

# MVC - Parte 2

## Pasta generic: Autoload.php

Esse código implementa o autoloading de classes no PHP de uma maneira muito simples.

A função dele é carregar automaticamente o arquivo de uma classe quando ela é usada pela primeira vez, eliminando a necessidade de fazer include ou require manualmente para cada arquivo de classe.

## Pasta generic: Autoload.php

```
1  <?php
2  spl_autoload_register(function($class){
3
4      include $class.".php";
5
6  });
```

# Acao.php

```
Humberto Patrick Lacerda Ribeiro, 5 months ago | 1 author (Humberto Patrick  
<?php  
namespace generic;  
  
Humberto Patrick Lacerda Ribeiro, 5 months ago | 1 author (Humberto Patrick  
class Acao{  
    private $classe;  
    private $metodo;  
  
    public function __construct($classe,$metodo)  
    {  
        $this->classe = "controller\\".$classe;  
        $this->metodo = $metodo;  
    }  
  
    public function executar(){  
        $obj = new $this->classe();  
        $obj->{$this->metodo}();  
    }  
}
```

Humberto Patrick Lacerda Ribeiro, 5 months ago

# MysqlFactory

You, last week | 2 authors (Humberto Patrick Lacerda Ribeiro and one other)

```
<?php
```

```
namespace generic;
```

You, last week | 2 authors (Humberto Patrick Lacerda Ribeiro and one other)

```
class MysqlFactory{
```

```
    public MysqlSingleton $banco;
```

```
    public function __construct()
```

```
    {
```

```
        $this->banco=MysqlSingleton::getInstance();
```

```
    }
```

```
}
```

# MysqlSingleton

```
class MysqlSingleton{
    private $usuario = 'root';
    private $senha = '';

    private function __construct(){
        if($this->conexao == null){
            $this->conexao = new PDO($this->dsn,$this->usuario,$this->senha);
        }
    }

    public static function getInstance(){
        if(self::$instance == null){
            self::$instance = new MysqlSingleton();
        }

        return self::$instance;
    }

    public function executar($query,$param = array()){
        if($this->conexao){
            $sth = $this->conexao->prepare($query);
            foreach($param as $k => $v){
                $sth->bindValue($k,$v);
            }

            $sth->execute();
            return $sth->fetchAll(PDO::FETCH_ASSOC);
        }
    }
}
```

# Controller

```
<?php
namespace generic;
Humberto Patrick Lacerda Ribeiro, 5 months ago | 1 author (Humberto Patrick Lacerda Ribeiro)
class Controller{
    private $arrChamadas=[];
    public function __construct()
    {
        $this->arrChamadas = [
            "cliente/lista" =>new Acao("Cliente","listar"),
            "cliente/formulario" =>new Acao("Cliente","formulario"),
            "cliente/formularioalterar" =>new Acao("Cliente","alterarForm"),
            "cliente/inserir" =>new Acao("Cliente","inserir")
        ];
    }

    public function verificarChamadas($rota){

        if(isset($this->arrChamadas[$rota])){
            //acao da rota
            $acao = $this->arrChamadas[$rota];
            $acao->executar();
            return ;
        }

        echo "Rota não existe!";
    }
}
```

Humberto Patrick Lacerda Ribeiro, 5 months ago • primeira versao

## Raiz: Index.php

```
<?php
include "generic/Autoload.php";

use generic\Controller;

if (isset($_GET["param"])) {
    $controller = new Controller();
    $controller->verificarChamadas($_GET["param"]);
}

| Ctd+L to chat Ctd+K to generate
```



DAO

# IClienteDAO

```
<?php      Humberto Patrick Lacerda Ribeiro, 5 months ago • primeira
namespace dao;
Humberto Patrick Lacerda Ribeiro, 5 months ago | 1 author (Humberto Patrick Lacerda Ribeiro)
interface IClienteDAO{
    public function listar();
    public function inserir($nome,$endereco);
    public function listarId($id);
    public function alterar($id,$nome,$endereco);
}
```

# ClienteDAO

```
<?php      Humberto Patrick Lacerda Ribeiro, 5 months ago • primeira ver
namespace dao\mysql;

use dao\IClienteDAO;
use generic\MysqlFactory;

Humberto Patrick Lacerda Ribeiro, 5 months ago | 1 author (Humberto Patrick Lacerda Ribeiro)
class ClienteDAO extends MysqlFactory implements IClienteDAO{

    public function listar(){
        $sql = "select id,nome,endereco from clientes";
        $retorno = $this->banco->executar($sql);
        return $retorno;
    }

    public function listarId($id){
        $sql = "select id,nome,endereco from clientes where id=:id";
        $param = [
            ":id" =>$id
        ];
        $retorno = $this->banco->executar($sql,$param);
        return $retorno;
    }
}
```

