

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO NORTE
CAMPUS NATAL-CENTRAL

DIRETORIA ACADÊMICA DE GESTÃO E TECNOLOGIA DA INFORMAÇÃO

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

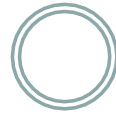


Sistemas Corporativos

Visão Geral e Arquitetura

Prof. Fellipe Aleixo (fellipe.aleixo@ifrn.edu.br)

O que são Sistemas Corporativos

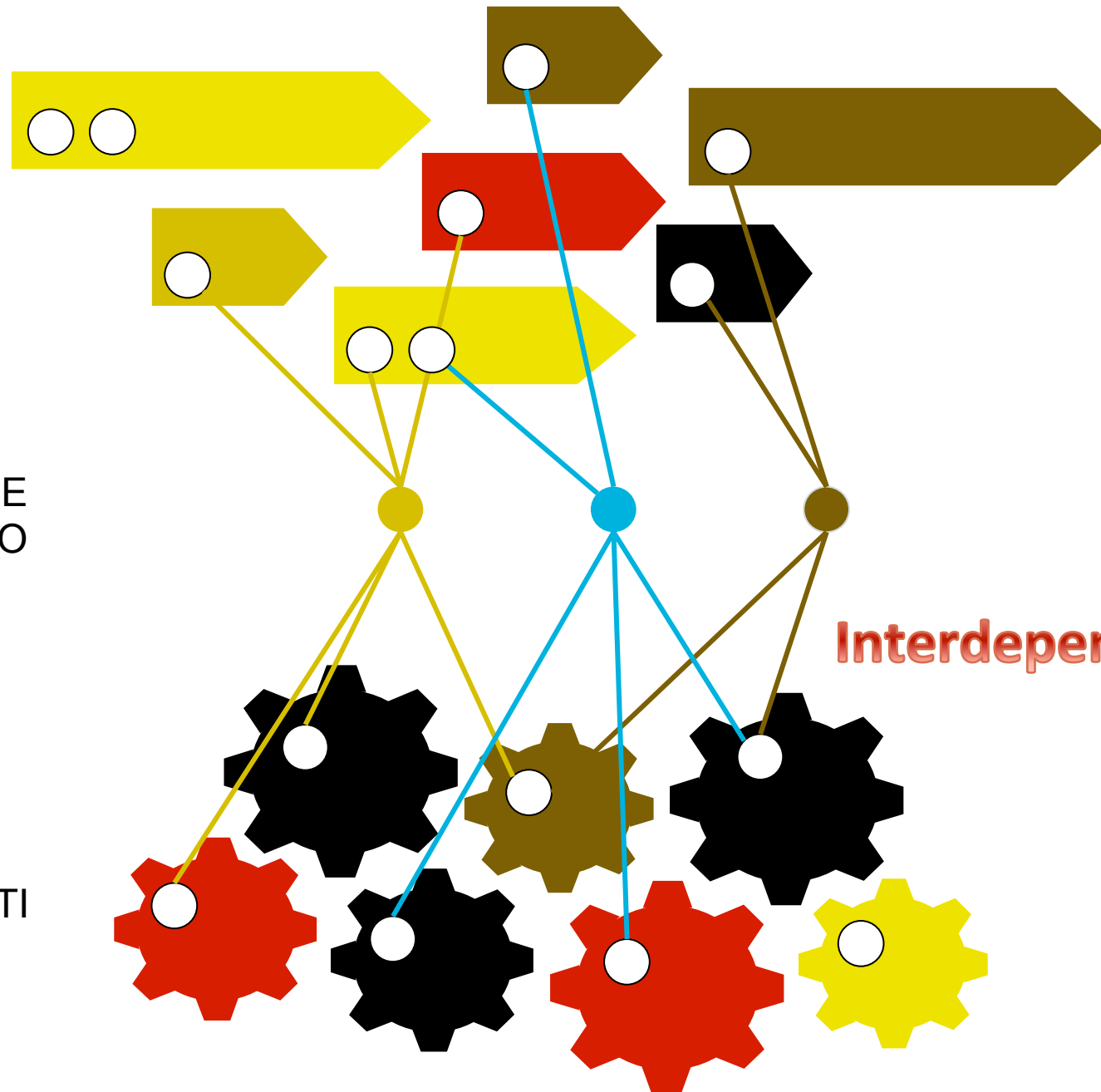


- Sistemas Corporativos (ou *Enterprise Systems*) são sistemas utilizados em empresas, geralmente com foco bastante diferente das aplicações voltadas para usuários finais
- São criadas para necessidades tipicamente empresariais, tais como folha de pagamento, catálogo de produtos, sistemas de cobrança, sistemas de segurança, etc.
- Segundo Martin Fowler:
 - *"Enterprise applications are about the display, manipulation, and storage of large amounts of often complex data and the support or automation of business processes with that data"*

