

PHP Orientado a Objetos

Node Studio (Rodrigo Santos de Oliveira)

https://www.youtube.com/watch?v=hzy_P_H-1CQ&list=PLwXQLZ3FdTVEau55kNj_zLgpXL4JZUg8I

Resumo do curso feito por Roberto Pinheiro

Aula 2 - Classes, atributos e métodos

aula2.php

```
require_once("classes/aula2/Pessoa.php");

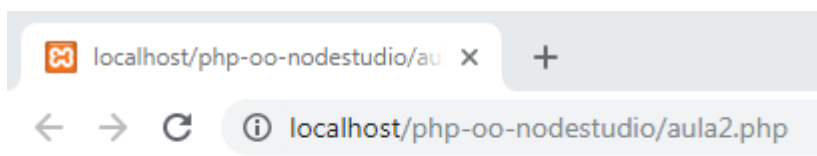
$rodrigo = new Pessoa();
$rodrigo->nome = "Rodrigo Santos de Oliveira";
$rodrigo->idade = 25;
$rodrigo->Falar();
// var_dump($rodrigo);
echo "<p>$rodrigo->nome</p>";
echo "<p>$rodrigo->idade</p>";
```

classes/aula2/Pessoa.php

```
<?php

class Pessoa {
    public $nome;
    public $idade;

    public function Falar(){
        echo "<p>". $this->nome . ", idade: ". $this->idade . " anos, acabou de
falar!</p>";
    }
}
```



Rodrigo Santos de Oliveira, idade: 25 anos, acabou de falar!

Rodrigo Santos de Oliveira

25

Aula 3 - Getters and Setters

aula3a.php

```
<?php

include_once 'classes/aula3a/Login.php';

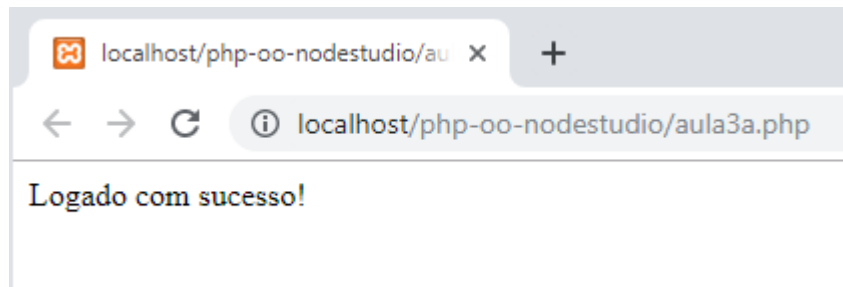
$logar = new Login();
$logar->email = "teste@teste.com";
$logar->senha = "123456";
$logar->Logar();
```

classes/aula3a/Login.php

```
<?php

class Login {
    public $email;
    public $senha;

    public function Logar(){
        if ($this->email == "teste@teste.com" and $this->senha == "123456"):
            echo "Logado com sucesso!";
        else:
            echo "Dados inválidos!";
        endif;
    }
}
```



aula3b.php

```
<?php

include_once 'classes/aula3b/Login.php';

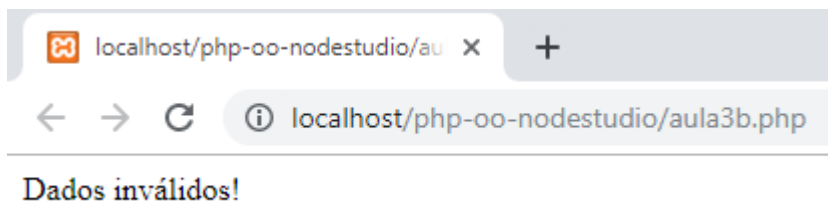
$login = new Login();
$login->email = "teste@teste.com";
$login->senha = "1234567";
$login->Logar();
```

classes/aula3b/Login.php

```
<?php

class Login {
    public $email;
    public $senha;

    public function Logar(){
        if ($this->email == "teste@teste.com" and $this->senha == "123456"):
            echo "Logado com sucesso!";
        else:
            echo "Dados inválidos!";
        endif;
    }
}
```



aula3c.php

```
<?php

include_once 'classes/aula3c/Login.php';

$login = new Login();
$login->setEmail("teste@teste.com");
$login->setSenha("123456");
$login->Logar();
echo "<p>Email: " . $login->getEmail() . "</p>";
echo "<p>Senha: " . $login->getSenha() . "</p>";
```

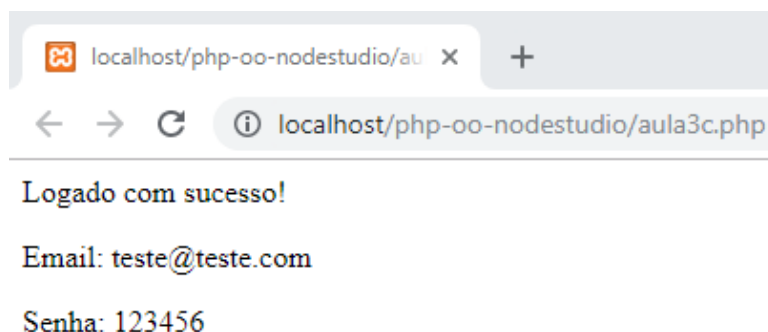
classes/aula3c/Login.php

```
<?php

class Login {
    private $email;
    private $senha;

    public function getEmail(){
        return $this->email;
    }
    public function setEmail($email){
        $email_limpo = filter_var($email, FILTER_SANITIZE_EMAIL);
        $this->email = $email_limpo;
    }
    public function getSenha(){
        return $this->senha;
    }
    public function setSenha($senha){
        $this->senha = $senha;
    }

    public function Logar(){
        if ($this->email == "teste@teste.com" and $this->senha == "123456"):
            echo "<p>Logado com sucesso!</p>";
        else:
            echo "<p>Dados inválidos!</p>";
        endif;
    }
}
```



aula3d.php

```
<?php

include_once 'classes/aula3d/Login.php';

$login = new Login();
$login->setEmail("teste()/@teste.com");
$login->setSenha("123456");
$login->Logar();
echo "<p>Email: " . $login->getEmail() . "</p>";
echo "<p>Senha: " . $login->getSenha() . "</p>";
```

classes/aula3d/Login.php

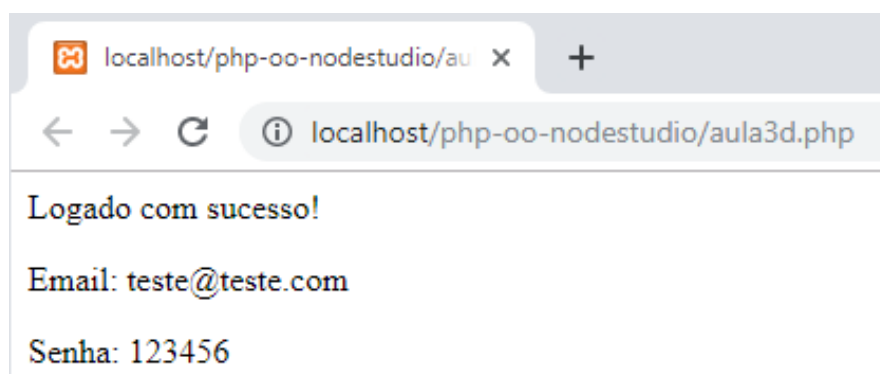
```
<?php

class Login {
    private $email;
    private $senha;

    public function getEmail(){
        return $this->email;
    }
    public function setEmail($email){
        $email_limpo = filter_var($email, FILTER_SANITIZE_EMAIL);
        $this->email = $email_limpo;
    }

    public function getSenha(){
        return $this->senha;
    }
    public function setSenha($senha){
        $this->senha = $senha;
    }

    public function Logar(){
        if ($this->email == "teste@teste.com" and $this->senha == "123456"):
            echo "<p>Logado com sucesso!</p>";
        else:
            echo "<p>Dados inválidos!</p>";
        endif;
    }
}
```



Aula 4 - Constructor

aula4.php

```
<?php

include_once 'classes/aula4/Login.php';

$logar = new Login("teste@teste.com", "123456", "Rodrigo Oliveira");
$logar->Logar();
echo "<p>Email: " . $logar->getEmail() . "</p>";
echo "<p>Senha: " . $logar->getSenha() . "</p>";
echo "<p>Nome: " . $logar->getNome() . "</p>";
```

classes/aula4/Login.php

```
<?php

class Login {
    private $email;
    private $senha;
    private $nome;

    public function __construct($email, $senha, $nome){
        $this->email = $email;
        $this->senha = $senha;
        $this->nome = $nome;
    }

    public function getEmail(){
        return $this->email;
    }

    public function setEmail($email){
        $this->email = $email;
    }

    public function getSenha(){
        return $this->senha;
    }

    public function setSenha($senha){
        $this->senha = $senha;
    }

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

    public function setNome($nome){
        $this->nome = $nome;
    }

    public function Logar(){
        if ($this->email == "teste@teste.com" and $this->senha == "123456"):
            echo "<p>Logado com sucesso!</p>";
        else:
            echo "<p>Dados inválidos!</p>";
        endif;
    }
}
```

localhost/php-oo-nodestudio/au x +

← → ↻ ⓘ localhost/php-oo-nodestudio/aula4.php

Logado com sucesso!

Email: teste@teste.com

Senha: 123456

Nome: Rodrigo Oliveira

Aula 5 - Herança

aula5.php

```
<?php

include_once 'classes/aula5/Carro.php';
include_once 'classes/aula5/Moto.php';

$carro = new Carro();
$carro->modelo = "Gol";
$carro->cor = "Vermelho";
$carro->ano = 2018;
$carro->Andar();
$carro->ligarLimpador();
print_r($carro);

$moto = new Moto();
$moto->modelo = "Honda Biz";
$moto->cor = "Azul";
$moto->ano = 2017;
$moto->Parar();
$moto->darGrau();
print_r($moto);
```

classes/aula5/Veiculo.php

```
<?php

class Veiculo {
    public $modelo;
    public $cor;
    public $ano;

    public function Andar(){
        echo "<p>Andou</p>";
    }

    public function Parar(){
        echo "<p>Parou</p>";
    }
}
```


classes/aula5/Carro.php

```
<?php

include_once 'classes/aula5/Veiculo.php';

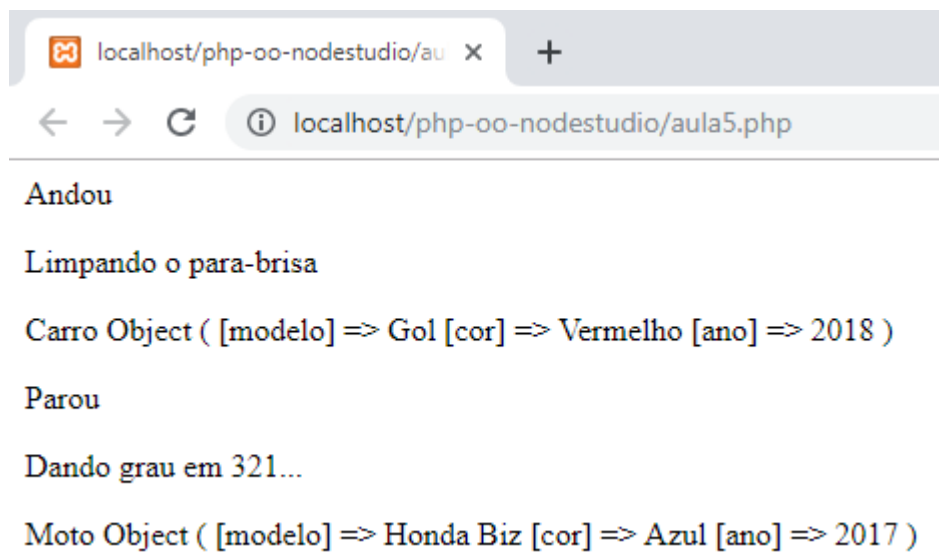
class Carro extends Veiculo {
    public function ligarLimpador(){
        echo "<p>Limpando o para-brisa</p>";
    }
}
```

classes/aula5/Moto.php

```
<?php

include_once 'classes/aula5/Veiculo.php';

class Moto extends Veiculo {
    public function darGrau(){
        echo "<p>Dando grau em 321...</p>";
    }
}
```



