

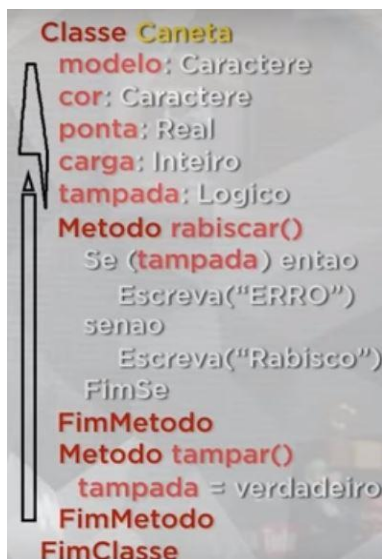
# Curso POO-PHP

## Curso em Vídeo (Gustavo Guanabara)

[https://www.youtube.com/watch?v=KIIL63MeyMY&list=PLHz\\_AreHm4dmGuLI3tsvryMMD7VgcT7x](https://www.youtube.com/watch?v=KIIL63MeyMY&list=PLHz_AreHm4dmGuLI3tsvryMMD7VgcT7x)

Resumo do curso feito por Roberto Pinheiro

### Aula 02 – Criando classes e objetos em PHP



#### Caneta.php

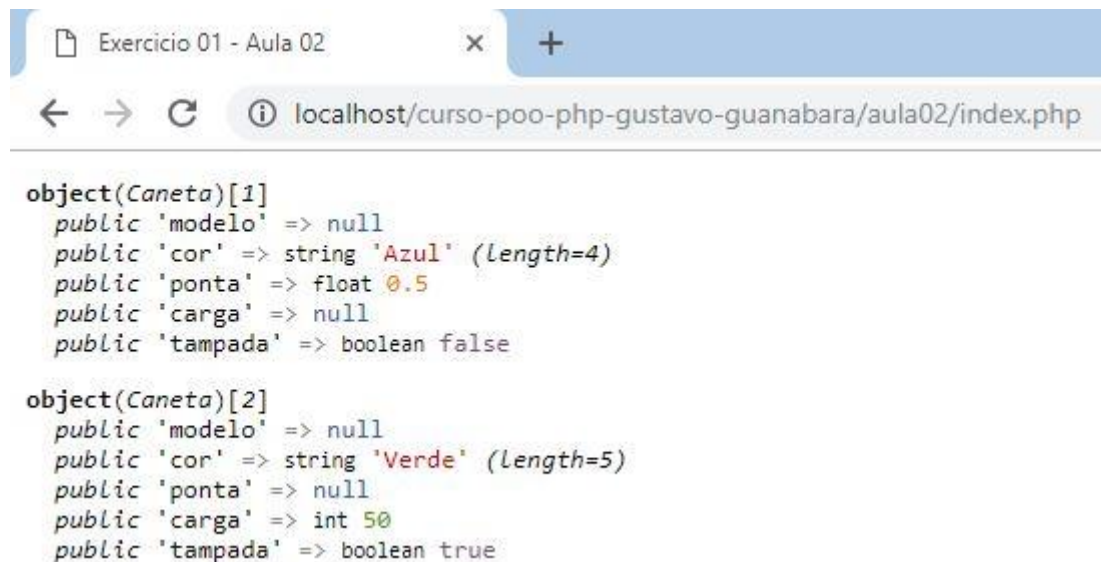
```
<?php
class Caneta {
    var $modelo;
    var $cor;
    var $ponta;
    var $carga;
    var $tampada;
    function rabiscar() {
        if ($this->tampada == true) {
            echo "<p>Erro! Não posso rabiscar.</p>";
        } else {
            echo "<p>Estou Rabiscando...</p>";
        }
    }
}
function tampar() {
    $this->tampada = true;
}
function destampar() {
    $this->tampada = false;
}
}
?>
```

## index.php

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Exercicio 01 - Aula 02</title>
  </head>
  <body>
    <?php
      require_once 'Caneta.php';
      $c1 = new Caneta;
      $c1->cor = "Azul";
      $c1->ponta = 0.5;
      $c1->destampar();

      $c2 = new Caneta;
      $c2->cor = "Verde";
      $c2->carga = 50;
      $c2->tampar();

      var_dump($c1);
      var_dump($c2);
    ?>
  </body>
</html>
```



## Aula 03 – Configurando visibilidade de atributos e métodos

**Classe Caneta**

**publico** modelo: Caractere  
**publico** cor: Caractere  
**privado** ponta: Real  
**protegido** carga: Inteiro  
**protegido** tampada: Logico  
**publico** Metodo escrever()  
  
**FimMetodo**  
**privado** Metodo rabiscar()  
  
**FimMetodo**  
**FimClasse**

Caneta
+ modelo + cor - ponta # carga # tampada
+ escrever() + rabiscar() + pintar() - tampar() - destampar()

### Caneta.php

```
<?php
class Caneta {
    public $modelo;
    public $cor;
    private $ponta;
    protected $carga;
    protected $tampada;

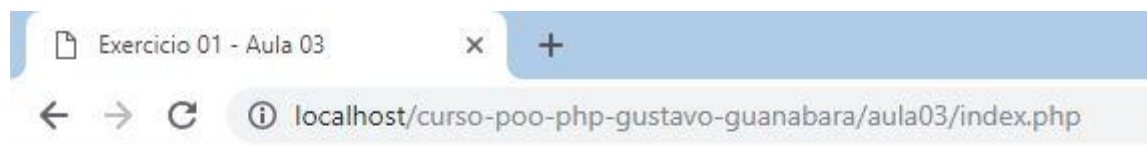
    public function rabiscar() {
        if ($this->tampada == true) {
            echo "<p>Erro! Não posso rabiscar.</p>";
        } else {
            echo "<p>Estou Rabiscando...</p>";
        }
    }
    public function tampar() {
        $this->tampada = true;
    }
    public function destampar() {
        $this->tampada = false;
    }
}

?>
```

## index.php

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Exercicio 01 - Aula 03</title>
  </head>
  <body>
    <?php
      require_once 'Caneta.php';
      $c1 = new Caneta;
      $c1->modelo = "BIC Crystal";
      $c1->cor = "Azul";
      //$c1->ponta = 0.5;
      //$c1->carga = 99;
      //$c1->tampada = true;
      $c1->rabiscar();
      $c1->tampar();
      $c1->destampar();

      var_dump($c1);
    ?>
  </body>
</html>
```



Estou Rabiscando...

```
object(Caneta)[1]
  public 'modelo' => string 'BIC Crystal' (length=11)
  public 'cor' => string 'Azul' (length=4)
  private 'ponta' => null
  protected 'carga' => null
  protected 'tampada' => boolean false
```

## Aula 04 – Métodos Getter, Setter e Construtor

Caneta
+ modelo - ponta
+ getModelo() + setModelo(m) + getPonta() + setPonta(p)

### Caneta.php

```
<?php
class Caneta {
    private $modelo;
    private $cor;
    private $ponta;
    private $tampada;

    public function __construct ($m, $c, $p) {
        $this->modelo = $m;
        $this->cor = $c;
        $this->ponta = $p;
        $this->tampar();
    }
    public function tampar() {
        $this->tampada = true;
    }
    public function getModelo() {
        return $this->modelo;
    }
    public function setModelo($m) {
        $this->modelo = $m;
    }
    public function getPonta() {
        return $this->ponta;
    }
    public function setPonta($p) {
        $this->ponta = $p;
    }
}
?>
```

## index.php

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Exercicio 01 - Aula 04</title>
  </head>
  <body>
    <?php
      require_once 'Caneta.php';

