PROVAS DE TI

TUDO PARA VOCÊ PASSAR

JPAHIB – Em exercícios

Rodrigo Macedo - http://www.itnerante.com.br/profile/RodrigoMacedo

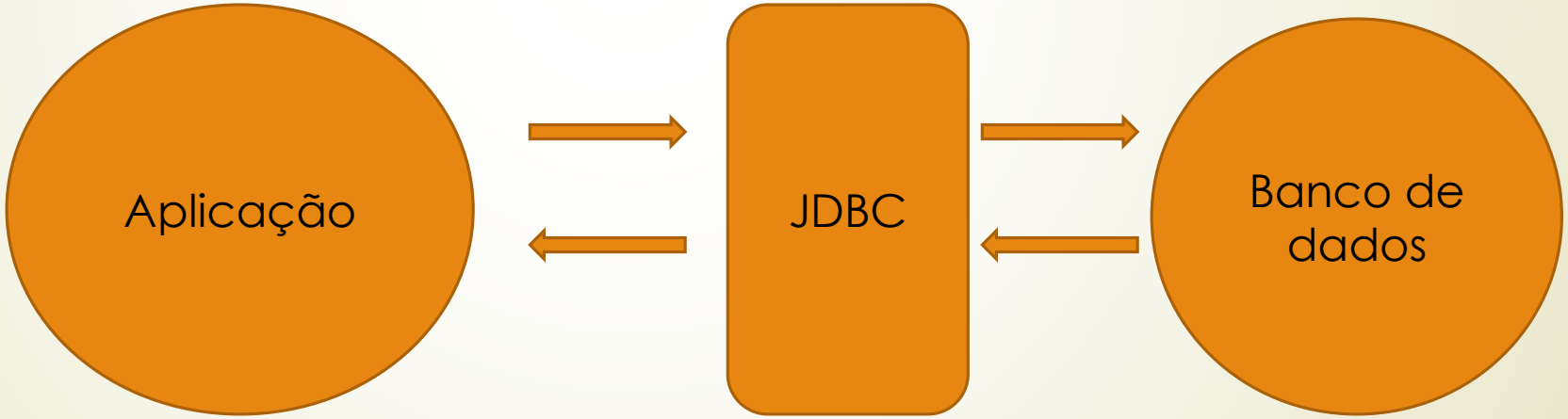
gabaritou <TI>

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.

