

## Exercício final – EJB 3

Deseja-se criar uma plataforma de gestão de avaliações para atender todos os alunos dos cursos de pós-graduação da FIAP. Essa plataforma será utilizada para aplicar as avaliações eletrônicas ao final de cada disciplina. Conforme mencionado, esse sistema proposto será adotado inicialmente nos cursos de pós-graduação mas, em seguida será estendido para todos os cursos da FIAP e de outras instituições de ensino. Este deverá suportar a quantidade de alunos das mesmas.

Estima-se que, num primeiro momento, serão realizadas 3.000 avaliações por mês. Mas, ao final do projeto, a estimativa é de aproximadamente 200.000 acessos mensais para a realização das provas eletrônicas.

O escopo inicial do projeto é criar uma estrutura para suportar somente avaliações com questões de múltipla escolha. Os demais tipos de avaliação serão implementados num segundo momento.

A interface do sistema não será única. Certamente a maioria dos acessos será realizado via web mas estão previstos também acessos via aplicação Java Swing/AWT e, principalmente, integrações com outras plataformas (Flex, Dot Net, Mobile, etc.).

Sendo assim o sistema deverá suportar requisitos básicos como escalabilidade, disponibilidade e desempenho. Todos os dados deverão, obrigatoriamente, ser armazenados em banco de dados ([disponibilizado pelo professor](#)).

Para testar a funcionalidade proposta deverão ser construídas 02 interfaces para recuperar as questões com suas devidas alternativas de resposta (uma para consumo RMI/IIOP e outra via SOAP/WS).

Para o cliente RMI/IIOP, o resultado deve ser algo parecido com a saída abaixo. Utilize o Console ([System.out.println](#)) para testar a chamada ao EJB.

1- Qual o tipo de objeto resultante da execução de um comando SQL usando a API JDBC?

- 1- Prepared Statement
- 2- Statement
- 3- ResultSet
- 4- Collection
- 5- Transaction

2- Qual o método do objeto Connection é usado para desabilitar a transação automática?

- 1- setAutoCommit(false)
- 2- setAutoCommit(true)
- 3- setTransaction(true)
- 4- setTransaction(false)
- 5- setRollback()

A interface **Avaliacao.java** e os objetos envolvidos na construção do serviço de obtenção de questões estão disponibilizados no projeto **avaliacao** com os nomes de **Questao.java** e **Resposta.java**.

```
package br.com.fiap;

import java.util.List;

public interface Avaliacao {

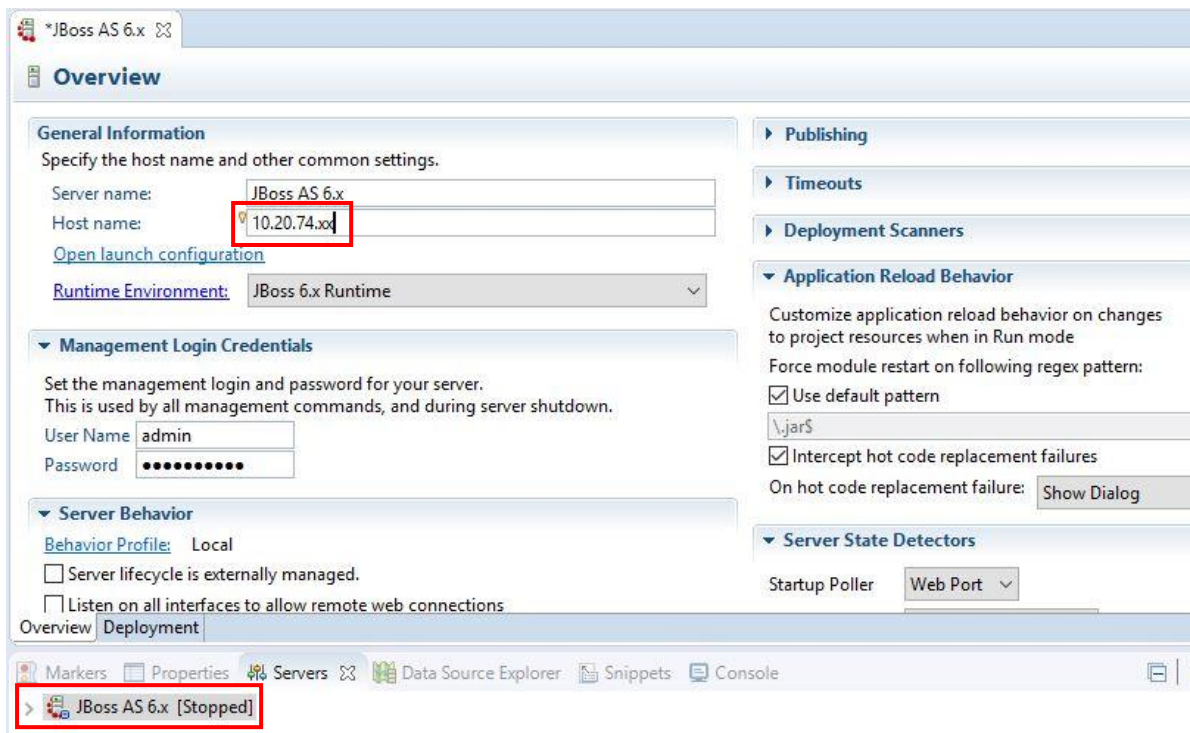
    public List obterQuestoes(int codigoAvaliacao);
    public void removerEJB();
}
```

### Método de avaliação:

1. Encaminhar ao professor o IP da máquina e o nome JNDI do EJB criado.
2. Encaminhar ao professor o caminho do WebService que disponibiliza a lista de questões/respostas da avaliação de código "1" previamente cadastrada no banco de dados MySql.
3. Disponibilizar o projeto no Git e enviar a URL (fazer upload .txt com a URL).
4. **(opcional)** Compactar o projeto do eclipse (exceto a pasta .metadata) juntamente com o arquivo de configuração mysql-ds.xml e fazer o upload no portal do aluno.

### Observações:

1. Não altere o nome dos pacotes e nem adicione anotações na interface Avaliacao.java. Para fazer a referência na classe do EJB utilize a anotação @Remote.
2. Para testes remotos favor substituir a entrada 'localhost' pelo IP da máquina no campo identificado abaixo dentro das propriedades do servidor no Eclipse.



\*JBoss AS 6.x

### Overview

**General Information**  
Specify the host name and other common settings.

Server name: JBoss AS 6.x

Host name: 10.20.74.xd

[Open launch configuration](#)

[Runtime Environment:](#) JBoss 6.x Runtime

**Management Login Credentials**  
Set the management login and password for your server.  
This is used by all management commands, and during server shutdown.

User Name: admin

Password: .....

**Server Behavior**  
[Behavior Profile:](#) Local

☐ Server lifecycle is externally managed.

☐ Listen on all interfaces to allow remote web connections

**Publishing**

**Timeouts**

**Deployment Scanners**

**Application Reload Behavior**  
Customize application reload behavior on changes to project resources when in Run mode  
Force module restart on following regex pattern:

☒ Use default pattern

\\.jar\$

☒ Intercept hot code replacement failures

On hot code replacement failure: Show Dialog

**Server State Detectors**

Startup Poller: Web Port

Overview | Deployment

Markers | Properties | Servers | Data Source Explorer | Snippets | Console

> JBoss AS 6.x [Stopped]