      $c1 = new Caneta ("BIC", "Azul", 0.5);
      $c2 = new Caneta ("KKK", "Verde", 1.0);

      var_dump($c1);
      var_dump($c2);
    ?>
  </body>
</html>
```



```
object(Caneta) [1]
  private 'modelo' => string 'BIC' (length=3)
  private 'cor' => string 'Azul' (length=4)
  private 'ponta' => float 0.5
  private 'tampada' => boolean true

object(Caneta) [2]
  private 'modelo' => string 'KKK' (length=3)
  private 'cor' => string 'Verde' (length=5)
  private 'ponta' => float 1
  private 'tampada' => boolean true
```

## Aula 05 – Exemplo prático em PHP

### Conta.php

```
<?php
/*
    Define a conta do banco
*/
class ContaBanco {
    //Atributos
    public $numConta;
    protected $tipo;
    private $dono;
    private $saldo;
    private $status;

    //Construtor
    function __construct() {
        $this->setSaldo(0);
        $this->setStatus(false);
    }

    //Metodos
    public function abrirConta($t) {
        $this->setTipo($t);
        $this->setStatus(true);
        if ($t == 'CC') {
            $this->setSaldo(50);
        } elseif ($t == 'CP') {
            $this->setSaldo(150);
        }
    }
    public function fecharConta() {
        if ($this->getSaldo() > 0) {
            echo "Conta com dinheiro";
        } elseif ($this->getSaldo() < 0) {
            echo "A conta está em debito";
        } else {
            $this->setStatus(false);
        }
    }
    public function depositar($val) {
        if ($this->getStatus()) {
            $this->setSaldo($this->getSaldo() + $val);
            echo "<p>Deposito de R$ $val na conta de " . $this->getDono() . "</p>";
        } else {
            echo "A conta está fechada";
        }
    }
}
```

```

public function sacar($val) {
    if ($this->getStatus()) {
        if ($this->getSaldo() >= $val) {
            $this->setSaldo($this->getSaldo() - $val);
            echo "<p>Saque de R$ $val autorizado na conta de " . $this->getDono() . "</p>";
        }
        elseif ($this->getSaldo() < $val) {
            echo "Saldo insuficiente";
        } else {
            echo "A Conta está fechada";
        }
    }
}

public function pagarMensal() {
    $v;
    if ($this->getTipo() == 'CC') {
        $v = 12;
    } elseif ($this->getTipo() == 'CP') {
        $v = 20;
    }
    if ($this->getStatus()) {
        if ($this->getSaldo() > $v) {
            $this->setSaldo($this->getSaldo() - $v);
        } elseif ($this->getSaldo() < $v) {
            echo "Saldo insuficiente";
        } else {
            echo "Conta fechada";
        }
    }
}

```

//Setters e Getters

```

public function setNumConta($n) {
    $this->numConta= $n;
}

public function getNumConta() {
    return $this->numConta;
}

public function setTipo($t) {
    $this->tipo = $t;
}

public function getTipo() {
    return $this->tipo;
}

public function setDono($d) {
    $this->dono = $d;
}

public function getDono() {
    return $this->dono;
}

public function setSaldo($s) {
    $this->saldo = $s;
}

public function getSaldo() {
    return $this->saldo;
}

```



```

    }
    public function setStatus($s) {
        $this->status = $s;
    }
    public function getStatus() {
        return $this->status;
    }
}
?>

```

### Arquivo index.php

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Exercicio 01 - Aula 05</title>
    </head>
    <body>
        <?php
            require_once 'Conta.php';

            $contaJubileu = new ContaBanco();
            $contaCreuza = new ContaBanco();
            $contaJubileu->abrirConta("CC");
            $contaJubileu->setDono("Jubileu");
            $contaJubileu->setNumConta(1111);
            $contaJubileu->depositar(300);
            $contaJubileu->pagarMensal();
            $contaJubileu->sacar(338);
            $contaJubileu->fecharConta();

            $contaCreuza->abrirConta("CP");
            $contaCreuza->setDono("Creuza");
            $contaCreuza->setNumConta(2222);
            $contaCreuza->depositar(500);
            $contaCreuza->pagarMensal();
            $contaCreuza->sacar(630);
            $contaCreuza->fecharConta();

            var_dump($contaJubileu);
            var_dump($contaCreuza);
        ?>
    </body>
</html>

```



Deposito de R\$ 300 na conta de Jubileu

Saque de R\$ 338 autorizado na conta de Jubileu

Deposito de R\$ 500 na conta de Creuza

Saque de R\$ 630 autorizado na conta de Creuza

```
object(ContaBanco)[1]
  public 'numConta' => int 1111
  protected 'tipo' => string 'CC' (length=2)
  private 'dono' => string 'Jubileu' (length=7)
  private 'saldo' => int 0
  private 'status' => boolean false
```

```
object(ContaBanco)[2]
  public 'numConta' => int 2222
  protected 'tipo' => string 'CP' (length=2)
  private 'dono' => string 'Creuza' (length=6)
  private 'saldo' => int 0
  private 'status' => boolean false
```

## Aula 06 – Encapsulamento

### ControleRemoto.php

```
<?php
/*
Classe para o modelo do controle remoto
*/
require_once 'Controlador.php';
class ControleRemoto implements Controlador {
    //Atributos da classe
    private $volume;
    private $ligado;
    private $tocando;

    public function __construct() {
        $this->volume = 50;
        $this->ligado = false;
        $this->tocando = false;
    }
    private function getVolume() {
        return $this->volume;
    }
    private function getLigado() {
        return $this->ligado;
    }
    private function getTocando() {
        return $this->tocando;
    }
    private function setVolume($volume) {
        $this->volume = $volume;
    }
    private function setLigado($ligado) {
        $this->ligado = $ligado;
    }
    private function setTocando($tocando) {
        $this->tocando = $tocando;
    }
    public function ligar() {
        $this->setLigado(true);
    }
    public function desligar() {
        $this->setLigado(false);
    }
    public function abrirMenu() {
        echo "<br />Está ligado? " . ($this->getLigado()? "SIM":"NÃO");
        echo "<br />Está tocando? " . ($this->getTocando()? "SIM":"NÃO");
        echo "<br />Volume: " . $this->getVolume() . " ";
        for($i=0; $i <= $this->getVolume(); $i+=10) {
```

```

        Echo "|";
    }
    echo "<br />";
}
public function fecharMenu() {
    echo "<br /> Fechando menu...";
}
public function maisVolume () {
    if ($this->getLigado()) {
        $this->setVolume($this->getVolume() + 10);
    } else {
        echo "Erro não posso aumentar o volume";
    }
}
public function menosVolume () {
    if ($this->getLigado()) {
        $this->setVolume($this->getVolume() - 10);
    } else {
        echo "Erro não posso diminuir o volume";
    }
}
public function ligarMudo() {
    if ($this->getLigado() && $this->getVolume > 0) {
        $this->setVolume(0);
    }
}
public function desligarMudo() {
    if ($this->getLigado() && $this->getVolume == 0) {
        $this->setVolume(50);
    }
}
public function play () {
    if ($this->getLigado() && !($this->getTocando()))
        $this->setTocando(true);
}
public function pause() {
    if ($this->getLigado() && $this->getTocando()) {
        $this->setTocando(false);
    }
}
}
?>

```

## Controlador.php

```
<?php
//Interface do controlador

interface Controlador {
    public function ligar();
    public function desligar();
    public function abrirMenu();
    public function fecharMenu();
    public function maisVolume();
    public function menosVolume();
    public function ligarMudo();
    public function desligarMudo();
    public function play();
    public function pause();
}
?>
```

## index.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Exercicio 01 - Aula 06</title>
</head>
<body>
    <pre>
        <?php
            echo "<h1>Projeto Controle Remoto</h1>";
            require_once 'ControleRemoto.php';
            $c = new ControleRemoto();

