PROVAS DE TI

TUDO PARA VOCÊ PASSAR

JPA e HIBERNATE

Rodrigo Macedo – rcbm539@gmail.com

http://www.itnerante.com.br/profile/RodrigoMacedo

gabaritou <TI>

Apresentação

- Formado em Sistemas de Informação pela Unibalsas – Faculdade de Balsas – MA em 2014.
- Pós graduado em Tecnologia para Educação à Distância pela Funbralu - Fundacao Brasil Ludovicense em 2015.
- Perfil no itnerante: http://www.itnerante.com.br/profile/RodrigoMacedo.

Bibliografia

- http://hibernate.org/orm/documentation/5.1/

- Ebok JPA e Hibernate – Algaworks

Pré-Requisitos

Público - Alvo

Ementa

Curso JPA e Hibernate	
1° Bateria de questões	JPA – Conceitos gerais.
2° Bateria de questões	JPA – Ciclo de vida.
3° Bateria de questões	JPA – ORM e JPQL.
4° Bateria de questões	Hibernate – Conceitos gerais.
5° Bateria de questões	Hibernate – Configuração e Mapeamento.
6° Bateria de questões	Hibernate – Consulta de dados.
7° Bateria de questões	Hibernate – Ciclo de vida.
8° Bateria de questões	Hibernate – Carga de dados.
9° Bateria de questões	Hibernate – Questões extras.

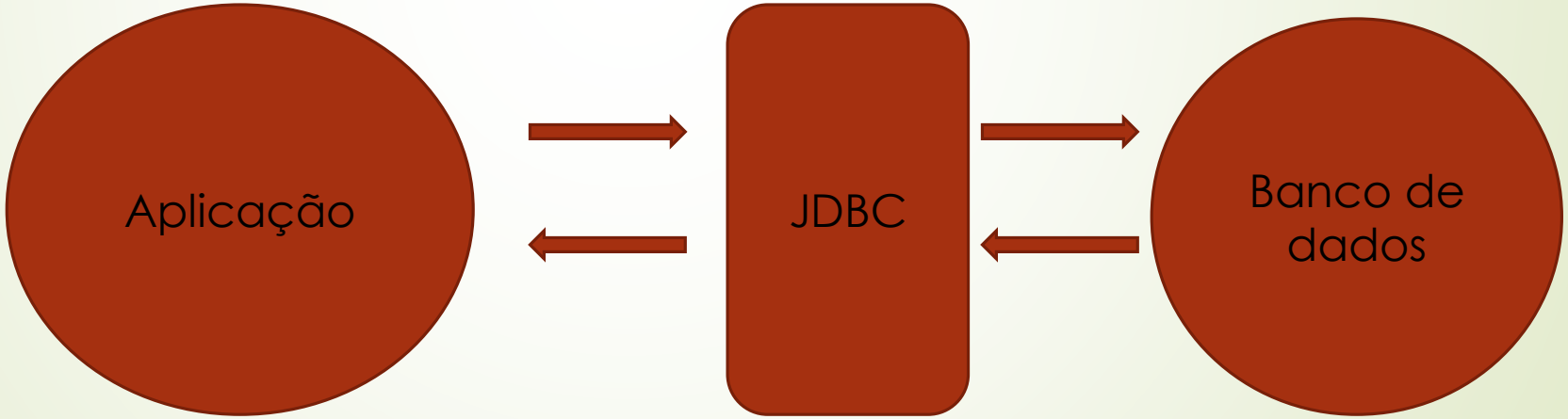

LET'S GO!
shutterstock
IMAGE ID: 113388922
www.shutterstock.com

Persistência de dados no Java

- Conceito muito fundamental que é aplicado em diferentes softwares.
- Formas de persistir dados:
  - XML, arquivos de texto, JSON, DB.
- Formas de persistir dados nos banco de dados relacionais em Java:
  - JDBC.
  - JPA.

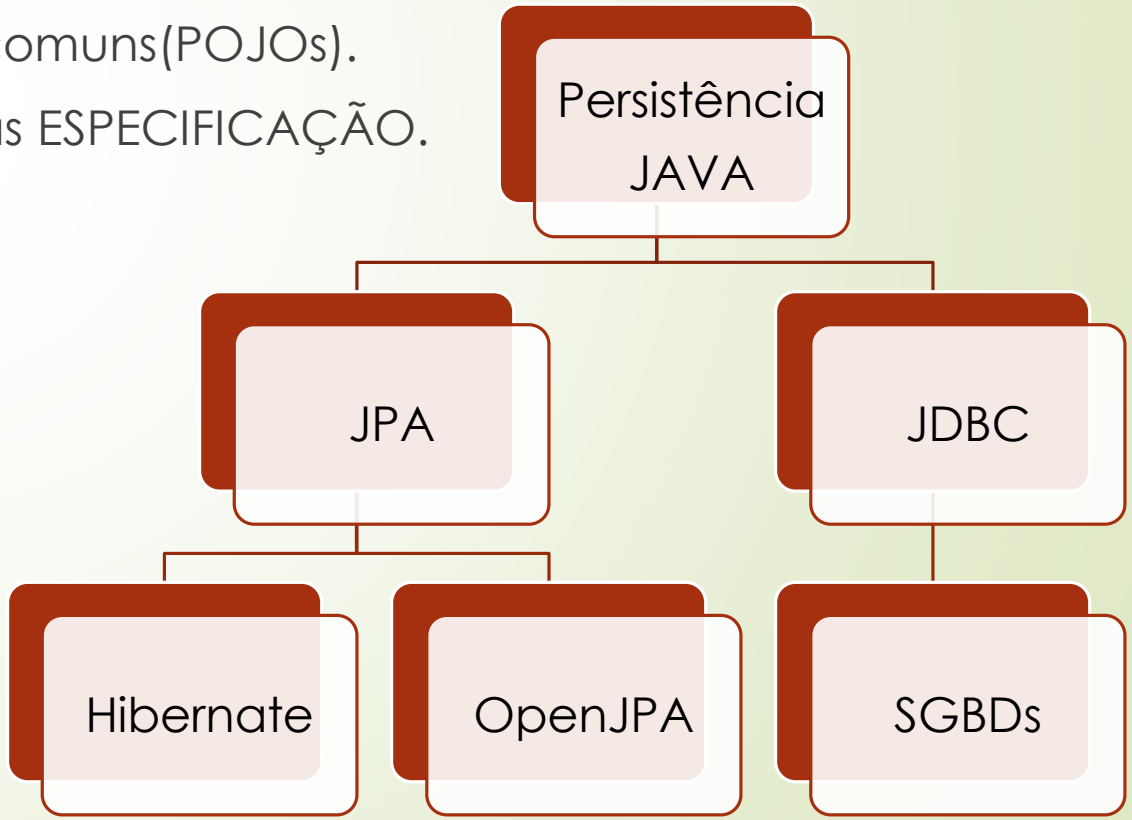
JDBC

- API padrão da linguagem Java para conectar aplicações Java aos diversos bancos de dados relacionais.
- Utiliza driver externo para comunicação dos diferentes SGBDs.
- Dependência direta dos SGBD's.
- Muito utilizado nas aplicações standalone Java – Java SE.
- Interação direta entre o SGBD e aplicação Java.



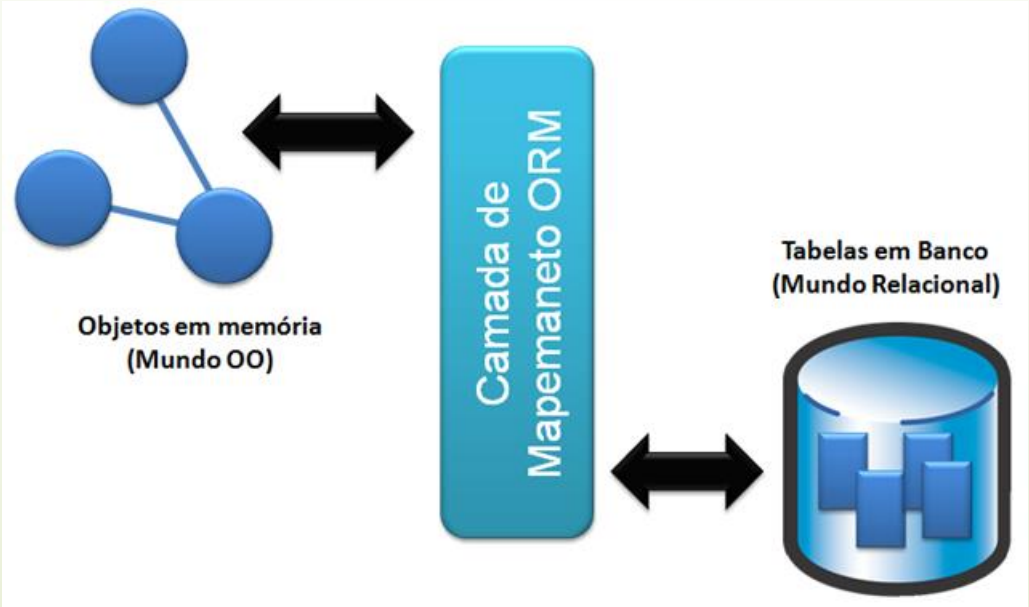
JPA – Conceitos Fundamentais

- Alternativa ao JDBC.
- Há uma alternativa ao utilizar SQL diretamente na aplicação.
- Persistência através de um mapeamento objeto – relacional (ORM).
- Provê esse mapeamento automático, independente de SGBD.
- Baseado em objetos simples e comuns(POJOs).
- Não é IMPLEMENTAÇÃO, apenas ESPECIFICAÇÃO.

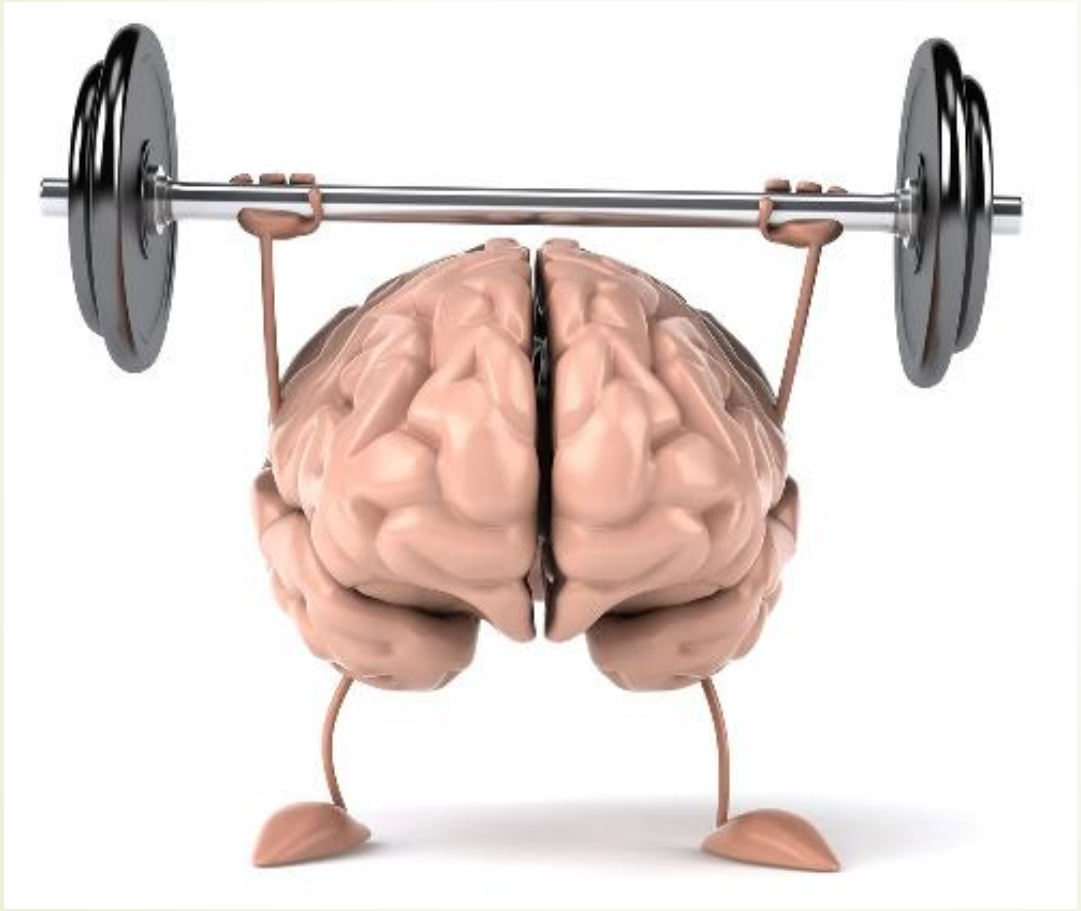


ORM – Mapeamento objeto - relacional

Modelo Relacional	Modelo OO
Tabela	Classe
Linha / Tupla	Objeto
Coluna	Atributo
Chave estrangeira	Associação



Primeira Bateria de Questões



Q1 - [CESPE TRE MS 2013] Assinale a opção correspondente ao elemento que, além de ser utilizado para definir um meio de mapeamento objeto-relacional para objetos

Java simples e comuns (POJOs), denominados beans de entidade, também é utilizado para gerenciar o desenvolvimento de entidades do modelo relacional em plataforma nativa Java SE e Java EE.

a) JSF

b) SVN

c) JPA

d) spring

e) Struts

Q1 - [CESPE TRE MS 2013] Assinale a opção correspondente ao elemento que, além de ser utilizado para definir um meio de mapeamento objeto-relacional para objetos

Java simples e comuns (POJOs), denominados beans de entidade, também é utilizado para gerenciar o desenvolvimento de entidades do modelo relacional em plataforma nativa Java SE e Java EE.

a) JSF

b) SVN

c) JPA

d) spring

e) Struts

Q2 - [CESPE CPPM 2013] Java persistence API (JPA) é uma solução para persistência de dados, utilizada, inclusive, quando há mapeamento do modelo orientado a objeto para bancos de dados relacionais.

Q3 - [CESPE TRT8 2016] Para persistir objetos Java a partir do framework com base em POJOS (plain old Java objects), utiliza-se

a)TDD (test driven development).

b)refactoring.

c)JSF (JavaServer faces).

d)struts.

e)JPA.

Q2 - [CESPE CPPM 2013] Java persistence API (JPA) é uma solução para persistência de dados, utilizada, inclusive, quando há mapeamento do modelo orientado a objeto para bancos de dados relacionais. CERTO.

Q3 - [CESPE TRT8 2016] Para persistir objetos Java a partir do framework com base em POJOS (plain old Java objects), utiliza-se

a)TDD (test driven development).

b)refactoring.

c)JSF (JavaServer faces).

d)struts.

e)JPA.

Q4 - [FCC 2007 TRE/SE] Sendo um grupo de classes e componentes responsáveis pelo armazenamento e recuperação de dados, esta camada inclui necessariamente um modelo das entidades do domínio de negócios (mesmo que seja somente um modelo de metadados). No âmbito do mapeamento objeto relacional (hibernate) esta é a camada de

A) negócio.
B) restrição.
C) apresentação.
D) consistência.
E) persistência.

Q4 - [FCC 2007 TRE/SE] Sendo um grupo de classes e componentes responsáveis pelo armazenamento e recuperação de dados, esta camada inclui necessariamente um modelo das entidades do domínio de negócios (mesmo que seja somente um modelo de metadados). No âmbito do mapeamento objeto relacional (hibernate) esta é a camada de

A) negócio.
B) restrição.
C) apresentação.
D) consistência.
E) persistência.

Q5 - [MPE RS 2012] O conjunto de classes e interfaces, escrito em JAVA, que faz o envio de instruções SQL para qualquer banco de dados relacional é denominado de

A) JPA.
B) JDBC.
C) Web.
D) XML.
E) HTML.

Q5 - [MPE RS 2012] O conjunto de classes e interfaces, escrito em JAVA, que faz o envio de instruções SQL para qualquer banco de dados relacional é denominado de

A) JPA.
B) JDBC.
C) Web.
D) XML.
E) HTML.

Q6 - [CESPE CNJ 2013] Os objetos mapeados na linguagem Java que devem ser persistidos como objetos precisam utilizar JPA (Java persistence API), pois o JPA permite realizar o mapeamento objeto/relacional automatizado e transparente e sua persistência em um banco de dados relacional.

Q7 - [CESPE TRE MS 2013 ADAP] JPA é um framework MVC de aplicações web que se destina a simplificar o desenvolvimento de interfaces de usuário embasadas em web.

Q8 - [CESPE TJ RO 2012 ADAP] JPA, um framework utilizado na camada de persistência, define uma forma para mapear POJO (plain old Java objects) para um banco de dados.

Q6 - [CESPE CNJ 2013] Os objetos mapeados na linguagem Java que devem ser persistidos como objetos precisam utilizar JPA (Java persistence API), pois o JPA permite realizar o mapeamento objeto/relacional automatizado e transparente e sua persistência em um banco de dados relacional. CERTO.

Q7 - [CESPE TRE MS 2013 ADAP GAB E] JPA é um framework MVC de aplicações web que se destina a simplificar o desenvolvimento de interfaces de usuário embasadas em web. ERRADO.

Q8- [CESPE TJ RO 2012 ADAP] JPA, um framework utilizado na camada de persistência, define uma forma para mapear POJO (plain old Java objects) para um banco de dados. CERTO.

Q9 - [IESES TRE MA 2015] Considere as seguintes afirmativas:

I. É a API padrão de mercado para conectividade entre a linguagem JAVA e vários tipos de bases de dados. Usando essa API, pode-se acessar praticamente qualquer fonte

de dados, de bases de dados relacionais a planilhas.

II. É uma API que oferece um meio de mapeamento objeto/relacional para que desenvolvedores JAVA gerenciem dados relacionais em aplicações JAVA.

As afirmativas I e II podem estar corretamente se referindo, respectivamente, às APIs:

a) JNDI e Hibenate.

b) JNDI e JPA.

c) JTA e JSR.

d) JDBC e JPA.

Q9 - [IESES TRE MA 2015] Considere as seguintes afirmativas:

I. É a API padrão de mercado para conectividade entre a linguagem JAVA e vários tipos de bases de dados. Usando essa API, pode-se acessar praticamente qualquer fonte

de dados, de bases de dados relacionais a planilhas.

II. É uma API que oferece um meio de mapeamento objeto/relacional para que desenvolvedores JAVA gerenciem dados relacionais em aplicações JAVA.

As afirmativas I e II podem estar corretamente se referindo, respectivamente, às APIs:

a) JNDI e Hibenate.

b) JNDI e JPA.

c) JTA e JSR.

d) JDBC e JPA.

GABARITO

Q1 – LETRA C.

Q2 – C (CERTO).

Q3 – LETRA E.

Q4 – LETRA E.

Q5 – LETRA B.

Q6 – C (CERTO).

Q7 – E (ERRADO).

Q8 - C (CERTO).

Q9- LETRA D.

