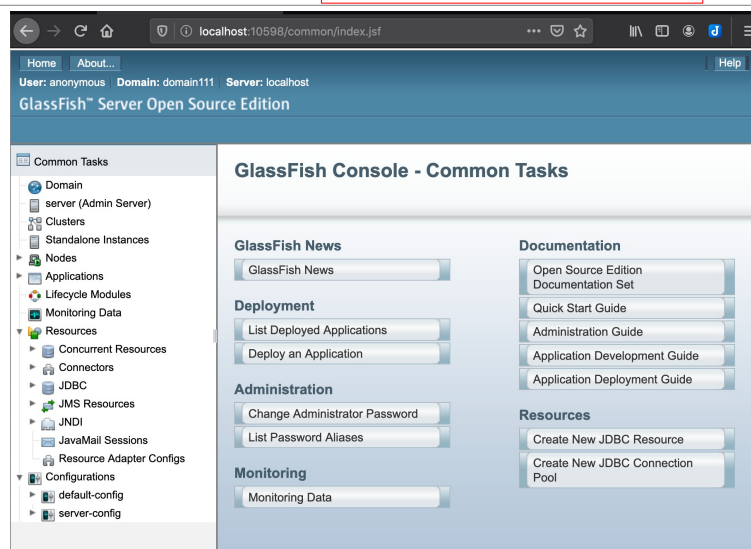


✓ No navegador digital (porta HTTP):
 ✓ `http://localhost:8080`
 ✓ `http://localhost:10630`

33

Eclipse Glassfish

✓ Direciona para a porta ADMIN:
 ✓ `http://localhost:4848`
 ✓ `http://localhost:10598`



34

Eclipse Glassfish

✓ Lista com todas as aplicações instaladas

The screenshot shows the Eclipse Glassfish web console. The left sidebar has a tree view with 'Applications' selected and highlighted by a red box. The main panel shows the 'Applications' section with a table of deployed applications. A red text box above the sidebar says '✓ Lista com todas as aplicações instaladas'.

Select	Name	Deployment Order	Enabled	Engines	Action
<input type="checkbox"/>	TesteNB12v1	100	✓	web	Launch Redeploy Reload

Prof. Dr. Razer A N R Montaña

JAVA WEB

35

35

Eclipse Glassfish

✓ Implantar uma nova aplicação

The screenshot shows the Eclipse Glassfish web console. The left sidebar has a tree view with 'Applications' selected. The main panel shows the 'Applications' section with a table of deployed applications. A red box highlights the 'Deploy...' button in the table's toolbar. A red text box above the sidebar says '✓ Implantar uma nova aplicação'.

Select	Name	Deployment Order	Enabled	Engines	Action
<input type="checkbox"/>	TesteNB12v1	100	✓	web	Launch Redeploy Reload