Processos

SISTEMAS DE
INFORMAÇÃO

Recursos de TI

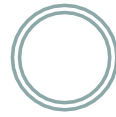


Exemplo de Sistemas Corporativos



**SISTEMAS DE INFORMAÇÃO CORPORATIVOS
DO ESTADO DE MINAS GERAIS**

Visão Geral



- São os grandes sistemas para a gestão do Governo do Estado de Minas Gerais
- São sistemas do tipo SIT (Sistema de Informação Transacional) e fornecem dados para utilização em *Data Warehouse* (armazém de informações)
- São utilizados por usuários dos diversos órgãos do governo, espalhados geograficamente em todo o
- Possuem uma integração que permite a troca de informações entre eles

Os Principais Sistemas



- SIAFI - Sistema Integrado de Administração Financeira
- SISAP - Sistema de Administração de Pessoal
- SISAD – Sistema de Avaliação de Desempenho
- SIPRO – Sistema Integrado de Protocolo
- SigPlan - Sistema de Informações Gerenciais e Planejamento
- Orçamento - Sistema para elaboração do orçamento anual
- SIAD - Sistema de Administração de Materiais

SIAFI – Sistema Integrado de Administração Financeira



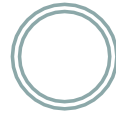
- Objetivo: Administração financeira do Governo do Estado de Minas Gerais
- Órgão Gestor: SEF
- Volume:
 - 1.345 Unidades Executoras
 - 3.000 Pontos de Acesso
 - 12.000 Usuários
 - 560.000 Empenhos/Ano
 - 750.000 Liquidações/Ano
 - 850.000 Pagamentos/Ano
 - 20 Milhões Lançamentos Contábeis/Ano
 - 800 Mil Transações com Bancos/Ano
 - 54 Milhões Operações/Ano (Acessos)

SIAFI – Sistema Integrado de Administração Financeira



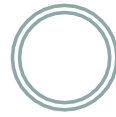
- Módulos
 - Rotina Administrativa:
 - Reconciliação de Conta Arrecadadora
 - Especificação de Compra/serviço
 - Contrato/Convênio
 - Rotina Anual:
 - Apropriação Orçamento Anual
 - Restos a Pagar
 - Encerramento Contábil do Exercício
 - DIRF
 - Anulação saldo de Empenho
 - Transferência Saldo Contábil
 - Balanço Anual
 - Movimentação Orçamentária
 - Movimentação da Receita
 - Movimentação da Despesa
 - Movimentação Financeira Escritural
 - Movimentação Financeira Bancária
 - Movimentação Contábil
 - Consultas e Relatórios
 - Tabelas

SISAP - Sistema de Administração de Pessoal



- Objetivo: Administração de pessoal do Governo do Estado de Minas Gerais
- Órgão Gestor: SEPLAG
- Principais Funções
 - Ingresso/Desligamento;
 - Movimentações Funcionais;
 - Eventos Funcionais;
 - Pagamento de Pessoal;
 - Agenda Médica;
 - Nomeação;
 - Informações Curriculares;
 - Quadro de Vagas;
 - Contagem de Tempo.

SIPRO – Sistema Integrado de Protocolo



- Objetivo: Gerenciamento do tramite de documentos no governo
- Órgão Gestor: SEPLAG
- Volume:
 - 34 Órgãos utilizam o SIPRO
 - Média de 2.250 Processos cadastrados / dia
 - Média de 6.200.Tramitações / dia
 - 7 milhões de Processos cadastrados
 - 20 Milhões de Tramitações cadastradas

SIPRO – Sistema Integrado de Protocolo



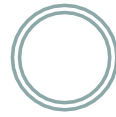
- Principais funções
 - Relativas a Processos:
 - Cadastramento de Processos,
 - Cadastramento de Solicitante
 - Arquivamento/Desarquivamento de Processo
 - Auditoria de Processo
 - Recibo de Protocolo
 - Relativas à Tramitação:
 - Inclusão/Alteração
 - Recebimento
 - Emissão de Guia
 - Consultas/Relatórios