JPA – Aprofundando

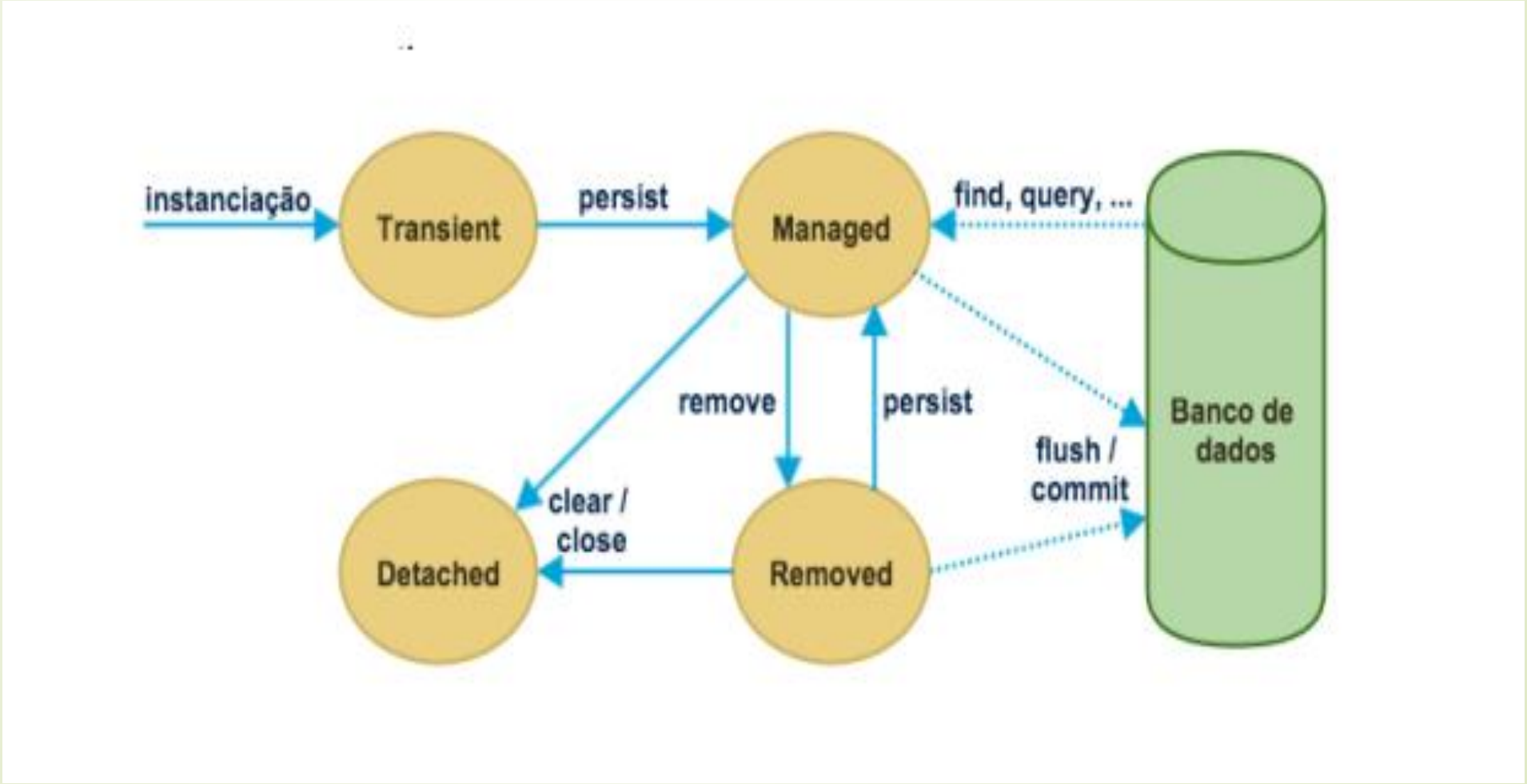
- Iniciou como parte da especificação JSR 220.
- Está na sua versão 2.x.
  - Versão 2.0 – especificação JSR 317.
  - Versão 2.1 – especificação JSR 338
- Utiliza duas formas para mapear os objetos:
  - Via XML.
  - Via anotação: podem ser representadas por anotações par-valor.
- Surgiu na especificação 3.0 do EJB substituindo os Entity Beans. Mas não se limitou à especificação EJB.
- Formas de mapeamento:
  - Top down: Objetos java para registros no banco de dados relacional.
  - Bottom up: Registros no banco de dados relacional para objetos java.
- Especificação é suportada em plataformas Java SE e Java EE.
- Presente nativamente no Java a partir da versão 1.5.

Data Binding

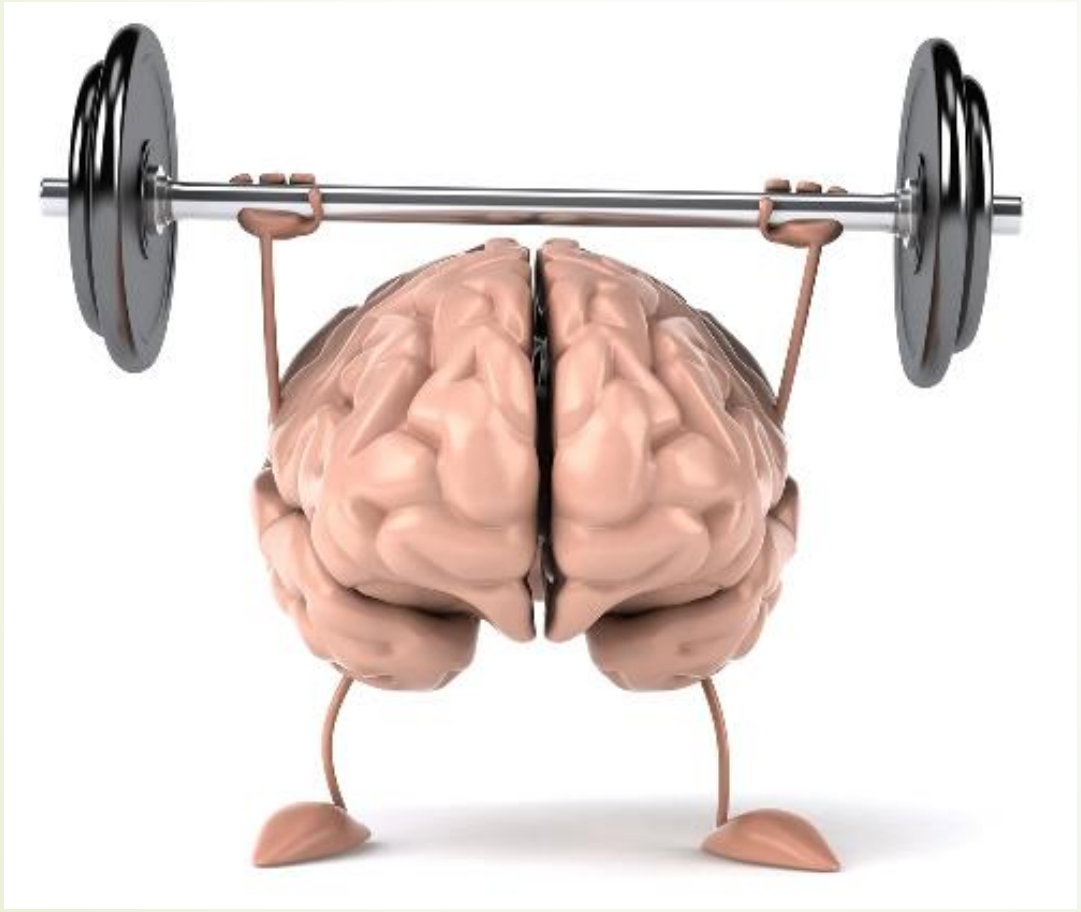
- Controle de binding especificado em: **javax.persistence.FetchType.**
- Estabelece conexão entres os dados de uma associação.
- O JPA permite duas formas de carregamento de dados:
  - **Lazy load.**
  - **Eager load**.

Ciclo de Vida

- **@Transient :** Objetos que ainda não estão associados ao banco de dados. Não estão nem sequer instanciados.
  - Para tanto devem ser instanciados usando o operador *new*.
- **@Managed:** Estão sempre associados a um contexto de persistência.
  - Qualquer alteração nos objetos são refletidos na base de dados. Sincronismo direto.
- **@Detached:** Continua existindo a instância do objeto, porém o gerenciador de entidades está fechado. Não há mais sincronia com o banco de dados.
- **@Removed:** É um objeto marcado para ser removido e o é no momento da sincronização física com o banco de dados.


instanciação
Transient
persist
Managed
find, query, ...
Banco de dados
remove
persist
flush / commit
clear / close
Detached
Removed

Segunda Bateria de Questões



Q10 – [CESPE TRE MS 2013 ADAP] JPA lida com a forma como dados relacionais são mapeados para objetos Java e com a forma como esses objetos são armazenados em um banco de dados relacional.

Q11 - [CESPE TJ SE 2014] A JPA, que foi criada como alternativa para o Hibernate para conexão com os sistemas gerenciadores de banco de dados, está nativa no Java SE a partir da versão 1.3.

Q12 - [CESPE TRE PI 2016] Assinale a opção correspondente à enumeração correta usada para o controle de binding em JPA.

a)javax.persistence.CascadeTime

b)javax.persistence.FetchType

c)javax.persistence.ElementCollection

d)javax.persistence.ElementBinding

e)javax.persistence.ObjectBinding

Q10 – [CESPE TRE MS 2013 ADAP] JPA lida com a forma como dados relacionais são mapeados para objetos Java e com a forma como esses objetos são armazenados em um banco de dados relacional. CERTO.

Q11 - [CESPE TJ SE 2014] A JPA, que foi criada como alternativa para o Hibernate para conexão com os sistemas gerenciadores de banco de dados, está nativa no Java SE a partir da versão 1.3. ERRADO.

Q12 - [CESPE TRE PI 2016] Assinale a opção correspondente à enumeração correta usada para o controle de binding em JPA.

a)javax.persistence.CascadeTime

b)javax.persistence.FetchType

c)javax.persistence.ElementCollection

d)javax.persistence.ElementBinding

e)javax.persistence.ObjectBinding

Q13 - [FCC TRT 19 2011] Os estados do ciclo de vida de uma instância de uma entidade, definidos na JPA 2.0, são .

a)novo (new), gerenciado (managed), destacado (detached) e removido (removed).

b)ativo (active), inativo (inactive) e removido (removed).

c)novo (new), temporário (temporary), permanente (permanent) e destacado (detached).

d)novo (new), temporário (temporary) e destacado (detached)

e)gerenciado (managed), temporário (temporary), permanente (permanent) e destacado (detached).

Q13 - [FCC TRT 19 2011] Os estados do ciclo de vida de uma instância de uma entidade, definidos na JPA 2.0, são .

a)novo (new), gerenciado (managed), destacado (detached) e removido (removed).

b)ativo (active), inativo (inactive) e removido (removed).

c)novo (new), temporário (temporary), permanente (permanent) e destacado (detached).

d)novo (new), temporário (temporary) e destacado (detached)

e)gerenciado (managed), temporário (temporary), permanente (permanent) e destacado (detached).

Q14 - [FEPES JUCESC 2013] Em relação à JPA e Hibernate, considere as seguintes afrmativas.

1. JPA Especifca uma JSR
2. Hibernate Especifca uma JSR
3. Hibernate cuida da camada de persistência enquanto JPA da camada de transação
4. Hibernate é uma implementações de JSR
5. JPA é uma Implementação de JSR .

Assinale a alternativa que indica todas as afirmativas corretas.

a)São corretas apenas as afrmativas 1 e 4.

b)São corretas apenas as afrmativas 2 e 3.

c)São corretas apenas as afrmativas 3 e 4.

d)São corretas apenas as afrmativas 1, 2 e 3.

e)São corretas apenas as afrmativas 3, 4 e 5

Q14 - [FEPES JUCESC 2013] Em relação à JPA e Hibernate, considere as seguintes afrmativas.

1. JPA Especifca uma JSR
2. Hibernate Especifca uma JSR
3. Hibernate cuida da camada de persistência enquanto JPA da camada de transação
4. Hibernate é uma implementações de JSR
5. JPA é uma Implementação de JSR .

Assinale a alternativa que indica todas as afirmativas corretas.

a)São corretas apenas as afrmativas 1 e 4.

b)São corretas apenas as afrmativas 2 e 3.

c)São corretas apenas as afrmativas 3 e 4.

d)São corretas apenas as afrmativas 1, 2 e 3.

e)São corretas apenas as afrmativas 3, 4 e 5

Q15 - [FCC 2011 TCE PR] A JPA

a)pode ser usada fora de componentes EJB e fora da plataforma Java EE, em aplicações Java SE.

b)utiliza persistência gerenciada por contêiner (CMP), ou seja, as classes de entidade e persistência necessitam de um contêiner presente em um servidor de aplicações

para serem executadas.

c)utiliza descritores XML para especificar informações do mapeamento relacional de objeto, mas não oferece suporte a anotações.

d)suporta consultas dinâmicas nomeadas nas classes de entidade que são acessadas apenas por instruções SQL nativas.

e)possui uma interface EntityBeans que padroniza operações Create Read Update Delete (CRUD) que envolvem tabelas.

Q15 - [FCC 2011 TCE PR] A JPA

a)pode ser usada fora de componentes EJB e fora da plataforma Java EE, em aplicações Java SE.

b)utiliza persistência gerenciada por contêiner (CMP), ou seja, as classes de entidade e persistência necessitam de um contêiner presente em um servidor de aplicações

para serem executadas.

c)utiliza descritores XML para especificar informações do mapeamento relacional de objeto, mas não oferece suporte a anotações.

d)suporta consultas dinâmicas nomeadas nas classes de entidade que são acessadas apenas por instruções SQL nativas.

e)possui uma interface EntityBeans que padroniza operações Create Read Update Delete (CRUD) que envolvem tabelas.

Q16 - [FCC TJ PE 2012] Em uma classe de entidade de uma aplicação que utiliza JPA, a anotação que define um atributo que não será salvo no banco de dados é a

a)@Optional.

b)@Transient.

c)@Stateless.

d)@Stateful.

e)@Local.

Q16 - [FCC TJ PE 2012] Em uma classe de entidade de uma aplicação que utiliza JPA, a anotação que define um atributo que não será salvo no banco de dados é a

a)@Optional.

b)@Transient.

c)@Stateless.

d)@Stateful.

e)@Local.

Q17 - [FCC TRE PB 2015] Em aplicações que usam JPA, quando um objeto de uma classe de entidade do banco de dados é instanciado pela primeira vez ele está no estado novo e para passá-lo para o estado gerenciado, utiliza-se um método da interface EntityManager chamado:

a)datashed.

b)managed.

c)persist.

d)flush.

e)Commit.

Q17 - [FCC TRE PB 2015] Em aplicações que usam JPA, quando um objeto de uma classe de entidade do banco de dados é instanciado pela primeira vez ele está no estado novo e para passá-lo para o estado gerenciado, utiliza-se um método da interface EntityManager chamado:

a)datashed.

b)managed.

c)persist.

d)flush.

e)Commit.

GABARITO

Q10 - C (CERTO).

Q11 - E (ERRADO).

Q12 - LETRA B

Q13 - LETRA A

Q14 - LETRA A

Q15 - LETRA A

Q16 - LETRA B

Q17 - LETRA C

ORM

Mapeamento ORM - JPA

- **@Entity**: Principal anotação JPA. Mapeia toda a classe como uma entidade no banco de dados.
- **@Table**: Utiliza-se caso queira especificar o nome da tabela no banco de dados que seja diferente do nome da classe

```java
@Id
@GeneratedValue
public Long getCodigo() {
    return codigo;
}
```

```java
@Entity
@Table(name = "tab_veiculo")
public class Veiculo {

    private Long codigo;
    private String fabricante;
    private String modelo;
    private Integer anoFabricacao;
    private Integer anoModelo;
```

- **@Id:** Define o atributo java como chave primária na tabela que se está mapeando.
  - **@Column**: Define atributos relacionados aos atributos no BD.
    - Name: Nome do atributo
    - Nullable: Pode ou não ser nulo. Valor booleano.
- **@GeneratedValue**: Atribui para esse id a propriedade autoincremento.
  - **Strategy= GenerationType.Auto**: Padrão.
  - **Strategy= GenerationType.Identity:** Valores gerados exclusivos por hierarquia.
  - **Strategy= GenerationType.Sequence:** Deve-se informar o valor da propriedade generator. Esse generator pode ser um @SequenceGenerator ou @TableGenerator.

- **@Temporal**: Utilizado para campos como data, hora e etc.
  - TemporalType.Date.
  - TemporalType.Time.
  - TemporalType.Timestamp.
- **@Version**: Especifica a versão de um campo a fim de verificar integridade deste.
- **@Transient**: É muito utilizado para trazer informações que não devem estar persistidas no banco de dados.
- **@NamedQuery**: Consulta definida estaticamente numa string de consulta.
  - Melhora a organização do código.
  - Permite reutilizar queries.
  - Reforça o uso de parâmetros em consultas.

Criação da NamedQuery:

1 - @NamedQuery(

2 - name="findAllEmployeesByFirstName",

3 - queryString="SELECT OBJECT(emp) FROM Employee emp WHERE emp.firstName = :firstname")

Utilização da NamedQuery:

1 - Query queryEmployeesByFirstName = em.createNamedQuery("findAllEmployeesByFirstName");

2 - queryEmployeeByFirstName.setParameter("firstName", "John");

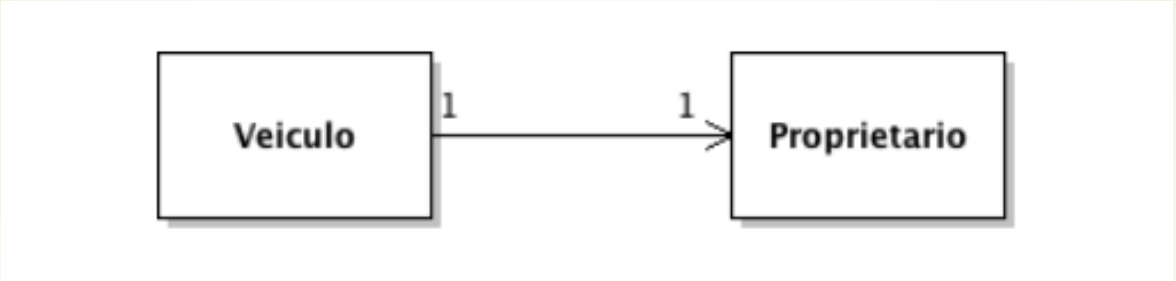
3 - Collection employees = queryEmployessByFirstName.getResultList()

Entity Class

- Deve ser uma classe de nível superior, ou seja não pode ser classe aninhada.
- Construtor público ou protegido sem argumentos.
- Métodos e variáveis de instância devem prezar pelo encapsulamento.
- Não pode ser final:
  - A classe.
  - Os métodos.
  - E as variáveis de instância.