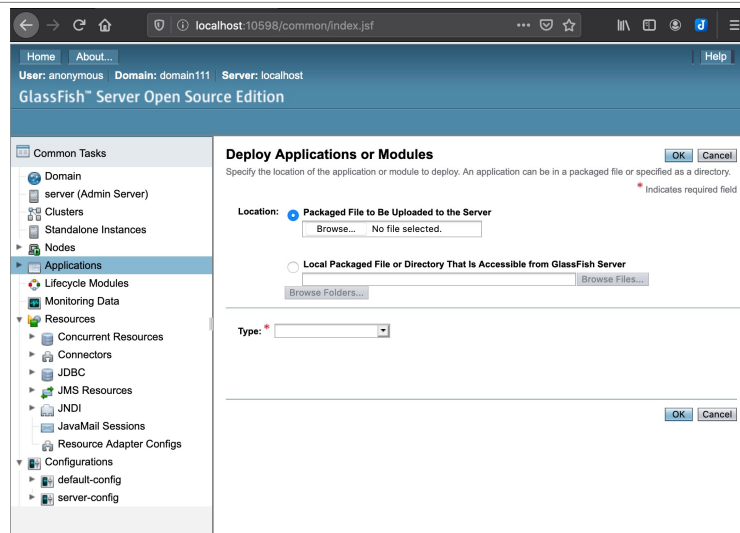
Prof. Dr. Razer A N R Montaña

JAVA WEB

36

36

Eclipse Glassfish



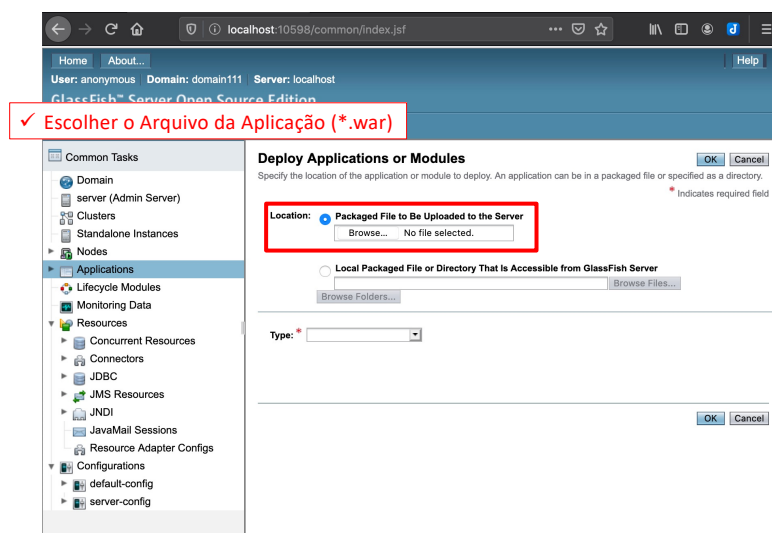
Prof. Dr. Razer A N R Montano

JAVA WEB

37

37

Eclipse Glassfish



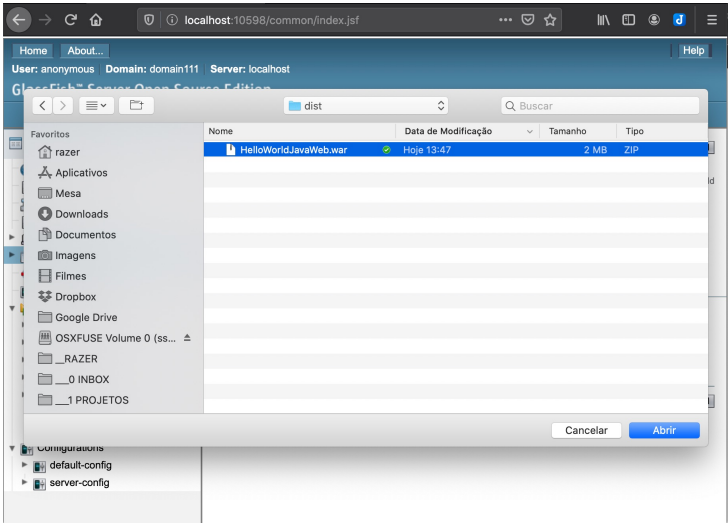
Prof. Dr. Razer A N R Montano

JAVA WEB

38

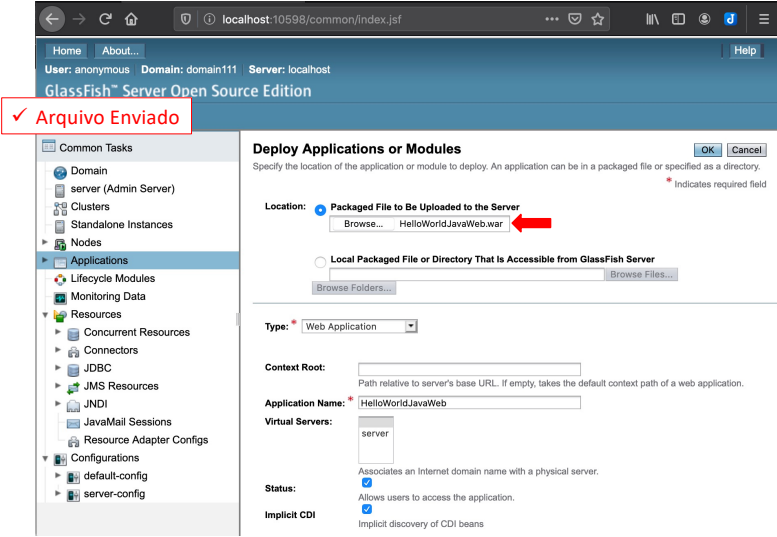
38

Eclipse Glassfish



39

Eclipse Glassfish



40

Eclipse Glassfish

✓ Setar o Context Root (para ele não colocar um número aleatório)

Deploy Applications or Modules

Specify the location of the application or module to deploy. An application can be in a packaged file or specified as a directory. * Indicates required field

Location: ☒ Packaged File to Be Uploaded to the Server
Browse... HelloWorldJavaWeb.war

☐ Local Packaged File or Directory That is Accessible from GlassFish Server
Browse Folders... Browse Files...

