Etec Zona Leste

Maria Luiza Mendes Rodrigues

Programação Web

Orientação a Objetos em PHP e JavaScript: fundamentos, aplicações e boas práticas no desenvolvimento web.

São Paulo

2025

Maria Luiza Mendes Rodrigues

Programação Web

Orientação a Objetos em PHP e JavaScript: fundamentos, aplicações e boas práticas no desenvolvimento web.

Atividades requisitadas para avaliação de Programação Web 2 do curso de Desenvolvimento de Sistemas.

Orientadora: Palloma Dias Nascimento

São Paulo

2025

Sumário

1. Introdução ................................................................................................................................... 4

2. Desenvolvimento ........................................................................................................................ 5

2.1 Conceitos básicos da Programação Orientada a Objetos ....................................................... 5

2.2 Diferenças e semelhanças entre POO em PHP e em JS ......................................................... 6

2.3 Exemplos de uso prático de POO em aplicações web ........................................................... 7

3. Considerações finais ................................................................................................................... 8

4. Exercício Prático ......................................................................................................................... 9

5. Referências bibliográficas ......................................................................................................... 10

1. Introdução

Na Programação Orientada a Objetos aplicada ao desenvolvimento web, alguns princípios são fundamentais para assegurar a qualidade e a eficiência do código. O primeiro é a **abstração**, que permite modelar elementos do mundo real dentro do sistema. Em seguida, temos o **encapsulamento**, responsável por proteger os dados e métodos de uma classe, restringindo acessos indevidos.

Outro conceito essencial é a **herança**, que possibilita criar novas classes a partir de outras já existentes, favorecendo a reutilização e simplificação da manutenção. Já o **polimorfismo** garante que diferentes classes possam responder de maneira própria a uma mesma operação, trazendo flexibilidade ao código.

Quando aplicados corretamente no desenvolvimento web, esses princípios tornam os sistemas mais organizados, escaláveis e de fácil manutenção, contribuindo diretamente para a eficiência e a robustez do software.

1. Desenvolvimento

2.1 Conceitos básicos da Programação Orientada a Objetos

Classe e Objeto  
Uma classe é uma forma de definir um novo tipo de dado em uma linguagem orientada a objetos. Ela reúne atributos (dados) e métodos (comportamentos). Depois que a classe é criada, podem ser instanciados diversos objetos baseados nela. Por exemplo, a classe Empresa pode ter como atributos nome, endereço, CNPJ, data de fundação e faturamento, além do método imprimir, que exibe as informações da empresa.

Herança  
A herança estabelece um relacionamento em que uma classe “é um” tipo de outra. Por exemplo, a classe Funcionário é uma Pessoa, logo Funcionário herda características de Pessoa. Em algumas linguagens, como C#, existe a herança múltipla, permitindo que uma classe herde de várias outras. Já em Java isso não é possível, pois cada classe só pode herdar de uma única classe.

Encapsulamento  
O encapsulamento consiste em restringir o acesso direto aos atributos de uma classe. Isso ocorre por dois motivos principais:

1. Evitar uso incorreto: por exemplo, em uma classe com um método de divisão, se o atributo divisor fosse acessado livremente, poderia receber o valor zero e causar erros. Com o encapsulamento, isso pode ser evitado por meio de métodos get e set com validações.
2. Manter a lógica interna protegida: em uma classe como Conta, pode ser mais seguro não permitir acesso direto ao atributo saldo, nem mesmo com get e set. Em vez disso, o acesso deve ocorrer apenas por métodos específicos, como saque, depósito e consultar saldo.

Polimorfismo  
O polimorfismo permite que, dentro de uma hierarquia de classes, métodos com a mesma assinatura sejam implementados de formas diferentes. Isso possibilita escrever um código genérico que funcione para várias classes sem a necessidade de múltiplas versões específicas. O principal objetivo do polimorfismo é reduzir a redundância de código, tornando o sistema mais claro e de fácil manutenção.

2.2 Diferenças e semelhanças entre POO em PHP e em JavaScript

A Programação Orientada a Objetos em PHP e em JavaScript compartilha várias semelhanças. Ambas as linguagens oferecem suporte aos conceitos fundamentais da POO, como classes, objetos, herança, encapsulamento e polimorfismo. Isso permite representar entidades do mundo real por meio de atributos e métodos, garantindo organização, reutilização e modularidade no desenvolvimento de aplicações web. Em ambos os casos, a aplicação desses princípios torna os sistemas mais claros, escaláveis e de fácil manutenção.

Por outro lado, existem diferenças significativas entre as duas linguagens. O PHP adota o modelo clássico baseado em classes, com suporte nativo para modificadores de acesso como *public*, *private* e *protected*. Já o JavaScript tem origem em um paradigma baseado em protótipos e apenas com o ES6 passou a oferecer a sintaxe de classes, que funciona como uma forma simplificada para os desenvolvedores, mas que internamente continua sendo prototípica. Além disso, o encapsulamento em JavaScript foi durante muito tempo feito por convenções ou closures, e só mais recentemente ganhou suporte a atributos privados com o símbolo #.

Outra diferença importante está no ambiente de execução. Enquanto o PHP é amplamente utilizado no lado do servidor, cuidando de tarefas como autenticação, integração com banco de dados e geração dinâmica de páginas, o JavaScript atua principalmente no lado do cliente, sendo responsável pela interatividade na interface. Entretanto, com o uso do Node.js, também é possível executar JavaScript no servidor, ampliando sua aplicação no desenvolvimento web.

2.3 Exemplos de uso prático de POO em aplicações web

Exemplo aplicado ao tema Música: modelagem de um sistema de catálogo musical. Classes possíveis: Artista, Álbum, Faixa, Usuário. Cada classe encapsula atributos relevantes (nome, ano, duração, gênero) e métodos para manipulação (ex.: reproduzir, pausar, buscar por gênero). Abaixo, um resumo de exemplo prático em PHP (aplicação entregue separadamente em arquivos PHP):

- Classe Usuário (base) com atributos id, nome, e-mail e métodos para login e logout.  
- Classe Cliente (herda Usuário) com comportamento adicional como criar playlists.  
- Classe Administrador (herda Usuário) com métodos para gerenciar catálogo (adicionar/editar/remover faixas e álbuns).

1. Considerações finais

Para iniciantes, a programação orientada a objetos oferece uma maneira mais amigável de entender como os diferentes componentes de um sistema interagem, criando uma base sólida.

À medida que os desenvolvedores ganham experiência, a programação orientada a objetos permite a criação de sistemas complexos de forma mais gerenciável. Isso facilita a escalabilidade à medida que os projetos crescem.

1. Exercício Prático

Tela de computador com fundo preto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto. Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

1. Referências bibliográficas

<https://3t.dev.br/programacao-orientada-a-objetos-aplicada-no-desenvolvimento-web/>

<https://www.devmedia.com.br/principais-conceitos-da-programacao-orientada-a-objetos/32285>

<https://wiki.ifsc.edu.br/mediawiki/images/4/4f/ApostilaPOO.pdf>

<https://www.dio.me/articles/a-importancia-da-programacao-orientada-a-objetos-para-desenvolvedores-de-todos-os-niveis>