Aula 6 - Modificadores de acesso

aula6a.php

```
<?php

include_once 'classes/aula6a/Veiculo.php';

$veiculo = new Veiculo();
$veiculo->modelo = "Gol";
$veiculo->cor = "Vermelho";
$veiculo->ano = 2018;
echo "<p>Veiculo modelo: " . $veiculo->modelo . "</p>";
echo "<p>Veiculo cor: " . $veiculo->cor . "</p>";
echo "<p>Veiculo ano: " . $veiculo->ano . "</p>";
var_dump($veiculo);
```

classes/aula6a/Veiculo.php

```
<?php

class Veiculo {
    public $modelo;
    public $cor;
    public $ano;

    public function Andar(){
        echo "<p>Andou</p>";
    }

    public function Parar(){
        echo "<p>Parou</p>";
    }
}
```

classes/aula6a/Carro.php

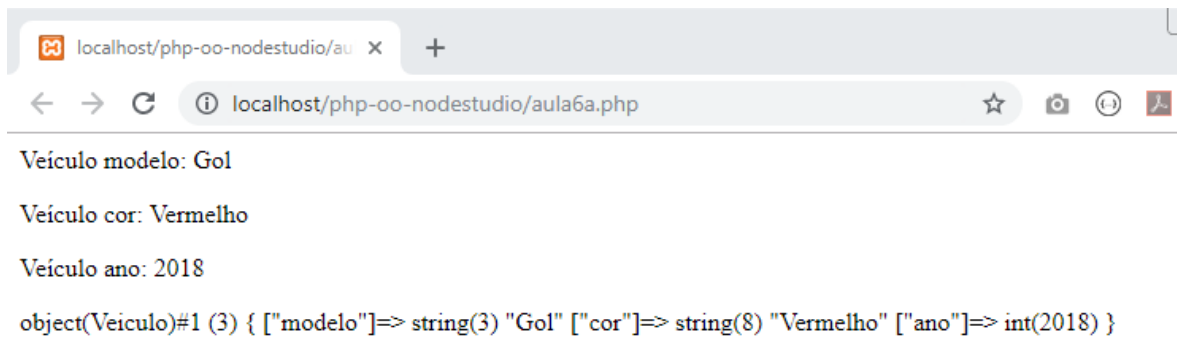
```
<?php

include_once 'classes/aula6a/Veiculo.php';

class Carro extends Veiculo {
    public function ligarLimpador(){
        echo "<p>Limpando o para-brisa</p>";
    }
}
```

classes/aula6a/Moto.php

```
<?php  
  
include_once 'classes/aula6a/Veiculo.php';  
  
class Moto extends Veiculo {  
    public function darGrau(){  
        echo "<p>Dando grau em 321...\</p>";  
    }  
}
```



aula6b.php

```
<?php

include_once 'classes/aula6b/Veiculo.php';

$veiculo = new Veiculo();
$veiculo->modelo = "Gol";
$veiculo->cor = "Vermelho";
$veiculo->ano = 2018;
echo "<p>Veículo modelo: " . $veiculo->modelo . "</p>";
echo "<p>Veículo cor: " . $veiculo->cor . "</p>";
echo "<p>Veículo ano: " . $veiculo->ano . "</p>";
var_dump($veiculo);
```

classes/aula6b/Veiculo.php

```
<?php

class Veiculo {
    private $modelo;
    public $cor;
    public $ano;

    public function Andar(){
        echo "<p>Andou</p>";
    }

    public function Parar(){
        echo "<p>Parou</p>";
    }
}
```

classes/aula6b/Carro.php

```
<?php

include_once 'classes/aula6b/Veiculo.php';

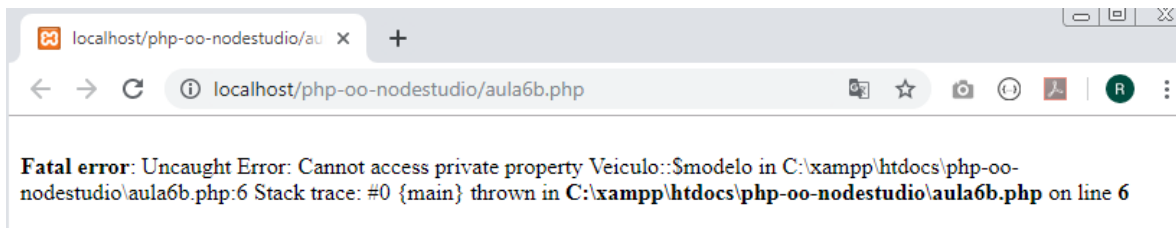
class Carro extends Veiculo {
    public function ligarLimpador(){
        echo "<p>Limpando o para-brisa</p>";
    }
}
```

classes/aula6b/Moto.php

```
<?php

include_once 'classes/aula6b/Veiculo.php';

class Moto extends Veiculo {
    public function darGrau(){
        echo "<p>Dando grau em 321...\</p>";
    }
}
```



aula6c.php

```
<?php

include_once 'classes/aula6c/Veiculo.php';

$veiculo = new Veiculo();
$veiculo->setModelo("Hilux");
echo "<p>Veículo modelo: " . $veiculo->getModelo() . "</p>";
var_dump($veiculo);
```

classes/aula6c/Veiculo.php

```
<?php

class Veiculo {
    private $modelo;
    public $cor;
    public $ano;

    public function getModelo(){
        return $this->modelo;
    }

    public function setModelo($modelo){
        $this->modelo = $modelo;
    }

    public function Andar(){
        echo "<p>Andou</p>";
    }

    public function Parar(){
        echo "<p>Parou</p>";
    }
}
```

classes/aula6c/Carro.php

```
<?php

include_once 'classes/aula6c/Veiculo.php';

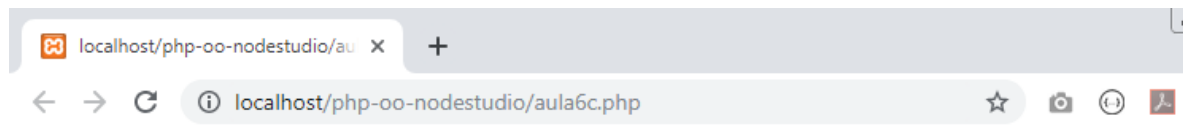
class Carro extends Veiculo {
    public function ligarLimpador(){
        echo "<p>Limpando o para-brisa</p>";
    }
}
```

classes/aula6c/Moto.php

```
<?php

include_once 'classes/aula6c/Veiculo.php';

class Moto extends Veiculo {
    public function darGrau(){
        echo "<p>Dando grau em 321...</p>";
    }
}
```



Veículo modelo: Hilux

```
object(Veiculo)#1 (3) { ["modelo":"Veiculo":private]=> string(5) "Hilux" ["cor"]=> NULL ["ano"]=> NULL }
```

aula6d.php

```
<?php

include_once 'classes/aula6d/Carro.php';

$carro = new Carro();
$carro->setModelo("Hilux");
echo "<p>Carro modelo: " . $carro->getModelo() . "</p>";
var_dump($carro);
```

classes/aula6d/Veiculo.php

```
<?php

class Veiculo {
    private $modelo;
    public $cor;
    public $ano;

    public function Andar(){
        echo "<p>Andou</p>";
    }

    public function Parar(){
        echo "<p>Parou</p>";
    }
}
```

classes/aula6d/Carro.php

```
<?php

include_once 'classes/aula6d/Veiculo.php';

class Carro extends Veiculo {

    public function getModelo(){
        return $this->modelo;
    }

    public function setModelo($modelo){
        $this->modelo = $modelo;
    }

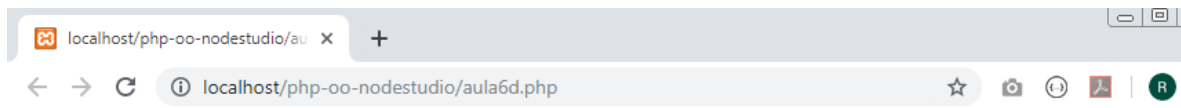
    public function ligarLimpador(){
        echo "<p>Limpando o para-brisa</p>";
    }
}
```


classes/aula6d/Moto.php

```
<?php

include_once 'classes/aula6d/Veiculo.php';

class Moto extends Veiculo {
    public function darGrau(){
        echo "<p>Dando grau em 321...\</p>";
    }
}
```



Carro modelo: Hilux

```
object(Carro)#1 (4) { ["modelo":"Veiculo":private]=> NULL ["cor"]=> NULL ["ano"]=> NULL ["modelo"]=> string(5) "Hilux" }
```

Aula 7 - Modificadores de acesso - 2

aula7a.php

```
<?php

include_once 'classes/aula7a/Veiculo.php';

$carro = new Veiculo();
$carro->Andar();
```

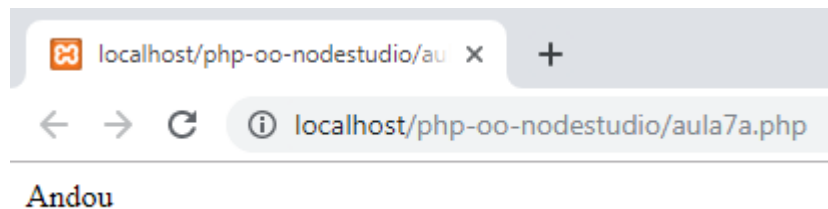
classes/aula7a/Veiculo.php

```
<?php

class Veiculo {
    public $modelo;
    public $cor;
    public $ano;

    public function Andar() {
        echo "<p>Andou</p>";
    }

    public function Parar() {
        echo "<p>Parou</p>";
    }
}
```



aula7b.php

```
<?php

include_once 'classes/aula7b/Veiculo.php';

$carro = new Veiculo();
$carro->Andar();
```

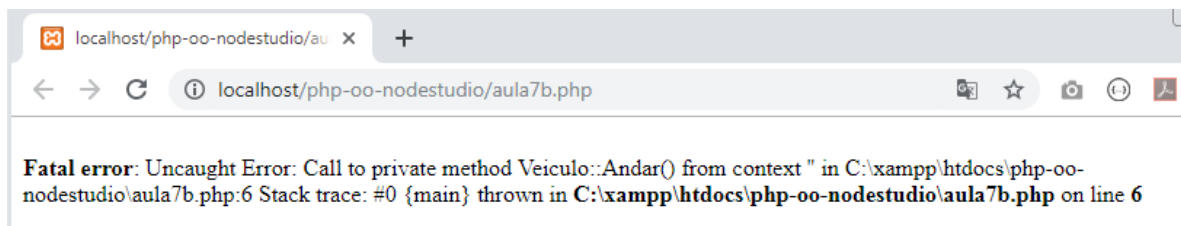
classes/aula7b/Veiculo.php

```
<?php

class Veiculo {
    private $modelo;
    public $cor;
    public $ano;

    private function Andar() {
        echo "<p>Andou</p>";
    }

    public function Parar() {
        echo "<p>Parou</p>";
    }
}
```



aula7c.php

```
<?php

include_once 'classes/aula7c/Veiculo.php';

$carro = new Veiculo();
$carro->mostrarAcao();
```

classes/aula7c/Veiculo.php

```
<?php

class Veiculo {
    private $modelo;
    public $cor;
    public $ano;

    private function Andar(){
        echo "<p>Andou</p>";
    }

    public function Parar(){
        echo "<p>Parou</p>";
    }

    public function mostrarAcao(){
        $this->Andar();
    }
}
```

classes/aula7c/Carro.php

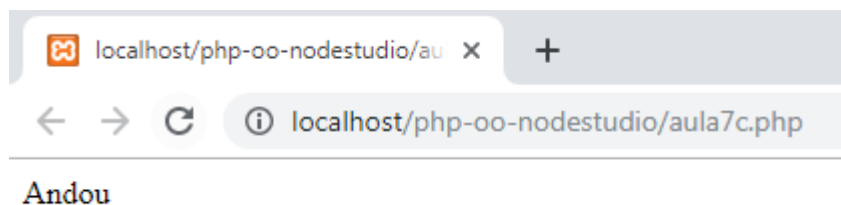
```
<?php

include_once 'classes/aula7c/Veiculo.php';

class Carro extends Veiculo {

    public function ligarLimpador(){
        echo "<p>Limpando o para-brisa</p>";
    }

}
```



aula7d.php

```
<?php

include_once 'classes/aula7d/Carro.php';

$carro = new Carro();
$carro->mostrarAcao();
```

classes/aula7d/Veiculo.php

```
<?php

class Veiculo {
    private $modelo;
    public $cor;
    public $ano;

    //    private function Andar(){
    //        echo "<p>Andou</p>";
    //    }

    protected function Andar(){
        echo "<p>Andou</p>";
    }

    public function Parar(){
        echo "<p>Parou</p>";
    }
}
```

classes/aula7d/Carro.php

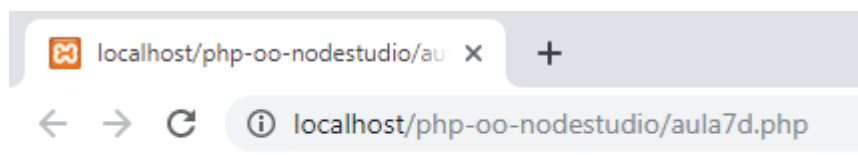
```
<?php

include_once 'classes/aula7d/Veiculo.php';

class Carro extends Veiculo {

    public function ligarLimpador(){
        echo "<p>Limpando o para-brisa</p>";
    }

    public function mostrarAcao(){
        $this->Andar();
    }
}
```