Relacionamentos ORM

- **@OneToOne**: Relação de um para um. Cada entidade apenas referencia uma ocorrência da outra entidade.



- **Associação Unidirecional**: Permite-se obter as ocorrências das entidades em uma única direção.
- **Associação Bidirecional**: Permite-se obter o veículo a partir do proprietário, e o proprietário a partir do veículo.
  - Para isso deve-se utilizar a propriedade **mappedBy** dentro da anotação.
  - O valor dessa propriedade deve ser igual ao nome que é referenciado que se usa numa entidade para se associar a outra.
- Por padrão a chave estrangeira é nomeada pelo **nome do atributo da associação + '_' nome do atributo identificador da entidade destino**.
  - Utiliza-se a notação @JoinColumn para personalizar o nome desta chave estrangeira.

Veículo:

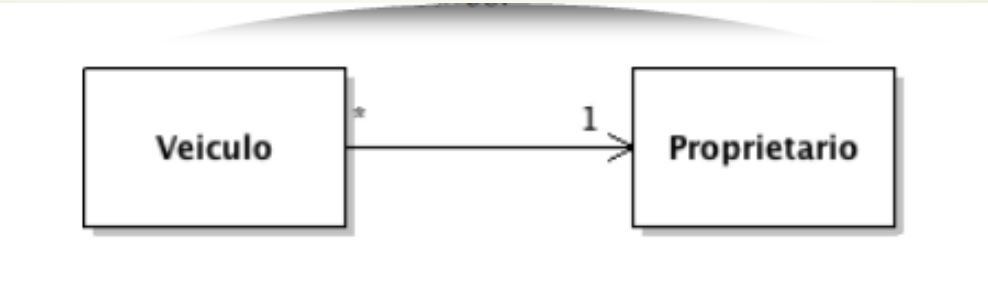
```
1 - @Entity
2 - @Table(name="veiculo")
3 - public class Veiculo{
4 -  private Proprietario proprietario;
5 -   @OneToOne
6 -  public Proprietario getProprietario(){
7 -   return proprietario;
  }
}
```

Proprietário:

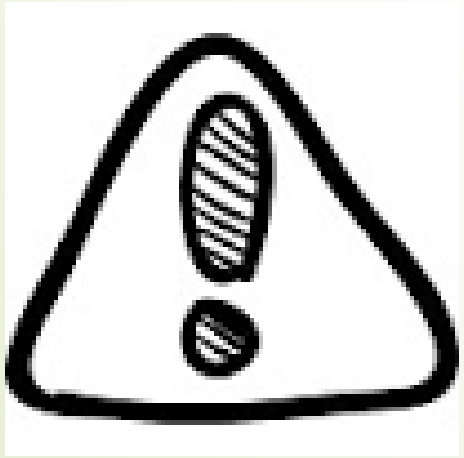
```
1 - @Entity
2 - @Table(name="proprietario")
3 - public class Proprietario{
4 -   private Veiculo veiculo;
5 - @OneToOne(mappedBy="proprietario")
6 -     public Veiculo getVeiculo(){
7 -         return veiculo;
    }
}
```

- A propriedade mappedBy é o segredo para os relacionamentos bidirecionais dentro das anotações de relacionamentos ORM.

- **@ManyToOne**: Relação de muitos para um. Cada entidade pode ser referenciada N vezes.



- **@OneToMany:** Relação de um para muitos. Cada entidade pode referenciar N outras entidades.
- Nesses dois casos só é possível também obter as ocorrências das entidades pela entidade do lado N ou "muitos".
  - Para resolver isso basta utilizar a propriedade mappedBy.



A utilização do MappedBy vai além do relacionamento definido na anotação. Tem a ver com o sentido da relação ser unidirecional ou bidirecional.

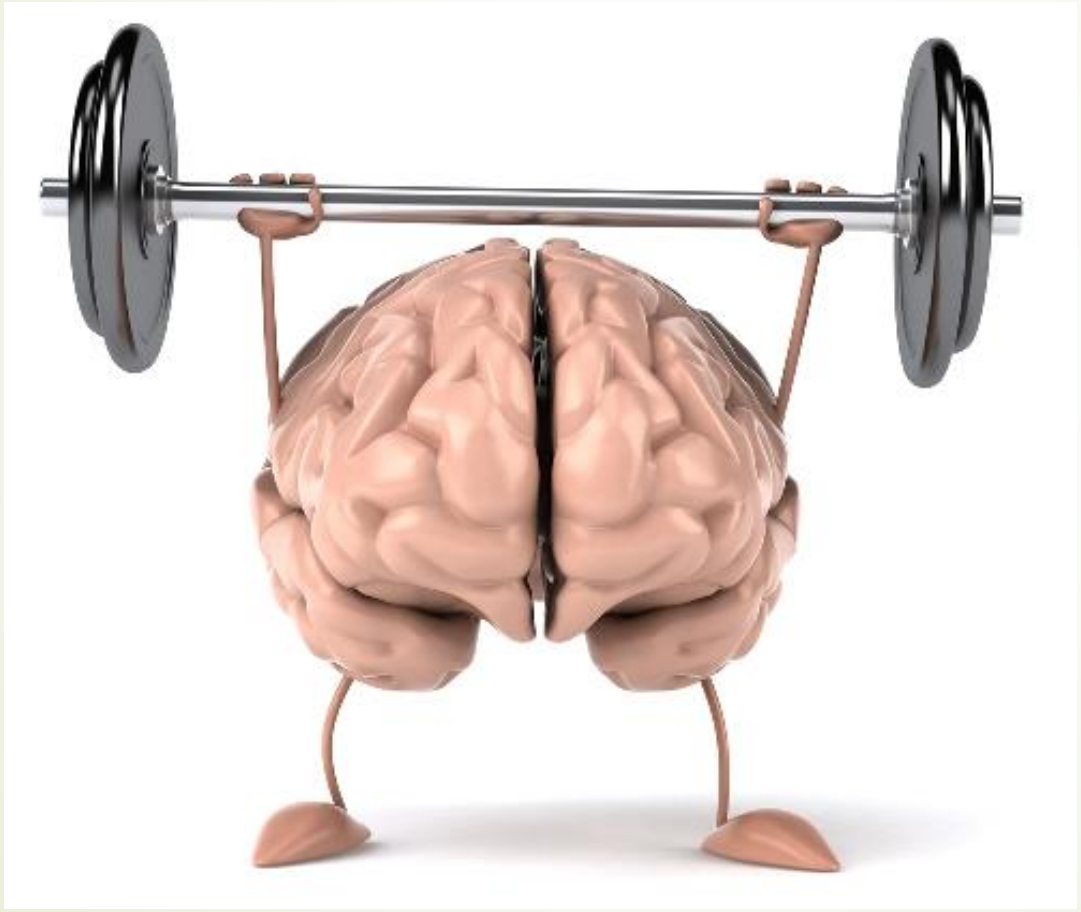
- **@ManyToMany**: Relação de muitos para muitos. Ambas entidades podem se referenciar N vezes.

- Por padrão os relacionamentos ManyToMany criam uma tabela adicional, enquanto os outros apenas se adicionava a chave estrangeira do lado "muitos", esse cria uma tabela adicional com a chave primárias das duas entidades.
- Por padrão vem a tabela vem nomeada com o **nome da primeira entidade** + "_" **+ nome da segunda entidade. Ex:** veiculo_acessorio.
  - Pode-se personalizar o nome através da propriedade name da anotação @JoinTable.
  - Pode-se personalizar o nome das chaves primárias desta tabela:
    - joinColumns = **@JoinColumn**(name = "")
    - inverseJoinColumns = **@JoinColumn**(name = "")
- OBS: **@ManyToMany não garante ao relacionamento que ele seja bidirecional.**

JPQL

- O JPA possui vários métodos que já fazem consultas prontas.
- Porém em casos de consultas específicas deve-se utilizar o recurso que a JPA chama de JPQL.
- Consulta ocorre nas entidades mapeadas não nas tabelas.
- Sintaxe bem próxima ao SQL.
- Se enquadra dentro da DML, nada de DDL muito menos DCL.
- Consultas portáveis, independente de SGBD.
- Pode se utilizar a consulta partindo do from. Não precisa de select.
- Dá suporte a utilização de parâmetros nomeados:
  - Evita-se que os valores informados sejam concatenados numa String, sendo alvo fácil de SQL Injection.
  - Ex: "from Veiculo where codigo = :id". Onde código é um campo da tabela de veículo e :id é o parâmetro nomeado nesta consulta./
  - O método setParameter é o responsável pela passagem de parâmetros. Os parâmetros desse método primeiro é o nome do parâmetro e outro o valor.
  - Por último, após a criação da query e da passagem de parâmetros, basta utilizar o método getResultList().

Terceira Bateria de Questões



Q18 - [CESPE TRE GO 2015] Ao se declarar uma coluna que seja a chave primária de uma tabela, é necessário utilizar a anotação @Id.

Q19 - [CESPE TRE GO 2015] A anotação @Entity significa que determinada classe Java é uma entidade do banco de dados. Caso essa entidade tenha nome que não seja o da tabela, será necessário utilizar a anotação @Table.

Q20 - [FCC TRT 23 2011] Considere:

Em relação à JPA (Java Persistence API) é **INCORRETO** afirmar que

a)@NamedQuery é aplicada para definir várias consultas.

b)@Entity define que haverá correspondência da classe com uma tabela do banco de dados.

c)@Id define que o atributo que está mapeado com tal anotação corresponderá à chave primária da tabela.

d)@Column(name = "id", nullable = false) define que o atributo da classe mapeado com tal anotação deve estar associado à coluna cujo nome é "id", além de definir que tal campo não pode ser nulo.

e)@OneToMany indica que o atributo contém um conjunto de entidades que a referenciam.

Q18 - [CESPE TRE GO 2015] Ao se declarar uma coluna que seja a chave primária de uma tabela, é necessário utilizar a anotação @Id..

Q19 - [CESPE TRE GO 2015] A anotação @Entity significa que determinada classe Java é uma entidade do banco de dados. Caso essa entidade tenha nome que não seja o da tabela, será necessário utilizar a anotação @Table. CERTO.

Q20 - [FCC TRT 23 2011] Considere:

Em relação à JPA (Java Persistence API) é INCORRETO afirmar que

a)@NamedQuery é aplicada para definir várias consultas.

b)@Entity define que haverá correspondência da classe com uma tabela do banco de dados.

c)@Id define que o atributo que está mapeado com tal anotação corresponderá à chave primária da tabela.

d)@Column(name = "id", nullable = false) define que o atributo da classe mapeado com tal anotação deve estar associado à coluna cujo nome é "id", além de definir que tal campo não pode ser nulo.

e)@OneToMany indica que o atributo contém um conjunto de entidades que a referenciam.

Q21 - [BIO-RIO IFRJ 2015] Com relação aos conceitos de JPA, avalie se são verdadeiras (V) ou falsas (F) as afirmativas a seguir:

- Até a versão J2EE 1.4, a plataforma Java não possuía uma forma simples de mapear objetos em um banco de dados, uma forma mais complexa podia ser utilizada, exigindo um contêiner EJB. Com a JPA (Java Persistence API), houve uma padronização do mapeamento de objeto/relacional na plataforma Java.
- A JPA é baseada no conceito POJO (Pain Old Java Object) em que os objetos persistentes são denominados entidades que são objetos que representam conjuntos de dados persistidos no banco. Como entidades não são definidas por classes Java comuns, sem relação com frameworks ou bibliotecas, elas podem ser abstratas ou herdar de outras classes, sem restrições.
- As classes e interfaces da JPA estão localizadas no pacote javax.persistence, com isso pode-se fazer o mapeamento da aplicação, utilizando anotações. Neste contexto, uma entidade é rotulada pela anotação @Entity, uma tabela por @Table, a chave primária por @Id e cada coluna por @Column.

As afirmativas são respectivamente:

a)F, V e F

b)V, F e V.

c)V, F e F.

d)V, V e F.

e)F, F e V.

Q21 - [BIO-RIO IFRJ 2015] Com relação aos conceitos de JPA, avalie se são verdadeiras (V) ou falsas (F) as afirmativas a seguir:

- Até a versão J2EE 1.4, a plataforma Java não possuía uma forma simples de mapear objetos em um banco de dados, uma forma mais complexa podia ser utilizada, exigindo um contêiner EJB. Com a JPA (Java Persistence API), houve uma padronização do mapeamento de objeto/relacional na plataforma Java.
- A JPA é baseada no conceito POJO (Pain Old Java Object) em que os objetos persistentes são denominados entidades que são objetos que representam conjuntos de dados persistidos no banco. Como entidades não são definidas por classes Java comuns, sem relação com frameworks ou bibliotecas, elas podem ser abstratas ou herdar de outras classes, sem restrições.
- As classes e interfaces da JPA estão localizadas no pacote javax.persistence, com isso pode-se fazer o mapeamento da aplicação, utilizando anotações. Neste contexto, uma entidade é rotulada pela anotação @Entity, uma tabela por @Table, a chave primária por @Id e cada coluna por @Column.

As afirmativas são respectivamente:

a)F, V e F

b)V, F e V.

c)V, F e F.

d)V, V e F.

e)F, F e V.

Q22 - [CETAP MPCM 2015] Analise o código concernente à tecnologia JPA a seguir

```
1. @Entity
2. //inserir o código 1 aqui
3. public class Cliente {
4.
5. //inserir o código 2 aqui
6. private Long id;
7.
8. private String temporario;
9 }
```

Qual das opções a seguir pode, respectivamente, substituir o comentário 1 e 2 para produzir corretamente a classe Cliente?

a)@Table(name="cliente") / @Identificator

b)@Table(name="cliente") / @Id

c)@Table("cliente")/@Id

d)@Table("clÍente") / @Identificator

e)@Identificator/@Temp

Q22 - [CETAP MPCM 2015] Analise o código concernente à tecnologia JPA a seguir

```
1. @Entity
2. //inserir o código 1 aqui
3. public class Cliente {
4.
5. //inserir o código 2 aqui
6. private Long id;
7.
8. private String temporario;
9 }
```

Qual das opções a seguir pode, respectivamente, substituir o comentário 1 e 2 para produzir corretamente a classe Cliente?

a)@Table(name="cliente") / @Identificator

b)@Table(name="cliente") / @Id

c)@Table("cliente")/@Id

d)@Table("clÍente") / @Identificator

e)@Identificator/@Temp

Q23 - [FGV PROCEMPA 2014] Considere a seguinte classe com anotações JPA:

```
@Entity
    @Table(name=”funcionario”)
     public class Funcionario implements Serializable
     {
          private static final long serialVersionUID =
     2L;
         @Id
         @Column(name=”id”, nullable=false)
          private Integer id;
         @Column(name=”nome”)
          private String primaryKey;
         @ManyToOne
          private Funcionario chefe;
      //    Restante da classe...
      }
```

Sobre essa classe anotada, analise as afirmativas a seguir.

I. A anotação @Table é dispensável, neste caso.

II. A chave primária da tabela associada à classe Funcionario é nome.

III. A anotação @ManyToOne introduz, neste exemplo, um autorrelacionamento.

Após o exame das afirmativas, verifica-se que

a)somente I e II são verdadeiras.

b)somente I e III são verdadeiras.

c)somente II e III são verdadeiras.

d)somente II é verdadeira.

e)somente I é verdadeira.

Sobre essa classe anotada, analise as afirmativas a seguir.

I. A anotação @Table é dispensável, neste caso.

II. A chave primária da tabela associada à classe Funcionario é nome.

III. A anotação @ManyToOne introduz, neste exemplo, um autorrelacionamento.

Após o exame das afirmativas, verifica-se que

a)somente I e II são verdadeiras.

b)somente I e III são verdadeiras.

c)somente II e III são verdadeiras.

d)somente II é verdadeira.

e)somente I é verdadeira.

Q24 - [CESPE STJ 2015] JPQL (Java Persistence Query Language) é uma linguagem de manipulação de dados adotada para criar, alterar estrutura de tabelas e gatilhos utilizados na especificação JPA (Java Persistence API).

Q25 - [CESPE TRE GO 2015] Para que uma classe Java efetue consultas em determinada entidade do banco de dados, é necessário elaborar o SQL e, depois, convertê-lo para JPQL (Java persistence query language).

Q26 - [FCC TRE AP 2015] Considere o fragmento de código a seguir, presente em uma classe ideal de acesso a dados de uma aplicação que utiliza JPA.

```
String jpql = "select e from Empregado e where e.cargo = :c";
Query q = entityManager.createQuery(jpql, Empregado.class);
...I...
List <Empregados> empregados = q.getResultList ();
```

Para completar corretamente o fragmento de código de forma que a consulta retorne os empregados cujo cargo seja Gerente, a lacuna I deve ser preenchida por

a)q.setString("c", "Gerente");

b)q.setParameter("cargo", "Gerente");

c)q.setString(c, "Gerente");

d)q.setAttribute(c, "Gerente");

e)q.setParameter("c", "Gerente");

Q24 - [CESPE STJ 2015] JPQL (Java Persistence Query Language) é uma linguagem de manipulação de dados adotada para criar, alterar estrutura de tabelas e gatilhos utilizados na especificação JPA (Java Persistence API). ERRADO.

Q25 - [CESPE TRE GO 2015] Para que uma classe Java efetue consultas em determinada entidade do banco de dados, é necessário elaborar o SQL e, depois, convertê-lo para JPQL (Java persistence query language). ERRADO.

Q26 - [FCC TRE AP 2015] Considere o fragmento de código a seguir, presente em uma classe ideal de acesso a dados de uma aplicação que utiliza JPA.

```
String jpql = "select e from Empregado e where e.cargo = :c";
Query q = entityManager.createQuery(jpql, Empregado.class);
...I...
List <Empregados> empregados = q.getResultList ();
```

Para completar corretamente o fragmento de código de forma que a consulta retorne os empregados cujo cargo seja Gerente, a lacuna I deve ser preenchida por

a)q.setString("c", "Gerente");

b)q.setParameter("cargo", "Gerente");

c)q.setString(c, "Gerente");

d)q.setAttribute(c, "Gerente");

e)q.setParameter("c", "Gerente");

Q27 - [NC-UFPR COPEL 2015] Em relação ao mapeamento objeto-relacional usando JPA (Java Persistence API) 2.0, assinale a alternativa correta.

a)Quando se usa a anotação @OneToOne(mappedBy="pai"), entende-se que a chave estrangeira desse mapeamento aponta para uma tabela chamada pai.

b)A anotação @Temporal indica que esse atributo é temporário, ou seja, é um sinônimo da anotação @Transient.

c)Quando usamos a anotação @GeneratedValue(strategy=GenerationType.SEQUENCE), devemos informar o valor da propriedade generator, que pode apontar para um @TableGenerator ou @SequenceGenerator.

d)A anotação @Version está Deprecated e portanto não deve ser utilizada, já que entra em conflito com a JTA (Java Transaction API)

e)A anotação @ManyToMany indica que o relacionamento é bidirecional, e mesmo que seja informado em apenas uma das classes, será possível realizar a navegação (e obter suas respectivas coleções) em ambos os lados.