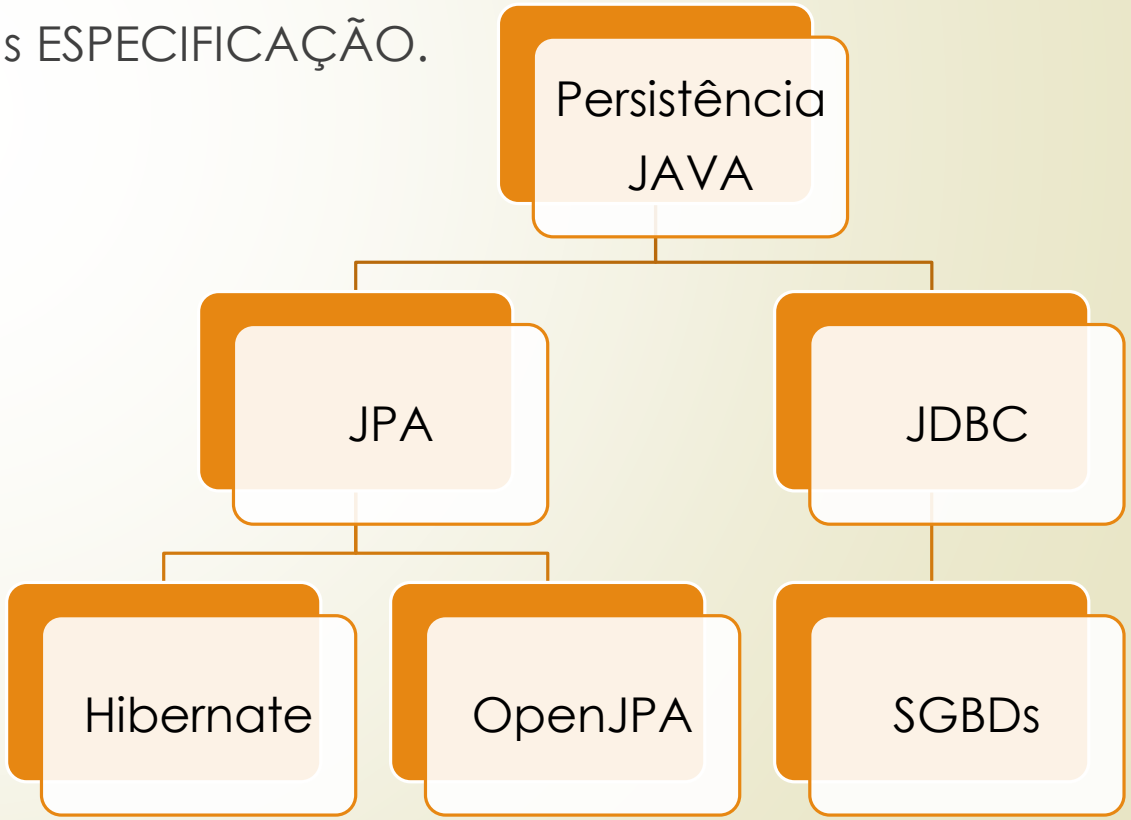
JDBC

- 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.