Type: * Web Application

Context Root: HelloWorldJavaWeb
Path relative to server's base URL. If empty, takes the default context path of a web application.

Application Name: * HelloWorldJavaWeb

Virtual Servers: server
Associates an Internet domain name with a physical server.

Status: ☒ Allows users to access the application.

Implicit CDI: ☒ Implicit discovery of CDI beans

Prof. Dr. Razer A N R Montano

JAVA WEB

41

41

Eclipse Glassfish

✓ Confirmar a Instalação

Deploy Applications or Modules

Specify the location of the application or module to deploy. An application can be in a packaged file or specified as a directory. * Indicates required field

Location: ☒ Packaged File to Be Uploaded to the Server
Browse... HelloWorldJavaWeb.war

☐ Local Packaged File or Directory That is Accessible from GlassFish Server
Browse Folders... Browse Files...

Type: * Web Application

Context Root: HelloWorldJavaWeb
Path relative to server's base URL. If empty, takes the default context path of a web application.

Application Name: * HelloWorldJavaWeb

Virtual Servers: server
Associates an Internet domain name with a physical server.

Status: ☒ Allows users to access the application.

Implicit CDI: ☒ Implicit discovery of CDI beans

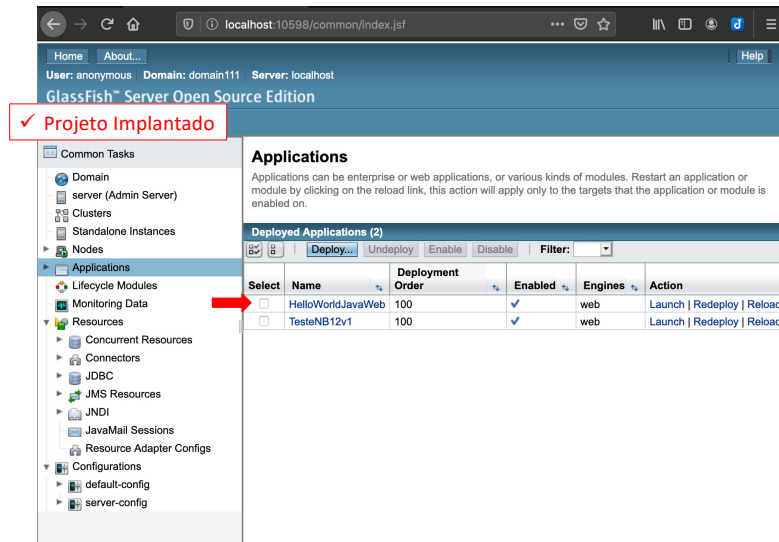
Prof. Dr. Razer A N R Montano

JAVA WEB

42

42

Eclipse Glassfish



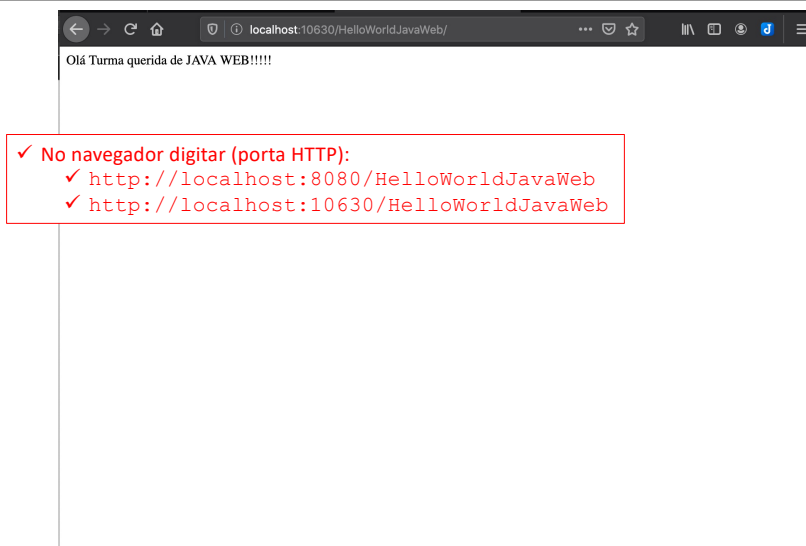
Prof. Dr. Razer A N R Montañó

JAVA WEB

43

43

Eclipse Glassfish.



Prof. Dr. Razer A N R Montañó

JAVA WEB

44

44

Outros Servidores de Aplicação..

- Servidores
 - Apache Tomcat – Servlet Container
 - 10.0.x (Java 8+) – Servlet 5.0 / JSP 3.0 / EL 4.0 (Jakarta EE 9 development)
 - *** 9.0.x (Java 8+) – Servlet 4.0 / JSP 2.3 / EL 3.0 (Jakarta EE 8)
 - ~~8.x (Java 7+) – Servlet 3.1 / JSP 2.3 / EL 3.0~~
 - Eclipse Glassfish – Application Server (Servlet+EJB)
 - 6.x – Jakarta EE 9
 - *** 5.1.0 – Jakarta EE 8
 - 4.x.y – Java EE 7
 - Wildfly
 - *** 20.0.1 (Java 8+) - Jakarta EE 8
 - 19.x (Java 8+) - Jakarta EE 8
 - 18.x (Java 8+) - Jakarta EE 8

45

4 Aplicação Web

46

Arquitetura de uma Aplicação

Baseada em Requisição e Resposta (*request / response*)

Pelo menos dois agentes participando:

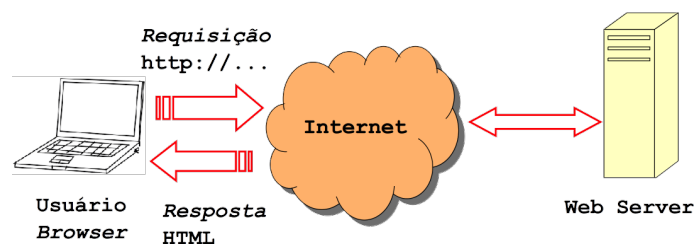
- **Cliente** : normalmente navegador, faz as requisições. Ex.: firefox, chrome, safari
- **Servidor** : aplicação servidora que responde às requisições, enviando as respostas. Ex.: Glassfish, Tomcat, Wildfly

Sobre HTTP

- Agentes geograficamente dispersos
- Mesmo que localmente, deve-se acessar a aplicação pelo navegador:
`http://localhost:8080/app/index.jsp`

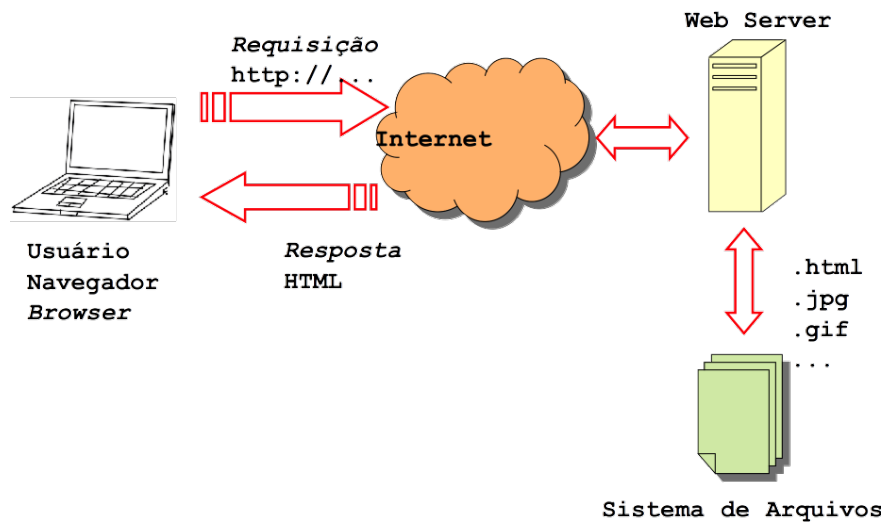
47

Arquitetura de uma Aplicação



48

Arquitetura de uma Aplicação



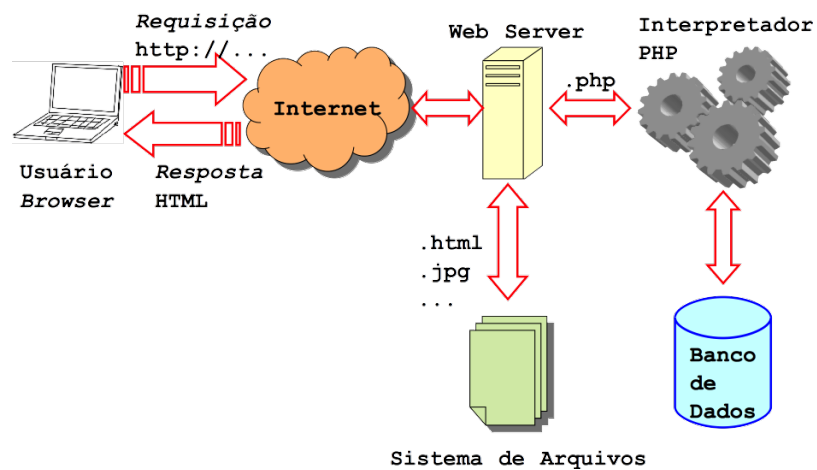
Prof. Dr. Razer A N R Montaña

JAVA WEB

49

49

Arquitetura de uma Aplicação



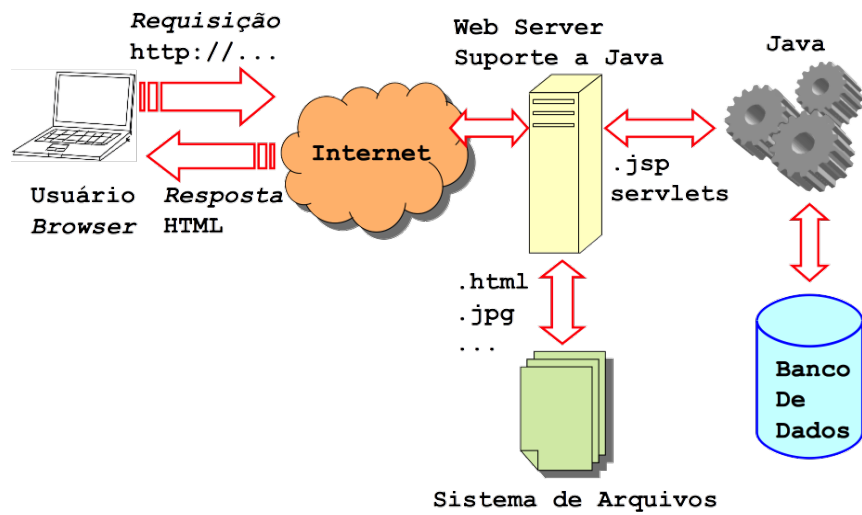
Prof. Dr. Razer A N R Montaña

JAVA WEB

50

50

Arquitetura de uma Aplicação..



51

5 Ferramentas

52

Ferramentas..

IDEs

- (pré-apache) Netbeans 8.2: <https://netbeans.org/downloads/old/8.2/>
- (apache) Netbeans 12: <https://netbeans.apache.org/download/index.html>
- Eclipse IDE for Enterprise Java Developers: <http://www.eclipse.org/downloads/eclipse-packages/>

Bancos de Dados

- PostgreSQL: <https://www.postgresql.org/download/>
- MySQL: <https://dev.mysql.com/downloads/>

Web Profile e Servlet-Only

- (Servlet Container) Tomcat 9.0.x: <https://tomcat.apache.org/download-90.cgi>
- Wildfly Servlet-Only
- Glassfish Web Profile

Servidores Jakarta EE 8

- Glassfish 5.0.1: <https://projects.eclipse.org/projects/ee4j.glassfish/downloads>
 - Full Platform
- Wildfly 20.0.1: <http://wildfly.org/downloads/>
 - Java EE Full & Web Distribution