Q27 - [NC-UFPR COPEL 2015] Em relação ao mapeamento objeto-relacional usando JPA (Java Persistence API) 2.0, assinale a alternativa correta.

a)Quando se usa a anotação @OneToOne(mappedBy="pai"), entende-se que a chave estrangeira desse mapeamento aponta para uma tabela chamada pai.

b)A anotação @Temporal indica que esse atributo é temporário, ou seja, é um sinônimo da anotação @Transient.

c)Quando usamos a anotação @GeneratedValue(strategy=GenerationType.SEQUENCE), devemos informar o valor da propriedade generator, que pode apontar para um @TableGenerator ou @SequenceGenerator.

d)A anotação @Version está Deprecated e portanto não deve ser utilizada, já que entra em conflito com a JTA (Java Transaction API)

e)A anotação @ManyToMany indica que o relacionamento é bidirecional, e mesmo que seja informado em apenas uma das classes, será possível realizar a navegação (e obter suas respectivas coleções) em ambos os lados.

Q28 - [CESGRANRIO PETROBRAS 2012] Em aplicações Java Enterprise Edition 6, é comum o uso da API JPA. Nessa API, há o conceito de classe de entidade (entity class).Por definição, uma classe de entidade deve, obrigatoriamente, cumprir os seguintes requisitos, **EXCETO**

a)estar anotada com a anotação Entity ou representada em um descritor XML.

b)não ser qualificada com final.

c)ter as variáveis de instância persistentes qualificadas com private, protected, ou package-private.

d)ter ao menos um construtor, este sem argumentos (no-arg constructor).

e)ter o mesmo nome da tabela correspondente do banco de dados.

Q28 - [CESGRANRIO PETROBRAS 2012] Em aplicações Java Enterprise Edition 6, é comum o uso da API JPA. Nessa API, há o conceito de classe de entidade (entity class).Por definição, uma classe de entidade deve, obrigatoriamente, cumprir os seguintes requisitos, **EXCETO**

a)estar anotada com a anotação Entity ou representada em um descritor XML.

b)não ser qualificada com final.

c)ter as variáveis de instância persistentes qualificadas com private, protected, ou package-private.

d)ter ao menos um construtor, este sem argumentos (no-arg constructor).

e)ter o mesmo nome da tabela correspondente do banco de dados.

[Q29 - FCC TRE AP 2015] Em uma classe de entidade do banco de dados presente em uma aplicação que utiliza JPA existem as seguintes instruções:

```
@NamedQuery(name="Cliente.listarTodos",query="select c from Cliente c")
@Entity
public class Cliente {
}
```

Considere que os atributos e métodos da classe Cliente estão implementados e mapeados adequadamente para a tabela Cliente do banco de dados.Em uma classe de acesso a dados da mesma aplicação, que possui um objeto em válido do tipo EntityManager, para executar a query da classe de entidade Cliente e obter os dados retornados em uma lista, utiliza-se:

a)
```
Query query = em.createNamedQuery("Cliente.listarTodos",Cliente.class);
    List <Clientes> clientes = query.getResultList();
```

b)
```
ResultSet query = em.createQuery("Cliente.listarTodos",Cliente.class);
   ArrayList <Clientes> clientes = query.getResultList();
```

c)
```
Query query = em.createNativeQuery("Cliente.listarTodos",Cliente.class);
    List <Clientes> clientes = query.getList();
```

d)
```
Query query = em.createQuery("Cliente.listarTodos",Cliente.class);
    List <Clientes> clientes = query.getResultList();
```

e)
```
List query = em.createNamedQuery("Cliente.listarTodos",Cliente.class);
   ArrayList clientes = query.getResultSet();
```

[Q29 - FCC TRE AP 2015] Em uma classe de entidade do banco de dados presente em uma aplicação que utiliza JPA existem as seguintes instruções:

```
@NamedQuery(name="Cliente.listarTodos",query="select c from Cliente c")
@Entity
public class Cliente {
}
```

Considere que os atributos e métodos da classe Cliente estão implementados e mapeados adequadamente para a tabela Cliente do banco de dados.Em uma classe de acesso a dados da mesma aplicação, que possui um objeto em válido do tipo EntityManager, para executar a query da classe de entidade Cliente e obter os dados retornados em uma lista, utiliza-se:

a)
```
Query query = em.createNamedQuery("Cliente.listarTodos",Cliente.class);
 List <Clientes> clientes = query.getResultList();
```

b)
```
ResultSet query = em.createQuery("Cliente.listarTodos",Cliente.class);
 ArrayList <Clientes> clientes = query.getResultList();
```

c)
```
Query query = em.createNativeQuery("Cliente.listarTodos",Cliente.class);
 List <Clientes> clientes = query.getList();
```

d)
```
Query query = em.createQuery("Cliente.listarTodos",Cliente.class);
 List <Clientes> clientes = query.getResultList();
```

e)
```
List query = em.createNamedQuery("Cliente.listarTodos",Cliente.class);
 ArrayList clientes = query.getResultSet();
```

[Q30 - UERJ 2015] Java Persistence Query Language (JPQL) é uma linguagem de consulta que faz parte da especificação JPA. Considere uma aplicação em Java que usa JPA, na qual está definida uma classe de entidade denominada br.app.acme.Cliente.
Além disso, essa aplicação contém o trecho de código abaixo, que cria um objeto do tipo javax.persistence.Query, cuja referência é qry.

```
javax.persistence.Query qry = entityManager.createQuery(
            "SELECT OBJECT(c) FROM br.app.acme.Cliente c " +
            " WHERE c.uf = :uf");
qry.setParameter("uf", "Rio de Janeiro");
```

A expressão adequada para execução da consulta em JPQL representada pela referência qry é:

a) java.util.Collection clientes = qry.getResultList()

b) java.util.Collection clientes = qry.executeQuery()

c) br.app.acme.Cliente[] clientes = qry.getResultList()

d) java.util.Collection clientes = qry.getSingleResult()

[Q30 - UERJ 2015] Java Persistence Query Language (JPQL) é uma linguagem de consulta que faz parte da especificação JPA. Considere uma aplicação em Java que usa JPA, na qual está definida uma classe de entidade denominada br.app.acme.Cliente.
Além disso, essa aplicação contém o trecho de código abaixo, que cria um objeto do tipo javax.persistence.Query, cuja referência é qry.

```
javax.persistence.Query qry = entityManager.createQuery(
            "SELECT OBJECT(c) FROM br.app.acme.Cliente c " +
            " WHERE c.uf = :uf");
qry.setParameter("uf", "Rio de Janeiro");
```

A expressão adequada para execução da consulta em JPQL representada pela referência qry é:

a) java.util.Collection clientes = qry.getResultList()

b) java.util.Collection clientes = qry.executeQuery()

c) br.app.acme.Cliente[] clientes = qry.getResultList()

d) java.util.Collection clientes = qry.getSingleResult()

GABARITO

Q18 - C (CERTO).
Q19 - C (CERTO).
Q20 - LETRA A.
Q21 - LETRA B.
Q22 - LETRA B.
Q23 - LETRA B.
Q24 - E (ERRADO).
Q25 - E (ERRADO).
Q26 - LETRA E.
Q27 - LETRA C.
Q28 - LETRA E.
Q29 – LETRA A.
Q30 – LETRA A.


HIBERNATE

Informes

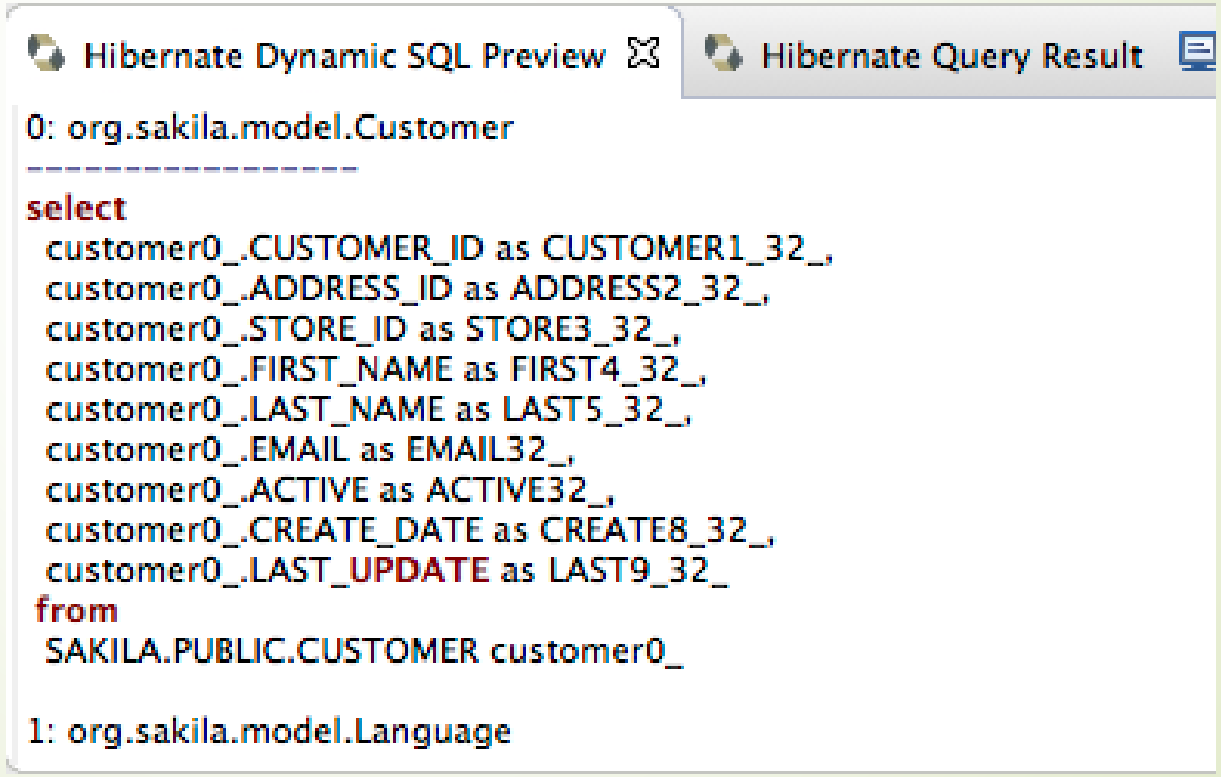
Bateria de Questões – Hibernate	
4 ° Bateria de questões	Questões conceitos gerais.
5 ° Bateria de questões	Configuração e Unidade de persistência.
6 ° Bateria de questões	Consulta de dados.
7 ° Bateria de questões	Ciclo de Vida.
8 ° Bateria de questões	Carga de dados.
9 ° Bateria de questões	Questões extras.

Conceitos Gerais

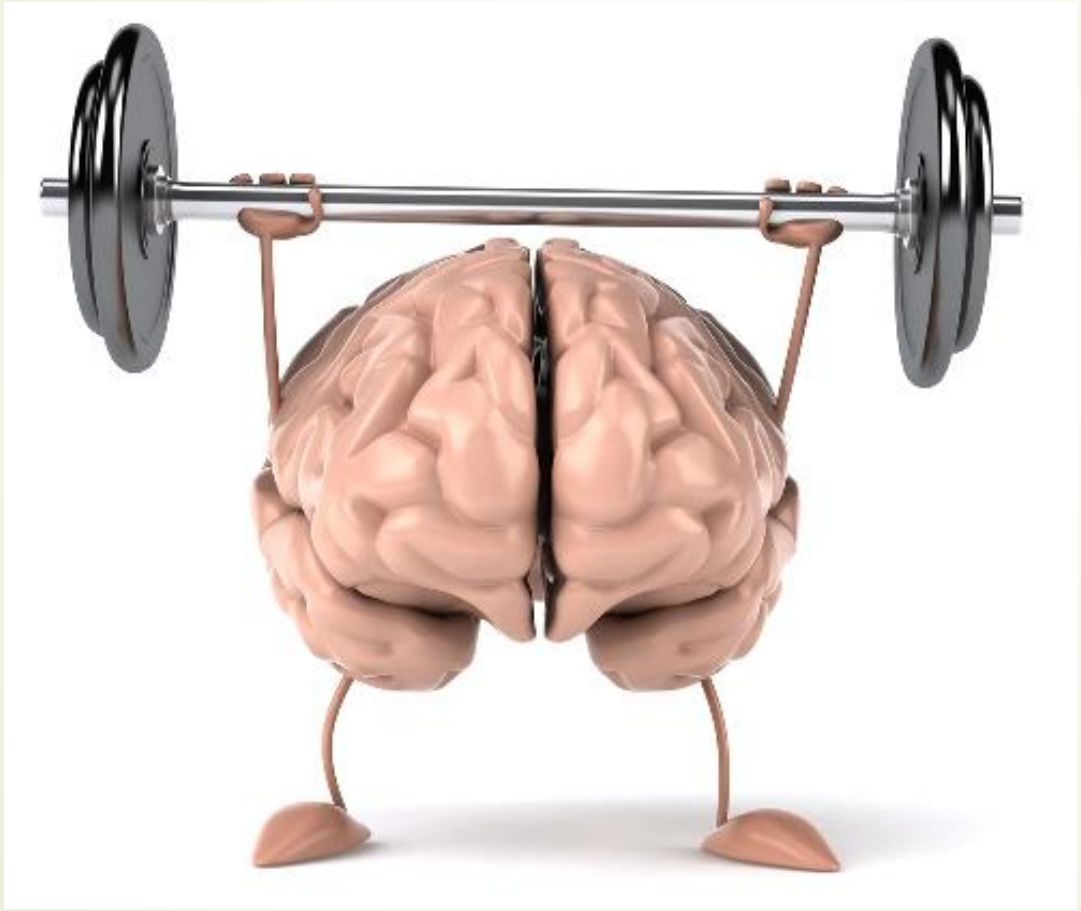
- Framework de implementação do ORM escrito em Java.
- Software – livre com liçenca LGPL.
- Existe a versão hibernate para .NET - NHibernate.
- Pode ser utilizado em aplicações standalone (JSE) e distribuídas (JEE).
- Num modelo arquitetural MVC representa a camada do modelo.
- Diminui complexidade entre aplicações Java que precisem trabalhar com banco de dados relacionais.
- Mapeia classes java em tabelas no banco de dados.
- Tecnologia de implementação da especificação JPA.
- Abordagem:
  - Top-Down
  - Bottom Up.

Utilitários

- **Hibernate Dynamic SQL Preview**: Mostra em tela a consulta SQL correspondente a query HQL implementada.
- **Hibernate Query Result**: Apresenta o resultado da query que foi informada no código.
- **Hibernate Query Cache**: Realiza o cache de consultas que são muito utilizadas.



Quarta Bateria de Questões



Q31 - [FCC TRT 19 2011] Em sua essência, o Hibernate é um framework para

a)mapeamento objeto-relacional (ORM).

b)desenvolvimento de aplicações de Internet rica (Rich Internet Application - RIA).

c)implementação da camada Controller do padrão MVC (Model - View - Controller).

d)construção de aplicações utilizando-se inversão de controle (IoC).

e)injeção de dependência (dependency injection) em aplicativos.

Q32 - [CESPE TRE MG 2009 ADAP]O Hibernate é um framework alternativo que integra os modelos orientados a objetos que não utilizam bancos de dados no

modelo relacional.

Q31 - [FCC TRT 19 2011] Em sua essência, o Hibernate é um framework para

a)mapeamento objeto-relacional (ORM).

b)desenvolvimento de aplicações de Internet rica (Rich Internet Application - RIA).

c)implementação da camada Controller do padrão MVC (Model - View - Controller).

d)construção de aplicações utilizando-se inversão de controle (IoC).

e)injeção de dependência (dependency injection) em aplicativos.

Q32 - [CESPE TRE MG 2009 ADAP]O Hibernate é um framework alternativo que integra os modelos orientados a objetos que não utilizam bancos de dados no modelo relacional. ERRADO.

Q33 - [CONSULPLAN TSE 2012] Por suas características, Hibernate 3.5 constitui uma ferramenta com a finalidade de realizar o seguinte tipo de mapeamento

a)objeto/relacional para Java.

b)gerencial/operacional para sites interativos textuais.

c)entidade/relacionamento para modelagem de dados.

d)lógico/físico para desenvolvimento por meio da prototipação.

Q34 - [CESPE TRE MS 2013 ADAP] Hibernate é um framework para mapeamento objeto/relacional para aplicações em Java que permite realizar, por meio de arquivos DHTML, o mapeamento de classes Java em tabelas do banco de dados relacionais e vice-versa.

Q33 - [CONSULPLAN TSE 2012] Por suas características, Hibernate 3.5 constitui uma ferramenta com a finalidade de realizar o seguinte tipo de mapeamento

a)objeto/relacional para Java.

b)gerencial/operacional para sites interativos textuais.

c)entidade/relacionamento para modelagem de dados.

d)lógico/físico para desenvolvimento por meio da prototipação.

Q34 - [CESPE TRE MS 2013 ADAP] Hibernate é um framework para mapeamento objeto/relacional para aplicações em Java que permite realizar, por meio de arquivos DHTML, o mapeamento de classes Java em tabelas do banco de dados relacionais e vice-versa. ERRADO.

Q35 - [FCC TRT 22 2010] Hibernate é um framework

a)que separa as funções que envolvem a construção de aplicações Web, através da associação dos eventos do lado cliente com os manipuladores dos eventos do lado do servidor.

b)pelo qual o programador utiliza a infraestrutura do servidor de aplicação voltada para o desenvolvimento de aplicações de missão crítica e de aplicações empresariais em geral.

c)no qual as questões de infraestrutura, segurança, disponibilidade e escalabilidade são responsabilidade do servidor de aplicações, permitindo que o programador se concentre, apenas, nas necessidades do negócio do cliente.

d)que permite ao desenvolvedor de páginas para internet produzir aplicações que acessem o banco de dados, manipulem arquivos no formato texto e capturem informações a partir de formulários.

e)cujo objetivo é diminuir a complexidade entre os programas Java que precisam trabalhar com um banco de dados do modelo relacional.

Q35 - [FCC TRT 22 2010] Hibernate é um framework

a)que separa as funções que envolvem a construção de aplicações Web, através da associação dos eventos do lado cliente com os manipuladores dos eventos do lado do servidor.

b)pelo qual o programador utiliza a infraestrutura do servidor de aplicação voltada para o desenvolvimento de aplicações de missão crítica e de aplicações empresariais em geral.

c)no qual as questões de infraestrutura, segurança, disponibilidade e escalabilidade são responsabilidade do servidor de aplicações, permitindo que o programador se concentre, apenas, nas necessidades do negócio do cliente.

d)que permite ao desenvolvedor de páginas para internet produzir aplicações que acessem o banco de dados, manipulem arquivos no formato texto e capturem informações a partir de formulários.

e)cujo objetivo é diminuir a complexidade entre os programas Java que precisam trabalhar com um banco de dados do modelo relacional.