            $c->ligar();
            $c->maisVolume();
            $c->abrirMenu();
        ?>
    </pre>
</body>
</html>
```



## Projeto Controle Remoto

Está ligado? SIM  
Está tocando? NÃO  
Volume: 60 |||||

## Aula 07 – Objetos compostos em PHP

### Lutador.php

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Exercicio 01 - Aula 06</title>
  </head>
  <body>
    <pre>
      <?php
        echo "<h1>Projeto Controle Remoto</h1>";
        require_once 'ControleRemoto.php';
        $c = new ControleRemoto();

        $c->ligar();
        $c->maisVolume();
        $c->abrirMenu();
      ?>
    </pre>
  </body>
</html>
```

## index.php

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ultra Emoji Combat</title>
  </head>
  <body>
    <?php
      require_once 'Lutador.php';
      /* Para criar um novo lutador lembre-se do método construtor.
      Ele se reproduz nessa ordem
      nome, nacionalidade, idade, altura, peso, categoria, vitórias, derrotas, empates.
      */
      $lutador = array();

      $lutador[0] = new Lutador ('Pretty Boy', 'França', 31, 1.75, 68.9 , 11, 2, 1);
      $lutador[1] = new Lutador ('Putscript', 'Brasil', 29, 1.68, 57.8 , 14, 2, 3);
      $lutador[2] = new Lutador ('Snapshadow', 'EUA', 35, 1.65, 80.9 , 12, 2, 1);
      $lutador[3] = new Lutador ('Dead Code', 'Austrália', 28, 1.93, 81.6 , 13, 0, 2);
      $lutador[4] = new Lutador ('Ufocobol', 'Brasil', 37, 1.70, 119.3 , 5, 4, 3);
      $lutador[5] = new Lutador ('Nerdaard', 'EUA', 30, 1.81, 105.7 , 12, 2, 4);

      $lutador[0]->apresentar();
      $lutador[2]->status();
      $lutador[1]->ganharLuta();
      $lutador[0]->empatarLuta();

      var_dump($lutador[0]);
      var_dump($lutador[1]);
      var_dump($lutador[2]);
      var_dump($lutador[3]);
      var_dump($lutador[4]);
      var_dump($lutador[5]);

    ?>
  </body>
</html>
```



O lutador Pretty Boy é do(a) França tem 31 de idade, altura de 1.75m, pesa 68.9kg lutando pela categoria Leve

O Lutador Snapshadow tem 12 Vitórias, 2 Derrotas e 1 Empates.

O lutador Putscript ganhou uma luta e agora possui 14 vitórias.

O lutador Pretty Boy empatou uma luta e agora possui 1 empates.

```
object(Lutador) [1]
  private 'nome' => string 'Pretty Boy' (length=10)
  private 'nacionalidade' => string 'França' (length=7)
  private 'idade' => int 31
  private 'altura' => float 1.75
  private 'peso' => float 68.9
  private 'categoria' => string 'Leve' (length=4)
  private 'vitorias' => int 11
  private 'derrotas' => int 2
  private 'empates' => int 1
  public 'setEmpates' => int 2

object(Lutador) [2]
  private 'nome' => string 'Putscript' (length=9)
  private 'nacionalidade' => string 'Brasil' (length=6)
  private 'idade' => int 29
  private 'altura' => float 1.68
  private 'peso' => float 57.8
  private 'categoria' => string 'Leve' (length=4)
  private 'vitorias' => int 14
  private 'derrotas' => int 2
  private 'empates' => int 3
  public 'setVitorias' => int 15

object(Lutador) [3]
  private 'nome' => string 'Snapshadow' (length=10)
  private 'nacionalidade' => string 'EUA' (length=3)
  private 'idade' => int 35
  private 'altura' => float 1.65
  private 'peso' => float 80.9
  private 'categoria' => string 'Médio' (length=6)
  private 'vitorias' => int 12
  private 'derrotas' => int 2
  private 'empates' => int 1

object(Lutador) [4]
  private 'nome' => string 'Dead Code' (length=9)
  private 'nacionalidade' => string 'Austrália' (length=10)
  private 'idade' => int 28
  private 'altura' => float 1.93
  private 'peso' => float 81.6
  private 'categoria' => string 'Médio' (length=6)
  private 'vitorias' => int 13
  private 'derrotas' => int 0
  private 'empates' => int 2

object(Lutador) [5]
  private 'nome' => string 'Ufocobol' (length=8)
  private 'nacionalidade' => string 'Brasil' (length=6)
  private 'idade' => int 37
  private 'altura' => float 1.7
  private 'peso' => float 119.3
  private 'categoria' => string 'Pesado' (length=6)
  private 'vitorias' => int 5
  private 'derrotas' => int 4
  private 'empates' => int 3

object(Lutador) [6]
  private 'nome' => string 'Nerdaard' (length=8)
  private 'nacionalidade' => string 'EUA' (length=3)
  private 'idade' => int 30
  private 'altura' => float 1.81
  private 'peso' => float 105.7
  private 'categoria' => string 'Pesado' (length=6)
  private 'vitorias' => int 12
  private 'derrotas' => int 2
  private 'empates' => int 4
```

## Aula 08 – Agregação entre objetos em PHP

### Lutador.php

```
<?php
```

```
class Lutador {
    //Atributos
    private $nome;
    private $nacionalidade;
    private $idade;
    private $altura;
    private $peso;
    private $categoria; #Peso leve, Peso Médio e Peso Pesado
    private $vitorias;
    private $derrotas;
    private $empates;

    // Método construtor
    public function __construct($nome, $nacionalidade, $idade, $altura, $pe, $vitorias,
    $derrotas, $empates) {
        $this->nome = $nome;
        $this->nacionalidade = $nacionalidade;
        $this->idade = $idade;
        $this->altura = $altura;
        $this->setPeso($pe);
        $this->vitorias = $vitorias;
        $this->derrotas = $derrotas;
        $this->empates = $empates;
    }
    //Getter e Setter [Métodos especiais]
    function getNome() {
        return $this->nome;
    }
    function getNacionalidade() {
        return $this->nacionalidade;
    }
    function getIdade() {
        return $this->idade;
    }
    function getAltura() {
        return $this->altura;
    }
    function getPeso() {
        return $this->peso;
    }
    function getCategoria() {
        return $this->categoria;
    }
}
```

```

function getVitorias() {
    return $this->vitorias;
}
function getDerrotas() {
    return $this->derrotas;
}
function getEmpates() {
    return $this->empates;
}
function setNome($nome) {
    $this->nome = $nome;
}
function setNacionalidade($nacionalidade) {
    $this->nacionalidade = $nacionalidade;
}
function setIdade($idade) {
    $this->idade = $idade;
}
function setAltura($altura) {
    $this->altura = $altura;
}
function setPeso($pe) {
    $this->peso = $pe;
    $this->setCategoria();
}
function setCategoria() {
    if ($this->peso < 52.2) {
        $this->categoria = "Invalido";
    } elseif ($this->peso <= 70.3) {
        $this->categoria = "Leve";
    } elseif ($this->peso <= 83.9) {
        $this->categoria = "Médio";
    } elseif ($this->peso <= 120.2) {
        $this->categoria = "Pesado";
    } else {
        $this->categoria = "Invalido";
    }
}
function setVitorias($vit) {
    $this->vitorias = $this->getVitorias() + $vit;
}
function setDerrotas($der) {
    $this->derrotas = $this->getDerrotas() + $der;
}
function setEmpates($emp) {
    $this->empates = $this->getEmpates + $emp;
}

```

//Métodos