## ClienteDAO

```
public function inserir($nome,$endereco){
    $sql = "insert into clientes (nome,endereco) values (:nome,:endereco)";
    $param=[
        ":nome" => $nome,
        ":endereco" =>$endereco
    ];
    $retorno = $this->banco->executar($sql,$param);
    return $retorno;
}

public function alterar($id,$nome,$endereco){
    $sql = "update clientes set nome=:nome,endereco = :endereco whrere id=:id";
    $param=[
        ":nome" => $nome,
        ":endereco" =>$endereco,
        ":id" =>$id
    ];
    $retorno = $this->banco->executar($sql,$param);
    return $retorno;
}
}
```

Service

# ClienteService

```
<?php
namespace service;

use dao\mysql\ClienteDAO;

Humberto Patrick Lacerda Ribeiro, 5 months ago | 1 author (Humberto Patrick Lacerda Ribeiro)
class ClienteService extends ClienteDAO{
    public function listarCliente(){
        return parent::listar();
    }

    public function inserir($nome,$endereco){
        return parent::inserir($nome,$endereco);
    }
    public function alterar($id, $nome, $endereco)
    {
        return parent::alterar($id,$nome,$endereco);
    }
    public function listarId($id)
    {
        return parent::listarId($id);
    }
}
```

Controller

# Cliente

```
<?php

namespace controller;

use service\ClienteService;
use template\ClienteTemp;
use template\ITemplate;

Humberto Patrick Lacerda Ribeiro, 5 months ago | 1 author (Humberto Patrick Lacerda Ribeiro)
class Cliente
{
    private ITemplate $template;
    public function __construct()
    {
        $this->template = new ClienteTemp();
    }

    public function listar()
    {
        $service = new ClienteService();
        $resultado= $service->listarCliente();
        $this->template->layout("\\public\\cliente\\listar.php",$resultado);
    }
}
```



# Cliente

```
public function inserir()
{
    $nome = $_POST["nome"];
    $endereco = $_POST["endereco"];
    $service = new ClienteService();
    $resultado= $service->inserir($nome,$endereco);
    header("location: /mvc20251/cliente/lista?info=1");
}

public function formulario()
{
    $this->template->layout("\\public\\cliente\\form.php");
}

public function alterarForm()
{
    $id=$_GET["id"];
    $service = new ClienteService();
    $resultado =$service->listarId($id);

    $this->template->layout("\\public\\cliente\\form.php",$resultado);
}
}
```

Template

# ITemplate

```
<?php
namespace template;
Humberto Patrick Lacerda Ribeiro, 5 months ago | 1 author (Humberto Patrick Lacerda Ribeiro)
interface ITemplate{
    public function cabecalho();
    public function rodape();
    public function layout($caminho,$parametro =null);
}
Humberto Patrick Lacerda Ribeiro, 5 months ago • concluindo
```

# ClienteTemp

```
<?php
```

```
namespace template;
```

Humberto Patrick Lacerda Ribeiro, 5 months ago | 1 author (Humberto Patrick Lacerda Ribeiro)

```
class ClienteTemp implements ITemplate
```

```
{
```

```
    public function cabecalho()
```

```
    {
```

```
        echo "<div> Cabeçalho </div>";
```

```
    }
```

```
    public function rodape()
```

```
    {
```

```
        echo "<div> Rodapé </div>";
```

```
    }
```

```
    public function layout($caminho, $parametro = null)
```

```
    {
```

```
        $this->cabecalho();
```

```
        include $_SERVER["DOCUMENT_ROOT"]."\mvc20251".$caminho;
```

```
        $this->rodape();
```

```
    }
```

```
}
```

Public

## Listar.php

```
<a href="/mvc20251/cliente/formulario">Cadastrar</a>
<table>
  <tr>
    <th>ID</th>
    <th>Nome</th>
    <th>Endereco</th>
    <th>Ações</th>
  </tr>
  <?php
  foreach( $parametro as $p){
    ?>
    <tr>
      <td><?= $p["id"] ?></td>
      <td><?= $p["nome"] ?></td>
      <td><?= $p["endereco"] ?></td>
      <td><a href="/mvc20251/cliente/formularioalterar?id=<?= $p["id"] ?>">Alterar</a></td>
    </tr>

    <?php
  }
  ?>
</table>
```

## Form.php

```
<form method="POST" action="/mvc20251/cliente/inserir">
    <label>Nome: </label>
    <input type="text" name="nome"
    value="<?= ($parametro != null) ? $parametro[0]["nome"] : "" ?>" />
    <br />
    <label>Endereço: </label>
    <input type="text" name="endereco"
    value="<?= ($parametro != null) ? $parametro[0]["endereco"] : "" ?>" />
    <br />
    <?php
    if ($parametro != null) {
    ?>
        <input type="hidden" name="id" value="<?= $parametro[0]["id"] ?>" />
    <?php
    }
    ?>
    <input type="submit" value="enviar" />
</form>
```