Q36 - [ESAF 2015]

O Hibernate é um framework para o mapeamento objeto-relacional. Em relação ao Hibernate, é incorreto afirmar que:

a)o seu objetivo é diminuir a complexidade entre os programas Java que precisam trabalhar com um banco de dados do modelo relacional, em especial, no desenvolvimento de consulta e atualização de dados.

b)pode ser utilizado em aplicações Java standalone ou em aplicações Java EE, utilizando Servlet ou sessões Enterprise Java Beans.

c)possibilita desenvolver classes consistentes usando Java convencional.

d)se trata de um software livre de código aberto distribuído com a licença LGPL (GNU Lesser General Public License).

e)sua principal característica é a transformação das classes em Java para tabelas de dados (e dos tipos de dados Java para os da SQL).

Q36 - [ESAF 2015]

O Hibernate é um framework para o mapeamento objeto-relacional. Em relação ao Hibernate, é incorreto afirmar que:

a)o seu objetivo é diminuir a complexidade entre os programas Java que precisam trabalhar com um banco de dados do modelo relacional, em especial, no desenvolvimento de consulta e atualização de dados.

b)pode ser utilizado em aplicações Java standalone ou em aplicações Java EE, utilizando Servlet ou sessões Enterprise Java Beans.

c)possibilita desenvolver classes consistentes usando Java convencional.

d)se trata de um software livre de código aberto distribuído com a licença LGPL (GNU Lesser General Public License).

e)sua principal característica é a transformação das classes em Java para tabelas de dados (e dos tipos de dados Java para os da SQL).

Q37 - [FUNCAB PRODAM AM 2014] Muitos sistemas coorporativos atuais são desenvolvidos utilizando linguagem de programação orientada a objetos e banco de dados relacional. Para simplificar a utilização em conjunto desses dois diferentes paradigmas, pode ser aplicado um framework de mapeamento objeto relacional, como por exemplo:

a)Struts.

b)Hibernate

c)Spring.

d)Java Server Faces.

e)JQuery.

Q38 - [CESPE MEC 2015] O framework Hibernate, uma solução para se fazer o mapeamento objeto relacional (ORM) em ambientes Java, cuida do mapeamento de classes

para tabelas de banco de dados e de tipos de dados Java para tipos de dados SQL.

Q37 - [FUNCAB PRODAM AM 2014] Muitos sistemas coorporativos atuais são desenvolvidos utilizando linguagem de programação orientada a objetos e banco de dados relacional. Para simplificar a utilização em conjunto desses dois diferentes paradigmas, pode ser aplicado um framework de mapeamento objeto relacional, como por exemplo:

a)Struts.

b)Hibernate

c)Spring.

d)Java Server Faces.

e)JQuery.

Q38 - [CESPE MEC 2015] O framework Hibernate, uma solução para se fazer o mapeamento objeto relacional (ORM) em ambientes Java, cuida do mapeamento de classes

para tabelas de banco de dados e de tipos de dados Java para tipos de dados SQL. CERTO.

Q39 - [CESPE SECONT-ES 2009] O Hibernate, um framework de mapeamento objeto relacional (ORM), cria uma camada persistência na solução desenvolvida, o que permite ligar os objetos aos bancos de dados relacionais. Entre seus serviços, o Hibernate provê um meio de se controlar transações, por meio de métodos de suas interfaces session e transaction, tendo ainda suporte a herança e polimorfismo. É distribuído sob a licença LGPL, o que permite seu uso em projetos comerciais ou open source.

Q40 - [CESPE INPI 2013] No Hibernate, caso o nome da classe seja diferente do nome da tabela mapeada, é necessário informar, na anotação @Table, o nome da tabela, por meio do atributo name.

Q41 - [CESPE TRE BA 2010] O Hibernate, um framework para o mapeamento objeto- relacional, é escrito na linguagem Java e, por isso, somente pode ser executado no ambiente Java.

Q39 - [CESPE SECONT-ES 2009] O Hibernate, um framework de mapeamento objeto relacional (ORM), cria uma camada persistência na solução desenvolvida, o que permite ligar os objetos aos bancos de dados relacionais. Entre seus serviços, o Hibernate provê um meio de se controlar transações, por meio de métodos de suas interfaces session e transaction, tendo ainda suporte a herança e polimorfismo. É distribuído sob a licença LGPL, o que permite seu uso em projetos comerciais ou open source. CERTO.

Q40 - [CESPE INPI 2013] No Hibernate, caso o nome da classe seja diferente do nome da tabela mapeada, é necessário informar, na anotação @Table, o nome da tabela, por meio do atributo name. CERTO

Q41 - [CESPE TRE BA 2010] O Hibernate, um framework para o mapeamento objeto- relacional, é escrito na linguagem Java e, por isso, somente pode ser executado no ambiente Java. ERRADO.

Q42 - [CESPE ANATEL 2014] O Hibernate permite operações de banco de dados relacionais em ambiente Java e a persistência transparente de classes POJO (plain old Java objects), que devem ter um construtor com apenas um argumento, para referenciar o servidor SQL no qual os dados manipulados pela classe serão persistidos.

Q43 - [CESPE DETRAN-ES 2010] O Hibernate, framework utilizado no desenvolvimento de consultas e atualização de dados em um banco relacional, foi criado para facilitar a integração entre programas em Java, funcionando também em ambientes .Net (NHibernate).

Q44 - [CESPE CORREIOS 2011] No Hibernate, o recurso Query Cache possibilita fazer o cache de queries que são executadas várias vezes.

Q42 - [CESPE ANATEL 2014] O Hibernate permite operações de banco de dados relacionais em ambiente Java e a persistência transparente de classes POJO (plain old Java objects), que devem ter um construtor com apenas um argumento, para referenciar o servidor SQL no qual os dados manipulados pela classe serão persistidos. ERRADO.

Q43 - [CESPE DETRAN-ES 2010] O Hibernate, framework utilizado no desenvolvimento de consultas e atualização de dados em um banco relacional, foi criado para facilitar a integração entre programas em Java, funcionando também em ambientes .Net (NHibernate). CERTO.

Q44 - [CESPE CORREIOS 2011] No Hibernate, o recurso Query Cache possibilita fazer o cache de queries que são executadas várias vezes. CERTO.

Q45 - [CONSULPLAN TSE 2012] O editor permite as funcionalidades descritas a seguir.

- A digitação de uma query HQL, com a correspondente visualização numa VIEW-1 da instrução SQL gerada automaticamente pelo Hibernate.
- A visualização do resultado da execução da query digitada, em Run HQL, numa VIEW-2. Neste caso, quando o resultado é selecionado, é possível a visualização dos

dados na view Properties.

As VIEW-1 e VIEW-2 são conhecidas, respectivamente, por

a)Hibernate Query HQL Preview e Hibernate Query Result.

b)Hibernate Dynamic SQL Preview e Hibernate Query Result.

c)Hibernate Query HQL Preview e Hibernate Data Generator.

d)Hibernate Dynamic SQL Preview e Hibernate Data Generator.

Q45 - [CONSULPLAN TSE 2012] O editor permite as funcionalidades descritas a seguir.

- A digitação de uma query HQL, com a correspondente visualização numa VIEW-1 da instrução SQL gerada automaticamente pelo Hibernate.
- A visualização do resultado da execução da query digitada, em Run HQL, numa VIEW-2. Neste caso, quando o resultado é selecionado, é possível a visualização dos dados na view Properties.

As VIEW-1 e VIEW-2 são conhecidas, respectivamente, por

a)Hibernate Query HQL Preview e Hibernate Query Result.

b)Hibernate Dynamic SQL Preview e Hibernate Query Result.

c)Hibernate Query HQL Preview e Hibernate Data Generator.

d)Hibernate Dynamic SQL Preview e Hibernate Data Generator.

GABARITO

Q31 - LETRA A.

Q32 - E (ERRADO).

Q33 - LETRA A.

Q34 - E (ERRADO).

Q35 - LETRA E.

Q36 – LETRA C.

Q37 – LETRA B.

Q38 - C (CERTO).

Q39 - C (CERTO).

Q40 - C (CERTO).

Q41 - E (ERRADO).

Q42 – E (ERRADO).

Q43 – E (ERRADO).

Q44 – C (CERTO).

Q45 – LETRA B.

Arquivo de configuração

- Define as unidades de persistências ou persistent units.
- Arquivo por padrão escrito em XML. **Pode ter o nome persistence.xml ou hibernate.cfg.xml**.
- Todas as propriedades da configuração ficam dentro da tag <persistence-unit>
  - Possui o atributo name que especifica o nome desta unidade de persistência.
- A tag provider diz qual implementação será usada como provedor de persistência.
- Properties: Atributos chave-valor.
  - **Hibernate.dialect**: Define o dialeto a ser usado nas construção de comandos SQL.
  - **Hibernate.show_sql**: Recebe um valor boolean. Informa se a query será ou não apresentada no console da IDE.
  - **Hibernate.hbm2ddl.auto**:
    - **Create:** Exclui tudo e cria novamente o schema.
    - **Create-drop:** Bem semelhante ao create, remove o schema no final da sessão.
    - **Update:** Faz as alterações no banco de dados sem precisar remover o schema.
    - **Validate:** Apenas valida o schema, se tiver algo errado lança uma exception.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1"
    xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
        http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
    <persistence-unit name="AlgaWorksPU">
        <provider>org.hibernate.ejb.HibernatePersistence</provider>

        <properties>
            <property name="javax.persistence.jdbc.url"
                value="jdbc:mysql://localhost/ebook-jpa" />
            <property name="javax.persistence.jdbc.user"
                value="usuario" />
            <property name="javax.persistence.jdbc.password"
                value="senha" />
            <property name="javax.persistence.jdbc.driver"
                value="com.mysql.jdbc.Driver" />

            <property name="hibernate.dialect"
                value="org.hibernate.dialect.MySQL5Dialect" />
            <property name="hibernate.show_sql" value="true" />
            <property name="hibernate.format_sql" value="true" />
            <property name="hibernate.hbm2ddl.auto" value="update" />
        </properties>
    </persistence-unit>

</persistence>
```

Utilização persistence-unit

- No método createEntityManagerFactory passa como parâmetro o nome da persistence-unit definida no arquivo persistence.xml.

Exemplo:

```
1 – import javax.persistence.Persistence;
2 -  public Class CriaTabelas{
3 -         public static void main(String[] args){
4 -                 Persistence. createEntityManagerFactory ("AlgaworksPU");
5 -         }
6 - }
```

Arquivo de mapeamento

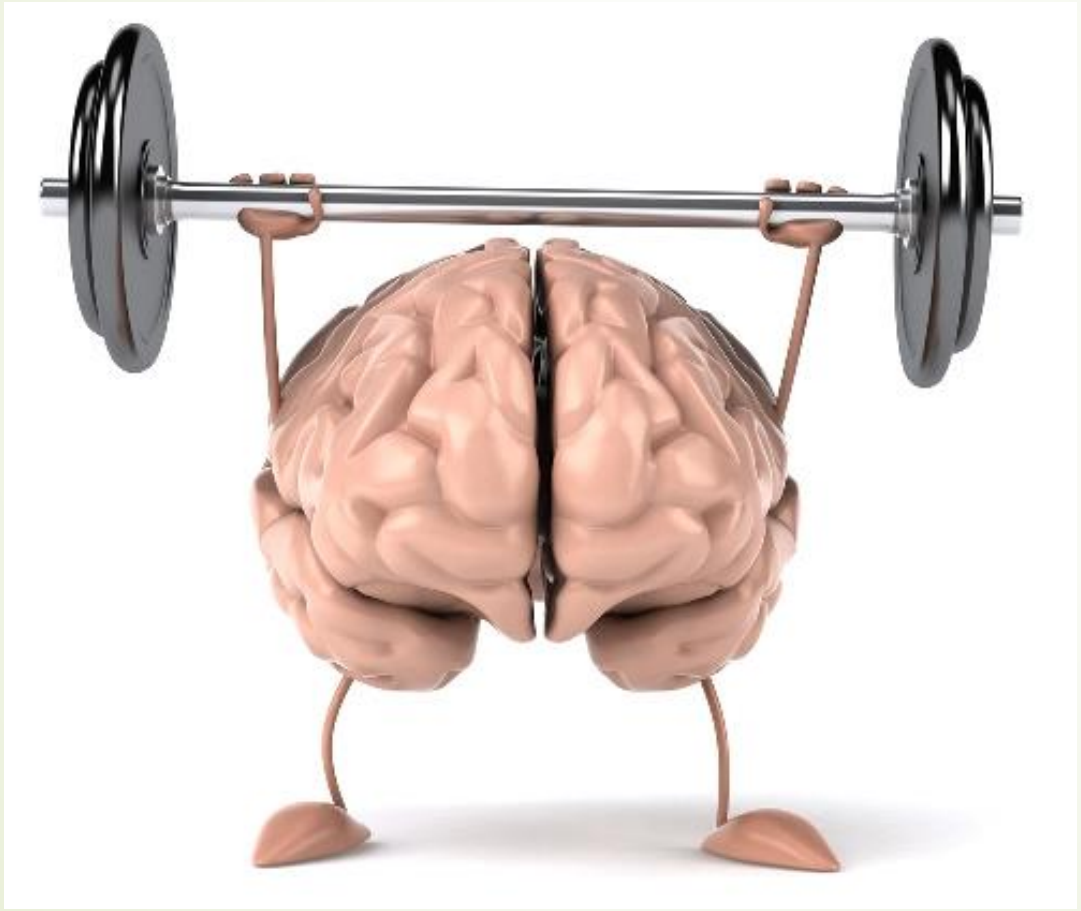
- Mapeia tabelas e colunas.
- Semelhante a utilização das anotações no JPA.
- Criado para cada classe persistente e que tem por nomenclatura o **nome da classe + 'hbm.xml'**.

Exemplo:

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
        "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
        "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd" [
    <!ENTITY types SYSTEM "classpath://your/domain/types.xml">
]>

<hibernate-mapping package="your.domain">
    <class name="MyEntity">
        <id name="id" type="my-custom-id-type">
            ...
        </id>
    <class>
    &types;
</hibernate-mapping>
```

Quinta Bateria de Questões



Q46 - [FCC TJ PE 2012] Quando se utiliza JPA, um EntityManager mapeia um conjunto de classes a um banco de dados particular. Este conjunto de classes, definido

em um arquivo chamado persistence.xml, é denominado

a)persistence context.

b)persistence unit.

c)entity manager factory.

d)entity transaction.

e)persistence provider.

Q47 - [CESPE TJDFT 2015] Para o usuário do JPA (Java Persistence API), o arquivo hbm.xml é responsável pela configuração do funcionamento do sistema e especifica a conexão com o banco de dados, seu dialeto e a linguagem de consulta.

Q46 - [FCC TJ PE 2012] Quando se utiliza JPA, um EntityManager mapeia um conjunto de classes a um banco de dados particular. Este conjunto de classes, definido

em um arquivo chamado persistence.xml, é denominado

a)persistence context.

b)persistence unit.

c)entity manager factory.

d)entity transaction.

e)persistence provider.

Q47 - [CESPE TJDFT 2015] Para o usuário do JPA (Java Persistence API), o arquivo hbm.xml é responsável pela configuração do funcionamento do sistema e especifica a conexão com o banco de dados, seu dialeto e a linguagem de consulta. ERRADO.

Q48 - [FCC TRT 19 2015]

Em uma aplicação que utiliza JPA e Hibernate, no arquivo persistence.xml

a)são definidas as configurações do servidor de aplicação, como número de conexões simultâneas permitidas e dados de log.

b)é definido o mapeamento objeto-relacional entre as tabelas do banco de dados e as classes de entidade da aplicação.

c)são definidas as queries nomeadas, que são queries pré-definidas que podem ser chamadas pelo nome a partir de classes de acesso a dados da aplicação.

d)há o mapeamento de componentes da camada de apresentação para os respectivos componentes da camada de acesso a dados da aplicação.

e)são definidas as propriedades de conexão com o banco de dados, como o driver JDBC, a URL de conexão, o nome do usuário e a senha.

Q48 - [FCC TRT 19 2015]

Em uma aplicação que utiliza JPA e Hibernate, no arquivo persistence.xml

a)são definidas as configurações do servidor de aplicação, como número de conexões simultâneas permitidas e dados de log.

b)é definido o mapeamento objeto-relacional entre as tabelas do banco de dados e as classes de entidade da aplicação.

c)são definidas as queries nomeadas, que são queries pré-definidas que podem ser chamadas pelo nome a partir de classes de acesso a dados da aplicação.

d)há o mapeamento de componentes da camada de apresentação para os respectivos componentes da camada de acesso a dados da aplicação.

e)são definidas as propriedades de conexão com o banco de dados, como o driver JDBC, a URL de conexão, o nome do usuário e a senha.

Q49 - [FCC TRE PB 2015] Em uma aplicação que utiliza JPA há a seguinte instrução: *EntityManagerFactory emf = Persistence.createEntityManagerFactory("teste");*

A palavra "teste" refere-se

a)à unidade de persistência definida na tag persistence-property do arquivo persistence.xml.

b)ao nome do arquivo de banco de dados que se deseja conectar.

c)a um objeto de uma classe Data Access Object, que estabelece a conexão com o banco de dados.

d)à unidade de persistência definida na tag persistence-unit do arquivo persistence.xml.

e)ao objeto da interface EntityManager que contém uma conexão válida com um banco de dados relacional.

Q50 - [CESPE TJ SE 2014] Um arquivo de mapeamento (mapping file) informa ao Hibernate que tabela do banco de dados deverá ser acessada e quais as colunas da tabela deverão ser usadas por ele.

Q49 - [FCC TRE PB 2015] Em uma aplicação que utiliza JPA há a seguinte instrução: *EntityManagerFactory emf = Persistence.createEntityManagerFactory("teste");*

A palavra "teste" refere-se

a)à unidade de persistência definida na tag persistence-property do arquivo persistence.xml.

b)ao nome do arquivo de banco de dados que se deseja conectar.

c)a um objeto de uma classe Data Access Object, que estabelece a conexão com o banco de dados.

d)à unidade de persistência definida na tag persistence-unit do arquivo persistence.xml.

e)ao objeto da interface EntityManager que contém uma conexão válida com um banco de dados relacional.

Q50 - [CESPE TJ SE 2014] Um arquivo de mapeamento (mapping file) informa ao Hibernate que tabela do banco de dados deverá ser acessada e quais as colunas da tabela deverão ser usadas por ele. CERTO.