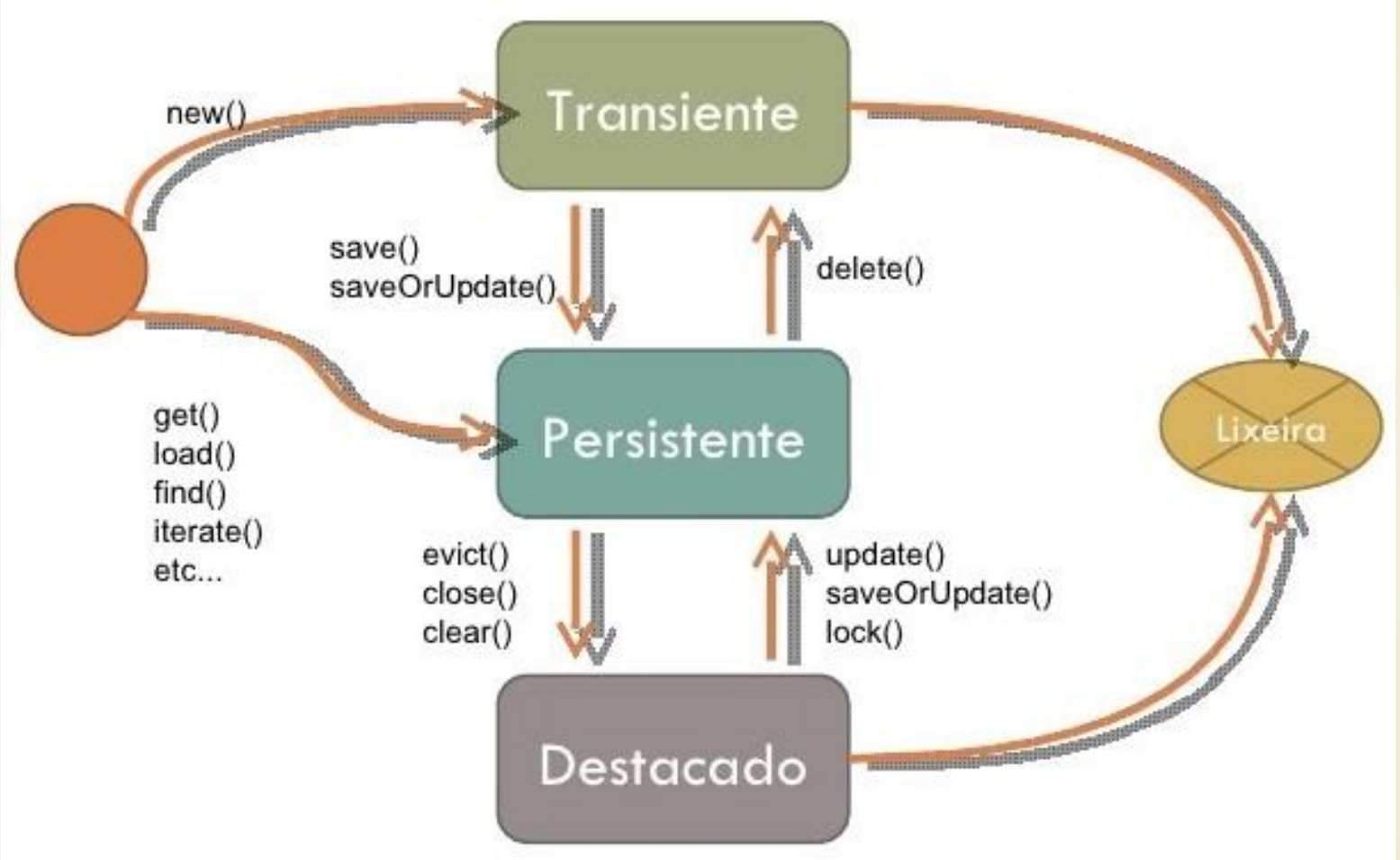
JPA – Conceitos Fundamentais

- 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.
- Não é IMPLEMENTAÇÃO, apenas ESPECIFICAÇÃO.



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


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

Q1- [CESPE SLU DF 2019] No que se refere a banco de dados relacional (SQL) e não relacional (NoSQL) e ao *framework* JPA, julgue item subsecutivo.

Situação hipotética: Na utilização de JPA (Java Persistence API) para realizar o mapeamento objeto-relacional (ORM) em uma aplicação Java, surgiu a necessidade de criar uma propriedade idade não serializável em uma classe serializável. **Assertiva**: Uma maneira de informar que essa propriedade não deve ser persistida em banco de dados é utilizar a anotação @Transient em seu método getIdade().

Q1- [CESPE SLU DF 2019] No que se refere a banco de dados relacional (SQL) e não relacional (NoSQL) e ao *framework* JPA, julgue item subsecutivo.

Situação hipotética: Na utilização de JPA (Java Persistence API) para realizar o mapeamento objeto-relacional (ORM) em uma aplicação Java, surgiu a necessidade de criar uma propriedade idade não serializável em uma classe serializável. Assertiva: Uma maneira de informar que essa propriedade não deve ser persistida em banco de dados é utilizar a anotação @Transient em seu método getIdade(). CERTO

Q2- [CONSULPLAN TRE RJ 2017] Ao utilizar JPA (*Java Persistence* API), para que um objeto torne-se um objeto persistente, precisa-se passá-lo para o estado *Managed* (gerenciado). Para isso, deve-se chamar o método _____.” Assinale a alternativa que completa corretamente a afirmativa anterior.

a) Insert
b) Create
c) Persist
d) CreateQuery

Q3) [CESPE MPE PI 2018] O Hibernate é uma solução tecnológica para ORM que aceita o uso da JPA e que permite padronizar as implementações de ORM em Java, embora ainda seja possível mapear as classes utilizando-se o XML.