Andou

Aula 8 - Abstração

aula8.php

```
<?php

include_once 'classes/aula8/Banco.php';
include_once 'classes/aula8/Itau.php';

$itau = new Itau();
$itau->setSaldo(1000);
echo "Saldo: " . $itau->getSaldo() . "<hr>";
$itau->sacar(240);
echo "Saldo: " . $itau->getSaldo() . "<hr>";
$itau->depositar(100);
echo "Saldo: " . $itau->getSaldo() . "<hr>";
```

classes/aula8/Banco.php

```
<?php

abstract class Banco {
    protected $saldo;
    protected $limiteSaque;
    protected $juros;

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

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

    abstract protected function Sacar($valor);
    abstract protected function Depositar($valor);
}
```

classes/aula8/Itau.php

```
<?php
include_once 'classes/aula8/Banco.php';

class Itau extends Banco {
    public function Sacar($valor){
        $this->saldo -= $valor;
        echo "Sacou $valor <hr>";
    }
    public function Depositar($valor){
        $this->saldo += $valor;
        echo "Depositou $valor <hr>";
    }
}
```

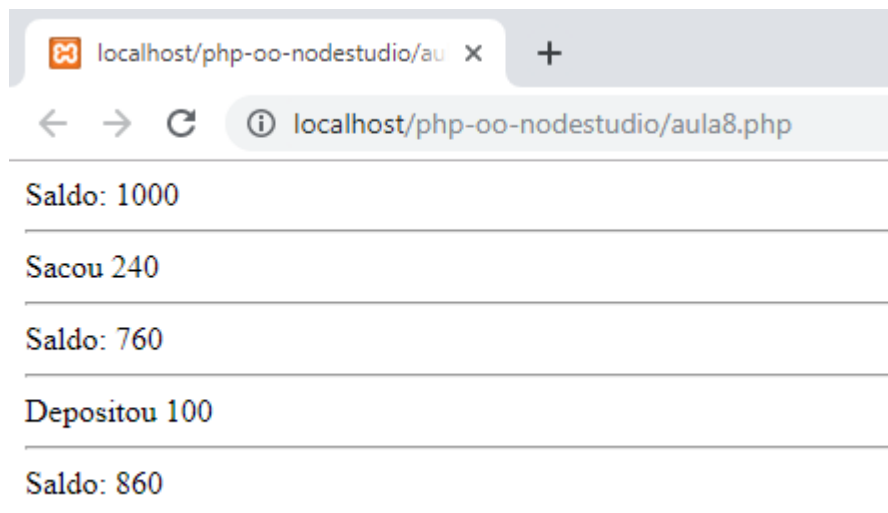
classes/aula8/Bradesco.php

```
<?php
include_once 'classes/aula8/Banco.php';

class Bradesco extends Banco {

    public function Sacar($valor){
        $this->saldo -= $valor;
        echo "Sacou $valor <hr>";
    }

    public function Depositar($valor){
        $this->saldo += $valor;
        echo "Depositou $valor <hr>";
    }
}
```



Aula 9 - Constantes, self e parent

aula9a.php

```
<?php

include_once 'classes/aula9a/Pessoa.php';

$ pessoa = new Pessoa();
$ pessoa->exibirNome();
```

classes/aula9a/Pessoa.php

```
<?php

class Pessoa {
    const nome = "Rodrigo";

    public function exibirNome(){
        echo self::nome;
    }
}
```

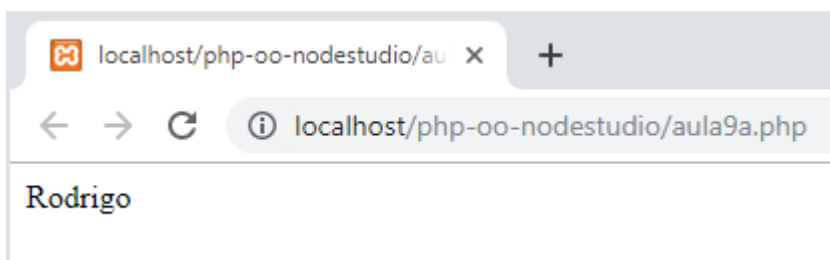
classes/aula9a/Rodrigo.php

```
<?php

include_once 'classes/aula9a/Pessoa.php';

class Rodrigo extends Pessoa {
    const nome = "Oliveira";

    public function exibirNome(){
        echo self::nome;
    }
}
```



aula9b.php

```
<?php

include_once 'classes/aula9b/Rodrigo.php';

$rodrigo = new Rodrigo();
$rodrigo->exibirNome();
```

classes/aula9b/Pessoa.php

```
<?php

class Pessoa {
    const nome = "Rodrigo";

    public function exibirNome(){
        echo self::nome;
    }
}
```

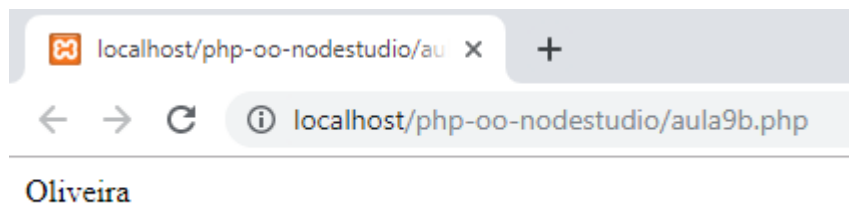
classes/aula9b/Rodrigo.php

```
<?php

include_once 'classes/aula9a/Pessoa.php';

class Rodrigo extends Pessoa {
    const nome = "Oliveira";

    public function exibirNome(){
        echo self::nome;
    }
}
```



aula9c.php

```
<?php

include_once 'classes/aula9c/Rodrigo.php';

$rodrigo = new Rodrigo();
$rodrigo->exibirNome();
```

classes/aula9c/Pessoa.php

```
<?php

class Pessoa {
    const nome = "Rodrigo";

    public function exibirNome(){
        echo self::nome;
    }
}
```

classes/aula9c/Rodrigo.php

```
<?php

include_once 'classes/aula9a/Pessoa.php';

class Rodrigo extends Pessoa {
    const nome = "Oliveira";

    public function exibirNome(){
        echo parent::nome;
    }
}
```

Aula 10 - Métodos e atributos estáticos

aula10.php

```
<?php

include_once 'classes/aula10/Login.php';

Login::$user = "Admin";
Login::verificaLogin();

$login = new Login();
$login ->sairSistema();
```

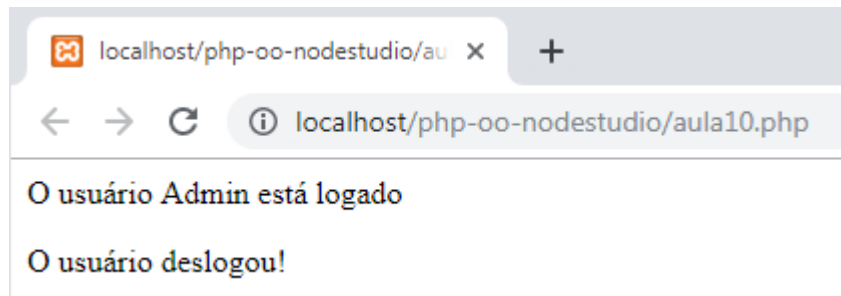
classes/aula10/Login.php

```
<?php

class Login {
    public static $user;

    public static function verificaLogin(){
        echo "<p>O usuário " . self::$user . " está logado</p>";
    }

    public function sairSistema(){
        echo "<p>O usuário deslogou!</p>";
    }
}
```



Aula 11 - Polimorfismo

aula11a.php

```
<?php

include_once 'classes/aula11a/Cavalo.php';

$animal = new Cavalo();
$animal->Andar();
```

classes/aula11a/Animal.php

```
<?php

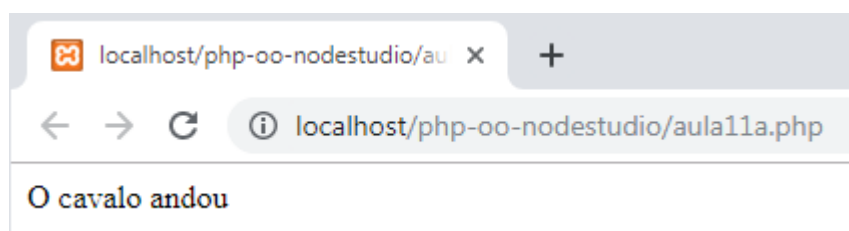
class Animal {
    public function Andar(){
        echo "O animal andou";
    }
}
```

classes/aula11a/Cavalo.php

```
<?php

include_once 'classes/aula11a/Animal.php';

class Cavalo extends Animal{
    public function Andar(){
        echo "O cavalo andou";
    }
}
```



aula11b.php

```
<?php

include_once 'classes/aula11b/Cavalo.php';

$animal = new Cavalo();
$animal->Andar();
```

classes/aula11b/Animal.php

```
<?php

class Animal {

    public function Andar(){
        $this->Correr();
    }

    public function Correr(){
        echo "O animal correu";
    }
}
```

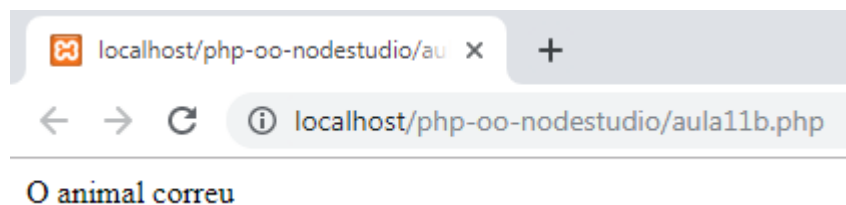
classes/aula11b/Cavalo.php

```
<?php

include_once 'classes/aula11b/Animal.php';

class Cavalo extends Animal{

    public function Andar(){
        $this->Correr();
    }
}
```



Aula 12 - Interfaces

aula12.php

```
<?php

include_once 'classes/aula12/Noticia.php';

$noticia = New Noticia();
$noticia->create();
$noticia->read();
$noticia->update(3);
$noticia->delete();
```

classes/aula12/Crud.php

```
<?php

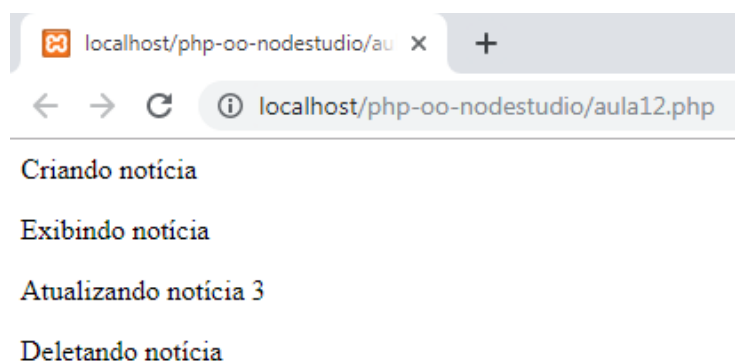
interface Crud {
    public function create();
    public function read();
    public function update($data);
    public function delete();
}
```

classes/aula12/Noticia.php

```
<?php

include_once 'classes/aula12/Crud.php';

class Noticia implements Crud{
    public function create(){
        echo "<p>Criando notícia</p>";
    }
    public function read(){
        echo "<p>Exibindo notícia</p>";
    }
    public function update($data){
        echo "<p>Atualizando notícia " . $data . "</p>";
    }
    public function delete(){
        echo "<p>Deletando notícia</p>";
    }
}
```



Aula 13 - Namespaces

aula13.php

```
<?php

require_once 'classes/aula13/classes/Produto.php';
require_once 'classes/aula13/models/Produto.php';

// use models\Produto;

use models\Produto as productModel;
use classes\Produto as productClass;

// $produto = new Produto();
$produto = new productModel();
$produto->mostrarDetalhes();

echo "<br>";

$produto2 = new productClass();
$produto2->mostrarDetalhes();
```

classes/aula13/classes/Produto.php

```
<?php

namespace classes;

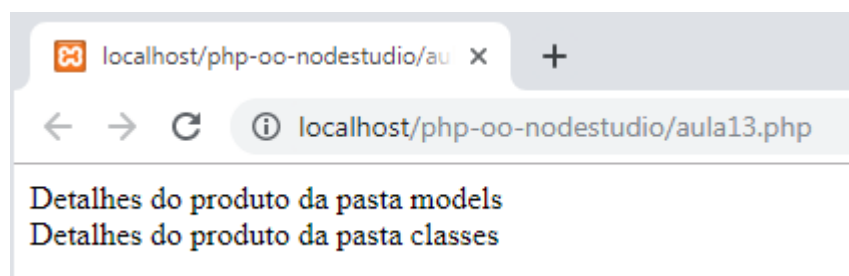
class Produto {
    public function mostrarDetalhes(){
        echo "Detalhes do produto da pasta classes";
    }
}
```

classes/aula13/models/Produto.php

```
<?php

namespace models;

class Produto {
    public function mostrarDetalhes(){
        echo "Detalhes do produto da pasta models";
    }
}
```