```

public function apresentar() {

```

```

        echo "<br />O lutador " . $this->getNome() . " é do(a) " . $this->getNacionalidade()
        . " tem " . $this->getIdade() . " de idade, altura de " . $this->getAltura() . "m, pesa " .
        $this->getPeso() . "kg lutando pela categoria " . $this->getCategoria();
    }
    public function status() {
        echo "<br />O Lutador " . $this->getNome() . " tem " . $this->getVitorias() . "
        Vitorias, " . $this->getDerrotas() . " Derrotas e " . $this->getEmpates() . " Empates.";
    }
    public function ganharLuta() {
        $this->setVitorias = $this->getVitorias() + 1;
        echo "<br />O lutador " . $this->getNome() . " ganhou uma luta e agora possui " .
        $this->getVitorias() . " vitorias.";
    }
    public function perderLuta() {
        $this->setDerrotas = $this->getDerrotas() + 1;
        echo "<br />O lutador " . $this->getNome() . " perdeu uma luta e agora possui " .
        $this->getDerrotas() . " derrotas.";
    }
    public function empatarLuta() {
        $this->setEmpates = $this->getEmpates() + 1;
        echo "<br />O lutador " . $this->getNome() . " empatou uma luta e agora possui " .
        $this->getEmpates() . " empates.";
    }
}
?>

```

## Luta.php

```

<?php
require_once 'Lutador.php';
class Luta {
    #Atributos
    private $desafiado;
    private $desafiante;
    private $rounds;
    private $aprovada;

    #Métodos Especiais , GETTER E SETTER
    private function getDesafiado() {
        return $this->desafiado;
    }
    private function getDesafiante() {
        return $this->desafiante;
    }
    private function getRounds() {
        return $this->rounds;
    }
    private function getAprovada() {
        return $this->aprovada;
    }
}

```

```

    }
    private function setDesafiado($desafiado) {
        $this->desafiado = $desafiado;
    }
    private function setDesafiante($desafiante) {
        $this->desafiante = $desafiante;
    }
    private function setRounds($rounds) {
        $this->rounds = $rounds;
    }
    private function setAprovada($aprovada) {
        $this->aprovada = $aprovada;
    }
}

```

#Métodos padrões

```

    public function marcarLuta($lutador1, $lutador2) {
        if ($lutador1->getCategoria() == $lutador2->getCategoria() && $lutador1 !=
        $lutador2) {
            $this->aprovada = true;
            $this->desafiado = $lutador1;
            $this->desafiante = $lutador2;
        } else {
            $this->aprovada = false;
            $this->desafiado = null;
            $this->desafiante = null;
        }
    }
    public function lutar() {
        if ($this->aprovada === true) {
            $this->desafiado->apresentar();
            $this->desafiante->apresentar();
            $checkvit = rand(0, 2);
            switch ($checkvit) {
                case 0: //Empate
                    echo "<br />Empatou";
                    $this->desafiado->empatarLuta();
                    $this->desafiante->empatarLuta();
                    break;
                case 1: //Ganhou desafiado
                    echo "<br />" . $this->desafiado->getNome() . " Desafiado Ganhou";
                    $this->desafiado->ganharLuta();
                    $this->desafiante->perderLuta();
                    break;
                case 2: //Ganhou desafiante
                    echo "<br />" . $this->desafiante->getNome() . " Desafiante Venceu";
                    $this->desafiado->perderLuta();
                    $this->desafiante->ganharLuta();
                    break;
            }
        } else {

```

```

        echo "A luta não pode acontecer";
    }
}
?>

```

## index.php

```

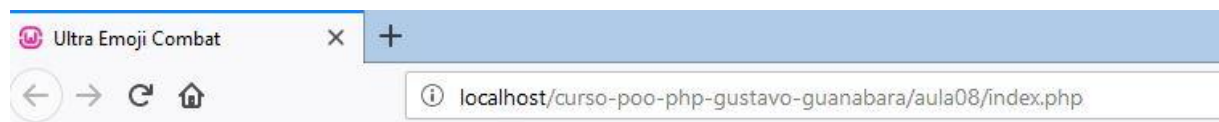
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Ultra Emoji Combat</title>
</head>
<body>
<?php
require_once 'Lutador.php';
require_once 'Luta.php';
/* Para criar um novo lutador lembre-se do método construtor.
Ele se reproduz nessa ordem
nome, nacionalidade, idade, altura, peso, categoria, vitórias, derrotas, empates.
*/
$lutador = array();

$lutador[0] = new Lutador ('Pretty Boy', 'França', 31, 1.75, 68.9 , 11, 2, 1);
$lutador[1] = new Lutador ('Putscript', 'Brasil', 29, 1.68, 57.8 , 14, 2, 3);
$lutador[2] = new Lutador ('Snapshadow', 'EUA', 35, 1.65, 80.9 , 12, 2, 1);
$lutador[3] = new Lutador ('Dead Code', 'Austrália', 28, 1.93, 81.6 , 13, 0, 2);
$lutador[4] = new Lutador ('Ufocobol', 'Brasil', 37, 1.70, 119.3 , 5, 4, 3);
$lutador[5] = new Lutador ('Nerdaard', 'EUA', 30, 1.81, 105.7 , 12, 2, 4);

$UEC01 = new Luta ();
$UEC01->marcarLuta($lutador[4], $lutador[5]);
$UEC01->lutar();

?>
</body>
</html>

```



O lutador Ufocobol é do(a) Brasil tem 37 de idade, altura de 1.7m, pesa 119.3kg lutando pela categoria Pesado  
O lutador Nerdaard é do(a) EUA tem 30 de idade, altura de 1.81m, pesa 105.7kg lutando pela categoria Pesado  
Empatou  
O lutador Ufocobol empatou uma luta e agora possui 3 empates.  
O lutador Nerdaard empatou uma luta e agora possui 4 empates.

## Aula 09 – Exercício prático POO em PHP



### Livro.php

```
<?php
require_once 'Pessoa.php';
require_once 'Publicacao.php';

class Livro implements Publicacao{
    private $titulo;
    private $autor;
    private $totPaginas;
    private $pagAtual;
    private $aberto;
    private $leitor;

    function __construct($ti, $au, $pag,$lei) {
        $this->titulo = $ti;
        $this->autor = $au;
        $this->totPaginas = $pag;
        $this->pagAtual = 1;
```

```

$this->aberto = false;
$this->leitor = $lei;
}
function getTitulo () {
    return $this->titulo;
}
function getAutor () {
    return $this->autor;
}
function getTotPaginas () {
    return $this->totPaginas;
}
function getPagAtual () {
    return $this->pagAtual;
}
function getAberto () {
    return $this->aberto;
}
function getLeitor () {
    return $this->leitor;
}
function setTitulo ($ti) {
    $this->titulo = $ti;
}
function setAutor ($au) {
    $this->autor = $au;
}
function setTotPaginas ($tot) {
    $this->totPaginas = $tot;
}
function setPagAtual ($pa) {
    $this->pagAtual = $pa;
}
function setAberto ($ab) {
    $this->aberto = $ab;
}
function setLeitor ($lei) {
    $this->leitor = $lei;
}
public function detalhes() {
    echo "Livro: " . $this->titulo . " escrito por " . $this->autor;
    echo "<br />P ginas: " . $this->totPaginas . " Atual: " . $this->getPagAtual();
    echo "<br />Sendo lido por " . $this->leitor->getNome();
}
public function abrir(){
    if ($this->getAberto() == true) {
        echo "J  est  aberto";
    } else {
        $this->setAberto(true);
    }
}

```