Q2- [CONSULPLAN TRE RJ 2017] Ao utilizar JPA (*Java Persistence* API), para que um objeto torne-se um objeto persistente, precisa-se passá-lo para o estado *Managed* (gerenciado). Para isso, deve-se chamar o método _____.” Assinale a alternativa que completa corretamente a afirmativa anterior.

a) Insert

b) Create

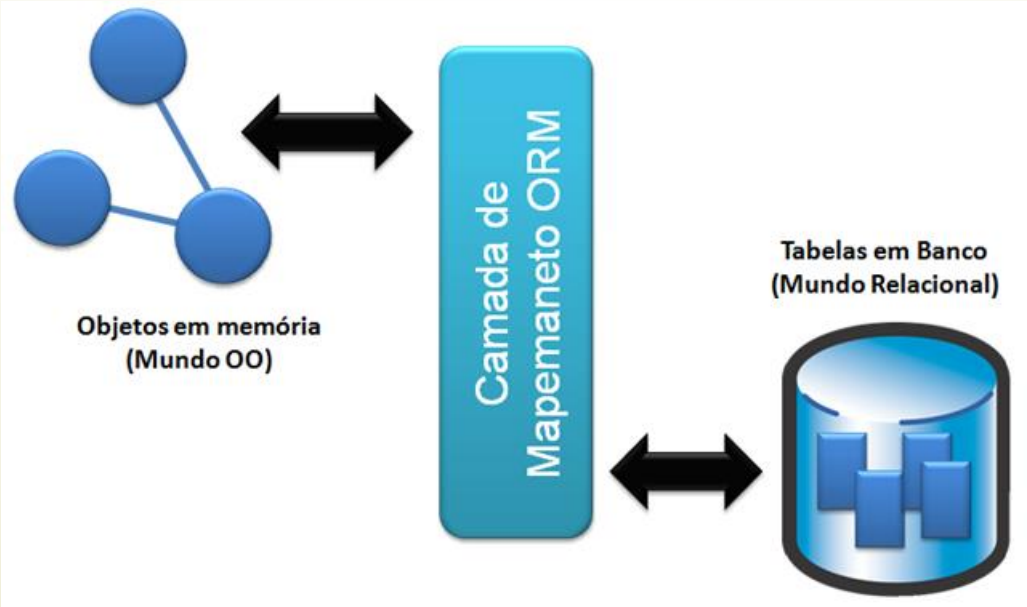
c) Persist

d) CreateQuery

Q3) [CESPE MPE PI 2018] O Hibernate é uma solução tecnológica para ORM que aceita o uso da JPA e que permite padronizar as implementações de ORM em Java, embora ainda seja possível mapear as classes utilizando-se o XML. CERTO.

ORM – Mapeamento objeto - relacional

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



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.

- **@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.

Q4- [FCC SABESP 2018] Considere a classe de entidade de um banco de dados a seguir, criada utilizando JPA.

```
 I
public class Cliente {
   II
   private String cpf;
   // outros atributos
}
```

Para indicar que é uma classe que representa uma tabela do banco de dados e que o atributo cpf representa a chave primária da tabela, as lacunas **I** e **II** devem ser preenchidas, correta e respectivamente, com

a) @Table e @Primary.
b) @Data e @PrimaryKey.
c) @Entity e @Id.
d) @Table e @Index
e) @Entity e @PrimaryKey.

Q4- [FCC SABESP 2018] Considere a classe de entidade de um banco de dados a seguir, criada utilizando JPA.

```
 I
public class Cliente {
   II
   private String cpf;
   // outros atributos
}
```

Para indicar que é uma classe que representa uma tabela do banco de dados e que o atributo cpf representa a chave primária da tabela, as lacunas **I** e **II** devem ser preenchidas, correta e respectivamente, com

a) @Table e @Primary.

b) @Data e @PrimaryKey.

c) @Entity e @Id.

d) @Table e @Index

e) @Entity e @PrimaryKey.

Q5- [FCC DPE AM 2018] Em uma aplicação que utiliza JPA, uma classe de entidade Processo.java relaciona-se com a tabela processo em um banco de dados relacional. Assim, essa classe de entidade precisa ter algumas anotações especiais, como @Entity que especifica que se trata de uma classe de entidade, e

a) @Field, que especifica que um atributo refere-se a um campo da tabela do banco de dados.

b) @Override, que especifica que o método da classe de entidade executará uma operação SQL na tabela do banco de dados.

c) @SetSQL, que especifica que o método *setter* executará uma operação de inclusão ou alteração de dados na tabela do banco de dados.

d) @EntityManager, que especifica que a classe é uma entidade gerenciada por uma unidade de persistência.

e) @Id, que indica qual atributo na classe de entidade representa a chave primária na tabela do banco de dados.