Aula 14 - Referência e clonagem de objetos

aula14.php

```
<?php

include_once 'classes/aula14/Pessoa.php';

$pessoa = new Pessoa;
$pessoa->idade = 25;

// $pessoa2 = $pessoa;
$pessoa2 = clone $pessoa;
$pessoa2->idade = 35;

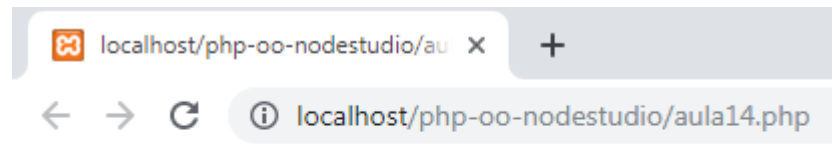
echo $pessoa->idade;
```

classes/aula14/Pessoa.php

```
<?php

class Pessoa {
    public $idade;

    public function __clone() {
        echo "<p>Clonagem de objetos</p>";
    }
}
```



Clonagem de objetos

25

Aula 15 - Tratamento de exceções

aula15.php

```
<?php

include_once 'classes/aula15/Newsletter.php';

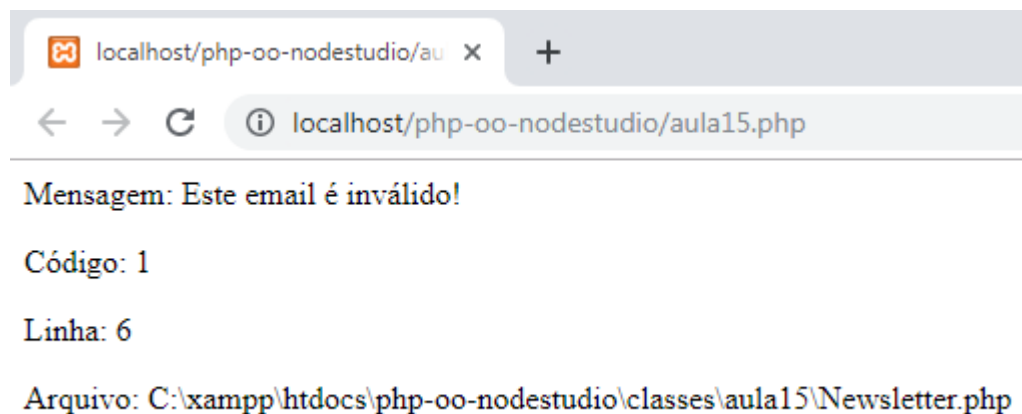
$newsletter = new Newsletter();

try {
    $newsletter->cadastrarEmail("contato@");
} catch (Exception $exc) {
    echo "<p>Mensagem: " . $exc->getMessage() . "</p>";
    echo "<p>Código: " . $exc->getCode() . "</p>";
    echo "<p>Linha: " . $exc->getLine() . "</p>";
    echo "<p>Arquivo: " . $exc->getFile() . "</p>";
}
```

classes/aula15/Newsletter.php

```
<?php

class Newsletter {
    public function cadastrarEmail($email){
        if(!filter_var($email, FILTER_VALIDATE_EMAIL)):
            throw new Exception("Este email é inválido!", 1);
        else:
            echo "Email cadastrado com sucesso!";
        endif;
    }
}
```



Aula 16 - Relação entre objetos (Associação)

1. Associação é a relação mais comum entre dois objetos;
2. Na associação, um objeto possui uma referência à outro objeto, podendo visualizar seus atributos ou mesmo acionar uma de suas funcionalidades (métodos);
3. A forma mais comum de implementar uma associação é ter um objeto como atributo de outro;
4. No exemplo, foi criado um objeto do tipo Cliente e outro do tipo Pedido;
5. Um dos atributos do pedido é o cliente;
6. Leia-se um objeto - está relacionado com - outro objeto;

aula16.php

```
<?php

include_once 'classes/aula16/Cliente.php';
include_once 'classes/aula16/Pedido.php';

$cliente = new Cliente();
$cliente->nome = "Rodrigo Oliveira";
$cliente->endereco = "Rua das Acácias, 58";

$pedido = new Pedido();
$pedido->numero = "123";
$pedido->cliente = $cliente; // Relacionamento por ASSOCIAÇÃO

$dados = array(
    'numero_pedido' => $pedido->numero,
    'cliente' => $pedido->cliente->nome,
    'endereco_cliente' => $pedido->cliente->endereco,
);

// var_dump($dados);
print_r($dados);
```

classes/aula16/Cliente.php

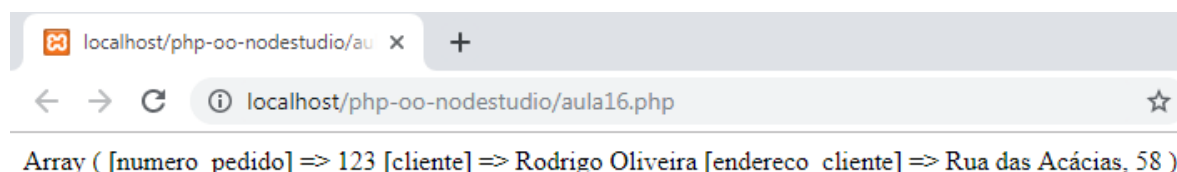
```
<?php

class Cliente {
    public $nome;
    public $endereco;
}
```

classes/aula16/Pedido.php

```
<?php

class Pedido {
    public $numero;
    public $cliente;
}
```



Aula 17 - Relação entre objetos (Agregação)

Na agregação uma classe precisa de outra para executar uma ação.

Na agregação o objeto agrega outro objeto, ou seja, torna um objeto externo parte de si. Assim o objeto pai poderá utilizar funcionalidades do objeto agregado, mas ele não depende dele (agregado) para existir.

aula17.php

```
<?php

include_once 'classes/aula17/Produtos.php';
include_once 'classes/aula17/Carrinho.php';

$produto1 = new Produtos("Notebook", "1500");
$produto2 = new Produtos("Mouse", "50");

$carrinho = new Carrinho();
$carrinho->adiciona($produto1);
$carrinho->adiciona($produto2);
$carrinho->exibe();
```

classes/aula17/Produtos.php

```
<?php

class Produtos {
    public $nome;
    public $valor;

    public function __construct($nome, $valor) {
        $this->nome = $nome;
        $this->valor = $valor;
    }
}
```

classes/aula17/Carrinho.php

```
<?php

class Carrinho {
    public $produtos;

    // Agregação
    public function adiciona(Produtos $produto){
        $this->produtos[] = $produto;
    }

    public function exibe(){
        foreach($this->produtos as $produto){
            echo "$produto->nome<br>";
            echo "$produto->valor<hr>";
        }
    }
}
```

localhost/php-oo-nodestudio/au x +

← → ↻ ⓘ localhost/php-oo-nodestudio/aula17.php

Notebook
1500

Mouse
50

Aula 18 - Relação entre objetos (Composição)

Na composição, uma instância cria a classe de outra classe dentro de si própria, quando ela for destruída, a outra classe também será.

A diferença da composição para a agregação é que quando o objeto "todo" é destruído, suas partes também são, exatamente por terem sido criadas pelo objeto "todo". Podemos dizer que na composição o objeto-pai "possui" as instâncias de suas partes.

aula18.php

```
<?php

include_once 'classes/aula18/Exibe.php';
include_once 'classes/aula18/Pessoa.php';

$exibe = new exibe("José");
$exibe->exibeNome();
```

classes/aula18/Pessoa.php

```
<?php

class Pessoa {
    public function atribuiNome($nome){
        return "O nome da pessoa é " . $nome;
    }
}
```

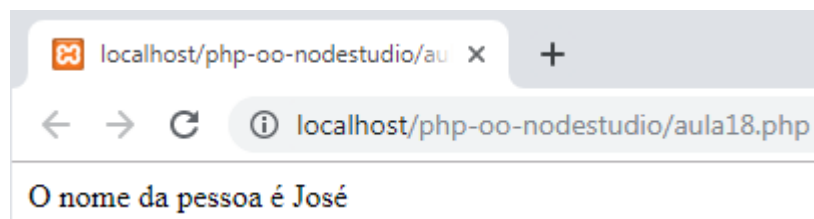
classes/aula18/Exibe.php

```
<?php

class Exibe {
    public $pessoa;
    public $nome;

    public function __construct($nome) {
        $this->pessoa = new Pessoa(); // Composição
        $this->nome = $nome;
    }

    public function exibeNome(){
        echo $this->pessoa->atribuiNome($this->nome);
    }
}
```



Aula 19 - Métodos mágicos

aula19a.php

```
<?php

include_once 'classes/aula19a/Pessoa.php';

$ pessoa = New Pessoa();
$ pessoa->nome = "Danilo";
echo $ pessoa->nome;
```

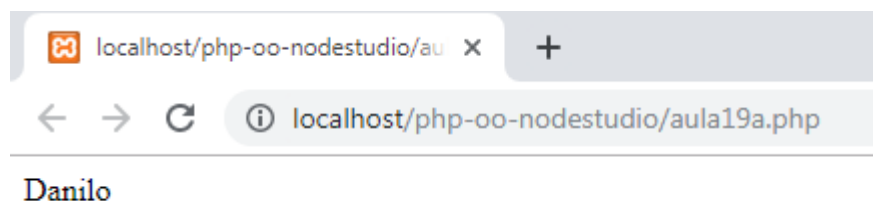
classes/aula19a/Pessoa.php

```
<?php

class Pessoa {
    private $nome;

    public function __set($nome, $valor) {
        $this->nome = $valor;
    }

    public function __get($nome){
        return $this->nome;
    }
}
```



aula19b.php

```
<?php

include_once 'classes/aula19b/Pessoa.php';

$ pessoa = New Pessoa();
$ pessoa->nome = "Danilo";
$ pessoa->idade = 50;
$ pessoa->sexo = "M";

echo $ pessoa->nome . "<br>";
echo $ pessoa->idade . "<br>";
echo $ pessoa->sexo . "<br>";

echo $ pessoa . "<br>";

echo $ pessoa();
```

classes/aula19b/Pessoa.php

```
<?php

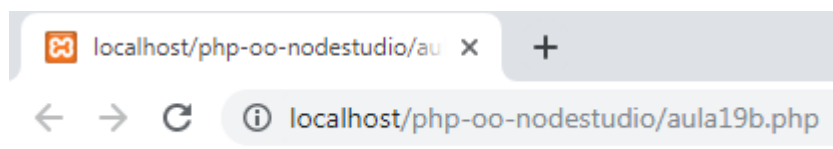
class Pessoa {
    private $dados = array();

    public function __set($nome, $valor) {
        $this->dados[$nome] = $valor;
    }

    public function __get($nome){
        return $this->dados[$nome];
    }

    public function __toString() {
        return "Tentei imprimir o objeto";
    }

    public function __invoke() {
        return "Objeto como função";
    }
}
```



Danilo

50

M

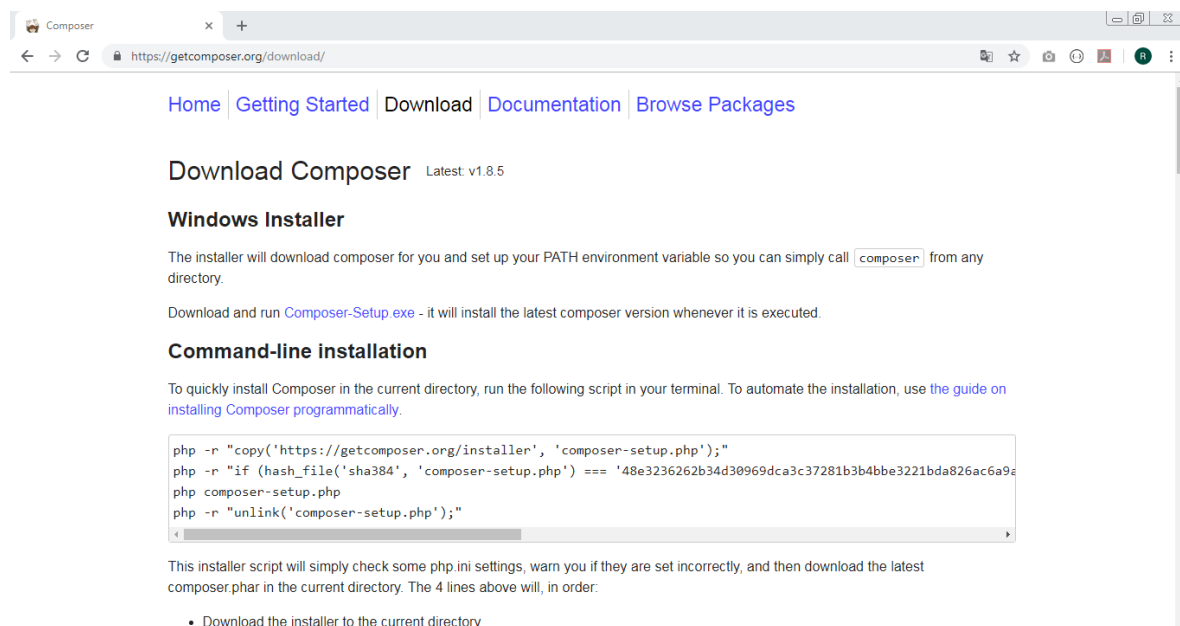
Tentei imprimir o objeto

Objeto como função

Aula 20 - Composer

O Composer é um gerenciador de dependências do PHP.