```

    }
    public function fechar(){
        if ($this->getAberto() == false) {
            echo "Já estou fechado";
        } else {
            $this->setAberto(false);
        }
    }
    public function folhear($p){
        if ($this->aberto == true && $p <= $this->getTotPaginas()) {
            $this->pagAtual = $p;
        } else {
            echo "O livro está fechado ou estou indo para uma página inacessível";
        }
    }
    public function avancarPag(){
        if ($this->aberto == false or $this->getPagAtual() == $this->getTotPaginas()) {
            echo "Não posso avançar estou fechado ou no fim do livro";
        } else {
            $this->setPagAtual($this->getPagAtual() + 1);
        }
    }
    public function voltarPag(){
        if ($this->aberto == false or $this->getPagAtual() == 1) {
            echo "Não posso voltar estou fechado ou na página inicial";
        } else {
            $this->setPagAtual($this->getPagAtual() - 1);
        }
    }
}

?>

```

## Pessoa.php

```

<?php

class Pessoa {
    private $nome;
    private $idade;
    private $sexo;

    function __construct ($no, $id, $se) {
        $this->nome = $no;
        $this->idade = $id;
        $this->sexo = $se;
    }
    function getNome() {
        return $this->nome;
    }
}

```

```

    }
    function getIdade() {
        return $this->idade;
    }
    function getSexo() {
        return $this->sexo;
    }
    function setNome($no) {
        $this->nome = $no;
    }
    function setIdade($id) {
        $this->idade = $id;
    }
    function setSexo($se) {
        $this->sexo = $se;
    }
}

public function fazerAniver() {
    $this->setIdade($this->getIdade() + 1);
    echo $this->getNome() . " fez aniversário agora tem " . $this->getIdade() . " anos
de idade. Parabéns.";
}
}
?>

```

### **Publicacao.php**

```

<?php
interface Publicacao {
    public function abrir();
    public function fechar();
    public function folhear($p);
    public function avancarPag();
    public function voltarPag();
}
?>

```

### **index.php**

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Exercicio 01 - Aula 09</title>
</head>
<body>
    <?php
        require_once 'Pessoa.php';
    >

```

```
require_once 'Livro.php';

$p[0] = new Pessoa ("Pedro", "22", "M");
$p[1] = new Pessoa ("Maria", 31, "F");

$l[0] = new Livro ("PHP básico", "José da Silva", 300, $p[0]);
$l[1] = new Livro ("POO com PHP", "Maria de Souza", 500, $p[0]);
$l[2] = new Livro ("PHP avançado", "Ana Paula", 800, $p[1]);

$l[0]->abrir();
$l[0]->folhear(80);
$l[0]->detalhes();
echo "<br />";
$l[1]->detalhes();
echo "<br />";
$l[2]->detalhes();

?>
</body>
</html>
```



## Aula 10 – Herança (parte 1)

### Pessoa.php

```
<?php
```

```
class Pessoa {  
    private $nome;  
    private $idade;  
    private $sexo;  
  
    function fazerAniv() {  
        $this->getIdade() + 1;  
    }  
    function getNome() {  
        return $this->nome;  
    }  
    function getIdade() {  
        return $this->idade;  
    }  
    function getSexo() {  
        return $this->sexo;  
    }  
}
```

```
function setNome($no) {  
    $this->nome = $no;  
}  
function setIdade($id) {  
    $this->idade = $id;  
}  
function setSexo($se) {  
    $this->sexo = $se;  
}  
}  
?>
```

## Funcionario.php

```
<?php
require_once 'Pessoa.php';
class Funcionario extends Pessoa {
    private $setor;
    private $trabalhando;
    function mudarTrabalho() {
        $this->setTrabalhando($this->getTrabalhando());
    }
    function getSetor() {
        return $this->setor;
    }
    function getTrabalhando() {
        return $this->trabalhando;
    }
    function setSetor($set) {
        $this->setor = $set;
    }
    function setTrabalhando($trab) {
        $this->trabalhando = ! $trab;
    }
}
?>
```

## Aluno.php

```
<?php
require_once 'Pessoa.php';
class Aluno extends Pessoa {
    private $matr;
    private $curso;
    function cancelarMatr() {
        echo "Matricula será cancelada";
    }
    function getMatr() {
        return $this->matr;
    }
    function getCurso() {
        return $this->curso;
    }
    function setMatr($ma) {
        $this->matr = $ma;
    }
    function setCurso($cu) {
        $this->curso = $cu;
    }
}
?>
```

## Professor.php

```
<?php
require_once 'Pessoa.php';
class Professor extends Pessoa {
    private $esp;
    private $sal;

    function receberAum($aum) {
        $this->setSal($this->getSal() + $aum);
    }
    function getEsp() {
        return $this->esp;
    }
    function getSal() {
        return $this->sal;
    }
    function setEsp($esp) {
        $this->esp = $esp;
    }
    function setSal($sal) {
        $this->sal = $sal;
    }
}
?>
```

## index.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Exercicio 01 - Aula 10</title>
</head>
<body>
    <?php
        require_once 'Pessoa.php';
        require_once 'Aluno.php';
        require_once 'Professor.php';
        require_once 'Funcionario.php';

        $p[0] = new Pessoa();
        $p[1] = new Aluno();
        $p[2] = new Professor();
        $p[3] = new Funcionario();

        $p[0]->setNome("Pedro");
        $p[1]->setNome("Maria");
        $p[2]->setNome("Cláudio");
```

```

    $p[3]->setNome("Fabiana");

    $p[0]->setSexo("M");
    $p[3]->setSexo("F");

    $p[1]->setCurso("Informática");
    $p[2]->setSal(2500.75);
    $p[3]->setSetor("Estoque");

    $p[2]->receberAum(550.20);
    $p[3]->mudarTrabalho();
    $p[1]->cancelarMatr();

    var_dump($p[0], $p[1], $p[2], $p[3]);
    ?>
</body>
</html>

```

```

Matricula será cancelada

object(Pessoa) [1]
  private 'nome' => string 'Pedro' (length=5)
  private 'idade' => null
  private 'sexo' => string 'M' (length=1)

object(Aluno) [2]
  private 'matr' => null
  private 'curso' => string 'Informática' (length=12)
  private 'nome' (Pessoa) => string 'Maria' (length=5)
  private 'idade' (Pessoa) => null
  private 'sexo' (Pessoa) => null

object(Professor) [3]
  private 'esp' => null
  private 'sal' => float 3050.95
  private 'nome' (Pessoa) => string 'Cláudio' (length=8)
  private 'idade' (Pessoa) => null
  private 'sexo' (Pessoa) => null

object(Funcionario) [4]
  private 'setor' => string 'Estoque' (length=7)
  private 'trabalhando' => boolean true
  private 'nome' (Pessoa) => string 'Fabiana' (length=7)
  private 'idade' (Pessoa) => null
  private 'sexo' (Pessoa) => string 'F' (length=1)

```



## Aula 11 – Herança (parte 2)

### Pessoa.php

```
<?php
abstract class Pessoa {
    private $nome;
    private $idade;
    private $sexo;

    public final function fazerAniversario(){
        $this->idade ++;
    }
    function getNome() {
        return $this->nome;
    }
    function getIdade() {
        return $this->Idade;
    }
    function getSexo() {
        return $this->sexo;
    }
    function setNome($n) {
        $this->nome = $n;
    }
    function setIdade($i) {
        $this->idade = $i;
    }
    function setSexo ($s) {
        $this->sexo = $s;
    }
}
?>
```

## Aluno.php

```
<?php
require_once 'Pessoa.php';
class Aluno extends Pessoa {
    private $matricula;
    private $curso;

    public function pagarMensalidade() {
        echo "<p>Pagando mensalidade do aluno: </p>" . $this->getNome();
    }
    function getMatricula() {
        return $this->matricula;
    }
    function getCurso() {
        return $this->curso;
    }
    function setMatricula($m) {
        $this->matricula = $m;
    }
    function setCurso ($c) {
        $this->curso = $c;
    }
}
?>
```

## Bolsista.php

