

PHP e Design Patterns

Em **Engenharia de Software**, um padrão de projeto (**Design Pattern**) é uma solução geral para um problema que ocorre com frequência dentro de um determinado contexto no projeto de software. Um padrão de projeto é composto, basicamente de 4 elementos:

- **Nome do Padrão**
- **Problema a ser resolvido**
- **Solução dada pelo padrão**
- **Consequências**

SINGLETON em PHP

- **Problema a ser resolvido:** Permitir apenas uma única instância para a classe. Os objetos são criados apenas uma vez na memória.
- **Solução dada pelo padrão:** Verificar a existência da instância para a classe. Se existir, retorna a instância existente. Se não existir, cria a instância.
- **Consequências:** *Positivas:* Previne stack overflow (estouro de pilha) e sobrecarga da memória. Facilita o gerenciamento do objeto, pois só há uma instância. *Negativas:* Aumenta o acoplamento da classe, não permite o uso de interfaces.

Aplicações

Sempre que houver a necessidade de garantir apenas uma única instância de uma classe para cada execução, o Singleton pode ser aplicado. Em PHP, usa-se muito este padrão para gerar a conexão com o banco de dados.

Implementação em `php`

```
<?php
class Singleton {
    private static $instance; // Armazena a instância
    // Impede o uso do construtor
    private function __construct(){}
    // Impede o clone de instância
    private function __clone(){}
    // Impede a criação da instância via unserialize()
    private function __wakeup(){}
    // Cria o método que cria ou retorna a instância
    public static function getInstance() {
        if(self::$instance === null){
            self::$instance = new self;
        }
        return self::$instance;
    }
}
```

Onde usar Singleton?

Aplicando o padrão Singleton

Podemos dizer que este é um dos projetos mais utilizados em qualquer projeto, isso porque ele previne um possível stack overflow (estouro de pilha), ainda mais em aplicações Web.

Uma aplicação comum são as instâncias para a criação de conexão de banco de dados.

```
private static $instance;  
private static $URL= "mysql:host=127.0.0.1;dbname=pweb";  
private static $user = 'root';  
private static $password = '';  
  
public static function getInstance() {  
    if (!isset(self::$instance)) {  
        self::$instance = new PDO($URL, $user, $password);  
    }  
  
    return self::$instance;  
}
```

Ver classe completa:

<https://github.com/joseolinda/SingletonPDO>

Mais padrões de projeto PHP

Ver outros padrões de projeto implementados em PHP em:

<https://github.com/joseolinda/SingletonPDO>