#### **Programação Cliente Servidor Aula 08**

# **Mapeamento Objeto-Relacional**

Os bancos de dados relacionais são a tecnologia de armazenamento mais consolidada no mercado, mas como pudemos observar, a forma de tratar as operações sobre o banco envolve a criação de classes que viabilizem a organização e reúso do código podendo ser facilmente padronizadas; é com base nesta premissa que surge a ideia por trás do **mapeamento objeto-relacional**.

Também chamado de **ORM** (do inglês *Object-Relational Mapping*), a técnica envolve uma reinterpretação dos bancos relacionais, fazendo com que as tabelas sejam representadas por classes e os registros destas tabelas, por instâncias destas classes.

# **Java Persistence API**

Um dos maiores avanços do Java foi a definição do **JPA** (Java Persistence API), o qual unifica os diversos frameworks de persistência em uma arquitetura única baseada em código anotado e apenas um arquivo de configuração, o **persistence.xml**.

**Persistence.xml**

responsável pela configuração da conexão, indicando a URL do banco, driver JDBC e framework utilizados

**JPQL** (Java Persistence Query Language)

bastante semelhante ao **SQL**, mas trata com objetos e coleções, ao invés de conjuntos de tuplas.

# **Enterprise Java Beans**

Um Enterprise Java Bean (**EJB**) é um componente corporativo utilizado de forma indireta, dentro de um ambiente de objetos distribuídos

O primeiro tipo de EJB a ser considerado é o de sessão, responsável por efetuar processos de negócios de forma síncrona. Ele pode assumir três comportamentos diferentes:

Stateless

Quando não guarda valores entre chamadas sucessivas. .Utilizamos **Stateless** quando não precisamos de nenhuma informação de processos anteriores ao corrente

Sateful

guarda valores entre chamadas sucessivas. Já o **Sateful** deve ser utilizado quando precisamos de informações anteriores

Singleton

Quando utiliza apenas uma instância por JVM. Quanto ao **Singleton**, ele é utilizado quando queremos compartilhar dados entre todos os usuários conectados ao aplicativo

1. Em termos de programação para banco de dados, em linguagens orientadas a objetos, é bastante aconselhável efetuar uma transformação dos registros para instâncias de classes de entidade. Este processo ficou conhecido como:

X a) ORM

b) POO

c) RUP

d) AUML

e) SQL

2. No uso de JPA, qual classe fica responsável por inserir os dados da entidade no banco de dados, através do método persist?

a) EntityManagerFactory

b) Query

c) CriteriaQuery

d) NamedQuery

X e) EntityManager

3. Existem diversos tipos de EJBs, mas um deles é voltado exclusivamente para o tratamento de mensagens recebidas a partir de mensagerias. Qual é este tipo de EJB?

a) *Stateless Session Bean*

b) *Message Driven Bean*

c) *Stateful Session Bean*

d) *Singleton Bean*

X e) *Entity Bean*

|  |  |
| --- | --- |
| |  | | --- | | Sobre POJO é correto afirma que: | |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | É uma classe sem métodos de negócios, mas com atributos definidos de forma publica e métodos de acesso público. |
|  |  | É uma classe com métodos de negócios e com atributos definidos de forma privada e métodos de acesso privado. |
|  |  | É uma classe com métodos de negócios e atributos definidos de forma privada e métodos de acesso público. |
| Certo |  | É uma classe sem métodos de negócios, mas com atributos definidos de forma privada e métodos de acesso público. |
|  |  | É uma classe sem métodos de negócios, mas com atributos definidos de forma privada e métodos de acesso privado. |

|  |  |
| --- | --- |
| |  | | --- | | O tipo de bean utilizado para tratamento de regras de negócio síncronas, onde ocorre a necessidade de guardar informações entre chamadas sucessivas seria: | |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | Message Driven Bean |
|  |  | Async Queued Bean |
| Certo |  | Stateful Session Bean |
|  |  | Entity Bean |
|  |  | Stateless Session Bean |

|  |  |
| --- | --- |
| |  | | --- | | Os EJBs do tipo Session Bean podem ser divididos em três tipos, sendo que um deles permite a ocorrência de apenas uma instância por máquina virtual. Qual é este tipo? | |
|  |

|  |  |  |
| --- | --- | --- |
| Certo |  | Singleton |
|  |  | Stateful |
|  |  | Message Driven |
|  |  | Stateless |
|  |  | Entity |

|  |  |
| --- | --- |
| |  | | --- | | Entre os diferente tipos de EJBs, os MDBs são responsáveis pelo comportamento assíncrono baseado em mensagerias. Assinale a alternativa correta com relação aos MDBs. | |
|  |

|  |  |  |
| --- | --- | --- |
| Certo |  | Apresentam um único método para tratamento de mensagens denominado onMessage. |
|  |  | Eles podem guardar informações de estado, como os Statefull Session Beans. |
|  |  | Necessitam de interfaces @Local e @Remote. |
|  |  | Precisam implementar a interface SessionListener. |
|  |  | São definidos pela anotação @Message. |

|  |  |
| --- | --- |
| |  | | --- | | O que é JPA? | |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | É uma API Padrão da linguagem Python que descreve uma interface comum para frameworks de persistência de dados. |
|  |  | É uma API Padrão da linguagem Ruby que descreve uma interface comum para frameworks de persistência de dados. |
|  |  | É uma API Padrão da linguagem PHP que descreve uma interface comum para frameworks de persistência de dados. |
| Certo |  | É uma API Padrão da linguagem Java que descreve uma interface comum para frameworks de persistência de dados. |
|  |  | É uma API Padrão da linguagem SQL que descreve uma interface comum para frameworks de persistência de dados. |

|  |  |
| --- | --- |
| |  | | --- | | Complete as lacunas no trecho de código-fonte abaixo com os termos corretos:  \_\_\_\_\_\_\_\_\_\_ emf = Persistence.createEntityManagerFactory("PU");       \_\_\_\_\_\_\_\_\_\_ em = emf.createEntityManager();      em.getTransaction().begin();     c = new Curso(3, "EE");     em.\_\_\_\_\_\_\_\_\_\_(c);   em.getTransaction().commit(); | |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | EntityManager; Command; insert |
|  |  | Database; Entity; put |
| Certo |  | EntityManagerFactory; EntityManager; persist |
|  |  | Connection; Statement; execute |
|  |  | EntityFactory; EntityManager; save |

|  |  |
| --- | --- |
| |  | | --- | | Sobre as redes de Petri é correto afirmar que: | |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | Formam uma linguagem de programação. |
| Certo |  | São uma representação matemática para sistemas distribuídos discretos. |
|  |  | Definem graficamente a estrutura de um sistema distribuído como um autômato. |
|  |  | Possuem nós de posição, nós de transição, e arcos não-direcionados conectando posições com transições |
|  |  | Define um grafo estocástico de elementos essenciais |

|  |
| --- |
|  |