

# INE5646 – Programação para Web

Unidade III – Seção VI

## Acesso a Sistemas Legados

Prof. Frank Siqueira – Turma A

Prof. Leandro Komosinski – Turma B

# Conteúdo

- Sistemas Legados
- Abordagens para Integração
- Tecnologias de Integração
  - *Java Connectors*

# Sistemas Legados

- Definição de “Legado”:  
“Algo que gerações passadas transmitiram às atuais”
- Definição de “Sistemas Legados”:  
“São sistemas computacionais antigos, por vezes ultrapassados, mas que apesar disso fornecem serviços necessários para o funcionamento de organizações.”

# Sistemas Legados

- Exemplos:
  - ERP (*Enterprise Resource Planning*): gerenciam os recursos operacionais de uma organização
  - CRM (*Customer Relationship Management*): gerenciam as relações com clientes da empresa
  - Sistemas de banco de dados: armazenam dados vitais para o funcionamento da empresa
  - Diversos outros sistemas vitais para a atividade de uma organização, como sistemas de reservas de passagens e de hotéis, sistemas de automação de força de vendas, etc.

# Sistemas Legados

- Sistemas legados impõem vários desafios para desenvolvedores e administradores de sistemas:
  - O sistema pode ter sido desenvolvido utilizando tecnologias obsoletas, para as quais não há mais suporte, manutenção e pessoal qualificado
  - Os responsáveis pela criação e manutenção do sistema legado podem já ter deixado a empresa
  - O sistema pode não ter sido bem documentado, ou a documentação pode ter se perdido com o tempo
  - Equipamentos nos quais rodam os sistemas legados podem ser incompatíveis com novos softwares

# Sistemas Legados

- Organizações podem manter Sistemas Legados em funcionamento por várias razões:
  - O sistema funciona bem. Não há porquê trocá-lo.
  - Os custos para troca do sistema são proibitivos.
  - Treinamento de pessoal para usar o novo sistema seria caro e levaria a perda de horas de trabalho.
  - O sistema é vital para o trabalho da organização. Construir outro sistema confiável seria difícil.
  - A reconstrução do sistema seria muito complexa por falta de conhecimento sobre o sistema (não há documentação, desenvolvedores já saíram da empresa, ...)

# Abordagens para Integração

- Captura de Tela (*Screen Scrapping*)
  - Usado na integração com sistemas legados com interface de texto e comandos pelo teclado
  - A interface de texto do sistema legado pode ser substituída por uma interface Web mais amigável
  - Necessita de código de integração entre a aplicação e o sistema legado
  - As ações do usuário executadas na aplicação resultam no envio de comandos ao sistema legado
  - As saídas geradas pelo sist. legado são capturadas, processadas e exibidas na aplicação Web

# Abordagens para Integração

- Integração Ponto a Ponto
  - Consiste em utilizar uma interface para integração com o sistema legado, que pode ser fornecida por seu fabricante ou criada pelo desenvolvedor
  - Códigos de integração são adicionados à aplicação e ao sistema legado, criando um canal de comunicação para envio de dados e comandos
  - Com isso, uma aplicação Web pode enviar comandos ao sistema legado, e dados oriundos deste podem ser exibidos na aplicação Web