```
<?php
require_once 'Aluno.php';
class Bolsista extends Aluno {
    private $bolsa;

    function renovarBolsa() {
        echo "Bolsa Renovada";
    }
    public function pagarMensalidade() {
        echo "<p>" . $this->getNome() . " é Bolsista! Então paga com desconto!</p>";
    }
    function getBolsa() {
        return $this->bolsa;
    }
    function setBolsa($b) {
        $this->bolsa = $b;
    }
}
?>
```

## Professor.php

```
<?php

require_once 'Pessoa.php';

class Professor extends Pessoa {

    private $especialidade;
    private $salario;

    public function receberAumento($valor) {
        $this->setSalario($this->getSalario() + $val);
    }

    function getEspecialidade() {
        return $this->especialidade;
    }

    function getSalario() {
        return $this->salario;
    }

    function setEspecialidade($e) {
        $this->especialidade = $e;
    }

    function setSalario($v) {
        $this->salario = $v;
    }
}
?>
```

## Visitante.php

```
<?php
require_once 'Pessoa.php';
class Visitante extends Pessoa {

}
?>
```

## Tecnico.php

```
<?php
require_once 'Aluno.php';
class Tecnico extends Aluno {
    private $registroProfissional;

    public function praticar() {
        echo "O aluno técnico " . $this->getNome() . " praticou a profissão";
    }
    function getRegistroProfissional() {
        return $this->registroProfissional;
    }
    function setRegistroProfissional($rp) {
        $this->registroProfissional = $rp;
    }
}
?>
```

## index.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Exercicio 01 - Aula 11</title>
</head>
<body>
    <?php
    /*require_once 'Visitante.php';
    $v1 = new Visitante();
    $v1->setNome("Juvenal");
    $v1->setIdade(33);
    $v1->setSexo("M");*/

    require_once 'Aluno.php';
    $a1 = new Aluno();
    $a1->setNome("Pedro");
    $a1->setMatricula(1111);
    $a1->setIdade(16);
    $a1->setSexo("M");
    $a1->setCurso("Informática");
    $a1->pagarMensalidade();
    var_dump($a1);

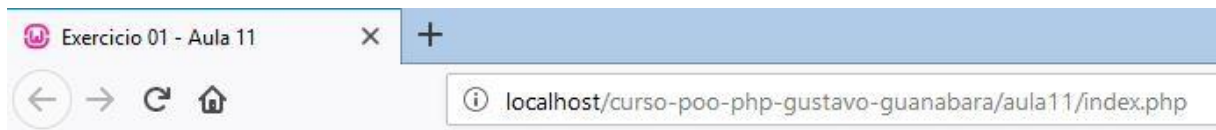
    require_once 'Bolsista.php';
    $b1 = new Bolsista();
    $b1->setMatricula(1112);
    $b1->setNome("Jubileu");
```

```

$b1->setBolsa(12.5);
$b1->setCurso("Adm");
$b1->setIdade(17);
$b1->pagarMensalidade();

var_dump($b1);
?>
</body>
</html>

```



Pagando mensalidade do aluno:

Pedro

```

object (Aluno) [1]
  private 'matricula' => int 1111
  private 'curso' => string 'Informática' (length=12)
  private 'nome' (Pessoa) => string 'Pedro' (length=5)
  private 'idade' (Pessoa) => int 16
  private 'sexo' (Pessoa) => string 'M' (length=1)

```

Jubileu é Bolsista! Então paga com desconto!

```

object (Bolsista) [2]
  private 'bolsa' => float 12.5
  private 'matricula' (Aluno) => int 1112
  private 'curso' (Aluno) => string 'Adm' (length=3)
  private 'nome' (Pessoa) => string 'Jubileu' (length=7)
  private 'idade' (Pessoa) => int 17
  private 'sexo' (Pessoa) => null

```

## Aula 12 – Polimorfismo em PHP

### Animal.php

```
<?php
```

```
abstract class Animal {
    protected $peso;
    protected $idade;
    protected $membros;

    abstract function locomover();
    abstract function alimentar();
    abstract function emitirSom();

    function getPeso() {
        return $this->peso;
    }
    function getIdade() {
        return $this->idade;
    }
    function getMembros() {
        return $this->membros;
    }
    function setPeso($peso) {
        $this->peso = $peso;
    }
    function setIdade($idade) {
        $this->idade = $idade;
    }
    function setMembros($membros) {
        $this->membros = $membros;
    }
}
?>
```

## Ave.php

```
<?php
require_once 'Animal.php';

class Ave extends Animal {
    private $corPena;

    function fazerNinho(){
        echo "Construindo um ninho";
    }
    public function alimentar() {
        echo "Comendo Frutas";
    }
    public function emitirSom() {
        echo "Som de Ave";
    }
    public function locomover() {
        echo "Voando";
    }
    function getCorPena() {
        return $this->corPena;
    }
    function setCorPena($corPena) {
        $this->corPena = $corPena;
    }
}

?>
```

## Reptil.php

```
<?php
require_once 'Animal.php';
class Reptil extends Animal {
    private $corEscama;
    public function alimentar() {
        echo "Comendo Vegetais";
    }
    public function emitirSom() {
        echo "Som de Réptil";
    }
    public function locomover() {
        echo "Rastejando";
    }
    function getCorEscama() {
        return $this->corEscama;
    }
    function setCorEscama($corEscama) {
        $this->corEscama = $corEscama;
    }
}
?>
```

## Mamifero.php

```
<?php
require_once 'Animal.php';
class Mamifero extends Animal {
    private $corPelo;
    public function alimentar() {
        echo "Mamando";
    }
    public function emitirSom() {
        echo "Som de Mamifero";
    }
    public function locomover() {
        echo "Correndo";
    }
    function getCorPelo() {
        return $this->corPelo;
    }
    function setCorPelo($corPelo) {
        $this->corPelo = $corPelo;
    }
}
?>
```



## Peixe.php

```
<?php
require_once 'Animal.php';

class Peixe extends Animal {
    private $corEscama;

    public function soltarBolha() {
        echo "Soltou uma Bolha";
    }
    public function alimentar() {
        echo "Comendo substâncias";
    }
    public function emitirSom() {
        echo "Peixe não faz som";
    }
    public function locomover() {
        echo "Nadando";
    }
    function getCorEscama() {
        return $this->corEscama;
    }
    function setCorEscama($corEscama) {
        $this->corEscama = $corEscama;
    }
}
;?>
```

## Arara.php

```
<?php
require_once 'Ave.php';
class Arara extends Ave {

}
?>
```

## Cachorro.php

```
<?php
require_once 'Mamifero.php';
class Cachorro extends Mamifero {
    public function emitirSom() {
        echo " AU AU AU AU ";
    }
}
?>
```

### **Canguru.php**

```
<?php
require_once 'Mamifero.php';
class Canguru extends Mamifero {
    public function locomover() {
        echo "Saltando";
    }
}
?>
```

### **Cobra.php**

```
<?php
require_once 'Reptil.php';
class Cobra extends Reptil {

}
?>
```

### **Tartaruga.php**

```
<?php
require_once 'Reptil.php';
class Tartaruga extends Reptil {
    public function locomover() {
        echo "Andando Beeeem devagar";
    }
}
?>
```

