

## Classe Usuário

Na aula de hoje daremos sequência ao desenvolvimento do nosso projeto completo no padrão MVC com PHP Orientado a Objetos. Vamos modelar a classe Usuário.

A classe Usuário que será responsável por gerenciar os dados do usuário no projeto. Seu nome será "*Usuario*", então novamente dentro da pasta *Model*, crie este arquivo PHP, não se esquecendo de colocar os delimitadores php "<?php ... ?>". O desenvolvimento desta classe será baseado no diagrama a seguir:



Diagrama da Classe Usuário

Dentro dos delimitadores PHP, devemos criar a classe e seus atributos:

- idusuario - código único de cada registro de usuário;
- nome - nome do usuário;
- cpf - CPF do usuário;
- dataNascimento - data de nascimento do usuário.
- email - e-mail do usuário
- senha - senha para acesso ao site.

A codificação deverá ficar dessa forma:

```
class Usuario {
    private $id;
    private $nome;
    private $cpf;
    private $email;
    private $dataNascimento;
    private $senha;
}
```

O próximo passo é criar todos os métodos *getters* e *setters* para acesso aos atributos privados da classe. Então, seguindo os padrões de projeto vamos codificá-los:

```
//ID
public function setID($id)
{
    $this->id = $id;
}
public function getID()
{
    return $this->id;
}
//nome
public function setNome($nome)
{
    $this->nome = $nome;
}
```

```
public function getNome()
{
    return $this->nome;
}
//cpf
public function setCPF($cpf)
{
    $this->cpf = $cpf;
}
public function getCPF()
{
    return $this->cpf;
}
//Email
public function setEmail($email)
{
    $this->email = $email;
}
public function getEmail()
{
    return $this->email;
}
//Data de nascimento
public function setDataNascimento($dataNascimento)
{
    $this->dataNascimento = $dataNascimento;
}
public function getDataNascimento()
{
    return $this->dataNascimento;
}
// Senha
public function setSenha($senha)
{
    $this->senha = $senha;
}
public function getSenha()
{
    return $this->senha;
}
```

Agora vamos partir para a codificação dos métodos específicos, que são três: *inserirBD*, *carregarUsuário* e *atualizarBD*.

Todos estão relacionados a operações com o Banco de Dados.

### Método **inserirBD**

Primeiramente vamos codificar o método "*inserirBD*", sua função será inserir no Banco de Dados as informações do usuário quando for invocado, mas neste primeiro momento não vamos inserir todas as informações.

Ao realizar o primeiro contato com a página e, conseqüentemente, realizar o cadastro, o usuário precisará apenas fornecer: Nome, CPF, e-mail e senha. Esse método é bem popular para agilizar o cadastro e o restante das informações podem ser inseridas em futuras atualizações. Então vamos codificar.

```
public function inserirBD()
{
    require_once 'ConexaoBD.php';
    $con = new ConexaoBD();
    $conn = $con->conectar();
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sql = "INSERT INTO usuario (nome, cpf, email, senha) VALUES
        ('".$this->nome."', '".$this->cpf."', '".$this->email."', '".$this->senha."')";
    if ($conn->query($sql) === TRUE) {
        $this->id = mysqli_insert_id($conn);
        $conn->close();
        return TRUE;
    }
    else {
        $conn->close();
        return FALSE;
    }
}
```

Perceba que o método foi definido como público para ser possível sua utilização após a instância de um objeto da classe Usuário, depois é necessária a inclusão da classe *conexaoBD*, para que seja possível a utilização da mesma.

Então, na sequência devemos:

- Instanciar objeto: `$con = new ConexaoBD();`
- Conectar ao Banco de dados e retornar essa conexão, atribuindo a um novo objeto: `$conn = $con->conectar();`
- Verificar se a conexão foi realizada: `if ($conn->connect_error);`

Se tudo ocorrer corretamente, sabemos que a conexão foi realizada e assim devemos escrever a sentença SQL:

```
$sql = "INSERT INTO usuario (nome, cpf, email, senha) VALUES
        ('".$this->nome."', '".$this->cpf."', '".$this->email."', '".$this->senha."')";
```

➡ Perceba que utilizamos os atributos da classe para montar essa sentença, ou seja, o programador será forçado a popular os atributos por meio dos métodos *getters and setters* para que o desenvolvimento fique padronizado.

Por fim, executamos a sentença SQL e verificamos se tudo ocorreu bem:

```
if ($conn->query($sql) === TRUE).
```

- Caso positivo, obtemos o id gerado no banco de dados para que seja inserido no objeto, fechamos a conexão e retornamos *TRUE*.
- Caso negativo, fechamos a conexão e retornamos *FALSE*.

Pronto, o primeiro método específico foi finalizado!

Vamos para o próximo método:

**Método carregarUsuario.**

```
public function carregarUsuario($cpf)
{
    require_once 'ConexaoBD.php';
    $con = new ConexaoBD();
    $conn = $con->conectar();
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sql = "SELECT * FROM usuario WHERE cpf = ".$cpf ;
    $re = $conn->query($sql);
    $r = $re->fetch_object();
    if($r != null)
    {
        $this->id = $r->idusuario;
        $this->nome = $r->nome;
        $this->email = $r->email;
        $this->cpf = $r->cpf;
        $this->dataNascimento = $r->dataNascimento;
        $this->senha = $r->senha;
        $conn->close();
        return true;
    }
    else
    {
        $conn->close();
        return false;
    }
}
```

Este método segue o mesmo padrão:

- Inclusão da Classe *ConexaoBD*;
- Instância do Objeto da Classe *ConexaoBD*;
- Conexão ao Banco de Dados, com verificação do sucesso ou não;
- Confeção da sentença SQL;
- Execução da sentença, com verificação do sucesso ou não.

A diferença é que em caso positivo, deve-se popular os dados dos objetos com resultado da consulta ao Banco de Dados, fechando, em seguida, a conexão e por fim retornando a *TRUE*.

Agora vamos ao último método da classe Usuário.

**Método atualizarBD**

Concluindo, o último método específico da classe Usuário: atualizarBD, terá como função atualizar os dados do usuário no Banco de Dados.

```
public function atualizarBD()
{
    require_once 'ConexaoBD.php';
    $con = new ConexaoBD();
    $conn = $con->conectar();
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sql = "UPDATE usuario SET nome = '". $this->nome."', cpf = '".
        $this->cpf."', dataNascimento = '". $this->dataNascimento."',
        email='". $this->email."' WHERE idusuario = '". $this->id. "'";
    if ($conn->query($sql) === TRUE) {
        $conn->close();
        return TRUE;
    }
    else {
        $conn->close();
        return FALSE;
    }
}
```

Seguindo o mesmo padrão:

- Inclusão da classe *ConexaoBD*;
- Instância do objeto da Classe *ConexaoBD*;
- Conexão ao Banco de Dados, com verificação do sucesso ou não;
- Confeção da sentença SQL;
- Execução da sentença com verificação do sucesso ou não.

A diferença é que em caso positivo, atualizam-se os dados do Banco com os dados importados no objeto instanciado, em seguida, fecha-se a conexão, retornando a TRUE. Pronto! Finalizamos a classe Usuário.

Na próxima aula daremos sequência ao desenvolvimento deste projeto.

Até lá! :)