Q51 - [FCC TRE CE 2012] Com relação ao framework Hibernate é correto afirmar:

a)Permite fazer a persistência automatizada dos objetos em uma aplicação Java para as tabelas de um banco de dados relacional, utilizando metadados (descrição dos dados) que descrevem o mapeamento entre os objetos e o banco de dados.

b)É uma boa opção apenas para sistemas que fazem muito uso de stored procedures, triggers ou que implementam a maior parte da lógica da aplicação no banco de dados vai se beneficiar mais com o uso do Hibernate.

c)Permite enviar unidirecionalmente uma representação de dados de um banco de dados relacional para um modelo de objeto utilizando um esquema baseado exclusivamente em Hibernate Query Language (HQL).

d)A Java Persistence API (JPA) implementa o Hibernate, que é parte do Enterprise JavaBeans 4.0.

e)Em uma aplicação criada com Hibernate, para cada classe de persistência é necessário criar um arquivo de mapeamento XML que deve ser salvo obrigatoriamente com o nome da classe seguido pelo sufixo .map.xml.

Q51 - [FCC TRE CE 2012] Com relação ao framework Hibernate é correto afirmar:

a)Permite fazer a persistência automatizada dos objetos em uma aplicação Java para as tabelas de um banco de dados relacional, utilizando metadados (descrição dos dados) que descrevem o mapeamento entre os objetos e o banco de dados.

b)É uma boa opção apenas para sistemas que fazem muito uso de stored procedures, triggers ou que implementam a maior parte da lógica da aplicação no banco de dados vai se beneficiar mais com o uso do Hibernate.

c)Permite enviar unidirecionalmente uma representação de dados de um banco de dados relacional para um modelo de objeto utilizando um esquema baseado exclusivamente em Hibernate Query Language (HQL).

d)A Java Persistence API (JPA) implementa o Hibernate, que é parte do Enterprise JavaBeans 4.0.

e)Em uma aplicação criada com Hibernate, para cada classe de persistência é necessário criar um arquivo de mapeamento XML que deve ser salvo obrigatoriamente com o nome da classe seguido pelo sufixo .map.xml.

Q52 - [FCC TRT 23 2016] Em uma aplicação que utiliza Hibernate como implementação da JPA, para definir suporte ao conjunto de instruções SQL específico de um determinado Sistema Gerenciador de Banco de Dados - SGBD, é necessário definir o dialeto SQL para esse SGBD. Isso normalmente é feito

a)pela tag dialect no arquivo persistence.xml.

b)pela tag property no arquivo web.xml.

c)no método de conexão da classe de acesso a dados.

d)pela tag property no arquivo persistence.xml.

e)como parte da instrução SQL que se deseja executar.

Q53 - [CESPE TJ SE 2014] Um dialeto encapsula todas as diferenças da forma com que o Hibernate se comunica com um banco de dados em particular para executar alguma tarefa, como recuperar um valor de uma sequência ou estruturar uma consulta SQL. A propriedade hibernate.dialect.property com a subclasse org.hibernate.dialect.Dialect define o dialeto para o banco de dados que se deseja utilizar.

Q52 - [FCC TRT 23 2016] Em uma aplicação que utiliza Hibernate como implementação da JPA, para definir suporte ao conjunto de instruções SQL específico de um determinado Sistema Gerenciador de Banco de Dados - SGBD, é necessário definir o dialeto SQL para esse SGBD. Isso normalmente é feito

a)pela tag dialect no arquivo persistence.xml.

b)pela tag property no arquivo web.xml.

c)no método de conexão da classe de acesso a dados.

d)pela tag property no arquivo persistence.xml.

e)como parte da instrução SQL que se deseja executar.

Q53 - [CESPE TJ SE 2014] Um dialeto encapsula todas as diferenças da forma com que o Hibernate se comunica com um banco de dados em particular para executar alguma tarefa, como recuperar um valor de uma sequência ou estruturar uma consulta SQL. A propriedade hibernate.dialect.property com a subclasse org.hibernate.dialect.Dialect define o dialeto para o banco de dados que se deseja utilizar. CERTO.

Q54 - [CESPE TCE RO 2013] Para ajustar o comportamento do Hibernate, pode ser utilizada a propriedade denominada hibernate.xml2ddl.auto, que executa a linguagem de manipulação de dados e pode assumir os valores update-valid, create-update ou create-drop.

Q55 - [CESPE TRT 17] A opção hbm2ddl.auto no arquivo hibernate.cfg.xml habilita a geração automática de esquemas da base de dados diretamente na base de dados. Essa opção pode ser naturalmente desligada apenas por meio da remoção da opção de configuração.

Q54 - [CESPE TCE RO 2013] Para ajustar o comportamento do Hibernate, pode ser utilizada a propriedade denominada hibernate.xml2ddl.auto, que executa a linguagem de manipulação de dados e pode assumir os valores update-valid, create-update ou create-drop. ERRADO.

Q55 - [CESPE TRT 17] A opção hbm2ddl.auto no arquivo hibernate.cfg.xml habilita a geração automática de esquemas da base de dados diretamente na base de dados. Essa opção pode ser naturalmente desligada apenas por meio da remoção da opção de configuração. CERTO.

Q56 - [FCC TRT 3 2014] Considere a seguinte classe desenvolvida em uma aplicação Java que utiliza JPA/Hibernate:

```
import javax.persistence.*;
public class NewClassDao {
    public void conectar(){
        EntityManagerFactory a = Persistence.createEntityManagerFactory("conectar");
        EntityManager b = a.createEntityManager();
        EntityTransaction c = b.getTransaction();
        c.begin();
    }
}
```

É correto afirmar que os diversos métodos para executar operações de inserção, consulta, alteração e exclusão de registros no Banco de Dados (persist, find, merge, remove, createQuery etc) podem ser acessados por meio do objeto I . A String "conectar" refere-se ao nome da II.

As lacunas I e II são preenchidas correta e respectivamente com

a) c - unidade de persistência

b) c - interface remota

c) a - interface local

d) b - interface local

e) b - unidade de persistência

Q56 - [FCC TRT 3 2014] Considere a seguinte classe desenvolvida em uma aplicação Java que utiliza JPA/Hibernate:

```
import javax.persistence.*;
public class NewClassDao {
    public void conectar(){
        EntityManagerFactory a = Persistence.createEntityManagerFactory("conectar");
        EntityManager b = a.createEntityManager();
        EntityTransaction c = b.getTransaction();
        c.begin();
    }
}
```

É correto afirmar que os diversos métodos para executar operações de inserção, consulta, alteração e exclusão de registros no Banco de Dados (persist, find, merge, remove, createQuery etc) podem ser acessados por meio do objeto I . A String "conectar" refere-se ao nome da II.

As lacunas I e II são preenchidas correta e respectivamente com

a) c - unidade de persistência

b) c - interface remota

c) a - interface local

d) b - interface local

e) b - unidade de persistência

GABARITO

Q46 - LETRA B.
Q47 - E (ERRADO).
Q48 - LETRA E.
Q49 - LETRA D.
Q50 - C(CERTO).
Q51 - LETRA A.
Q52 - LETRA D.
Q53 - C(CERTO).
Q54 - E(ERRADO).
Q55 - C(CERTO).
Q56 - LETRA E.


SQL


HQL

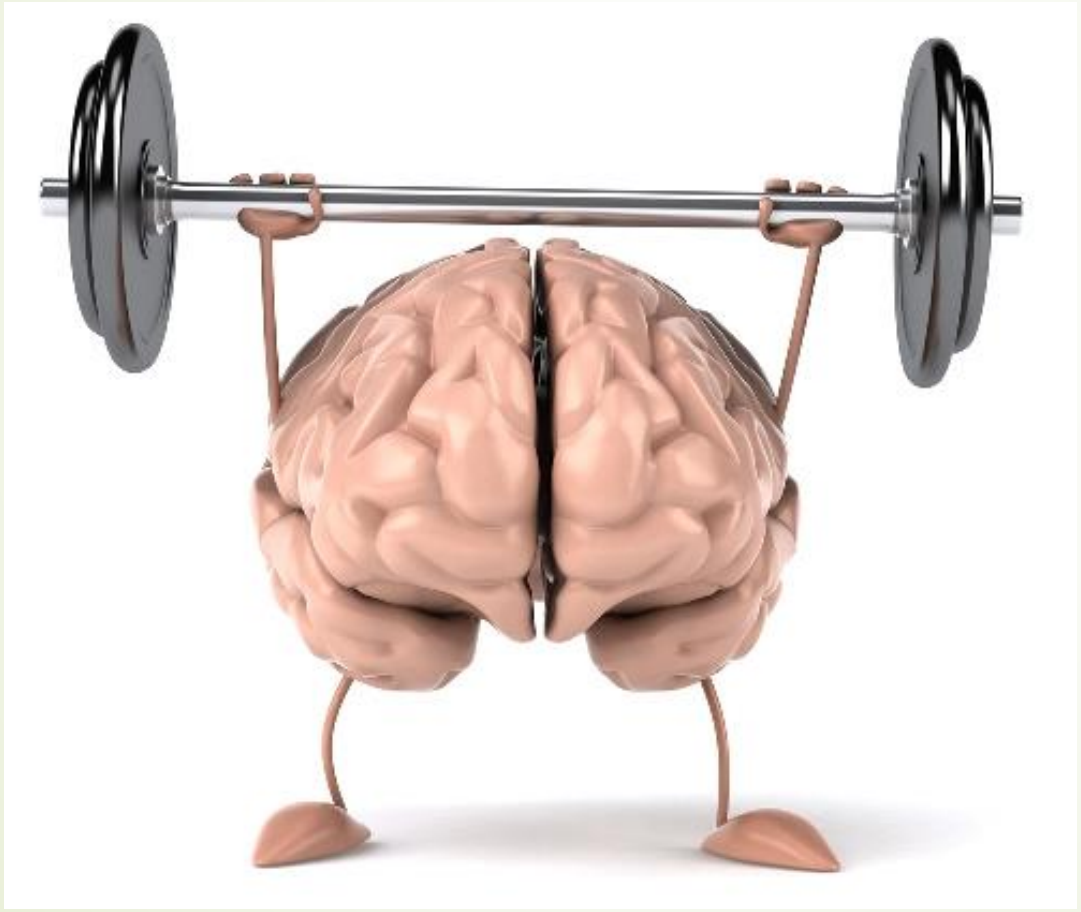
Api Criteria

- Erros podem ser verificados em tempo de compilação. Algo que não é possível nem no SQL nem HQL.
- Constrói as consultas de forma programática.
- Une as vantagens do HQL com praticidade de se escrever consultas em Java através da interface org.hibernate.Criteria.
- As consultas ficam menos imunes a alguns ataques como sql injection.
- Queries dinâmicas ao invés de strings longas.
- Como desvantagem pode ter uma complexidade em relação a HQL e SQL para a construção de consultas.
- Permite também criar restrições dentro de uma consulta ao retornar objetos.
  - eq(): Retorna objeto em conformidade com o valor passado.
  - ne(): Retornam os objetos que são diferentes do valor passado.
  - Like
  - ilike: Mesma função do like mas leva em consideração o case-sensitive.
  - sqlRestriction: Permite especificar diretamente uma query SQL na API criteria.

HQL

- Consulta acontece nas entidades que foram mapeadas e não na própria tabela.
- Independem de SGBD.
- Consulta pode começar a partir do ‘from’.
- Totalmente orientado à objetos, utiliza herança, polimorfismo e associações.
- Aceita parâmetros nomeados.
- **Regras case-sensitive**: Teste <> teste
  - Nome de classes Java.
  - Propriedades.

Sexta Bateria de Questões



Q57 - [FCC TRT 19 2011] Linguagem de queries, fornecida pelo Hibernate, que é similar em aparência ao SQL e que, no entanto, é orientada a objeto e compreende noções como herança, polimorfismo e associação. Trata-se de

a)HiBD-QL.

b)OOQL.

c)ORM-QL.

d)HQL.

e)JEEQL.

Q58 - [CESPE TCU 2010] A tecnologia Hibernate 3.5 é apropriada para o sistema a ser desenvolvido: entre as características que a credenciam, está o fato de ela possibilitar a recuperação de objetos por meio da formulação de queries em linguagens HQL (hibernate query language) e SQL (structured query language), bem como pelo uso de APIs (application programming interfaces) de busca por critério, entre outras.

Q57 - [FCC TRT 19 2011] Linguagem de queries, fornecida pelo Hibernate, que é similar em aparência ao SQL e que, no entanto, é orientada a objeto e compreende noções como herança, polimorfismo e associação. Trata-se de

a)HiBD-QL.

b)OOQL.

c)ORM-QL.

d)HQL.

e)JEEQL.

Q58 - [CESPE TCU 2010] A tecnologia Hibernate 3.5 é apropriada para o sistema a ser desenvolvido: entre as características que a credenciam, está o fato de ela possibilitar a recuperação de objetos por meio da formulação de queries em linguagens HQL (hibernate query language) e SQL (structured query language), bem como pelo uso de APIs (application programming interfaces) de busca por critério, entre outras. CERTO.

Q59 - [CESPE INPI 2013] A interface Criteria do Hibernate é utilizada para realizar consultas ao banco de dados.

Q60 - [CESPE TRE BA 2013] No Hibernate, apenas a linguagem de consulta HQL (hibernate query language) pode ser utilizada. A HQL executa os pedidos SQL sobre as classes de persistência do Java em vez de tabelas no banco de dados, o que diminui a distância entre o desenvolvimento das regras de negócio e o banco de dados.

Q61 - [CESPE MEC 2015] Uma das desvantagens da utilização do Hibernate é o aumento do tempo de desenvolvimento, já que todas as consultas SQL deverão ser transcritas para a linguagem HQL para poderem ser mapeadas em classes Java.

Q62 - [CESPE MEC 2015] Ao se utilizar o Hibernate, não é preciso que se reescrevam consultas HQL durante a migração entre bancos de dados diferentes.

Q59 - [CESPE INPI 2013] A interface Criteria do Hibernate é utilizada para realizar consultas ao banco de dados. CERTO.

Q60 - [CESPE TRE BA 2013] No Hibernate, apenas a linguagem de consulta HQL (hibernate query language) pode ser utilizada. A HQL executa os pedidos SQL sobre as classes de persistência do Java em vez de tabelas no banco de dados, o que diminui a distância entre o desenvolvimento das regras de negócio e o banco de dados. ERRADO.

Q61 - [CESPE MEC 2015] Uma das desvantagens da utilização do Hibernate é o aumento do tempo de desenvolvimento, já que todas as consultas SQL deverão ser transcritas para a linguagem HQL para poderem ser mapeadas em classes Java. ERRADO.

Q62 - [CESPE MEC 2015] Ao se utilizar o Hibernate, não é preciso que se reescrevam consultas HQL durante a migração entre bancos de dados diferentes. CERTO.

Q63 - [FUNCAB PRODAM AM 2012] Em uma query escrita em HQL(Hibernate Query Language), são case-sensitive:

a)somente nomes de classes Java e propriedades e palavras reservadas do HQL.

b)somente palavras reservadas do HQL

c)todos elementos da query

d)Nenhum elemento da query.

e)somente nomes de classes Java e propriedades.

Q64 - [CESPE TC DF 2014] O comando abaixo permite recuperar informação HQL (Hibernate Query Language), linguagem de consulta do Hibernate que permite a consulta e recuperação de informação.

retrieve (a=count(y.i by y.d where y.str = "ii" or y.str = "foo"),b=max(count(y.i by y.d)))*

Q63 - [FUNCAB PRODAM AM 2012] Em uma query escrita em HQL(Hibernate Query Language), são case-sensitive:

a)somente nomes de classes Java e propriedades e palavras reservadas do HQL.

b)somente palavras reservadas do HQL

c)todos elementos da query

d)Nenhum elemento da query.

e)somente nomes de classes Java e propriedades.

Q64 - [CESPE TC DF 2014] O comando abaixo permite recuperar informação HQL (Hibernate Query Language), linguagem de consulta do Hibernate que permite a consulta e recuperação de informação.

retrieve (a=count(y.i by y.d where y.str = "ii" or y.str = "foo"),b=max(count(y.i by y.d)))*. ERRADO.

GABARITO

Q57 - LETRA D.

Q58 - C(CERTO).

Q59 - C(CERTO).

Q60 - E(ERRADO).

Q61 - E(ERRADO).

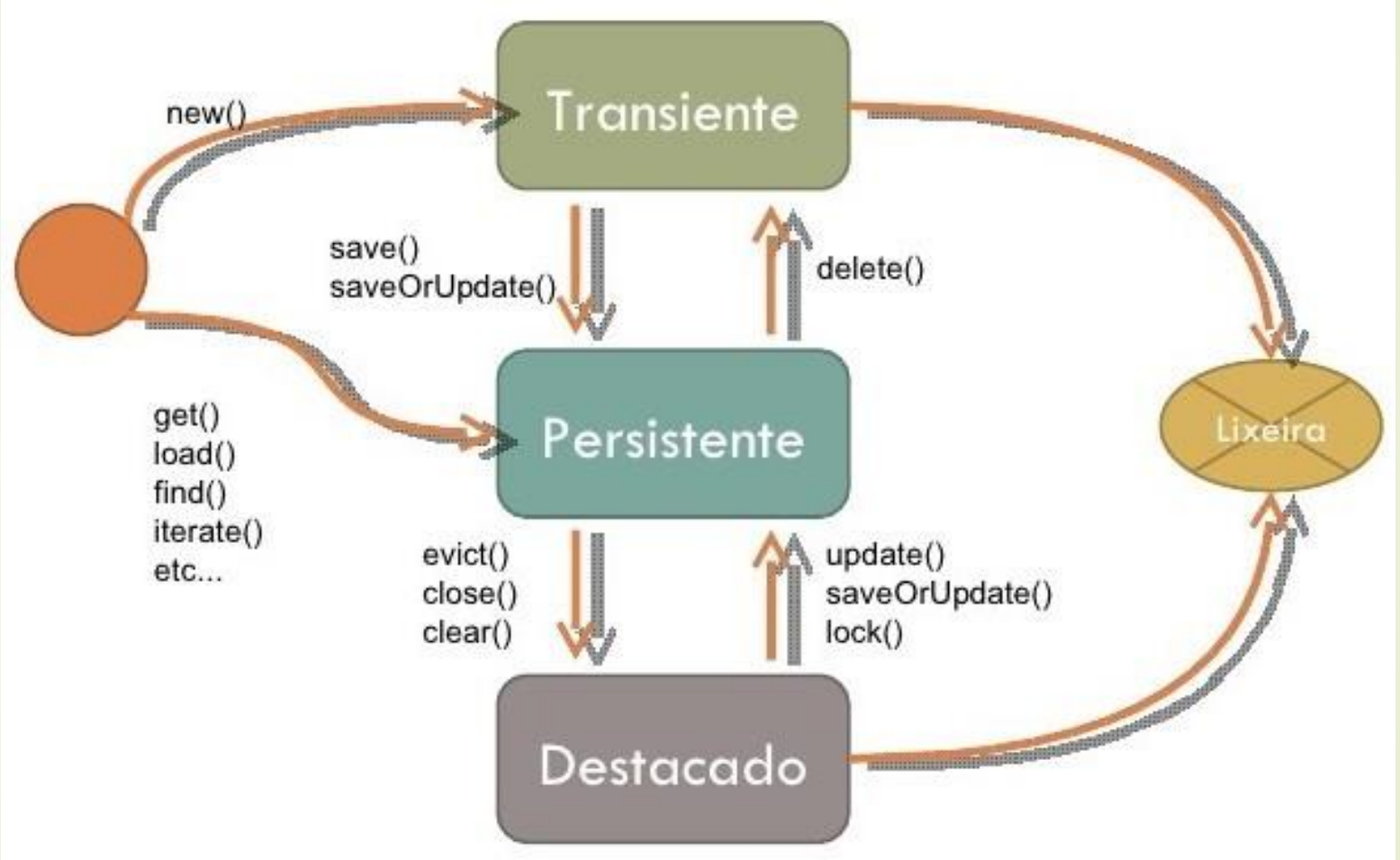
Q62 - C(CERTO).

