

## Persistência de Objetos

Fausto Maranhão Ayres

### 7 Persistência com JPA (Atributos)

#### Mapeamento de tabela e coluna

**@Table** especifica a tabela e **@Column** especifica a coluna

**@Entity**

Renomeia a tabela  
pessoa para gente

```
@Table(name="gente", uniqueConstraints=  
    @UniqueConstraint(columnNames = {"nome", "mae"}))
```

```
public class Pessoa {
```

```
    @Id
```

```
    @Column(name="codigo")
```

Renomeia a coluna  
id para codigo

```
    private int id;
```

```
    @Column(length=100, unique=false,  
            nullable=false,  
            updatable=false)
```

```
    private String nome; //length=100
```

```
    private String mae; //length=255
```

```
    ...
```

```
}
```

```
create table gente(  
    codigo integer,  
    nome varchar(100) not null  
    mae varchar(255)  
    ...  
)
```

Tamanho default  
para String

# Mapeamento de Data (API javatime)

```
@Entity
public class Pessoa {
    @Id
    private int id;
    private String nome;

    @Column(columnDefinition = "DATE")
    private LocalDate data = LocalDate.now();

    @Column(columnDefinition = "TIMESTAMP")
    private LocalDateTime datahora = LocalDateTime.now();

    ...
}
```

fausto.ayres@ifpb.edu.br

3

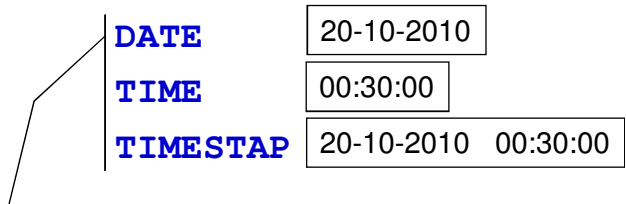
# Mapeamento de Data (API Date)

**@Temporal** configura o tipo de data

```
@Entity
public class Pessoa {
    @Id
    private int id;
    private String nome;

    @Temporal(TemporalType.DATE)
    private Date dtcadastro = new Date();

    ...
}
```



The diagram illustrates the mapping of the `@Temporal(TemporalType.DATE)` annotation to different database types and their formats:

Type	Format
DATE	20-10-2010
TIME	00:30:00
TIMESTAMP	20-10-2010 00:30:00

fausto.ayres@ifpb.edu.br

4

# Mapeamento de Enumeração

**@Enumerated** inclui o texto da enumeração. Sem ela, apenas o índice (0,1,2) será armazenado

```
public class Pessoa
...
@Enumerated(EnumType.STRING)
private Sexo sexo;
```

```
public enum Sexo{
    MASCULINO, FEMININO, OUTRO;
}
```

Sexo.java

# Mapeamento de Campo longo

**@Lob** - para imagens, textos longos, etc

Ex:

```
@Lob
private byte[] foto;
```

```
@Lob
private String texto;
```

# Atributo não persistente

## @Transient – para atributos temporários

```
@Transient  
private int idade;
```

OBS:

O atributo **idade** não deve ser gravado no banco, pois muda com o tempo.

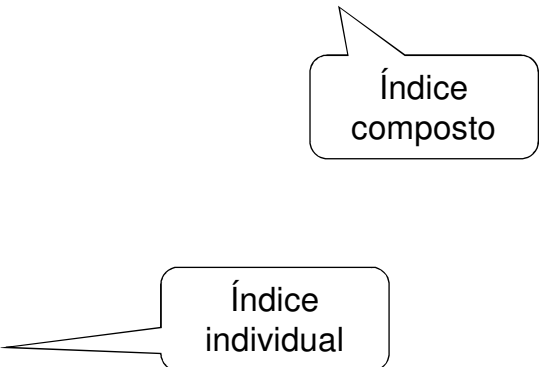
Ele deve ser recalculado toda vez que um objeto Pessoa for lido do banco, usando a sua data de nascimento.

fausto.ayres@ifpb.edu.br

7

## Criação de Índices (individual e composto)

```
@Entity  
@Index(name="IND",columnNames={"nome","sobrenome"})  
public class Empregado{  
    @Id  
    @GeneratedValue...  
    private int id;  
  
    private String nome;  
  
    @Index  
    private String sobrenome;
```



SQL:

**create index** IND on empregado (nome, sobrenome)

**create index** INDEX\_EMPREGADO\_SOBRENOME on empregado (sobrenome)

fausto.ayres@ifpb.edu.br

8

# Objetos embutidos (não-entidades)

- **Entidade** é uma classe com as anotações `@Entity` e `@Id`
- **Não-entidade** é uma classe sem as anotações `@Entity` e `@Id` - sem chave primária
  - Esses objetos (não-entidades) não terão tabela própria e serão **embutidos** dentro de uma tabela (de entidades)

fausto.ayres@ifpb.edu.br

9

## Embutindo uma Não-entidade em uma Entidade

```
@Embeddable // @Entity
public class Logradouro {
    private String rua;
    private String bairro;
    public Logradouro() {}
    public Logradouro(...) ..
}
```

Não-Entidade

```
@Entity
public class Pessoa {
    @Id
    private int id;
    private String nome;
    @Embedded
    private Logradouro endereco;
}
```



