# **ESTUDO DE CASO**

O foco desse trabalho é desenvolver um framework que atenda as necessidades dos programadores e concilie com a facilidade de utilização.

## Tecnologias

Para o desenvolvimento do ORM a linguagem PHP foi escolhida. PHP é uma sigla em inglês que significa *Hypertext PreProcessor* (ou em português Pré-Processador de Hipertexto).

O PHP é uma linguagem de *script*, largamente utilizada, principalmente para desenvolvimento web. Seu uso é extremamente simples para iniciantes e oferece diversos recursos para desenvolvedores profissionais.

Outro motivo para o grande uso do PHP é o fato de ser *open source* e poder ser utilizado na maioria dos sistemas operacionais, incluindo Linux, variantes Unix, Mac OS e Microsoft Windows.

## Mapeamento

Para realizar o mapeamento dos modelos de dados, foi utilizado o conceito de *Annotations*.

As *annotations* adicionam às classes do modelo e às suas propriedades, comportamentos e informações adicionais sobre as elas. Ou seja, através do uso de *annotations*, pode-se adicionar às classes informações para mapear tabelas do banco de dados, e adicionar às propriedades da classe para mapear as colunas de uma tabela do banco de dados.

### **Mapeamento Simples**

Como mostrado na Listagem 1, pode-se ver um exemplo de um mapeamento simples da classe “Pessoa”:

1. namespace App;
2. /\*\*
3. \* @ORM/Entity
4. \*/
5. class Pessoa {
6. /\*\*
7. \* @ORM/Id
8. \* @ORM/Generated
9. \* @ORM/Column(name=pessoa\_id, type=int)
10. \*/
11. public $id;
12. /\*\*
13. \* @ORM/Column(name=nome, type=string, length=50)
14. \*/
15. public $nome;
16. /\*\*
17. \* @ORM/Column(name=data\_nasc, type=date)
18. \*/
19. public $dataNasc;
20. }

Listagem 1 – Classe Pessoa Mapeada

Na linha 5, a *annotation* “@ORM\Entity” informa para o ORM que a classe pessoa deve ser tratada como uma “Entity” (em português, Entidade), ou seja, a classe “Pessoa” representa uma tabela com o nome de “pessoa” no banco de dados.

Na linha 9, a *annotation* “@ORM/Id” informa que a propriedade “$id” (declarada na linha 13) é a chave primária da tabela “pessoa”. Essa propriedade será utilizada pelo ORM quando for necessário fazer consultas relacionando múltiplas tabelas.

Na linha 10, a *annotation* “@ORM/Generated” informa que a propriedade “$id” é um valor de geração automática e sequencial. O ORM deverá resolver o valor desse campo de acordo com o *Driver* utilizado para a conexão com banco de dados utilizado. Os *Drivers* serão abordados mais profundamente futuramente.

Nas linhas 11, 16, 21, a *annotation* “@ORM/Column” informa respectivamente que as propriedades “$id”, “$nome”, “$dataNasc”, estão mapeando colunas específicas da tabela “pessoa”.

A *annotation* “@ORM/Column” possui propriedades, a propriedade “name” define o nome da coluna mapeada, como no exemplo acima, as propriedades da classe “Pessoa” representam as colunas da tabela “pessoa” com os nomes “pessoa\_id”, “nome”, “data\_nasc”.

A propriedade “type” define os tipos dos dados a serem mapeados, como no exemplo acima, as colunas mapeadas acima são respectivamente os tipos “int”, “string” e “date”.

A propriedade “length” define o tamanho do campo do tipo “string”, no caso da propriedade “$nome” mostrada no exemplo acima, a propriedade “length” define que o tamanho do campo é igual à 50. Essa informação auxilia o ORM a criar a tabela, caso esse comportamento seja definido. Esse assunto será abordado mais à frente.

Além das propriedades citadas acima, também existem as propriedades “scale”, “precision”, “unique”, “nullable”. Essas propriedades auxiliam o ORM na tarefa de criar a tabela. As propriedades “scale” e “precision” definem o tamanho e a precisão de uma coluna do tipo número real. A propriedade “unique” define que a coluna deve ser indexada como única na tabela, ou seja, não deve possuir valores que repitam na tabela. E por fim, a propriedade “nullable” define que a coluna não deve aceitar valores nulos.

Além dos tipos citados no exemplo acima, outros tipos podem ser utilizados, esses tipos são definidos no Driver utilizado para a conexão. Esse assunto será abordado mais a fundo na seção de *Drivers*.

### **Mapeamento um Relacionamento**

Na linha 10, a *annotation* “@ORM/Generated” informa que a propriedade “$id” é um valor de geração automática e sequencial. O ORM deverá resolver o valor desse campo de acordo com o *Driver* utilizado para a conexão com banco de dados utilizado. Os *Drivers* serão abordados mais profundamente futuramente.

#### **Um para Um**

Texto.

#### **Um para Muitos**

Texto.

#### **Muitos para Muitos**

Texto.

## Conexões

### **Driver**

### **Criação das tabelas**

Script:

Open Source: