

Web Technologies

Tecnologias de Middleware
2004/2005

Hugo Simões
hsimoes@di.fc.ul.pt

1

Tópicos da apresentação

- A Web
- Tecnologias Web para suporte a clientes remotos (Applets, CGI, Servlets)
- Servidores Aplicacionais (J2EE)
- Tecnologias Web para integração de aplicações (ORBs, EDIFACT, XML)

2

A Web

- Como um meio de partilha de informação
- Como um meio de comunicação entre clientes remotos e aplicações na internet
- Como um meio de integração de aplicações através da internet

3

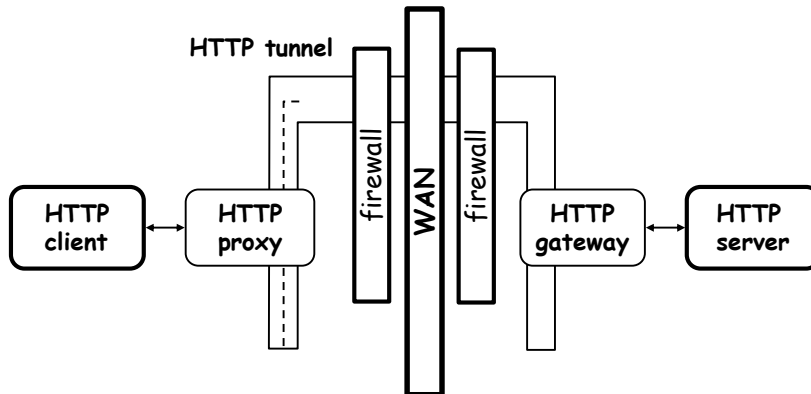
A Web

- Tecnologias centrais
 - HTTP: protocolo de comunicação
 - HTML: ling. de apresentação(hipertexto)
 - Web servers: servidor(disponibiliza serviços)
 - Web browsers: cliente(usa serviços)

4

A Web

- Exemplo de um esquema de comunicação entre um cliente HTTP e um servidor HTTP



5

Tecnologias Web para suporte a clientes remotos

- Documentos estáticos
 - Applets(programa java): *client side*
 - Documento resultante: HTML(estático) + Applets
- Documentos dinâmicos
 - CGI(programa num leque variado de ling.)
 - Servlets(programa java c/ threads)
 - Ambos:
 - *Server side*
 - Documento resultante: HTML(dinâmico)

6

Servidores Aplicacionais

A Web como meio de acesso generalizado,
a sistemas de informação



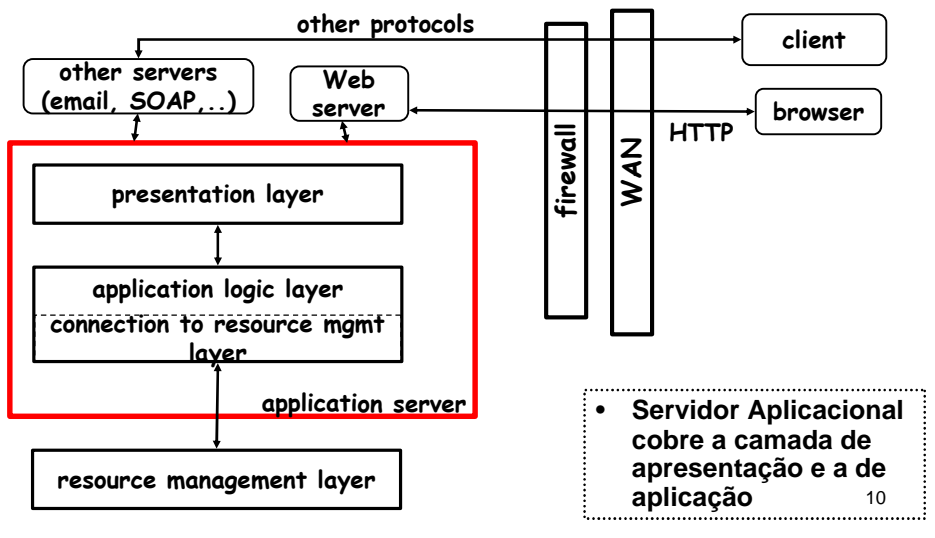
As plataformas de middleware fornecessem
suporte para acesso Web



Servidores aplicativos (J2EE, .NET, ...)