SigPlan - Sistema de Informações Gerenciais e de Planejamento



- Objetivo: Criação e acompanhamento do Plano Plurianual do Governo do Estado de Minas Gerais - PPAG.
- Órgão Gestor: SEPLAG
- Volume:
 - 126 unidades de Planejamento Cadastradas
 - 278 Usuários ativos cadastrados
- Principais Funções:
 - Cadastramento, Gerenciamento, Monitoramento e Avaliação das informações relativas aos Programas de Governo, suas ações e Locais de Execução

Orçamento - Sistema para elaboração do orçamento anual



- Objetivo: Elaboração e registro das informações relativas à Proposta Orçamentária Anual dos órgãos que compõem o Governo do Estado de Minas Gerais conforme legislação estabelecida.
- Órgão Gestor: SEPLAG
- Volume:
 - 126 Unidades Orçamentárias cadastradas
 - 238 Usuários ativos cadastrados

Orçamento - Sistema para elaboração do orçamento anual



- Principais Funções:
 - Permite o Registro e o tratamento de :
 - Orçamento da Receita
 - Metas e Programas de Trabalho
 - Orçamento da Despesa
 - Detalhamento das Obras
 - Obras Por Município
 - Gastos com Pessoal
 - Repasse de Recursos

SIAD - Sistema de Administração de Materiais



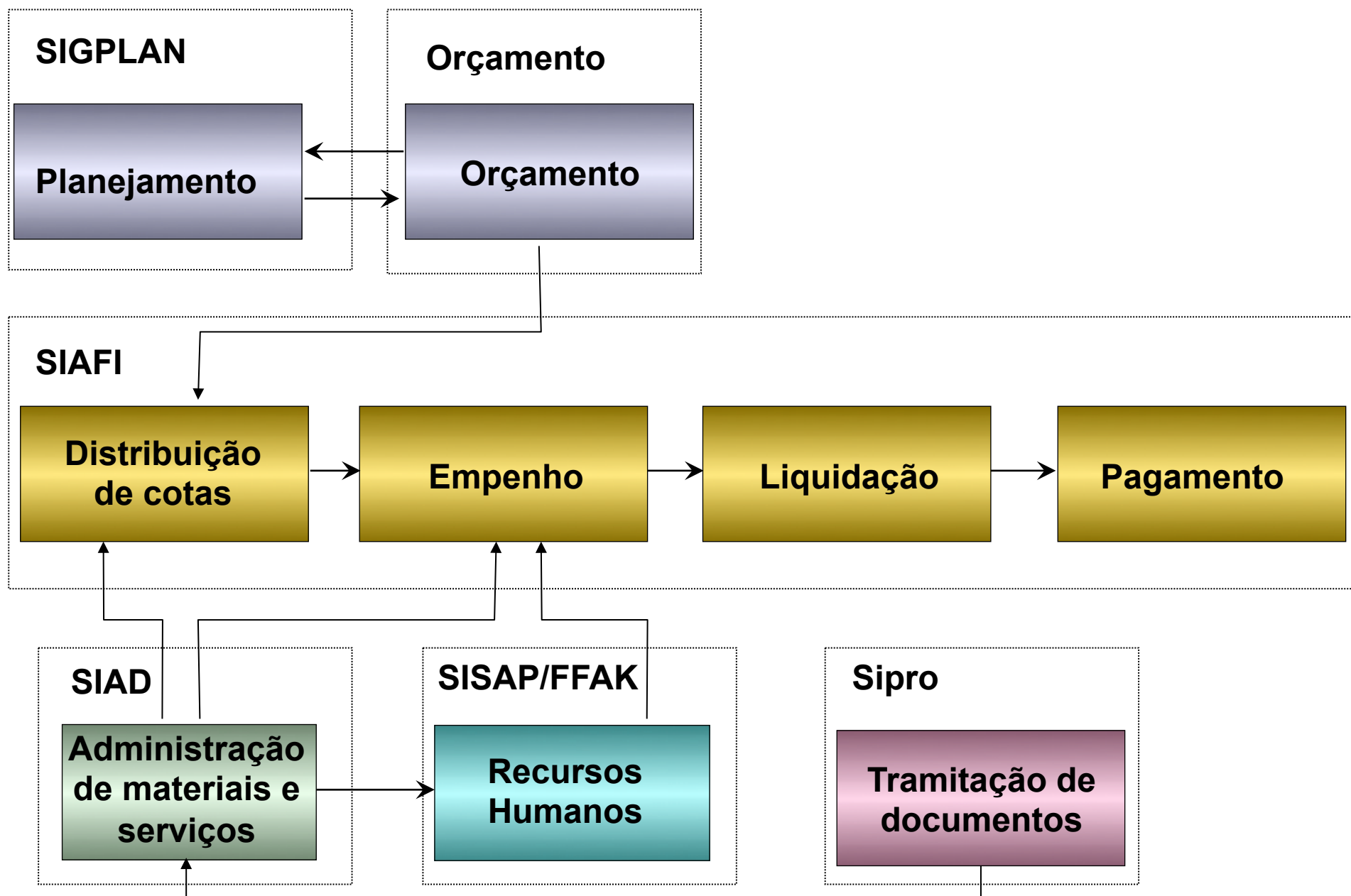
- Objetivo: Gerenciamento de todo ciclo de materiais e serviços (requisição, aquisição, utilização e encerramento)
- Órgão Gestor: SEPLAG