### **Goldfish.php**

```
<?php
require_once 'Peixe.php';
class Goldfish extends Peixe {

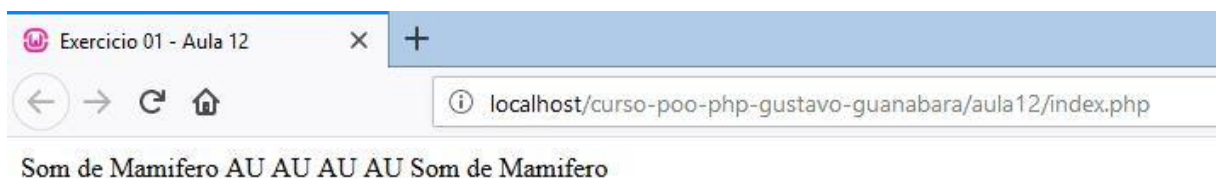
}
?>
```

## index.php

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Exercicio 01 - Aula 12</title>
  </head>
  <body>
    <?php
      require_once 'Mamifero.php';
      require_once 'Reptil.php';
      require_once 'Peixe.php';
      require_once 'Ave.php';
      require_once 'Canguru.php';
      require_once 'Cachorro.php';
      require_once 'Cobra.php';
      require_once 'Tartaruga.php';
      require_once 'Goldfish.php';
      require_once 'Arara.php';

      $m = new Mamifero();
      $a = new Ave();
      $r = new Reptil();
      $c = new Canguru();
      $k = new Cachorro();
      $t = new Tartaruga();

      $m->emitirSom();
      $k->emitirSom();
      $c->emitirSom();
    ?>
  </body>
</html>
```



## Aula 13 – Polimorfismo Sobrecarga (Parte 2)

### Animal.php

```
<?php

abstract class Animal {
    protected $peso, $idade, $membros;
    public abstract function emitirSom();
    function getPeso () {
        return $this->peso;
    }
    function getIdade() {
        return $this->idade;
    }
    function getMembros() {
        return $this->membros;
    }
    function setPeso($peso) {
        $this->peso = $peso;
    }
    function setIdade($idade) {
        $this->idade = $idade;
    }
    function setMembros($membros) {
        $this->membros = $membros;
    }
}
?>
```

### Mamifero.php

```
<?php
require_once 'Animal.php';
class Mamifero extends Animal {
    protected $corPelo;
    function emitirSom() {
        echo "Som de mamifero ";
    }
    function getCorPelo() {
        return $this->corPelo;
    }
    function setCorPelo($corPelo) {
        $this->corPelo = $corPelo;
    }
}
?>
```

## Lobo.php

```
<?php
require_once 'Mamifero.php';
class Lobo extends Mamifero {
    function emitirSom() {
        echo "Auuuuuuuuu!";
    }
}
?>
```

## Cachorro.php

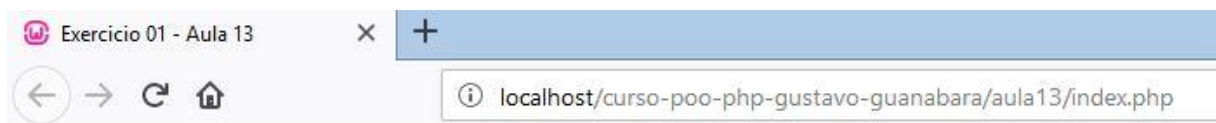
```
<?php
require_once 'Lobo.php';
class Cachorro extends Lobo {
    function emitirSom() {
        echo "Au! Au! Au!";
    }
    function reagirFrase($frase) {
        if ($frase == "toma comida" || $frase == "Olá") {
            echo "<br />Abanar e Latir ";
        } else {
            echo "<br />Rosnar ";
        }
    }
    function reagirHora($hora, $min) {
        if ($hora < 12) {
            echo "<br />Abanar ";
        } elseif ($hora >= 18) {
            echo "<br />Ignorar ";
        } else {
            echo "<br />Abanar e Latir ";
        }
    }
    function reagirDono($dono) {
        if ($dono == true) {
            echo "<br />Abanar ";
        } else {
            echo "<br />Rosnar e latir ";
        }
    }
    function reagirIdadePeso($idade, $peso) {
        if ($idade < 5) {
            if ($peso < 10) {
                echo "<br />Abanar";
            } else {
                echo "<br />Latir";
            }
        } else {
            if ($peso < 10) {
                echo "<br />Rosnar";
            } else {
                echo "<br />Ignorar";
            }
        }
    }
}
```

```
}  
}  
}  
}
```

?>

## index.php

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>Exercicio 01 - Aula 13</title>  
  </head>  
  <body>  
    <pre>  
    <?php  
    require_once 'Cachorro.php';  
    $c = new Cachorro();  
    $c->reagirFrase('Olá');  
    $c->reagirFrase('Vai apanhar');  
    $c->reagirHora(11, 45);  
    $c->reagirHora(21, 00);  
    $c->reagirDono(true);  
    $c->reagirDono(false);  
    $c->reagirIdadePeso(2, 12.5);  
    $c->reagirIdadePeso(17, 4.5);  
    var_dump($c);  
    ?>  
    </pre>  
  </body>  
</html>
```



```
Abanar e Latir  
Rosnar  
Abanar  
Ignorar  
Abanar  
Rosnar e latir  
Latir  
Rosnar  
  
object(Cachorro) [1]  
  protected 'corPelo' => null  
  protected 'peso' => null  
  protected 'idade' => null  
  protected 'membros' => null
```

## Aula 14 – Projeto final em PHP (parte 1)

AcoesVideo.php

```
<?php
interface AcoesVideo {
    public function play();
    public function pause();
    public function like();
}
?>
```

**Video.php**

```
<?php
require_once 'AcoesVideo.php';
class Video implements AcoesVideo {
    private $titulo;
    private $avaliacao;
    private $views;
    private $curtidas;
    private $reproduzindo;

    public function __construct($titulo) {
        $this->titulo = $titulo;
        $this->avaliacao = 1;
        $this->views = 0;
        $this->curtidas = 0;
        $this->reproduzindo = false;
    }

    public function play(){
        $this->reproduzindo = true;
    }
    public function pause(){
        $this->reproduzindo = false;
    }
    public function like(){
        $this->curtidas ++;
    }

    function getTitulo() {
        return $this->titulo;
    }
    function getAvaliacao() {
        return $this->avaliacao;
    }
    function getViews() {
        return $this->views;
    }
}
```

```

    }
    function getCurtidas() {
        return $this->curtidas;
    }
    function getReproduzindo() {
        return $this->reproduzindo;
    }
    function setTitulo($titulo) {
        $this->titulo = $titulo;
    }
    function setAvaliacao($avaliacao) {
        $this->avaliacao = $avaliacao;
    }
    function setViews($views) {
        $this->views = $views;
    }
    function setCurtidas($curtidas) {
        $this->curtidas = $curtidas;
    }
    function setReproduzindo($reproduzindo) {
        $this->reproduzindo = $reproduzindo;
    }
}
?>

```

## Pessoa.php

```
<?php
```

```

abstract class Pessoa {
    protected $nome;
    protected $idade;
    protected $sexo;
    protected $experiencia;

    public function __construct($nome, $idade, $sexo) {
        $this->nome = $nome;
        $this->idade = $idade;
        $this->sexo = $sexo;
        $this->experiencia = 0;
    }

    protected function ganharExp($n) {
        $this->experiencia += $n;
    }

    function getNome() {
        return $this->nome;
    }
    function getIdade() {

```



```

        return $this->idade;
    }
    function getSexo() {
        return $this->sexo;
    }
    function getExperiencia() {
        return $this->experiencia;
    }
    function setNome($nome) {
        $this->nome = $nome;
    }
    function setIdade($idade) {
        $this->idade = $idade;
    }
    function setSexo($sexo) {
        $this->sexo = $sexo;
    }
    function setExperiencia($experiencia) {
        $this->experiencia = $experiencia;
    }
}
?>