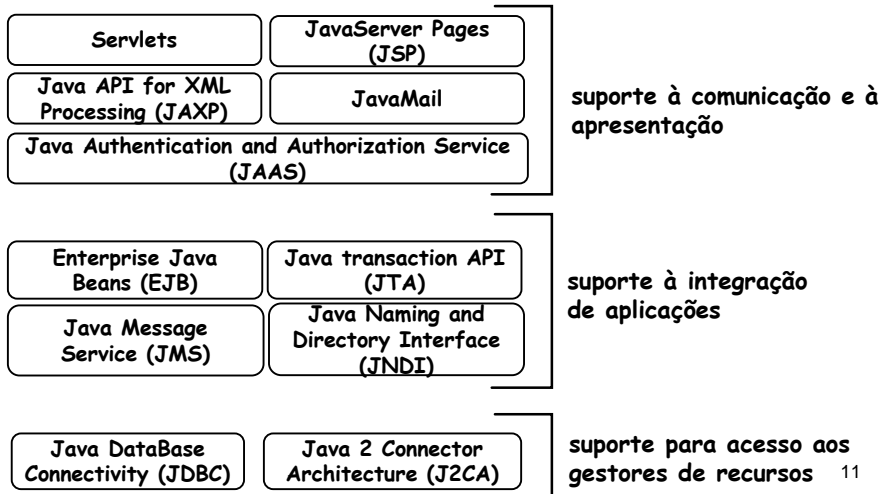
9

Servidores Aplicacionais



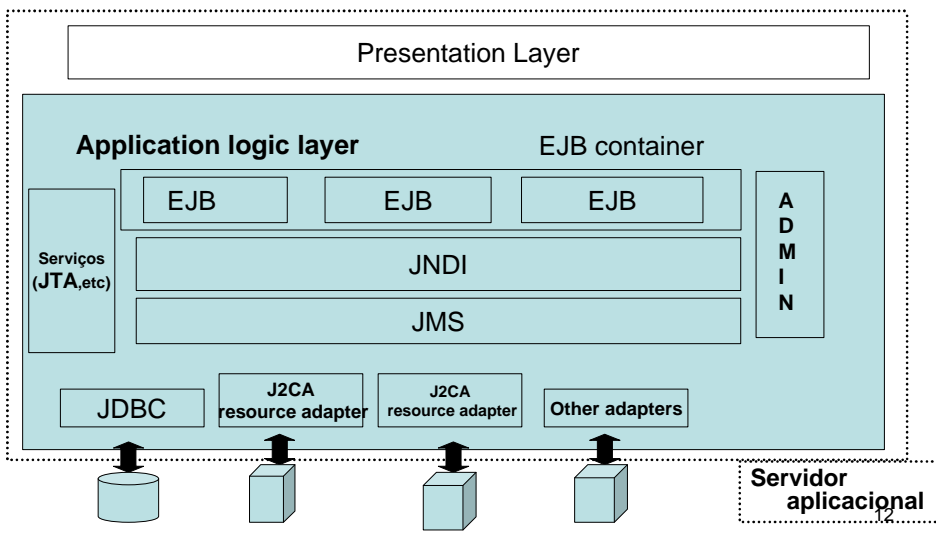
Servidores Aplicacionais

J2EE: especificação principal



Servidores Aplicacionais

J2EE: camada de aplicação



Servidores Aplicacionais

J2EE: camada de aplicação

- EJB (Enterprise JavaBeans)
 - Java API
 - Encapsula funcionalidades lógicas de negócio
 - Define uma arquitectura baseada em componentes que podem ser acedidos localmente ou remotamente
 - Há 3 tipos de EJBs
 - Session beans
 - Entity beans
 - Message-driven beans

13

Servidores Aplicacionais

J2EE: camada de aplicação

- EJB: descrição dos 3 tipos
 - Session beans
 - Encapsula processos e regras lógicas de negócio
 - Trata de operações com um cliente
 - Uma dada sessão permite executar operações apenas com um cliente
 - Numa dada sessão podem ser invocados outros EJBs e recursos para extrair informação de uma base de dados
 - Exemplo: um Session Bean pode calcular os custos de envio da compra de um livro via Amazon

14

Servidores Aplicacionais

J2EE: camada de aplicação

- EJB: descrição dos 3 tipos
 - Entity beans
 - Representam objectos persistentes(ie. Uma tabela de numa base de dados)
 - Podem ser acedidos por múltiplos utilizadores
 - Podem invocar recursos para acesso a uma base de dados
 - Não podem fazer chamadas a outros EJBs

15

Servidores Aplicacionais

J2EE: camada de aplicação

- EJB: descrição dos 3 tipos
 - Message-driven beans (MDB)
 - Objectos persistentes
 - Processam mensagens assíncronas
 - Respondem a mensagens que estão numa fila, enviadas por outros EJBs
 - Exemplo:
um MDB para responder a uma entidade “Inventário de Stock” avisando que o um produto está esgotado