Entidade

Pessoa			
id	nome	rua	bairro

Não-entidade

fausto.ayres@ifpb.edu.br

10

# CONFIGURAÇÕES

fausto.ayres@ifpb.edu.br


11

## Eclipselink + postgresql

```
<persistence-unit name="agenda-eclipselink" transaction-type="RESOURCE_LOCAL">
  <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
  <exclude-unlisted-classes>false</exclude-unlisted-classes>

  <properties>
    <property name="javax.persistence.jdbc.driver" value="org.postgresql.Driver" />
    <property name="javax.persistence.jdbc.url" value="jdbc:postgresql://localhost:5432/agenda" />
    <property name="javax.persistence.jdbc.user" value="postgres" />
    <property name="javax.persistence.jdbc.password" value="ifpb" />

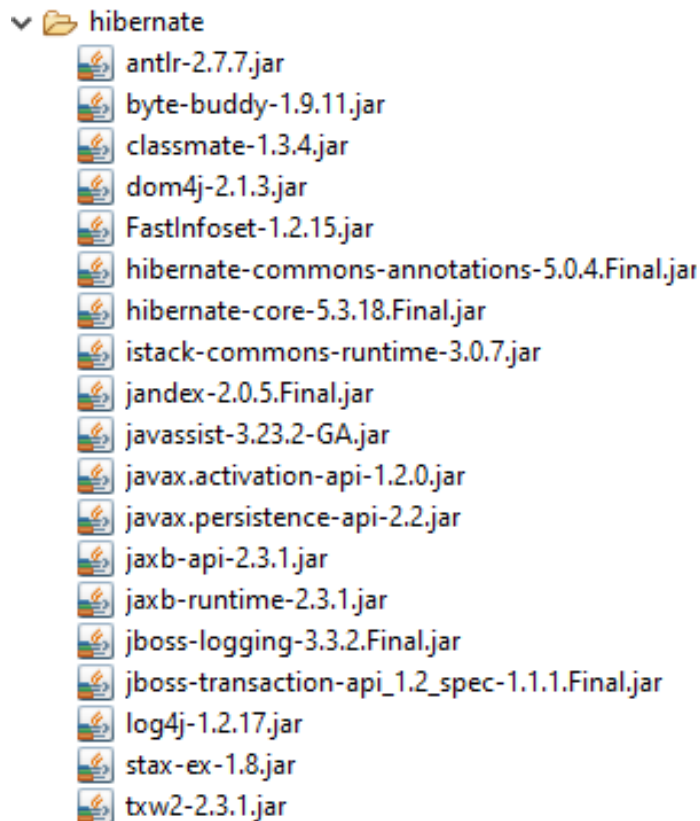
    <property name="javax.persistence.schema-generation.database.action" value="create"/>
    <property name="javax.persistence.schema-generation.scripts.action" value="create"/>
    <property name="javax.persistence.schema-generation.scripts.create-target" value="ddl.txt"/>
    <property name="eclipselink.logging.level" value="fine"/>
    <property name="eclipselink.logging.file" value="log.txt"/>
  </properties>
```



fausto.ayres@ifpb.edu.br

12

# Bibliotecas do Hibernate (5.3.18)



fausto.ayres@ifpb.edu.br

13

## Hibernate + postgresql

```
<persistence-unit name="agenda-hibernate" transaction-type="RESOURCE_LOCAL">
  <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
  <exclude-unlisted-classes>false</exclude-unlisted-classes>
  <validation-mode>NONE</validation-mode>
  <properties>
    <property name="javax.persistence.jdbc.driver" value="org.postgresql.Driver" />
    <property name="javax.persistence.jdbc.url" value="jdbc:postgresql://localhost:5432/agenda" />
    <property name="javax.persistence.jdbc.user" value="postgres" />
    <property name="javax.persistence.jdbc.password" value="ifpb" />

    <property name="hibernate.dialect" value="org.hibernate.dialect.PostgreSQLDialect" />
    <property name="hibernate.hbm2ddl.auto" value="update" />
    <property name="hibernate.generate_statistics" value="true" />
    <property name="hibernate.format_sql" value="true" />
    <property name="hibernate.show_sql" value="false" />
    <property name="hibernate.use_sql_comments" value="true" />
  </properties>
```

O arquivo log.txt é configurado separadamente

fausto.ayres@ifpb.edu.br

14

# Hibernate + mysql

```
<properties>
  <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver" />
  <property name="javax.persistence.jdbc.url" value="jdbc:mysql://127.0.0.1:3306/agenda" />
  <property name="javax.persistence.jdbc.user" value="root" />
  <property name="javax.persistence.jdbc.password" value="" />

  <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect" />
  <property name="hibernate.hbm2ddl.auto" value="update" />
  <property name="hibernate.show_sql" value="true" />
  <property name="hibernate.use_sql_comments" value="false" />
  <property name="hibernate.format_sql" value="true" />
</properties>
```

## Configuração do log (Log4j.properties)

```
log4j.rootLogger=debug, file
# Direct log messages to a log file
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=log.txt
log4j.appender.file.MaxFileSize=1MB
log4j.appender.file.MaxBackupIndex=1
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n
```