O site do Composer é: <https://getcomposer.org/download/>



Para fazer o download, basta clicar no link: [Composer-Setup.exe](https://getcomposer.org/Composer-Setup.exe) (https://getcomposer.org/Composer-Setup.exe)

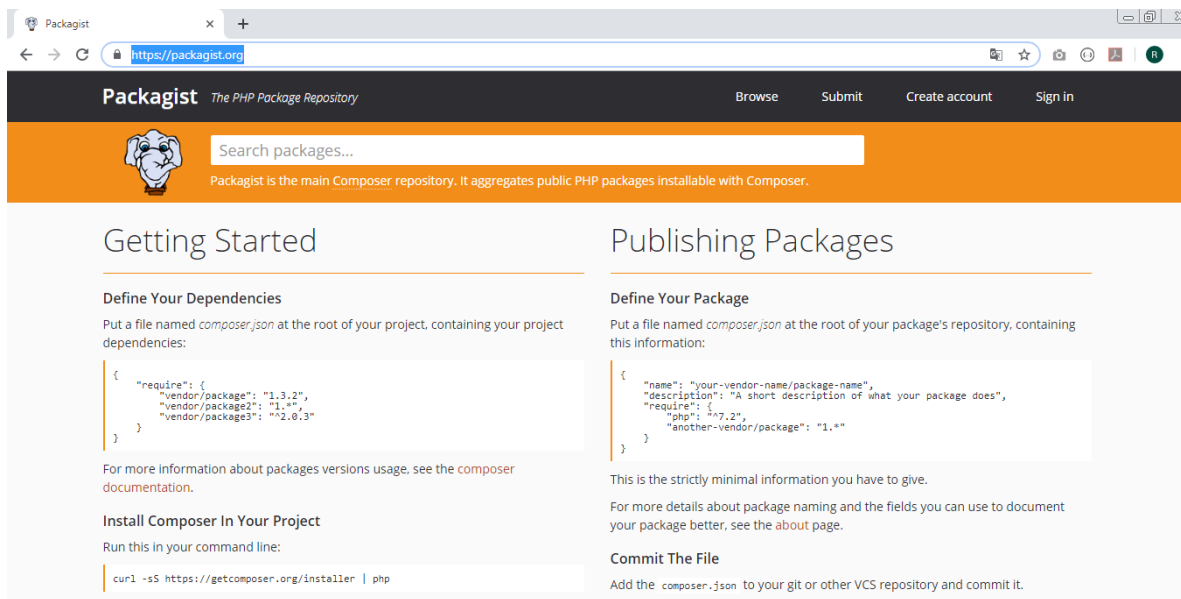
- Instalar no modo desenvolvedor.
- Reiniciar o computador e inicializar o prompt de comando (Cmd)
- Digite **composer** e pressione a tecla enter. Se estiver sendo instalado corretamente será exibido algo semelhante a:

```
Composer
Composer version 1.8.2 2019-01-29 15:00:53
Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
      --ansi                Force ANSI output
      --no-ansi             Disable ANSI output
  -n, --no-interaction      Do not ask any interactive question
      --profile             Display timing and memory usage information
      --no-plugins          Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  -vvvv, --verbose          Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug
```

Repositório de bibliotecas de terceiros:

<https://packagist.org/>



The screenshot shows the Packagist website, which is the main Composer repository for PHP. The page has a dark blue header with the Packagist logo and navigation links: Browse, Submit, Create account, and Sign in. Below the header is an orange banner with a search bar and the text "Packagist is the main Composer repository. It aggregates public PHP packages installable with Composer." The main content area is divided into two columns: "Getting Started" and "Publishing Packages".

Getting Started

Define Your Dependencies

Put a file named `composer.json` at the root of your project, containing your project dependencies:

```
{
  "require": {
    "vendor/package": "1.3.2",
    "vendor/package2": "1.*",
    "vendor/package3": "^2.0.3"
  }
}
```

For more information about packages versions usage, see the [composer documentation](#).

Install Composer In Your Project

Run this in your command line:

```
curl -sS https://getcomposer.org/installer | php
```

Publishing Packages

Define Your Package

Put a file named `composer.json` at the root of your package's repository, containing this information:

```
{
  "name": "your-vendor-name/package-name",
  "description": "A short description of what your package does",
  "require": {
    "php": "^7.2",
    "another-vendor/package": "1.*"
  }
}
```

This is the strictly minimal information you have to give.

For more details about package naming and the fields you can use to document your package better, see the [about](#) page.

Commit The File

Add the `composer.json` to your git or other VCS repository and commit it.

Aula 21 - Variáveis de ambiente e servidor embutido

No Windows, clique em:

Computador-> Propriedades -> Configurações avançadas de sistema
->Variáveis de Ambiente

Clique na variável Path e verifique que quando o composer foi instalado, automaticamente foi incluído no path para ser acessado de qualquer pasta do sistema:

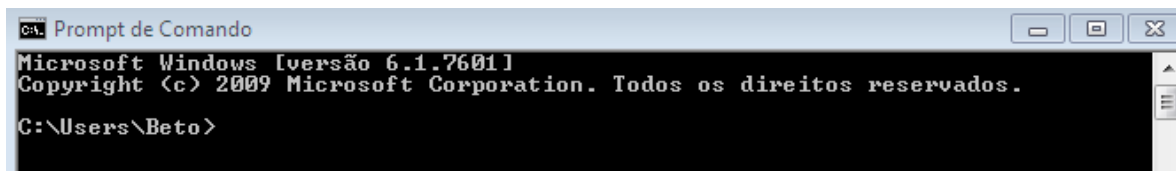
Caminho do PHP:

C:\xampp\php;

Caminho do Composer:

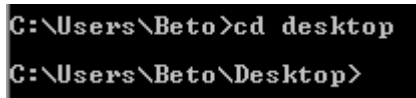
C:\ProgramData\ComposerSetup\bin;

O prompt de comando do Windows é aberto na pasta do usuário:



```
C:\ Prompt de Comando
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.
C:\Users\Beto>
```

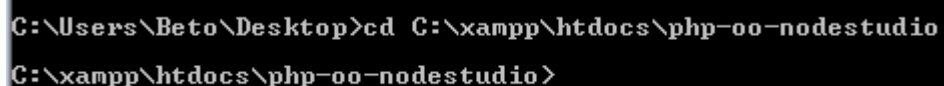
Para navegar até a área de trabalho:



```
C:\Users\Beto>cd desktop
C:\Users\Beto\Desktop>
```

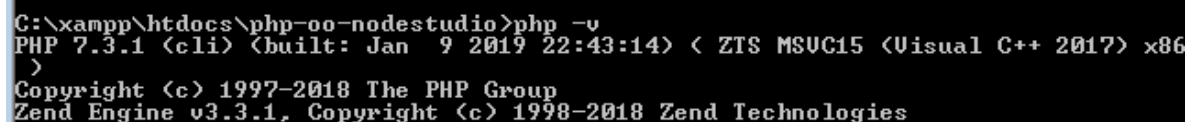
Para navegar para a pasta de projetos:

cd C:\xampp\htdocs\php-oo-nodestudio



```
C:\Users\Beto\Desktop>cd C:\xampp\htdocs\php-oo-nodestudio
C:\xampp\htdocs\php-oo-nodestudio>
```

Versão instalada do PHP:



```
C:\xampp\htdocs\php-oo-nodestudio>php -v
PHP 7.3.1 (cli) (built: Jan 9 2019 22:43:14) < ZTS MSVC15 (Visual C++ 2017) x86_64
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.1, Copyright (c) 1998-2018 Zend Technologies
```

Rodando o servidor embutido do PHP

Desative o servidor XAMPP ou WAMP e no terminal de comando entre com:

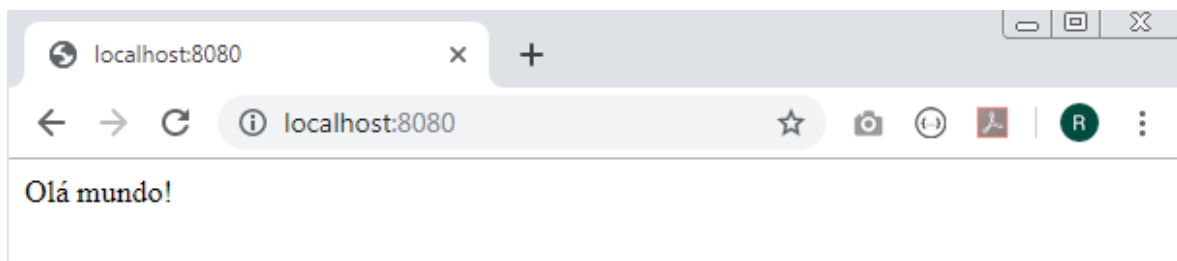
```
php -S localhost:8080
```

```
C:\xampp\htdocs\php-oo-nodestudio>php -S localhost:8080
PHP 7.3.1 Development Server started at Mon May 13 03:52:09 2019
Listening on http://localhost:8080
Document root is C:\xampp\htdocs\php-oo-nodestudio
Press Ctrl-C to quit.
[Mon May 13 03:53:46 2019] [::1]:50014 [404]: / - No such file or directory
[Mon May 13 03:53:47 2019] [::1]:50015 [404]: /favicon.ico - No such file or directory
[Mon May 13 03:54:53 2019] [::1]:50018 [200]: /
```

index.php

```
<?php
```

```
echo "Olá mundo!";
```



Para parar o servidor, pressione: **Ctrl-C**

Aula 22 - Composer Init, Require e Update

Configurando o Composer

```
composer init
```

```
Package name: nodestudio/composer
Description: Teste Composer
Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br> <Enter>
Minimum Stability []: <Enter>
Package Type: <Enter>
License: MIT
Would you like to define your dependencies <require> interactively [yes]? yes
Search for a package: dompdf/dompdf
Enter the version constraint to require (or leave blank to use the latest
version): <Enter>
Search for a package: <Enter>
Would you like to define your dev dependencies interactively: no
Add PSR-4 autoload mapping? Maps namespace "Nodestudio/Composer" to the entered
relative path. [src/, n to skip]: n
Do you confirm generation [yes]? yes
```

```
C:\xampp\htdocs\php-oo-nodestudio>composer init
```

```
Welcome to the Composer config generator
```

```
This command will guide you through creating your composer.json config.
```

```
Package name (<vendor>/<name>) [betol/php-oo-nodestudio]: nodestudio/composer
Description []: Teste Composer
Author [Roberto Pinheiro <betopinheiro1005@yahoo.com.br>, n to skip]:
Minimum Stability []:
Package Type (e.g. library, project, metapackage, composer-plugin) []:
License []: MIT
```

```
Define your dependencies.
```

```
Would you like to define your dependencies (require) interactively [yes]? yes
```

```
Search for a package: dompdf/dompdf
```

```
Enter the version constraint to require (or leave blank to use the latest version):
```

```
Using version ^1.1 for dompdf/dompdf
```

```
Search for a package:
```

```
Would you like to define your dev dependencies (require-dev) interactively [yes]? no
```

```
Add PSR-4 autoload mapping? Maps namespace "Nodestudio\Composer" to the entered relative path. [src/, n to skip]: n
```

```
{
  "name": "nodestudio/composer",
  "description": "Teste Composer",
  "require": {
    "dompdf/dompdf": "^1.1"
  },
  "license": "MIT",
  "authors": [
    {
      "name": "Roberto Pinheiro",
      "email": "betopinheiro1005@yahoo.com.br"
    }
  ]
}
```

```
Do you confirm generation [yes]? yes
```

```

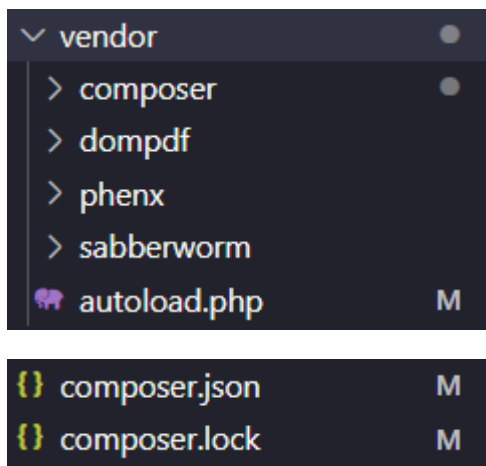
Would you like to install dependencies now [yes]? yes
Loading composer repositories with package information
Updating dependencies
Lock file operations: 4 installs, 0 updates, 0 removals
- Locking dompdf/dompdf (v1.1.1)
- Locking phenx/php-font-lib (0.5.4)
- Locking phenx/php-svg-lib (0.3.4)
- Locking sabberworm/php-css-parser (8.4.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 4 installs, 0 updates, 0 removals
- Installing sabberworm/php-css-parser (8.4.0): Extracting archive
- Installing phenx/php-svg-lib (0.3.4): Extracting archive
- Installing phenx/php-font-lib (0.5.4): Extracting archive
- Installing dompdf/dompdf (v1.1.1): Extracting archive
3 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating autoload files

```

Serão gerados dois arquivos:

- `composer.json`
- `composer.lock` (não alterar)

E a uma pasta chamada `vendor` (onde são armazenadas as dependências)