Q63 - LETRA E.

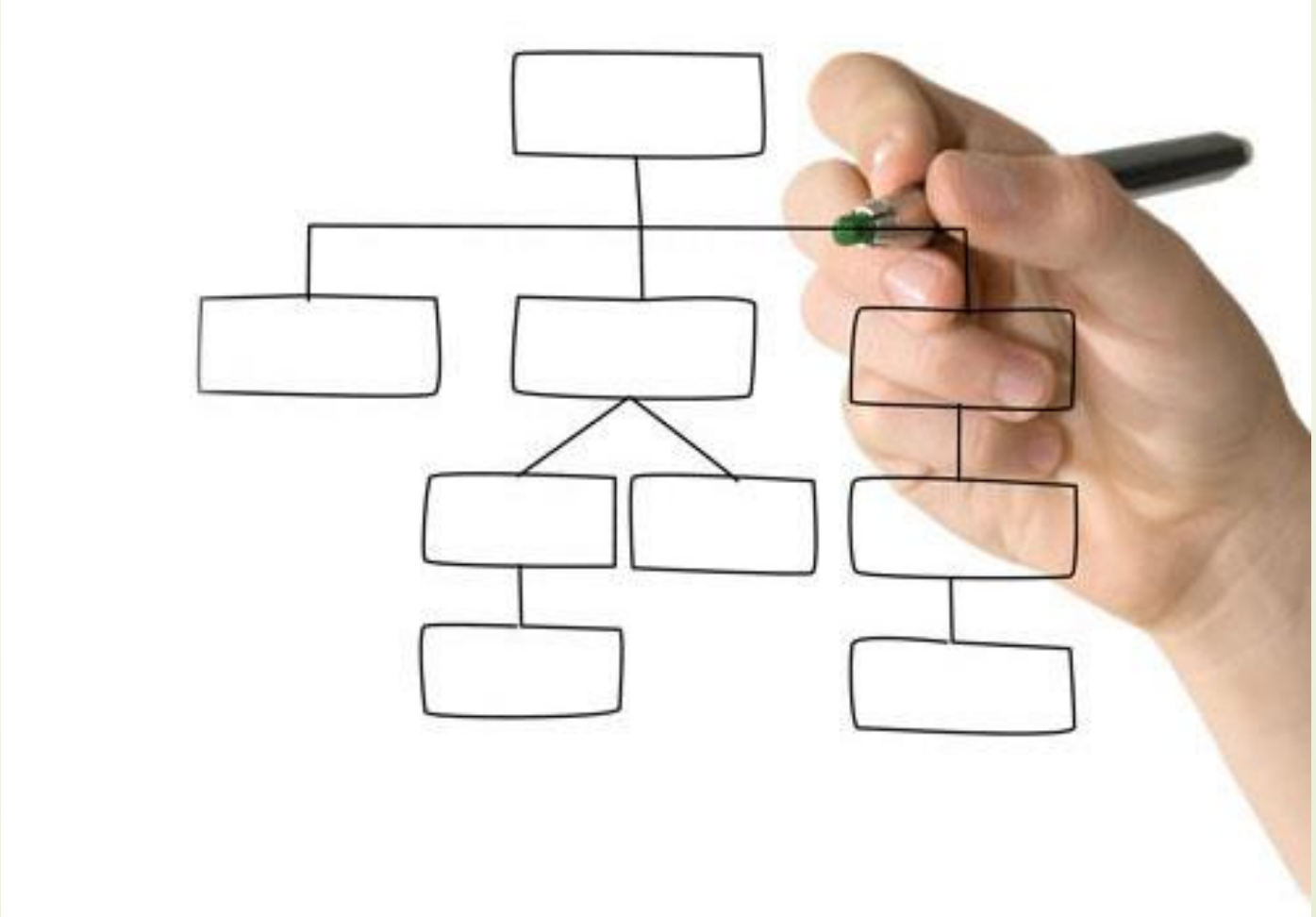
Q64 - E(ERRADO).

Ciclo de Vida

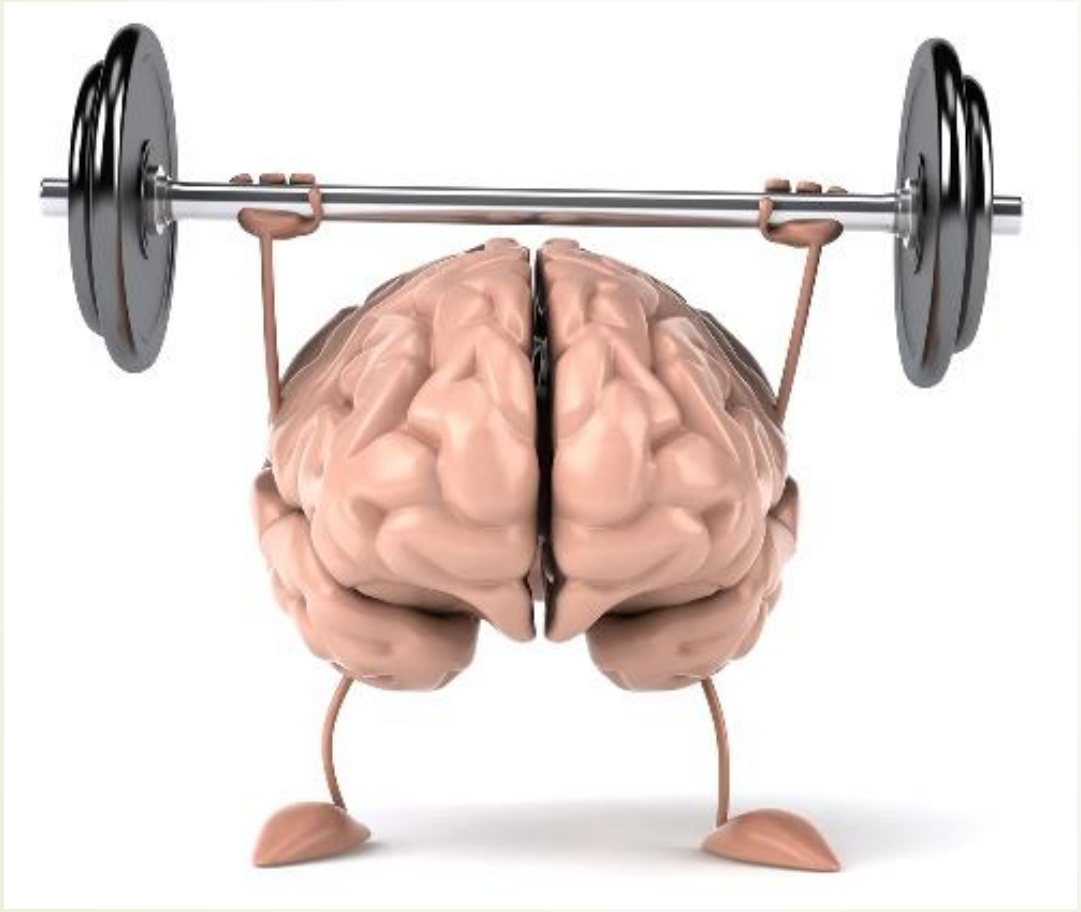
- **Transient:** Objetos que ainda não estão associados ao banco de dados. Não estão nem sequer instanciados.
  - São instanciados com o operador new.
- **Persistent**: Estão sempre associados a um contexto de persistência.
  - Qualquer alteração nos objetos são refletidos na base de dados. Sincronismo direto.
- **Detashed**: Continua existindo a instância do objeto, porém o gerenciador de entidades está fechado.
  - Não há mais sincronia com o banco de dados


Transiente
new()
save()
saveOrUpdate()
delete()
Persistente
get()
load()
find()
iterate()
etc...
Lixeira
evict()
close()
clear()
update()
saveOrUpdate()
lock()
Destacado

D – P - T



Sétima Bateria de Questões



Q65 - [CESPE CEHAP-PB 2009] No framework Hibernate, é comum que uma instância de uma classe persistente tenha três estados específicos. Assinale a opção

que contém esses três estados.

a)plugged, disconnected, timewait

b)connected, disconnected, detached

c)transient, persistent, detached

d)transient, connected, timewait

Q66 - [FCC MPU 2007] Objetos que têm uma representação no banco de dados, mas não fazem mais parte de uma sessão do Hibernate, o que significa que o seu estado pode não estar mais sincronizado com o banco de dados, são do tipo

a)transient.

b)detached.

c)attached.

d)persistent.

e)consistent

Q65 - [CESPE CEHAP-PB 2009] No framework Hibernate, é comum que uma instância de uma classe persistente tenha três estados específicos. Assinale a opção

que contém esses três estados.

a)plugged, disconnected, timewait

b)connected, disconnected, detached

c)transient, persistent, detached

d)transient, connected, timewait

Q66 - [FCC MPU 2007] Objetos que têm uma representação no banco de dados, mas não fazem mais parte de uma sessão do Hibernate, o que significa que o seu estado pode não estar mais sincronizado com o banco de dados, são do tipo

a)transient.

b)detached.

c)attached.

d)persistent.

e)consistent

Q67 -[FCC TRT 8 2010] Os três estados de objeto definidos pelo framework Hibernate são:

a)Temporário (Temporary), Permanente (Permanent) e Resiliente (Resilient).

b)Transiente (Transient), Persistente (Persistent) e Resiliente (Resilient).

c)Temporário (Temporary), Persistente (Persistent) e Destacado (Detached).

d)Transiente (Transient), Persistente (Persistent) e Destacado (Detached).

e)Transiente (Transient), Permanente (Permanent) e Resiliente (Resilient).

Q68 - [CESPE STJ 2015] O Hibernate define um objeto transient com uma instância de um objeto que tenha persistido e que esteja em transição para consulta e utilização pela aplicação.

Q67 -[FCC TRT 8 2010] Os três estados de objeto definidos pelo framework Hibernate são:

a)Temporário (Temporary), Permanente (Permanent) e Resiliente (Resilient).

b)Transiente (Transient), Persistente (Persistent) e Resiliente (Resilient).

c)Temporário (Temporary), Persistente (Persistent) e Destacado (Detached).

d)Transiente (Transient), Persistente (Persistent) e Destacado (Detached).

e)Transiente (Transient), Permanente (Permanent) e Resiliente (Resilient).

Q68 - [CESPE STJ 2015] O Hibernate define um objeto transient com uma instância de um objeto que tenha persistido e que esteja em transição para consulta e utilização pela aplicação. ERRADO.

Q69 - [FCC TRT 4 2015] Uma aplicação que trabalha com Hibernate e EJB possui uma classe POJO - Plain Old Java Object utilizada no mapeamento objeto-relacional com uma tabela do banco de dados. Nessa classe, há um atributo calculado chamado valorTotalPedido que, para ser utilizado apenas em tempo de execução e descartado após finalizar o seu serviço temporário, deverá ser anotado com

a)@Embedded

b)@TemporaryAttribute

c)@GeneratedValue

d)@Transient

e)@Basic

Q69 - [FCC TRT 4 2015] Uma aplicação que trabalha com Hibernate e EJB possui uma classe POJO - Plain Old Java Object utilizada no mapeamento objeto-relacional com uma tabela do banco de dados. Nessa classe, há um atributo calculado chamado valorTotalPedido que, para ser utilizado apenas em tempo de execução e descartado após finalizar o seu serviço temporário, deverá ser anotado com

a)@Embedded

b)@TemporaryAttribute

c)@GeneratedValue

d)@Transient

e)@Basic

GABARITO

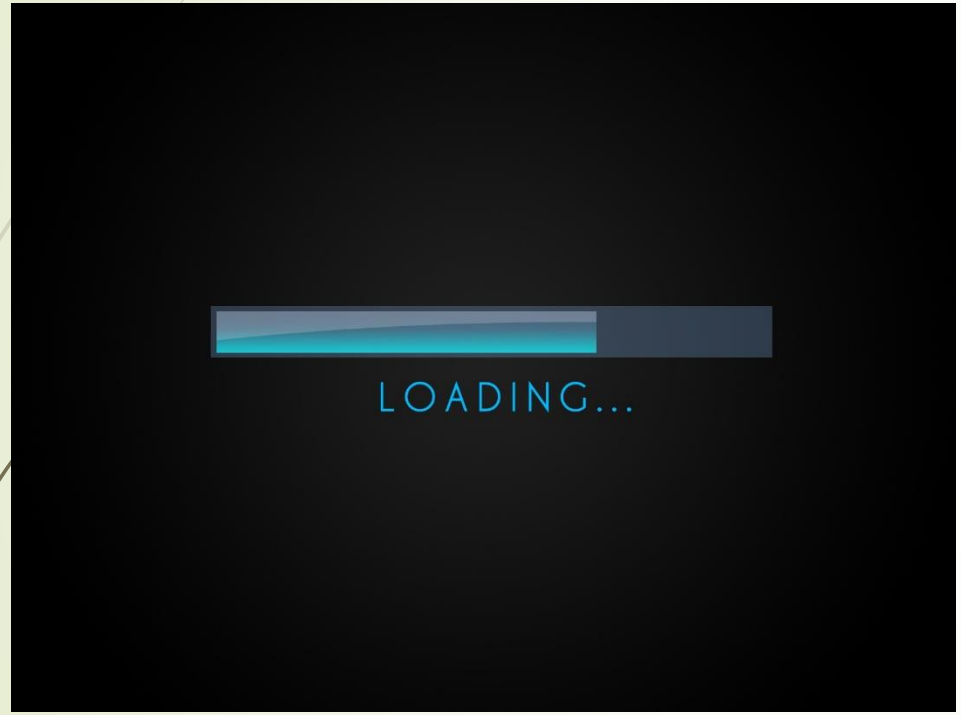
Q65 - LETRA C.

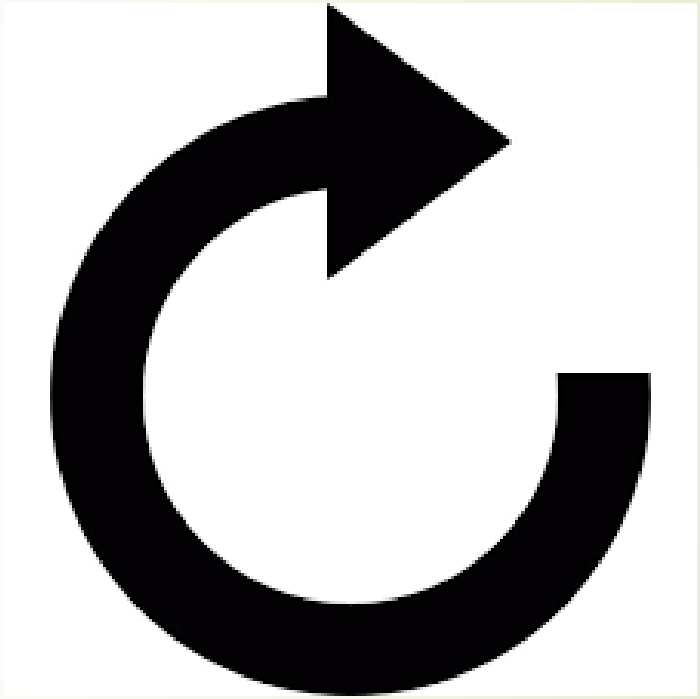
Q66 - LETRA D.

Q67 - LETRA D.

Q68 - E(ERRADO).

Q69 - LETRA D.


LOADING...



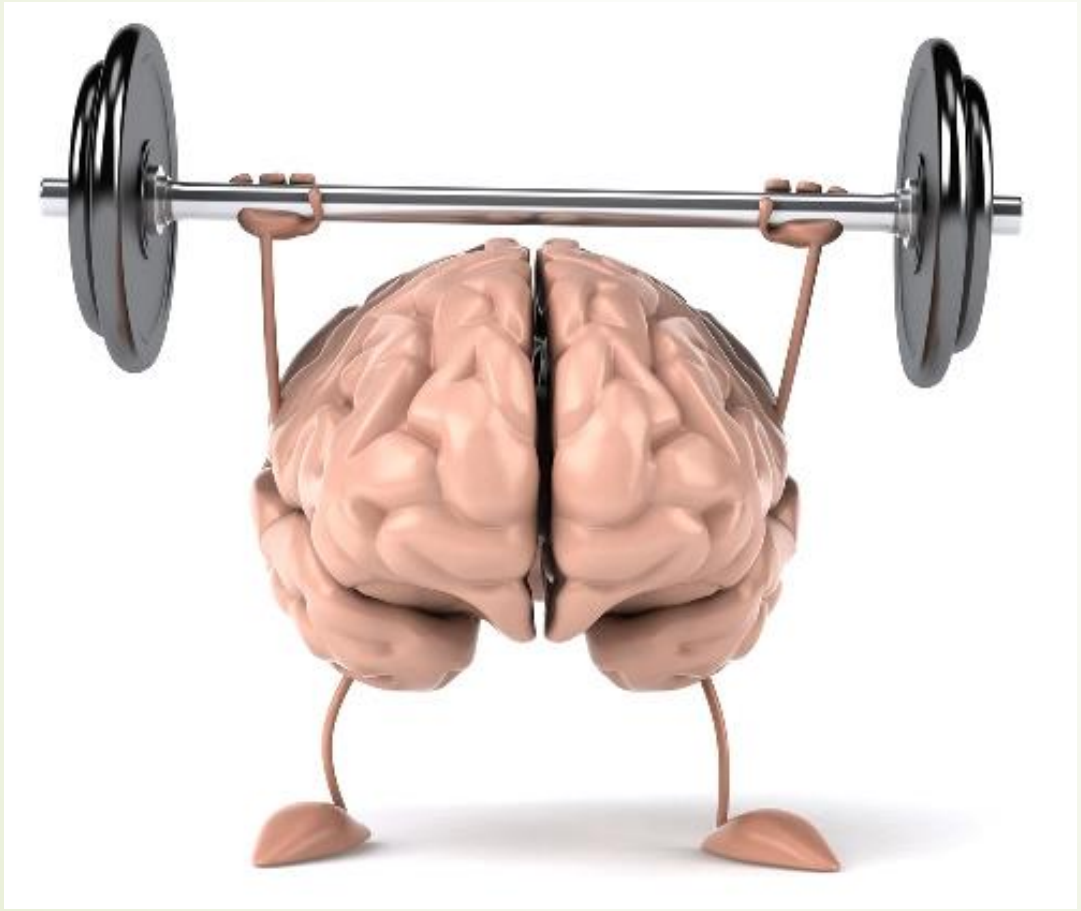
Eager Loading

- Relacionamento "x para um". Ex: @ManyToOne, @OneToOne.
- Esses relacionamentos utilizam o eager loading como padrão.
- Conhecido também por "busca ansiosa".
- Muitas das vezes o eager carrega informações sem levar em consideração se elas serão ou não utilizadas.
- Geralmente, uma consulta que utilizar busca ansiosa é formada por **uma única query contendo um join com a outra tabela relacionada.**

Lazy Loading

- Relacionamento "x para muitos". Ex: OneToMany, @ManyToMany.
- Esses relacionamentos utilizam o lazy loading como padrão.
- Conhecido também por "busca preguiçosa".
- Por outro lado o lazy load só vai carregar informações do lado N quando for realmente necessário.
- Geralmente numa consulta que utiliza busca preguiçosa é **formada por 2 querys, onde a query da entidade do lado N só é construída quando necessário.**
- Esse atributo lazy dentro da tag map pode receber como valores:
  - **True:** É o valor padrão, informa que determinada consulta será lazy loading.
  - **False:** Informa que determinada consulta não será lazy loading.
  - **Extra:** Hibernate não busca a coleção inteira na memória a menos que seja necessária. (Apropriado para grandes coleções).