Q5- [FCC DPE AM 2018] Em uma aplicação que utiliza JPA, uma classe de entidade Processo.java relaciona-se com a tabela processo em um banco de dados relacional. Assim, essa classe de entidade precisa ter algumas anotações especiais, como @Entity que especifica que se trata de uma classe de entidade, e

a) @Field, que especifica que um atributo refere-se a um campo da tabela do banco de dados.

b) @Override, que especifica que o método da classe de entidade executará uma operação SQL na tabela do banco de dados.

c) @SetSQL, que especifica que o método *setter* executará uma operação de inclusão ou alteração de dados na tabela do banco de dados.

d) @EntityManager, que especifica que a classe é uma entidade gerenciada por uma unidade de persistência.

e) @Id, que indica qual atributo na classe de entidade representa a chave primária na tabela do banco de dados.

Q6- [IBFC EMBASA 2017] Dadas as definições abaixo, elas correspondem respectivamente às seguintes siglas

I. É uma especificação Java para a construção de interfaces de usuário baseadas em componentes para aplicações web.

II. Define um meio de mapeamento objeto-relacional para objetos Java simples e comuns (POJOs).

III. É um conjunto de classes e interfaces (API) escritas em Java que fazem o envio de instruções SQL para qualquer banco de dados relacional.

a) I-JDBC, II-JPA, III-JSF.

b) I-JPA, II-JSF, III-JDBC.

c) I-JSF, II-JPA, III-JDBC.

d) I-JSF, II-JDBC, III-JPA.

Q6- [IBFC EMBASA 2017] Dadas as definições abaixo, elas correspondem respectivamente às seguintes siglas

I. É uma especificação Java para a construção de interfaces de usuário baseadas em componentes para aplicações web.

II. Define um meio de mapeamento objeto-relacional para objetos Java simples e comuns (POJOs).

III. É um conjunto de classes e interfaces (API) escritas em Java que fazem o envio de instruções SQL para qualquer banco de dados relacional.

a) I-JDBC, II-JPA, III-JSF.

b) I-JPA, II-JSF, III-JDBC.

c) I-JSF, II-JPA, III-JDBC.

d) I-JSF, II-JDBC, III-JPA.

Q7- [IBFC EMBASA 2017] A ferramenta Hibernate

a) cuida do mapeamento de classes Java para tabelas de banco de dados.

b) possui uma API nativa incompatível com a Java Persistence API.

c) foca na modelagem de bancos de dados objeto-relacional.

d) impede a herança e o polimorfismo nas classes persistentes.

Q8- [QUADRIX CFO DF 2017] Acerca do *framework* Hibernate e da linguagem de modelagem UML 2.0, julgue o item subsequente.

O Hibernate é um *framework* que facilita muito o trabalho dos desenvolvedores. No entanto, como qualquer outra ferramenta ou tecnologia, ele possui algumas desvantagens, como, por exemplo, não integrar à parte essencial do *framework* as anotações utilizadas pela linguagem Java.

Q7- [IBFC EMBASA 2017] A ferramenta Hibernate

a) **cuida do mapeamento de classes Java para tabelas de banco de dados.**

b) possui uma API nativa incompatível com a Java Persistence API.

c) foca na modelagem de bancos de dados objeto-relacional.

d) impede a herança e o polimorfismo nas classes persistentes.

Q8- [QUADRIX CFO DF 2017] Acerca do *framework* Hibernate e da linguagem de modelagem UML 2.0, julgue o item subsequente.

O Hibernate é um *framework* que facilita muito o trabalho dos desenvolvedores. No entanto, como qualquer outra ferramenta ou tecnologia, ele possui algumas desvantagens, como, por exemplo, não integrar à parte essencial do *framework* as anotações utilizadas pela linguagem Java. ERRADO.