`composer.json`

```

{
    "name": "nodestudio/composer",
    "description": "Teste Composer",
    "require": {
        "dompdf/dompdf": "^1.1"
    },
    "autoload": {
        "psr-4": {
            "Nodestudio\\Composer\\": "src/"
        }
    },
    "authors": [
        {
            "name": "Roberto Pinheiro",
            "email": "betopinheiro1005@yahoo.com.br"
        }
    ]
}

```

vendor/autoload.php

Faz o carregamento automático e dinâmico dos pacotes da pasta **vendor**.

```
<?php

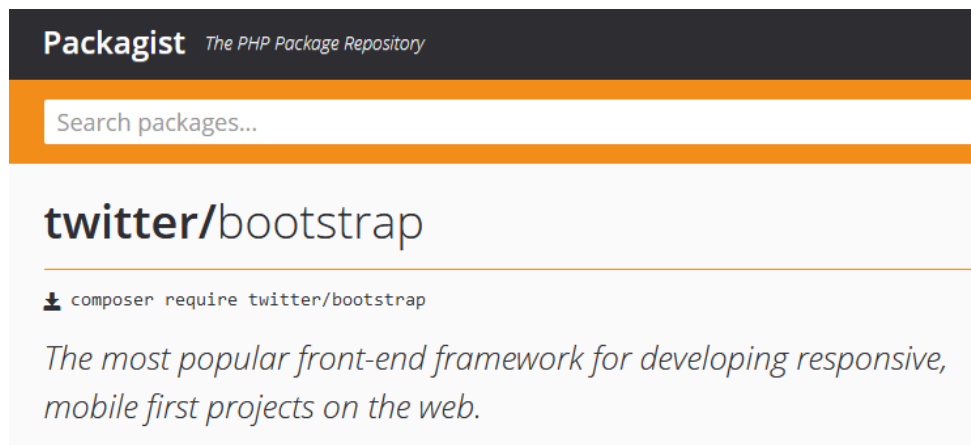
// autoload.php @generated by Composer

require_once __DIR__ . '/composer/autoload_real.php';

return ComposerAutoloaderInit8b2f9258d2dae6123e83ea8eb181d41d::getLoader();
```

Instalando um novo pacote

Para instalar o pacote bootstrap:

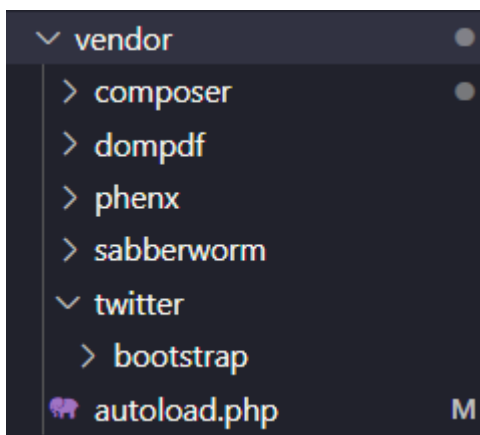


```
composer require twitter/bootstrap
```

```
C:\xampp\htdocs\php-oo-nodestudio>composer require twitter/bootstrap
Using version ^5.1 for twitter/bootstrap
./composer.json has been updated
Running composer update twitter/bootstrap
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
  - Locking twitter/bootstrap (v5.1.3)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Downloading twitter/bootstrap (v5.1.3)
  - Installing twitter/bootstrap (v5.1.3): Extracting archive
Generating autoload files
```

composer.json

```
{
  "name": "nodestudio/composer",
  "description": "Teste Composer",
  "require": {
    "dompdf/dompdf": "^1.1",
    "twitter/bootstrap": "^5.1"
  },
  "license": "MIT",
  "authors": [
    {
      "name": "Roberto Pinheiro",
      "email": "betopinheiro1005@yahoo.com.br"
    }
  ]
}
```



Removendo um pacote

Para remover o pacote **domPDF** exclua a linha correspondente no arquivo **composer.json**

composer.json

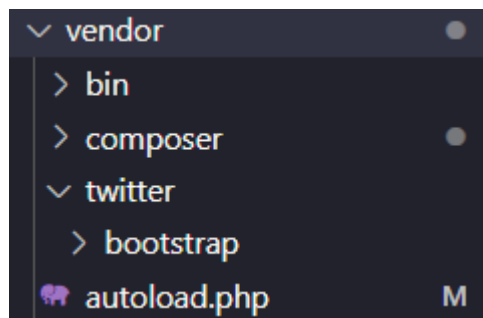
```
{
  "name": "nodestudio/composer",
  "description": "Teste Composer",
  "require": {
    "twitter/bootstrap": "^5.1"
  },
  "license": "MIT",
  "authors": [
    {
      "name": "Roberto Pinheiro",
      "email": "betopinheiro1005@yahoo.com.br"
    }
  ]
}
```

E em seguida execute o seguinte comando:

```
composer update
```



```
C:\xampp\htdocs\php-oo-nodestudio>composer update
Loading composer repositories with package information
Updating dependencies
Lock file operations: 0 installs, 0 updates, 4 removals
- Removing dompdf/dompdf (v1.1.1)
- Removing phenx/php-font-lib (0.5.4)
- Removing phenx/php-svg-lib (0.3.4)
- Removing sabberworm/php-css-parser (8.4.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 0 installs, 0 updates, 4 removals
- Removing sabberworm/php-css-parser (8.4.0)
- Removing phenx/php-svg-lib (0.3.4)
- Removing phenx/php-font-lib (0.5.4)
- Removing dompdf/dompdf (v1.1.1)
Generating autoload files
```



Aula 23 - Composer Install

Instalando as dependências manualmente

- Inicialmente apague todos os arquivos criados anteriormente pelo Composer.
- Em seguida, crie o arquivo `composer.json` com o seguinte conteúdo:

`composer.json`

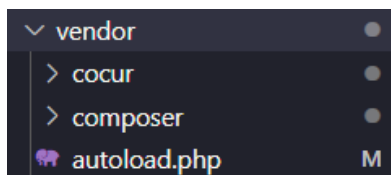
```
{
    "require": {
        "cocur/slugify": "3.2"
    }
}
```

slug é usado para criar uma URL amigável.

Agora, entre com o comando:

```
composer install
```

```
C:\xampp\htdocs\php-oo-nodestudio>composer install
No composer.lock file present. Updating dependencies to latest instead of installing from lock file. See https://getcomposer.org/install for more information.
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
  - Locking cocur/slugify (v3.2)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Downloading cocur/slugify (v3.2)
  - Installing cocur/slugify (v3.2): Extracting archive
Generating autoload files
```



`index.php`

```
<?php
```

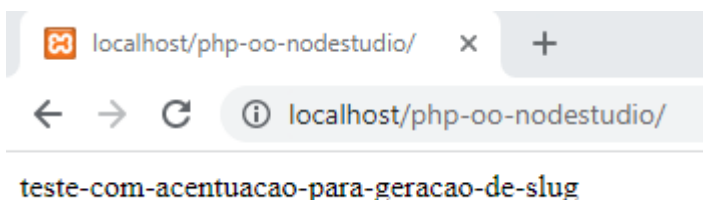
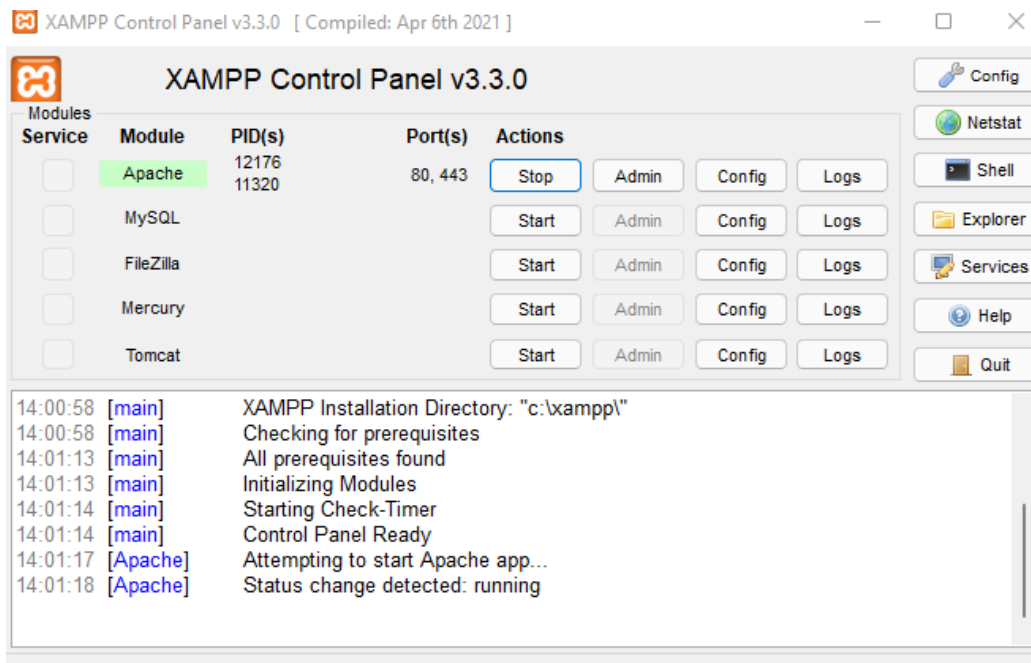
```
    // echo "Olá mundo!";

    require_once 'vendor/autoload.php';

    use Cocur\Slugify\Slugify;

    $slug = new Slugify();
    echo $slug->slugify('Teste com acentuação para geração de slug!');
```

- Baixe e instale o `XAMPP` em seu computador.
- Inicie o servidor `Apache` no `XAMPP`.



Alterando o caractere separador

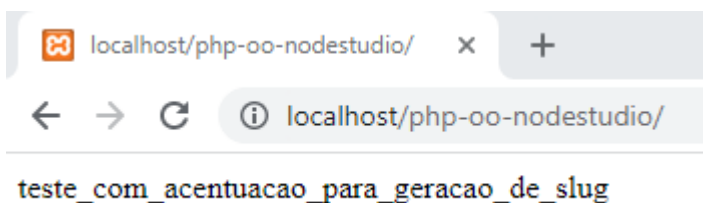
index.php

```
<?php
```

```
    require_once 'vendor/autoload.php';

    use Cocur\Slugify\Slugify;

    $slug = new Slugify();
    echo $slug->slugify('Teste com acentuação para geração de slug!', '_');
```



Substituindo texto no slug

index.php

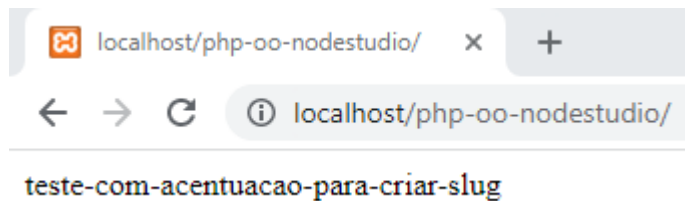
```
<?php

    // echo "Olá mundo!";

    require_once 'vendor/autoload.php';

    use Cocur\Slugify\Slugify;

    $slug = new Slugify();
    $slug->addRule('geração de', 'criar');
    echo $slug->slugify('Teste com acentuação para geração de slug!');
```



Aula 24 - CRUD com PDO, Singleton e Autoload - Parte 1

CRUD\App\Model\Connection.php

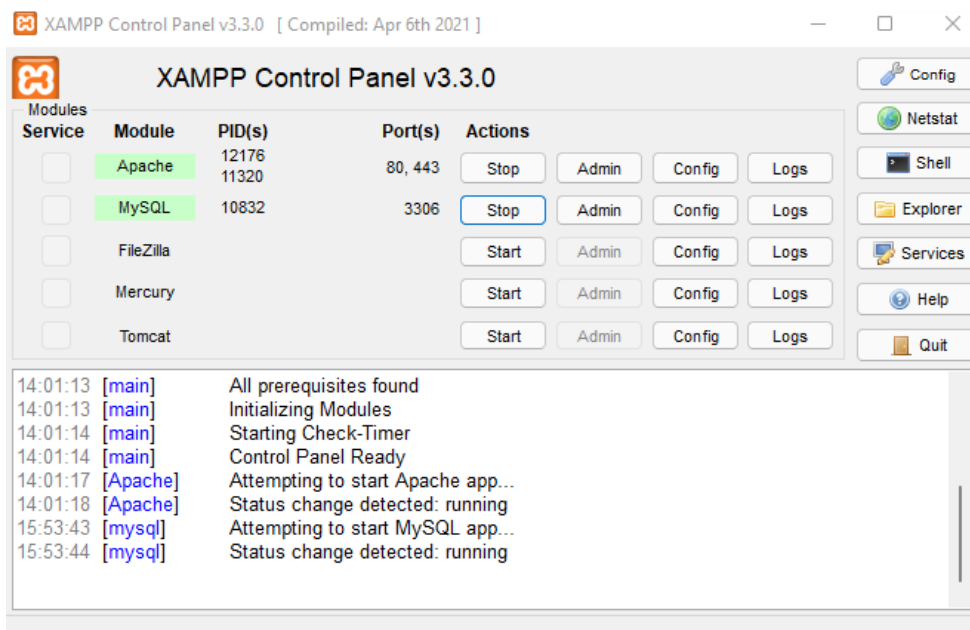
```
<?php

//DAO - Data Access Object
//self é usado para acessar membros estáticos.

class Connection{

    private static $instance;

    public static function getConn(){
        if(!isset(self::$instance)){
            self::$instance =
                new PDO('mysql:host=localhost;dbname=pdo;charset=utf8','root','');
        } else {
            return self::$instance;
        }
    }
}
```