SIAD - Sistema de Administração de Materiais



- Principais Funções:
 - Compras
 - Cotação Eletrônica
 - Contrato
 - Material de Consumo
 - Material Permanente
 - Bolsa de Materiais
 - Patrimônio
 - Estoque
 - Frota de veículos
 - Alienação
 - Órgãos e entidades
 - Fornecedores
 - Licitanet
 - Registro de Preços

Integração dos Sistemas Corporativos

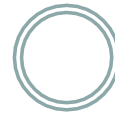


Plataforma Java EE



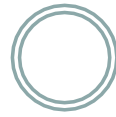
JAVA ENTERPRISE EDITION

O Padrão JEE



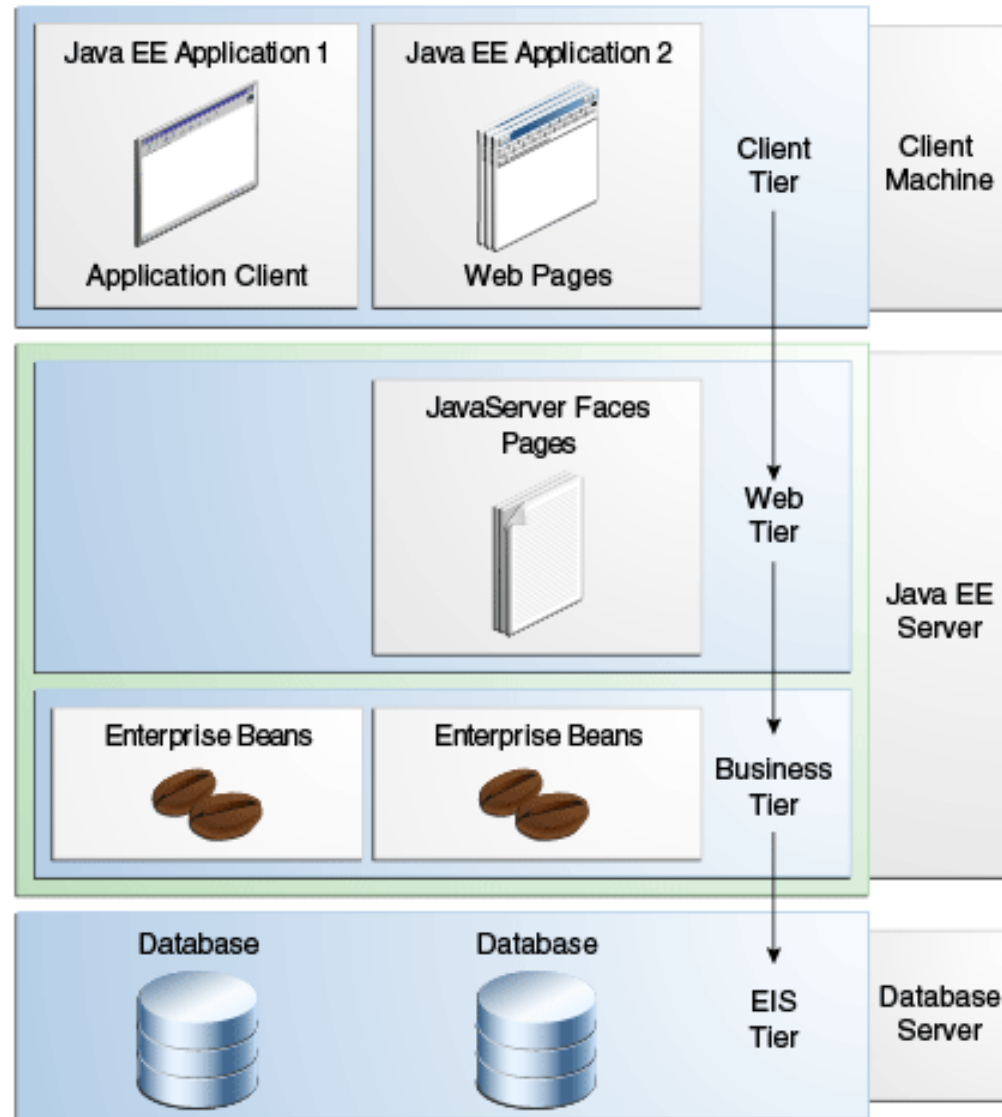
- Para suportar a criação de aplicações Java Corporativas, foi criado o *Java Enterprise Edition* (JEE)
- O padrão JEE acrescenta várias APIs ao SDK do Java, que incorporam funcionalidades de
 - tolerância a falhas,
 - aplicações distribuídas,
 - aplicações multicamadas,
 - entre outras, a serem executadas em um servidor de aplicação
- O padrão JEE é uma especificação, ou seja, somente determina o que deve ser feito. O como fica a cargo do implementador