```

### **Gafanhoto.php**

```

<?php
require_once 'Pessoa.php';

class Gafanhoto extends Pessoa {
    private $login;
    private $totAssistido;

    public function __construct($nome, $idade, $sexo, $login) {
        parent::__construct($nome, $idade, $sexo);
        $this->login = $login;
        $this->totAssistido = 0;
    }
    function getLogin() {
        return $this->login;
    }
    function getTotAssistido() {
        return $this->totAssistido;
    }
    function setLogin($login) {
        $this->login = $login;
    }
    function setTotAssistido($totAssistido){
        $this->totAssistido = $totAssistido;
    }
    public function viuMaisUm() {

```

```
        $this->totAssistido ++;
    }
}
?>
```

## **index.php**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Exercicio 01 - Aula 14</title>
  </head>
  <body>
    <?php
      require_once 'Video.php';
      require_once 'Gafanhoto.php';
      $v[0] = new Video ("Aula 1 de POO");
      $v[1] = new Video ("Aula 12 de PHP");
      $v[2] = new Video ("Aula 15 de HTML5");
      $g[0] = new Gafanhoto("Tiago", 21, "M", "Kakou");
      $g[1] = new Gafanhoto("Creuza", 12, "F", "Creuzita");
    ?>
  </body>
</html>
```

## Aula 15 – Projeto final em PHP (parte 2)

### AcoesVideo.php

```
<?php
interface AcoesVideo {
    public function play();
    public function pause();
    public function like();
}
?>
```

### Video.php

```
<?php
require_once 'AcoesVideo.php';
class Video implements AcoesVideo {
    private $titulo;
    private $avaliacao;
    private $views;
    private $curtidas;
    private $reproduzindo;

    public function __construct($titulo) {
        $this->titulo = $titulo;
        $this->avaliacao = 1;
        $this->views = 0;
        $this->curtidas = 0;
        $this->reproduzindo = false;
    }

    public function play(){
        $this->reproduzindo = true;
    }
    public function pause(){
        $this->reproduzindo = false;
    }
    public function like(){
        $this->curtidas ++;
    }

    function getTitulo() {
        return $this->titulo;
    }
    function getAvaliacao() {
        return $this->avaliacao;
    }
    function getViews() {
        return $this->views;
    }
}
```

```

    }
    function getCurtidas() {
        return $this->curtidas;
    }
    function getReproduzindo() {
        return $this->reproduzindo;
    }
    function setTitulo($titulo) {
        $this->titulo = $titulo;
    }
    function setAvaliacao($avaliacao) {
        $media = ($this->avaliacao + $avaliacao) / $this->views;
        $this->avaliacao = $media;
    }
    function setViews($views) {
        $this->views = $views;
    }
    function setCurtidas($curtidas) {
        $this->curtidas = $curtidas;
    }
    function setReproduzindo($reproduzindo) {
        $this->reproduzindo = $reproduzindo;
    }
}
?>

```

## Pessoa.php

```
<?php
```

```

abstract class Pessoa {
    protected $nome;
    protected $idade;
    protected $sexo;
    protected $experencia;

    public function __construct($nome, $idade, $sexo) {
        $this->nome = $nome;
        $this->idade = $idade;
        $this->sexo = $sexo;
        $this->experencia = 0;
    }

    protected function ganharExp($n) {
        $this->experencia += $n;
    }

    function getNome() {
        return $this->nome;
    }
}

```

```

function getIdade() {
    return $this->idade;
}
function getSexo() {
    return $this->sexo;
}
function getExperiencia() {
    return $this->experiencia;
}
function setNome($nome) {
    $this->nome = $nome;
}
function setIdade($idade) {
    $this->idade = $idade;
}
function setSexo($sexo) {
    $this->sexo = $sexo;
}
function setExperiencia($experiencia) {
    $this->experiencia = $experiencia;
}
}
?>

```

## Gafanhoto.php

```

<?php
require_once 'Pessoa.php';

class Gafanhoto extends Pessoa {
    private $login;
    private $totAssistido;

    public function __construct($nome, $idade, $sexo, $login) {
        parent::__construct($nome, $idade, $sexo);
        $this->login = $login;
        $this->totAssistido = 0;
    }
    function getLogin() {
        return $this->login;
    }
    function getTotAssistido() {
        return $this->totAssistido;
    }
    function setLogin($login) {
        $this->login = $login;
    }
    function setTotAssistido($totAssistido){
        $this->totAssistido = $totAssistido;
    }
}

```

```

public function viuMaisUm() {
    $this->totAssistido ++;
}
}
?>

```

## Visualizacao.php

```

<?php
require_once 'Gafanhoto.php';
require_once 'Video.php';
class Visualizacao {
    private $espectador;
    private $filme;

    public function __construct($espectador, $filme) {
        $this->espectador = $espectador;
        $this->filme = $filme;
        $this->filme->setViews($this->filme->getViews() + 1);
        $this->espectador->setTotAssistido($this->espectador->getTotAssistido() + 1);
    }

    public function avaliar() {
        $this->filme->setAvaliacao(5);
    }
    public function avaliarNota($nota) {
        $this->filme->setAvaliacao($nota);
    }
    public function avaliarPorc($porc) {
        $nova = 0;
        if ($porc <= 20) {
            $nova= 3;
        } elseif ($porc <= 50) {
            $nova = 5;
        } elseif ($porc <= 90) {
            $nova = 8;
        } else {
            $nova = 10;
        }
        $this->filme->setAvaliacao($nova);
    }
    function getEspectador() {
        return $this->espectador;
    }
    function getFilme() {
        return $this->filme;
    }
    function setEspectador($espectador) {
        $this->espectador = $espectador;
    }
}

```

```

function setFilme($filme) {
    $this->filme = $filme;
}
}
?>

```

## index.php

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Exercicio 01 - Aula 14</title>
</head>
<body>
    <?php
        require_once 'Video.php';
        require_once 'Gafanhoto.php';
        require_once 'Visualizacao.php';
        $v[0] = new Video ("Aula 1 de POO");
        $v[1] = new Video ("Aula 12 de PHP");
        $v[2] = new Video ("Aula 15 de HTML5");
        $g[0] = new Gafanhoto("Tiago", 21, "M", "Kakou");
        $g[1] = new Gafanhoto("Creuza", 12, "F", "Creuzita");
        $vis[0] = new Visualizacao($g[0], $v[2]);
        $vis[1] = new Visualizacao($g[0], $v[1]);
        $vis[0]->avaliar();
        $vis[1]->avaliarPorc(85);

        var_dump($vis[0], $vis[1]);
    ?>
</body>
</html>

```

```
object(Visualizacao)[6]
  private 'espectador' =>
    object(Gafanhoto)[4]
      private 'login' => string 'Kakou' (length=5)
      private 'totAssistido' => int 2
      protected 'nome' => string 'Tiago' (length=5)
      protected 'idade' => int 21
      protected 'sexo' => string 'M' (length=1)
      protected 'experiencia' => int 0
  private 'filme' =>
    object(Video)[3]
      private 'titulo' => string 'Aula 15 de HTML5' (length=16)
      private 'avaliacao' => int 6
      private 'views' => int 1
      private 'curtidas' => int 0
      private 'reproduzindo' => boolean false

object(Visualizacao)[7]
  private 'espectador' =>
    object(Gafanhoto)[4]
      private 'login' => string 'Kakou' (length=5)
      private 'totAssistido' => int 2
      protected 'nome' => string 'Tiago' (length=5)
      protected 'idade' => int 21
      protected 'sexo' => string 'M' (length=1)
      protected 'experiencia' => int 0
  private 'filme' =>
    object(Video)[2]
      private 'titulo' => string 'Aula 12 de PHP' (length=14)
      private 'avaliacao' => int 9
      private 'views' => int 1
      private 'curtidas' => int 0
      private 'reproduzindo' => boolean false
```