- No PHPMyAdmin, crie uma BD chamada **pdo** (Agrupamento: utf8_general_ci)
- E dentro dessa BD crie uma tabela chamada **produtos** com 3 colunas: id, name e description

Nome	Tipo	Tamanho/Valores*	Predefinido	Agrupamento (Collation)	Atributos	Nulo	Índice	Comentários	Virtualidade
id	INT		Nenhum			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
name	VARCHAR	150	Nenhum			<input type="checkbox"/>	---	<input type="checkbox"/>	
description	TEXT		Nenhum			<input type="checkbox"/>	---	<input type="checkbox"/>	

Estrutura

Comentários da tabela: Colação: Motor de armazenamento: InnoDB

Definição da PARTIÇÃO:

Partição por: (Expressão ou lista de colun)

Partições:

Pré-visualizar SQL Guarda

CRUD\App\Model\Product.php

```
<?php
```

```
class Product{

    private $id, $name, $description;

    public function getId()
    {
        return $this->id;
    }

    public function setId($id)
    {
        $this->id = $id;
    }

    public function getName()
    {
        return $this->name;
    }

    public function setName($name)
    {
        $this->name = $name;
    }

    public function getDescription()
    {
        return $this->description;
    }

    public function setDescription($description)
    {
        $this->description = $description;
    }

}
```

CRUD\App\Model\ProductDao.php

```
<?php
```

```
class ProductDao{

    public function create(Product $p){

    }

    public function read(){

    }

    public function update(Product $p){

    }

    public function delete($id){

    }

}
```

Aula 25 - CRUD com PDO, Singleton e Autoload - Parte 2

CRUD\App\Model\Product.php

```
<?php

namespace App\Model;

class Product{

    private $id, $name, $description;

    public function getId()
    {
        return $this->id;
    }

    public function setId($id)
    {
        $this->id = $id;
    }

    public function getName()
    {
        return $this->name;
    }

    public function setName($name)
    {
        $this->name = $name;
    }

    public function getDescription()
    {
        return $this->description;
    }

    public function setDescription($description)
    {
        $this->description = $description;
    }

}
```


CRUD\App\Model\ProductDao.php

```
<?php

namespace App\Model;

class ProductDao{

    public function create(Product $p){

    }

    public function read(){

    }

    public function update(Product $p){

    }

    public function delete($id){

    }

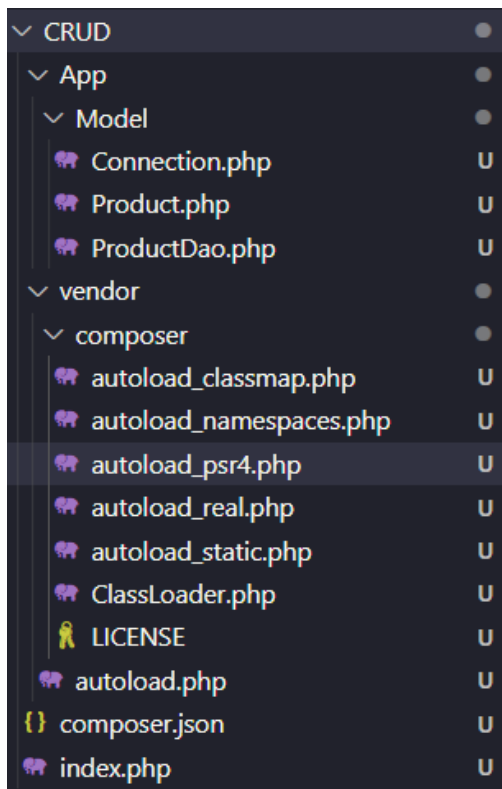
}
```

CRUD\composer.json

```
{
    "autoload": {
        "psr-4": {
            "App\\": "App/"
        }
    }
}
```

```
composer dump-autoload
```

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/php-oo-nodestudio/CRUD (master)
$ composer dump-autoload
Generating autoload files
Generated autoload files
```



CRUD\vendor\composer\autoload_psr4.php

```
<?php

// autoload_psr4.php @generated by Composer

$vendorDir = dirname(dirname(__FILE__));
$baseDir = dirname($vendorDir);

return array(
    'App\\' => array($baseDir . '/App'),
);
```

CRUD\index.php

```
<?php

require_once('vendor/autoload.php');
```

Aula 26 - CRUD com PDO, Singleton e Autoload - Parte 3

CRUD\App\Model\Connection.php

```
<?php

namespace App\Model;

//DAO - Data Access Object
//self é usado para acessar membros estáticos.

class Connection{

    private static $instance;

    public static function getConn(){
        if(!isset(self::$instance)):
            self::$instance =
                new PDO('mysql:host=localhost;dbname=pdo;charset=utf8','root','');
            endif;

        return self::$instance;
    }
}
```

CRUD\App\Model\ProductDao.php

```
<?php

namespace App\Model;

use PDOException;

class ProductDao{

    public function create(Product $p){
        $sql = 'INSERT INTO products(name,description) VALUES (?,?)'; //bind values
        $stmt = Connection::getConn()->prepare($sql);

        $stmt->bindValue(1,$p->getName());
        $stmt->bindValue(2,$p->getDescription());

        $stmt->execute();
    }
}
```

CRUD\index.php

```
<?php
```

```
// SETS
```

```
require_once('vendor/autoload.php');
```

```
$product = new \App\Model\Product();  
$product->setName('Notebook DELL');  
$product->setDescription('i5, 4gb');
```

```
$productDao = new \App\Model\ProductDao();  
$productDao->create($product);
```

```
$product = new \App\Model\Product();  
$product->setName('Microfone AT2020');  
$product->setDescription('Microfone condensador');
```

```
$productDao = new \App\Model\ProductDao();  
$productDao->create($product);
```

 A mostrar registos de 0 - 1 (2 total, A consulta demorou 0,0017 segundos.)

`SELECT * FROM `products``

☐ Perfil [[Editar em linha](#)] [[Edita](#)] [[Explicar SQL](#)] [[Criar código PHP](#)] [[Atualizar](#)]

☐ Mostrar tudo | Número de registos: | Filtrar registos: | Ordenar pela chave:

+ Opções

				id	name	description
<input type="checkbox"/>	 Edita	 Copiar	 Apagar	1	Notebook DELL	i5, 4gb
<input type="checkbox"/>	 Edita	 Copiar	 Apagar	2	Microfone AT2020	Microfone condensador

 ☐ [Marcar todos](#) | Com os seleccionados:  [Edita](#)  [Copiar](#)  [Apagar](#)  [Exportar](#)

Aula 27 - CRUD com PDO, Singleton e Autoload - Parte 4

CRUD\App\Model\Connection.php

```
<?php

namespace App\Model;

//DAO - Data Access Object
//self é usado para acessar membros estáticos.

class Connection{

    private static $instance;

    public static function getConn(){
        if(!isset(self::$instance)):
            self::$instance =
                new \PDO('mysql:host=localhost;dbname=pdo;charset=utf8','root','');
            endif;

        return self::$instance;
    }
}
```

CRUD\App\Model\Product.php

```
<?php

namespace App\Model;

class Product{

    private $id, $name, $description;

    public function getId()
    {
        return $this->id;
    }

    public function setId($id)
    {
        $this->id = $id;
    }

    public function getName()
    {
        return $this->name;
    }

    public function setName($name)
    {
        $this->name = $name;
    }

    public function getDescription()
    {
        return $this->description;
    }

    public function setDescription($description)
    {
        $this->description = $description;
    }
}
```

CRUD\App\Model\ProductDao.php

```
<?php

namespace App\Model;

use PDOException;

class ProductDao{

    public function create(Product $p){
        try{
            $sql = 'INSERT INTO products(name,description) VALUES (?,?)';//bind values
            $stmt = Connection::getConn()->prepare($sql);

            $stmt->bindValue(1,$p->getName());
            $stmt->bindValue(2,$p->getDescription());

            $stmt->execute();

            echo 'Produto Cadastrado com Sucesso!';

            header('Location: ../index.php');
        }catch(PDOException $error){
            echo $error->getMessage();
        }
    }

    public function read(){
        try{
            $sql = 'SELECT * FROM products';
            $stmt = Connection::getConn()->prepare($sql);
            $stmt->execute();

            if($stmt->rowCount() > 0){
                $result = $stmt->fetchAll(\PDO::FETCH_ASSOC);//retorna um array com todos os
registros
                return $result;
            }else{
                return [];
            }
        }catch(PDOException $error){
            echo $error->getMessage();
        }
    }

    public function update(Product $p){
        try{
            $sql = 'UPDATE products SET name=?, description=? WHERE id=?';
            $stmt = Connection::getConn()->prepare($sql);
            $stmt->bindValue(1,$p->getName());
            $stmt->bindValue(2,$p->getDescription());
            $stmt->bindValue(3,$p->getId());

            $stmt->execute();

            header('Location: ../index.php');

        }catch(PDOException $error){
            echo $error->getMessage();
        }
    }

    public function delete($id){
        try{
            $sql = 'DELETE FROM products WHERE id = ?';
            $stmt = Connection::getConn()->prepare($sql);
            $stmt->bindValue(1,$id->getId());
```

```

        $stmt->execute();

        header('Location: ../index.php');

    } catch(PDOException $error){
        echo $error->getMessage();
    }
}
}

```

CRUD\index.php

```

<?php
    require_once('vendor/autoload.php');
    session_start();
?>

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Formulario</title>
</head>
<style>
    *{
        font-family: Verdana;
    }
    input[type="text"]{
        display: block;
        margin-bottom: 5px;
    }
    button{
        padding: 2px 10px;
        cursor: pointer;
        margin: 5px;
    }
    a{
        text-decoration: none;
        cursor: pointer;
    }
    table,
    tr,
    td {
        border: 1px solid black;
        border-collapse: collapse;
        padding: 5px;
    }
    theader tr td{
        text-align: center;
    }
</style>

<body>
    <a href="Views/form_create.php"><button><p>Cadastrar Produto</p></button></a>

    <?php
        $product = new \App\Model\ProductDao();
        $product->read();
    ?>

    <h1>Produtos Cadastrados</h1>

    <?php if ($product->read() == []) { ?>
        <h4>Nenhum produto cadastrado</h4>
    <?php } else { ?>
        <table>

```

```

        <tr>
            <td>Id</td>
            <td>Nome</td>
            <td>Descrição</td>
            <td style="text-align:center" colspan="2">Ação</td>
        </tr>
        <?php foreach ($product->read() as $item) { ?>
            <tr>
                <td><?php echo $item['id']; ?></td>
                <td><?php echo $item['name']; ?></td>
                <td><?php echo $item['description']; ?></td>
                <td style="text-align:center"><a href="Views/form_edit.php?id=<?php echo
$item['id']; ?>&name=<?php echo $item['name']; ?>&description=<?php echo
$item['description']; ?>"><button>Editar</button></a>
                <a href="Controller/Delete.php?id=<?php echo $item['id'];
?>"><button>Excluir</button></a></td>
            </tr>
        <?php } ?>
    </table>

    <br />
    <?php } ?>

</body>

</html>

```

CRUD\Controller\Create.php

```

<?php

require_once('../vendor/autoload.php');

use App\Model\Product;
use App\Model\ProductDao;

if(isset($_POST['name']) && isset($_POST['description'])) {

    $name = $_POST['name'];
    $description = $_POST['description'];

    if($name != null && $description != null) {
        $product = new Product();

        $product->setName($name);
        $product->setDescription($description);

        $productDao = new ProductDao();
        $productDao->create($product);
    } else {
        echo "<h3 style='color: #f00;'>É necessário preencher todos os campos do
formulário!</h3>";
        echo "<a href='../index.php'><button style='padding: 10px 20px;'>Produtos
cadastrados</button></a>";
    }
}

?>

```


CRUD\Controller\Read.php

```
<?php
require_once(' ../vendor/autoload.php');

use \App\Model\ProductDao;

$productDao = new ProductDao();
$productDao->read();
```

CRUD\Controller\Update.php

```
<?php

require_once(' ../vendor/autoload.php');

use App\Model\Product;
use App\Model\ProductDao;

if(isset($_POST['name']) && isset($_POST['description'])) {

    $id = $_POST['id'];
    $name = $_POST['name'];
    $description = $_POST['description'];

    if($name != null && $description != null) {
        $product = new Product();
        $product->setId($id);
        $product->setName($name);
        $product->setDescription($description);

        $productDao = new ProductDao();
        $productDao->update($product);
    } else {
        echo "<h3 style='color: #f00;'>É necessário preencher todos os campos do
formulário!</h3>";
        echo "<a href='../index.php'><button style='padding: 10px 20px;'>Produtos
cadastrados</button></a>";
    }
}
```

CRUD\Controller\Delete.php