Q9- [COMPERVE UFRN 2018] As anotações são mecanismos importantes para configuração do funcionamento do *hibernate* para um sistema específico. Sobre essa temática, analise as afirmativas abaixo.

I Em um relacionamento bidirecional, é possível fazer uso das anotações @OneToMany, @ManyToOne e @JoinColumn.

II A anotação @ManyToMany não é suportada pelo *hibernate*.

III @PrimaryKeyJoinColumn indica que a chave primária de uma entidade é usada como chave estrangeira.

IV @Todo indica um ponto de extensão em linhas de produtos de software.

Estão corretas as afirmativas:

a) I e III.

b) I e II.

c) II e IV.

d) III e IV.

Q9- [COMPERVE UFRN 2018] As anotações são mecanismos importantes para configuração do funcionamento do *hibernate* para um sistema específico. Sobre essa temática, analise as afirmativas abaixo.

I Em um relacionamento bidirecional, é possível fazer uso das anotações @OneToMany, @ManyToOne e @JoinColumn.

II A anotação @ManyToMany não é suportada pelo *hibernate*.

III @PrimaryKeyJoinColumn indica que a chave primária de uma entidade é usada como chave estrangeira.

IV @Todo indica um ponto de extensão em linhas de produtos de software.

Estão corretas as afirmativas:

a) I e III.

b) I e II.

c) II e IV.

d) III e IV.

Q10- [CCV-UFC UFC 2019] No Hibernate 5, qual das opções abaixo contém a *annotation* responsável por especificar o nome da coluna que define a chave estrangeira relacionada com uma associação "muitos para um" (*Many To One*)?

a) @Embedded.

b) @JoinColumn.

c) @ManyToOne.

d) @ReferColumn.

e) @OrderColumn.

Q11 – [CESPE STM 2018] A respeito de construção de sistemas, julgue o item subsequente.

Em sistemas desenvolvidos em Java, o objetivo do framework Hibernate é a transformação das classes para tabelas de dados, com a geração dos comandos SQL.

Q10- [CCV-UFC UFC 2019] No Hibernate 5, qual das opções abaixo contém a *annotation* responsável por especificar o nome da coluna que define a chave estrangeira relacionada com uma associação "muitos para um" (*Many To One*)?

a) @Embedded.

b) @JoinColumn.

c) @ManyToOne.

d) @ReferColumn.

e) @OrderColumn.

Q11 – [CESPE STM 2018] A respeito de construção de sistemas, julgue o item subsequente.

Em sistemas desenvolvidos em Java, o objetivo do framework Hibernate é a transformação das classes para tabelas de dados, com a geração dos comandos SQL. CERTO.

Q12- [COMPERVE UFRN 2018] O *hibernate* é uma tecnologia bastante utilizada para o mapeamento objeto-relacional de sistemas Java. Sobre essa tecnologia, analise as afirmativas abaixo.

I @Entity deve ser utilizado para anotar classes persistentes.

II @Table é uma anotação utilizada em atributos de classes.

III @Id anota atributos cujos valores podem ser gerados automaticamente.

IV @TableKey anota atributos que representam chaves primárias.

Estão corretas as afirmativas:

a) III e IV.

b) I e II.

c) II e IV.

d) I e III.

Q13 – [QUADRIX CFO-DF 2017] Acerca do *framework* Hibernate e da linguagem de modelagem UML 2.0, julgue o item subsequente.

O Hibernate é uma ferramenta eficaz na implementação de associações entre objetos Java.

Q12- [COMPERVE UFRN 2018] O *hibernate* é uma tecnologia bastante utilizada para o mapeamento objeto-relacional de sistemas Java. Sobre essa tecnologia, analise as afirmativas abaixo.

I @Entity deve ser utilizado para anotar classes persistentes.

II @Table é uma anotação utilizada em atributos de classes.

