

JPA ANNOTATIONS

CRIANDO ANOTAÇÕES

```
@Target({ElementType.FIELD, ElementType.METHOD})
@Retention(RetentionPolicy.RUNTIME)
public @interface Coluna {
    String nome();
    boolean obrigatorio();
    int tamanho() default 255;
}
```

@interface - Palavra reservada para definir uma classe como anotação.
Definir parâmetros de configurações da anotação:
Ex: String nome();
boolean obrigatorio();
int tamanho() default 255; // default - definir padrão quando uso opcional

@Retention - Anotação para classe anotação que determina até quando estará presente.
Valores: SOURCE - até .java
CLASS - até .class
RUNTIME - até jvm

@Target - Escopo da anotação, definindo onde a anotação pode ser utilizada.
Valores: TYPE - classe
FIELD - atributos
METHOD - metodos

ANOTAÇÕES DE CLASSE

```
@Entity
@Table(name = "TAB_CLIENTE")
@SequenceGenerator(name = "cliente", sequenceName = "SQ_TAB_CLIENTE", allocationSize = 1)
public class Cliente {}
```

@Entity - Defini a classe java como uma entidade no banco de dados.

@Table - Indica qual será o nome da tabela no banco de dados.

@SequenceGenerator - Define que a tabela terá um SEQUENCE para chave-primária da tabela
name - nome do SequenceGenerator que será usado no atributo que corresponde a chave-primária
sequenceName - nome da SEQUENCE no banco de dados
allocationSize - incremento para a SEQUENCE

ANOTAÇÕES DE ATRIBUTOS

```
@Id
@Column(name = "cd_cliente")
@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "cliente")
private int codigo;

@Column(name = "nm_cliente", nullable = false, length = 100)
private String nome;
```

@Id - Indica o atributo como chave-primário no banco de dados.

@Column - Define o nome da coluna no banco de dados.

Obs.: não é necessário passar o parâmetro 'nullable' por já estar anotado com @Id

@GeneratedValue - Indica o atributo que receberá o SequenceGenerator definido na classe
strategy - indica como a persistência deve atribuir a chave-primária
generator - SEQUENCE no java, definido no 'name' da SequenceGenerator

OUTRAS ANOTAÇÕES DE ATRIBUTOS

@Temporal - Utilizado com atributos do tipo Calendar ou Date para datas no banco de dados.
Ex: @Temporal(TemporalType.DATE)

Valores: DATE - somente a data
TIME - somente as horas e minutos
TIMESTAMP - data e horas

@CreationTimestamp - Gravar a data e hora de cadastro no banco de dados de forma automática.

@Enumerated - Utilizado com atributos do tipo Enum
Ex: @Enumerated(EnumType.STRING)

Valores: ORDINAL - armazena a posição do elemento da enum
STRING - armazena o texto do elemento da enum

@Lob - Mapear um campo BLOB no oracle para armazenar arquivos no banco de dados.
O atributo deve ser um array de byte
Ex: @Lob
Private byte[] foto;

@Transient - Anotação que informa atributo que não será mapeado no banco de dados.

ANOTAÇÕES DE MÉTODOS

@PrePersist - Utilizado para executar o método antes de persistir no banco de dados.