```
<?php

require_once(' ../vendor/autoload.php');

use App\Model\Product;
use App\Model\ProductDao;

if(isset($_GET['id'])) {
    $id = $_GET['id'];
}

$product = new Product();
$product->setId($id);

$productDao = new ProductDao();
$productDao->delete($product);
```

CRUD\Views\form_create.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    *{
      font-family: Verdana;
    }
    input[type="text"]{
      display: block;
      margin-bottom: 5px;
      padding: 5px;
    }
    label, textarea{
      display: block;
      margin-bottom: 5px;
    }
    button{
      margin-top: 5px;
      padding: 10px 25px;
      cursor: pointer;
    }
    .info{
      color: #555;
      display: block;
      padding-bottom: 10px;
      font-size: 12px;
    }
  </style>
  <title>Cadastro de Produto</title>
</head>
<body>
  <h1>Cadastrar Produto</h1>

  <p class="info">* O preenchimento de todos os campos é obrigatório</p>

  <form action="../Controller/Create.php" method="post">
    <label for="name">Nome do produto:</label>
    <input type="text" name="name" id="name" size="60">
    <label for="description">Descrição do produto:</label>
    <textarea name="description" id="description" rows="10" cols="61"></textarea>
    <button type="submit">Cadastrar</button>
  </form>

</body>
</html>
```

CRUD\Views\form_edit.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    *{
      font-family: Verdana;
    }
    input[type="text"]{
      display: block;
      margin-bottom: 5px;
    }
    label, textarea{
      display: block;
      margin-bottom: 5px;
    }
    button{
      margin-top: 5px;
    }
    .info{
      color: #555;
      display: block;
      padding-bottom: 10px;
      font-size: 12px;
    }
  </style>
  <title>Formulário de edição</title>
</head>
<body>

  <?php
    $id = $_GET['id'];
    $name = $_GET['name'];
    $description = $_GET['description'];
  ?>

  <h1>Editar Produtos</h1>

  <p class="info">* O preenchimento de todos os campos é obrigatório</p>

  <form action="../Controller/Update.php" method="post">
    <input type="hidden" size="3" name="id" id="id" value="<?php echo $id; ?>">

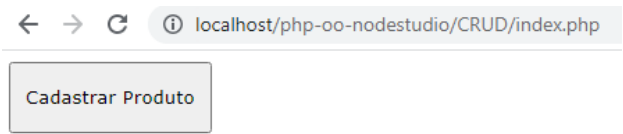
    <label for="name">Nome do produto:</label>
    <input type="text" size="60" name="name" id="name" value="<?php echo $name; ?>">

    <label for="description">Descrição do produto:</label>
    <textarea name="description" id="description" rows="10" cols=60><?php echo
$description; ?></textarea>
    <button type="submit">Atualizar</button>
  </form>

</body>
</html>
```

Exibindo os produtos cadastrados

http://localhost/php-oo-nodestudio/CRUD/index.php



Produtos Cadastrados

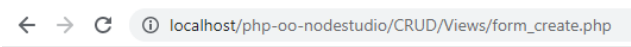
Nenhum produto cadastrado

Cadastrando produtos

Nome: Notebook Dell Inspiron i15-3501

Descrição: Com o notebook Dell Inspiron I15-3501-A46S você aproveita ao máximo o seu dia. Ideal para as tarefas do dia a dia, ele conta com memória e armazenamento pensados para o dia a dia: alterne entre aplicativos abertos com 8 GB de memória. O SSD de 256GB proporciona maior tempo de bateria, rápida resposta e performance mais silenciosa.

A tela de 15.6 polegadas, antirreflexo de alta definição com dois lados de bordas finas, oferece uma imagem clara e brilhante que é agradável aos olhos. Mobilidade e design: mais leve e fácil de levar com você para todos os lugares, seu exterior elegante faz dele um item indispensável no seu dia a dia.



Cadastrar Produto

* O preenchimento de todos os campos é obrigatório

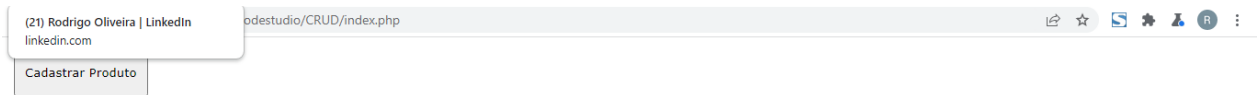
Nome do produto:

Text input field containing: Notebook Dell Inspiron i15-3501

Descrição do produto:

Text area containing: Com o notebook Dell Inspiron I15-3501-A46S você aproveita ao máximo o seu dia. Ideal para as tarefas do dia a dia, ele conta com memória e armazenamento pensados para o dia a dia: alterne entre aplicativos abertos com 8 GB de memória. O SSD de 256GB proporciona maior tempo de bateria, rápida resposta e performance mais silenciosa. A tela de 15.6 polegadas, antirreflexo de alta definição com dois lados de bordas finas, oferece uma imagem clara e brilhante que é agradável aos olhos. Mobilidade e design: mais leve e fácil de levar com você para todos os lugares, seu exterior elegante faz dele um item indispensável no seu dia a dia.

Button labeled: Cadastrar



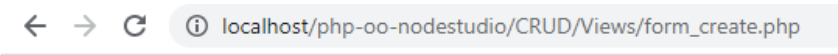
Produtos Cadastrados

Id	Nome	Descrição	Ação
1	Notebook Dell Inspiron i15-3501	Com o notebook Dell Inspiron I15-3501-A46S você aproveita ao máximo o seu dia. Ideal para as tarefas do dia a dia, ele conta com memória e armazenamento pensados para o dia a dia: alterne entre aplicativos abertos com 8 GB de memória. O SSD de 256GB proporciona maior tempo de bateria, rápida resposta e performance mais silenciosa. A tela de 15.6 polegadas, antirreflexo de alta definição com dois lados de bordas finas, oferece uma imagem clara e brilhante que é agradável aos olhos. Mobilidade e design: mais leve e fácil de levar com você para todos os lugares, seu exterior elegante faz dele um item indispensável no seu dia a dia.	<div>Editar</div> <div>Excluir</div>

Nome: Mouse Sem Fio - Multilaser MO251

Descrição: Mouse com um design anatômico, com um alcance de 10 metros com plug and play e com um super dpi de 1200. Informações;- tecnologia 2.4Ghz- design anatômico- alcance: 10 metros- alimentação: 2 pilhas AAA (não inclusas)imagens meramente ilustrativas. Para um bom funcionamento do produto, certifique-se de que a bateria esteja inserida corretamente. Caso o computador não reconheça o mouse, tente uma nova entrada USB. Se você estiver usando um mouse wireless próximo, isto pode gerar alguma interferência e não funcionar adequadamente. Caso esteja desligar o mouse próximos para verificar se está gerando alguma interferência no funcionamento.

http://localhost/php-oo-nodestudio/CRUD/Views/form_create.php



Cadastrar Produto

* O preenchimento de todos os campos é obrigatório

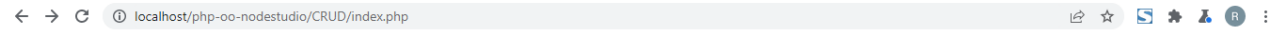
Nome do produto:

Mouse Sem Fio - Multilaser MO251

Descrição do produto:

Mouse com um design anatômico, com um alcance de 10 metros com plug and play e com um super dpi de 1200. Informações;- tecnologia 2.4Ghz- design anatômico- alcance: 10 metros- alimentação: 2 pilhas AAA (não inclusas)imagens meramente ilustrativas. Para um bom funcionamento do produto, certifique-se de que a bateria esteja inserida corretamente. Caso o computador não reconheça o mouse, tente uma nova entrada USB. Se você estiver usando um mouse wireless próximo, isto pode gerar alguma interferência e não funcionar adequadamente. Caso esteja desligar o mouse próximos para verificar se está gerando alguma interferência no funcionamento.

Cadastrar



Cadastrar Produto

Produtos Cadastrados

Id	Nome	Descrição	Ação
1	Notebook Dell Inspiron i15-3501	Com o notebook Dell Inspiron I15-3501-A46S você aproveita ao máximo o seu dia. Ideal para as tarefas do dia a dia, ele conta com memória e armazenamento pensados para o dia a dia: alterne entre aplicativos abertos com 8 GB de memória. O SSD de 256GB proporciona maior tempo de bateria, rápida resposta e performance mais silenciosa. A tela de 15.6 polegadas, antirreflexo de alta definição com dois lados de bordas finas, oferece uma imagem clara e brilhante que é agradável aos olhos. Mobilidade e design: mais leve e fácil de levar com você para todos os lugares, seu exterior elegante faz dele um item indispensável no seu dia a dia.	<div>Editar</div> <div>Excluir</div>
2	Mouse Sem Fio - Multilaser MO251	Mouse com um design anatômico, com um alcance de 10 metros com plug and play e com um super dpi de 1200. Informações;- tecnologia 2.4Ghz- design anatômico- alcance: 10 metros- alimentação: 2 pilhas AAA (não inclusas)imagens meramente ilustrativas. Para um bom funcionamento do produto, certifique-se de que a bateria esteja inserida corretamente. Caso o computador não reconheça o mouse, tente uma nova entrada USB. Se você estiver usando um mouse wireless próximo, isto pode gerar alguma interferência e não funcionar adequadamente. Caso esteja desligar o mouse próximos para verificar se está gerando alguma interferência no funcionamento.	<div>Editar</div> <div>Excluir</div>

Editando os dados de um produto

localhost/php-oo-nodestudio/CRUD/Views/form_edit.php?id=

Editar Produtos

* O preenchimento de todos os campos é obrigatório

Nome do produto:

Mouse Sem Fio - Multilaser MO251

Descrição do produto:

Mouse com um design anatômico, com um alcance de 10 metros com plug and play e com um super dpi de 1200. Informações;- tecnologia 2.4Ghz- design anatômico- alcance: 10 metros- alimentação: 2 pilhas AAA (não inclusas)imagens meramente ilustrativas.Para um bom funcionamento do produto, certifique-se de que a bateria esteja inserida corretamente. Caso o computador não reconheça o mouse, tente uma nova entrada USB. Se você estiver usando um mouse wireless próximo, isto pode gerar alguma interferência e não funcionar adequadamente. Caso esteja desligar o mouse próximos para verificar se está gerando alguma interferência no funcionamento.

Enviar

Editar Produtos

* O preenchimento de todos os campos é obrigatório

Nome do produto:

Mouse Sem Fio - Multilaser MO251

Descrição do produto:

Mouse com um design anatômico, com um alcance de 10 metros com plug and play e com um super dpi de 1200. Informações;- tecnologia 2.4Ghz- design anatômico- alcance: 10 metros- alimentação: 2 pilhas AAA (não inclusas).

Atualizar

localhost/php-oo-nodestudio/CRUD/index.php

Cadastrar Produto

Produtos Cadastrados

Id	Nome	Descrição	Ação
1	Notebook Dell Inspiron i15-3501	Com o notebook Dell Inspiron i15-3501-A46S você aproveita ao máximo o seu dia. Ideal para as tarefas do dia a dia, ele conta com memória e armazenamento pensados para o dia a dia: alterne entre aplicativos abertos com 8 GB de memória. O SSD de 256GB proporciona maior tempo de bateria, rápida resposta e performance mais silenciosa. A tela de 15.6 polegadas, antirreflexo de alta definição com dois lados de bordas finas, oferece uma imagem clara e brilhante que é agradável aos olhos. Mobilidade e design: mais leve e fácil de levar com você para todos os lugares, seu exterior elegante faz dele um item indispensável no seu dia a dia.	<div>Editar</div> <div>Excluir</div>
2	Mouse Sem Fio - Multilaser MO251	Mouse com um design anatômico, com um alcance de 10 metros com plug and play e com um super dpi de 1200. Informações;- tecnologia 2.4Ghz- design anatômico- alcance: 10 metros- alimentação: 2 pilhas AAA (não inclusas).	<div>Editar</div> <div>Excluir</div>

Excluindo um produto cadastrado

- Excluindo o mouse (id: 2)

← → ↻ ⓘ localhost/php-oo-nodestudio/CRUD/index.php

Cadastrar Produto

Produtos Cadastrados

Id	Nome	Descrição	Ação
1	Notebook Dell Inspiron I15-3501	Com o notebook Dell Inspiron I15-3501-A46S você aproveita ao máximo o seu dia. Ideal para as tarefas do dia a dia, ele conta com memória e armazenamento pensados para o dia a dia: alterne entre aplicativos abertos com 8 GB de memória. O SSD de 256GB proporciona maior tempo de bateria, rápida resposta e performance mais silenciosa. A tela de 15.6 polegadas, antirreflexo de alta definição com dois lados de bordas finas, oferece uma imagem clara e brilhante que é agradável aos olhos. Mobilidade e design: mais leve e fácil de levar com você para todos os lugares, seu exterior elegante faz dele um item indispensável no seu dia a dia.	<div>Editar</div> <div>Excluir</div>