16

Servidores Aplicacionais

J2EE: camada de aplicação

- JMS: Java Message Service
 - É uma maneira de todas aplicações java comunicarem entre si através de mensagens
 - É um conjunto de interfaces que permitem a criação, o envio e a recepção de mensagens entre vários componentes, em ambientes distribuídos
 - O MDB usa o JMS

17

Servidores Aplicacionais

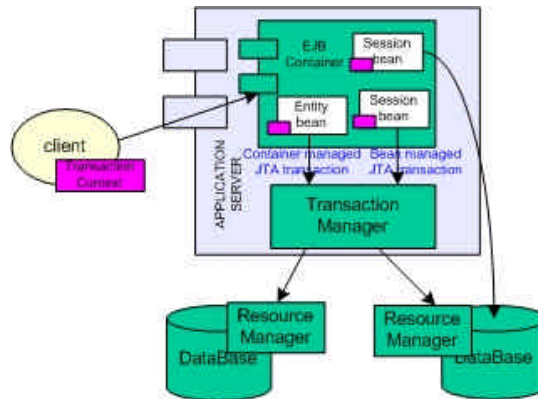
J2EE: camada de aplicação

- JNDI: Java Naming and Directory Interface
 - Fornece um mecanismo para localização de recursos
 - Permite que uma aplicação localize componentes num determinado contexto
 - Estes recursos podem ser objectos remotos, serviços de directórios, etc.

18

Servidores Aplicacionais J2EE: camada de aplicação

- JTA: Java Transaction API
 - Fornece um serviço para transacções



19

Servidores Aplicacionais J2EE: camada de aplicação

- JDBC: Java Database Connectivity
 - API para ligação a bases de dados relacionais
 - Um aplicação aceder às base de dados, executar instruções SQL e obter resultados
 - Exemplo

```
private String dbName = "java:comp/env/jdbc/BankDB";  
InitialContext ic = new InitialContext();  
DataSource ds = (DataSource) ic.lookup(dbName);  
Connection con = ds.getConnection();
```

20

Servidores Aplicacionais

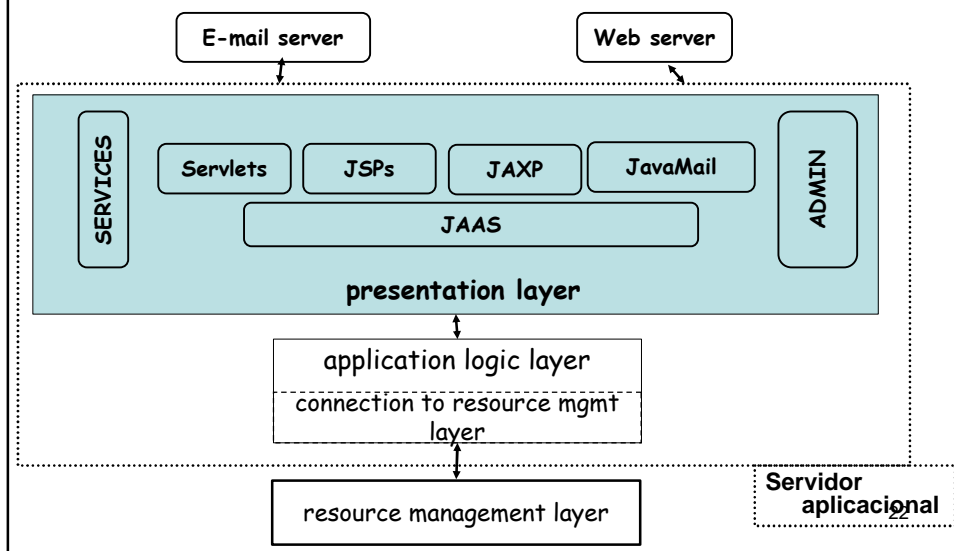
J2EE: camada de aplicação

- J2CA: J2EE Connector Architecture
 - Mais genérico do que o JDBC (não assume que os dados são apenas relacionais)
 - Permite a ligação com dados de diferentes tipos e de diferentes sistemas

21

Servidores Aplicacionais

J2EE: camada de apresentação



Servidores Aplicacionais

J2EE: camada de apresentação

- JSPs: JavaServer Pages
 - Geração dinâmica de documentos
 - Server e platform independent
 - Separação da interface e do conteúdo
 - Usa tags tipo XML para encapsular o conteúdo (*server-side resident code*)
- JAXP: Java API for XML Processing
 - API de suporte a XML
 - Fornece interface de *parsing e transforming*