Oitava Bateria de Questões



Q70 - [FCC TRE PB 2015] Quando temos uma relação um-para-muitos entre classes de entidade na aplicação que utiliza JPA, por padrão, ao buscarmos um objeto do lado um, não são trazidos os objetos relacionados do lado muitos da relação. Porém, se desejarmos trazer todos os objetos relacionados, devemos utilizar na classe de entidade do lado um da relação a anotação:

a)@OneToMany(fetch=FetchType.LAZY)

b)@ManyToOne(Cascade.JOIN)

c)@OneToMany(fetch=Fetch.JOIN)

d)@ManyToOne(Cascade=true)

e)@OneToMany(fetch=FetchType.EAGER)

Q71 - [CESPE MPU 2013] O Hibernate sempre usa a estratégia de busca preguiçosa para coleções que precisará buscar no banco de dados inteiro da memória em cada transação, ou seja, são executas consultas para cada entidade ou coleção associada ao mesmo comando SELECT.

Q70 - [FCC TRE PB 2015] Quando temos uma relação um-para-muitos entre classes de entidade na aplicação que utiliza JPA, por padrão, ao buscarmos um objeto do lado um, não são trazidos os objetos relacionados do lado muitos da relação. Porém, se desejarmos trazer todos os objetos relacionados, devemos utilizar na classe de entidade do lado um da relação a anotação:

a)@OneToMany(fetch=FetchType.LAZY)

b)@ManyToOne(Cascade.JOIN)

c)@OneToMany(fetch=Fetch.JOIN)

d)@ManyToOne(Cascade=true)

e)@OneToMany(fetch=FetchType.EAGER)

Q71 - [CESPE MPU 2013 GAB C] O Hibernate sempre usa a estratégia de busca preguiçosa para coleções que precisará buscar no banco de dados inteiro da memória em cada transação, ou seja, são executas consultas para cada entidade ou coleção associada ao mesmo comando SELECT. ERRADO.

Q72 - [FEPESE JUCESC 2013] Assinale a alternativa correta com relação ao atributo lazy.

a)Contém uma biblioteca de métodos prontos que visam agilizar o processo de desenvolvimento com hibernate.

b)Define uma forma de recuperação de dados empregada pelo hibernate que otimiza a performance por default ou padrão.

c)Armazena dinamicamente erros de programação decorrentes de mau uso das melhores práticas de desenvolvimento em tempo de execução.

d)Decorre da otimização de código com menos linhas, automatizando o desenvolvimento e economizando tempo e código.

e)Otimiza a performance do hibernate ao tratar exceções e possibilitar a continuidade da execução pois armazena alternativas à exceção ocorrida.

Q73 - [CESPE MPU 2013] O Hibernate sempre usa a estratégia de busca preguiçosa para coleções que precisará buscar no banco de dados inteiro da memória em cada transação, ou seja, são executas consultas para cada entidade ou coleção associada ao mesmo comando SELECT.

Q72 - [FEPESE JUCESC 2013] Assinale a alternativa correta com relação ao atributo lazy.

a)Contém uma biblioteca de métodos prontos que visam agilizar o processo de desenvolvimento com hibernate.

b)Define uma forma de recuperação de dados empregada pelo hibernate que otimiza a performance por default ou padrão.

c)Armazena dinamicamente erros de programação decorrentes de mau uso das melhores práticas de desenvolvimento em tempo de execução.

d)Decorre da otimização de código com menos linhas, automatizando o desenvolvimento e economizando tempo e código.

e)Otimiza a performance do hibernate ao tratar exceções e possibilitar a continuidade da execução pois armazena alternativas à exceção ocorrida.

Q73 - [CESPE MPU 2013] O Hibernate sempre usa a estratégia de busca preguiçosa para coleções que precisará buscar no banco de dados inteiro da memória em cada transação, ou seja, são executas consultas para cada entidade ou coleção associada ao mesmo comando SELECT. ERRADO.

Q74 - [FEPESE JUCESC 2013] Assinale a alternativa que indica a palavra-chave empregada em conjunto com fetch que defne sua forma de funcionamento de forma a ignorar a estratégia definida de lazy do Hibernate no mapping document deste..

a)JOIN

b)SELECT

c)SUBSELECT

d)BATCH

e)ALL

Q75 - [FUNCAB PRODAM AM 2014] Durante um mapeamento utilizando Hibernate 4.1, além dos valores true e false, pode ser escolhido um terceiro valor diferente para o elemento lazy de uma tag map:

a)extra.

b)half.

c)join.

d)select.

e)asc.

Q74 - [FEPESE JUCESC 2013] Assinale a alternativa que indica a palavra-chave empregada em conjunto com fetch que defne sua forma de funcionamento de forma a ignorar a estratégia definida de lazy do Hibernate no mapping document deste..

a)JOIN

b)SELECT

c)SUBSELECT

d)BATCH

e)ALL

Q75 - [FUNCAB PRODAM AM 2014] Durante um mapeamento utilizando Hibernate 4.1, além dos valores true e false, pode ser escolhido um terceiro valor diferente para o elemento lazy de uma tag map:

a)extra.

b)half.

c)join.

d)select.

e)asc.

GABARITO

Q70 - LETRA E.

Q71 - E(ERRADO).

Q72 - LETRA B.

Q73 - E(ERRADO).

Q74 - LETRA A.

Q75 – LETRA A.

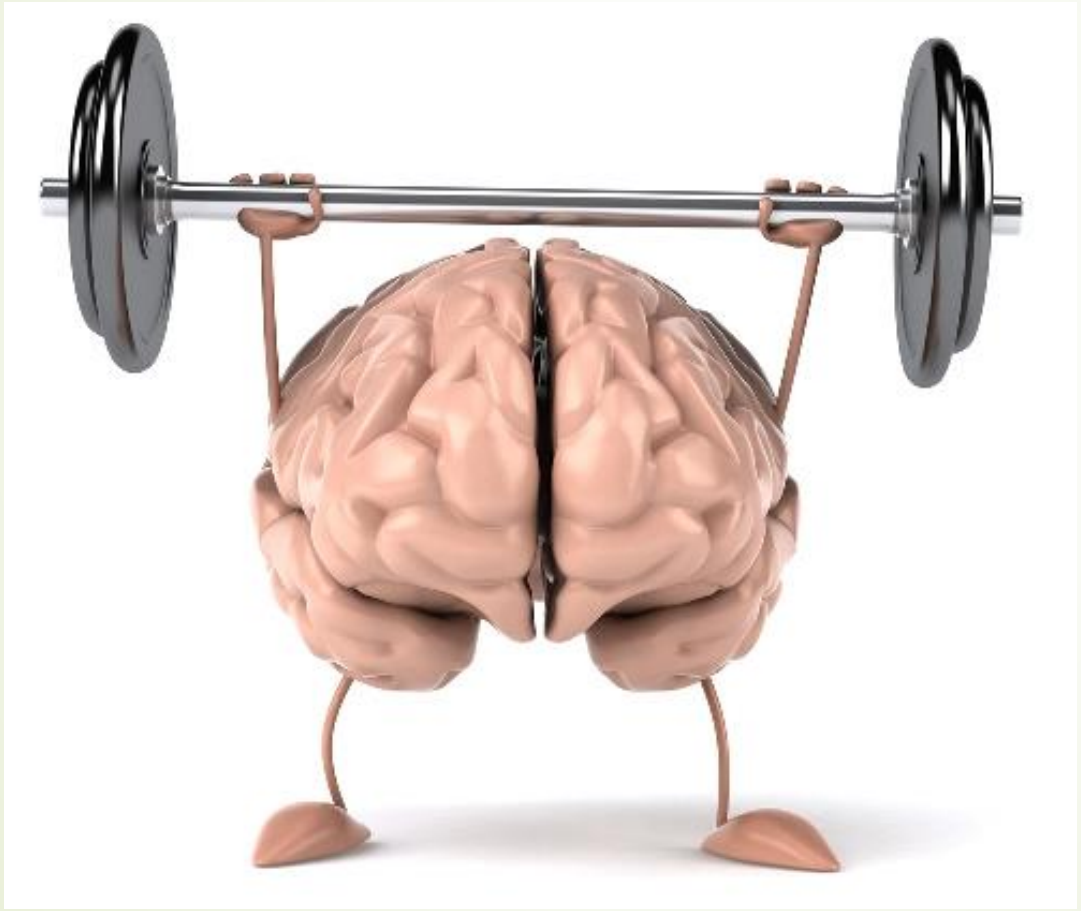
Conceitos Extras – Configuração e Mapeamento

- Mapeamento de herança pode ser feito de 3 formas:
  - Tabela única por hierarquia.
  - Tabela por subclasse.
  - Tabela por classe concreta.
- Além do arquivo de configuração outra forma de configurar o hibernate são:
  - Programaticamente.
  - Através do arquivo hibernate.properties.

Conceitos Extras – Session e Session Factory

- **Session**: objeto single-threaded permite conversação entre a aplicação e o armazenamento persistente.
- **Session Factory**: Cache imutável composto de mapeamentos compilados para um único banco de dados.
- Sessão é sempre criada através de uma Session Factory.
  - Geralmente se tem uma instância de um Session Factory por aplicação.
  - Sessions não são thread-safe.

Nona Bateria de Questões



Q76- [CESGRANRIO TJ RO 2008] Sobre o framework de mapeamento objeto/relacional Hibernate, é correto afirmar que

a)após modificar um objeto que já está vinculado a uma session (sessão) aberta, é necessário utilizar o método update() do objeto session para que as modificações sejam gravadas no banco de dados.

b)um objeto se encontra no estado detached quando tiver sido criado com o construtor new, estando ainda desvinculado de qualquer session (sessão) do Hibernate.

c)um objeto gerenciado pelo Hibernate se encontra no estado transient quando tiver sido criado vinculado a uma session (sessão) do Hibernate, mas ainda não tiver sido gravado no banco de dados.

d)um objeto gerenciado pelo Hibernate se encontra no estado persistent quando a session (sessão) a que o mesmo estava vinculado foi fechada e, em conseqüência, o objeto já foi gravado no banco de dados.

e)uma session do Hibernate não é thread-safe, o que significa que pode gerar resultados imprevisíveis e instabilidade se uma instância for utilizada, simultaneamente, por mais de uma thread da aplicação sem a devida sincronização do acesso à mesma.

Q76 - [CESGRANRIO TJ RO 2008] Sobre o framework de mapeamento objeto/relacional Hibernate, é correto afirmar que

a)após modificar um objeto que já está vinculado a uma session (sessão) aberta, é necessário utilizar o método update() do objeto session para que as modificações sejam gravadas no banco de dados.

b)um objeto se encontra no estado detached quando tiver sido criado com o construtor new, estando ainda desvinculado de qualquer session (sessão) do Hibernate.

c)um objeto gerenciado pelo Hibernate se encontra no estado transient quando tiver sido criado vinculado a uma session (sessão) do Hibernate, mas ainda não tiver sido gravado no banco de dados.

d)um objeto gerenciado pelo Hibernate se encontra no estado persistent quando a session (sessão) a que o mesmo estava vinculado foi fechada e, em conseqüência, o objeto já foi gravado no banco de dados.

e)uma session do Hibernate não é thread-safe, o que significa que pode gerar resultados imprevisíveis e instabilidade se uma instância for utilizada, simultaneamente, por mais de uma thread da aplicação sem a devida sincronização do acesso à mesma.

Q77 - [FCC PGE RJ 2009] Usando Hibernate, o mapeamento de uma relação de herança pode ser feito usando uma tabela por

a)classe concreta, subclasse ou hierarquia.

b)hierarquia, apenas.

c)subclasse, apenas.

d)classe concreta ou subclasse, apenas.

e)classe concreta, apenas.

Q77 - [FCC PGE RJ 2009] Usando Hibernate, o mapeamento de uma relação de herança pode ser feito usando uma tabela por

a)classe concreta, subclasse ou hierarquia.

b)hierarquia, apenas.

c)subclasse, apenas.

d)classe concreta ou subclasse, apenas.

e)classe concreta, apenas.

Q78 – [FCC TRE CE 2012] Considere:

I - Cache imutável composto de mapeamentos compilados para um único banco de dados.

II – Objeto *single-threaded*, de vida curta, que representa uma conversação entre o aplicativo e o armazenamento persistente.

III - Abstrai a aplicação dos Datasource ou DriverManager adjascentes.

Em relação à arquitetura do *Hibernate*, os itens I, II e III, associam-se, respectivamente, a

a) SessionFactory, Session e Connection Provider.

b) Transaction Factory, Session Factory e Connection Provider.

c) Session, Connection Provider e Extension Interfaces.

d) Connection Provider, Extension Interfaces e Session Factory.

e) Connection Provider, Session e Session Factory.

Q78 – [FCC TRE CE 2012] Considere:

I - Cache imutável composto de mapeamentos compilados para um único banco de dados.

II – Objeto *single-threaded*, de vida curta, que representa uma conversação entre o aplicativo e o armazenamento persistente.

III - Abstrai a aplicação dos Datasource ou DriverManager adjascentes.

Em relação à arquitetura do *Hibernate*, os itens I, II e III, associam-se, respectivamente, a

a) SessionFactory, Session e Connection Provider.

b) Transaction Factory, Session Factory e Connection Provider.

c) Session, Connection Provider e Extension Interfaces.

d) Connection Provider, Extension Interfaces e Session Factory.

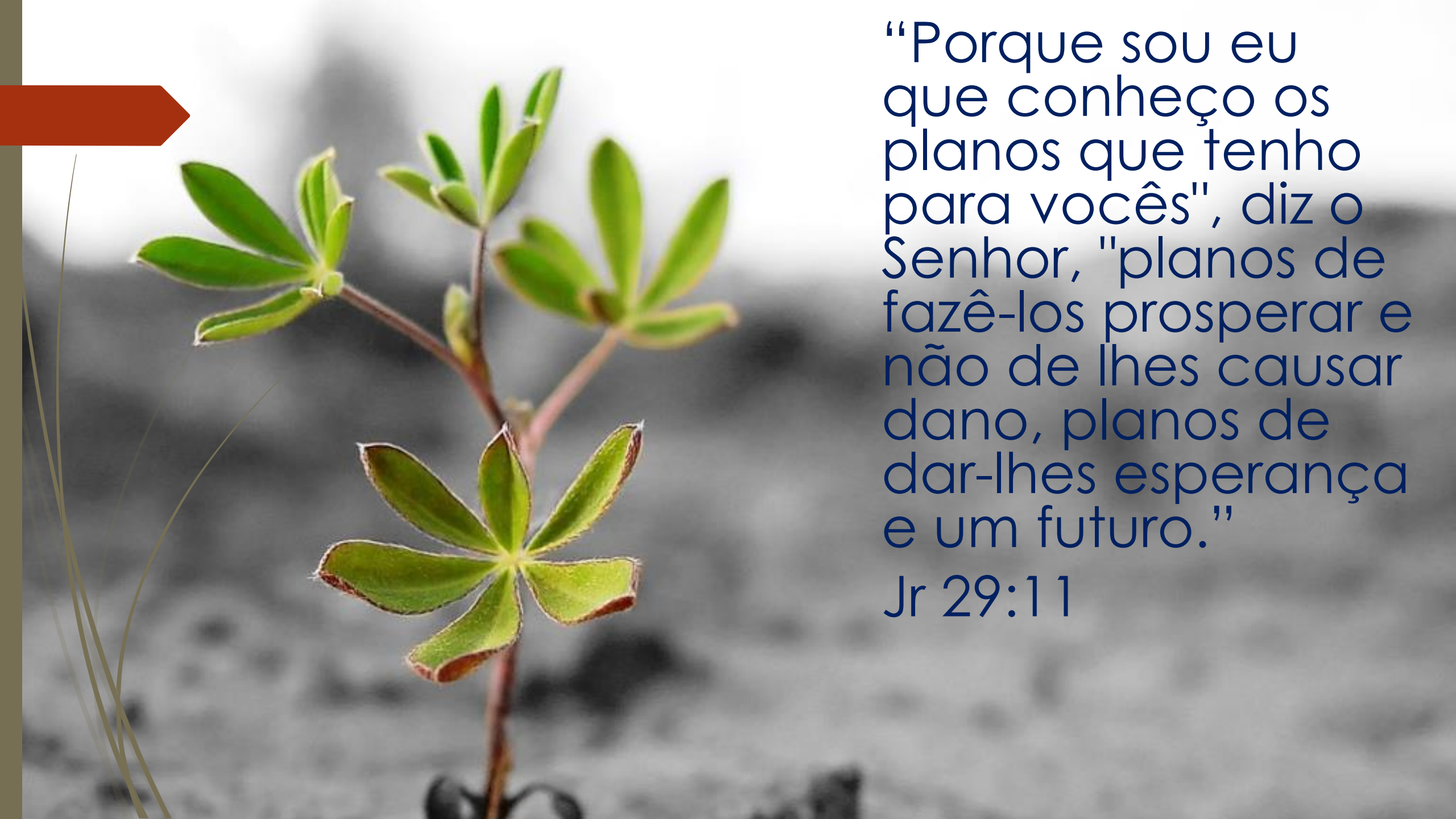
e) Connection Provider, Session e Session Factory.

GABARITO

Q76 - LETRA E.

Q77- LETRA A.

Q78- LETRA A.



“Porque sou eu que conheço os planos que tenho para vocês", diz o Senhor, "planos de fazê-los prosperar e não de lhes causar dano, planos de dar-lhes esperança e um futuro.”

Jr 29:11