III @Id anota atributos cujos valores podem ser gerados automaticamente.

IV @TableKey anota atributos que representam chaves primárias.

Estão corretas as afirmativas:

a) III e IV.

b) I e II.

c) II e IV.

d) I e III.

Q13 – [QUADRIX CFO-DF 2017] Acerca do *framework* Hibernate e da linguagem de modelagem UML 2.0, julgue o item subsequente.

O Hibernate é uma ferramenta eficaz na implementação de associações entre objetos Java. CERTO.

Q14- [CCV-UFC 2016] O framework de persistência de dados Hibernate possui as seguintes annotations para designar os tipos de associação entre entidades:

a) @ManyToOne, @AllToMany, @OneToOne, @ManyToMany.

b) @ManyToOne, @OneToMany, @OneToOne, @ManyToMany.

c) @ManyToOne, @OneToMany, @OneToOne, @ManyToAll.

d) @ManyToOne, @OneToAll, @OneToOne, @ManyToMany.

e) @ManyToAll, @OneToAll, @OneToOne, @ManyToMany.

Q15 – [CESPE TCE PA] No que se refere ao desenvolvimento de sistemas e às linguagens de programação Java e JavaScript, julgue o item a seguir.

Empresa de desenvolvimento que opte pela utilização do Hibernate em seus sistemas enfrentará dificuldades à medida que seus projetos forem crescendo, devido ao fato de o Hibernate ser considerado inapropriado para a execução de trabalhos em uma arquitetura altamente escalável.

Q14- [CCV-UFC 2016] O framework de persistência de dados Hibernate possui as seguintes annotations para designar os tipos de associação entre entidades:

a) @ManyToOne, @AllToMany, @OneToOne, @ManyToMany.

b) @ManyToOne, @OneToMany, @OneToOne, @ManyToMany.

c) @ManyToOne, @OneToMany, @OneToOne, @ManyToAll.

d) @ManyToOne, @OneToAll, @OneToOne, @ManyToMany.

e) @ManyToAll, @OneToAll, @OneToOne, @ManyToMany.

Q15 – [CESPE TCE PA] No que se refere ao desenvolvimento de sistemas e às linguagens de programação Java e JavaScript, julgue o item a seguir.

Empresa de desenvolvimento que opte pela utilização do Hibernate em seus sistemas enfrentará dificuldades à medida que seus projetos forem crescendo, devido ao fato de o Hibernate ser considerado inapropriado para a execução de trabalhos em uma arquitetura altamente escalável. ERRADO.

Q16- [FAURGS HCPA 2016] Qual é a anotação no Hibernate que permite marcar uma propriedade como identificador?

a) @Entity.

b) @Id.

c) @Identifier.

d) @Person.

e) @Key.

Q16- [FAURGS HCPA 2016] Qual é a anotação no Hibernate que permite marcar uma propriedade como identificador?

a) @Entity.

b) @Id.

c) @Identifier.

d) @Person.

e) @Key.

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>
```

Q17- [CCV-UFC UFC 2016] Analise o mapeamento Hibernate abaixo:

```
@TAG_SUB(
        name = "buscaCandidatoPorCpf",
        query = "from Candidato c where c.cpf = :cpf"
)
@Entity
@Table(name = "candidato")
public class Candidato implements java.io.Serializable {
...
```

A tag que substitui corretamente @TAG_SUB é:

a) @Access
b) @NamedQuery
c) @Embeddable
d) @GenerateQuery
e) @SecondaryTable

Q17- [CCV-UFC UFC 2016] Analise o mapeamento Hibernate abaixo:

```
@TAG_SUB(
        name = "buscaCandidatoPorCpf",
        query = "from Candidato c where c.cpf = :cpf"
)
@Entity
@Table(name = "candidato")
public class Candidato implements java.io.Serializable {
...
```

A tag que substitui corretamente @TAG_SUB é:

a) @Access

b) @NamedQuery

c) @Embeddable

d) @GenerateQuery

e) @SecondaryTable

Q18- [FUNCAB CREA AC 2016] Em um projeto de software com base em Java, quando se utiliza o framework Hibernate, deseja-se:

a) automatizar os processos de construção de classes e também os testes de distribuição.

b) criar um container para instanciar classes java, definindo as dependências entre elas.

c) fornecer um controlador para facilitar a escrita de moldes padronizados no formato XML.

d) permitir que a camada de apresentação contenha cabeçalho e rodapé independentes.

e) transformar as classes em Java para tabelas de dados, gerando chamadas SQL.

Q18- [FUNCAB CREA AC 2016] Em um projeto de software com base em Java, quando se utiliza o framework Hibernate, deseja-se:

a) automatizar os processos de construção de classes e também os testes de distribuição.

b) criar um container para instanciar classes java, definindo as dependências entre elas.

c) fornecer um controlador para facilitar a escrita de moldes padronizados no formato XML.

d) permitir que a camada de apresentação contenha cabeçalho e rodapé independentes.

e) transformar as classes em Java para tabelas de dados, gerando chamadas SQL.

Q19- [CCV-UFC UFC 2016] O Hibernate é um ORM (*Object Relational Mapping*) com o intuito de tornar o mapeamento dos objetos da aplicação na base de dados mais automatizada. Sobre esse ORM, é correto afirmar:

a) Para configurar o acesso ao banco através do Hibernate, é necessário criar o arquivo de configuração (hibernate.cfg.xml), contendo as informações de acesso a base de dados.

b) Por ser um ORM para ser utilizado com o Java, o Hibernate implementa os recursos especificados no JPA (Java Persistence API), não proporcionando recursos adicionais.

c) No mapeamento das entidades no arquivo XML, não importa qual o tipo do atributo do objeto, visto que tal informação é dinamicamente identificada durante a execução.

d) Para ser possível a utilização de anotações (annotations) para mapear as classes, é obrigatório utilizar as bibliotecas do JPA juntamente com as bibliotecas do Hibernate.

e) O mapeamento dos objetos com o Hibernate somente pode ser feito em arquivo XML. Tais arquivos geralmente possuem a extenção .hbm.xml.

Q19- [CCV-UFC UFC 2016] O Hibernate é um ORM (*Object Relational Mapping*) com o intuito de tornar o mapeamento dos objetos da aplicação na base de dados mais automatizada. Sobre esse ORM, é correto afirmar:

a) **Para configurar o acesso ao banco através do Hibernate, é necessário criar o arquivo de configuração (hibernate.cfg.xml), contendo as informações de acesso a base de dados.**

b) Por ser um ORM para ser utilizado com o Java, o Hibernate implementa os recursos especificados no JPA (Java Persistence API), não proporcionando recursos adicionais.

c) No mapeamento das entidades no arquivo XML, não importa qual o tipo do atributo do objeto, visto que tal informação é dinamicamente identificada durante a execução.

d) Para ser possível a utilização de anotações (annotations) para mapear as classes, é obrigatório utilizar as bibliotecas do JPA juntamente com as bibliotecas do Hibernate.

e) O mapeamento dos objetos com o Hibernate somente pode ser feito em arquivo XML. Tais arquivos geralmente possuem a extenção .hbm.xml.

Q20- [FCC TRE MT 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.

Q20- [FCC TRE MT 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.

GABARITO

Q1 – CERTO.

Q2 - LETRA C.

Q3 - CERTO.

Q4 - LETRA C.

Q5 - LETRA E.

Q6 - LETRA C.

Q7 – LETRA A.

Q8 – ERRADO.

Q9 - LETRA A.

Q10 - LETRA B.

Q11 – CERTO.

Q12 - LETRA D.

Q13 - CERTO.

Q14 – LETRA B.

Q15 – ERRADO.

Q16 - LETRA B.

Q17 – LETRA B.

Q18 – LETRA E.

Q19 - LETRA A.

Q20 – LETRA D.