23

Servidores Aplicacionais

J2EE: camada de apresentação

- JavaMail: API para sistemas de correio electrónico a partir de aplicações J2EE

Exemplo

```
public void sendMail(String address) throws java.rmi.RemoteException {
    Session session = null;
    try {
        session = (Session)PortableRemoteObject.narrow(
            new InitialContext().lookup("java:Mail"), Session.class);
    } catch (javax.naming.NamingException e) {
        e.printStackTrace();
    }

    try {
        MimeMessage m = new MimeMessage(session);
        m.setFrom();
        Address[] to = new InternetAddress[] {
            new InternetAddress(address)
        };
        m.setRecipients(Message.RecipientType.TO, to);
        m.setSubject("JavaMail Test");
        m.setSentDate(new Date());
        m.setContent("Test from inside EJB Using JBoss",
            "text/plain");
        Transport.send(m);
    } catch (javax.mail.MessagingException e) {
        e.printStackTrace();
    }
}
```

24

Servidores Aplicacionais

J2EE: camada de apresentação

- JAAS: Java Authentication and Authorization Service
 - Autenticação de utilizadores
 - Quem está a executar código
 - Autorização de utilizadores
 - Que pedaço de código está autorizado a executar
 - Sob a forma de plug-in
 - Extensão da arquitectura de segurança tradicional, na execução de código

25

Tecnologias Web para integração de aplicações

As plataformas de middleware foram desenhadas ao nível de uma LAN



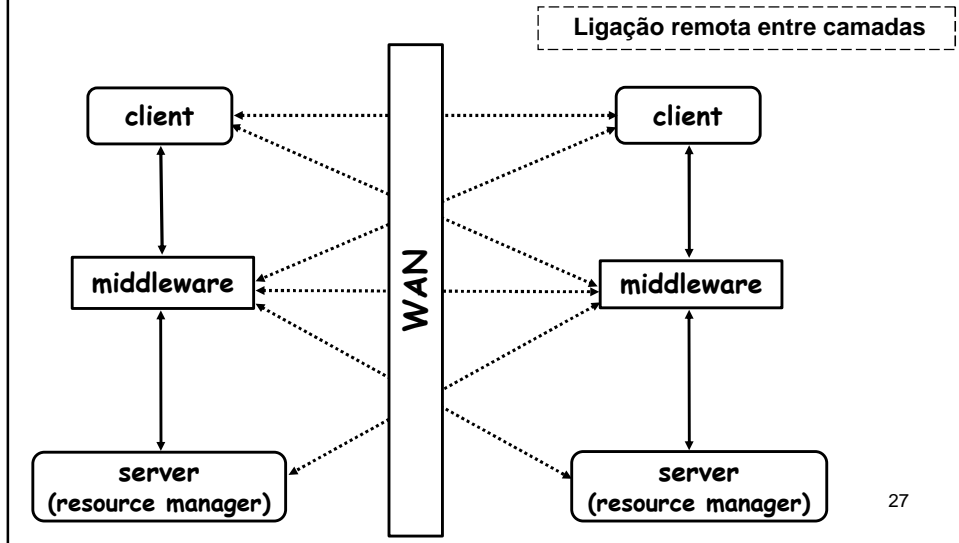
Diferentes plataformas de middleware tem de comunicar entre si (ie. Entre WANs)



Comunicação entre CORBA ORBs para integração de aplicações através da internet

26

Tecnologias Web para integração de aplicações



Tecnologias Web para integração de aplicações

- Problema: Comunicação entre aplicações
- A integração é conseguida à custa do GIOP (General Inter-Orb protocol)
 - Protocolo abstracto usado para comunicação entre CORBA ORBs
 - Especifica como são feitas as chamadas entre um ORB e o outro
 - Especifica os formatos das mensagens a trocar

Tecnologias Web para integração de aplicações

- O IIOP é uma implementação do GIOP
 - Permite implementar CORBA sobre a internet (TCP/IP)
 - Permite que os browsers e os servidores comuniquem entre si não apenas usando texto, mas sim, usando representações de dados do tipo inteiros, arrays, objectos, etc.

29

Tecnologias Web para integração de aplicações

- Outros standard de representação de mensagens e dados para comunicações entre aplicações
 - EDIFACT:
 - Electronic Data Interchange for Administration, Commerce and Transport
 - Fornece templates para as mensagens e para o seu conteúdo
 - ISO/TS 20625:2002 (EDIFACT+XML)
 - XML
 - Estrutura perfeita para uma automatização do processo

30

Referências

- J2EE
 - http://java.sun.com/j2ee/tutorial/1_3-fcs/
- CORBA ORBs + GIOP + IIOP
 - <http://www.omg.org/technology/documents/>
- EDI + EDIFACT + XML
 - <http://www.apedi.pt/>
 - <http://www.unece.org/trade/untdid/>
 - <http://www.dcti.iscte.pt/vextra/index.html>