# Abordagens para Integração

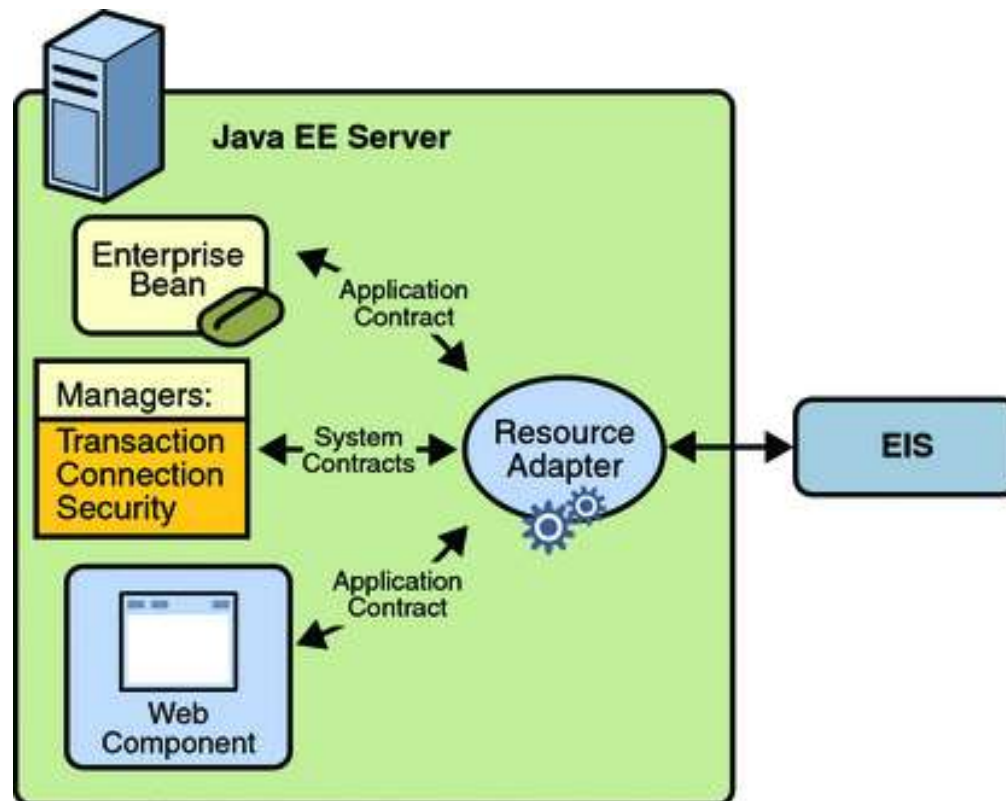
- Empacotamento OO (*OO Wrapping*)
  - Consiste em desenvolver um objeto empacotador (*wrapper*) para "encapsular" o sistema legado
  - Os métodos do objeto empacotador permitem a execução das funcionalidades do sistema legado
  - O empacotador interage com o sistema legado através de uma interface nativa fornecida por ele
  - O empacotador permite que as funcionalidades do sistema legado sejam invocadas a partir do código de uma aplicação orientada a objetos

# Tecnologias de Integração

- Várias tecnologias podem ser usadas para integração com sistemas legados, entre elas:
  - *Sockets*: comandos podem ser enviados para uma interface de integração ponto a ponto provida pelo sistema legado
  - CORBA: sistema legado pode ser encapsulado como um objeto CORBA e invocado remotamente
  - *Web Services*: podem encapsular um sistema legado, dando a ele uma interface SOAP ou REST
  - *Java EE Connector*: permite que aplicações Java EE interajam com sistemas legados usando um adaptador fornecido pelo seu fabricante ou por terceiros

# Tecnologias de Integração

- *Java EE Connector Architecture (JCA)*



# Tecnologias de Integração

- *Java EE Connector*
  - O *Resource Adapter* intermedia a comunicação entre o EIS e a aplicação Web
  - A JCA provê gerenciamento de transações, conexões e segurança
  - Contratos de serviço devem ser respeitados
  - Formas de comunicação possíveis:
    - *Outbound*: aplicação → EIS
    - *Inbound*: EIS → aplicação
    - Bidirecional: fluxo nos dois sentidos

# Tecnologias de Integração

- *Java EE Connector API*
  - Define um conjunto de classes e interfaces usadas no desenvolvimento de conectores
    - *Connection*: representa conexões com o sistema legado
    - *ConnectionFactory*: usada para criação de conexões
    - *ConnectionSpec*: propriedades da conexão
    - *Interaction*: modela uma interação com o sist. legado
    - *InteractionSpec*: propriedades de uma interação
    - *Record*: representa dados trocados com o sist. legado
    - *RecordFactory*: usada para criar conjuntos de dados

# Tecnologias de Integração

- *Java EE Connector* – Exemplo de Uso:

```
int count;
ConnectionSpec spec = new CciConnectionSpec(user, passwd);
Connection con = cf.getConnection(spec);
Interaction ix = con.createInteraction();
CciInteractionSpec iSpec = new CciInteractionSpec();
iSpec.setSchema(user);
iSpec.setFunctionName("EMPLOYEECOUNT");
RecordFactory rf = cf.getRecordFactory();
IndexedRecord iRec = rf.createIndexedRecord("InputRecord");
Record rec = ix.execute(iSpec, iRec);
Iterator iter = ((IndexedRecord)rec).iterator();
while(iter.hasNext()) {
    Object obj = iter.next();
    if(obj instanceof Integer)
        count = ((Integer)obj).intValue();
}
con.close();
System.out.println("Employee count is: " + count);
```