**Roteiro 3**

**Objetivos:**

*\* Apresentar a sintaxe básica para utilização dos métodos* ***\_\_get*** *e* ***\_\_set*** *em php.*

**Ferramentas necessárias**

*Navegador, NetBeans, Servidor Apache.*

1. Métodos \_\_get e \_\_set

*A versão 5 do PHP incorporou alguns métodos muito interessantes para utilzação em Orietação a Objeto (OO), os métodos \_\_get e \_\_set. A técnica chama-se overloading e permite que criemos uma forma padrão de acesso e a atualização de métodos sem precisar criar manualmente o get e o set para cada atributo.*

*Anteriormente criávamos os objetos obedecendo a seguinte estrutura:*

class Destaque{

private $link;

private $id\_destaque;

function getLink() {

return $this->link;

}

function setLink($link) {

$this->link = $link;

}

function getId\_destaque() {

return $this->id\_destaque;

}

function setId\_destaque($id\_destaque) {

$this->id\_destaque = $id\_destaque;

}

}

*Repare que para cara atributo que é declarado como private, eu declaro duas funções, uma como get e outra como set. Se for só dois atributos beleza, mas como a maioria dos dados representados possui vários atributos o processo é trabalhoso.*

*Para instânciar e utilizar o objeto é necessário mais as seguintes linhas:*

$destaque = new Destaque(); //instância da classe destaque

$destaque->setId\_destaque (21); // passando para o atributo Id\_destaque o valor 21

echo $destaque->getId\_destaque(); //exibindo o que foi carregado no atributo

*Seguindo a técnica de overloading, utilizando os métodos mágicos do PHP, a estrutura fica da seguinte forma:*

class Actor\_bean {  
 private $id;  
 private $name;

//método de captura de valores para o objeto  
 public function \_\_get($key){  
 return $this->$key;  
 }//end \_\_get()

//método de retorno de valores do objeto  
 public function \_\_set($key, $value){  
 $this->$key = $value;  
 }//end \_\_set()

}

Observe que há somente um método \_\_get e um método \_\_set que servirá para todos os atributos da classe. Quando é utilizado o método \_\_set, ele captura o valor e o carrega no atributo correspondente. Quando é invocado o método \_\_get ele retorna o que está carregado. Ganha-se com isso muito tempo, pois para cada atributo novo acrescenta-se somente uma linha, que é a declaração privada do atributo e o método é montado dinamicamente. Outro benefício é a forma de chamar o atributo:

$actor = new Actor\_bean(); //a instância da classe é feita igual

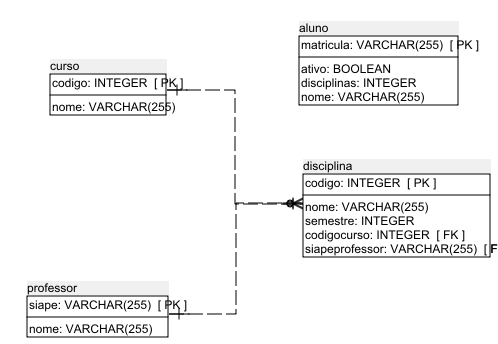
$actor->name = ‘Morgan Freeman’; //método \_\_set captura a string e carrega o atributo name

echo $actor->name //retorna o que foi carregado anteriormente

Repare que após a instância da classe, tratamos o atributo direto pelo nome ao invés de chamar o nome do método. Isso torna o código mais limpo, os erros mais tratáveis e economiza tempo de desenvolvimento.

***Atividade***

1. *Criar classes, com os métodos \_\_get e \_\_set para todos atributos que represente a estrutura das seguintes tabelas, sendo que o padrão para criação das classes é* ***NomeTabela.class.php****.*

**