O Padrão JEE



- Na especificação JEE, as extensões ao Java padrão ficam todas no pacote “javax”, são elas:
 - faces → construção de interfaces componentizadas
 - servlet → lida com requisições HTTP
 - enterprise → lida com injeção de dependência
 - **ejb → acesso a objetos**
 - validation → validação de objetos
 - transaction → controle de transações
 - jms → comunicação assíncrona
 - resource → conectores para integração entre sistemas

Aplicação Distribuída e Multicamadas



Definição de EJB



“A arquitetura Enterprise JavaBeans – EJB – é uma arquitetura para o desenvolvimento e a implantação de aplicativos de negócio distribuídos baseados em componentes. Aplicativos escritos utilizando a arquitetura EJB são escalonáveis, transacionais e seguros com multiusuários. Esses aplicativos podem ser escritos uma vez e então implantados em qualquer plataforma de servidor que suporta a especificação EJB”

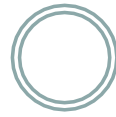
Sun Microsystems

O Padrão JEE



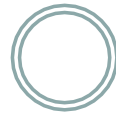
- Além destes, tem-se pacotes relativos à:
 - manipulação de XML,
 - e-mail,
 - além de especificações que se aplicam aos componentes JEE, como Conectores, Portlets, JavaBeans, etc.
- Sendo uma especificação, cada fornecedor cria a implementação do jeito que melhor convier
 - Assim, devemos ter em mente que não devemos esperar que duas implementações sejam iguais, somente que ambas façam a mesma coisa

Servidores de Aplicação



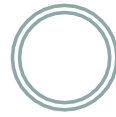
- Para utilizarmos as funcionalidades JEE, precisamos de um Servidor de Aplicações
- Um *Application Server* (AS) é um servidor capaz de prover implementações das funcionalidades JEE
 - Um servidor que tenha implementações para toda especificação JEE é chamado de *Full Compliant*
- Exemplos destes servidores: JBoss, Oracle Glassfish, IBM WebSphere e Oracle Weblogic

Servidores de Aplicação



- Alguns servidores JEE, no entanto, provém implementações somente para a parte Web, chamados de *Web Profile Compliant*
- Um AS que seja *Web Compliant* permite o *deploy* de aplicações Web; porém, o suporte a funcionalidades extras depende de cada servidor.
 - Por exemplo, o JBoss é um servidor *Web Compliant* porque não provê funcionalidade JMS total
 - Alguns exemplos de servidores Web Compliant: JBoss, Glassfish Web Profile, Caucho Resin e Apache TomEE

Servidores de Aplicação



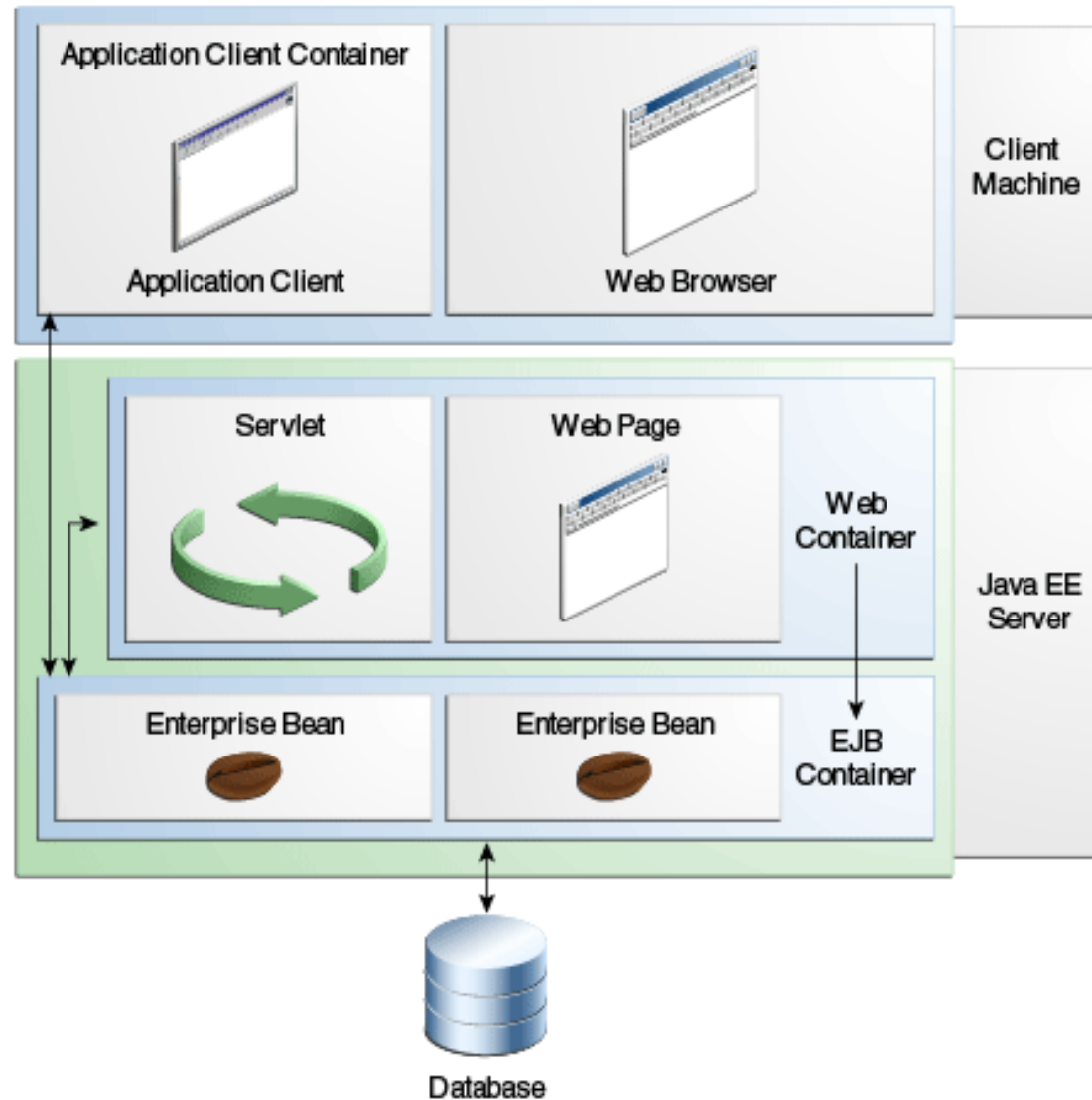
- Alguns servidores provêm a funcionalidade de *servlet containers*: é possível realizar o *deploy* de aplicações Web, mas somente gerenciam este aspecto das aplicações
- Com isso, podemos criar aplicações que respondem à requisições Web, mas nada além disso!
 - Por exemplo, não podemos fazer comunicação JMS com um servidor deste tipo
- Estes servidores não são Application Servers!
 - Como exemplo, temos o Jetty e o Tomcat.

Servidores de Aplicação



- Um servidor de aplicação é composto por vários contêineres – manipulação dos tipos específicos de componentes
 - Contêiner Web
 - Contêiner EJB
 - (Contêiner JPA)
 - (Contêiner JMS)

Servidores de Aplicação



Gerenciamento de Recursos



- Um contêiner EJB gerencia os recursos utilizados pelos EJBs
 - Como utilizam a memória
 - Threads
 - Conexões de banco de dados
 - Poder de processamento
 - Entre outros...

Gerenciamento de Recursos



- Serviços primários suportados pelo contêiner:
 - (1) Concorrência
 - (2) Gerenciamento de transação
 - (3) Persistência
 - (4) Distribuição de objetos
 - (5) Atribuição de nomes
 - (6) Segurança
- Serviços adicionais suportados: (i) sistema de mensagens assíncronas e (ii) serviço e temporização

Gerenciamento de Recursos



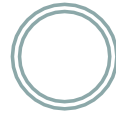
- Idéia de escala:
 - Sistema corporativo \cong milhares ou mesmo milhões de objetos em uso simultaneamente
- Mecanismos para gerenciar beans em execução:
 - Pool de instâncias – instâncias carregadas na memória, prontas para serem compartilhadas
 - Ativação – conservação de instâncias com estado
- Gerência de conexões – Java EE Connector Architecture

Gerenciamento de Recursos



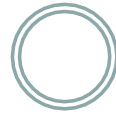
- Estados de um bean de sessão sem informações de estado:
 - Sem estado – ainda não foi instanciada
 - Pooled – foi instanciada pelo contêiner mas ainda não foi associada a uma solicitação EJB
 - Ready (pronto) – foi associada a uma solicitação EJB e está pronta para responder a invocações
- Como não guardam informações de estado – podem ser compartilhados

Gerenciamento de Recursos



- MDBs e Pool de instâncias:
 - Não guardam estado para uma solicitação específica, como os beans de sessão sem informação de estado
 - Na maioria dos contêiners, cada tipo de bean baseado em mensagem tem o seu próprio pool de instâncias
 - JMS-MDBs se inscrevem em um destino específico
 - Um destino equivale a um endereço utilizado para enviar e receber mensagens
 - Ao receber uma mensagem para um destino o contêiner EJB determina que JMS-MDB, que está no pool relativo aquele destino, irá tratar a mensagem

Gerenciamento de Recursos



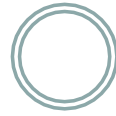
- Mecanismo de ativação:
 - Aplicado a beans de sessão com informações de estado – manutenção do estado conversacional
 - Se um bean de sessão com informações de estado for retirado da memória, o estado do mesmo é serializado e armazenado em disco – passivação
 - O cliente não tem a informação da passivação
 - Ativar um bean é o ato da restauração do estado de uma instância de um bean que sofreu a passivação

Java EE Connector Architecture



- Define uma interface entre Enterprise Information Systems (EISs) e contêiners Java EE
- EIS: termo genérico para qualquer sistema de informações
 - Sistemas de gerenciamento de banco de dados
 - Sistema de mensagens assíncronas
 - CORBA
 - Sistemas de ERP
 - Sistemas legados (p.ex.: IMS e CICS)

Java EE Connector Architecture



- Java EE define algumas APIs corporativas padronizadas, independentes de fornecedor, para a conexão com sistemas de informações
 - JDBC – gerenciadores de banco de dados relacional
 - JMS – middleware orientado a mensagens
 - JNDI – serviços de diretórios e atribuição de nomes
 - Java IDL – CORBA
 - JavaMail – sistemas de correio eletrônico

Java EE Connector Architecture



- Quando um EJB utiliza tais APIs é de responsabilidade do contêiner funções como:
 - Colocar no pool e manter conexões EISs
 - Registrar o EIS nas transações
 - Propagar as credenciais de segurança
- Tais interações exigem um nível de interação não coberto pelas APIs padronizadas

Serviços Primários



- Principais serviços gerenciados pelo contêiner:
 - (1) Concorrência
 - (2) Transações
 - (3) Persistência
 - (4) Objetos distribuídos
 - (5) Sistema de mensagens assíncronas
 - (6) EJB Timer Service
 - (7) Atribuição de nomes
 - (8) Segurança

Concorrência



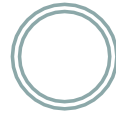
- Aspectos da especificação EJB:
 - Como os servidores EJB tratam a concorrência, os métodos de um *bean* não precisam ser thread-safe
 - Proíbe o uso da palavra-chave *synchronized*
 - Proíbe que os *beans* criem seus próprios threads
- Os *beans* de sessão não suportam acesso concorrente

Concorrência



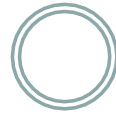
- Concorrência com *beans* de entidade
 - Representam dados que são compartilhados
 - Para tornar possível o acesso concorrente o provedor de persistência precisa bloquear os dados
 - Na especificação JPA – é criada uma cópia da instância do *bean* por transação
- Concorrência com *beans* de mensagens
 - Processamento de mais de uma mensagem por vez – diferentes instâncias do MDB podem trabalhar simultaneamente

Transações



- Conjunto atômico (indivisível) de tarefas
 - Exemplo anterior: (a) criação de uma locação e (b) realização do pagamento da referida locação
- Gerenciadas automaticamente pelo contêiner, sem a necessidade de programação nenhuma
 - Nos beans pode haver a declaração de atributos transacionais, usados em tempo de implantação
- O EJB provê também um mecanismo para o gerenciamento explícito de transações

Persistência



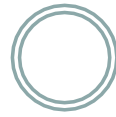
- Beans de entidade:
 - São implementados como POJOs
 - Tem o seu estado salvo em um banco de dados
 - Podem ser criados fora do contêiner
 - Gerenciados pelo EntityManager
 - Podem estar acoplados ao gerenciamento do contêiner ou desacoplados
 - Uma vez desacoplados, podem ser reacoplados através do método merge – alterações sincronizadas com o banco

Persistência



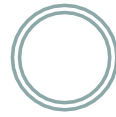
- Nas versões mais recentes a persistência deixou de fazer parte da plataforma, passando a ser especificada a parte – Java Persistence API – JPA
- JPA é uma abstração superior à API JDBC
 - Os objetos são mapeados em tabelas de banco dados, de modo que possam ser consultados, carregados, atualizados ou removidos sem que necessário utilizar a API JDBC
 - Mapeia classes simples (com *gets* e *sets*) – Beans de Entidade

Objetos Distribuídos



- O cliente tem acesso a um EJB através da interface remota ou interface *endpoint*
 - Os demais elementos são abstraídos, inclusive o mecanismo de suporte a objetos distribuídos
- Na especificação EJB, os *beans* de sessão podem ser acessados através de RMI-IIOP
- Também é requerido o suporte ao SOAP 1.2 pela API JAX-RPC

Objetos Distribuídos



- Tendência na computação distribuída
- “Aplicativos modulares auto descritos e autocontidos que podem ser publicados, localizados e invocados pela Web”
 - Independentes de plataforma
 - SOAP – gramática XML para protocolo de aplicativo
 - WSDL – gramática XML para definição de interface
- O EJB permite o desenvolvimento de Serviços Web através da API JAX-WS

Sistema de Mensagens Assíncronas



- Além do suporte a objetos distribuídos baseados em RMI, o EJB suporta também um sistema de mensagens assíncronas
 - Uma mensagem é um pacote autocontido de dados do negócio e cabeçalhos de roteamento de rede
 - Mensagens assíncronas podem ser transmitidas entre um aplicativo e outro em uma rede, utilizando o Message-Oriented Middleware – MOM
 - O MOM assegura tolerância a falhas, escalabilidade, balanceamento de carga e suporte a transações

Sistema de Mensagens Assíncronas



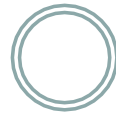
- O contêiner EJB roteia confiavelmente mensagens de clientes JMS para JMS-MDBs
- Mensagens corporativas podem ser serializadas e armazenadas em disco ou em banco de dados até que possam ser adequadamente entregues
 - O armazenamento secundário das mensagens permite a tolerância a falhas, em caso de travamento do servidor EJB
- O sistema de mensagens assíncronas é transacional

EJB Timer Service



- Utilizado para agendar notificações a serem enviadas a beans
- Serviços de temporização são úteis em vários domínios de problemas: (a) vencimento de prestações, (b) expiração de cadastros de usuários e etc.
- Temporizadores podem ser configurados em todo o tipo de EJB, exceto beans de sessão com informação de estado

Atribuição de Nomes



- Fornece aos clientes um mecanismo para localizar recursos ou objetos distribuídos
- Serviços fornecidos:
 - Vinculação de objeto
 - Associação de um objeto distribuídos a um identificador
 - API de pesquisa
 - Fornece ao cliente uma interface para o sistema de atribuição de nomes
 - No EJB é obrigatório o uso da JNDI – suporta qualquer tipo de serviços de diretórios e de atribuição de nomes

Atribuição de Nomes



- Exemplo de utilização da API JNDI:

```
javax.naming.Context jndiContext = new  
  
javax.naming.InitialContext();  
Object ref = jndiContext.lookup("AgLocacaoRemote");  
AgLocacaoRemote agente = (AgLocacaoRemote)  
    PortableRemoteObject.narrow  
    (ref, AgLocacaoRemote.class);  
  
...  
Locacao locacao = agente.criarLocacao(...);
```

Segurança



- Os servidores Enterprise JavaBeans suportam três tipos de segurança:
 - Autenticação
 - Validar a identidade do usuário
 - Suporte a vários mecanismos (tela de login, cartões de identificação digital, certificados de segurança, etc.)
 - Autorização
 - Diretivas definem o que os usuários podem, ou não, fazer
 - Comunicação segura
 - Definição de um canal de comunicação seguro